

# Parte I

## Ottimizzazione nei sistemi di controllo I







# Indice

<b>I Ottimizzazione nei sistemi di controllo I</b>	<b>1</b>
<b>1 Ottimizzazione Dinamica</b>	<b>9</b>
1.1 Introduzione	9
1.2 Loot-Sizing	9
1.3 Controllo Ottimo	10
1.4 Programmazione Dinamica	11
1.4.1 Principio di Ottimalità	11
1.4.2 Problema cammino minimo ( SPP )	11
1.4.3 Interpretazione di SPP con DP	12
<b>2 Equazioni di Bellman</b>	<b>15</b>
2.1 Problema di Knapsak	15
2.2 Formulazione Generale	15
2.2.1 Processo decisionale multi-stage	15
2.2.2 Funzione valore	16
2.2.3 Equazione di Bellman	16
2.2.4 Soluzione Ricorsiva Equazione Di Bellman	16
2.2.5 Lot-Sizing DP	17
<b>3 Controllo Ottimo Lineare Quadratico a tempo discreto e MPC</b>	<b>19</b>
3.1 Introduzione	19
3.2 Controllo Ottimo Lineare Quadratico di sistemi a tempo discreto	19
Soluzione a “batch” di ottimizzazione dinamica LQ a tempo discreto	21
Soluzione Batch vs Equazione di Riccati	22
3.3 Model Predictive Control MPC	22
Introduzione al Model Predictive Control	22
IDEA	22
STRATEGIA DI MPC	22
3.3.1 MPC VINCOLATO	24
<b>Definizione 3.13.</b> <i>Model Predictive Controllo in presenza di vincoli</i>	24
3.3.2 Stabilità e Ammissibilità di MPC	24
Vincolo di stabilità per MPC	24
3.3.3 Ottimizzazione On-line vs Ottimizzazione Off-line	25
QP multi-parametrico mp-QP	25
mp-QP per MPC esplicito	25
Struttura della soluzione ottima di mp-QP <sub>z</sub>	26
Regione Critica CR <sub>0</sub>	26
Esplorazione di X\CR <sub>0</sub>	27
<b>4 Reinforcement Learning</b>	<b>29</b>
4.1 Introduzione	29
4.1.1 Elementi principali di Reinforcement Learning	29
4.1.2 Schema di Reinforcement Learning	30
Marko Decision Process (MDP)	30

Proprietà Markoviana dello stato . . . . .	30
Policy stazionaria . . . . .	30
4.1.3 Obiettivo di Reinforcement Learning . . . . .	30
4.2 Value Function nel Reinforcement Learning . . . . .	31
4.2.1 Funzione Valore di Stato . . . . .	31
4.2.1.1 Equazione del valore di $v_\pi$ . . . . .	31
4.2.1.2 Soluzione dell'equazione del valore di $v_\pi$ . . . . .	31
4.2.2 Funzione valore di stato/azione . . . . .	32
4.2.3 Funzione Valore Ottima . . . . .	32
4.2.4 Policy ottima . . . . .	32
4.3 Policy e Value iteration . . . . .	33
4.3.1 Policy Evaluation . . . . .	33
4.3.1.1 Approssimazione Iterativa . . . . .	33
4.3.2 Policy Improvement . . . . .	33
4.4 Value Iteration . . . . .	34
4.5 Programmazione Dinamica Asincrona . . . . .	35
4.6 Processi Decisionali di Markov Stocastici . . . . .	35
4.6.1 Richiamo sulle Variabili Aleatorie . . . . .	35
4.6.2 Processo decisionale di Markov Stocastico . . . . .	36
4.6.3 Confronto Grafico . . . . .	37
4.7 Temporal-difference learning e Q-learning . . . . .	37
4.7.1 Temporal-difference learning . . . . .	37
4.7.2 Monte Carlo vs TD . . . . .	38
4.7.3 On-policy TD:SARSA . . . . .	38
4.7.4 Off-policy TD:Q-learning . . . . .	39
4.7.5 RL con approssimatori funzionali: Deep Reinforcement Learning . . . . .	39
<b>5 Riepilogo Teoria dei Sistemi . . . . .</b>	<b>41</b>
5.1 Sistema Dinamico Lineare . . . . .	41
5.1.1 Sistema Lineare a Tempo Continuo . . . . .	41
5.2 Proprietà di Stabilità ( con $u=0$ ) . . . . .	41
5.3 Teoria di Lyapunov . . . . .	42
5.3.1 Metodo diretto di Lyapunov . . . . .	42
5.3.2 Raggiungibilità e Controllabilità . . . . .	43
5.3.2.1 Insieme degli stati raggiungibili . . . . .	43
5.3.2.2 Assegnazione degli Autovalori . . . . .	43
5.3.2.3 Algoritmo di Mitter . . . . .	43
5.3.3 Proprietà di Osservabilità . . . . .	45
<b>6 Principio di Ottimalità e Equazione di Hamilton-Jacobi-Bellman . . . . .</b>	<b>47</b>
6.1 Introduzione . . . . .	47
6.2 Definizione di un Problema di Controllo Ottimo . . . . .	47
6.3 Principio di Ottimalità . . . . .	48
6.4 Commenti sul Principio di Ottimalità . . . . .	48
6.4.1 Analisi . . . . .	48
6.4.2 Sintesi . . . . .	49
6.5 Funzione Valore di un Problema di Controllo Ottimo . . . . .	49
6.5.1 Funzione Valore . . . . .	49
6.6 Condizioni Necessarie di Ottimalità - Equazione HJB . . . . .	49
6.6.0.1 Riepilogo Espansione di Taylor . . . . .	50
6.6.1 Equazione di Hamilton-Jacobi-Bellman . . . . .	50
6.7 Condizioni Sufficienti di Ottimalità . . . . .	50
6.8 Problema di Controllo Ottimo in presenza di vincoli sul controllo . . . . .	52
6.9 Problema di Controllo Ottimo ad Orizzonte Finito . . . . .	52
<b>7 Problema del Regolatore Lineare-Quadratico (LQR) . . . . .</b>	<b>53</b>

7.1	Problema del Regolatore Lineare-Quadratico (LQR)	53
7.2	Minimizzazione dell'equazione di HJB	53
7.3	Calcolo della Funzione Valore	53
7.4	Equazione Differenziale di Riccati	54
7.5	Esistenza Globale della Soluzione DRE	54
7.6	Considerazioni Computazionali e Sistema Hamiltoniano	55
<b>8</b>	<b>Matrice Hamiltoniana e Soluzione Dell'Equazione differenziale di Riccati</b>	<b>57</b>
8.1	Proprietà della matrice Hamiltoniana	57
8.2	Riepilogo sulla forma di Jordan	57
8.3	Soluzione di DRE tramite matrice Hamiltoniana	58
8.4	Modelo Macro-Economico Nuovo Keynesiano	59
8.4.1	Modello Nuovo-Keynesiano	59
<b>9</b>	<b>Regolatore Lineare-Quadratico su Orizzonte Infinito</b>	<b>61</b>
9.1	LQR su Orizzonte Infinito	61
9.2	Condizioni Sufficienti di Esistenza	61
9.2.1	Esistenza del limite della soluzione su orizzonte finito	61
9.2.2	Unicità della soluzione (stabilizzante) di ARE	63
9.2.3	Ottimalità della Soluzione di ARE	63
9.2.4	Stabilità del sistema a ciclo chiuso	64
<b>10</b>	<b>Algoritmo di Kleinman</b>	<b>67</b>
10.1	Approccio Risolutivo di HJB su orizzonte infinito	67
10.2	Algoritmo di Kleinman	67
<b>11</b>	<b>Tracking e Reiezione di Disturbi</b>	<b>69</b>
11.1	Problema di Tracking/Asservimento	69
11.2	Reiezione dei disturbi noti	70
11.3	Equazione di HJB per reiezione dei disturbi	70
11.4	Forma della funzione valore	70
11.5	Soluzione dell'equazione di HJB	70
11.6	Commenti sulla soluzione di HJB	71





# Capitolo 1

## Ottimizzazione Dinamica

### 1.1 Introduzione

Il problema di ottimizzazione è un problema di massimizzazione e/o minimizzazione in cui le variabili decisionale e o altre parametri variano nel tempo:

- per cause esterne;
- in conseguenza a scelte precedenti istantanee o non banali.

L'ottimizzazione dinamica è quindi caratterizzata da:

- funzione obiettivo cumulativo di costi o ricavi istantanei;
- vincoli dinamici.

Le tecniche di ottimizzazione dinamica sono di due tipo:

- controllo ottimo;
- programmazione dinamica;

L'obiettivo di entrambe le tecniche è quello di selezionare una legge di controllo in feedback al fine di minimizzare un costo.

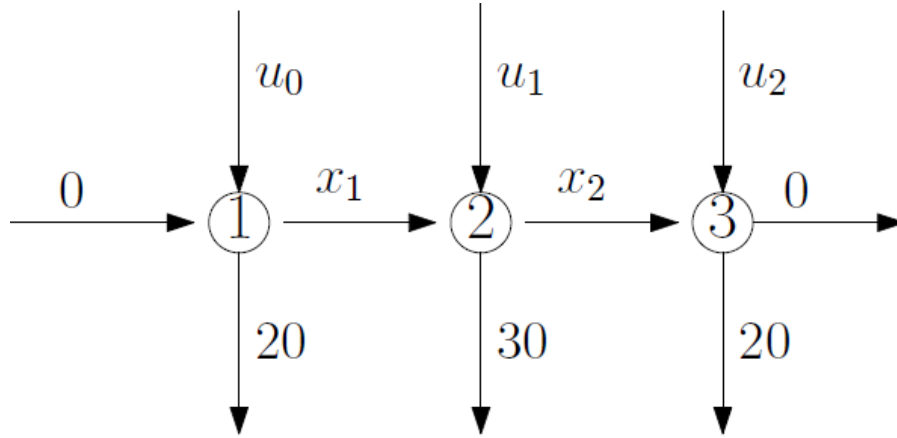
### 1.2 Loot-Sizing

Un esempio di ottimizzazione dinamica è quello del loot-sizing. Si vuole decidere quando e quanto produrre per soddisfare una domanda.

Assumiamo:

- costi fissi di produzione  $\rightarrow$  pochi lotti grandi;
- costi di immagazzinamento  $\rightarrow$  lotti piccoli quanto serve;

esempio:  $N=3$



$$\begin{cases} \min_{u_0, u_1, u_2} \sum_{i=0}^N (p_k u_k + A_k \eta(u_k) + h_k x_k) \\ 0 \leq u_k \leq M \quad k=0, 1, 2 \\ 0 \leq x_k \quad k=1, 2, 3 \\ \begin{cases} x_1 = u_0 - 20 \\ x_2 = x_1 + u_1 - 30 \\ 0 = x_2 + u_2 - 20 \end{cases} \text{ Rappresentazione esplicita} \end{cases}$$

In cui:

$p_k$  := costo di produzione al tempo  $k$

$h_k$  := costo di immagazzinamento

$A_k$  := costo di produzione

$M$  := limite di produzione

$$\eta(u_k) = \begin{cases} 0 & u_k = 0 \\ 1 & u_k > 0 \end{cases}$$

La **rappresentazione esplicita** dei **vincoli** permette di riassumere le conseguenze delle scelte precedenti. Tuttavia, maggiore sono le variabili del nostro problema, maggiori sono il numero di vincoli espliciti che dovremmo formulare, portando ad un aumento di difficoltà computazionale non indifferente.

Per risolvere questa problematica, occorre utilizzare la **rappresentazione implicita** dei vincoli. Da cui otteniamo:

$$x_{k+1} = x_k + u_k - d_k \quad x_0 = 0 \quad k \in \mathbb{N}$$

Quindi **ogni** qual volta in cui i vincoli variano nel tempo occorre utilizzare i **vincoli impliciti**.

### 1.3 Controllo Ottimo

Il **controllo ottimo** è una tecnica di ottimizzazione dinamica in cui si vuole determinare la legge di controllo in feedback  $u(t) = kx(t)$  con  $k \in \mathbb{R}^{m \times n}$  se lineare o  $u(t) = \pi(x(t))$  non lineare.

Questo controllo deve appunto minimizzare o massimizzare un **funzionale** di costo:

$$J(u) = \int_0^t l(x(t), u(t)) dt$$

## 1.4 Programmazione Dinamica

- La programmazione dinamica fornisce condizioni necessarie e sufficienti di ottimalità.
- Di solito sono condizioni ricorsive nei sistemi a tempo discreto.
- Equazioni alle derivate parziali → Equazioni di Hamiltone-Jacobi
- Equazione matriciale quadratica → Equazione di Riccati per sistemi lineari

Quindi la programmazione dinamica è una tecnica algoritmica utile a risolvere problemi di ottimizzazione.

L'idea di fondo è quella di dividere il problema in maniera opportuna in sottoproblemi che possono essere risolti più facilmente e la cui soluzione sia utile alla composizione della soluzione complessiva.

### **N.B.:**

Questa tecnica è necessaria ove si hanno problemi decisionali variabili nel tempo (dinamico) e le decisioni correnti influenzano lo scenario futuro.

### 1.4.1 Principio di Ottimalità

Una politica decisionale ottima ha la proprietà che, quale che sia la configurazione iniziale (**stato**) e le decisioni iniziali (**controllo**), se consideriamo un punto intermedio, le decisioni rimanenti della politica devono costituire una soluzione ottima rispetto alla configurazione raggiunta dalla prima parte delle decisioni.

#### **Oss.:**

Se conosciamo già una soluzione ottima → deve soddisfare il principio di ottimalità

Quindi preso qualsiasi istante intermedio, le decisioni rimanenti che devono essere prese, devono costituire la soluzione ottima del sottoproblema ristretto dal punto intermedio in poi, qualunque esso sia.

### 1.4.2 Problema cammino minimo ( SPP )

#### **Dati:**

- $C = \{1, \dots, n\}$   $n > 1$  nodi
- $D$  con distanza  $D(i, j)$  t.c.:

$$\left\{ \begin{array}{l} \text{è presente al più un legame tra coppie di nodi} \\ -\infty < D(i, j) \leq \infty \quad \forall i, j \in C \\ D(i, j) = \infty \quad \forall i \in C \\ D(i, j) = \infty \quad \text{se non c'è un arco tra } i \text{ e } j \end{array} \right.$$

#### **DEF**

$A(j) := \{i \in C: D(j, i) < \infty\}$   $j \in C$  insieme **successori** immediati di  $j$

$B(j) := \{i \in C: D(i, j) < \infty\}$   $j \in C$  insieme **predecessori** immediati di  $j$

#### **Obiettivo**

Determinare il percorso più breve tra origine e destinazione.

Assumiamo:

- $B(1) = \emptyset \rightarrow$  origine non ha predecessori;
- Possono essere presenti cicli;
- $D(i, j) \geq 0$  distanze non negative

Allora:

Esiste una soluzione ammissibile  $\Rightarrow$  Esiste almeno una **soluzione ottima (aciclica)**

A questo punto definiamo:

$$V(j) := \text{lunghezza del percorso più breve da 1 a } j \in C$$

$$V(1) = 0$$

Obiettivo:

Vogliamo determinare una  $V(n)$

Per risolvere questo problema dobbiamo ricorrere all'**algoritmo di Dijkstra**:

“Si definisce una procedura iterativa che cerca di migliorare una sovrastima  $\bar{V}(j)$  del valore esatto  $V(j)$  per ciascun nodo.”

Questo algoritmo è greedy poiché si vengono processati i nodi solamente 1 volta.

#### Inizializzazione

$$j = 1 \quad \bar{V}(1) = 0 \quad \bar{V}(i) = \infty, \forall i \in \{2, 3, \dots, n\} \quad U = C$$

#### Iterazione

$$\begin{aligned} &\text{while } |U| \geq 1 \text{ e } \bar{V}(j) < \infty \\ &\quad U \leftarrow U \setminus \{j\} && \text{si eliminano } i \text{ nodi da processare} \\ \bar{V}(j) &\leftarrow \min \{ \bar{V}(i), \bar{V}(i) + D(i, j) \} \forall i \in A(j) \cap U && \text{si confronta il costo attuale con quello che passa per } j \\ j &\leftarrow \arg \min_i \{ \bar{V}(i), i \in U \} && \text{Si seleziona il nodo più promettente} \end{aligned}$$

L'algoritmo termina in al più  $n-1$  iterazioni e  $\bar{V}(j) = V(j) \forall j \in C$

[Vedere applicazione algoritmo di Dijkstra sugli appunti cartacei]

### 1.4.3 Interpretazione di SPP con DP

Prima di andare ad applicare la **programmazione dinamica** occorre fornire le seguenti definizioni:

- **Equazione funzionale:** è una funzione che caratterizza la proprietà che deve essere soddisfatta dal problema di ottimizzazione dinamica;
- **Algoritmi:** permette di risolvere l'equazione funzionale.

Applicando l'idea di suddivisione in sottoproblemi e il principio di ottimalità:

$$V(j) = V(i) + D(i, j)$$

Secondo il principio di ottimalità, la distanza minima fino a  $j$  è pari alla minima distanza fino a  $i$  (predecessore di  $j$ ) più la lunghezza da  $i$  a  $j$ .

Quindi:

$$V(j) = \min_{i \in B(j)} \{ V(i) + D(i, j) \}, j \in C \setminus \{1\} \quad V(1) = 0$$

Rappresenta la proprietà che la soluzione ottima che deve essere soddisfatta.

Questa equazione è risolvibile tramite:

- metodi diretti;
- metodi con approssimazione successive.

#### **Metodi diretti**

I metodi diretti consistono nella risoluzione dell'equazione funzionale considerando la funzione di costo  $V(j)$ ,  $j \in C$  viene considerata una incognita.

#### **Metodi per approssimazioni successive**

Si approssima il valore cercato con una **stima**  $\bar{V}(\cdot)$  e si aggiorna ripetutamente migliorando la stima fino a quando non si converge ad un **punto fisso**. Definiamo un **punto fisso** come una funzione su cui si applica l'aggiornamento e si ottiene la stessa stima.

Esistono 2 filosofie di aggiornamento:

- **PULL** : si aggiorna il valore di  $j$  sulla base della stima del miglior percorso attualmente conosciuto per raggiungerlo.

$$\text{PULL } a.j: \bar{V}(j) \leftarrow \min_{i \in B(j)} \{ \bar{V}(i) + D(i, j) \}, j \in C, B(j) \neq \emptyset$$

- **PUSH:** si propaga un miglioramento della stima per raggiunge  $j$  a tutti i suoi successori.

$$\text{PUSH } aj: \bar{V}(j) \leftarrow \min \{V(i), \bar{V}(i) + D(j, i)\} \ i \in A(j)$$

In sintesi, nella politica **PULL** se  $\bar{V}(j)$  migliora, si processano i suoi successori  $Q = Q \cup A(j)$ ; nella politica **PUSH** se qualche  $i \in A(j)$  migliora la sua stima, processiamo anche i  $Q = Q \cup i$ .



# Capitolo 2

## Equazioni di Bellman

### 2.1 Problema di Knapsack

- Consideriamo uno zainetto di capacità  $k$ ;
- Possiamo riempire lo zaino con un numero intero di oggetti  $x_i$   $i = 1, \dots, N$ ;
- Ogni oggetto ha volume  $v_i$  e valore  $c_i$ ;

**Obiettivo:** massimizzare il valore dello zainetto.

$$P(N, K) = \begin{cases} \max \sum_{i=1}^N c_i x_i \\ \sum_{i=1}^N v_i x_i \leq k \\ x_i \geq 0 \quad i = 1, \dots, N \\ x_i \in \mathbb{N} \quad i = 1, \dots, N \end{cases}$$

La soluzione algoritmica prevede di risolvere  $NK$  sottoproblemi  $P(r, \lambda)$  con  $r \leq N$  oggetti e volume  $\lambda \leq K$ .

Per formulare il problema con **programmazione dinamica**(DP) si introduce:

$y :=$  volume libero dello zainetto

Inizialmente:

$y_o = K$  (zaino vuoto con tutto lo spazio a disposizione)

$y_{k+1} = y_k - v_k$  (a patto che  $v_k \leq y_k$ )

Definiamo:

$V(y) :=$  **valore massimo ottenibile** dato volume  $y$  ( $V(0) = 0$ )

$$V(y) = \max_{v_k \in y, k \in \{1, \dots, N\}} \{c_k + V(y - v_k)\} \quad \forall y \in \{1, \dots, K\}$$

Vogliamo massimizzare il valore aggiungendo un oggetto alla volta  $c_k$ , ma considerando anche l'effetto futuro della scelta.

[Pagina 8 appunti cartacei Esempio applicazione DP Knapstack]

### 2.2 Formulazione Generale

#### 2.2.1 Processo decisionale multi-stage

$$\begin{cases} \min_{u_1, \dots, u_{N-1}} \sum_{k=0}^{N-1} g_k(x_k, u_k) + g_N(x_N) \\ x_{k+1} = f(x_k, u_k) \quad k = 0, 1, \dots, N-1 \\ u_k \in U_k \\ x_k \in X \end{cases}$$

In cui:

- $k :=$  indice temporale;
- $N :=$  orizzonte temporale;
- $x_k :=$  stato del sistema;

- $u_k :=$  controllo da selezionare;
- $g_k :=$  costo corrente;  $g_N :=$  costo terminale;
- $U_k :=$  insieme controlli ammissibili al tempo  $k$ ;
- $X :=$  insieme di cardinalità finita;

### 2.2.2 Funzione valore

$V_k(v_k) :=$  è il costo migliore che possiamo ottenere dall passo  $k$  fino ad  $N$  data la condizione iniziale  $x_k$  al tempo  $k$ .

### 2.2.3 Equazione di Bellman

$$\begin{cases} v_k(x_k) = \min_{u_k \in U_k} \{g_k(x_k, u_k) + V_{k+1}(f(x_k, u_k))\} & \forall x_k \in X \quad k = \{0, 1, \dots, N\} \\ V_N(x_N) = g_N(x_N) & \forall x_N \in X \end{cases}$$

Questa equazione si risolve ricorsivamente all'indietro, cioè si parte dal punto finale e tornando indietro, si ricostruiscono gli ingressi ottimali. In altre parole:

Dopo aver calcolato il valore di lungo periodo, l'**azione ottima** è quella che trasferisce il sistema nello stato a maggior valore:

$$u_k^*(x_k) = \arg \min_{u_k \in U_k} \{g_k(x_k, u_k) + V_{k+1}(x_{k+1})\}$$

### 2.2.4 Soluzione Ricorsiva Equazione Di Bellman

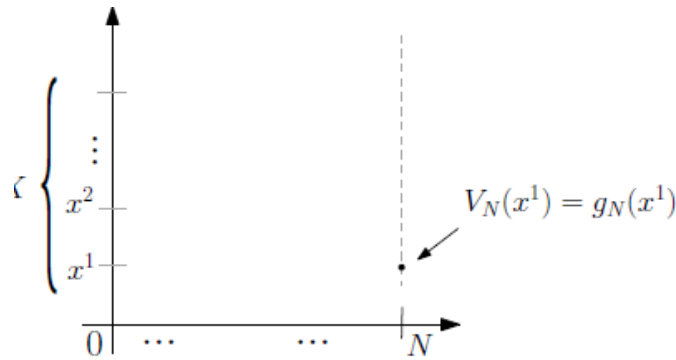


Figura 2.1.

**Consideriamo l'istante terminale  $N$** , il costo migliore a partire da  $x^1$  al tempo  $N$  è pari al costo terminale  $g_N$  valutato in  $x^1$ . Possiamo ora assegnare il costo ottimo al tempo  $N$  a ciascuno stato in  $X$ , calcolando  $V_N(\cdot)$ .

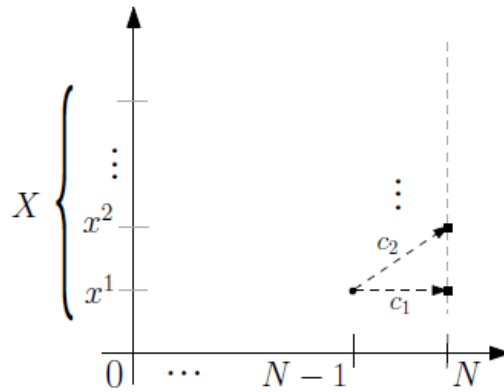


Figura 2.2.

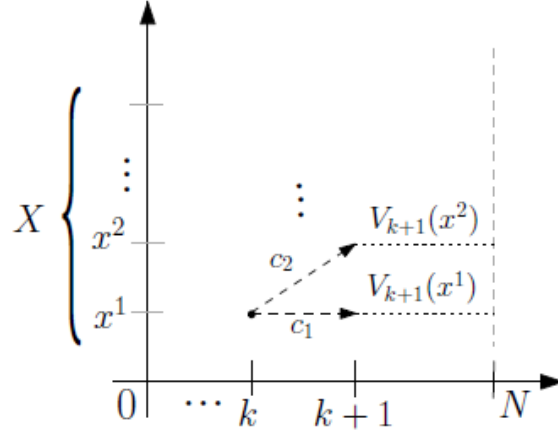


**Facendo un passo indietro a N-1**, il costo della scelta 1 è  $c_1 + V_N(x^1)$ .

Conosciamo tutti i costi  $V_N(x^i)$ , quindi possiamo determinare la migliore scelta da  $x^1$  al tempo N-1:

$$V_{N-1}(x^1) = \min_i \{c_i + V_N(x^i)\}$$

**Al generico istante k:**



**Figura 2.3.**

Per il principio di ottimalità, la coda della soluzione ottima da k deve necessariamente coincidere con  $V_{k+1}(x^1)$ . **Scelte greedy basate su considerazioni istantanee rispetto ad una funzione che racchiude anche le conseguenze future.**

### 2.2.5 Lot-Sizing DP

$$\begin{cases} \min_{u_1, \dots, u_{N-1}} \sum_{k=0}^{N-1} p u_k + A\eta(u_k) + h x_k + g_N(x_N) \\ x_{k+1} = x_k + u_k - d_k \quad k=0, 1, \dots, N-1 \\ u_k \in U_k \\ x_k \geq 0 \end{cases}$$

- **Capacità di immagazzinamento finita**  $M_m$

$$x_k + u_k - d_k \leq M_m \implies u_k \leq M_m + d_k - x_k$$

- **Capacità di produzione finita**  $M_p, u_k \leq M_p$

- **Vincolo di domanda**

$$x_k + u_k - d_k \geq 0 \implies u_k \geq d_k - x_k$$

$$U_k = [d_k - x_k, \min \{M_p, M_m + d_k - x_k\}] \quad X = \{0, \dots, M_m\}$$

- **Equazione di Bellman**

$$V_k(x_k) = \min_{u_k \in U_k} \{p u_k + A\eta(u_k) + h x_k + V_{k+1}(x_{k+1})\}, k=0, 1, \dots, N-1 \quad x_k=0, \dots, M_m$$

$$V_N(x_N) = g_N(x_N) \quad x_N=0, \dots, M_m$$



# Capitolo 3

## Controllo Ottimo Lineare Quadratico a tempo discreto e MPC

### 3.1 Introduzione

**Definizione 3.1.** *Matrici definite e semi-definite*

La matrice  $M \in \mathbb{R}^{n \times n}$ ,  $M = M^T$  è **definita positiva** (**semi-definita positiva**) se:

$$x^T M x > 0 (\geq 0), \forall x \in \mathbb{R}^n$$

Overo la forma quadratica associata è una funzione sempre positiva ( non-negativa ).

**Nota 3.2.** Se  $M$  non è simmetrica:

$$x^T M x = \frac{1}{2} x^T (M + M^T) x = \frac{1}{2} x^T (M + M^T) x + \frac{1}{2} x^T (M - M^T) x$$

Con  $M_s = M + M^T$  e  $M_a = M - M^T$  e

$$\frac{1}{2} x^T M_a x = \frac{1}{2} x^T M x - \frac{1}{2} x^T M^T x = 0$$

Nella forma quadratic "sopravviva solo la parte simmetrica  $M_s$ .

**Definizione 3.3. Criterio di Sylvester**

Supponiamo  $M$  simmetrica:

- $M$  è definita positiva se e solo se:
  - tutti gli autovalori di  $M$  sono positivi;
  - tutti i minori principali dominanti sono positivi;
- $M$  è semi-definita positiva se e solo se:
  - tutti gli autovalori di  $M$  sono non negativi;
  - tutti i minori principali sono non-negativi;

### 3.2 Controllo Ottimo Lineare Quadratico di sistemi a tempo discreto

Consideriamo un indice di costo **quadratico**:

$$J(u) = \sum_{k=k_0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) + x_N^T S x_N$$

$$Q = Q^T \in \mathbb{R}^{n \times n}, Q \succcurlyeq 0, R = R^T \in \mathbb{R}^{m \times m}, R \succ 0, S = S^T \in \mathbb{R}^{n \times n}, S \succcurlyeq 0$$

In presenza di un vincolo dinamico:

$$x_{k+1} = A x_k + B u_k \quad x_{k_0} = \bar{x}_0$$

**sistema lineare a tempo discreto**,  $x_k \in \mathbb{R}^n$  e  $u_k \in \mathbb{R}^m$ ,  $k \in \mathbb{Z}$  e il costo viene parametrizzato da  $k_0, N, \bar{x}_0$  ovvero  $J(k_0, N, \bar{x}_0, u(\cdot))$  da **minimizzare**.

Per il principio di ottimalità:

**Importante 3.4.** Quale che sia lo stato  $x_{N-1}$  raggiunto dal processo al tempo  $N-1$  sotto l'azione del controllo ottimo, il valore di  $u_{N-1}$  (ultimo campione di controllo) che rende minimo l'intero indice di costo  $J$ , **deve essere** tale da ottenere un costo residuo  $\tilde{J}$  nell'intervallo  $[N-1, N]$  **minimo**.

**Definizione 3.5.**

$$\tilde{J} = J(N-1, N, x_{N-1}, u_{N-1}) = x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + x_N^T S x_N$$

Per ricostruire ricorsivamente all'indietro la soluzione ottima, si deve determinare  $u_{N-1}$  che minimizza  $\tilde{J}$  **quale che sia**  $x_{N-1}$ .

**Nota 3.6.**

La condizione necessaria di minimizzazione non vincolata di una funzione  $f(x)$  è:

- $\nabla_x f(x^*) = 0$  Condizione Sufficiente
- Se la matrice **hessiana** è definita positiva, allora la condizione diventa sufficiente.

A questo punto sfruttiamo  $\tilde{J}(u_{N-1})$  e sostituiamo  $x_N$ :

$$\begin{aligned} \tilde{J}(u_{N-1}) &= x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + x_N^T S x_N \\ &= x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + (A x_{N-1} + B u_{N-1})^T S (A x_{N-1} + B u_{N-1}) = \\ &= x_{N-1}^T (Q + A^T S A) x_{N-1} + u_{N-1}^T (R + B^T S B) u_{N-1} + u_{N-1}^T B^T S A x_{N-1} + x_{N-1}^T A^T S B u_{N-1} \end{aligned}$$

Calcoliamo le derivate di  $\tilde{J}$  rispetto a  $u_{N-1}$ :

$$\begin{aligned} \nabla \tilde{J} &= 2(R + B^T S B) u_{N-1} + 2 B^T S A x_{N-1} \\ \nabla^2 \tilde{J} &= 2(R + B^T S B) \end{aligned}$$

Dal momento che  $\nabla^2 \tilde{J} = 2(R + B^T S B) > 0$ , il minimo si ottiene imponendo  $\nabla \tilde{J} = 0$ . Quindi:

$$u_{N-1}^* = -(R + B^T S B)^{-1} B^T S A x_{N-1}$$

è l'unico valore del controllo che rende minimo  $\tilde{J}$  all'istante  $N-1$  in funzione di  $x_{N-1}$ .

Il **valore minimo** del costo si ottiene sostituendo  $u_{N-1}^*$  in  $\tilde{J}$ :

$$\tilde{J} = x_{N-1}^T (Q + A^T S A - A^T S B (R + B^T S B)^{-1} B^T S A) x_{N-1} = x_{N-1}^T P_{N-1} x_{N-1}$$

Il problema di calcolare  $u_{N-2}^*$  può essere risolto similmente prima, ma  $u_{N-1}$  è **già fissato**. Quindi vogliamo minimizzare:

$$\tilde{J}(u_{N-2}) = x_{N-2}^T Q x_{N-2} + u_{N-2}^T R u_{N-2} + x_{N-1}^T P_{N-1} x_{N-1}$$

simile a  $\tilde{J}$  con slittamento dei tempi all'indietro di un campione  $\implies$  stessa soluzione

**Importante 3.7.** Iterando ricorsivamente all'indietro

Il **controllo ottimo** è:

$$u_k = -(R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A x_k$$

Con  $P$  calcolata ricorsivamente all'indietro da  $P_N = S$  e

$$P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

Inoltre, il **costo ottimo** è:

$$J^* = \bar{x}_0^T P_0 \bar{x}_0$$

**Definizione 3.8. Equazione alle differenze di Riccati**

Per  $k$  che va da  $N-1$  a  $0$ :

$$P_k = Q + A^T P_{k+1} A - A^T P_{k+1} B (R + B^T P_{k+1} B)^{-1} B^T P_{k+1} A$$

risolta ricorsivamente all'indietro a partire da:

$$P_N = S$$

**Nota 3.9.**

- L'ipotesi  $Q \succcurlyeq 0, S \succcurlyeq 0$  garantisce che  $(R + B^T P_{k+1} B)$  è **definita positiva** per ogni  $k$ :
  - stiamo effettivamente calcolando un punto di **minimo**;
  - garantisce anche che  $P_k \succcurlyeq 0$ , in quanto  $x^T P x$  descrive il valore del costo ottimo da  $x$  nell'intervallo  $[k, N]$
- L'ipotesi di cui sopra può essere rilassata a patto di verificare la condizione per ogni  $k$ .

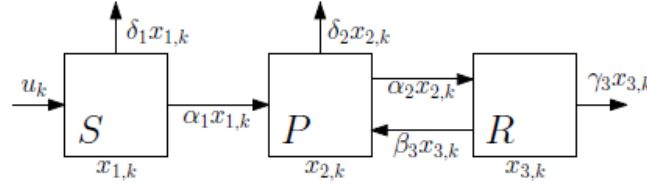
**Esempio 3.10. Supply-Chain(Matlab)**

Figura 3.1.

$$\begin{aligned}
 x_{1,k+1} &= (1 - \alpha_1 - \delta_1) x_{1,k} + u_k \\
 x_{2,k+1} &= \alpha_1 x_{1,k} + (1 - \alpha_2 - \delta_2) x_{2,k} + \beta_3 x_{3,k} \\
 x_{3,k+1} &= \alpha_2 x_{2,k} + (1 - \beta_3 - \gamma_3) x_{3,k} \\
 y_k &= \gamma_3 x_{3,k} \\
 \text{Goal: } &\text{massimizzare } \sum y_k \text{ e minimizzare } \sum u_k
 \end{aligned}$$

- **Supplier S** acquista una quantità  $u_k$  di materia prima ad ogni unità di tempo
- una porzione  $\delta_1$  viene scartata e una porzione  $\alpha_1$  viene inviata al **Producer P**
- Il **Producer** scarta una porzione  $\delta_2$ , invia una porzione  $\alpha_2$  di beni prodotti al **Retailer R** e riceve una porzioni  $\beta_3$  di prodotti difettosi
- Il **Retailer** vende una porzione  $\gamma_3$  di prodotti finiti

Figura 3.2.

**Soluzione a “batch” di ottimizzazione dinamica LQ a tempo discreto**

Abbiamo visto come risolvere il problema di ottimizzazione dinamica in maniera **ricorsiva** all'indietro attraverso l'equazioni alle differenze di Riccati.

Vogliamo ora determinare l'intera sequenza ottima di azioni:

$$U = [u_0^T \quad u_1^T \quad \dots \quad u_{N-1}^T] \in \mathbb{R}^s \quad s = mN$$

Dalla definizione di sistema dinamico a tempo discreto, notiamo che:

$$\begin{aligned}
 x_1 &= A x_0 + B u_0 \\
 x_2 &= A x_1 + B u_1 = A (A x_0 + B u_0) + B u_1 = A^2 x_0 + A B u_0 + B u_1 \\
 &\vdots \\
 x_k &= A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \quad \text{soluzione esplicita}
 \end{aligned}$$

In forma compatta:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ & \ddots & \ddots & \\ A^{N-1}B & A^{N-2}B & \dots & B \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix} + \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^N \end{pmatrix} x_0 = \bar{S}U + \bar{T}x_0$$

L'indice di **costo/prestazione** può essere scritto come:

$$\begin{aligned} J(U, x_0) &= x_0^T Q x_0 + \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}^T \begin{pmatrix} Q & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & Q & 0 \\ 0 & \dots & 0 & S \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} + U^T \begin{pmatrix} R & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & R & 0 \\ 0 & \dots & 0 & R \end{pmatrix} U = \\ &= x_0^T Q x_0 + (U^T \bar{S}^T + x_0^T \bar{T}^T) \bar{Q} (\bar{S}U + \bar{T}x_0) + U^T \bar{R} U = \\ &= \frac{1}{2} x_0^T 2 (Q + \bar{T}^T \bar{Q} \bar{T}) x_0 + x_0^T 2 \bar{T}^T \bar{Q} \bar{S} U + \frac{1}{2} U^T 2 (\bar{R} + \bar{S}^T \bar{Q} \bar{S}) U \end{aligned}$$

Quest'ultima forma compatta viene utilizzata nel **problema di minimizzazione statica non vincolata**:

$$\min_U J(U, x_0) = \frac{1}{2} U^T H U + x_0^T F^T U + \frac{1}{2} x_0^T Y x_0$$

La soluzione ottima si ottiene azzerando il gradiente  $\nabla_U J(U, x_0) = H U + F x_0$

$$U^* = \begin{pmatrix} u_0^* \\ \vdots \\ u_{N-1}^* \end{pmatrix} = -H^{-1} F x_0$$

### Soluzione Batch vs Equazione di Riccati

- La soluzione Batch non è **robusta**, infatti è ad open-loop e vuole calcolare la soluzione ottima in un passo in funzione di  $x_0$ ;
- La soluzione con Riccati diventa **onerosa** per N grandi, infatti occorre effettuare una ricorrenza all'indietro;
- La via di mezzo è quella di usare un modello **semplificato** per predire un comportamento futuro **orientativo** del sistema per selezionare un **buon** controllo.

## 3.3 Model Predictive Control MPC

### Introduzione al Model Predictive Control

#### IDEA

Supponiamo • e di voler minimizzare  $J(u) = \sum_{t=0}^{\infty} (x_t^T Q x_t + u_t^T R u_t)$ , l'idea del **model predictive controll** è quella che:

Ad ogni istante di tempo  $t \in \mathbb{N}$ :

1. Otteniamo misure aggiornate del valore dello stato corrente  $x(t) = x_t$ ;
2. Risolviamo un problema di ottimizzazione dinamica a "batch" su una finestra di  $N_u$  passi;
3. Implementiamo solo la prima azione ottima  $u_t = u_0^*$  e scartiamo tutti i valori successivi.

**Importante 3.11.** Si tratta di un problema di ottimizzazione **statica** a cui possiamo potenzialmente aggiungere vincoli.

### STRATEGIA DI MPC

**Notazione 3.12.**

- $x(t)$ : campione corrente dello stato;
- $x_k = x(t+k|t)$ : è l'evoluzione prevista dal modello data l'informazione all'istante  $t$ ;
- $u_k = u(t+k|t)$ : azione da eseguire all'istante  $t+k$  data l'informazione all'istante  $t$ ;

Ad ogni istante di tempo  $t \in \mathbb{N}$ :

- Modello di predizione:

$$x_{k+1} = A x_k + B u_k \quad x_0 = x(t)$$

$$y_k = C x_k$$

$$x_k \in \mathbb{R}^n, u_k \in \mathbb{R}^m, y \in \mathbb{R}^p$$

- Vincoli:

$$u_{\min} \leq u_i(t) \leq u_{\max} \quad i = 1, \dots, m$$

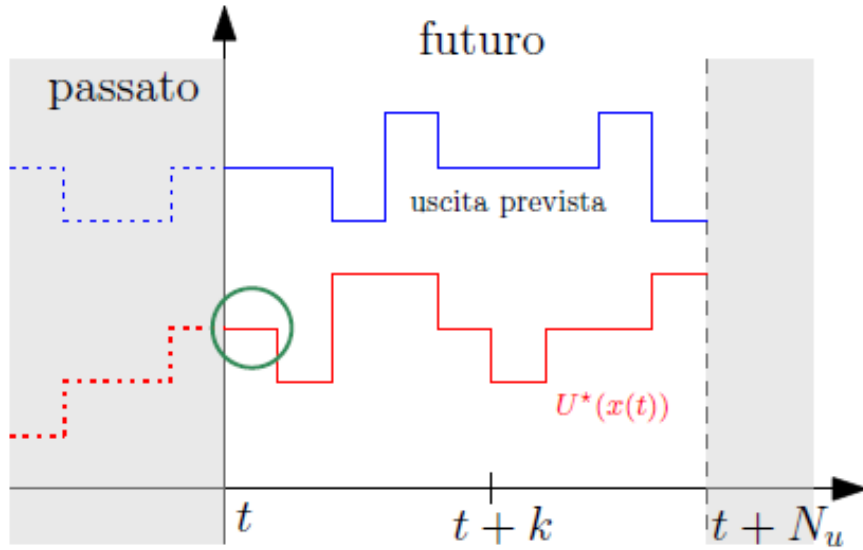
$$\Rightarrow u_{\min} \leq u_{i,k} \leq u_{\max} \quad k \in [0, N-1]$$

$$y_{\min} \leq y_j(t) \leq y_{\max} \quad j = 1, \dots, p$$

$$\Rightarrow y_{\min} \leq y_{j,k} \leq y_{\max} \quad k \in [0, N_u]$$

- Indice di prestazione ( a finestra mobile ):

$$V(x_0) = \min_u \sum_{k=0}^{N_u-1} (x_k^T Q x_k + u_k^T R u_k) + x_{N_u}^T S x_{N_u}$$



**Figura 3.3.** Finestra Mobile

- Vincoli sul controllo:

$$\begin{cases} u_{i,k} \leq u_{\max} \\ -u_{i,k} \leq -u_{\min} \end{cases} \Rightarrow \begin{pmatrix} I \\ -I \end{pmatrix} U \leq \begin{pmatrix} u_{\max} \\ -u_{\min} \end{pmatrix}$$

- Vincoli sull'uscita: ricordiamo che  $y_k = C A^k x_0 + \sum_{i=0}^{k-1} C A^i B u_{k-1-i}$

$$y_{k,i} \leq y_{\max} \Rightarrow \begin{pmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_u-1}B & \dots & CAB & CB \end{pmatrix} U \leq \begin{pmatrix} y_{\max} \\ y_{\max} \\ \vdots \\ y_{\max} \end{pmatrix} - \begin{pmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_u} \end{pmatrix} x_0$$

Relazione simile per  $-y_{k,i} \leq -y_{\min}$

### 3.3.1 MPC VINCOLATO

#### Definizione 3.15. *Model Predictive Controllo in presenza di vincoli*

Ad ogni istante di tempo  $t \in \mathbb{N}$ :

- Otteniamo misure aggiornate del valore dello stato corrente  $x(t)$ ;
- Determinano la soluzione del problema di Programmazione Quadratica:

$$(\text{QP}) := \begin{cases} V^*(x(t)) = \min_U \left\{ \frac{1}{2} U^T H U + x(t)^T F^T U \right\} + \frac{1}{2} x(t)^T Y x(t) \\ s.t. \\ G U \leq W + E x(t) \quad E x(t) := \text{feedback} \end{cases}$$

- Implementiamo solo la prima azione ottima  $u(t) = u_0^*$  di  $U^*$

**Nota 3.16.**

**QP è convesso:**

- funzione obiettivo quadratica (convessa con  $R > 0$ );
- vincoli lineari  $G U \leq W + E x(t)$  definiscono un poliedro. In cui  $G = \bullet, W = \bullet, E = \bullet$ .

### 3.3.2 Stabilità e Ammissibilità di MPC

Dobbiamo considerare 3 aspetti: **ammissibilità, stabilità** e complessità computazionale.

#### Vincolo di stabilità per MPC

Consideriamo il sistema lineare  $x_{t+1} = A x_t + B u_t, t = 0, 1, \dots, n$  e supponiamo che la strategia di MPC sia basata sul seguente problema QP:

$$\begin{aligned} V^*(x(t)) &= \min_u \sum_{k=0}^{N_u-1} x_k^T Q x_k + u_k^T R u_k \\ s.t. \\ x_{k+1} &= A x_k + B u_k, x_0 = x(t) \\ u_{\min} &\leq u_k \leq u_{\max} \\ y_{\min} &\leq C x_k \leq y_{\max} \\ x_{N_u} &= 0 \longrightarrow \text{vincolo terminale} \end{aligned}$$

con  $Q > 0, R > 0, u_{\min} < 0 < u_{\max}, y_{\min} < 0 < y_{\max}$ . Allora, se il problema QP è ammissibile al tempo  $t=0$ :

$$\lim_{t \rightarrow \infty} x_t = 0 \quad \lim_{t \rightarrow \infty} u_t = 0$$

ed i vincoli sono soddisfatti per ogni  $t$ .

#### Dimostrazione.

Supponiamo che  $U_t^* = [u_0^t, \dots, u_{N-1}^t]$  sia la soluzione ottima di QP al tempo  $t$  con valore  $V^*(x(t))$ . Dal momento che  $U_t^*$  è ammissibile, sappiamo che  $x_{t+N_u} = 0$ ;

Di conseguenza, la sequenza  $U_{t+1} = [u_1^t, \dots, u_{N-1}^t, 0]$  è ammissibile per QP al tempo  $t+1$ . Infatti:

- $x(t+1) = x(t+1|t)$  dato che abbiamo utilizzato  $u_0^t$ ;
- La sequenza  $U_{t+1}$  è tale che  $x(t+N_u|t+1) = x(t+N_u+1|t+1) = 0$

Ora, il costo  $U_{t+1}$  è:

$$V^*(x(t)) - x(t)^T Q x(t) - u(t)^T R u(t) \geq V^*(x(t+1))$$



è maggiore del valore ottimo a  $t + 1$ :

$\implies V^*(x(t))$  è monotonicamente decrescente e limitata inferiormente

$$\exists \lim_{t \rightarrow \infty} V^*(x(t)) =: V_\infty$$

Dunque  $0 \leq x(t)^T Q x(t) + u(t)^T R u(t) \leq V^*(x(t)) - V^*(x(t+1)) \rightarrow 0$  per  $t \rightarrow \infty$  e la tesi segue dal fatto che  $Q$  ed  $R$  sono matrici definite positive.  $\square$

### 3.3.3 Ottimizzazione On-line vs Ottimizzazione Off-line

**Definizione 3.17.** *MPC implicito vs MPC esplicito*

$$\begin{cases} V(x(t)) = \min_U \left\{ \frac{1}{2} U^T H U + x(t)^T F^T U \right\} + \frac{1}{2} x(t)^T Y x(t) \\ \text{s.t. } G U \leq W + E x(t) \end{cases}$$

Ottimizzazione **on-line**: dato  $x(t)$  risolviamo il problema ad ogni tempo  $t \in \mathbb{N}$ :

- Legge di controllo  $u = u_o^*$  è funzione **implicita** di  $x(t)$ ;
- Problema di Programmazione Quadratica (QP);

Ottimizzazione **off-line**: risolviamo il problema QP a priori per ogni  $x(t)$  in un dato insieme:

- La legge di controllo  $u = u_o^*(x(t))$  è funzione **esplicita** di  $x(t)$ ;
- QP multi-parametrico (mp-QP);

#### QP multi-parametrico mp-QP

Dato il problema di ottimizzazione

$$\begin{aligned} \min_z f(z, x) \quad & x \in \mathbb{R}^n \\ \text{s.t. } g(z, x) & \leq 0 \quad z \in \mathbb{R}^s \end{aligned}$$

e un insieme  $X \subseteq \mathbb{R}^n$  di parametri  $x$  di interesse, determinare:

- insieme dei **parametri ammissibili**  $X^* \subseteq X$  per i quali il problema ammette soluzioni;
- **funzione valore**  $V^*: X^* \rightarrow \mathbb{R}$  che associa il valore ottimo a ciascun  $x \in X^*$ ;
- una **funzione ottimizzante**  $z^*: X^* \rightarrow \mathbb{R}^s$  che associa la soluzione ottima a ciascun  $x \in X^*$

#### mp-QP per MPC esplicito

**Definizione 3.18.** *Funzione affine a tratti*

Una funzione  $U(x): X \rightarrow \mathbb{R}^s$ , dove  $X \subseteq \mathbb{R}^n$  è un poliedro, è **affine a tratti** se è possibile partizionare  $X$  in regioni poliedrali convesse  $C R_i$  e tali che:

$$U(x) = H^i x + k^i$$

per ogni  $x \in C R_i$ .

**Nota 3.19.** è una traslazione di una funzione lineare.

Come **passo preliminare** per l'algoritmo, introduciamo:

$$z = U + H^{-1} F x(t)$$

**Nota 3.20.** Si ha il vantaggio di passare da una funzione quadratica non omogenea con termine quadratico, lineare e costante in  $U$  ad una forma quadratica omogenea nella variabile  $x$ .

Il problema mp-QP diventa:

$$\text{mp-QP}_z := \begin{cases} V_z(x) = \min_z \frac{1}{2} z^T H z \iff V(x) = V_z(x) + \frac{1}{2} x^T (Y - F^T H^{-1} F) x \\ \text{s.t. } G z \leq W + M x \iff M = E + G H^{-1} F \end{cases}$$

Infatti, sostituendo la  $x$  nel costo e  $y = F^T H^{-1} F$ :

$$V_z = \frac{1}{2} (U^T + x^T F^T H^{-1}) H (U + H^{-1} F x) = \frac{1}{2} U^T H U + \frac{1}{2} x^T F^T H^{-1} F x + \frac{1}{2} U^T F x + \frac{1}{2} x^T F^T U$$

### Struttura della soluzione ottima di mp-QP<sub>z</sub>

Dati:

- un **insieme poliedrale**  $X := \{x: T x \leq Z\}$  di parametri per cui risolvere mp-QP<sub>z</sub>;
- un **vettore iniziale**  $x_0 \in X$  per cui mp-QP<sub>z</sub> ammette una soluzione ammissibile.

Allora:

- risolviamo mp-QP<sub>z</sub> per  $x = x_0$ , ottenendo la soluzione ottima  $z_0$ ;
- dal momento che  $H$  è definita positiva,  $z_0$  è unica;
- $z_0$  determina univocamente un sottoinsieme di **vincoli attivi**:

$$G z_0 \leq W + M x_0 \iff \tilde{G} z_0 = \tilde{W} + \tilde{M} x_0 \quad \hat{G} z_0 < \hat{W} + \hat{M} x_0 \\ \implies \tilde{G}, \tilde{W}, \tilde{M} \text{ caratterizzano i vincoli attivi della soluzione ottima } z_0 \text{ e } x_0$$

### Definizione 3.21. Struttura di $z^*(x)$

Supponiamo che le righe di  $\tilde{G}$  siano linearmente indipendenti. Sia  $C R_0$  (**regione critica**) l'insieme di tutti i vettori  $x$  per i quali la stessa combinazione di vincoli è attiva all'ottimo  $z^*(x)$ . Allora  $z^*(x)$  è una funzione **affine** di  $x$  univocamente definita in  $C R_0$  e data da:

$$z^*(x) = H^{-1} \tilde{G}^T (\tilde{G} H^{-1} \tilde{G})^{-1} (\tilde{W} + \tilde{M} x) \quad \forall x \in C R_0$$

### Dimostrazione.

Sia  $\lambda \in \mathbb{R}^q$  il moltiplicatore di Lagrange del problema di ottimizzazione vincolata mp-QP<sub>z</sub>, allora le condizioni al primo ordine di **Karush-Kuhn-Tucker** sono:

1.  $H z + G^T \lambda = 0$  := derivata funzione Lagrangiana;
2.  $\lambda_i (G^i z - W^i - M^i x) = 0 \quad i=1, \dots, q$  condizione di complementarità;
3.  $\lambda \geq 0$  condizioni di non-negatività;
4.  $G z \leq W + M x$  condizioni di ammissibilità.

Da (1)  $\implies z = -H^{-1} G^T \lambda$  e sostituendo in (2):

$$\lambda (-G H^{-1} G^T \lambda - W - M x) = 0$$

Siano  $\tilde{\lambda}$  e  $\hat{\lambda}$  i moltiplicatori a vincoli non-attivi e attivi, rispettivamente. Allora:

$$\implies \hat{\lambda} = 0 \text{ da (2) e } (G^i z - W^i - M^i x) < 0;$$

$$\implies \text{per i vincoli attivi } \tilde{G} z - \tilde{W} - \tilde{M} x = 0 \text{ implica che:}$$

$$-\tilde{G} H^{-1} \tilde{G}^T \tilde{\lambda} - \tilde{W} = 0 \implies \tilde{\lambda} = -(\tilde{G} H^{-1} \tilde{G}^T)^T (\tilde{W} + \tilde{M} x)$$

Quindi  $\lambda = (\tilde{\lambda}, \hat{\lambda}) = (-(\tilde{G} H^{-1} \tilde{G}^T)^{-1} (\tilde{W} + \tilde{M} x), 0)$  che sostituito in (1) fornisce  $z^*$ .  $\square$

### Regione Critica $C R_0$

Ora vogliamo **caratterizzare** esplicitamente la regione critica  $C R_0$ . Imponiamo ammissibilità prima e duale, ovvero:

1.  $G H^{-1} \tilde{G}^T (\tilde{G} H^{-1} \tilde{G}^T)^{-1} (\tilde{W} + \tilde{M} x) \leq W + M x$  **ammissibilità**  $\iff \hat{G} z(x) \leq \hat{W} + \hat{M} x$
2.  $-(\tilde{G} H^{-1} \tilde{G}^T)^{-1} (\tilde{W} + \tilde{M} x) \geq 0$  **vincolo di non negatività**

Allora i vincoli sono lineari in  $x$ .

Quindi:

**Definizione 3.22.** *Regione critica*

---

$$CR_0 := \{x \in X : A_0 x \leq b_0\}$$

con  $A_0, b_0$  ottenute da (1),(2) dopo aver eliminato i vincoli **ridondanti**:

**$CR_0$  è un sottoinsieme poliedrale di  $X$**

**Esplorazione di  $X \setminus CR_0$**

Dopo aver calcolato  $CR_0$  definiamo:

$$CR_{\text{rest}} = X \setminus CR_0$$

ovvero la regione di  $X$  non inclusa in  $CR_0$  su cui non conosciamo ancora la soluzione. Quindi **iteriamo**, cacciamo una regione critica in sottoinsiemi poliedrali di  $CR_{\text{rest}}$ .

**Definizione 3.23.** *Decomposizione poliedrale*

Sia  $X$  un poliedro e  $CR_0 := \{x \in X : A x \leq b\}$  sottoinsieme di  $X$ ,  $CR_0 \neq \emptyset$ . Inoltre, siano  $R_i = \{x \in X : A^i x > b^i, A^j x \leq b^j, \forall j < i\}, i = 1, \dots, \dim(b)$  e sia  $CR_{\text{rest}} = \bigcup_{i=1}^{\dim(b)} R_i$ . Allora:

$$1. CR_{\text{rest}} \cup CR_0 = X \quad CR_0 \cap R_i = \emptyset;$$

$$2. R_i \cap R_j = \emptyset \quad \forall i \neq j$$

$\implies \{CR_0, R_1, \dots, R_{\dim(b)}\}$  è una **partizione** di  $X$ .

**Dimostrazione.**

(1) Dato  $x \in X$ , vogliamo dimostrare che  $x$  appartiene o a  $CR_0$  o a  $CR_{\text{rest}}$  ovvero a  $R_i$  per qualche  $i$ :

- Se  $x \in CR_0$ , la conclusione è ovvia;
- Se  $x \notin CR_0$ , esiste un indice  $i$  tale che  $A^i x > b^i$ . Sia  $i^* = \min_{i \leq \dim(b)} \{i : A^i x > b^i\}$   
Allora  $x \in R_{i^*}$ , in quanto  $A^j x \leq b^j \quad \forall j < i^*$  per definizione di  $i^*$

(2)

- Sia  $x \in CR_0$ , allora non esiste  $i$  tale che  $A^i x > b^i \implies x \notin R_i$  per nessuna  $i$ ;
- Sia  $x \in R_i$  e prendiamo  $i > j$ . Dal momento che  $x \in R_i$ , per definizione di  $R_i$ ,  $A^j x \leq b^j$  che implica che  $x \notin R_j$ . □



# Capitolo 4

## Reinforcement Learning

### 4.1 Introduzione

#### Definizione 4.1. Reinforcement Learning (RL)

*Il reinforcement learning è un insieme di problemi di apprendimento in cui un **agente** impara come mappare ciascuno **stato** dell'ambiente in un'**azione** corrispondente per massimizzare la **somma delle ricompense** ricevute.*

#### Importante 4.2. Idea di RL

All'agente non viene detto esplicitamente quale azione eseguire, ma deve imparare dall'esperienza quale azione fornisce la ricompensa maggiore **provando ad eseguirle**.

#### Importante 4.3.

- RL è diverso da **apprendimento supervisionato** in cui l'agente impara da una serie di esempi di azioni corrette suggerite da un utente esperto;
- RL è diverso da **apprendimento non-Supervisionato**, in cui si vuole determinare pattern nascosti in un insieme di dati;

**Importante 4.4.** RL terzo paradigma di Machine Learning, caratterizzato da:

- Goal-oriented learning;
- Assenza di supervisor;
- Importanza del tempo;
- Trade-off tra exploration e exploitation.

#### Notazione 4.5.

- **Agente:** sistema di controllo;
- **Azione:** controllo;
- **Ambiente:** sistema;
- **Funzione valore ottima:** funzione valore;
- **Learning:** risolvere un problema di DP senza modello;
- **Policy:** legge di controllo in retro-azione;
- **Ricavo complessivo:** opposto di funzionale di costo;
- **Ricompensa:** opposto di costo corrente;
- **Problema formulato del tipo:**

$$\begin{aligned} \dot{x} &= Fx + Gu \\ y &= Hx \end{aligned}$$

#### 4.1.1 Elementi principali di Reinforcement Learning

- **Policy:** definisce il comportamento dell'agente per ogni istante di tempo e in funzione dello stato corrente;

- **Ricompensa:** (reward signal) definisce l'obiettivo di RL; ad ogni istante l'ambiente invia all'agente un segnale di ricompensa, in funzione di stato corrente e dell'azione eseguita;
- **Funzione valore:** specifica quali sono le azioni migliori nel lungo periodo (quantità totale di ricompensa che possiamo ottenere a partire da un determinato stato). Uno stato potrebbe fornire una ricompensa bassa (breve periodo) ma avere un elevato valore (lungo periodo);
- **Modello:** (dell'ambiente) da utilizzare eventualmente per pianificare scelte future, ma siamo interessati a metodi model-free.

#### 4.1.2 Schema di Reinforcement Learning

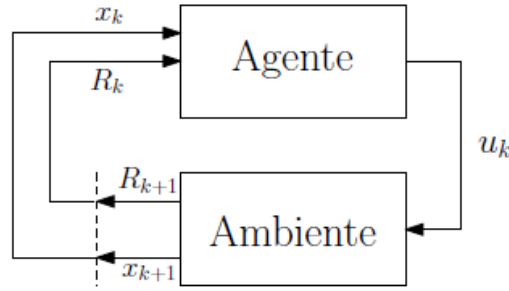


Figura 4.1.

L'**agente** si trova in  $x_k \in X$  (insieme dei possibili stati) e sceglie l'**azione**  $u_k \in U$  (insieme delle possibili azioni) e come conseguenza della sua azione riceve una **ricompensa**  $R_{k+1} \in \mathbb{R}$ .

#### Marko Decision Process (MDP)

Questo schema può essere modellizzato tramite un **Markov Decision Process (MPD)**.

Caso Deterministico:

- Il comportamento dell'ambiente è descritto da un modello (azione  $\rightarrow$  nuovo stato)

$$f: X \times U \rightarrow X \quad x_{k+1} = f(x_k, u_k)$$

#### Proprietà Markoviana dello stato

In RL, lo stato deve essere tale da riassumere in maniera compatta tutta l'informazione passata in modo da determinare in maniera univoca il valore successivo.

- Il comportamento dell'agente è descritto da una policy (stato  $\rightarrow$  azione)

$$\pi: X \rightarrow U \quad u_k = \pi(x_k)$$

#### Policy stazionaria

Per ciascuno stato fornisce sempre la stessa azione indipendentemente dal punto della sequenza in cui si incontra lo stato.

- L'ambiente genera una ricompensa (stato, azione  $\rightarrow$  ricompensa)

$$\varrho: X \times U \rightarrow \mathbb{R} \quad R_{k+1} = \varrho(x_k, u_k)$$

#### 4.1.3 Obiettivo di Reinforcement Learning

**Obiettivo:** massimizza la somma delle ricompense ricevute, ovvero il **ricavo** complessivo dall'istante  $k$  e dallo stato  $x \in X$ :

$$G_k(x) = R_{k+1} + R_{k+2} + \dots + R_N$$

La definizione di  $G_k(x)$  ha senso soli se esiste una conclusione ragionavelo del problema. Quindi si parla di **apprendimento episodico**: quando viene raggiunto uno stato terminale in  $X^+$ , lo stato viene resettato ad uno stato iniziale in  $X$ .

Nel caso di **apprendimento continuo**:

$$G_k(x) = R_{k+1} + \gamma R_{k+2} + \gamma^2 R_{k+3} + \dots + \sum_{i=0}^{\infty} \gamma^i R_{k+i+1} \quad 0 \leq \gamma \leq 1, \text{ fattore di sconto}$$

**Nota 4.6.** Nel caso in cui  $\gamma = 0$ , l'agente è interessato solo alla ricompensa immediata.

## 4.2 Value Function nel Reinforcement Learning

### 4.2.1 Funzione Valore di Stato

Tutti gli algoritmi di RL si basano sulla **stima** della **funzione valore**. Essa fornisce il **valore** di uno stato, ovvero quanto ricavo si potrebbe ottenere in futuro per il fatto di trovarsi in quel particolare stato.

**Definizione 4.7.** *Valore di  $\pi$  da  $x$*

*Si definisce **funzione valore di stato per la policy  $\pi$** :*

$$v_{\pi}(x) = G_k(x) = \sum_{i=0}^{\infty} \gamma^i R_{k+i+1}$$

In cui  $R_{k+i+1}$  sono ottenute ad ogni passo seguendo la policy  $\pi$ .

Selezionando, ad ogni passo  $u_i = \pi(x_i) \rightarrow x_{i+1} = f(x_i, \pi(x_i))$  e  $R_{i+1} = \rho(x_i, \pi(x_i))$

#### 4.2.1.1 Equazione del valore di $v_{\pi}$

La funzione valore soddisfa una condizione di **consistenza ricorsiva**, otteniamo l'**equazione del valore di  $v_{\pi}$** :

$$\begin{aligned} v_{\pi}(x) &= \sum_{i=0}^{\infty} \gamma^i R_{k+i+1} = R_{k+1} + \sum_{i=1}^{\infty} \gamma^i R_{k+i+1} \\ &= R_{k+1} + \gamma \sum_{i=0}^{\infty} \gamma^i R_{k+i+2} \\ &= R_{k+1} + \gamma v_{\pi}(f(x, \pi(x))) = R_{k+1} + \gamma v_{\pi}(x_{k+1}) \end{aligned}$$

#### 4.2.1.2 Soluzione dell'equazione del valore di $v_{\pi}$

Assumendo  $X$  insieme degli stati ammissibili di cardinalità finita,  $|X| = m$

$\Rightarrow$  L'equazione per il valore di  $\pi$  è **lineare** e può essere risolta in forma chiusa.

Infatti, **data una policy  $\pi: X \rightarrow U$** :

- La transizione da uno stato all'altro è fissata univocamente  $x_{k+1} = f(x_k, \pi(x_k))$

**Esempio 4.8.** Se ci troviamo nello stato 2 il successore è determinato da  $f(2, \pi(2)) = 4$

- La ricompensa ottenuta è fissata univocamente  $R_{k+1} = \rho(x_k, \pi(x_k))$

**Esempio 4.9.** Se ci troviamo nello stato due la ricompensa è determinata da  $\rho(2, \pi(2)) = 1$

**Esempio 4.10.**

Dunque, l'equazione del valore di  $v_{\pi}$  per  $x = 2$  è:

$$v_{\pi}(2) = 1 + \gamma v_{\pi}(4)$$

Ovvero:

$$v_{\pi}(2) = 1 + \gamma [0, 0, 0, 1, 0, \dots] \begin{pmatrix} v_{\pi}(1) \\ v_{\pi}(2) \\ v_{\pi}(3) \\ v_{\pi}(4) \end{pmatrix}$$

**Definizione 4.11.**

Possiamo definire la **matrice di transizione**  $T_\pi$  e il vettore delle ricompense  $R_\pi$ :

- La riga  $i$ -esima di  $T_\pi$  ha 1 nella colonna  $f(i, \pi(i))$  e 0 altrimenti;
  - La riga  $i$ -esima del vettore  $R_\pi$  è uguale a  $\varrho(i, \pi(i))$
- Quindi:

$$v_\pi(x) = R_{k+1} + \gamma v_\pi(x_{k+1}) \quad \forall x \iff v = R_\pi + \gamma T_\pi v \implies v = (I - \gamma T_\pi)^{-1} R_\pi$$

$$\text{dove } v = \begin{pmatrix} v_\pi(1) \\ v_\pi(2) \\ v_\pi(3) \\ v_\pi(4) \end{pmatrix}$$

**4.2.2 Funzione valore di stato/azione**

**Definizione 4.12.** *Funzione Valore dell'azione  $u$  con  $\pi$  da  $x$*

$$\begin{aligned} q_\pi(x, u) &= G_k(x)|_{u_k=u} = R_{k+1} + \sum_{i=1}^{\infty} \gamma^i R_{k+i+1} \\ &= \varrho(x, u) + \gamma v_\pi(x_{k+1}) \end{aligned}$$

Dove la ricompensa  $R_{k+1}$  è ottenuta implementando l'azione  $u$  all'istante  $k$  e seguendo la policy  $\pi$  da  $x_{k+1} = f(x, u)$  in poi.

**Nota 4.13.**

Di norma la funzione valore di stato/azione  $>$  funzione valore di stato

**4.2.3 Funzione Valore Ottima**

La funzione valore  $v_\pi$  permette un **ordinamento parziale** delle policy:

$$\pi \geq \pi' \iff v_{\pi'} > v_\pi \quad \forall x \in X$$

Quindi possiamo definire la **policy ottima** e la **funzione valore ottima** corrispondere se e solo se:

$$v_*(x) = \max_{\pi} v_\pi(x)$$

Allora il massimo ricavo ottenibile da  $x$ :

$$q_*(x, u) = \max_{\pi} q_\pi(x, u) = R_{k+1} + \gamma v_*(x_{k+1})$$

Coincide con il massimo ricavo ottenibile da  $x$ , dopo aver deciso di eseguire in ogni caso l'azione  $u$  al tempo  $k$ .

**4.2.4 Policy ottima**

**Se conosciamo  $v_*$ :** l'azione che appare migliore in un singolo passo è la soluzione ottima dell'intero problema decisionale:

$\implies$  **strategia greedy** rispetto a  $v_*$ :

$$u_k \in \arg \max_u \{R_{k+1} + \gamma v_*(f(x, u))\} \quad \forall x \in X$$

**Se conosciamo  $q_*$ :**

$$u_k \in \arg \max_u q_*(x, u) \quad \forall x \in X$$

**Nota 4.14.**

L'ultima equazione comporta:

- maggiore costo di memoria poiché occorre salvare valori per ciascuna coppia  $(x, u)$ ;
- Non dobbiamo conoscere più nulla di possibili successori  $(f(x, u))$  o ricompense  $R_{k+1}$ ;



## 4.3 Policy e Value iteration

Il problema di determinazione di policy ottima può essere risolto tramite la raccolta di informazioni dell'ambiente in maniera iterativa. Le tecniche usate sono la **policy iteration** e **value iteration**.

### 4.3.1 Policy Evaluation

Data una policy  $\pi$ , vogliamo calcolare il suo valore  $v_\pi$  da ciascuno stato:

$$v_\pi(x) = R_{k+1} + \gamma R_{k+2} + \dots = R_{k+1} + \gamma v_\pi(x')$$

In cui  $x' = f(x, \pi(x))$  è il **successo di  $x$  seguendo  $\pi$** .

**Nota 4.15.**

Questo è un sistema di  $|X|$  equazioni lineari e la sua inversa potrebbe essere non banale

#### 4.3.1.1 Approssimazione Iterativa

Consideriamo la sequenza  $v_0, v_1, \dots$  di approssimazioni di  $v_\pi$ ,  $v_i: X \rightarrow \mathbb{R}$ , con  $v_0$  arbitraria tranne  $v_0(x) = 0 \quad \forall x \in X^+$  definita da:

$$v_{i+1}(x) = R_{k+1} + \gamma v_i(x') \quad \forall x \in X^+, i = 0, 1, 2, \dots$$

- Se ad un passo non aggiorniamo più il valore per nessuno stato, ovvero  $v_{i+1}(x) = v_i(x)$ , allora  $v_i$  è detto **punto fisso** e risolve:

$$v_\pi(x) = R_{k+1} + \gamma R_{k+2} + \dots = R_{k+1} + \gamma v_\pi(x');$$

- Si dimostra che  $v_i \rightarrow v_\pi$  per  $i \rightarrow \infty$ ;

**Nota 4.16.** Implementazione

- Abbiamo bisogno di due vettore per salvare  $v_{i+1}$  e  $v_i$  (**full backup**), ma possiamo anche usarne solo uno utilizzando il valore  $v_i(x')$  se già disponibili;
- Abbiamo bisogno di uno **stopping criterion**:  $\max_{x \in X} |v_{i+1}(x) - v_i(x)| < \varepsilon$  con  $\varepsilon > 0$  sufficientemente piccolo.

**Algoritmo 4.1**

**Input:**  $\pi$ , policy da valutare,  $\varepsilon > 0$

**Inizializzazione:**  $V(x)$  tale che  $V(x) = 0, \forall x \in X^+, \Delta = \varepsilon$

**Iterazioni:**

```

    while  $\Delta \geq \varepsilon$ 
         $\Delta = 0$ 
        for each  $x \in X$ 
             $v = V(x)$ 
             $V(x) = \rho(x, \pi(x)) + \gamma V(x')$ 
             $\Delta = \max \{ \Delta, |v - V(x)| \}$ 
        end
    end
end
```

### 4.3.2 Policy Improvement

Tramite 4.3.1 possiamo calcolare il valore di una policy. A questo punto supponiamo di avere una policy  $\pi$  e di aver calcolato il suo valore  $v_\pi$ , ci chiediamo se per qualche stato non sarebbe meglio un'azione  $u \neq \pi(x)$ .

**Definizione 4.17.**

Possiamo implementare  $u$  nello stato  $x$  e poi seguire  $\pi$ , ovvero sfruttare la funzione  $q_\pi(x, u)$ :

$$q_\pi(x, u) = R_{k+1} + \gamma v_\pi(x')$$

Se  $q_\pi(x, u) \geq v_\pi(x) \implies$  conviene eseguire  $u$  in  $x$  invece di  $v_\pi(x)$ .

**Dimostrazione.**

Siano  $\pi$  e  $\hat{\pi}$  due policy, se vale che:

$$q_\pi(x, \hat{\pi}(x)) \geq v_\pi(x) \quad \forall x \in X$$

Allora, sfruttando la disuguaglianza iterativamente per ogni stato:

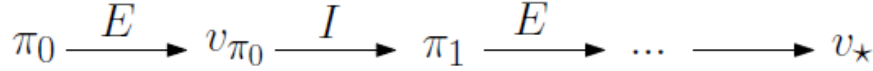
$$\begin{aligned} v_\pi(x) &\leq q_\pi(x, \hat{\pi}(x)) = R_{k+1} + \gamma v_\pi(x') \\ &\leq R_{k+1} + \gamma q_\pi(x', \hat{\pi}(x')) = R_{k+1} + \gamma R_{k+2} + \gamma^2 v_\pi(x'') \leq \dots \leq v_{\hat{\pi}}(x) \end{aligned} \quad \square$$

Eseguendo la stessa operazione per ogni stato, possiamo definire l'intera policy  $\hat{\pi}$ :

$$\begin{aligned} \hat{\pi} &= \arg \max_u q_\pi(x, u) \quad \forall x \in X \\ &= \arg \max_u \{ \rho(x, u) + \gamma / v_\pi(f(x, u)) \} \end{aligned}$$

Quindi si ha che  $\hat{\pi}$  è definita aggiornando  $\pi$  in modo **greedy** rispetto a  $q_\pi$ .

Se non esiste uno stato  $x \in X$  per cui  $v_{\hat{\pi}}(x) > v_\pi(x)$  all'ora  $v_{\hat{\pi}}(x) = v_\pi(x)$  è un **punto fisso**. Alternando Policy Evaluation e Policy Improvement si ottiene la funzione valore ottima associata alla policy ottima.



**Figura 4.2.**

**Algoritmo 4.2**

**Inizializzazione:**  $V(x), \pi(x)$

**Iterazioni:**

1. **Policy Evaluation**

2. **Policy Improvement**

$policy-stable = \text{TRUE}$

for each  $x \in X$

$a = \pi(x)$

$\pi(x) \in \arg \max_u \{ \rho(x, u) + \gamma V(x, u) \}$

if  $a \neq \pi(x)$ , then  $policy-stable = \text{FALSE}$

if  $policy-stable$ , stop and return  $V, \pi$

else goto 1. Policy Evaluation

## 4.4 Value Iteration

Poiché in **policy iteration** in ogni passo si deve eseguire *policy evaluation* e la convergenza è assicurata per un numero infinito di passi, ma **realisticamente** ne possiamo effettuare un numero **finito**.

Si è dimostrato che Policy Iteration converge anche se si esegue una Policy Evaluation troncata, cioè in un numero finito di passi:

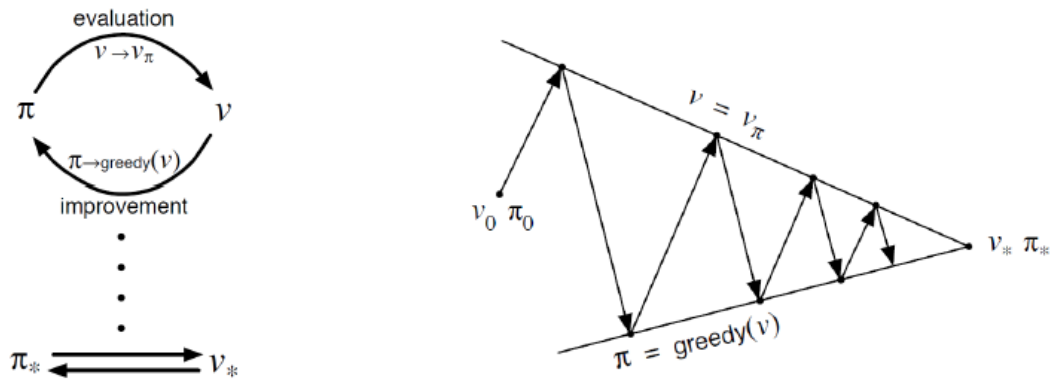


Figura 4.3.

Portanto al “limite” questa considerazioni, si può pensare di fare solamente una sola iterazione:  
**Value Iteration**

$$v_{i+1}(x) = \max_u \{R_{k+1} + \gamma v_i(x')\} \quad \forall x \in X$$

Quindi l’equazione di Bellman diventa la legge di aggiornamento del nostro algoritmo.  
 [Vedere esempio Lezione 7 slide 11]

## 4.5 Programmazione Dinamica Asincrona

Lo svantaggio di **Value Iteration** è che ad ogni iterazione abbiamo considerato un *full backup* di tutti gli stati. Non è sempre possibile farlo.

Tramite la **programmazione dinamica asincrona (DP asincrona)**, è possibile aggiornare solo gli stati coinvolti nell’episodio appena concluso:

- il valore di alcuni stati potrebbe essere aggiornato molte volte prima che altri lo siano anche una sola volta;
- bisogna garantire che tutti gli stati siano sufficientemente aggiornati;

## 4.6 Processi Decisionali di Markov Stocastici

### 4.6.1 Richiamo sulle Variabili Aleatorie

**Definizione 4.18.** *Spazio delle probabilità*

Lo **spazio delle probabilità** è la terna  $(\Omega, A, Pr)$  dove:

- $\Omega$ , spazio campionario;
- $A$ , spazio degli eventi;
- $Pr$ , misura di probabilità per gli eventi in  $A$

**Definizione 4.19.** *Probabilità*

Per ogni evento  $A \in A$  si definisce la **probabilità di  $A$** :

$$Pr[A]$$

**Assioma 4.20.**

- $Pr[A] \geq 0$  per ogni evento  $A \in A$ ;
- $Pr[\Omega] = 1$ ;

- $\Pr[A_1 \cup A_2 \cup \dots \cup A_n] = \Pr[A_1] + \Pr[A_2] + \dots$  per eventi disgiunti;

Nei **processi decisionali di markov** consideriamo spazi di probabilità **discreti**:

$$\Omega = \{w_1, w_2, \dots, w_1\}$$

**Definizione 4.21. Probabilità condizionata**

La **probabilità condizionata** confronta la probabilità di un evento con quella dello stesso evento data l'informazione aggiuntiva:

$$\Pr[A|B] = \frac{\Pr[A \cap B]}{\Pr[B]}$$

**Definizione 4.22. Variabile Aleatoria Discreta  $X$**

Dato uno spazio di probabilità discreto  $(\Omega, \mathcal{A}, \Pr)$ ,  $X$  è una funzione da  $\Omega \rightarrow \mathbb{R}$

$$x = X(w)$$

$\Rightarrow X(\cdot)$  è la **variabile aleatoria** e  $x$  è il valore;

$\Rightarrow X$  rappresenta un modo più semplice di rappresentare eventi.

**Definizione 4.23. Funzione di massa di probabilità**

$$p_X(x) = \Pr[X = x]$$

**Definizione 4.24. Valore Atteso**

$$E[x] = \sum_x x p_X(x)$$

**Definizione 4.25. Valore Atteso Condizionato**

$$E[X|A] = \sum_x x p_{X|A}(x) \quad p_{X|A}(x) = \Pr[X = x|A]$$

## 4.6.2 Processo decisionale di Markov Stocastico

Fino ad ora abbiamo visto che, dato il valore dello stato  $x$  e l'azione  $u$  all'istante  $k$ , allora il valore dello stato successivo  $x'$  è **univocamente** fissato dal modello deterministico  $f$ :

$$x' = f(x, u)$$

e di ricevere la ricompensa:

$$r = R_{k+1} = \rho(x, u)$$

Ora vogliamo generalizziamo nel contesto **non deterministico**. Per fare ciò occorre che:

- lo stato successivo sia  $x'$  e la ricompensa  $r$ , specificando la **funzione di massi di probabilità**:

$$p(x', r|x, u) = \Pr[R_{k+1} = r, X_{k+1} = x'|X_k, U_k]$$

In cui  $R, X, U$  rappresentano le variabili aleatorie discrete da cui possiamo calcolare la **probabilità di transizione di stato**:

$$p(x', |x, u) = \sum_r p(x', r|x, u)$$

- si descriva una **policy** come una distribuzione di probabilità  $\pi(u|x)$
- l'**obbiettivo** sia di massimizzazione del valore atteso del ricavo:

$$E_\pi[G_k|X_k = x] = E_\pi\left[\sum_{i=0}^{\infty} R_{k+i+1}|X_k = x\right]$$

Ottenendo la corrispettiva funzione valore  $v_\pi$ :

$$v_\pi = \sum_u \pi(u|x) \sum_{x'} \sum_r p(x', r|x, u) [r + \gamma v_\pi(x')]$$

e la funzione valore stato:

$$q_\pi(x, u) = \sum_{x'} \sum_r p(x', r|x, u) [r + \gamma v_\pi(x')]$$

Nel caso ottimo  $v_*$ :

$$v_* = \max_u \sum_{x'} \sum_r p(x', r|x, u) [r + \gamma v_\pi(x')]$$

### 4.6.3 Confronto Grafico

Di seguito è proposto il confronto dell'equazione di Bellman nel caso deterministico e non:

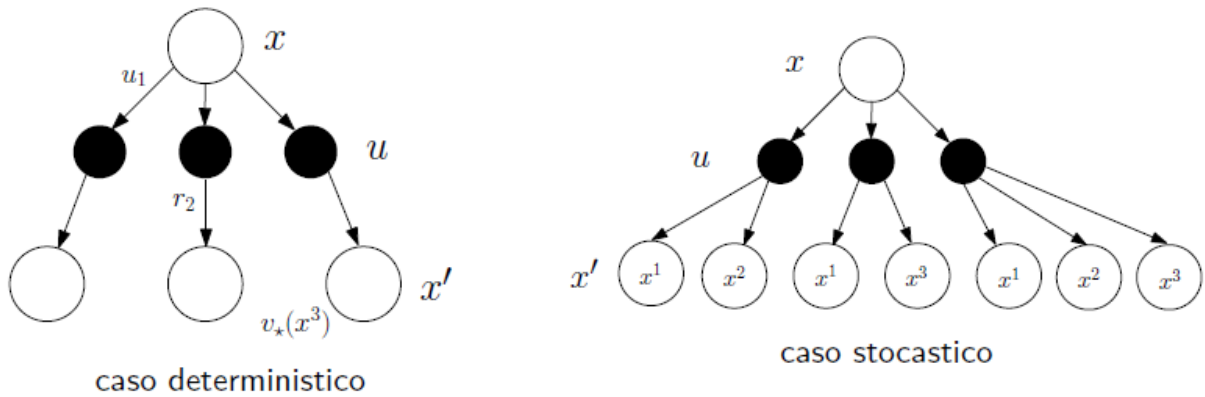


Figura 4.4.

[Esempio Lezione 8, slide 11-12]

## 4.7 Temporal-difference learning e Q-learning

Sono tecniche che estendono il reinforcement learning in metodi detti **data-driven** cioè non sapendo il modello e dalla logica di reward. Le fasi di ploci evaluatione e improvemente si basano su dati raccolti tramite esperimenti sul modello vero oppure applicati su un modello simulativo.

### 4.7.1 Temporal-difference learning

Il **temporal-difference learning**, ha come idea centrale del RL:

- imparare direttamente dall'esperienza senza modello *model-free*;
- aggiorna i valori correnti anche sulla base di informazioni stimati *bootstrapping*;

Quindi, data l'esperienza, cioè le ricompense ricevute, seguendo una policy  $\pi$ , si aggiorna la stima del valore di uno stato intermedio:

- metodi **Monte Carlo**: si aspetta la fine dell'episodio, si raccoglie il vero ricavo ottenututo  $x_k$ , ovvero  $G_k = \sum_{i=0}^N \gamma^i R_{k+i+1}$  e si usa tale valore come **target** per aggiornare la stima dello stato  $x_k$ :

$$V(x_k) = V(x_k) + \alpha [G_k - V(x_k)] \quad \alpha \in (0, 1)$$

- **TD**: si aspetta solo l'istente successivo e si usa la sua stima attuale come target:

$$V(x_k) = V(x_k) + \alpha [R_{k+1} + \gamma V(x_{k+1}) - V(x_k)]$$

### 4.7.2 Monte Carlo vs TD

**Problema 4.1.** Tornare a casa dall'università

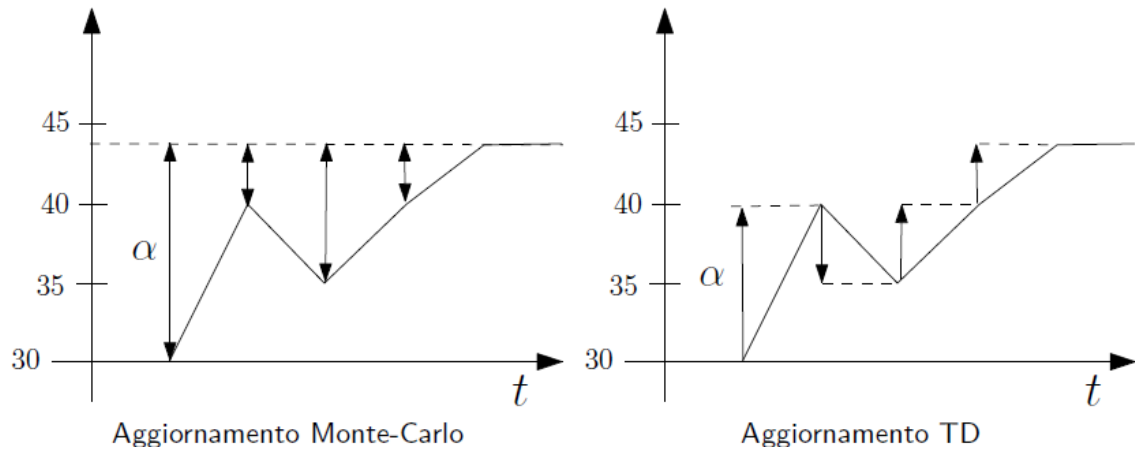
⇒ le ricompense corrispondono al tempo trascorso;

⇒ stati: lasciare università, arrivare alla macchina, lasciare il GRA, strada interna, strada di casa, arrivare a casa.

Quindi:

stato	tempo trascorso	stima tempo rimasto	stima tempo totale
1	0	30	30
2	5	35	40
3	20	15	35
4	30	10	40
5	40	3	43
6	43	0	43

Vediamo l'aggiornamento delle due metodologie:



- Si può dimostrare che la stima converge al valore vero  $V(x)$ , a patto che il passo di aggiornamento (tempo-variante)  $\alpha_k$  soddisfi:

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Cerchiamo di usare TD per il **controllo** utilizzando lo schema di policy iteration con la fase di evaluation eseguita con TD.

### 4.7.3 On-policy TD:SARSA

Con TD stimiamo direttamente  $q_{\pi}(x, u)$  invece di  $v_{\pi}(x)$ :

$$Q(x_k, u_k) = Q(x_k, u_k) + \alpha [R_{k+1} + \gamma Q(x_{k+1}, u_{k+1}) - Q(x_k, u_k)]$$

La legge di aggiornamento utilizza tutti gli elementi della quintupla:

$$(x_k, u_k, R_{k+1}, x_{k+1}, u_{k+1})$$

Da qui il nome SARSA: Stato-Azione-Reward-Stato-Azione.

#### Algoritmo 4.3

- Ci troviamo in uno stato  $x_k$ ;

- Scegliamo un'azione  $u_k$  **in base a Q**  $\Rightarrow u_k \in \arg \max_u Q(x_k, u)$ ;
- Riceviamo una ricompensa  $R_{k+1}$  e andiamo a finire in  $x_{k+1}$ ;
- Scegliamo un'azione  $u_{k+1}$  **in base a Q**  $\Rightarrow u_{k+1} \in \arg \max_u Q(x_{k+1}, u)$ ;
- Aggiorniamo la stima di Q con TD

**Nota 4.26.**

Se  $x_{k+1}$  è uno stato terminale poniamo  $Q(x_{k+1}, u_{k+1}) = 0$ .

Per non ripetere sempre le stesse azioni esistono due metodologie:

- **exploring start:** si deve garantire che ciascuna coppia di stato e azione sia selezionata come la prima di un episodio un numero infinito di volte;
- **$\epsilon$ -greedy policy:** senza *exploring start* una policy deterministica ci potrebbe sempre alle stesse azioni. Per questo occorre introdurre una **policy non-deterministica**:
  - con probabilità  $1 - \epsilon$  scegliamo  $u_k$  in maniera greedy rispetto a Q;
  - con probabilità  $\epsilon$  scegliamo  $u_k$  casualmente;

**4.7.4 Off-policy TD:Q-learning**

Se conosciamo per ciasucno stato il valore di tutte le possibili azioni e non solo quelle che siamo esplorando:

$$Q(x_k, u_k) = Q(x_k, u_k) + \alpha \left[ R_{k+1} + \gamma \max_u Q(x_{k+1} + u_{k+1}) - Q(x_k, u_k) \right]$$

Non è implementabile *on-line* ma solo su dati precedentemente raccolti.

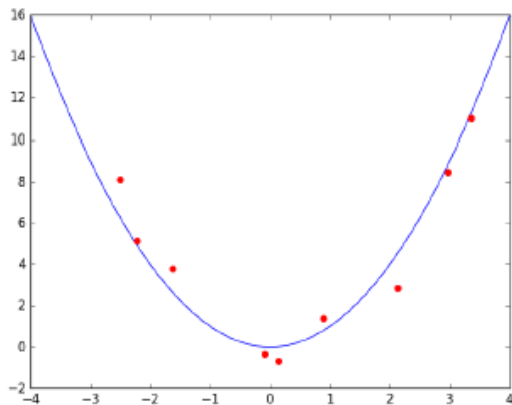
**4.7.5 RL con approssimatori funzionali: Deep Reinforcement Learning**

Data l'impossibilità di tenere traccia di un numero enorme di stato con i metodi precedenti, si utilizza un **approssimatore funzionale**:

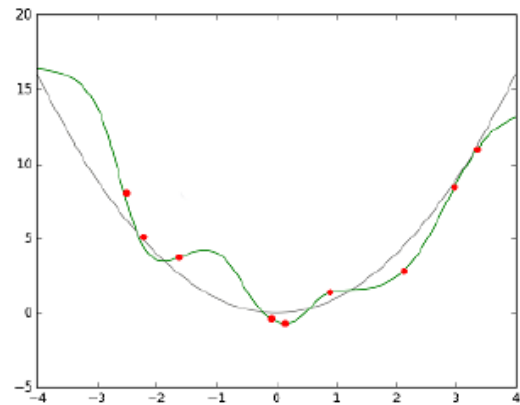
Data una collezione di coppia  $x_i, y_i = f(x_i) \quad i = 1, \dots, N$

Consideriamo una funzione  $s(v, w)$  dipendente da un numero ristretto di coefficienti  $w$

Vogliamo determinare il valori di  $w$  t.c.  $s(x_i, \bar{w})$  approssimi o interpoli  $y_i \quad \forall i \Rightarrow$  **Rete Neurale**



approssimazione



interpolazione





# Capitolo 5

## Riepilogo Teoria dei Sistemi

### 5.1 Sistema Dinamico Lineare

#### 5.1.1 Sistema Lineare a Tempo Continuo

Rappresenta il vincolo di ottimizzazione del nostro problema.

Un **sistema lineare a tempo continuo** è definito:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) & x(0) &= x_0 \quad t \in \mathbb{R} \\ y(t) &= Cx(t) + Du(t) & t &\in \mathbb{R}\end{aligned}$$

In cui,  $x(t) \in \mathbb{R}^n$  stato,  $u(t) \in \mathbb{R}^p$  controllo,  $y(t) \in \mathbb{R}^q$  uscita. Inoltre, definiamo:

$$A \in \mathbb{R}^{n \times n} := \text{matrice dinamica}$$

$$B \in \mathbb{R}^{n \times p} := \text{matrice di ingresso}$$

$$C \in \mathbb{R}^{q \times n} := \text{matrice di uscita}$$

$$D \in \mathbb{R}^{q \times p} := \text{legame diretto ingresso/uscita}$$

Inoltre, devono essere date le condizioni al contorno al tempo 0.

La soluzione **esiste** sempre ed è **unica** ed è della forma:

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau$$

In cui:

- $e^{At}x_0 := \text{evoluzione libera}$
- $\int_0^t e^{A(t-\tau)} Bu(\tau) d\tau := \text{evoluzione forzata}$

Verifichiamo ora che la  $x(t)$  è la soluzione facendone la derivata rispetto al tempo:

$$\begin{aligned}\dot{x}(t) &= Ae^{At}x_0 + Bu(t) + \int_0^t Ae^{A(t-\tau)} Bu(\tau) d\tau \\ &= A \left( e^{At}x_0 + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau \right) + Bu(t) \\ &= Ax(t) + Bu(t)\end{aligned}$$

Sostituendo  $t=0$ :

$$x(0) = e^{A0}x_0 + \int_0^0 e^{A(0-\tau)} Bu(\tau) d\tau = x_0$$

### 5.2 Proprietà di Stabilità ( con $u=0$ )

#### Definizione 5.1. Stato di Equilibrio

Uno stato  $x_e \in \mathbb{R}^n$  nel quale il sistema rimane indefinitamente in assenza di perturbazione è detto di **equilibrio** se:

$$x(0) = x_e \implies x(t) = x_e \quad \forall t \quad (\text{derivata nulla, } Ax_e = 0 \implies x_e \in \text{Ker}(A))$$

**Definizione 5.2.** *Stabilità di  $x_e$* 

Un punto di equilibrio si dice **stabile** se piccoli spostamenti di  $x_e$  comportano solo piccoli moti intorno all'equilibrio, e tanto più piccoli quanto più è piccolo lo scostamento iniziale.

$$\forall \varepsilon > 0, \exists \delta_\varepsilon > 0: \|x_0 - x_e\| < \delta_\varepsilon \implies \|x(t) - x_e\| < \varepsilon, \forall t \geq 0$$

**Definizione 5.3.** *Instabilità di  $x_e$* 

$$\exists \varepsilon > 0, \forall \delta_\varepsilon > 0: \|x_0 - x_e\| < \delta_\varepsilon \nRightarrow \|x(t) - x_e\| < \varepsilon, \forall t \geq 0$$

**Definizione 5.4.** *Attrattività di  $x_e$* 

Un punto di equilibrio  $x_e$  è detto **attrattivo** se i moti del sistema riportano lo stato nella configurazione di equilibrio.

$$\exists \delta_a > 0: \|x_0 - x_e\| < \delta_a \implies \lim_{t \rightarrow \infty} \|x(t) - x_e\| = 0$$

**Importante 5.5.**

Stabilità + Attrattività  $\rightarrow$  **Stabilità Asintotica**

## 5.3 Teoria di Lyapunov

La **teoria di Lyapunov** permette di stabilire le proprietà di stabilità di  $x_e$  senza dover calcolare tutti i moti perturbati.

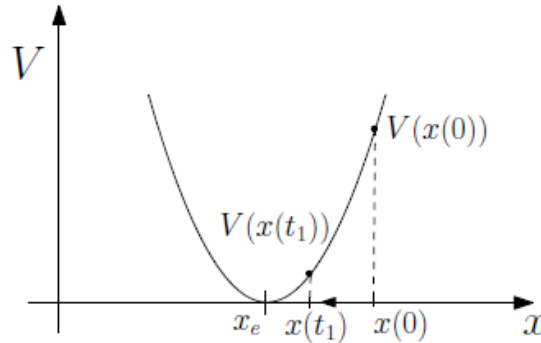
**Teorema 5.6.** *Metodo diretto di Lyapunov*

Per uno stato di equilibrio  $x_e$  cerchiamo di trovare una opportuna funzione di energia  $V(x)$  che assume il valore minimo in  $x_e$  e che venga dissipata lungo il moto del sistema.

Ciò significa valutare la funzione  $V$  lungo le traiettorie del sistema e vedere che essa decresce sempre. Quindi, se l'energia converge ad un minimo  $\implies x(t)$  converge a  $x_e$  punto di minimo.

In particolare, valutare la funzione di Lyapunov lungo le traiettorie del sistema, significa calcolare  $V(x(t))$ :

$$\dot{x} = -(x - x_e), V(x) = (x - x_e)^2$$



Mentre per verificare la decrescenza della funzione di Lyapunov occorre calcolare la derivata:

$$\dot{V}(x(t)) = \frac{d}{dt} V(x(t)) = \frac{\partial V}{\partial x} \dot{x} = \frac{\partial V}{\partial x} A x < 0$$

### 5.3.1 Metodo diretto di Lyapunov

**Definizione 5.7.** *Funzione di Lyapunov*

Supponiamo  $V(0) = 0, V(x) > 0$  per ogni  $x \neq 0$  e  $V \in C^1$  derivabile con continuità, allora:

- $\dot{V} < 0 \quad \forall x \neq 0$ , implica che  $x = 0$  è asintoticamente stabile;
- $\dot{V} \leq 0 \quad \forall x \neq 0$ , impliche che  $x = 0$  è stabile.

Cerchiamo di indovinare la struttura della funzione di Lyapunov  $V = x^T P x, P = P^T \succ 0$

$$\implies \dot{V} = \frac{\partial V}{\partial x} \dot{x} = 2x^T P(Ax) = x^T (PA + A^T P)x$$

Quindi,  $x=0$  è asintoticamente stabile se e solo se per ogni  $Q=Q^T$  esiste una  $P$  tale che:

$$PA + A^T P = -Q \quad \text{Equazione di Lyapunov}$$

Inoltre, l'equazione di Lyapunov ammette una soluzione  $P$  quale che sia  $Q$  se e solo se tutti gli **autovalori** di  $A$  sono a parte reale negativa.

### 5.3.2 Raggiungibilità e Controllabilità

Ci chiediamo se sia possibile imporre la proprietà di stabilità applicando al sistema un controllo  $u = Fx$

#### Definizione 5.8. Raggiungibilità

Uno stato  $\bar{x}$  si dice **raggiungibile** se esistono un istante finito  $\bar{t} > 0$  e una funzione di ingresso  $u(\cdot)$  tali che:

$$\bar{x} = \int_0^{\bar{t}} e^{A(t-\tau)} B u(\tau) d\tau$$

Il sistema si dice **raggiungibile** se tutti gli stati sono raggiungibili.

#### Definizione 5.9. Controllabilità

Uno stato  $\bar{x}$  si dice **controllabile** se esistono un istante finito  $\bar{t} > 0$  e una funzione di ingresso  $u(\cdot)$  tali che:

$$0 = e^{A\bar{t}} \bar{x} + \int_0^{\bar{t}} e^{A(t-\tau)} B u(\tau) d\tau$$

Il sistema si dice **controllabile** se tutti gli stati sono controllabili.

#### 5.3.2.1 Insieme degli stati raggiungibili

Consideriamo il sistema  $\dot{x} = Ax + Bu$  e definiamo la **matrice Gramiana di raggiungibilità**  $G(t) \in \mathbb{R}^{n \times n}$ :

$$G(t) = \int_0^t e^{A\tau} B B^T e^{A^T \tau} d\tau$$

Inoltre definiamo la **matrice di raggiungibilità**  $P \in \mathbb{R}^{n \times (np)}$  per ogni  $t \neq 0$ ,  $\text{im}(G) = \text{im}(P)$

#### Definizione 5.10. Stati raggiungibili

L'insieme  $X_r$  degli stati raggiungibili, che coincide con quello degli stati controllabili per sistemi a tempo continuo, è il sottospazio di  $\mathbb{R}^n$  definito con  $X_r = \text{Im}(G(t)) = \text{Im}(P)$ .

Quindi il sistema è raggiungibile/controllabile se e solo se  $\text{rank}(G(t)) = \text{rank}(P) = n$ .

#### 5.3.2.2 Assegnazione degli Autovalori

Se applichiamo al sistema controllato  $\dot{x} = Ax + Bu$  il feedback  $u = Fx(+v)$  otteniamo il sistema a ciclo chiuso:

$$\dot{x} = (A + BF)x(+Bv)$$

#### Importante 5.11. Assegnazione arbitraria degli autovalori

Gli autovalori di  $A + BF$  possono essere assegnati arbitrariamente tramite  $F$  se e solo se il sistema è controllabile/raggiungibile.

#### 5.3.2.3 Algoritmo di Mitter

##### Ipotesi

- Supponiamo che la matrice  $A$  possieda tutti gli autovalori distinti;
- Vogliamo spostare l'autovalore reale  $\lambda_a$  di  $A$  nell'autovalore reale  $\gamma_a$  di  $A + BF$

##### Algoritmo 5.1

1. Calcoliamo un autovettore sinistro  $v_a$  di  $A$  relativo a  $\lambda_a$  ovvero:

$$v_a \neq 0 \quad v_a^T A = \lambda_a v_a^T$$

2. Per la controllabilità:

$$v_a^T B \neq 0$$

3. Introduciamo l'incognita  $f_a \in \mathbb{R}^p$  e risolviamo:

$$v_a^T B f_a = \gamma_a - \lambda_a$$

4. Poniamo  $u = F_a x + v$  con  $F_a = f_a v_a^T$

**Verifica**

$$v_a^T (A + B F_a) = v_a^T A + v_a^T B f_a v_a^T = \lambda_a v_a^T + (\gamma_a - \lambda_a) v_a^T = \gamma_a v_a^T \text{ (per autovalore considerato)}$$

$$(A + B F_a) w_i = A w_i + B f_a v_a^T w_i = \lambda_i w_i \quad (\text{per tutti gli altri autovalori})$$

Per esempi:

Consideriamo le matrici

$$A = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -8 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

verifichiamo se la stabilizzazione sia già risolta con  $u = 0$ , calcolando gli autovalori di  $A$

$$\begin{aligned} \det(A - \lambda I) &= \det \left( \begin{bmatrix} -1-\lambda & 0 & 1 \\ 1 & -8-\lambda & 0 \\ 0 & 0 & 2-\lambda \end{bmatrix} \right) = (2-\lambda) \det \left( \begin{bmatrix} -1-\lambda & 0 \\ 1 & -8-\lambda \end{bmatrix} \right) \\ &= (2-\lambda)(-1-\lambda)(-8-\lambda) \Rightarrow \text{autovalori} := \{2, -1, -8\} \end{aligned}$$

$\Rightarrow$  dobbiamo spostare l'autovalore  $\lambda_a = 2$  in, ad esempio,  $\gamma_a = -6$

verifichiamo se il sistema è controllabile

$\Rightarrow$  dobbiamo costruire la matrice  $\mathcal{P} = [B \quad AB \quad A^2B]$  e verificare che abbia rango pari a 3

costruiamo la matrice  $\mathcal{P}$  per passi (aggiungendo colonne) e verifichiamo il rango via via:

$$\mathcal{P}_1 = B = \begin{bmatrix} 2 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow \text{può avere rango 3?}$$

$$\mathcal{P}_2 = [B \quad AB] = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 1 & 2 & -8 \\ 1 & 0 & 2 & 0 \end{bmatrix} \Rightarrow \text{scegliamo le prime 3 colonne...}$$

$$\det \left( \begin{bmatrix} 2 & 0 & -1 \\ 0 & 1 & 2 \\ 1 & 0 & 2 \end{bmatrix} \right) = \begin{bmatrix} 2 & 0 & -1 & | & 2 & 0 \\ 0 & 1 & 2 & | & 0 & 1 \\ 1 & 0 & 2 & | & 1 & 0 \end{bmatrix} = \underbrace{4 + 0 + 0}_{\text{diagonali}} - \overbrace{(0 + 0 - 1)}^{\text{antidiagonali}} = 5 \neq 0$$

(applicando la regola di Sarrus...)

$\Rightarrow$  Il sistema è controllabile!

Calcoliamo un autovettore sinistro di  $A$  relativo a  $\lambda_a = 2$ , incognita  $v_a^T = [a, b, c]$

$$[a, b, c] \begin{bmatrix} -1 & 0 & 1 \\ 1 & -8 & 0 \\ 0 & 0 & 2 \end{bmatrix} \Rightarrow \begin{cases} -3a + b = 0 \\ -10b = 0 \\ a = 0 \end{cases} \Rightarrow v_a = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Verifichiamo che  $v_a^T B = [1, 0] \neq 0$

Introduciamo l'incognita  $f_a = [f_1, f_2]^T$  e risolviamo

$$[1, 0] \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = -6 - 2 = -8 \Rightarrow \text{ad esempio } f_a = \begin{bmatrix} -8 \\ 0 \end{bmatrix}$$

Selezioniamo  $F = f_a v_a^T$

$$\begin{bmatrix} -8 \\ 0 \end{bmatrix} [0, 0, 1] = \begin{bmatrix} 0 & 0 & -8 \\ 0 & 0 & 0 \end{bmatrix}$$

$\Rightarrow$  calcolare autovalori di  $A + BF \dots$

### 5.3.3 Proprietà di Osservabilità

Un sistema si dice osservabile dall'uscita  $y$  se esiste un intervallo finito di tempo  $[0, \bar{t}]$  tale che, conoscendo senza errori  $y(t)$  e  $u(t)$  per  $t \in [0, \bar{t}]$  e le matrici  $A, B, C$  e  $D$ , risulti possibile in ogni caso individuare **univocamente** il valore dello stato all'inizio di tale intervallo ovvero  $x(0)$

**Matrice Gramiana di Osservabilità/Matrice di Osservabilità**

$$L(t) = \int_0^t e^{A^T \tau} C^T C e^{A \tau} d\tau \quad O = \begin{pmatrix} C \\ C A \\ C A^2 \\ \vdots \\ C A^{n-1} \end{pmatrix}$$

**Definizione 5.12. Stati Inosservabili**

L'insieme degli stati inosservabili  $X_i$  è  $X_i = \ker(L(t)) = \ker(O)$ . Quindi il sistema è osservabile se e solo se  $\text{rank}(L(t)) = \text{rank}(O) = n$ .



# Capitolo 6

## Principio di Ottimalità e Equazione di Hamilton-Jacobi-Bellman

### 6.1 Introduzione

Vogliamo definire un problema di controllo ottimo o ottimizzazione dinamica su un orizzonte finito a tempo continuo. Il generico sistema può essere espresso come nel capitolo precedente. Per ottenere una soluzione unica abbiamo necessità di avere una condizione al contorno per  $t = t_0$ .

### 6.2 Definizione di un Problema di Controllo Ottimo

Consideriamo un sistema non-lineare a tempo continuo:

$$\dot{x}(t) = f(x(t), u(t), t) \quad x(t_0) = x_0 \quad f(0, 0, t) = 0 \quad \forall t$$

Supponiamo che  $f$  sia **localmente Lipschitz** rispetto a  $x$  e continua rispetto ad  $u$  e  $t$ . Allora esiste una soluzione ed è unica.

#### Definizione 6.1. Funzione Lipschitz

Una funzione  $g: \mathbb{R}^n \rightarrow \mathbb{R}^n$  è **localmente Lipschitz** se per ogni  $x$  esiste un intorno  $U$  di  $x$  e una costante  $L$  tale che:

$$\|g(x) - g(y)\| \leq L \|x - y\|$$

per ogni  $y \in U$ .

L'obiettivo è **controllare** il sistema in un intervallo di tempo  $[t_0, T]$  fissato a priori, ovvero determinare una funzione  $u^*(t), t \in [t_0, T]$  tale che la corrispondente soluzione  $x^*(t)$  minimizzi l'**indice di costo**:

$$J(u) = \int_{t_0}^T l(x(t), u(t), t) dt + m(x(T)) \quad \textbf{Problema di Bolza}$$

#### Nota 6.2.

- $\int_{t_0}^T l(x(t), u(t), t) dt :=$  **termine di Lagrange**;
- $m(x(T)) :=$  **termine di Mayer**;
- **costo corrente** :=  $l: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}, l(0, 0, t) = 0 \quad \forall t$  di classe  $C^2$
- **costo terminale** :=  $m: \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}, m(0) = 0$

Limitiamo la scelta dei controlli ammissibili a **retro-azioni** dallo stato  $x$  al tempo  $t$ :

$$u = \varphi(x, t) \quad \varphi: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m$$

## 6.3 Principio di Ottimalità

### **Teorema 6.3.** *Principio di Ottimalità*

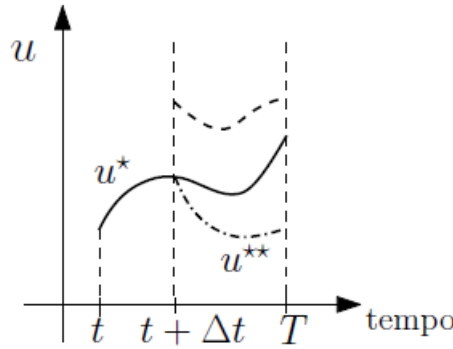
Consideriamo un Problema di Bolza. Se  $u^*$  è ottima sull'intervallo  $[t, T]$  a partire dallo stato  $x(t)$ , allora  $u^*(\tau), \tau \in [t + \Delta t, T]$  è necessariamente ottima per un problema di bolza ristretto all'intervallo  $[t + \Delta t, T]$  a partire da  $x^*(t + \Delta t)$  per ogni  $\Delta t$  tale che  $0 < \Delta t \leq T - t$ .

### **Dimostrazione.**

Supponiamo **per assurdo** che esista  $u^{**}$  che fornisce un valore minore dell'indice di costo del problema di Bolza ristretto:

$$\tilde{J}(u) = \int_{t+\Delta t}^T l(x(\tau), u(\tau), \tau) d\tau + m(x(T))$$

rispetto a  $u^*$  in  $[t + \Delta t, T]$ . Quindi  $\tilde{J}(u^{**}) < \tilde{J}(u^*)$ . Se questa ipotesi è vera



definiamo un nuovo controllo  $\hat{u}$ :

$$\hat{u}(\tau) = \begin{cases} u^*(\tau) & \text{se } t \leq \tau \leq t + \Delta t \\ u^{**}(\tau) & \text{se } t + \Delta t \leq \tau \leq T \end{cases}$$

Allora, sull'intero intervallo  $[t, T]$  abbiamo:

$$\begin{aligned} J(\hat{u}) &= \int_t^T l(\hat{x}(s), \hat{u}(s), s) ds + m(\hat{x}(T)) \\ &= \int_t^{t+\Delta t} l(x^*(s), u^*(s), s) ds + \int_{t+\Delta t}^T l(x^{**}(s), u^{**}(s), s) ds + m(x^{**}(T)) \\ &< \int_t^{t+\Delta t} l(x^*(s), u^*(s), s) ds + \int_{t+\Delta t}^T l(x^*(s), u^*(s), s) ds + m(x^*(T)) = J(u^*) \end{aligned}$$

### **Nota 6.4.**

- $\int_{t+\Delta t}^T l(x^{**}(s), u^{**}(s), s) ds + m(x^{**}(T)) = \tilde{J}(u^{**})$
- $\int_{t+\Delta t}^T l(x^*(s), u^*(s), s) ds + m(x^*(T)) = \tilde{J}(u^*)$

Abbiamo così contraddetto l'ipotesi che  $u^*$  sia la soluzione ottima su  $[t, T]$

□

## 6.4 Commenti sul Principio di Ottimalità

### 6.4.1 Analisi

**Analisi** di una soluzione ottima:



Per ogni istante di tempi intermedio  $\bar{t} \in [t, T]$ , se  $x^*(t)$  è lo stato raggiunto dalla traiettoria ottima, allora le decisioni rimanenti da  $\bar{t}$  a  $T$  devono rappresentare una strategia ottima a partire da  $x^*(t)$

### 6.4.2 Sintesi

- Dividiamo l'intervallo  $[t, T]$  nei due sotto-intervalli  $[t, t + \Delta t]$  e  $[t + \Delta t, T]$ ;
- Supponiamo di avere uno strumento che ci restituisce il costo ottimo da tutti i possibili stati raggiunti all'istante  $t + \Delta t$ ;
- Grazie al principio di ottimalità possiamo preoccuparci solamente della parte di analisi poiché lo strumento ci fornisce le rimanenti decisioni ottime che devono necessariamente coincidere con quelle che avrei avuto minimizzando su tutto l'intervallo. Ciò viene realizzato tramite la **funzione valore**.

## 6.5 Funzione Valore di un Problema di Controllo Ottimo

### 6.5.1 Funzione Valore

**Definizione 6.5.** *Funzione Valore*  
Definiamo la **funzione valore**:

$$V^*(x, t), V: \mathbb{R}^n \times \mathbb{R} \longrightarrow \mathbb{R}$$

Consideriamo un problema di Bolza:

$$\begin{cases} \min_u \{J(u)\} = \min_u \left\{ \int_{t_0}^T l(x(\tau), u(\tau), \tau) d\tau + m(x(T)) \right\} \\ \dot{x} = f(x, u, t) \\ x(t_0) = x_0 \end{cases}$$

Data la condizione iniziale  $x_0$  e il tempo iniziale  $t_0$  la funzione valore restituisce il minimo valore dell'indice di costo, ovvero:

$$V^*(x_0, t_0) = \min_u \left\{ \int_{t_0}^T l(x(\tau), u(\tau), \tau) d\tau + m(x(T)) \right\}$$

Vogliamo scrivere un'equazione che caratterizzi la funzione  $V^*$ .

## 6.6 Condizioni Necessarie di Ottimalità - Equazione HJB

Cerchiamo di caratterizzare un'equazione per la funzione valore:

$$\begin{aligned} V^*(x(t), t) &= \min_{u(\tau), \tau \in [t, T]} \left\{ \int_t^T l(x(s), u(s), s) ds + m(x(T)) \right\} \\ &= \min_{u(\tau), \tau \in [t, T]} \left\{ \int_t^{t+\Delta t} l(x(s), u(s), s) ds + \int_{t+\Delta t}^T l(x(s), u(s), s) ds + m(x(T)) \right\} \\ &= \min_{u(\tau), \tau \in [t, t+\Delta t]} \left\{ \int_t^{t+\Delta t} l(x(s), u(s), s) ds + V^*(x(t+\Delta t), t+\Delta t) \right\} \end{aligned}$$

**Nota 6.6.**

La parte rossa viene ottenuta tramite l'utilizzo della definizione del principio di ottimalità. La conseguenza è che abbiamo ristretto i tempi.

Cerchiamo di ricavare un'equazione differenziale per il problema proposto. A tale scopo occorre derivare la funzione valore e utilizziamo l'espansione di Taylor.

### 6.6.0.1 Riepilogo Espansione di Taylor

Consideriamo una funzione derivabile con continuità  $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ . Allora l'espansione in serie di Taylor al primo ordine intorno al punto  $x = a$ :

$$f(x) = f(a) + \frac{\partial f}{\partial x}|_{x=a}(x-a) + o(x-a)$$

in cui  $o(x-a)$  è un infinitesimo di ordine superiore a  $(x-a)$  ovvero  $\lim_{x \rightarrow a} \frac{o(x-a)}{\|x-a\|} = 0$ .

Consideriamo le seguenti espansioni al primo ordine:

- $x(t + \Delta t)$  intorno a  $t$ :

$$x(t + \Delta t) = x(t) + \dot{x}\Delta t + o(\Delta t)$$

- $V^*(x(t + \Delta t), t + \Delta t)$  intorno a  $(x(t), t)$

$$V^*(x(t + \Delta t), t + \Delta t) = V^*(x(t), t)[x(t + \Delta t) - x(t)] + \frac{\partial V^*}{\partial t}(x(t), t)\Delta t$$

- $\int_t^{t+\Delta t} l(x(s), u(s), s)ds$  intorno a  $t$ :

$$\int_t^{t+\Delta t} l(x(s), u(s), s)ds = l(x(t), u(t), t)\Delta t + o(\Delta t)$$

In cui il termine:

$$\int_t^t l(x(s), u(s), s)ds = 0$$

### 6.6.1 Equazione di Hamilton-Jacobi-Bellman

Sostituiamo gli sviluppi in serie precedentemente calcolati, dividiamo per  $\Delta t$ , effettuiamo il limite per  $\Delta t \rightarrow 0^+$ :

$$0 = \min_{u(t)} \left\{ l(x(t), u(t), t) + \frac{\partial V^*}{\partial t}(x(t), t) + \frac{\partial V^*}{\partial x}(x(t), t) f(x(t), u(t), t) \right\}$$

Dato che dobbiamo determinare la soluzione per ogni  $t$  e per ogni  $x(t)$ , possiamo finalmente scrivere l'**equazione di Hamilton-Jacobi-Bellman**:

$$\begin{cases} -\frac{\partial V^*}{\partial t}(x(t), t) = \min_{u(t)} \left\{ l(x(t), u(t), t) + \frac{\partial V^*}{\partial x}(x(t), t) f(x(t), u(t), t) \right\} & \forall x \in \mathbb{R}^n, t \in [t_0, T] \\ V^*(x, T) = m(x) \end{cases}$$

## 6.7 Condizioni Sufficienti di Ottimalità

Vogliamo dimostrare che l'equazione di Hamilton-Jacobi-Bellman fornisce anche condizioni sufficienti di ottimalità:

“Se una funzione  $\hat{V}$  risolve HJB, allora è la funzione valore”

### Teorema 6.7.

Supponiamo che la funzione  $\hat{V}: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ , derivabile con continuità, risolve la HJB, ovvero:

$$\begin{cases} -\frac{\partial \hat{V}}{\partial t}(x, t) = \min_u \left\{ l(x, u, t) + \frac{\partial \hat{V}}{\partial x}(x, t) f(x, u, t) \right\} & \forall x \in \mathbb{R}^n, t \in [t_0, T] \\ \hat{V}(x, T) = m(x), \forall x \in \mathbb{R}^n \end{cases}$$

e supponiamo che il minimo sia raggiunto per:

$$\hat{u} = \arg \min_u \left\{ l(x, u, t) + \frac{\partial \hat{V}}{\partial x}(x, t) f(x, u, t) \right\} \quad \forall x, \forall t$$

Allora  $\hat{V}(x_0, t_0)$  è il **costo ottimo** e  $\hat{u}(t)$  è il **controllo ottimo**.

**Dimostrazione.**

Se  $\hat{V}$  risolve HJB e  $\hat{u}$  raggiunge il minimo del termine di destra di HJB, allora lungo la traiettoria  $\hat{x}(t)$  corrispondente al controllo  $\hat{u}(t)$ ,  $t \geq 0$  a partire dalla condizione iniziale  $\hat{x}(t_0) = x_0$ , si ha:

$$-\frac{\partial \hat{V}}{\partial t}(\hat{x}(t), t) = l(\hat{x}(t), \hat{u}(t), t) + \frac{\partial \hat{V}}{\partial x}(\hat{x}(t), t) f(\hat{x}(t), t) f(\hat{x}(t), \hat{u}(t), t) \quad (1)$$

Ricordando poi che:

$$\frac{d}{dt} \hat{V}(\hat{x}(t), t) = \frac{\partial}{\partial x} \hat{V}(\hat{x}(t), t) f(\hat{x}(t), \hat{u}(t), t) + \frac{\partial}{\partial t} \hat{V}(\hat{x}(t), t)$$

L'equazione (1) può essere scritta come:

$$0 = l(\hat{x}(t), \hat{u}(t), t) + \frac{d}{dt} \hat{V}(\hat{x}(t), t)$$

A questo punto, integrando:

$$0 = \int_{t_0}^T l(\hat{x}(t), \hat{u}(t), t) dt + \int_{t_0}^T \frac{d}{dt} \hat{V}(\hat{x}(t), t) dt$$

Quindi:

$$\begin{aligned} 0 &= \int_{t_0}^T l(\hat{x}(t), \hat{u}(t), t) dt + \int_{t_0}^T \frac{d}{dt} \hat{V}(\hat{x}(t), t) dt \\ &= \int_{t_0}^T l(\hat{x}(t), \hat{u}(t), t) dt + \hat{V}(\hat{x}(T), T) - \hat{V}(\hat{x}(t_0), t_0) \\ &= J(\hat{u}) - \hat{V}(\hat{x}(t_0), t_0) \end{aligned}$$

Ottenuto grazie al teorema fondamentale del calcolo integrale e alle condizioni al contorno.

Abbiamo dunque dimostrato che:

$$\hat{V}(x_0, t_0) = J(\hat{u})$$

A questo punto dobbiamo confrontare questo costo con quello di qualsiasi altra  $u$ . A tale scopo ripetiamo i passaggi di prima con un generico  $u$ , con  $x(t)$  traiettoria corrispondente a  $u(t)$  da  $x(t_0) = x_0$ :

$$-\frac{\partial \hat{V}}{\partial t}(x(t), t) \leq l(x(t), u(t), t) + \frac{\partial \hat{V}}{\partial x}(x(t), u(t), t) \quad (2)$$

L'equazione (2) può essere scritta come:

$$0 \leq l(x(t), u(t), t) + \frac{d}{dt} \hat{V}(x(t), t)$$

Integrando:

$$\begin{aligned} 0 &\leq \int_{t_0}^T l(x(t), u(t), t) dt + \int_{t_0}^T \frac{d}{dt} \hat{V}(x(t), t) dt \\ &= \int_{t_0}^T l(x(t), u(t), t) dt + \hat{V}(x(T), T) - \hat{V}(x_0, t_0) \end{aligned}$$

Quindi:

$$\hat{V}(x_0, t_0) \leq J(u)$$

che combinato con la relazione precedente per  $\hat{u}$ :

$$J(\hat{u}) \leq J(u) \quad \forall u$$

□

## 6.8 Problema di Controllo Ottimo in presenza di vincoli sul controllo

Consideriamo il caso in cui sono presenti vincoli sul controllo:

$$\begin{cases} \min_u \int_{t_0}^T l(x(\tau), u(\tau), \tau) d\tau + m(x(T)) \\ \dot{x}(t) = f(x(t), u(t), t) & x(t_0) = x_0 \\ u(t) \in U & t \in [t_0, T] \end{cases}$$

Le condizioni necessarie e sufficienti rimangono valie se consideriamo:

$$\begin{cases} -\frac{\partial \hat{V}}{\partial t}(x, t) = \min_{u \in U} \left\{ l(x, u, t) + \frac{\partial \hat{V}}{\partial x}(x, t) f(x, u, t) \right\} & \forall x \in \mathbb{R}^n, t \in [t_0, T] \\ \hat{V}(x, T) = m(x), \forall x \in \mathbb{R}^n \end{cases}$$

## 6.9 Problema di Controllo Ottimo ad Orizzonte Finito

Supponiamo che sia il sistema che il costo corrente siano **stazionari**. Quindi:

$$f(x, u), l(x, u) = 0$$

$$l(x, u), l(0, 0) = 0$$

Supponiamo che non sia presente un **costo terminale**,  $m(x) = 0 \quad \forall x$ .

Dunque, consideriamo il problema di controllo ottimo:

$$\begin{cases} \min_u \int_{t_0}^{\infty} l(x(\tau), u(\tau), \tau) d\tau \\ \dot{x}(t) = f(x(t), u(t)) & x(t_0) = x_0 \end{cases}$$

Allora il minimo costo ottenibile non dipende dal tempo iniziale e l'equazione di HBJ diventa:

$$\begin{cases} 0 = \min_u \left\{ l(x, u) + \frac{\partial V^*}{\partial x}(x) f(x, u) \right\} & \forall x \in \mathbb{R}^n \\ 0 = V^*(0) \end{cases}$$

# Capitolo 7

## Problema del Regolatore Lineare-Quadratico (LQR)

### 7.1 Problema del Regolatore Lineare-Quadratico (LQR)

Consideriamo un problema di controllo ottimo su orizzonte finito descritto da un **sistema lineare** e indice di **costo quadratico**:

$$\min_u J(u) = \left\{ \frac{1}{2} \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt + \frac{1}{2} x(T)^T M x(T) \right\}$$

$$\dot{x}(t) = A x(t) + B u(t) \quad x(0) = x_0$$

con  $Q = Q^T \succcurlyeq 0, M = M^T \succcurlyeq 0, R = R^T \succcurlyeq 0$ .

L'equazione di HJB scritta per il problema di controllo ottimo LQ diventa:

$$\begin{cases} -\frac{\partial V^*}{\partial t}(x, t) = \min_u \left\{ \frac{1}{2} x^T Q x + \frac{1}{2} u^T R u + \frac{\partial V^*}{\partial x}(x, t) (A x + B u) \right\} & \forall x \in \mathbb{R}^n, t \in [0, T] \\ V^*(x, T) = \frac{1}{2} x^T M x & \forall x \in \mathbb{R}^n \end{cases}$$

### 7.2 Minimizzazione dell'equazione di HJB

Il primo passo per la risoluzione dell'equazione consiste nel minimizzare il termine di destra rispetto ad  $u$ . Dato che quest'ultimo è una funzione **strettamente convessa** in  $u$ , possiamo effettuare la derivata rispetto ad  $u$  e imporre pari a zero:

$$u^T R + \frac{\partial V^*}{\partial x}(x, t) B = 0 \quad \implies u^* = -R^{-1} B^T \left( \frac{\partial V^*}{\partial x}(x, t) \right)^T$$

Tuttavia dobbiamo conoscere l'espressione della funzione valore.

### 7.3 Calcolo della Funzione Valore

Dato che al tempo finale la funzione valore deve soddisfare le condizioni al contorno:

$$V^*(x, T) = \frac{1}{2} x^T M x \quad \forall x \in \mathbb{R}^n$$

Proviamo ad **ipotizzare** una **forma quadratica** per la funzione  $V^*$ , ovvero:

$$V^* = \frac{1}{2} x^T P(t) x$$

con  $P(\cdot): \mathbb{R} \longrightarrow \mathbb{R}^{n \times n}, P(t) = P(t)^T \succcurlyeq 0, t \in [0, T]$  che dobbiamo determinare.

Per sostituire la funzione  $V^*$  nell'equazione di HJB, calcoliamo le derivate parziali:

$$\frac{\partial V^*}{\partial t} = \frac{1}{2} x^T \dot{P}(t) x$$

$$\frac{\partial V^*}{\partial x} = x^T P(t)$$

$$\Rightarrow u^* = -R^{-1} B^T P(t) x \quad (\text{è lineare e non stazionario})$$

## 7.4 Equazione Differenziale di Riccati

Sostituendo in HJB le derivate parziali di  $V^*$  e il controllo  $u^*$ , otteniamo:

$$-\frac{1}{2} x^T \dot{P} x = \frac{1}{2} x^T Q x + x^T P(t) A x - \frac{1}{2} x^T P(t) B R^{-1} B^T P(t) x$$

Dal momento che l'equazione deve valere per ogni  $t \in [0, T]$  e per ogni  $x \in \mathbb{R}^n$ , la matrice  $P(t)$  deve essere tale da soddisfare la seguente **equazione alle derivate ordinarie** (**Equazione differenziale di Riccati**):

$$-\dot{P}(t) = P(t) A + A^T P(t) - P(t) B R^{-1} B^T P(t) + Q \quad P(T) = M$$

**Nota 7.1.**

- Equazione alle derivate ordinarie da integrare all'indietro a partire dalla condizione terminale  $P(T) = M$ ;
- La DRE è una equazione quadratica in  $P(t)$ ;

## 7.5 Esistenza Globale della Soluzione DRE

### Esistenza Locale

Sappiamo che la soluzione  $P(t)$  esiste per tempi sufficientemente vicini a  $T$ , integrando all'indietro.

#### Esistenza Globale in $[0, T]$

Supponiamo che  $\frac{1}{2} x^T P(t) x$  sia la funzione valore, allora la soluzione dell'equazione differenziale di Riccati esiste per ogni  $t$ .

#### Dimostrazione.

Supponiamo per assurdo che esista un istante  $\hat{t} < T$  tale che  $P(t)$  esiste sull'intervallo aperto  $(\hat{t}, T]$  ma un suo elemento  $p_{i,j}(t)$  diventa **illimitato** per  $t$  che converge a  $\hat{t}$  da destra:

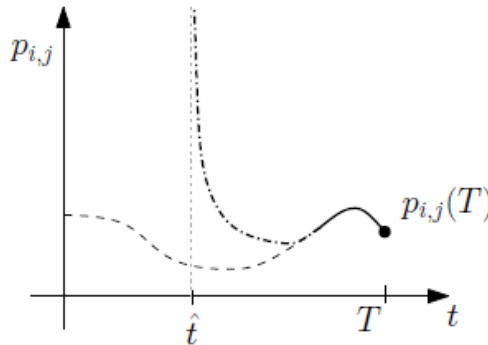


Figura 7.1.

Distinguiamo due casi:

1. L'elemento  $p_{i,j}$  è **fuori dalla diagonale** di  $P(t)$ :

Dal momento che  $P(t)$  è semidefinita positiva tutti i suoi minori principali sono maggiori o uguali a zero.

Consideriamo il minore di ordine 2 ottenuto scegliendo proprio le righe e colonne nell'insieme  $\{i, j\}^2$ :

$$\det \begin{pmatrix} p_{i,i}(t) & p_{i,j}(t) \\ p_{i,j}(t) & p_{j,j}(t) \end{pmatrix} = p_{i,i}(t)p_{j,j}(t) - p_{i,j}(t)^2$$

Otteniamo una contraddizione poiché il minore diventa negativo per  $t$  che tende a  $\hat{t}$  da destra, perché  $p_{i,j}$  diventa illimitato mentre  $p_{i,i}$  e  $p_{j,j}$  restano limitati.

2. L'elemento  $p_{i,j}$  **appartiene alla diagonale**:

Consideriamo l' $i$ -esimo elemento della base canonica di  $\mathbb{R}^n$  ovvero:

$$\eta_i = [0, \dots, 0, 1, 0, \dots, 0]$$

Il costo ottimo da  $\eta_i$  al tempo  $t$  dunque dovrebbe essere:

$$V^*(\eta_i, t) = \frac{1}{2} \eta_i^T P(t) \eta_i = \frac{1}{2} P_{i,i}(t) \longrightarrow \infty \text{ per } t \rightarrow \hat{t}$$

Ma il costo ottimo non può essere maggiore del costo ottimo di  $u=0$ . Allora:

$$u=0, \dot{x}=Ax \rightarrow x(\tau) = e^{A(\tau-t)} \eta_i$$

E quindi il costo diventa:

$$J(0) = \frac{1}{2} \int_t^T \eta_i^T e^{A(\tau-t)} \eta_i d\tau + \frac{1}{2} \eta_i^T e^{A^T(T-t)} M e^{A(T-t)} \eta_i < \infty$$

per ogni  $t$ , visto che si tratta di un integrale di una funzione continua su un intervallo limitato.

□

## 7.6 Considerazioni Computazionali e Sistema Hamiltoniano

L'equazione matriciale Differenziale di Riccati consiste di  $\frac{n(n+1)}{2}$  equazioni differenziali ordinali non lineare. Cerchiamo, quindi, di riscrivere le equazioni in una forma diversa che permetta di avere più equazioni lineari.

### **Definizione 7.2.** *Sistema Hamiltoniano*

La soluzione dell'equazione differenziale di Riccati può essere ottenuta come  $P(t) = Y(t) X(t)^{-1}$  con  $Y \in \mathbb{R}^{n \times n}$  e  $X \in \mathbb{R}^{n \times n}$  soluzioni di:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} A & -B R^{-1} B^T \\ -Q & -A^T \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$$

Con condizioni al contorno:

$$\begin{pmatrix} X(T) \\ Y(T) \end{pmatrix} = \begin{pmatrix} I \\ M \end{pmatrix}$$

### **Dimostrazione.**

- Notiamo che:

$$\frac{d}{dt}(YX^{-1}) = Y \frac{dX^{-1}}{dt} + \frac{dY}{dt} X^{-1}$$

- Derivando la relazione  $X(t) X(t)^{-1} = I$  otteniamo:

$$X(t) \frac{dX(t)^{-1}}{dt} + \frac{dX(t)}{dt} X(t)^{-1} = 0 \implies \frac{dX(t)}{dt} = -X(t) \frac{dX(t)^{-1}}{dt} X(t)^{-1}$$

Quindi:

$$\frac{d}{dt}(YX^{-1}) = -YX^{-1}\dot{X}X^{-1} + \dot{Y}X^{-1}$$

$$P = YX^{-1}$$

$$= YX^{-1}A + YX^{-1}BR^{-1}B^TYX^{-1} - Q - A^TY - 1$$

Inoltre:

$$P(T) = Y(T)X(T)^{-1} = M$$

□



# Capitolo 8

## Matrice Hamiltoniana e Soluzione Dell'Equazione differenziale di Riccati

### 8.1 Proprietà della matrice Hamiltoniana

**Definizione 8.1.** *Sistema Hamiltoniano*

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} \quad \begin{pmatrix} X(T) \\ Y(T) \end{pmatrix} = \begin{pmatrix} I \\ M \end{pmatrix}$$

**Proprietà della matrice Hamiltoniana H**

- In generale se  $\lambda$  è un autovalore di una matrice  $A$  allora  $-\lambda$  è un autovalore di  $-A$ ;
- In generale  $A$  e  $A^T$  hanno lo stesso insieme di autovalori;
- $H^T$  e  $-H$  sono matrice **simili**, ovvero:

$$-H = JH^TJ^{-1} \quad \text{con } J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

- **Simmetrica Quadrantale:** se  $\lambda$  è un autovalore di  $H$  anche  $-\lambda$  è un autovalore di  $H$ .

### 8.2 Riepilogo sulla forma di Jordan

**Definizione 8.2.** *Matrice diagonalizzabile*

Una matrice  $A \in \mathbb{R}^{n \times n}$  è **diagonalizzabile** se è simile ad una matrice diagonale, ovvero esiste  $T$  non singolare e  $\Lambda$  diagonale tali che  $\Lambda = TAT^{-1}$ .

**Definizione 8.3.** *Diagonalizzabilità*

$A$  è diagonalizzabile se e solo se per ogni suo autovalore la molteplicità algebrica coincide con la molteplicità geometrica.

**Definizione 8.4.** *Forma di Jordan*

La forma di Jordan di una matrice quadrata  $A$  è una matrice triangolare  $J$  simile ad  $A$  che ha una struttura il più possibile simile ad  $A$  che ha una struttura il più possibile vicina ad una matrice diagonale.

$$J = \begin{pmatrix} J_1 & 0 & \dots & 0 \\ 0 & J_2 & \dots & 0 \\ \vdots & & \ddots & \\ 0 & \dots & & J_p \end{pmatrix} \quad \text{con } J_i = \begin{pmatrix} \lambda_i & 1 & \dots & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{pmatrix}$$

La matrice  $T$  è data da una collezione di **autovettori generalizzati**.

Supponiamo che  $\lambda$  sia autovalore di  $A$  con molteplicità algebrica  $k \geq 1$ , allora  $v$  è autovettore generalizzato se  $(A - \lambda I)^k v = 0$ ,  $(A - \lambda I)^{k-1} v \neq 0$ , ne esistono  $k-1$ . Allora definiamo una **catena di autovettori generalizzati** come  $(A - \lambda I)v_i = v_{i-1} \quad i = 2, \dots, k$ .

### 8.3 Soluzione di DRE tramite matrice Hamiltoniana

Supponiamo che  $H$  non abbia autovalori puramente immaginari, dunque  $n$  a parte reale positiva (instabili) e  $n$  a parte reale negativa (stabili).

Esiste sempre una trasformazione non-singolare  $U$  tale che:

$$U^{-1} H U = \begin{pmatrix} \Lambda_s & 0 \\ 0 & \Lambda_u \end{pmatrix}$$

in cui  $\Lambda_s/\Lambda_u$  raccolgono tutti i blocchi di Jordan associati ad autovalori **stabili/instabili**.

Partizioniamo, di conseguenza, anche  $U$  in maniera coerente come:

$$U = \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix}$$

dove:

$$\begin{pmatrix} U_{11} \\ U_{21} \end{pmatrix} / \begin{pmatrix} U_{12} \\ U_{22} \end{pmatrix}$$

hanno come colonne gli autovettori generalizzati di  $H$  corrispondenti agli autovalori **stabili/instabili**.

Ora, consideriamo il **cambio di coordinante** per il sistema Hamiltoniano:

$$\begin{pmatrix} X \\ Y \end{pmatrix} = U \begin{pmatrix} \hat{X} \\ \hat{Y} \end{pmatrix} \Rightarrow \begin{pmatrix} \hat{X} \\ \hat{Y} \end{pmatrix} = U^{-1} \begin{pmatrix} X \\ Y \end{pmatrix}$$

Il sistema nelle nuove coordinate diventa:

$$\begin{pmatrix} \dot{\hat{X}} \\ \dot{\hat{Y}} \end{pmatrix} = U^{-1} \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = U^{-1} H \begin{pmatrix} X \\ Y \end{pmatrix} = U^{-1} H U \begin{pmatrix} \hat{X} \\ \hat{Y} \end{pmatrix} = \begin{pmatrix} \Lambda_s & 0 \\ 0 & \Lambda_u \end{pmatrix} \begin{pmatrix} \hat{X} \\ \hat{Y} \end{pmatrix}$$

Questo è un sistema **lineare e disaccoppiato** per  $\hat{X}$  e  $\hat{Y}$ .

La soluzione al tempo  $T$  può essere trovata nel seguente modo:

$$\begin{aligned} \hat{X}(T) &= e^{\Lambda_s(T-t)} \hat{X}(t) \implies \hat{X}(t) = e^{-\Lambda_s(T-t)} \hat{X}(T) \\ \hat{Y}(T) &= e^{\Lambda_u(T-t)} \hat{Y}(t) \implies \hat{Y}(t) = e^{-\Lambda_u(T-t)} \hat{Y}(T) \end{aligned}$$

Ora imponiamo le condizioni al contorno ottenendo:

$$\begin{pmatrix} X(T) \\ Y(T) \end{pmatrix} = \begin{pmatrix} I \\ M \end{pmatrix} = \begin{pmatrix} U_{11} \hat{X}(T) + U_{12} \hat{Y}(T) \\ U_{21} \hat{X}(T) + U_{22} \hat{Y}(T) \end{pmatrix}$$

che si utilizzano per ricavare  $\hat{Y}(T)$  in funzione di  $\hat{X}(T)$ :

$$\begin{aligned} M(U_{11} \hat{X}(T) + U_{12} \hat{Y}(T)) &= U_{21} \hat{X}(T) + U_{22} \hat{Y}(T) \\ \Rightarrow \hat{Y}(T) &= -(U_{22} - M U_{12})^{-1} (U_{21} - M U_{11}) \hat{X}(T) = G \hat{X}(T) \end{aligned}$$

Dalla relazione  $[X(t)^T, Y(t)^T]^T = U [\hat{X}(t)^T, \hat{Y}(t)^T]^T$ , si ottiene:

$$\begin{aligned} X(t) &= U_{11} \hat{X}(t) + U_{12} \hat{Y}(t) = U_{11} e^{-\Lambda_s(T-t)} \hat{X}(t) + U_{12} e^{-\Lambda_u(T-t)} \hat{Y}(t) \\ &= U_{11} e^{-\Lambda_s(T-t)} \hat{X}(t) + U_{12} e^{-\Lambda_u(T-t)} G \hat{X}(t) = \\ &= [U_{11} + U_{12} e^{-\Lambda_u(T-t)} G e^{\Lambda_s(T-t)}] e^{-\Lambda_s(T-t)} \hat{X}(t) \end{aligned}$$

Ripetendo lo stesso ragionamento per  $\hat{Y}(t)$ :

$$Y(t) = [U_{21} + U_{22} e^{-\Lambda_u(T-t)} G e^{\Lambda_s(T-t)}] e^{-\Lambda_s(T-t)} \hat{X}(t)$$

Avendo ora calcolato esplicitamente  $X(t)$  e  $Y(t)$ , possiamo ottenere  $P(t) = Y(t) X(t)^{-1}$ :

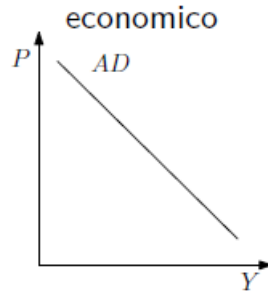
$$P(t) = [U_{21} + U_{22} e^{-\Lambda_u(T-t)} G e^{\Lambda_s(T-t)}] [U_{11} + U_{12} e^{-\Lambda_u(T-t)} G e^{\Lambda_s(T-t)}]^{-1}$$

Allora la soluzione della DRE ottenuta calcolando solo autovalore e autovettori di  $H$ .

## 8.4 Modello Macro-Economico Nuovo Keynesiano

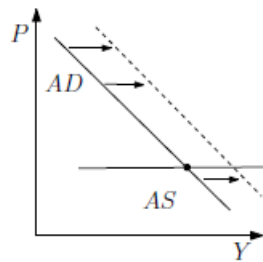
Vogliamo rispondere alla domanda: qual è il livello giusto di produzione/offerta? Per rispondere alla questione Keynesiana, ricordiamo:

**Domanda Aggregata (AD):** domanda di beni o servizi formulata da un sistema economico



curva di domanda aggregata

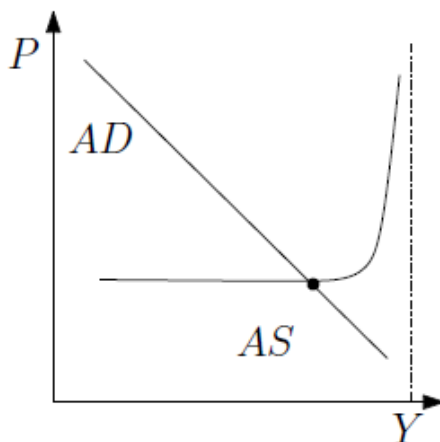
**Offerta Aggregata (AS):** capacità produttiva di un sistema economico



curva di offerta aggregata  
modello *Keynesiano* (breve periodo)

In cui  $P$ :prezzi medi e  $Y$ :produzione effettiva (PIL).

### 8.4.1 Modello Nuovo-Keynesiano



*composizione Domanda Aggregata*

- Gli stipendi sono più lenti dei prezzi ad adeguarsi;

- Se cala la domanda (AD) le aziende devono licenziare per adeguarsi ai prezzi;
- è necessario aumentare (AD) per diminuire il tasso di disoccupazione.

Domanda aggregata ( $Z(t) = \text{Consumi}(C(t)) + \text{Investimenti}(I(t)) + \text{Spesa Pubblica}(G(t)) + \text{Esportazioni}(N(t))$ )

- Se AD supera AS, le industrie aumentano la produzione in maniera proporzionale:

$$\dot{Y}(t) = \alpha(Z(t) - Y(t))$$

- La propensione al consumo è proporzionale all'offerta  $C(t) = \beta Y(t)$ ;
- Deve essere presente un capitale  $K(t)$  a sostenere la produzione, l'investimento è pari alla variazione del capitale  $I(t) = \dot{K}(t)$ ;
- Esiste un livello di capitale desiderato proporzionale alla produzione  $K^*(t) = \gamma Y(t)$  e il capitale deve essere aggiustato in modo da coincidere con  $K^*(t)$ :

$$\dot{K}(t) = \delta(K^*(t) - K(t))$$

- Il controllo è la spesa pubblica  $u(t) = G(t)$  con l'obiettivo di minimizzare la distanza tra  $K$  e  $K^*$  e tra  $Y$  e  $Y^*$

Dalle considerazioni sopra elencate, otteniamo:

$$\min_u \int_0^T ([y(t), k(t)] Q [y(t), k(t)]^T + R u(t)^2) dt$$

$$\begin{pmatrix} \dot{y}(t) \\ \dot{k}(t) \end{pmatrix} = \begin{pmatrix} \alpha(\beta + \delta\gamma - 1) & -\alpha\delta \\ \delta\gamma & -\delta \end{pmatrix} \begin{pmatrix} y(t) \\ k(t) \end{pmatrix} + \begin{pmatrix} \alpha \\ 0 \end{pmatrix} u(t)$$

dove:

- $y(t) = Y(t) - Y^*$ ;  $k(t) = K(t) - K^*(t)$  e  $u(t) = G(t)$
- $Q = I$  e  $R > 0$

# Capitolo 9

## Regolatore Lineare-Quadratico su Orizzonte Infinito

### 9.1 LQR su Orizzonte Infinito

Consideriamo un problema di controllo ottimo su **orizzonte infinito** descritto da un **sistema lineare** e indice di **costo quadratico**:

$$\min_u J(u) = \left\{ \frac{1}{2} \int_0^\infty (x(t)^T Q x(t) + u(t)^T R u(t)) dt \right\}$$

$$\dot{x}(t) = A x(t) + B u(t) \quad x(0) = x_0$$

Prima di verificare se la soluzione del problema su orizzonte infinito esista o meno, scriviamo la sua struttura:

- Soluzione,  $P(t) = [U_{21} + U_{22} e^{-\Lambda_u(T-t)} G e^{\Lambda_s(T-t)}][U_{11} + U_{12} e^{-\Lambda_u(T-t)} G e^{\Lambda_s(T-t)}]^{-1} \implies$   
 $\implies \lim_{T \rightarrow \infty} P(t) = U_{21} U_{11}^{-1} = \tilde{P} \quad u = -R^{-1} B^T \tilde{P} x$
- $-\dot{P}(t) = P(t) A + A^T P(t) - P(t) B R^{-1} B^T P(t) + Q \implies$   
 $0 = A^T \tilde{P} + \tilde{P} A + Q - \tilde{P} B R^{-1} B^T \tilde{P} \quad \text{Equazione algebrica di Riccati (ARE)}$

### 9.2 Condizioni Sufficienti di Esistenza

**Teorema 9.1.**

Consideriamo il problema LQR su orizzonte infinito, con  $M=0, R>0$  e  $Q=D^T D \geq 0$ . Supponiamo che la coppia  $(A, D)$  sia **osservabile** e  $(A, B)$  sia **controllabile**. Allora:

- Esiste un'unica soluzione definita positiva  $\tilde{P}$  di ARE;
- Il sistema a ciclo chiuso:

$$\dot{x} = (A - B R^{-1} B^T \tilde{P}) x$$

ha un equilibrio in zero asintoticamente stabile.

**Nota 9.2. Schema della dimostrazione**

1. **Esistenza** del limite della soluzione su orizzonte finito;
2. **Unicità** della soluzione di ARE;
3. **Ottimalità** del controllo stazionario su orizzonte infinito;
4. **Stabilità Asintotica** (Teorema di Lyapunov+Teorema di LaSalle)

#### 9.2.1 Esistenza del limite della soluzione su orizzonte finito

Definiamo la **famiglia su orizzonte finito parametrizzati rispetto a T**:

$$\min_u J(u) = \left\{ \frac{1}{2} \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt \right\}$$

$$\dot{x}(t) = A x(t) + B u(t) \quad x(0) = x_0$$

Se consideriamo due tempi terminali diversi  $T_1$  e  $T_2$  otteniamo due soluzioni **diverse**  $P_{T_1}(t)$  e  $P_{T_2}(t)$  dal momento che dobbiamo imporre la condizione  $P(T) = 0 (=M)$  in due istanti diversi:

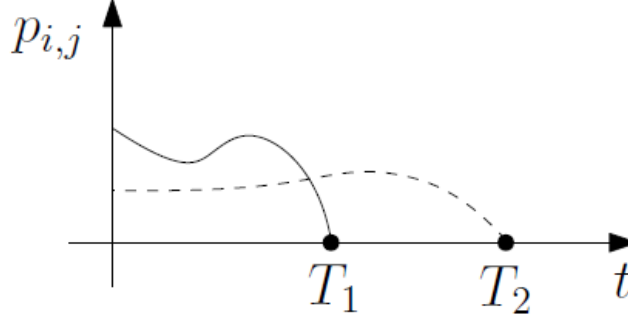


Figura 9.1.

Allora  $\{P_{T_i}(T)\}_{T_i}$ , successione di soluzioni su  $[0, T_i]$  al variare di  $i$ .

**Questa successione ha un limite per  $T \rightarrow \infty$  se:**

- i. è monotonicamente non decrescente;
- ii. ogni elemento è limitato superiormente.

**Dimostrazione.**

Per dimostrare i.: consideriamo due tempi terminali  $T_1, T_2$  tali che  $T_1 < T_2$ . Allora dalla definizione di funzione valore e ricordando che  $R > 0, Q \geq 0$ :

$$\begin{aligned} V_1^*(t, x) &= \int_t^{T_1} l(x_1(\tau), u_1(\tau)) d\tau \leq \int_t^{T_1} l(x_2(\tau), u_2(\tau)) d\tau \\ &\leq \int_t^{T_1} l(x_2(\tau), u_2(\tau)) d\tau + \int_{T_1}^{T_2} l(x_2(\tau) + u_2(\tau)) d\tau = V_2^*(t, x) \end{aligned}$$

dove  $l(x, u) = \frac{1}{2}(x^T Q x + u^T R u)$  ovvero:

$$V_1^*(t, x) = \frac{1}{2} x^T P_{T_1}(t) x \leq \frac{1}{2} x^T P_{T_2}(t) x = V_2^*(t, x) \quad \forall x \in \mathbb{R}^n$$

Quindi, sussiste una relazione d'ordine del tipo:

$$P_{T_1}(t) \leq P_{T_2}(t)$$

Per dimostrare ii.: per l'ipotesi di **controllabilità** di (A,B) esiste una matrice  $k$  tale che  $A + B K$  ha tutti autovalore a parte reale negativa, ovvero  $\sigma(A + B K) \subset \mathbb{C}^-$ .

Il sistema a ciclo chiuso con  $u = K x$  diventa  $\dot{x} = (A + B K) x \implies x(t) = e^{(A+BK)t} x_0$ :

$$\begin{aligned} J(u) &= \frac{1}{2} \int_0^T (x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau)) d\tau = \frac{1}{2} \int_0^T x(\tau)^T (Q + K^T R K) x(\tau) d\tau \\ &= \frac{1}{2} x_0^T \left( \int_0^T e^{(A+BK)^T \tau} (Q + K^T R K) e^{(A+BK) \tau} d\tau \right) x_0 \end{aligned}$$

Per dimostrare che è limitato dobbiamo effettuare il limite per  $T \rightarrow \infty$ . Questo limite viene posto come  $\hat{P}$ .

Quindi  $\forall x_0 \in \mathbb{R}^n$ :  $(\frac{1}{2} x_0^T P_T(0) x_0)$  costo ottimo su  $[0, T]$

$$\begin{aligned} \frac{1}{2} x_0^T P_T(0) x_0 &\leq \frac{1}{2} x_0^T \left( \int_0^T e^{(A+BK)^T \tau} (Q + K^T R K) e^{(A+BK) \tau} d\tau \right) x_0 \\ &\leq \frac{1}{2} x_0^T \left( \int_0^\infty e^{(A+BK)^T \tau} (Q + K^T R K) e^{(A+BK) \tau} d\tau \right) x_0 = \frac{1}{2} x_0^T \hat{P} x_0 \\ &\implies P_T(0) \leq \hat{P} \end{aligned}$$

Stesso ragionamento per un generico tempo iniziale  $t$ ,  $P_T(t) \leq \tilde{P}$ .

□

### 9.2.2 Unicità della soluzione (stabilizzante) di ARE

Supponiamo che esistano due matrici  $\tilde{P}_1$  e  $\tilde{P}_2$  che soddisfano:

$$\begin{aligned} 0 &= \tilde{P}_1 A + A^T \tilde{P}_1 + Q - \tilde{P}_1 B R^{-1} B^T \tilde{P}_1 & A_{cl,1} &= A - B R^{-1} B^T \tilde{P}_1 & \sigma(A_{cl,1}) &\subset \mathbb{C}^- \\ 0 &= \tilde{P}_2 A + A^T \tilde{P}_2 + Q - \tilde{P}_2 B R^{-1} B^T \tilde{P}_2 & A_{cl,2} &= A - B R^{-1} B^T \tilde{P}_2 & \sigma(A_{cl,2}) &\subset \mathbb{C}^- \end{aligned}$$

Sottraendo la seconda equazione dalla prima e sommando e sottraendo  $\tilde{P}_2 B R^{-1} B^T \tilde{P}_2$  otteniamo:

$$0 = (\tilde{P}_1 - \tilde{P}_2) A_{cl,1} + A_{cl,2}^T (\tilde{P}_1 - \tilde{P}_2) \quad \textbf{Equazione di Sylvester}$$

#### Definizione 9.3. Equazione di Sylvester

Un'equazione di Sylvester è un'equazione matriciale lineare nell'incognita  $X$  della forma

$$X A + B X = 0$$

$A, B, C$  matrici note di coefficienti. Quale che sia  $C$ , l'equazione ammette un'unica soluzione se e solo se  $\sigma(A) \cap \sigma(-B) = \emptyset$ ,  $\sigma(-A_{cl,2}^T) \subseteq \mathbb{C}^+$

L'unica soluzione è dunque :

$$(\tilde{P}_1 - \tilde{P}_2) = 0 \longrightarrow \tilde{P}_1 = \tilde{P}_2$$

### 9.2.3 Ottimalità della Soluzione di ARE

Dopo aver dimostrato che il limite esiste ed è unico, consideriamo i limiti di **funzione valore**  $V(x_0) = \frac{1}{2} x_0^T P x_0$  e **controllo**  $u^* = -R^{-1} B^T \tilde{P} x$ .

Ora dobbiamo verificare che **effettivamente**  $u^*$  sia la soluzione ottimale per il problema LQR su  $[0, \infty)$ . Cerchiamo di legare il costo su orizzonte finito alla  $\tilde{P}$ :

Per qualsiasi  $T$  vale

$$\frac{1}{2} x_0^T \tilde{P} x_0 - \frac{1}{2} x^{*T}(T) \tilde{P} x^*(T) = \frac{1}{2} \int_0^T (x^*(t)^T Q x^*(t) + u^*(t)^T R u^*(t)) dt$$

Infatti:

$$\begin{aligned} \frac{1}{2} x_0^T \tilde{P} x_0 - \frac{1}{2} x^{*T}(T) \tilde{P} x^*(T) &= -\frac{1}{2} \int_0^T \frac{d}{dt} [x^*(t)^T \tilde{P} x^*(t)] dt \\ &= -\frac{1}{2} \int_0^T [(A x^*(t) + B u^*(t))^T \tilde{P} x^*(t) + x^*(t)^T \tilde{P} (A x^*(t) + B u^*(t))] dt \\ &= \int_0^T x^*(t)^T (Q + \tilde{P} B R^{-1} R R^{-1} B^T \tilde{P}) x^*(t) dt = J_T(u^*) \end{aligned}$$

Quindi:

$$J_T(u^*) \text{ è il costo di } u^* \text{ su } [0, T]$$

Dato che  $\tilde{P} \geq 0 \implies J_T(u^*) \leq \frac{1}{2} x_0^T \tilde{P} x_0 \quad \forall T \implies J(u^*) \leq \frac{1}{2} x_0^T \tilde{P} x_0$

Inoltre, scegliendo un qualsiasi altro controllo  $u$  abbiamo:

$$\begin{aligned} \frac{1}{2} x_0^T P_T(0) x_0 &\leq \frac{1}{2} \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt \\ &\leq \frac{1}{2} \int_0^\infty (x(t)^T Q x(t) + u(t)^T R u(t)) dt = J(u) \quad \forall T \end{aligned}$$

Facendo il limite per  $T$  che tende ad infinito  $\frac{1}{2} x_0^T \tilde{P} x_0 \leq J(u)$ .

Quindi  $J(u^*) \leq J(u) \quad \forall u \implies u^*$  controllo ottimo su  $[0, \infty]$

Infine, scegliendo  $u = u^* \implies \frac{1}{2} x_0^T \tilde{P} x_0 \leq J(u^*) \implies V(x_0) = \frac{1}{2} x_0^T \tilde{P} x_0 = J(u^*)$

### 9.2.4 Stabilità del sistema a ciclo chiuso

Dimostriamo la stabilità asintotica utilizzando la funzione valore  $V(x) = \frac{1}{2}x^T P x$  come funzione di Lyapunov. Dobbiamo quindi dimostrare che :

1. La matrice  $\tilde{P}$  è **definita positiva**
2. La derivata  $\dot{V}$  è **definita negativa**

Per dimostrare (1), supponiamo per assurdo che  $\tilde{P}$  sia solo semidefinita positiva. Allora esiste una condizione iniziale  $x_0 \neq 0$  tale che:

$$\tilde{P}x_0 = 0 \implies x_0^T \tilde{P} x_0 = 0 \implies \int_0^\infty (x(t)^T D^T D x(t) + u(t)^T R u(t)) dt = 0$$

Dato che  $R \succ 0 \implies u(t) = 0 \forall t \implies \dot{x} = A x \implies x(t) = e^{At} x_0$

Quindi:

$$\int_0^\infty x(t)^T D^T D x(t) dt = x_0^T \left( \int_0^\infty e^{A^T t} D^T D e^{A t} dt \right) x_0 = 0$$

Tuttavia la coppia  $(A, D)$  è osservabile per ipotesi, dunque la matrice Gramiana di osservabilità è definita positiva e l'unica condizione  $x_0$  compatibile sarebbe  $x_0 = 0$ . Ciò giunge ad una contraddizione.

Per dimostrare (2), si considera la derivata della funzione valore lungo le traiettorie del sistema a ciclo chiuso:

$$\dot{V} = x^T \tilde{P} (A - B R^{-1} B^T \tilde{P}) x = \frac{1}{2} x^T (\tilde{P} A + A^T \tilde{P} - 2 \tilde{P} B R^{-1} \tilde{P}) x = -\frac{1}{2} x^T Q x - \frac{1}{2} x^T \tilde{P} B R^{-1} \tilde{P} x$$

Per ottenere  $\dot{V} < 0 \quad \forall x$  dovremmo avere almeno uno dei due termini  $> 0$ :

- $Q \succ 0$ , ma per ipotesi  $Q \succcurlyeq 0$
- $\tilde{P} B R^{-1} B^T \tilde{P} \succ 0$ , ma  $B R^{-1} B^T$  ha rango pieno solo se  $m=n$  (tanti ingressi quante sono le uscite)

In generale possiamo concludere solo che  $\dot{V} \preccurlyeq 0$ . Quindi **stabilità semplice**.

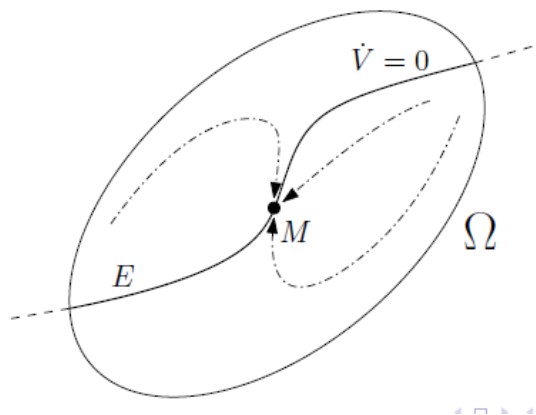
Per dimostrare la **stabilità asintotica** occorre ricorrere al **Teorema di LaSalle** per vedere l'attrattività dei moti.

#### **Teorema 9.4. Teorema di LaSalle**

Consideriamo un sistema autonomo  $\dot{x} = A x$ .

- Sia  $\Omega \subset \mathbb{R}^n$  un insieme compatto positivamente invariante per il sistema
- Sia  $V: \mathbb{R}^n \rightarrow \mathbb{R}$  una funzione  $C^1$  tale che  $\dot{V} = 0$
- Sia  $M$  il più grande insieme invariante contenuto in  $E$

Allora ogni soluzione converge a  $M$  per  $t \rightarrow \infty$



**Figura 9.2.**



**Applicazione del Teorema di LaSalle**

Sappiamo che:

- $V(x)$ ,  $V$  funzione quadratica, fissato un valore per la costante  $c$  gli insiemi di livello

$$\{x \in \mathbb{R}^n: V(x) \leq c\}$$

sono ellissoidi;

- $\dot{V} = -\frac{1}{2}x^T D^T D x - \frac{1}{2}u^T R u \leq 0 \quad \forall x$  dunque  $V$  decresce rispetto a  $t$  o:

$$V(x(t)) \leq V(x(0)) \quad \forall t \geq 0$$

Dunque possiamo scegliere:

$$\Omega = \{x \in \mathbb{R}^n: V(x) \leq V(x_0)\}$$

Compatto ed invariante. Quindi se è vera per ogni  $t$  per soddisfare il teorema deve rimanere in  $\Omega$ .

A questo punto cerchiamo di caratterizzare l'insieme  $E = \{x: \dot{V} = 0\}$  con  $\dot{V} = -\frac{1}{2}x^T D^T D x - \frac{1}{2}u^T R u$

Dato che  $R$  è definita positiva  $u$  deve essere nulla. Quindi:

$$\dot{V} = 0 \implies x(t)^T D^T D x(t) = 0 \implies D x(t) = 0 \quad \forall t$$

Dunque :

$$E = \{x \in \mathbb{R}^n: D e^{At} x = 0, \forall t\}$$

Ricordiamo che per l'ipotesi di osservabilità la coppia  $(A, D)$  dobbiamo poter distinguere le uscite di due qualsiasi condizioni iniziali  $x_1, x_2$  ovvero:

$$\exists t: D e^{At} x_1 \neq D e^{At} x_2 \quad \forall x_1, x_2$$

Infine:

- La condizione iniziale  $x_0 = 0$  fornisce  $D e^{At} x_0 = 0$  per ogni  $t$ ;
- è l'unico stato con tale proprietà per l'osservabilità;

Allora  $E = \{0\}$ ,  $M = \{0\}$

Per il teorema di LaSalle dimostriamo **attrattività** e quindi la stabilità asintotica.



# Capitolo 10

## Algoritmo di Kleinman

### 10.1 Approccio Risolutivo di HJB su orizzonte infinito

Data l'equazione di HJB su orizzonte infinito:

$$0 = \min_u \left\{ \frac{\partial V}{\partial x} f(x, u) + l(x, u) \right\} \quad \forall x \in \mathbb{R}^n \quad \text{lineare in } \frac{\partial V}{\partial x}$$

i passi algoritmici per risolverla in modo esatto sono:

- determinare  $u^*$  come  $\arg \min \Rightarrow u^* = \psi\left(x, \frac{\partial V}{\partial x}\right)$
- Sostituire  $u^*$  nell'equazione
- Risolvere in  $\frac{\partial V}{\partial x}$  l'equazione:

$$0 = \frac{\partial V}{\partial x} f\left(x, \psi\left(x, \frac{\partial V}{\partial x}\right)\right) + l\left(x, \psi\left(x, \frac{\partial V}{\partial x}\right)\right) \quad \forall x \in \mathbb{R}^n \quad \text{non lineare}$$

Se provassimo ad invertire l'ordine dei passi la soluzione non sarebbe ottima poiché il controllo non sarebbe  $\arg \min$ . L'obiettivo è quello di costruire una legge di controllo non inizialmente ottima per poi continuare ad iterare.

### 10.2 Algoritmo di Kleinman

Nel caso lineare quadratico su orizzonte infinito, abbiamo visto che la soluzione ottima è:

$$u = -R^{-1} B^T \bar{P} x = K x$$

in cui  $\bar{P} = \bar{P}^T > 0$  risolve l'equazione:

$$0 = A^T \bar{P} + \bar{P} A + Q - \bar{P} B R^{-1} B^T \bar{P}$$

Invece di risolvere la ARE, vogliamo provare ad approssimare iterativamente la soluzione  $\bar{P}$  risolvendo una sequenza di equazioni lineari.

Notiamo che:

$$\begin{aligned} 0 &= A^T \bar{P} + \bar{P} A + Q - \bar{P} B R^{-1} \bar{P} \\ &= A^T \bar{P} + \bar{P} A + Q - \bar{P} B R^{-1} B^T \bar{P} + \bar{P} B R^{-1} \bar{P} - \bar{P} B R^{-1} \bar{P} \\ &= A^T \bar{P} + \bar{P} A + Q + K^T B^T \bar{P} + K^T R K + \bar{P} B K \\ &= (A + B K)^T \bar{P} + \bar{P} (A + B K) + Q + K^T R K \end{aligned}$$

Definiamo  $A_{cl} = A + B K$

$$A_{cl}^T \bar{P} + \bar{P} A_{cl} = -Q - K^T R K$$

sembra un'equazione lineare di Lyapunov per il sistema a ciclo chiuso  $u = K x$ . Se fissiamo una qualsiasi  $K \neq -R^{-1} B^T \bar{P}$  possiamo risolvere l'equazione.

**Teorema 10.1.** *Algoritmo di Kleinman*

Consideriamo un problema LQR su orizzonte infinito con  $(A, B)$  controllabile e  $(A, D)$  osservabile. Sia  $u_0 = K_0 x$  un qualsiasi controllo che stabilizza il sistema, ovvero  $\sigma(A + BK_0) \subset \mathbb{C}^-$ . Definiamo  $S_i = A + BK_i$  e risolviamo l'equazione lineare di Lyapunov:

$$S_i \bar{P}_i + \bar{P}_i S_i = -Q - K_i^T R K_i$$

nell'incognita  $\bar{P}_i$  e aggiorniamo  $K_{i+1} = -R^{-1} B^T \bar{P}_i$ . Allora:

$$\lim_{t \rightarrow \infty} \bar{P}_i = \bar{P}$$

$$V^* \leq V_{k+1} \leq V_k \leq \dots$$

$$k = 0, 1, \dots \text{ con } V_i(x) = \frac{1}{2} x^T \bar{P}_i x$$

### Dimostrazione.

Sappiamo che  $K_0$  è tale che  $\sigma(A + BK_0) \in \mathbb{C}^-$ , quindi:

$$V_0(x) = \frac{1}{2} x^T \bar{P} x = \frac{1}{2} x^T \left( \int_0^\infty e^{S_0^T t} (Q + K_0 R K_0) e^{S_0 t} dt \right) x < \infty$$

Consideriamo orizzonte finito  $T$  e siano  $P_{0,T}, P_{1,T}$  le soluzioni di:

$$-P_{i,T} = S_i^T P_{i,T} + P_{i,T} S_i + Q + K_i^T R K_i \quad P_{i,T}(T) = 0 \quad i = 0, 1$$

Vogliamo confrontare i costi di  $u_0 = K_0 x$  e  $u_1 = K_1 x$  per un generico  $T$ :

$$J_T(u_0) - J_T(u_1) = \frac{1}{2} x^T (P_{0,T} - P_{1,T}(0)) x$$

Definiamo  $\delta P(t) = P_{0,T} - P_{1,T}(0)$  e studiamo la sua evoluzione nel tempo:

$$\dot{\delta} = -S_1^T \delta P - \delta P S_1 - (K_0 - K_1)^T (B^T P_{0,T} + R K_1) - (B^T P_{0,T} + R K_1)^T (K_0 - K_1) - (K_0 - K_1)^T R (K_0 - K_1)$$

A questo punto conosciamo  $\delta P(T) = 0$  e vogliamo conoscere  $\delta P(0)$ . Tramite la legge di aggiornamento dell'algoritmo  $K_1 = -R^{-1} B^T P_{0,T}$ :

$$\bar{Q} = -(K_0 - K_1)^T R (K_0 - K_1) \rightarrow \delta \dot{P} = -S_1^T \delta P - \delta P S_1 + \bar{Q} \quad \delta P(T) = 0$$

Corrisponde ad una **equazione matriciale lineare** della forma:

$$\dot{X} = A^T X + X A + Y \quad X(T) = 0 \implies X(t) = \int_t^T -e^{A^T(t-\tau)} Y e^{A(t-\tau)} d\tau$$

Nel nostro caso:

$$\delta P(0) = \int_0^T -e^{S_1^T \tau} \bar{Q} e^{S_1 \tau} d\tau$$

e dunque:

$$J_T(u_0) - J_T(u_1) = \frac{1}{2} x^T \delta P(0) x = \frac{1}{2} x^T \left( \int_0^T e^{S_1^T \tau} (K_0 - K_1)^T T (K_0 - K_1) e^{S_1 \tau} d\tau \right) x \geq 0$$

Ovvero:

$$P_{0,T}(0) \geq P_{1,T}(0) \quad \forall T$$

Al limite:

$$\bar{P}_0 \geq \bar{P}_1$$

Infine:

$$V_1(x) = \frac{1}{2} x^T \bar{P}_1 x < \infty \iff \sigma(A + BK_1) \subset \mathbb{C}^-$$

Possiamo iterare il ragionamento.

□

# Capitolo 11

## Tracking e Reiezione di Disturbi

### 11.1 Problema di Tracking/Asservimento

Il problema di tracking/asservimento è un problema di minimizzazione del tipo:

$$\min_u J(u) = \frac{1}{2} \int_0^T ((\xi(t) - \dot{\xi}(t))^T Q (\xi(t) - \dot{\xi}(t)) + u(t)^T R u(t)) dt$$

$$\dot{\xi}(t) = A\xi(t) + Bu(t), \xi(0) = \xi_0$$

In cui l'obiettivo è quello di avere lo stato del sistema più vicino possibile ad un segnale nel tempo in un intervallo di tempo fissato. Alcune tipologie di questo problema possono essere ricondotte ad un problema di Regolatore Lineare Quadratico (LQR).

A tale scopo, supponiamo che la funzione desiderata del tempo  $\tilde{\xi}(t)$  soddisfi:

$$A\tilde{\xi} - \dot{\tilde{\xi}} = 0 \implies \tilde{\xi} = \sum_{i=1}^{\nu} \sum_{j=0}^{\mu_i} c_{ij} e^{\lambda_i t} t^j$$

per qualche  $c_{ij}$  (**funzioni delle condizioni iniziali**) dove:

- $\nu$  è il numero di autovalori distinti della matrice  $A$ ;
- $\lambda_i$  è l'autovalore  $i$ -esimo della matrice  $A$ ;
- $\mu_i$  è la molteplicità algebrica dell'autovalore  $\lambda_i$ .

**Teorema 11.1. Principio del modello interno**

*Per avere tracking perfetto, il processo deve contenere una copia (modello) dei modi (funzioni del tempo) del segnale di riferimento  $\tilde{\xi}(t)$ .*

Quindi se vale questo principio, è ragionevole considerare il cambio di coordinate:

$$x = \xi - \tilde{\xi} \rightarrow \xi = x + \tilde{\xi}$$

E conseguentemente, nelle nuove coordinate:

- **Indice di costo:**

$$J(u) = \frac{1}{2} \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) d\tau$$

- **Sistema dinamico:**

$$\begin{aligned} \dot{x} &= \dot{\xi} - \dot{\tilde{\xi}} = A\xi + Bu - \dot{\tilde{\xi}} \\ &= A(x + \tilde{\xi}) + Bu - \dot{\tilde{\xi}} = Ax + Bu + (A\tilde{\xi} - \dot{\tilde{\xi}}) \\ &= Ax + Bu \end{aligned}$$

In termini di problema LQR, la soluzione ottima è data da:

$$u^* = -R^{-1} B^T P(t) x(t) = -R^{-1} B^T P(t) (\xi(t) - \tilde{\xi}(t))$$

## 11.2 Reiezione dei disturbi noti

Nel caso in cui il principio del modello interno non venga soddisfatto, possiamo considerare il cambio di coordinate ottenendo:

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

ottenendo così un **sistema affine**.

In questa casistica possiamo definire il seguente problema:

**Problema 11.1.** Problema di Reiezione dei disturbi noti

Problema di minimizzazione di indice di costo quadratico:

$$J(u) = \frac{1}{2} \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt$$

con un vincolo dinamico **affine**, cioè in presenza  $w(t)$  disturbo misurabile **noto**.

## 11.3 Equazione di HJB per reiezione dei disturbi

Consideriamo il problema di controllo ottimo:

$$\min_u J(u) = \frac{1}{2} \int_0^T (x(t)^T Q x(t) + u(t)^T R u(t)) dt$$

$$\dot{x} = Ax + Bu + w$$

Non si tratta di un problema LQ per la presenza di  $w$ , ma rientra nella teoria dei sistemi non lineari.

**L'equazione di HJB fornisce condizioni necessarie e sufficienti:**

$$-\frac{\partial V}{\partial t} = \min_u \left\{ \frac{\partial V}{\partial x} (Ax + Bu + w(t)) + \frac{1}{2} x^T Q x + \frac{1}{2} u^T R u \right\}, V(x, T) = 0 \quad \forall x \in \mathbb{R}^n, t \in [0, T]$$

## 11.4 Forma della funzione valore

Da un'analisi preliminare, considerando la classica funzione valore, si nota che è valida solamente per  $w(t) = 0$ ; quindi occorre considerare la **funzione valore completa**:

$$V(x, t) = \frac{1}{2} x^T P(t) x + b(t)^T x + c(t), P(t) \in \mathbb{R}^{n \times n}, b(t) \in \mathbb{R}^{n \times 1}, c(t) \in \mathbb{R}$$

con derivate parziali:

$$\begin{cases} \frac{\partial V}{\partial t} = \frac{1}{2} x^T \dot{P}(t) x + \dot{b}(t)^T x + \dot{c}(t) \\ \frac{\partial V}{\partial x} = x^T P(t) x + b(t)^T \end{cases}$$

Dunque, calcolando il minimo rispetto ad  $u$  di:

$$\frac{1}{2} x^T Q x + \frac{1}{2} u^T R u + (x^T P(t) + b(t)^T) (Ax + Bu + w)$$

Otteniamo:

$$u^* = -R^{-1} B^T (P(t) x + b(t)) = -R^{-1} B^T P(t) x - R^{-1} B^T b(t)$$

## 11.5 Soluzione dell'equazione di HJB

Sostituendo  $u^*$  nell'equazione di HJB, otteniamo:

$$-\frac{1}{2} x^T \dot{P}(t) x - \dot{b}(t)^T x - \dot{c}(t) = \frac{1}{2} x^T Q x + \frac{1}{2} (x^T P(t) + b(t)^T) B R^{-1} R R^{-1} B^T (P(t) x + b(t))$$

$$+ x^T P(t) A x + b(t)^T A x - (x^T P(t) + b(t)^T) B R^{-1} B^T (P(t) x + b(t)) + x^T P(t) w(t) + b(t)^T w(t)$$

Per ogni  $t$  e ogni  $x \in \mathbb{R}^n$  se e solo se:

- Coefficiente Quadratico:  $-\dot{P}(t) = P(t)A + A^T P(t) + Q - P(t)BR^{-1}B^T P(t)$ ,  $P(T) = 0$ ;
- Coefficiente Lineare:  $\dot{b}(t) = -(A - BR^{-1}B^T P(t))^T b(t) - P(t)w(t)$ ,  $b(T) = 0$ ;
- Coefficiente Costante:  $\dot{c}(t) = \frac{1}{2}b(t)^T BR^{-1}B^T b(t) - b(t)^T w(t)$ ,  $c(T) = 0$

Le condizioni al contorno derivano da  $V(x, T) = 0$ .

## 11.6 Commenti sulla soluzione di HJB

- Le equazioni in  $P$ ,  $b$  e  $c$  possono essere risolte in maniera sequenziale;
- Il sistema in  $\dot{b}$  diventa non stazionario:

$$\dot{b} = -(A - BR^{-1}B^T P(t))b \implies b(t) = \Phi_b(t, \tau) b(\tau)$$

Allora:

$$b(t) = \int_t^T \Phi_b(t, \tau) P(\tau) w(\tau) d\tau$$

Il termine in feedforwards è non causale rispetto al disturbo.

- La funzione  $c(t)$  non compare in  $u^*$ , ma  $V(0, 0) = c(0)$ . Il costo dovuto al disturbo da  $x_0 = 0$ .