

POLITECNICO DI MILANO

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Corso di Laurea Magistrale in Ingegneria dell'Automazione



**TECNICHE RANDOMIZZATE DI PROGRAMMAZIONE LINEARE
PER IL CONTROLLO OTTIMO DI
SISTEMI IBRIDI STOCASTICI**

Relatore:

prof. Maria PRANDINI

Tesi di Laurea di:

Anacleto PETRETTI

Matr. 770836

Anno Accademico 2012/2013

*Alla mia famiglia, a chi ha costruito
e a chi costruirà con me.*

INDICE

INDICE	3
ELENCO DELLE FIGURE	5
ELENCO DELLE TABELLE	8
1 INTRODUZIONE	10
2 PROGRAMMAZIONE DINAMICA PER IL CONTROLLO OTTIMO DI UNA CATENA DI MARKOV	12
2.1 Catena di Markov controllata.....	12
2.2 Controllo ottimo di una catena di Markov.....	14
2.3 Soluzione mediante la programmazione dinamica.....	16
2.3.1 Value iteration.....	18
2.3.2 Policy iteration.....	20
2.3.3 Programmazione lineare.....	21
2.4 Programmazione dinamica approssimata.....	22
2.4.1 Value iteration approssimata.....	23
2.4.2 Policy iteration approssimata.....	24
2.4.3 Programmazione lineare approssimata.....	24

3	PROGRAMMAZIONE LINEARE APPROSSIMATA PER IL CONTROLLO DI SISTEMI IBRIDI STOCASTICI	27
3.1	Sistemi ibridi stocastici.....	27
3.2	Controllo ottimo di un sistema ibrido stocastico.....	30
3.3	Value iteration approssimata ibrida.....	33
3.4	Programmazione lineare approssimata ibrida.....	36
3.4.1	Scelta delle funzioni di base.....	38
3.4.2	Scelta della funzione peso sullo stato.....	44
3.4.3	Riduzione del numero di vincoli.....	46
3.4.4	Soluzione randomizzata basata sull'approccio a scenario.....	47
3.5	Applicazione al problema dell'invarianza controllata.....	51
4	ANALISI DELLA HALP A SCENARIO IN UN CASO DI STUDIO	55
4.1	Descrizione del caso di studio.....	55
4.2	Soluzione mediante HALP e approccio a scenario.....	60
4.3	Risultati nel caso di una singola stanza.....	64
4.3.1	Funzioni di base gaussiane.....	66
4.3.2	Funzioni di base polinomiali.....	73
4.4	Risultati nel caso di due stanze adiacenti.....	77
4.5	Risultati nel caso di più di due stanze.....	85
5	CONCLUSIONI E SVILUPPI FUTURI	88
A	APPENDICE	90
A.1	Codice Matlab per n_s stanze con funzioni di base e funzione peso gaussiane	90
A.2	Codice Matlab per n_s stanze con funzioni di base polinomiali e funzione peso uniforme	99

ELENCO DELLE FIGURE

2.1	Pseudo-codice dell'algoritmo Value Iteration.	20
2.2	Pseudo-codice dell'algoritmo della Policy Iteration.	21
3.1	Rappresentazione grafica del Teorema.	50
3.2	Algoritmo di HALP con Approccio a Scenario.	51
4.1	Possibile disposizione relativa di 4 stanze.	58
4.2	Funzione peso gaussiana.	65
4.3	Funzione peso uniforme.	66
4.4	Confronto tra value function ottenuta per grigliatura (magenta modo1, verde modo 2) . 68 e value function ottenuta mediante HALP (rosso modo 1, nero modo 2) nel caso di scelta causale della media e della varianza delle funzioni approssimanti.	68
4.5	Confronto tra value function ottenuta per grigliatura (magenta modo1, verde modo 2) . 69 e value function ottenuta mediante HALP (rosso modo 1, nero modo 2) nel caso si scelgano valori di media e varianza delle funzioni approssimanti pari a quelli ottenuti mediante fitting.	69

- 4.6** Confronto tra value function ottenuta per grigliatura (magenta modo 1, verde modo 2) . 70
e value function ottenuta mediante HALP (rosso modo 1, nero modo 2) utilizzando
un numero elevato di funzioni di base.
- 4.7** Scelta di media e varianza nel caso di approssimazione mediante un numero elevato ... 70
di funzioni di base gaussiane.
- 4.8** Value function approssimata ottenuta con HALP utilizzando una funzione peso 72
gaussiana.
- 4.9** Value function approssimata ottenuta con HALP utilizzando una funzione peso 72
uniforme.
- 4.10** Value function $V^w(\cdot)$ ottenuta con HALP utilizzando una funzione peso gaussiana. ... 73
- 4.11** Value function $V^w(\cdot)$ ottenuta con HALP utilizzando una funzione peso uniforme. 74
- 4.12** Value function $V^w(\cdot)$ ottenuta con HALP utilizzando un polinomio di terzo grado 75
come approssimante all'esterno dell'insieme A .
- 4.13** Value function $V^w(\cdot)$ ottenuta con HALP utilizzando un polinomio di quinto grado ... 75
come approssimante all'esterno dell'insieme A .
- 4.14** Politica ottima ottenuta mediante il metodo della value iteration approssimata con 79
grigliatura della componente continua dello spazio di stato.
- 4.15** Value function $\hat{V}(\cdot)$ approssimata (modo 1) ottenuta mediante il metodo della value .. 80
iteration approssimata con grigliatura della componente continua dello spazio di
stato.
- 4.16** Politica ottima ottenuta mediante il metodo della HALP a scenario nel caso 81
 $\sigma_{11^t} = \sigma_{12^t} = 10$ e $\sigma_{11^s} = \sigma_{12^s} = 1$.
- 4.17** Politica ottima ottenuta mediante il metodo della HALP a scenario nel caso 81
 $\sigma_{11^t} = \sigma_{12^t} = 15$ e $\sigma_{11^s} = \sigma_{12^s} = 1$.

- 4.18** Posizione delle gaussiane approssimanti nel caso di una griglia 3×3 82
- 4.19** Value function approssimata $V^w(\cdot)$ (modo 1) ottenuta con una griglia di funzioni 83
di base 3×3 .
- 4.20** Value function approssimata $V^w(\cdot)$ (modo 1) ottenuta con una griglia di funzioni 83
di base 4×4 .
- 4.21** Value function approssimata $V^w(\cdot)$ (modo 1) ottenuta con una griglia di funzioni 83
di base 5×5 .
- 4.22** Value function approssimata $V^w(\cdot)$ (modo 1) ottenuta con una griglia di funzioni 84
di base 6×6 .

ELENCO DELLE TABELLE

4.1 Valori di media e varianza scelti per l'esempio in Figura 4.4.....	67
4.2 Valori di media e varianza delle funzioni di base ottenute mediante fitting.....	69
4.3 Confronto tra i tempi di calcolo per gli esempi in Figura 4.5 e Figura 4.6.....	71
4.4 Confronto tra i tempi di calcolo nel caso di funzioni di base gaussiane e nel caso di..... funzioni di base polinomiali.	76
4.5 Tempi di calcolo per le soluzioni in Figura 4.19 e in Figura 4.21.....	84
4.6 Tempi di calcolo nel caso in cui $n_s=3$	86
4.7 Tempi di calcolo nel caso in cui $n_s=4$	87

SOMMARIO

In questo lavoro si affronta il problema del controllo ottimo di sistemi ibridi stocastici, caratterizzando la politica ottima attraverso le equazioni della programmazione dinamica e proponendo l'utilizzo del metodo della *programmazione lineare approssimata* (ALP) per la loro soluzione. Tale metodo porta alla formulazione di un problema di programmazione lineare (LP) semi-infinito, difficile da risolvere a causa del numero infinito di vincoli. L'idea sviluppata in questo lavoro è quella di applicare l'*approccio a scenario* per ottenere una soluzione approssimata al problema LP, tramite la riduzione dei vincoli ad un numero finito N . La teoria a scenario garantisce che, se N è scelto in modo opportuno, allora anche gli altri vincoli non considerati risultano comunque soddisfatti, ad eccezione di una porzione predefinita. L'approccio LP a scenario viene applicato al problema dell'invarianza probabilistica controllata in un caso di studio, e confrontato con un metodo di programmazione dinamica approssimata alternativo, già noto in letteratura.

1

INTRODUZIONE

Un *sistema ibrido* è un sistema dinamico caratterizzato dalla presenza sia di variabili continue che di variabili discrete, le cui dinamiche sono interallacciate. Nel caso in cui l'evoluzione del sistema sia affetta da incertezza caratterizzata in modo probabilistico si parla di *sistemi ibridi stocastici*. I sistemi ibridi stocastici sono stati oggetto di grande attenzione negli ultimi anni, sia per quanto riguarda lo studio di aspetti teorici che l'utilizzo in ambiti applicativi ([4],[8]). Un esempio particolarmente significativo è rappresentato dalle reti elettriche di nuova generazione, caratterizzate da variabili discrete legate allo stato on/off di generatori e carichi collegati alla rete, variabili continue come tensioni e correnti, e componenti aleatorie come quelle che caratterizzano la produzione di energia elettrica da parte di fonti rinnovabili (eolico e solare). Come controparte della loro capacità espressiva, i sistemi ibridi stocastici sono più complessi da analizzare ed i problemi di controllo che li coinvolgono più difficili da risolvere ([9]).

In questa tesi si affronta un problema di *controllo ottimo su orizzonte infinito* per la classe dei sistemi ibridi stocastici a tempo discreto. L'obiettivo è determinare attraverso tecniche di *programmazione dinamica approssimata* una politica di feedback sullo stato che sia ottima secondo una cifra di merito media su orizzonte infinito.

Il termine controllo ottimo cominciò ad essere usato negli anni '50 per indicare il problema del progetto di una legge di controllo che fosse in grado di ottimizzare una misura del comportamento dinamico del sistema sotto controllo su un assegnato intervallo temporale. Basandosi sulle teorie di Hamilton e Jacobi, Richard Bellman propose un approccio al problema basato sulla definizione della cosiddetta *value function* e sulla sua caratterizzazione mediante un'equazione funzionale denominata, in suo onore, *equazione di Bellman* ([14]). La programmazione dinamica approssimata si occupa appunto della soluzione numerica dell'equazione di Bellman.

In questa tesi si focalizza l'attenzione sul metodo della programmazione lineare approssimata ibrida (HALP), che ha come obiettivo quello di determinare un'approssimazione della *value function* utilizzando una classe funzionale parametrizzata linearmente. Il problema della determinazione dei parametri viene ricondotto ad un problema di programmazione lineare (LP) semi-infinito, e viene proposto l'utilizzo dell'*approccio a scenario* per la sua soluzione tramite riduzione del numero di vincoli a finito.

I risultati ottenibili mediante l'utilizzo congiunto di HALP e approccio a scenario vengono analizzati in un caso di studio tratto da [2] relativo al controllo di temperatura di un certo numero di stanze in presenza di vincoli sugli elementi riscaldanti a disposizione.

La tesi è organizzata come segue: nel Capitolo 2 si richiamano alcuni concetti ben noti in letteratura sul controllo ottimo di catene di Markov controllate e sulla sua soluzione attraverso tecniche di programmazione dinamica. Nel Capitolo 3 si descrive la nozione di sistema ibrido stocastico e si estendono ad esso i risultati mostrati nel Capitolo 2, soffermandosi in particolare sul metodo della programmazione lineare approssimata. Viene poi introdotto l'approccio a scenario per la riduzione del numero di vincoli coinvolti nel problema di programmazione lineare ibrida approssimata e rendere il problema affrontabile in pratica. Nel Capitolo 4 si applica il metodo proposto ad un caso di studio in cui il problema è quello di mantenere entro un intervallo prestabilito la temperatura di un certo numero di stanze. Il sistema è modellizzabile mediante un sistema ibrido stocastico e l'obiettivo formulabile come problema di invarianza probabilistica controllata. Il Capitolo 5 contiene, infine, alcune considerazioni conclusive sui risultati raggiunti e discute possibili sviluppi e miglioramenti.

2

PROGRAMMAZIONE DINAMICA PER IL CONTROLLO OTTIMO DI UNA CATENA DI MARKOV

In questo capitolo vengono richiamati alcuni concetti e risultati relativi al controllo ottimo di catene di Markov. Le catene di Markov controllate sono ormai diventate uno standard nella rappresentazione di problemi di decisione affetti da incertezza in numerosi campi quali, ad esempio, l'economia, la biologia e l'ingegneria. Sono molti gli esempi di problemi risolti mediante il loro utilizzo ([4]).

Nel seguito, dopo aver introdotto la definizione di catena di Markov controllata e richiamato le proprietà di cui i processi decisionali di Markov godono, verranno descritti i metodi utilizzati per la loro risoluzione. In particolare si tratteranno i problemi di controllo ottimo su orizzonte infinito soffermandosi sull'approccio della Programmazione Dinamica per la loro soluzione. Ci si soffermerà sul metodo della Programmazione Lineare Approssimata. Tale metodo costituirà la base dell'approccio risolutivo adottato per affrontare i problemi di controllo ottimo per sistemi ibridi stocastici.

2.1 CATENA DI MARKOV CONTROLLATA

Le catene di Markov controllate costituiscono un formalismo matematico per la modellizzazione di processi stocastici a tempo discreto, la cui evoluzione dipende da una variabile di controllo. Formalmente una catena di Markov controllata è descritta da una

tripletta di elementi $M=(S,U,T_s)$ dove S rappresenta lo spazio di stato discreto, U è lo spazio di controllo delle azioni possibili e $T_s:S\times S\times U\times\mathbb{N}\rightarrow[0,1]$ la probabilità di transizione tra gli stati. Una catena di Markov controllata gode della *proprietà di Markov*:

L'evoluzione nel futuro del sistema è indipendente dal passato, condizionatamente al presente.

Da un punto di vista formale:

$$Pr(s_{t+1}|s_t, u_t, s_{t-1}, u_{t-1}, \dots, s_0, u_0) = Pr(s_{t+1}|s_t, u_t) \quad \forall s_{t+1}, s_t, u_t, s_{t-1}, u_{t-1}, \dots, s_0, u_0$$

dove s_t e u_t sono rispettivamente gli stati e le azioni all'istante t . Analizziamo ora in dettaglio gli elementi costituenti una catena di Markov controllata.

Spazio di stato. Lo stato di un sistema ne descrive completamente e univocamente la condizione in un determinato istante di tempo. Assegnato l'istante di tempo $t\in T$, S_t rappresenta l'insieme di tutti i possibili valori che lo stato del sistema può assumere all'istante t . L'insieme $S=\cup_t S_t$ comprende quindi tutti i valori che lo stato del sistema può assumere nel corso della sua evoluzione. In molti casi si ha che: $S_t=S \forall t$.

Spazio di controllo. L'insieme di tutte le azioni di controllo che si possono applicare al sistema in un istante di tempo t viene detto spazio di controllo ed è indicato nella nostra trattazione con U_t .

In alcuni casi l'insieme di controllo all'istante t dipende dal valore dello stato in quell'istante. Si avrà quindi:

$$U_t = \cup_{s\in S_t} U_t(s)$$

Lo spazio di controllo è l'insieme di tutte le possibili azioni applicabili al sistema in ogni istante di tempo:

$$U = \cup_t U_t.$$

Nel caso in cui gli insiemi di controllo siano invarianti rispetto al tempo avremo che:

$$U_t = U \quad \forall t$$

Probabilità di transizione. Le catene di Markov controllate costituiscono l'equivalente

probabilistico del concetto di sistema dinamico a tempo discreto. In un sistema dinamico l'evoluzione del sistema è descritta dalla *funzione di transizione*, in una catena controllata di Markov l'evoluzione è governata dalla *probabilità di transizione*:

$$T_s : S \times S \times U \times \mathbb{N} \rightarrow [0,1]$$

In particolare, assegnato l'istante di tempo t , lo stato s_t e l'azione u_t a quell'istante, lo stato del sistema all'istante $t+1$ è determinato in base alla distribuzione di probabilità:

$$T_s(\cdot | s_t, u_t, t)$$

detta *probabilità di transizione*. $T_s(s_{t+1} | s_t, u_t, t)$ indica la probabilità che lo stato del sistema all'istante $t+1$ sia s_{t+1} dato che all'istante t è s_t e viene applicata l'azione di controllo u_t . Varrà quindi la condizione:

$$\sum_{s' \in S_{t+1}} T_s(s' | s, u, t) = 1, \quad \forall s \in S_t, u \in U_t(s), t \in \mathbb{N}$$

Nel caso in cui la probabilità di transizione (e anche lo spazio di stato e quello di controllo) sia tempo invariante, si parla di *omogeneità della catena di Markov* che corrisponde alla nozione di tempo invarianza per i sistemi dinamici classici.

2.2 CONTROLLO OTTIMO DI UNA CATENA DI MARKOV

Supponiamo che ad ogni configurazione stato-azione sia assegnato un determinato valore tramite una funzione:

$$R : S \times U \rightarrow \mathbb{R}$$

Se $R(s, u)$ rappresenta un “reward” che si ottiene quando il sistema è nello stato s e viene applicata l'azione u , l'obiettivo sarà massimizzare il reward accumulato durante l'evoluzione del sistema.

Controllare una catena di Markov in modo ottimo significa scegliere una *politica*, ossia definire una regola che, per ogni stato del sistema, permetta di determinare l'azione di

controllo da applicare in quel particolare stato per massimizzare il reward accumulato. Indicata con:

$$\mu_t: S_t \rightarrow U_t$$

la *regola di decisione* che permette di scegliere, sulla base dello stato del sistema a quell'istante, l'azione da applicare all'istante t , la politica è definita come la sequenza delle regole di decisione:

$$\pi = \{\mu_0, \mu_1, \mu_2, \dots\}$$

Il problema diventa quello di scegliere π in modo da massimizzare una opportuna cifra di merito. Per esempio, assegnato un orizzonte di controllo $[0, T]$, la cifra di merito da massimizzare è rappresentata dalla somma dei valori assunti dalla funzione di reward negli istanti temporali da 0 a T , valutata in media:

$$E_\pi \left[\sum_{t=0}^T R(s_t, u_t) \right], \quad (2.1)$$

dove $E_\pi[\cdot]$ indica il valore atteso quando la politica applicata è π . Si noti infatti che, fissata π , risulta fissata la probabilità di transizione tra stati:

$$P_\pi(s_t = s' | s_t = s) = T_s(s' | s, \mu_t(s), t)$$

e la probabilità di osservare una certa traiettoria in $[0, T]$ è:

$$P_\pi(s_0 = \bar{s}_0, s_1 = \bar{s}_1, \dots, s_k = \bar{s}_k, \dots, s_T = \bar{s}_T) = \prod_{t=0}^{T-1} T_s(\bar{s}_{t+1} | \bar{s}_t, \mu_t(\bar{s}_t), t) P(\bar{s}_0)$$

dove $P: S_0 \rightarrow [0, 1]$ è la distribuzione di probabilità iniziale.

Nel caso in cui si scelga di perseguire un obiettivo di controllo su orizzonte infinito ($T = \infty$), la formulazione (2.1) non risulta utilizzabile in quanto il valore della cifra di merito potrebbe essere infinito per ogni scelta delle variabili di controllo (anche se la funzione reward assume valori limitati), impedendo quindi di identificare la scelta migliore.

Definiamo la cifra di merito su orizzonte infinito:

$$E_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, u_t) \right] \quad (2.2)$$

dove $\gamma \in [0,1)$ è detto fattore di “discount”. Il fattore γ permette di pesare i reward sull'orizzonte temporale considerato in modo che quanto “guadagnato” nel futuro remoto valga meno di quanto guadagnato nel futuro più prossimo. Nel caso in cui $\gamma < 1$ e $R(s, u)$ sia limitata $\forall s \in S, u \in U$ la sommatoria (2.2) assume un valore finito. Nel caso particolare in cui $\gamma = 0$, il controllore si dice “cieco” e l'obiettivo del controllo si riduce a scegliere l'ingresso all'istante $t=0$ in modo da massimizzare:

$$E_{\pi} [R(s_0, u_0)]$$

Di particolare interesse solo le politiche stazionarie, cioè le politiche per cui la regola di decisione è sempre la stessa in ogni istante considerato: $\pi = \{\mu, \mu, \mu, \dots\}$ indicate in modo sintetico semplicemente con μ . Se tutto è stazionario, cioè $S_t = S, U_t = U, \forall t$ e $T_s : S \times S \times U \rightarrow [0,1]$ non dipende dal tempo, allora la politica ottima nel caso della cifra di merito (2.2) è in effetti stazionaria.

Il problema della determinazione di una politica ottima può essere affrontato mediante la cosiddetta *Programmazione Dinamica* (*Dynamic Programming* o *DP*). Nella sezione seguente richiamiamo alcuni risultati fondamentali della programmazione dinamica con riferimento al caso stazionario e alla cifra di merito (2.2).

2.3 SOLUZIONE MEDIANTE LA PROGRAMMAZIONE DINAMICA

Fissata una politica stazionaria μ , la value function ad essa associata è definita:

$$V_{\mu}(s) = E_{\mu} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mu(s_t)) \middle| s_0 = s \right] \quad (2.3)$$

e indica il reward su orizzonte infinito ottenuto applicando la politica μ a partire dallo stato

iniziale s . È facile verificare che la value function soddisfa l'equazione ricorsiva:

$$\begin{aligned} V_\mu(s) &= R(s, \mu(s)) + \gamma \sum_{s' \in S} E_\mu \left[\sum_{t=1}^{\infty} \gamma^{t-1} R(s_t, \mu(s_t)) \middle| s_1 = s' \right] P_\mu(s_1 = s' | s_0 = s) \\ &= R(s, \mu(s)) + \gamma \left[\sum_{s' \in S} T_s(s' | s, \mu(s)) V_\mu(s') \right] \end{aligned} \quad (2.4)$$

Tale equazione è lineare nelle incognite $V_\mu(s), s \in S$. Si noti che la cifra di merito:

$$E_\mu \left[\sum_{t=1}^{\infty} \gamma^t R(s_t, \mu(s_t)) \right]$$

può essere espressa in funzione della value function e della distribuzione di probabilità iniziale nel modo seguente:

$$E_\mu \left[\sum_{t=1}^{\infty} \gamma^t R(s_t, \mu(s_t)) \right] = \sum_{s \in S} V_\mu(s) P(s_0 = s)$$

La politica ottima si può quindi ottenere massimizzando la value function $V_\mu(s)$ per ogni valore dello stato iniziale $s \in S$.

La funzione $V^*(s) = \sup_{\mu} V_\mu(s), s \in S$, rappresenta la value function ottima ed è l'unica soluzione dell'equazione di Bellman ([14]):

$$V^*(s) = \max_{u \in U} \left[R(s, u) + \gamma \sum_{s' \in S} T_s(s' | s, u) V^*(s') \right], \quad s \in S. \quad (2.5)$$

Sulla base di $V^*(s)$ è possibile calcolare la politica ottima come:

$$\mu^*(s) = \arg \max_{u \in U} \left[R(s, u) + \gamma \sum_{s'} T_s(s' | s, u) V^*(s') \right] \quad (2.6)$$

Se definiamo l'operatore di Bellman T :

$$T[V(\cdot)] = \max_{u \in U} \left[R(s', u) + \gamma \sum_{s' \in S} T_s(s' | s, u) V(s') \right], \quad \forall s \in S \quad (2.7)$$

allora l'equazione di Bellman (2.5) può essere riscritta in forma operatoriale come:

$$V^*(s) = T[V^*(\cdot)](s), \quad s \in S. \quad (2.8)$$

Esistono diversi metodi per la soluzione dell'equazione di Bellman, alcuni esatti altri approssimati. Introduciamo ora alcune delle metodologie più usate nell'ambito della programmazione dinamica, quando il modello della catena di Markov controllata e la funzione di reward sono note. Le estensioni al caso in cui esse non siano note sono denominate in letteratura tecniche di “Reinforcement Learning” ([15]) e non verranno trattate in questo lavoro.

2.3.1 VALUE ITERATION

Quello della value iteration è uno degli algoritmi basilari della programmazione dinamica. Esso sfrutta la proprietà dell'operatore di Bellman di essere una “contrazione” con fattore di contrazione pari al discount factor γ .

Definizione: Sia (F, ρ) uno spazio metrico e $B: F \rightarrow F$ una funzione che mappa F in se stesso. B è una contrazione se per qualche $\delta \in (0,1)$, vale la seguente condizione: $\rho(Bf, Bg) \leq \delta \rho(f, g)$ per ogni f e g in F .

Teorema: L'operatore di Bellman è un contrazione rispetto alla metrica:

$$\rho(V(\cdot), V'(\cdot)) = \|V - V'\|_\infty = \max_{s \in S} |V(s) - V'(s)|.$$

Dimostrazione. Osserviamo che l'operatore di Bellman soddisfa le seguenti proprietà:

- 1) monotonia $V(s) \leq V'(s), \forall s \in S \rightarrow T[V(\cdot)](s) \leq T[V'(\cdot)](s), \forall s \in S$
- 2) additività $T[V(\cdot) + \alpha](s) = T[V(\cdot)](s) + \gamma \alpha, \forall s \in S$

Posto $d = \max_{s \in S} |V(s) - V'(s)|$ allora:

$$V(s) - d \leq V'(s) \leq V(s) + d, \quad \forall s \in S$$

Applicando l'operatore di Bellman ad entrambi i membri e sfruttando le proprietà 1 e 2, si ha:

$$T[V(\cdot)](s) - \gamma d \leq T[V(\cdot)](s) \leq T[V(\cdot)](s) + \gamma d, \quad \forall s \in S$$

da cui segue:

$$|T[V(\cdot)](s) - T[V'(\cdot)](s)| \leq \gamma d, \quad \forall s \in S$$

e quindi:

$$\max_{s \in S} |T[V(\cdot)](s) - T[V'(\cdot)](s)| \leq \gamma \max_{s \in S} |V(s) - V'(s)|$$

Applicando iterativamente l'operatore di Bellman ad una funzione $\hat{V}^0(s)$ si ottiene:

$$\hat{V}^{(n+1)} = T[\hat{V}^{(n)}] = T[T[\hat{V}^{(n-1)}]] = \dots T^n[\hat{V}^0]$$

Il fatto che l'operatore di Bellman sia una contrazione garantisce che:

$$\lim_{n \rightarrow \infty} T^n[\hat{V}^0] = V^*$$

questo perché dalla proprietà contrattività applicata iterativamente segue che:

$$\sup_{s \in S} |T^n[\hat{V}^0(\cdot)](s) - T^n[V^*(\cdot)](s)| \leq \gamma^n \sup_{s \in S} |\hat{V}^0(s) - V^*(s)|$$

dove:

$$T^n[V^*(\cdot)](s) = V^*(s), \quad \forall s \in S, \forall n \in \mathbb{N}$$

e:

$$\lim_{n \rightarrow \infty} \gamma^n \sup_{s \in S} |\hat{V}^0(s) - V^*(s)| = 0$$

La value iteration si basa sull'applicazione ripetuta dell'operatore di Bellman. L'obiettivo è quello di determinare una politica stazionaria ϵ -ottima μ_ϵ che soddisfi:

$$\|\hat{V}_{\mu_\epsilon} - V^*\|_\infty = \max_{s \in S} |\hat{V}_{\mu_\epsilon}(s) - V^*(s)| \leq \epsilon$$

dove ε è l'errore di Bellman. Di seguito è riportato un algoritmo in pseudo codice che implementa la logica della value iteration.

Algoritmo Value Iteration

Input: Una catena di Markov controllata $M=(S,U,T_s)$, $\hat{V}^{(0)}(\cdot)$, $\beta > 0$

$k=1$

1) calcola:

$$\hat{V}^k = T[\hat{V}^{k-1}]$$

2) se $\sup_{s \in S} |\hat{V}^k(s) - \hat{V}^{k-1}(s)| < \beta$ allora calcola la politica $\hat{\mu}^* : S \rightarrow U$ tale che:

$$\hat{\mu}^*(s) = \arg \max_{u \in U} \{ R(s, u) + \gamma \sum_{s' \in S} T_s(s'|s, u) \hat{V}^k(s') \}, s \in S$$

esci, altrimenti incrementa k e torna al passo 1.

Output: $\hat{V}^k(s)$, $\hat{\mu}^*$

Figura 2.1: Pseudo-codice dell'algoritmo Value Iteration.

Se $|R(s, u)| \leq c, c \in \mathbb{R}$, scegliendo:

$$\beta = \epsilon \frac{1-\gamma}{2\gamma}$$

si ha che la politica $\hat{\mu}^*$ è ε -ottima.

2.3.2 POLICY ITERATION

L'algoritmo di policy iteration si basa sull'individuazione della politica ottima μ^* a partire da una politica iniziale $\mu^{(0)}$. Nello specifico, l'algoritmo consiste in due passi:

- determinare l'espressione della value function associata alla politica attualmente adottata;
- applicare l'operatore di Bellman per aggiornare la politica.

L'espressione aggiornata della value function si ricava dall'equazione (2.4), applicata ad ogni $s \in S$, ottenendo in questo modo un sistema di $|S|$ equazioni lineari. L'algoritmo termina quando si ricava consecutivamente la stessa politica: $\mu^{(t)} = \mu^{(t+1)} = \mu^*$. La convergenza della

soluzione alla politica ottima per la catena di Markov controllata è garantita ([12]) e questo dipende dal fatto che spazio di stato e di controllo sono finiti.

Per ogni catena di Markov controllata esistono $|U|^{|S|}$ possibili politiche e quindi il tempo richiesto per la convergenza, nel peggiore dei casi, può essere esponenziale in $|S|$.

Nella pratica, la policy iteration converge più velocemente degli altri algoritmi per la risoluzione delle catene di Markov controllate. Questo vantaggio è parzialmente oscurato dal carico computazionale richiesto, che è un $O(|S|^3 + |S|^2|U|)$ ad ogni iterazione.

Algoritmo Policy Iteration

Input: catena di Markov controllata $M=(S,U,T_s)$, $\mu^{(0)}$.

$t = -1$;
 ripeti

$t = t + 1$;

calcola $V_{\mu^{(t)}}$ risolvendo $|S|$ equazioni lineari:

$$V_{\mu^{(t)}}(s) = R(s, \mu^t(s)) + \gamma \sum_{s'} T_s(s'|s, \mu^t(s)) V_{\mu^{(t)}}(s') \quad \text{per ogni } s \in S$$

$$\mu^{(t+1)} = \arg \max_{u \in U} \left[R(s, u) + \gamma \sum_{s'} T_s(s'|s, u) V_{\mu^{(t)}}(s') \right]$$

fino a quando $\mu^{(t+1)} = \mu^{(t)}$

Output: politica ottima μ^* .

Figura 2.2: Pseudo-codice dell'algoritmo della Policy Iteration.

2.3.3 PROGRAMMAZIONE LINEARE

La *Programmazione Lineare* (indicata con la sigla LP) permette di riformulare il problema del calcolo della value function ottima, e conseguentemente della politica ottima, come un

problema di minimizzazione vincolato, secondo le seguenti equazioni:

$$\begin{aligned} \min_{V(\cdot)} \quad & \sum_{s \in S} \psi(s) V(s) \\ \text{soggetto a:} \quad & V(s) \geq R(s, u) + \gamma \sum_{s'} T_s(s' | s, u) V(s'), \quad \forall s \in S, u \in U \end{aligned} \quad (2.9)$$

in cui $\psi(s) \geq 0, \forall s \in S$ rappresenta un peso sullo stato del sistema, $V(s)$ le variabili di ottimizzazione del problema LP, una per ogni stato $s \in S$, e il numero dei vincoli è pari alla cardinalità dello spazio prodotto $S \times U$.

In effetti la $V(\cdot)$ che soddisfa i vincoli in (2.9), soddisfa anche:

$$V(s) \geq V^*(s), \quad \forall s \in S \quad (2.10)$$

dato che essi implicano:

$$V(s) \geq T[V(\cdot)](s), \quad \forall s \in S$$

e applicando le proprietà viste nella sezione 2.3.1:

$$V(s) \geq T[V(\cdot)](s) \geq T^2[V(\cdot)](s) \geq \dots \geq T^n[V(\cdot)](s) \geq \dots \geq V^*(s), \quad \forall s \in S.$$

Minimizzando $\sum_{s \in S} \psi(s) V(s)$ con $\psi(s) > 0, \forall s \in S$ si ottiene quindi $V^*(s), s \in S$.

Il metodo della programmazione lineare, come pure quello della value iteration o della policy iteration, ha lo svantaggio di richiedere uno sforzo computazionale che cresce esponenzialmente con la cardinalità dello spazio di stato. Il che rende di fatto inapplicabili queste metodologie ai problemi in larga scala. Per queste ragioni sono stati introdotti diversi approcci volti alla ricerca di una soluzione approssimata al problema della programmazione dinamica e a rendere fruibile tale teoria anche nel caso di sistemi con spazio di stato di grandi dimensioni.

2.4 PROGRAMMAZIONE DINAMICA APPROSSIMATA

Già nel caso di sistemi incerti descritti tramite catene di Markov caratterizzate da una cardinalità dello spazio di stato finita, l'applicazione dei metodi esatti per la soluzione dei

problemi di DP non è fattibile nello studio di casi pratici in cui anche la sola memorizzazione della value function in una look-up table può essere onerosa.

Un approccio comunemente utilizzato per rappresentare in modo più sintetico la value function è quello di approssimarla con una funzione $V^w(s)$, parametrizzata tramite il vettore $w=[w_1, w_2, \dots, w_k]$. Una delle approssimazioni più utilizzate è quella di tipo *lineare*:

$$V^w(s) = \sum_{i=1}^k w_i f_i(s), \quad s \in S \quad (2.11)$$

La funzione approssimante è cioè espressa come combinazione lineare di k funzioni $f_i: S \rightarrow \mathbb{R}$ pesate con il relativo parametro w_i .

L'idea dell'approssimazione della value function è stata sfruttata per l'elaborazione di versioni approssimate degli algoritmi esatti per la soluzione dei problemi di programmazione dinamica: si parlerà quindi di Value Iteration Approssimata, Policy Iteration Approssimata ed infine di Programmazione Lineare Approssimata (ALP).

2.4.1 VALUE ITERATION APPROSSIMATA

La value iteration approssimata costituisce un adattamento dell'algoritmo presentato nel paragrafo 2.3.1, in cui si sostituisce l'espressione di $\hat{V}(s)$ con una funzione approssimante del tipo indicato nella (2.11). In questo caso si aggiorna il vettore dei pesi w di $V^w(s)$, in modo da minimizzare una misura della distanza tra la funzione coi parametri aggiornati $V^{w^{(t+1)}}(s)$ e quella ottenuta applicando l'operatore di Bellman alla funzione approssimante calcolata al passo precedente $T[V^{w^{(t)}}(\cdot)](s)$. Tipicamente si minimizza l'espressione:

$$\sum_{s \in G} \psi(s) [V^{w^{(t+1)}}(s) - T[V^{w^{(t)}}(\cdot)](s)]^2 \quad (2.12)$$

risolvendo di fatto un problema di fitting della funzione $V^{w^{(t+1)}}$ a $T[V^{w^{(t)}}(\cdot)]$ tramite la minimizzazione dell'errore quadratico calcolato su un sottoinsieme $G = \{s^{(1)}, s^{(2)}, \dots, s^{(N)}\}$ di

valori per lo stato secondo un peso $\psi(s)$ attribuito a tali valori.

2.4.2 POLICY ITERATION APPROSSIMATA

Equivalentemente a quanto visto per la value iteration, anche l'algoritmo di policy iteration presenta una sua versione approssimata. Anche in questo caso si procede per due passi successivi: nel primo si aggiorna l'espressione di $V^w(s)$ sulla base della politica attuale $\mu^{(t)}$ e nel secondo si ricava la nuova politica $\mu^{(t+1)}$:

$$\mu^{(t+1)}(s) = \arg \max_{u \in U} \left[R(s, u) + \gamma \sum_{s' \in S} T_s(s' | s, u) V^{w^{(t)}}(s') \right] \quad (2.13)$$

e i due passi vengono reiterati fino a che $\mu^{(t+1)} = \mu^{(t)}$. Questo algoritmo ha buone prestazioni quando $V^w(\cdot)$ è in grado di approssimare bene la value function associata alla politica $\mu^{(t)}$ calcolata iterativamente.

Questo in pratica è difficile da garantire a priori, essendo generalmente ignota la forma della value function. A causa di ciò, l'algoritmo di policy iteration approssimato tende a generare rapidi miglioramenti dell'approssimazione nelle prime iterazioni per poi restare bloccato in un ciclo oscillatorio ([6]).

2.4.3 PROGRAMMAZIONE LINEARE APPROSSIMATA

Introduciamo ora l'algoritmo della *Programmazione Lineare Approssimata* (o *ALP*) applicata a sistemi descritti mediante catene di Markov controllate. Esso si basa sull'approssimazione della value function fornita dalla (2.11), richiamata qui per completezza:

$$V^w(s) = \sum_{i=0}^k w_i f_i(s), \quad s \in S$$

Il problema di ottimizzazione (2.9) introdotto nel paragrafo 2.3.3 assume la forma:

$$\begin{aligned} \min_{w_i, i=0, \dots, k} \quad & \sum_{s \in S} \psi(s) \sum_{i=0}^k w_i f_i(s) \\ \text{soggetto a:} \quad & \sum_{i=0}^k w_i f_i(s) \geq R(s, u) + \gamma \sum_{s' \in S} T_s(s' | s, u) \sum_{i=0}^k w_i f_i(s'), \forall s \in S, u \in U \end{aligned} \quad (2.14)$$

In questo caso $\mathbf{w} = [w_0, w_1, \dots, w_k]$ rappresenta il vettore delle $k + 1$ variabili di ottimizzazione del problema LP, mentre la funzione $\psi(s) \geq 0$ rappresenta un peso sullo stato che permette di decidere l'importanza relativa dei diversi valori dello stato.

Affinché il problema sia *feasible*, ossia affinché ammetta una soluzione che rispetti i vincoli, occorre che tra le funzioni di base scelte ci sia la funzione costante $f_0(\cdot) = 1$. Si assume che tale funzione sia sempre presente.

Mentre il problema LP completo esposto nel paragrafo 2.3.3 porta all'individuazione della funzione ottima V^* , in modo indipendente rispetto alla scelta del peso $\psi(\cdot)$, l'algoritmo ALP fornisce una funzione approssimante $V^w(s)$ in cui il vettore dei pesi \mathbf{w}^* risultante dal processo di ottimizzazione ha una chiara dipendenza da $\psi(\cdot)$. Questo perché dai vincoli in (2.14) e dalla monotonia dell'operatore di Bellman segue che:

$$V^w(s) \geq T[V^w(\cdot)](s) \geq T^2[V^w(\cdot)](s) \geq \dots \geq T^n[V^w(\cdot)] \quad \forall s \in S.$$

Dato che:

$$\lim_{n \rightarrow \infty} T^n[V^w(\cdot)](s) = V^*(s),$$

si ha quindi che:

$$V^w(s) \geq V^*(s), \forall s \in S$$

e minimizzare $\sum_{s \in S} \psi(s) V^w(s)$ equivale a minimizzare la norma 1 di $V^w(\cdot) - V^*(\cdot)$ pesata con $\psi(\cdot)$:

$$\sum_{s \in S} \psi(s) (V^w(s) - V^*(s)) = \sum_{s \in S} \psi(s) |V^w(s) - V^*(s)|$$

Attraverso un'opportuna scelta di $\psi(s)$, è possibile imporre che l'approssimazione sia migliore in certe zone dello spazio di stato rispetto ad altre, compatibilmente con la struttura di $V^w(\cdot)$.

Come per il caso completo, anche la formulazione approssimata dell'algoritmo di programmazione lineare coinvolge un numero di vincoli (della forma espressa nella (2.14)) pari a $|S \times U|$.

3

PROGRAMMAZIONE LINEARE APPROSSIMATA PER IL CONTROLLO DI SISTEMI IBRIDI STOCASTICI

In questo capitolo viene affrontato il problema del controllo ottimo di sistemi ibridi stocastici a tempo discreto. Dopo aver mostrato che un sistema ibrido stocastico può essere interpretato come un processo di Markov controllato con spazio di stato ibrido, si estenderanno gli approcci risolutivi basati sulla programmazione dinamica introdotti nel capitolo precedente. In particolare, si considererà la tecnica basata sulla programmazione lineare approssimata, proponendo l'utilizzo dell'*approccio a scenario* per risolvere il problema di ottimizzazione lineare semi-infinito risultante.

La tecnica sviluppata sarà testata in un caso di studio nel capitolo 4 e confrontata con una tecnica alternativa basata sulla grigliatura della componente continua dello spazio di stato ibrido.

3.1 SISTEMI IBRIDI STOCASTICI

I *Sistemi Ibridi Stocastici* (SHS o *Stochastic Hybrid Systems*) permettono di rappresentare

sistemi dinamici caratterizzati dall'interazione di una dinamica discreta e di una dinamica continua e affetti da incertezza descritta in modo probabilistico. Qui di seguito richiamiamo il modello di SHS a tempo discreto (DTSHS) introdotto in [1].

Definizione [DTSHS]. Un DTSHS è descritto dall'insieme di elementi

$H = (Q, n, U, T_x, T_q, R)$, dove:

- $Q = \{q_1, q_2, \dots, q_m\}$, $m \in \mathbb{N}$ rappresenta lo spazio di stato discreto del sistema; $q \in Q$ è anche detto *modo*;
- $n: Q \rightarrow \mathbb{N}$ assegna ad ogni modo $q \in Q$ la dimensione dello spazio continuo $\mathbb{R}^{n(q)}$ quando il sistema opera nel modo q . Lo spazio di stato ibrido sarà allora dato da $S = \cup_{q \in Q} \{q\} \times \mathbb{R}^{n(q)}$;
- U rappresenta lo spazio di controllo;
- $T_x: B(\mathbb{R}^{n(\cdot)}) \times S \times U \rightarrow [0,1]$ è la probabilità di transizione; $T_x(\cdot | s, u)$, $s \in S, u \in U$ caratterizza l'evoluzione della componente continua dello stato dato il valore corrente s dello stato ibrido e u dell'ingresso;
- $T_q: Q \times S \times U \rightarrow [0,1]$ è la probabilità di transizione $T_q(\cdot | s, u)$, $s \in S$ e $u \in U$, che caratterizza l'evoluzione della componente discreta dello stato dato il valore corrente dello stato ibrido e dell'ingresso u ;
- $R: B(\mathbb{R}^{n(\cdot)}) \times S \times U \times Q \rightarrow [0,1]$ assegna ad ogni $s \in S, u \in U$ e $q' \in Q$, una misura di probabilità sullo spazio boreliano $(\mathbb{R}^{n(q)}, B(\mathbb{R}^{n(q)}))$; $R(\cdot | (q, x), u, q')$ caratterizza il reset della componente continua dello stato a seguito di una transizione discreta da q a q' quando l'ingresso applicato è u .

La probabilità di transizione discreta T_q permette di modellizzare sia le *transizioni spontanee* da uno stato discreto all'altro che possono avvenire durante l'evoluzione dello stato continuo, sia le *transizioni forzate* che devono avvenire quando lo stato continuo esce da un dominio prescritto. In particolare affinché le transizioni spontanee possano avvenire quando un'azione di controllo $u \in U$ abilita una transizione dal modo corrente q al modo q' basta che

sia $T_q(q'|q, x, u) > 0$.

Relativamente alle transizioni forzate l'insieme $Inv(q)$ di tutti i valori ammissibili dello stato continuo relativamente al particolare modo $q \in Q$ può essere espresso in termini di T_q , forzando $T_q(q|(q, x), u) = 0$ per tutti i valori dello spazio continuo $x \in \mathbb{R}^{n(q)}$ all'esterno di $Inv(q)$, indipendentemente dal valore dell'azione di controllo $u \in U$.

Definizione [Esecuzione di un DTSHS]. Un processo stocastico $\{q_k, x_k\}_{k=0}^N$ è un'esecuzione del sistema ibrido stocastico a tempo discreto $H = (Q, n, U, T_x, T_q, R)$ sull'orizzonte temporale $\{0, 1, \dots, N\}$ associata alla sequenza di ingressi $\{\bar{u}_k\}_{k=0}^N$ e alla condizione iniziale $\bar{s}_0 \in S$ se le sue realizzazioni sono ottenute con il seguente algoritmo:

Poni $k=0$ e $(q_0, x_0) = \bar{s}_0$;
fintanto che $k < N$ esegui

estrai un valore $\bar{q}_{k+1} \in Q$ per q_{k+1} secondo la distribuzione di probabilità
 $T_q(\cdot | (\bar{q}_k, \bar{x}_k), \bar{u}_k)$;

se $\bar{q}_{k+1} = \bar{q}_k$, allora:

estrai un valore $\bar{x}_{k+1} \in X$ per x_{k+1} secondo la distribuzione di
probabilità $T_x(\cdot | (\bar{q}_k, \bar{x}_k), \bar{u}_k)$;

altrimenti

estrai un valore $\bar{x}_{k+1} \in X$ per x_{k+1} secondo la distribuzione di
probabilità $R(\cdot | (\bar{q}_k, \bar{x}_k), \bar{u}_k, \bar{q}_{k+1})$;

$k = k + 1$;

fine.

Introduciamo ora la probabilità di transizione dello stato ibrido che assegna ad ogni coppia $s = (q, x) \in S, u \in U$ una misura di probabilità, sullo spazio boreliano $(S, B(S))$, $T_s: B(S) \times S \times U \rightarrow [0, 1]$:

$$T_s((\cdot, q')|s, u) = \tau_x(\cdot|s, u, q')T_q(q'|s, u), \quad q' \in Q$$

dove $\tau_x: B(\mathbb{R}^{n(\cdot)}) \times S \times U \times Q \rightarrow [0, 1]$ è dato da:

$$\tau_x(\cdot|(q, x), u, q') = \begin{cases} T_x(\cdot|(q, x), u) & \text{se } q' = q \\ R(\cdot|(q, x), u, q') & \text{se } q' \neq q \end{cases}$$

Allora la definizione di esecuzione si semplifica nel modo seguente:

Posto $s_0 = \bar{s}_0$ e $k = 0$;
 fintanto che $k < N$
 estrai un valore $\bar{s}_{k+1} \in S$ per s_{k+1} secondo $T_s(\cdot|\bar{s}_k, \bar{u}_k)$;
 incrementa k ;
 fine

Questo rivela che la definizione formale di un SHS può essere ricondotta a quella di un catena di Markov controllata $H = \{S, U, T_s\}$, dove:

- S rappresenta lo spazio di stato ibrido;
- U rappresenta lo spazio di controllo;
- $T_s: B(S) \times S \times U \rightarrow [0,1]$ rappresenta la probabilità di transizione e assegna ad ogni $s = (q, x) \in S, u \in U$ una misura di probabilità $T_s(\cdot|s, u)$ sullo spazio boreliano $(S, B(S))$.

Essendo quindi un DTSHS un processo di Markov controllato, potremo affrontare il problema estendendo i metodi già presentati nel Capitolo 2 al caso di spazio di stato ibrido. Per semplicità ci riferiremo al caso in cui $n(q) = n, \forall q \in Q$, cioè la dimensione dello spazio di stato continuo è la stessa in ogni modo $q \in Q$.

3.2 CONTROLLO OTTIMO DI UN SISTEMA IBRIDO STOCASTICO

Supponiamo che ad ogni configurazione stato-azione sia assegnato un determinato valore tramite una funzione:

$$R: S \times U \rightarrow \mathbb{R}$$

Se $R(s, u)$ rappresenta un “reward” che si ottiene quando il sistema è nello stato s e viene

applicata l'azione u , l'obiettivo sarà massimizzare il reward accumulato durante l'evoluzione del sistema. Controllare un sistema ibrido in modo ottimo significa scegliere una *politica*, ossia definire una regola che, per ogni stato del sistema, permetta di determinare l'azione di controllo da applicare in quel particolare stato per massimizzare il reward accumulato.

Come per le catene di Markov, una politica per un DTSHS $H = \{S, U, T_s\}$ è una sequenza $\pi = \{\mu_0, \mu_1, \mu_2, \dots\}$ di funzioni:

$$\mu_t: S \rightarrow U, \quad t = 0, 1, 2, \dots,$$

dallo spazio di stato ibrido $S = Q \times X$ allo spazio di controllo U .

Consideriamo ora la seguente cifra di merito:

$$E_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, u_t) \right] \quad (3.1)$$

dove $E_\pi[\cdot]$ indica il valore atteso valutato quando la politica applicata è π e $\gamma \in (0, 1)$ è il fattore di “discount”. La politica π ottima è quella che massimizza la (3.1).

Se tutte le mappe μ sono uguali, cioè $\mu_t = \mu \forall t \geq 0$ allora la politica π si dice stazionaria e viene indicata con μ .

Fissata una politica stazionaria μ , la value function ad essa associata è:

$$V_\mu(s) = E_\mu \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mu(s_t)) \middle| s_0 = s \right] \quad (3.2)$$

e indica il reward su orizzonte infinito ottenuto applicando la politica μ a partire dallo stato iniziale s . Dato che :

$$E_\mu \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mu(s_t)) \right] = \int_S V_\mu(s) P_{s_0}(ds) ds$$

dove $P_{s_0}: B(S) \rightarrow [0, 1]$ è la distribuzione di probabilità dello stato iniziale s_0 , allora la politica ottima μ^* è tale che:

$$V_{\mu^*}(s) = \sup_{\mu} V_\mu(s), \quad \forall s \in S.$$

È facile verificare che la value function $V_\mu(\cdot)$ soddisfa l'equazione ricorsiva:

$$V_\mu(s) = R(s, \mu(s)) + \gamma E_{s' \sim T_s(\cdot|s, \mu(s))} [V_\mu(s')], \quad (3.3)$$

dove, data $V: S \rightarrow \mathbb{R}$, $E_{s' \sim T_s(\cdot|s, u)} [V(s')]$ indica il valore atteso di $V(s')$ quando s' ha distribuzione di probabilità condizionata $T_s(\cdot|s, u)$:

$$E_{s' \sim T_s(\cdot|s, u)} [V(s')] = \int_S V(s') T_s(ds'|s, u) = \sum_{q \in Q} T_q(q'|s, u) \int_X V(q', x') \tau_x(dx'|s, u, q') dx \quad (3.4)$$

L'equazione (3.3) è lineare nelle incognite $V_\mu(s)$, $s \in S$.

Introduciamo l'operatore di Bellman T :

$$T[V(\cdot)](s) = \sup_{u \in U} \{R(s, u) + \gamma E_{s' \sim T_s(\cdot|s, u)} [V(s')]\} \quad (3.5)$$

Posto $V^*(s) = \sup_{\mu} V_\mu(s)$, essa rappresenta la cifra di merito ottima quando $s_0 = s$ ed è l'unica soluzione dell'equazione di Bellman:

$$V^*(s) = T[V^*(s)] \quad (3.6)$$

Inoltre si ha che:

$$\lim_{n \rightarrow \infty} T^n[V(s)] = V^*(s), \quad s \in S$$

Anche nel caso di catene di Markov controllate con spazio di stato ibrido, l'operatore T gode delle proprietà enunciate nel capitolo precedente per il caso di spazio di stato discreto:

- 1) monotonia: $V(s) \leq V'(s)$, $\forall s \in S \rightarrow T[V(\cdot)](s) \leq T[V'(\cdot)](s)$, $\forall s \in S$
- 2) additività: $T[V(\cdot) + \alpha](s) = T[V(\cdot)](s) + \gamma \alpha$
- 3) è una contrazione per la metrica $\rho(V(\cdot), V'(\cdot)) = \sup_{s \in S} |V(s) - V'(s)|$.

La politica ottima $\mu^*: S \rightarrow U$ è data da:

$$\mu^*(s) = \arg \sup_{u \in U} \{R(s, u) + \gamma E_{s' \sim T_s(\cdot|s, u)} [V^*(s')]\} \quad (3.7)$$

Essa è ben definita se $\forall s \in S$ esiste \bar{u} tale che:

$$R(s, \bar{u}) + \gamma E_{s' \sim T_s(\cdot|s, \bar{u})}[V^*(s')] = \sup_{u \in U} \{R(s, u) + \gamma E_{s' \sim T_s(\cdot|s, u)}[V^*(s')]\}$$

Nel seguito si considererà il caso in cui lo spazio di stato U sia con cardinalità finita per il quale tale condizione è soddisfatta.

Introduciamo ora alcune tecniche utilizzate nell'ambito della programmazione dinamica nel caso di sistemi stocastici ibridi, soffermandoci in particolare sull'estensione a tali sistemi del metodo della programmazione lineare approssimata (ALP) presentato nel capitolo precedente. È importante osservare che, nel caso di sistemi ibridi, lo spazio di stato ha cardinalità infinita e quindi la value function non può essere rappresentata in modo esatto mediante look-up table. Sarà quindi necessario introdurre una rappresentazione approssimata riconducendosi al caso di spazio di stato finito tramite grigliatura oppure considerando una classe funzionale parametrizzata con un numero finito di parametri.

3.3 VALUE ITERATION APPROSSIMATA IBRIDA

Come per il caso di sistemi con spazio di stato discreto, anche in quello di sistemi con spazio di stato ibrido l'algoritmo della value iteration si basa sull'applicazione iterata dell'operatore di Bellman T e permette di ricavare una politica stazionaria ϵ -ottima μ_ϵ che soddisfa:

$$\|\hat{V}_{\mu_\epsilon} - V^*\|_\infty = \sup_{s \in S} |\hat{V}_{\mu_\epsilon}(s) - V^*(s)| \leq \epsilon$$

Ad ogni iterazione l'algoritmo calcola $\hat{V}^k(s)$ applicando l'operatore di Bellman a $\hat{V}^{k-1}(s)$, la value iteration si arresta quando $\hat{V}^k(s)$ soddisfa:

$$|\hat{V}^k(s) - \hat{V}^{k-1}(s)| < \beta, \quad \beta > 0$$

Come per il caso discreto, se lo spazio di controllo U ha cardinalità finita, vale ancora il

seguito risultato:

Se $|R(s, u)| \leq c, c \in \mathbb{R}$, scegliendo:

$$\beta = \epsilon \frac{1-\gamma}{2\gamma}$$

si ha che la politica:

$$\hat{u}(s) = \arg \min_{u \in U} [R(s, u) + \gamma E_{s' \sim T_s(\cdot|s, u)} \hat{V}^k(s')]$$

è ϵ -ottima.

Il problema dell'applicazione dell'algoritmo della value iteration al caso di spazio di stato ibrido risiede nel fatto che l'aggiornamento della funzione $\hat{V}^k(s)$ ad ogni iterazione deve essere effettuato sullo spazio di stato $S = Q \times X$ di cardinalità infinita. Per far fronte a questo problema si possono utilizzare due approcci alternativi:

- 1) **Approssimazione funzionale:** si approssima la funzione $\hat{V}(s)$ con una funzione $V^w(s)$ parametrizzata in $w \in \mathbb{R}^k$. Ad ogni passo dell'algoritmo si aggiorna il vettore dei pesi w in modo da minimizzare:

$$\sum_{s \in G} \psi(s) [V^{w^{(t+1)}}(s) - T[V^{w^{(t)}}(\cdot)](s)]^2$$

Ad ogni iterazione si richiede, quindi, l'applicazione dell'operatore di Bellman alla funzione $V^{w^{(t)}}(s)$ e per il calcolo della quantità $E_{s' \sim T_s(\cdot|s, u)}[V^{w^{(t)}}(s)]$ per $s \in G = \{s_1, s_2, \dots, s_N\}$. Ciò può essere fatto in modo analitico in alcuni casi (vedi Sezione 3.4.1) oppure utilizzando il metodo Monte Carlo. In quest'ultimo caso, l'errore di approssimazione introdotto può accumularsi in iterazioni successive;

- 2) **Grigliatura:** si approssima il sistema ibrido stocastico con una catena di Markov controllata mediante una discretizzazione della componente continua dello spazio di stato. Si considera una partizione dello spazio di stato continuo in N insiemi $X_i, i = 1, \dots, N$ con $\cup_{i=1}^N X_i = X$ e si prende un valore all'interno di ogni elemento della partizione. Si ottiene così lo spazio ibrido discretizzato:

$$G = \{s_1, s_2, \dots, s_{n_G}\}$$

con $n_G = N \cdot |Q|$, dove $s_i \in G$ è della forma $s_i = (x_i, q)$, $i = 1, \dots, N$, $q \in Q$. In questo modo si può rappresentare la value function come una funzione costante sui vari elementi della partizione $\hat{V}(s) = \hat{V}(s_i)$, $\forall s \in S_i = \{q\} \times X_i$, $i = 1, \dots, N$, $q \in Q$.

L'operatore di Bellman per $\hat{V}(s)$ può essere definito solo sugli stati $s_i \in G$ nel modo seguente:

$$T_G[\hat{V}(\cdot)](s_i) = \max_{u \in U} \left[R(s_i, u) + \gamma \sum_{j=1}^{n_G} T_{sG}(s_j | s_i, u) \hat{V}(s_j) \right], \quad i = 1, 2, \dots, n_G \quad (3.8)$$

dove $T_{sG}(s_j | s_i, u) = \int_{S_j} T_s(ds | s_i, u)$, con S_j elemento della partizione a cui appartiene s_j , è la probabilità di transizione da $s_i \in G$ ad $s_j \in G$ della catena di Markov approssimante il sistema ibrido di partenza.

In tal modo è possibile calcolare $V^{k+1}(s) = T_G[V^k(\cdot)](s)$, cioè svolgere un passo della value iteration, rappresentando \hat{V} con una look-up table e applicando l'operatore T_G definito nella (3.8).

Il metodo della grigliatura risulta computazionalmente oneroso per sistemi con dimensione dello spazio di stato continuo elevata, data la dipendenza esponenziale del numero di punti sulla griglia da tale dimensione. Una discretizzazione non sufficientemente accurata infatti comporta una scarsa qualità delle approssimazioni ottenute iterando l'operatore T_G . Risulta allora utile trovare delle tecniche che permettano di trattare opportunamente il caso di sistemi ibridi senza passare per una discretizzazione uniforme delle variabili di stato continue e senza amplificare l'errore di approssimazione tramite iterazioni successive dell'operatore di Bellman.

3.4 PROGRAMMAZIONE LINEARE APPROSSIMATA IBRIDA

La *Programmazione Lineare Approssimata Ibrida* (*Hybrid Approximate Linear Programming* o *HALP*) estende l'algoritmo della programmazione lineare approssimata (ALP) al caso di processi con spazio di stato ibrido. Come nel caso dell'ALP, si utilizza come funzione approssimante della value function ottima V^* la combinazione lineare di opportune *funzioni di base*. Data la natura ibrida dello stato, si adotta la seguente approssimazione:

$$V^w(q, x) = \sum_{i=0}^{n_q} w_{iq} f_{iq}(x), \quad (q, x) \in S, \quad (3.9)$$

in cui le funzioni di base $f_{iq}(x)$ sono funzioni della sola componente continua dello stato.

In tal modo si riduce il problema della determinazione di V^* all'ottimizzazione dei pesi delle funzioni di base la cui combinazione lineare costituisce la funzione approssimante. La formulazione dell'HALP è data da:

$$\begin{aligned} \min_{w_{iq}} \quad & E_{\psi(s)} \left[\sum_{i=0}^{n_q} w_{iq} f_{iq}(x) \right] \\ \text{soggetto a:} \quad & \sum_{i=0}^{n_q} w_{iq} f_{iq}(x) \geq R(s, u) + \gamma E_{s'=(q', x') \sim T_s(\cdot|s, u)} \left[\sum_{i=0}^{n_{q'}} w_{iq'} f_{iq'}(x') \right], \quad \forall s \in S, u \in U \end{aligned} \quad (3.10)$$

dove $\psi(s) \geq 0, s \in S$, soddisfa $\int_S \psi(s) ds = 1$ e:

$$\begin{aligned} E_{\psi(s)} \left[\sum_{i=0}^{n_q} w_{iq'} f_{iq'}(x) \right] &= \sum_{q \in Q} \sum_{i=0}^{n_q} w_{iq} \int_x \psi(q, x) f_{iq}(x) dx \\ E_{s' \sim T_s(\cdot|s, u)} \left[\sum_{i=0}^{n_q} w_{iq} f_{iq}(x) \right] &= \sum_{q \in Q} \sum_{i=0}^{n_{q'}} w_{iq'} T_q(q'|s, u) \int_x \tau_x(dx'|s, u, q') f_{iq'}(x') dx \end{aligned}$$

La (3.10) può essere allora espressa come:

$$\begin{aligned}
 \min_{w_{iq}} \quad & \sum_{q \in Q} \sum_{i=0}^{n_q} w_{iq} \int_x \psi(q, x) f_{iq}(x) dx \\
 \text{soggetto a:} \quad & \sum_{i=0}^{n_q} w_{iq} f_{iq}(x) \geq R(s, u) + \\
 & \gamma \sum_{q' \in Q} \sum_{i=0}^{n_{q'}} w_{iq'} T_q(q' | s, u) \int_x \tau_x(dx' | s, u, q') f_{iq'}(x') dx, \forall s \in S, u \in U.
 \end{aligned} \tag{3.11}$$

Il problema HALP è *feasible* se esiste $f_{0q}(\cdot) = 1, \forall q \in Q$. Assumiamo che questa condizione sia sempre soddisfatta.

Il problema di ottimizzazione (3.11) è lineare semi-infinito, ci sono cioè un numero finito di variabili di ottimizzazione ma un numero infinito di vincoli dato che lo spazio di stato $S = Q \times X$ ha cardinalità infinita. Per risolverlo è quindi necessario ricondurre il numero di vincoli ad essere finito. Questo problema verrà discusso nella Sezione 3.4.3 in cui viene proposto un metodo randomizzato per trovare una soluzione approssimata al problema (3.11) con qualità in termini di soddisfacimento dei vincoli fissata a priori.

Uno degli aspetti critici dell'approccio HALP consiste nella scelta delle funzioni di base da utilizzare per costruire l'approssimante della value function ottima. Ciò infatti incide in maniera decisiva su:

- qualità dell'approssimazione;
- onere computazionale nella risoluzione del problema di ottimizzazione.

Cifra di merito e vincoli nella (3.11) richiedono il calcolo di integrali su tutto lo spazio di stato continuo X delle funzioni $f_{iq}(x)$ moltiplicate per la funzione peso $\psi(q, x)$ e per la probabilità di transizione del processo $\tau_x(\cdot | s, u, q')$. Nella Sezione 3.4.1 si presenteranno alcuni esempi di funzioni di base che consentono di ricavare l'espressione di tali integrali in forma analitica quando la probabilità di transizione è gaussiana e la funzione peso è gaussiana o uniforme evitando quindi di utilizzare un'approccio numerico (e di conseguenza computazionalmente più oneroso) per la loro risoluzione.

Un altro aspetto importante nell'impostazione del problema di HALP è la scelta della funzione $\psi(s), s \in S$, che compare nella cifra di merito da minimizzare del problema (3.11).

Nella Sezione 3.4.2 si discuterà il ruolo di $\psi(s)$ e il suo impatto sulla soluzione V^w .

3.4.1 SCELTA DELLE FUNZIONI DI BASE

La scelta delle funzioni di base da utilizzare per l'approssimazione della value function ottima $V^*(\cdot)$ è uno degli aspetti più importanti della programmazione lineare approssimata in quanto influenza la qualità dell'approssimazione ottenuta. In linea di principio per ottenere una buona approssimazione bisognerebbe conoscere a priori la forma della funzione ottima e scegliere di conseguenza le funzioni di base da utilizzare. Di fatto ciò non è possibile in quanto la funzione $V^*(\cdot)$ da approssimare non è nota nemmeno in modo campionario.

In questa sezione si considerano due classi di funzioni di base: le funzioni gaussiane e le funzioni polinomiali. Le funzioni gaussiane costituiscono una buona scelta per l'approssimazione locale di una funzione, dato che la funzione gaussiana decresce rapidamente per valori della variabile che si allontanano dalla sua media. Le funzioni polinomiali invece consentono di ottenere una buona approssimazione globale attraverso un'opportuna scelta dei coefficienti delle potenze che compaiono nella funzione in quanto non presentano lo stesso difetto di località che caratterizza le funzioni gaussiane. Oltre che sulla qualità dell'approssimazione, le funzioni di base adottate hanno impatto anche sul costo computazionale per la soluzione del problema. Esse infatti, come indicato nella (3.11) sono coinvolte nel calcolo di integrali sullo spazio di stato continuo X , integrali in cui compaiono come prodotto con la probabilità di transizione caratteristica del processo. Tali integrali potrebbero certamente essere calcolati mediante metodi di integrazione numerica. Tuttavia tali approcci richiedono un campionamento che cresce come $1/\epsilon^2$, dove ϵ è l'accuratezza cercata. Affinché il problema possa essere risolto in maniera computazionalmente efficiente è opportuno che le funzioni di base $f_{iq}(x)$ siano scelte in modo tale che gli integrali che compaiono nella (3.11) ammettano una soluzione in forma analitica.

Introduciamo ora l'espressione analitica dell'integrale ottenuto scegliendo funzioni di base polinomiali e gaussiane nel caso in cui la probabilità di transizione τ_x e la funzione peso sullo

stato $\psi(q, \cdot)$ siano gaussiane.

Funzioni di base polinomiali. Sia:

$$f(x) = x^n$$

e:

$$P(x) = N(x; \mu, \sigma)$$

dove $N(x; \mu, \sigma)$ indica la gaussiana monodimensionale con media μ e varianza σ :

$$N(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Allora, il valore atteso $E_{P(x)}[f(x)]$ calcolato su un intervallo $[a, b]$ ha un'espressione di tipo ricorsivo:

$$\int_a^b f(x) P(x) dx = \mu \int_a^b x^{n-1} N(x|\mu, \sigma) dx + (n-1) \sigma^2 \int_a^b x^{n-2} N(x|\mu, \sigma) dx - \sigma^2 [x^{n-1} N(x|\mu, \sigma)]_a^b \quad (3.12)$$

Dimostrazione. Calcoliamo l'espressione della derivata:

$$\frac{d}{dx} [x^{n-1} N(x; \mu, \sigma)]$$

Essa sarà pari a:

$$\frac{d}{dx} [x^{n-1} N(x; \mu, \sigma)] = (n-1) x^{n-2} N(x; \mu, \sigma) + x^{(n-1)} \frac{d}{dx} N(x; \mu, \sigma) \quad (3.13)$$

In particolare, il termine relativo alla derivata della funzione gaussiana sarà pari a:

$$\frac{d}{dx} N(x; \mu, \sigma) = \frac{d}{dx} \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \right] = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \left[-\frac{x}{\sigma^2} + \frac{\mu}{\sigma^2} \right]$$

Sostituendo nella (3.13):

$$\frac{x^n}{\sigma^2} N(x; \mu, \sigma) = (n-1)x^{n-2} N(x; \mu, \sigma) + \frac{\mu}{\sigma^2} N(x; \mu, \sigma) - \frac{\partial}{\partial x} [x^{n-1} N(x; \mu, \sigma)]$$

Moltiplicando ambo i membri per σ^2 , si ha:

$$x^n N(x; \mu, \sigma) = \sigma^2 (n-1)x^{n-2} N(x; \mu, \sigma) + \mu N(x; \mu, \sigma) - \sigma^2 \frac{d}{dx} [x^{n-1} N(x; \mu, \sigma)]$$

Applicando l'operatore di integrale ad entrambi i membri si ottiene la (3.12):

$$\int_a^b x^n N(x; \mu, \sigma) dx = \mu \int_a^b x^{n-1} N(x; \mu, \sigma) dx + (n-1)\sigma^2 \int_a^b x^{n-2} N(x; \mu, \sigma) dx - \sigma^2 [x^{n-1} N(x; \mu, \sigma)]_a^b$$

Nel caso in cui $n=1$, la (3.12) si riconduce a:

$$\int_a^b x N(x; \mu, \sigma) dx = \mu \int_a^b N(x; \mu, \sigma) dx - \sigma^2 [N(x; \mu, \sigma)]_a^b$$

I casi in cui $n > 1$, possono essere risolti ricorsivamente partendo da tale risultato. Quando l'integrazione è effettuata sull'intero dominio di x , vale a dire quando gli estremi di integrazione sono $+\infty$ e $-\infty$ basta tener conto del fatto che:

$$\lim_{\substack{b \rightarrow +\infty \\ a \rightarrow -\infty}} [x^{n-1} N(x; \mu, \sigma)]_a^b = 0 \qquad \lim_{\substack{b \rightarrow +\infty \\ a \rightarrow -\infty}} [N(x; \mu, \sigma)]_a^b = 0$$

Funzioni di base gaussiane. Sia:

$$f(x) = N(x; \mu_1, \sigma_1), \quad P(x) = N(x; \mu_2, \sigma_2)$$

Allora, il termine di valore atteso $E_{P(x)}[f(x)]$ calcolato su un intervallo $[a, b]$ ha l'espressione:

$$\int_a^b f(x) P(x) dx = c_p \int_a^b N(x; \mu_p, \sigma_p) dx \tag{3.14}$$

dove:

$$c_P = \exp\left(\frac{-(\mu_1 - \mu_2)^2}{2(\sigma_1 + \sigma_2)} - \frac{1}{2} \ln(2\pi(\sigma_1 + \sigma_2))\right) \quad (3.15)$$

$$\sigma_P = \frac{\sigma_1 \sigma_2}{\sigma_1 + \sigma_2} \quad (3.16)$$

$$\mu_P = \frac{\mu_1 \sigma_2 + \mu_2 \sigma_1}{\sigma_1 + \sigma_2} \quad (3.17)$$

vale a dire che il prodotto di due funzioni gaussiane è ancora una gaussiana di media μ_P e varianza σ_P ma riscalata di un fattore c_P .

Dimostrazione. Assegnate due funzioni gaussiane:

$$N_1(x; \mu_1, \sigma_1) \quad N_2(x; \mu_2, \sigma_2)$$

Scriviamone il prodotto in forma esplicita:

$$\begin{aligned} N(x|\mu_1, \sigma_1) N(x|\mu_2, \sigma_2) &= \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \\ &= \frac{1}{\sqrt{2\pi}\sigma_1^2\sigma_2^2 2\pi} e^{-\frac{\sigma_2^2(x-\mu_1) - \sigma_1^2(x-\mu_2)^2}{2\sigma_1^2\sigma_2^2}} \end{aligned}$$

Essendo $(\sigma_1^2 + \sigma_2^2) > 0$, possiamo moltiplicare e dividere il secondo membro per questa quantità ottenendo:

$$\begin{aligned} & \frac{1}{\sqrt{2\pi} \frac{\sigma_1^2 \sigma_2^2}{\sigma_1 + \sigma_2} (\sigma_1^2 + \sigma_2^2) 2\pi} e^{\frac{-\sigma_2^2(x-\mu_1) - \sigma_1^2(x-\mu_2)^2}{\sigma_1^2 + \sigma_2^2} \frac{2\sigma_1^2 \sigma_2^2}{\sigma_1 + \sigma_2}} \\ &= \frac{1}{\sqrt{2\pi} \frac{\sigma_1^2 \sigma_2^2}{\sigma_1 + \sigma_2}} \frac{1}{\sqrt{2\pi} (\sigma_1^2 + \sigma_2^2)} e^{\frac{-\sigma_2^2(x-\mu_1) - \sigma_1^2(x-\mu_2)^2}{\sigma_1^2 + \sigma_2^2} \frac{2\sigma_1^2 \sigma_2^2}{\sigma_1 + \sigma_2}} \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{\sqrt{2\pi \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} e^{\ln\left(\frac{1}{\sqrt{2\pi(\sigma_1^2 + \sigma_2^2)}}\right)} e^{\frac{-\sigma_2^2(x-\mu_1) - \sigma_1^2(x-\mu_2)^2}{\sigma_1^2 + \sigma_2^2} - \frac{2\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \\
 &= \frac{1}{\sqrt{2\pi \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} e^{-\frac{1}{2} \ln(2\pi(\sigma_1^2 + \sigma_2^2)) + \frac{-\sigma_2^2(x-\mu_1) - \sigma_1^2(x-\mu_2)^2}{\sigma_1^2 + \sigma_2^2} - \frac{2\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}
 \end{aligned}$$

Concentriamoci ora sull'esponente di questa espressione:

$$\frac{-\frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \ln(2\pi(\sigma_1^2 + \sigma_2^2)) + \frac{-\sigma_2^2}{\sigma_1^2 + \sigma_2^2} (x - \mu_1)^2 - \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} (x - \mu_2)^2}{2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}$$

Espandendo il quadrato e scrivendo il numeratore come polinomio in x , otteniamo:

$$\frac{-x^2 + 2x\left(\frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right) - \frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2 + \ln(2\pi(\sigma_1^2 + \sigma_2^2))}{\sigma_1^2 + \sigma_2^2}}{2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}$$

Adesso ricordiamo che un polinomio della forma $-x^2 + 2Ax - (A^2 + C)$ può essere anche scritto come $-(x - A)^2 + C$ e poniamo:

$$\begin{aligned}
 A &= \frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2} \\
 A^2 + C &= \frac{\mu_1 \sigma_2^2 + \mu_2 \sigma_1^2 + \ln(2\pi(\sigma_1^2 + \sigma_2^2))}{\sigma_1^2 + \sigma_2^2}
 \end{aligned}$$

Sostituendo l'espressione di A e rielaborando:

$$C = 2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} (\mu_1 - \mu_2)^2 + 2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \ln(2\pi(\sigma_1^2 + \sigma_2^2))$$

Abbiamo ora l'espressione di A e C. Il prodotto delle tue gaussiane avrà quindi la forma:

$$\frac{1}{\sqrt{2\pi \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} e^{-\frac{(x-A)^2 - C}{2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} = \frac{1}{\sqrt{2\pi \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} e^{-\frac{(x-A)^2}{2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} e^{-\frac{-C}{2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} = \frac{1}{\sqrt{2\pi \sigma_p^2}} e^{-\frac{(x-\mu_p)^2}{2\sigma_p^2}} c_p$$

che è l'espressione di una gaussiana con media A pari alla (3.17) e varianza σ_p data dalla (3.16), riscalata di un fattore c_p dato dalla (3.15) essendo:

$$e^{-\frac{-C}{2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} = e^{\frac{2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} (\mu_1 - \mu_2)^2 + 2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \ln(2\pi(\sigma_1^2 + \sigma_2^2))}{2 \frac{\sigma_1 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}}} = e^{-\frac{-(\mu_1 - \mu_2)^2}{2(\sigma_1 + \sigma_2)} - \frac{1}{2} \ln(2\pi(\sigma_1 + \sigma_2))}$$

I conti effettuati si riferiscono al caso scalare in cui τ_x è una distribuzione gaussiana monodimensionale ($x \in \mathbb{R}$). Il caso in cui $x \in \mathbb{R}^n$, $n > 1$, può essere ricondotto a quello scalare se si effettua una trasformazione V $n \times n$, con $\det(V) \neq 0$, tale che la probabilità di transizione gaussiana:

$$\frac{1}{\sqrt[n]{2\pi \det \Sigma^2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)},$$

$\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n \times n}$ con $\Sigma = \Sigma^T > 0$, in $z = Vx$ diventi il prodotto di n gaussiane monodimensionali. Basta in effetti procedere nel modo seguente:

Fattorizzo Σ come $\Sigma = V^T V$, dove $V^{-1} = V^T$, e poi osservo che:

$$(x-\mu)^T \Sigma^{-1}(x-\mu) = (x-\mu)^T (V^{-1})(V^T)^{-1}(x-\mu) = (Vx - V\mu)^T (Vx - V\mu)$$

da cui segue:

$$N(x; \mu, \Sigma) = N(Vx; V\mu, I) \frac{1}{\det \Sigma^{\frac{1}{2}}}$$

dove $N(\cdot; \tilde{\mu}, I)$ è il prodotto di n gaussiane monodimensionali con media data da $\tilde{\mu}$ e varianza unitaria.

NOTA. Nel caso in cui $\psi(q, \cdot)$ sia una distribuzione di probabilità uniforme sull'intervallo $[a, b]$ e $f_{iq}(\cdot)$ sia gaussiana, allora per calcolare la cifra di merito in (3.11) si può utilizzare (caso monodimensionale):

$$\int_x U(a, b) f_{iq}(x) dx = \frac{1}{b-a} \int_a^b f_{iq}(x) dx$$

Se $f_{iq}(\cdot)$ è polinomiale gli integrali sono facilmente esprimibili in modo analitico.

3.4.2 SCELTA DELLA FUNZIONE PESO SULLO STATO

Nell'ambito dell'ALP la scelta della funzione peso sullo stato che compare nell'espressione della funzione obiettivo del problema di ottimizzazione influenza la qualità dell'approssimazione. Ricordiamo la formulazione del problema HALP:

$$\begin{aligned} \min_w \quad & E_{\psi(s)}[V^w(s)] \\ \text{soggetto a:} \quad & V^w(s) \geq R(s, u) + E_{s' \sim T_s(\cdot|s, u)}[V^w(s')], \forall s \in S, u \in U \end{aligned} \quad (3.18)$$

Vale allora il seguente lemma:

Lemma: Se un vettore \tilde{w} è soluzione del problema di ottimizzazione vincolato (3.18), allora è soluzione anche di:

$$\begin{aligned} \min_w \quad & \|V^w(s) - V^*(s)\|_{\psi(s)} \\ \text{soggetto a:} \quad & V^w(s) \geq T[V^w(\cdot)](s), \quad \forall s \in S, u \in U \end{aligned}$$

dove la norma $\|\cdot\|_{\psi(s)}$ è definita come:

$$\|V(s)\|_{\psi(s)} = \int_S \psi(s) |V(s)| ds$$

Dimostrazione. Consideriamo il vincolo nella (3.18) espresso come:

$$V^w(s) \geq T[V^w(\cdot)](s), \quad \forall s \in S, u \in U$$

Per la proprietà di monotonia dell'operatore di Bellman e di convergenza a $V^*(s)$ della sua ripetuta applicazione, si ha:

$$V(s) \geq T[V(\cdot)](s) \geq T^2[V(\cdot)](s) \geq \dots \geq T^n[V(\cdot)](s) \geq \dots \geq V^*(s), \quad \forall s \in S.$$

Si può quindi riformulare il problema (3.18) come:

$$\begin{aligned} \min_w \quad & E_{\psi(s)}[V^w(s)] \\ \text{soggetto a:} \quad & V^w(s) \geq V^*(s), \quad \forall s \in S, u \in U \end{aligned}$$

Possiamo sottrarre alla funzione obiettivo del problema ora introdotto la costante $E_{\psi(s)}[V^*(s)]$ senza modificare la soluzione che si ottiene risolvendo il problema:

$$\begin{aligned} \min_w \quad & E_{\psi(s)}[V^w(s)] - E_{\psi(s)}[V^*(s)] \\ \text{soggetto a:} \quad & V^w(s) \geq V^*(s), \quad \forall s \in S \end{aligned} \tag{3.19}$$

Concentriamoci sulla funzione obiettivo scrivendo in forma esplicita il valore atteso:

$$\begin{aligned} E_{\psi(s)}[V^w(s)] - E_{\psi(s)}[V^*(s)] &= \int_S \psi(s) V^w(s) ds - \int_S \psi(s) V^*(s) ds \\ &= \int_S \psi(s) [V^w(s) - V^*(s)] ds \end{aligned}$$

Dato che la soluzione della (3.19) soddisfa il vincolo $V^w(s) \geq V^*(s), \forall s \in S$, allora $[V^w(s) - V^*(s)] = |V^w(s) - V^*(s)|$ e quindi:

$$E_{\psi(s)}[V^w(s)] - E_{\psi(s)}[V^*(s)] = \int_S \psi(s) |V^w(s) - V^*(s)| ds$$

che è proprio la norma $\|V^w(s) - V^*(s)\|_{\psi(s)}$

Abbiamo quindi dimostrato che è possibile, attraverso la scelta di $\psi(s)$, attribuire una diversa importanza alla qualità dell'approssimazione in diverse regioni dello spazio di stato.

3.4.3 RIDUZIONE DEL NUMERO DI VINCOLI

Il numero dei vincoli coinvolti nella soluzione dell'ALP nel caso di ibrido è infinito in quanto lo stato assume valori in $S=Q \times X$ e X è uno spazio di stato continuo. Occorre dunque ridurre il numero dei vincoli a finito introducendo una formulazione *rilassata* del problema di partenza:

$$\begin{aligned} \min_w \quad & E_{\psi(s)}[V^w(s)] \\ \text{soggetto a:} \quad & V^w(s) \geq R(s, u) + \gamma E_{s' \sim T_s(\cdot|s, u)}[V^w(s')], \quad (s, u) \in C, \end{aligned}$$

in cui viene considerato solo un sottoinsieme C di coppie stato-azione (s, u) con $|C| < \infty$.

Descriviamo ora alcuni dei metodi più utilizzati per la soluzione al problema della riduzione del numero di vincoli.

Campionamento Monte Carlo. Un modo semplice di ridurre il numero di vincoli coinvolti nel problema di ottimizzazione è quello offerto dal campionamento secondo il metodo Monte Carlo. Nello specifico si campionano le coppie stato-azione secondo una distribuzione di probabilità assegnata φ . In questo modo lo spazio dei vincoli viene ridotto e si considera solo un sottoinsieme finito C di quelli di partenza. Relativamente ai modelli con spazio di stato discreto, Farias and Van Roy ([7]) hanno dimostrato in un teorema la qualità del risultato che si può ottenere mediante questo campionamento:

Teorema: Sia w^* la soluzione del problema di ALP e \hat{w} quella dello stesso problema rilassato ottenuto campionando le coppie stato-azione secondo una distribuzione φ . Allora esiste una distribuzione φ e un numero di campioni N , che soddisfa:

$$N \geq O\left(\frac{U\theta}{(1-\gamma)\epsilon} \left(K \ln \frac{U\theta}{(1-\gamma)\epsilon} + \ln \frac{1}{\beta}\right)\right)$$

tali che con probabilità almeno $(1-\beta)$, vale la seguente disuguaglianza:

$$\|V^* - V^{\hat{w}}\|_{1,\phi} \leq \|V^* - V^{w^*}\|_{1,\phi} + \epsilon \|V^*\|_{1,\phi}$$

dove $\|\cdot\|_{1,\phi}$ è una norma pesata secondo il peso ϕ , θ è una costante specifica del problema, U e K sono rispettivamente il numero delle azioni e delle funzioni di base, δ e ϵ degli scalari nell'intervallo $(0,1)$.

Trovare la distribuzione ϕ che soddisfa questo teorema tuttavia è complicato e richiede in pratica di conoscere la soluzione al problema. Inoltre il risultato non è stato ancora esteso al caso ibrido.

ϵ -HALP. Un metodo alternativo consiste nel procedere ad una discretizzazione delle variabili di stato continue. Le variabili vengono campionate secondo una griglia uniforme di ampiezza ϵ e nel problema di ottimizzazione si considerano soltanto i vincoli relativi ai valori dello stato sulla griglia.

Il metodo della grigliatura presenta lo svantaggio di essere computazionalmente costoso. Infatti esso comporta che vengano selezionati, tra quelli di partenza, un numero di vincoli che è un $O([1/\epsilon+1]^{n_x})$, dove n_x rappresenta la dimensione dello spazio di stato continuo.

3.4.4 SOLUZIONE RANDOMIZZATA BASATA SULL'APPROCCIO A SCENARIO

Un metodo alternativo per la riduzione del numero dei vincoli consiste nel ricorrere alla *Teoria a Scenario* ([11]), che viene qui brevemente richiamata. Consideriamo un problema del tipo:

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & c^T w \\ \text{soggetto a:} \quad & f_\delta(w) \leq 0 \quad \forall \delta \in \Delta \end{aligned} \quad (3.20)$$

in cui, per ogni valore del parametro incerto δ in Δ , $f_\delta(w)$ è una funzione convessa in $w \in \mathbb{R}^d$. Spesso accade che l'insieme Δ sia continuo e contenga infiniti elementi, così che il problema (3.20) risulta semi-infinito e di difficile risoluzione seppure convesso.

L'approccio a scenario prevede una riduzione del numero di vincoli del problema in esame ottenuto attraverso un opportuno campionamento dell'incertezza $\delta \in \Delta$. In particolare, esso consente di ottenere una soluzione approssimata del problema semi-infinito originario che soddisfa i vincoli associati alla maggior parte dei δ eccetto un insieme di dimensione prefissata.

Per applicare la teoria, bisogna fissare una distribuzione di probabilità Pr su Δ e campionare un numero N di scenari $\delta^{(i)}, i=1, \dots, N$, in accordo alla distribuzione di probabilità scelta. Il problema di ottimizzazione verrà risolto imponendo che la soluzione soddisfi soltanto i vincoli associati agli scenari $\delta^{(i)}, i=1, \dots, N$, estratti. Si passa in tal modo alla formulazione rilassata:

$$\begin{aligned} \min_{w \in \mathbb{R}^d} \quad & c^T w \\ \text{soggetto a:} \quad & f_{\delta^{(i)}}(w) \leq 0 \quad i=1, 2, \dots, N \end{aligned} \quad (3.21)$$

del problema semi-infinito iniziale (3.20).

In tal modo ci si è ricondotti ad un problema di ottimizzazione convesso *finito* che può essere risolto ricorrendo ai metodi standard con un costo computazionale ragionevole, a patto che N non sia eccessivamente elevato.

L'aspetto più importante della teoria a scenario risiede nel fatto che il numero di campioni è determinato dall'utente sulla base di due considerazioni: l'entità dei vincoli che si desidera siano soddisfatti e la probabilità con cui si vuole che ciò accada. Vale infatti il seguente teorema:

Teorema: Scelto un parametro di violazione $\epsilon \in (0,1)$ e un parametro di confidenza

$\beta \in (0,1)$, se N soddisfa:

$$\sum_{i=0}^{d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta, \quad (3.22)$$

dove d è il numero delle variabili di ottimizzazione, allora, con probabilità $1-\beta$, tutti i vincoli della (3.20) sono soddisfatti tranne al più una frazione ϵ .

Un'espressione esplicita della (3.22) (seppur conservativa) è la seguente:

$$N \geq \frac{2}{\epsilon} \left(\ln \frac{1}{\beta} + d \right) \quad (3.23)$$

Analizzando il teorema possiamo individuare meglio il significato dei due parametri ϵ e β . Trascurando per un momento quest'ultimo infatti, la (3.23) ci dice semplicemente che il numero di campioni da considerare cresce con il grado di robustezza ϵ con cui si vuole che la soluzione ottenuta si comporti rispetto agli scenari ammissibili per il processo. Diminuire tale parametro equivale a richiedere una soluzione più robusta, che non soddisfi solo una quantità molto limitata di scenari. Ovviamente vale che:

$$\lim_{\epsilon \rightarrow 0} N(\epsilon) = \infty$$

Riguardo al ruolo di β , si può notare come la soluzione del problema di ottimizzazione a scenario (3.21) sia una variabile casuale in quanto essa dipende dai vincoli estratti randomicamente.

Può accadere di estrarre N volte lo stesso vincolo e di conseguenza quanto affermato dal teorema non sarebbe più valido in quanto la porzione dei vincoli violati sarebbe molto più grande di ϵ . Per questo motivo si introduce il parametro β che controlla la probabilità di questi eventi "sfortunati" e il risultato vale con confidenza $1-\beta$. Anche in questo caso, il comportamento $\beta=0$ ideale implicherebbe un numero infinito di vincoli. Dalla (3.23) tuttavia si vede che il parametro β compare come argomento del logaritmo naturale. Quindi anche la scelta di valori per β quali 10^{-10} (praticamente nulli) non implica un valore di N

eccessivamente elevato. Nella figura seguente si riporta una rappresentazione grafica ed intuitiva di quanto espresso dal Teorema:

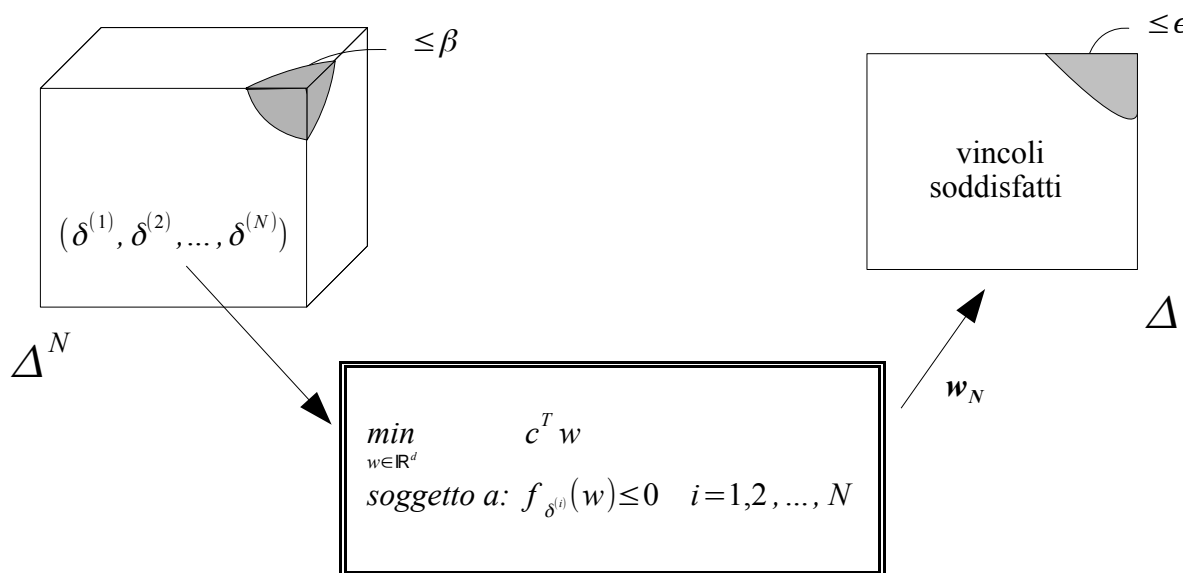


Figura 3.1: Rappresentazione grafica del Teorema.

Interpretando come incertezza $\delta \in \Delta$ il valore assunto da $(s, u) \in S \times U$, il metodo a scenario può essere applicato nella soluzione del problema HALP (3.11). In effetti, la funzione obiettivo da minimizzare è una combinazione lineare delle variabili di ottimizzazione w e i vincoli sono anch'essi funzioni lineari (e quindi convesse) di w per ogni valore di $(s, u) \in S \times U$.

Nella figura seguente si riporta la versione a scenario dell'HALP (3.11) nel caso in cui lo spazio di controllo U è finito e la funzione peso $\psi(s)$ soddisfa $\psi(q, \cdot) = \bar{\psi}(\cdot), \forall q \in Q$. La probabilità utilizzata per estrarre gli scenari è $\bar{\psi}(\cdot)$. Il problema così formulato può essere facilmente risolto mediante l'uso di algoritmi per la soluzione di problemi di programmazione lineare (ad esempio CPLEX).

HALP con Approccio a Scenario

1) Scegliere i parametri di violazione $\epsilon \in (0,1)$ e di confidenza $\beta \in (0,1)$;

2) Calcolare:

$$N = \left\lceil \frac{2}{\epsilon} \left(\ln \frac{1}{\beta} + d \right) \right\rceil$$

dove $d = \sum_{q \in Q} n_q$ è il numero dei parametri w_{iq} da ottimizzare;

3) Estrarre N valori dello stato continuo X in accordo con la distribuzione $\bar{\psi}(\cdot)$ ottenendo lo spazio continuo campionato $X_N = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$;

4) Risolvere il problema ALP rilassato:

$$\begin{aligned} \min_{w_{iq}} \quad & \sum_{q \in Q} \sum_{i=0}^{n_q} w_{iq} \int_X \psi(q, x) f_{iq}(x) dx \\ \text{soggetto a:} \quad & \sum_{i=0}^{n_q} w_{iq} f_{iq}(x) \geq R(s, u) + \\ & \gamma \sum_{q' \in Q} \sum_{i=0}^{n_{q'}} w_{iq'} T_q(q' | s, u) \int_X \tau_x(dx' | s, u, q') f_{iq'}(x') dx, \forall x \in X_N, q \in Q, u \in U. \end{aligned}$$

Figura 3.2: Algoritmo di HALP con Approccio a Scenario.

3.5 APPLICAZIONE AL PROBLEMA DELL'INVARIANZA CONTROLLATA

Consideriamo il problema di voler determinare una politica per un DTSHS che massimizzi la probabilità che il suo stato rimanga all'interno di un insieme A_s che indicheremo come insieme safe. Nel caso in cui l'orizzonte temporale considerato sia infinito e l'insieme A_s sia compatto non è sempre possibile determinare una politica che renda tale insieme un invariante rispetto alla dinamica stocastica in anello chiuso. Più precisamente se la probabilità di transizione dello stato continuo ha supporto non limitato e porta lo stato fuori da un compatto con probabilità $\geq \epsilon (> 0)$ ad ogni istante, allora con probabilità 1 lo stato prima o poi

uscirà da un qualsiasi insieme compatto nel corso della sua evoluzione. L'approccio che si può adottare allora è quello definito come strategia *self-recovery* in [13]. Non potendo impedire che lo stato del sistema abbandoni l'insieme safe A_s , si fa in modo che una volta che ciò accade lo stato torni il prima possibile all'interno di esso. Nello specifico tale obiettivo è raggiunto massimizzando il valore atteso di una funzione di reward su un orizzonte temporale infinito con un assegnato fattore di discount in cui le transizioni che riportano il sistema all'interno di A_s sono premiate, e tanto più quanto prima intervengono, mentre quelle che lo conducono all'esterno di esso vengono penalizzate, tanto più quanto prima avvengono.

Formalizziamo ora meglio l'obiettivo di controllo. Consideriamo il sistema stocastico a tempo discreto $H=(S, U, T_s)$, un insieme compatto $A_s \subset S$ di stati "safe" e una politica stazionaria $\pi=(\mu, \mu, \dots)$ con $\mu: S \rightarrow U$.

Per avere una politica di controllo che permetta di massimizzare la probabilità che lo stato di H rimanga in A_s si richiede che essa massimizzi il funzionale di reward scontato:

$$V_\mu(s) = E_\mu \left[\sum_{t=0}^{\infty} \gamma^t \rho(s_t, s_{t+1}) \middle| s_0 = s \right], \quad \forall s \in S \quad (3.24)$$

dove $\gamma \in (0,1)$ è il fattore di discount e $\rho: S \times S \rightarrow [-1,1]$ è definita come:

$$\rho(s, s') = \begin{cases} -1, & s \in A_s \text{ e } s' \notin A_s \\ 1, & s \notin A_s \text{ e } s' \in A_s \\ 0, & \text{altrimenti} \end{cases}$$

In tal modo, le transizioni che portano lo stato nell'insieme safe vengono premiate con un valore +1, quelle che lo portano all'esterno sono penalizzate con il valore -1, quelle che lo mantengono all'interno o all'esterno non sono né penalizzate né ricompensate.

La funzione di reward (3.24) può essere riscritta nella forma:

$$E_\mu \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mu(s_t)) \middle| s_0 = s \right]$$

definendo $R: S \times U \rightarrow \mathbb{R}$ nel modo seguente:

$$R(s, u) = \int_S \rho(s, s') T_s(ds' | s, u)$$

La politica ottima può essere quindi ricavata mediante le tecniche di programmazione dinamica approssimata viste nelle sezioni precedenti.

Un ruolo importante è svolto in (3.24) dal fattore di sconto γ che riscalda i premi/penalità rispetto al tempo. In particolare esso fa in modo che lo stato tenda a rientrare il prima possibile nell'insieme degli stati "safe" e ad uscirne il più tardi possibile. Così com'è formulato inoltre, il funzionale di costo fa in modo che il sistema cerchi di mantenere il suo stato all'interno dell'insieme safe A_s il più al lungo possibile. Per comprendere ciò, notiamo che la funzione di reward scontata (3.24) valutata in $s \in A_s$ può essere espressa anche come:

$$V_\mu(s) = \sum_{t=0}^{\infty} J_\mu(n, s) P_\mu(\tau_{I-uscita} = n | s_0 = s) \quad (3.25)$$

dove $P_\mu(\tau_{I-uscita} = n | s_0 = s)$ è la probabilità che lo stato esca per la prima volta da A_s all'istante n e:

$$J_\mu(n, s) = E_\mu \left[\sum_{t=0}^{\infty} \gamma^t \rho(s_t, s_{t+1} | \tau_{I-uscita} = n, s_0 = s) \right]$$

è la funzione di reward media scontata condizionata rispetto al fatto che lo stato è uscito dell'insieme A_s all'istante $\tau_{I-uscita} = n$. Se lo stato iniziale del sistema si trova in A_s ($s \in A_s$), la funzione $J_\mu^0(n) = E_\mu \left[\sum_{t=n-1}^{\infty} \gamma^t \rho(s_t, s_{t+1} | \tau_{I-uscita} = n, s_0 = s) \right]$ soddisfa la seguente relazione:

$$-\gamma^{n-1} \leq J_\mu(n, s) \leq -\gamma^{n-1} + \gamma^n = -\gamma^{n-1}(1 - \gamma)$$

Il limite inferiore corrisponde al caso in cui lo stato continua a restare fuori dall'insieme safe una volta uscito all'istante n , e il limite superiore al caso in cui esso rientra subito in A_s all'istante $n + 1$ essendone uscito all'istante n immediatamente precedente. Per ogni γ poi,

esiste $\Delta n \geq 1$ tale che:

$$J_{\mu}(n, s) \leq -\gamma^{n-1}(1-\gamma) \leq -\gamma^{n+\Delta n-1} \leq J_{\mu}(n+\Delta n, s)$$

In particolare, per $\gamma=0.5, \Delta n=1$ e $\{J_{\mu}(n, s), n=0,1, \dots\}$ che compare nella (3.25) cresce, saturando rapidamente a 0.

Per minimizzare la funzione di costo, quindi, la massa della distribuzione di probabilità dell'istante di prima uscita dall'insieme safe dovrebbe essere concentrata su valori elevati di $\tau_{I-uscita}$ e, allo stesso tempo, lo stato del sistema dovrebbe essere riportato all'interno di A_s dopo l'uscita da esso nel più breve tempo possibile per poter raggiungere il limite superiore di $J_{\mu}(n, s)$.

4

ANALISI DELLA HALP A SCENARIO IN UN CASO DI STUDIO

In questo capitolo le tecniche di programmazione lineare approssimata ibrida introdotte nel Capitolo 3 vengono applicate ad un caso di studio per valutare i risultati ottenibili mediante il loro utilizzo e l'impatto di diverse scelte delle funzioni di base e della funzione peso.

Nello specifico si farà riferimento al problema del controllo della temperatura di un numero assegnato di stanze presentato in [2] e si effettuerà un confronto con quanto ottenuto mediante il metodo della value iteration approssimata con grigliatura dello spazio di stato continuo illustrato nella Sezione 3.3 del Capitolo 3.

4.1 DESCRIZIONE DEL CASO DI STUDIO

Consideriamo il problema di controllare la temperatura in n_s stanze. Ipotizziamo che in ogni stanza sia presente un elemento riscaldante e che, in ogni istante, al più uno di essi possa essere in funzione. L'obiettivo del controllo consiste nell'individuare una strategia di switching che decida quale elemento riscaldante attivare sulla base delle misure delle

temperature delle n_s stanze in modo da massimizzare la probabilità che tali temperature rimangano in un range prestabilito.

Il sistema sotto controllo può essere modellizzato mediante un sistema ibrido stocastico a tempo discreto $H=(Q, n, U, T_x, T_q, R)$ i cui componenti sono descritti qui di seguito.

La componente discreta dello spazio di stato è definita come $Q=\{1,2,\dots,n_s+1\}$. Se $q=i, i=1,\dots,n_s$, la i -esima stanza viene riscaldata, se $q=n_s+1$ nessuna stanza viene riscaldata. La mappa $n:Q\rightarrow\mathbb{N}$, che definisce la dimensione della componente continua dello spazio di stato ibrido, è la mappa costante $n(q)=n_s, \forall q\in Q$. La componente continua $x\in\mathbb{R}^{n_s}$ infatti rappresenta la temperatura (media) nelle n_s stanze.

$U=\{1,\dots,n_s+1\}$ rappresenta lo spazio di controllo. In particolare l'ingresso $u=i, i=1,\dots,n_s$, indica il comando di attivazione dell'elemento riscaldante nell' i -esima stanza, mentre $u=n_s+1$ indica il comando di spegnimento di tutti gli elementi riscaldanti. In particolare, se il sistema sta operando nel modo $q=i$ e viene applicata l'azione di controllo $u=i$, l'elemento riscaldante nella stanza i , che era già acceso, continua ad esserlo lasciando la situazione invariata.

L'evoluzione dello stato continuo $x\in X=\mathbb{R}^{n_s}$ nel modo $q\in Q$ è governata dalla probabilità di transizione $T_x:B(X)\times S\rightarrow[0,1]$ che assegna ad ogni stato una misura di probabilità $T_x(\cdot|q, x)$ sullo spazio boreliano $(X, B(X))$. $T_x(\cdot|x, u)$ è definita implicitamente tramite le equazioni lineari alle differenze stocastiche:

$$x_{i,k+1}=x_{i,k}+\left[b_i(x_a-x_{i,k})+c_i h_{i,k}+\sum_{j=1,\dots,n_s; j\neq i} a_{ij}(x_{j,k}-x_{i,k})\right]\Delta t+\eta_{i,k}, \quad i=1,\dots,n_s, \quad (4.1)$$

ottenute discretizzando secondo il metodo Eulero-Maruyama a passo costante le corrispondenti equazioni differenziali stocastiche a tempo continuo. Nella (4.1), $x_{i,k}$ rappresenta la temperatura media nella stanza i all'istante k , x_a rappresenta la temperatura ambiente (che si assume costante) e $h_{i,k}$ è una funzione booleana che vale 1 quando $q_k=i$, cioè quando l'elemento riscaldante nella stanza i è acceso, e 0 altrimenti. Gli elementi

a_{ij}, c_i e b_i sono delle costanti non negative che rappresentano i coefficienti di scambio termico della stanza i con l'ambiente (b_i) e con le altre stanze (a_{ij}) e la potenza in ingresso alla stanza i (c_i) fornita dall'elemento riscaldante in essa presente, normalizzati rispetto alla capacità termica della stanza i . Il rumore $\eta_{i,k}$ sulla temperatura è un processo gaussiano costituito da una successione di variabili casuali indipendenti e identicamente distribuite con media nulla e varianza $v^2 \Delta t$. Si assume, per semplicità, che i rumori agenti sulla temperatura di stanze diverse siano indipendenti.

In accordo con l'equazione (4.1):

$$T_x(\cdot|q, x) = N(\cdot; x + \Xi x + \Gamma(q), v^2 \Delta t I), \quad (4.2)$$

dove Ξ è una matrice quadrata di dimensione n_s , $\Gamma(q)$ è un vettore colonna di dimensione n_s che dipende da $q \in Q$. $N(\cdot; m, \Sigma)$ indica una misura di probabilità sullo spazio $(\mathbb{R}^{n_s}, B(\mathbb{R}^{n_s}))$ associata ad una distribuzione gaussiana multivariata di media m e matrice di covarianza Σ . Il fatto che la probabilità di transizione T_x descritta nella (4.2) sia gaussiana deriva dalla linearità dell'equazione (4.1) e dal fatto che i disturbi stocastici $\eta_{i,k}$ agenti sulle temperature nelle stanze sono congiuntamente gaussiani.

Nello specifico, l'elemento della riga i e della colonna j della matrice Ξ è dato da:

- $[\Xi]_{ij} = a_{ij}$, se $j \neq i$, $i = 1, \dots, n_s$;
- $[\Xi]_{ij} = -b_i - \sum_{k \neq i, k \in Q} a_{ik}$, se $j = i$, $i = 1, \dots, n_s$;

Riguardo al vettore $\Gamma(q)$, le sue componenti sono date da:

- $[\Gamma(q)]_i = (b_i x_a + c_i)$, se $q = i$, $i = 1, \dots, n_s$;
- $[\Gamma(q)]_i = b_i x_a$, se $q \neq i$, $i = 1, \dots, n_s$

Si noti che i coefficienti a_{ij} possono essere interpretati come elementi di una matrice $A_{scambio}$ di dimensioni $n_s \times n_s$ che descrive la topologia e la disposizione relativa delle stanze:

$$A_{scambio} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n_s} \\ a_{21} & a_{22} & \cdots & a_{2n_s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_s 1} & \cdots & \cdots & a_{n_s n_s} \end{bmatrix}$$

Tale matrice avrà elementi sulla diagonale tutti nulli ($a_{ii}=0, i=1, \dots, n_s$, in quanto ogni camera non scambia calore con se stessa) ed è simmetrica (lo scambio di calore tra le camere è identico e quindi $a_{ij} = a_{ji}$). Infine, il fatto che $a_{ij} > 0$ implica che le stanze i e j sono adiacenti mentre $a_{ij} = 0$ significa che non lo sono. Per esempio se $n_s = 4$ e le 4 stanze sono disposte come in Figura 4.1 allora la matrice $A_{scambio}$ sarà di questa forma:

$$A_{scambio} = \begin{bmatrix} 0 & 0.3 & 0 & 0.5 \\ 0.3 & 0 & 0.4 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0.5 & 0 & 0 & 0 \end{bmatrix}$$

dove $a_{12} = a_{21} = 0.3$, $a_{23} = a_{32} = 0.4$ e $a_{14} = a_{41} = 0.5$.

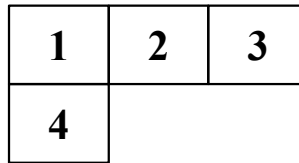


Figura 4.1: Possibile disposizione relativa di 4 stanze.

La probabilità di transizione T_q , che caratterizza l'evoluzione della componente discreta dello stato, non dipende dai valori assunti dalla componente continua. Essa è data da $T_q: Q \times Q \times U \rightarrow [0,1]$, dove $T_q(q'|q, u)$ indica la probabilità che il prossimo modo sia q' dato il modo q e il valore u dell'ingresso corrente. Assumiamo che T_q sia data da:

$$T_q(q'|q, u) = \begin{cases} 1, & u = q = q' \\ 1 - \alpha, & u \neq q, q' = q \\ \alpha, & u = q', q' \neq q \end{cases} \quad (4.3)$$

dove $\alpha \in (0,1)$ rappresenta la probabilità che il comando di commutazione (spegnimento di un

elemento riscaldante ed eventuale accensione di un altro) sia efficace.

Si assume che durante una transizione da un modo q ad un modo q' la temperatura continui ad evolvere secondo la dinamica della modo iniziale q . Ciò viene modellizzato definendo come distribuzione di probabilità per il reset:

$$R(\cdot|(q, x), q') = T_x(\cdot|(q, x)), \quad q, q' \in Q, x \in \mathbb{R}^n \quad (4.4)$$

Sulla base delle probabilità di transizione definite nella (4.2), nella (4.3) e nella (4.4) è possibile definire la probabilità di transizione controllata $T_s: B(S) \times S \times U \rightarrow [0,1]$ che assegna una misura di probabilità sullo spazio $(S, B(S))$, condizionata al valore corrente dello stato $s = (q, x) \in S$ e il valore $u \in U$ dell'ingresso, nel modo seguente:

$$T_s((q', dx)|(q, x), u) = T_x(dx|(q', x)) T_q(q'|q, u) \quad (4.5)$$

Il problema che viene affrontato è quello di progettare la politica stazionaria $\mu: S \rightarrow U$ tale da mantenere la temperatura di ciascuna stanza in un intervallo $[x_L, x_U]$ prefissato il più a lungo possibile e riportarla entro tale intervallo prima possibile una volta che essa ne è uscita. Si tratta quindi del problema dell'invarianza probabilistica descritto nella Sezione 3.5 dove l'insieme safe $A_s \subset S$ è definito come:

$$A_s = Q \times A, \quad A = [x_L, x_U]^n \quad (4.6)$$

e la cifra di merito da massimizzare è:

$$E_\mu \left[\sum_{k=0}^{\infty} \gamma^k \rho(s_k, s_{k+1}) \right]$$

dove:

$$\rho(s, s') = \begin{cases} -1, & s \in A_s \text{ e } s' \notin A_s \\ 1, & s \notin A_s \text{ e } s' \in A_s \\ 0, & \text{altrimenti} \end{cases} \quad (4.7)$$

4.2 SOLUZIONE MEDIANTE HALP E APPROCCIO A SCENARIO

Per risolvere il problema dell'invarianza controllata applicato all'esempio ora descritto verrà utilizzato il procedimento proposto nella Sezione 3.4.4 del Capitolo 3. Si formulerà il problema HALP e si utilizzerà l'approccio a scenario per la riduzione del numero dei vincoli. I risultati ottenuti verranno confrontati con quelli relativi ad un approccio alternativo basato sulla grigliatura della componente continua dello spazio di stato e sull' algoritmo della value iteration approssimata ibrida presentato nella Sezione 3.3 del Capitolo 3.

L'approccio ALP prevede innanzitutto la scelta della funzione peso sullo stato e delle funzioni di base da utilizzare per costruire la funzione approssimante della value function ottima.

Riguardo alla scelta della funzione peso $\psi(\cdot): S \rightarrow R$, si assume che:

$$\psi(q, x) = \bar{\psi}(x) \quad \forall q \in Q$$

ossia si fa l'ipotesi che la funzione peso dipenda unicamente dalla componente continua dello spazio di stato e non dal modo q considerato. Per valutare l'effetto sulla soluzione ottenuta mediante l'HALP della funzione $\bar{\psi}(\cdot)$ si considereranno due scelte possibili:

- 1) **Distribuzione normale:** La funzione peso è il prodotto di n_s funzioni gaussiane indipendenti:

$$\bar{\psi}(\mathbf{x}) = \prod_{i=1}^{n_s} \psi(x_i) = \prod_{i=1}^{n_s} N(x_i; \mu_i, \sigma_i),$$

dove $N(\cdot | \mu_i, \sigma_i)$ indica la densità di probabilità di una variabile casuale gaussiana come media μ_i e varianza σ_i .

- 2) **Distribuzione uniforme:** La funzione peso è il prodotto di n_s distribuzioni uniformi indipendenti sull'intervallo $[y_L, y_U]$. La funzione peso sarà quindi data da:

$$\bar{\psi}(\mathbf{x}) = \prod_{i=1}^{n_s} \psi(x_i) = \prod_{i=1}^{n_s} U_{[y_L, y_U]}(x_i) = U_{[y_L, y_U]}^{n_s}(\mathbf{x})$$

In entrambi i casi, l'ipotesi che sia possibile definire la funzione peso come prodotto di n_s funzioni, ognuna dipendente da una sola componente scalare dello spazio di stato, permette di utilizzare le espressioni in forma analitica presentate nel capitolo precedente per il calcolo degli integrali che appaiono nel problema HALP.

Per la scelta delle funzioni di base è opportuno fare una considerazione preliminare sulle caratteristiche della value function ottima $V^*(\cdot)$ di cui si vuole costruire un'approssimazione. $V^*(\cdot)$ è la value function associata alla politica ottima μ^* ed è quindi data dalla seguente espressione:

$$V^*(s) = V_{\mu^*}(\cdot)(s) = E_{\mu^*} \left[\sum_{k=0}^{\infty} \gamma^k \rho(s_k, s_{k+1}) \middle| s_0 = s \right]$$

Da questa espressione è facile capire che $V^*(s) \leq 0, \forall s \in A_s$ e $V^*(s) \geq 0, \forall s \notin A_s$. Questo perché se $s_0 = s \in A_s$ e si rimane in A_s indefinitamente allora:

$$\sum_{k=0}^{\infty} \gamma^k \rho(s_k, s_{k+1}) = 0,$$

mentre se $s_0 = s \in A_s$ e si esce da A_s ad un certo istante n :

$$\sum_{k=0}^{\infty} \gamma^k \rho(s_k, s_{k+1}) \leq -\gamma^n + \gamma^{n+1} < 0$$

perché $0 < \gamma < 1$. Viceversa se $s_0 = s \notin A_s$ e si rimane fuori da A_s , allora:

$$\sum_{k=0}^{\infty} \gamma^k \rho(s_k, s_{k+1}) = 0,$$

mentre se si rientra in A_s all'istante di tempo n :

$$\sum_{k=0}^{\infty} \gamma^k \rho(s_k, s_{k+1}) \leq \gamma^n - \gamma^{n+1} > 0.$$

Dato che se la temperatura \mathbf{x} delle stanze è vicina alla frontiera di A_s la probabilità che si verifichi una transizione che porti ad oltrepassare la frontiera è maggiore rispetto al caso in cui \mathbf{x} è lontano dalla frontiera, è prevedibile che anche l'andamento della value function

ottima presenti una discontinuità sulla frontiera dell'insieme A_s . Per questo motivo, le funzioni di base $f_{iq}:x \rightarrow \mathbb{R}$ scelte hanno come supporto A_s oppure il suo complementare $\bar{A}_s = X \setminus A_s$. Esse possono essere ottenute moltiplicando una funzione definita su X per la funzione indicatrice di A_s o \bar{A}_s . Indicheremo nel seguito con $f_{iq}^{A_s}(\cdot)$ e $f_{iq}^{\bar{A}_s}(\cdot)$ le due tipologie di funzioni, e con $n_q^{A_s}$ e $n_q^{\bar{A}_s}$ il numero di funzioni $f_{iq}^{A_s}(\cdot)$ e $f_{iq}^{\bar{A}_s}(\cdot)$ considerate. Ciò porta alla seguente espressione per la funzione approssimante:

$$V^w(q, x) = \sum_{i=0}^{n_q^{A_s}} w_{iq}^{A_s} f_{iq}^{A_s}(x) + \sum_{i=0}^{n_q^{\bar{A}_s}} w_{iq}^{\bar{A}_s} f_{iq}^{\bar{A}_s}(x), \quad (q, x) \in S. \quad (4.8)$$

Sulla base della (4.8) il problema di ALP può essere riformulato come:

$$\begin{aligned} \min_{w_{iq}} \quad & \sum_{q \in Q} \left(\sum_{i=0}^{n_q^{A_s}} \int_{A_s} \bar{\psi}(x) w_{iq}^{A_s} f_{iq}^{A_s}(x) dx + \sum_{i=0}^{n_q^{\bar{A}_s}} \int_{\bar{A}_s} \bar{\psi}(x) w_{iq}^{\bar{A}_s} f_{iq}^{\bar{A}_s}(x) dx \right) \\ \text{soggetto a:} \quad & \sum_{i=0}^{n_q^{A_s}} w_{iq}^{A_s} f_{iq}^{A_s}(x) + \sum_{i=0}^{n_q^{\bar{A}_s}} w_{iq}^{\bar{A}_s} f_{iq}^{\bar{A}_s}(x) \geq R(s, u) + \\ & \gamma \sum_{q' \in Q} \left(\sum_{i=0}^{n_q^{A_s}} T_q(q'|q, u) \int_{A_s} T_x(dx'|(q', x)) w_{iq'}^{A_s} f_{iq'}^{A_s}(x') dx + \right. \\ & \left. \sum_{i=0}^{n_q^{\bar{A}_s}} T_q(q'|q, u) \int_{\bar{A}_s} T_x(dx'|(q', x)) w_{iq'}^{\bar{A}_s} f_{iq'}^{\bar{A}_s}(x') dx \right), \quad \forall s = (q, x) \in S, u \in U \end{aligned} \quad (4.9)$$

dove la funzione $R(s, u) = \int_S \rho(s, s') T_s(ds'|s, u)$ sarà data da:

$$R((q, x), u) = \sum_{q' \in Q} T_q(q'|q, u) \int_X \rho((q, x), (q', x')) T_x(dx'|(q', x)) dx'$$

che, in base alla (4.7), può essere sviluppata come:

$$R((q, x), u) = \begin{cases} - \sum_{q' \in Q} T_q(q'|q, u) \int_{A_s} T_x(dx'|(q', x)) dx', & x \in A_s \\ \sum_{q' \in Q} T_q(q'|q, u) \int_{\bar{A}_s} T_x(dx'|(q', x)) dx', & x \notin A_s \end{cases} \quad (4.10)$$

Nel seguito si supporrà che $f_{i,q}^{A_s}(\cdot) = f_{i,q}^{\bar{A}_s}(\cdot)$ e $f_{i,q}^{\bar{A}_s}(\cdot) = f_{i,q}^{A_s}(\cdot)$, $\forall q \in Q$, cioè che le funzioni di base adottate non dipendono dal modo $q \in Q$, e quindi $n_q^{A_s} = n^{A_s}$, $\forall q \in Q$ e $n_q^{\bar{A}_s} = n^{\bar{A}_s}$, $\forall q \in Q$.

Affinché il problema (4.9) sia feasible assumiamo:

$$\begin{aligned} f_0^{A_s}(\cdot) &= 1_{A_s}(\cdot) \\ f_0^{\bar{A}_s}(\cdot) &= 1_{\bar{A}_s}(\cdot) \end{aligned}$$

dove $1_{A_s}(\cdot)$ è la funzione indicatrice di A_s e $1_{\bar{A}_s}(\cdot)$ è la funzione indicatrice di \bar{A}_s .

La versione a scenario del problema (4.9) può essere riscritta in maniera compatta nel seguente modo:

$$\begin{aligned} \min_{\mathbf{w}} \quad & S \mathbf{w} \\ \text{soggetto a:} \quad & F \mathbf{w} \geq R + E \mathbf{w} \end{aligned} \quad (4.11)$$

dove il vettore dei parametri \mathbf{w} è così organizzato:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_{A_s} \\ \mathbf{w}_{\bar{A}_s} \end{bmatrix}$$

Le componenti di \mathbf{w} sono date da:

$$\begin{aligned} \mathbf{w}_0 &= [w_{01, \dots}, w_{0(n_s+1)}, w_{01, \dots}, w_{0(n_s+1)}]^T \in \mathbb{R}^{2(n_s+1)} \\ \mathbf{w}_{A_s} &= [w_{11, \dots}, w_{n^{A_s}1}, \dots, w_{1(n_s+1)}, \dots, w_{(n^{A_s})(n_s+1)}]^T \in \mathbb{R}^{n^{A_s}(n_s+1)} \\ \mathbf{w}_{\bar{A}_s} &= [w_{11, \dots}, w_{n^{\bar{A}_s}1}, \dots, w_{1(n_s+1)}, \dots, w_{(n^{\bar{A}_s})(n_s+1)}]^T \in \mathbb{R}^{n^{\bar{A}_s}(n_s+1)} \end{aligned}$$

Nella (4.11) S è un vettore riga di dimensione:

$$n_w = \sum_{q \in Q} (n_q^{A_s} + n_q^{\bar{A}_s} + 2) = \sum_{q=1}^{n_s+1} n_q^{A_s} + \sum_{q=0}^{n_s+1} n_q^{\bar{A}_s} + 2(n_s + 1) = (n_s + 1)(n^{A_s} + n^{\bar{A}_s} + 2),$$

il cui generico elemento S_i è dato da:

$$S_i = \int_{A_s} \bar{\psi}(x) f_i^{A_s}(x) dx \quad \text{o} \quad S_i = \int_{\bar{A}_s} \bar{\psi}(x) f_i^{\bar{A}_s}(x) dx$$

I vincoli della (4.11) sono espressi mediante le matrici F , R e E che avranno un numero di righe pari a $n = N |Q| |U| = N \cdot (n_s + 1) \cdot (n_s + 1)$, dove N rappresenta il numero di campioni della componente continua dello spazio di stato estratti in accordo all'approccio a scenario (si veda la Sezione 3.4.4 del Capitolo 3). Nello specifico, per ogni riga delle matrici si fissa il campione x estratto, il modo q e l'ingresso u in accordo alla seguente logica:

- 1) Fisso il campione x estratto;
- 2) Per ogni campione estratto considero tutti i modi $q \in Q$;
- 3) Per ogni modo $q \in Q$ considero tutti gli ingressi $u \in U$;

in tal modo per ogni campione x estratto verranno introdotte un numero di righe pari a $|Q| \cdot |U|$ in ciascuna delle quali saranno fissati il valore di x , il modo q e l'ingresso u .

Il codice MATLAB per la costruzione delle matrici F , R ed E è riportato in appendice.

4.3 RISULTATI NEL CASO DI UNA SIGNOLA STANZA

Consideriamo come primo esempio numerico quello in cui la stanza sia soltanto una e in essa sia presente un elemento riscaldante. La stanza scambierà calore unicamente con l'esterno secondo l'equazione:

$$x_{1,k+1} = x_{1,k} + \Delta t b_1 (x_a - x_{1,k}) + \Delta t c_1 h_{1,k} + \eta_{1,k} \quad (4.12)$$

dove il passo di integrazione è $\Delta t = 1/30$, la temperatura ambiente $x_a = 6$, il coefficiente di

scambio termico con l'esterno $b_1=0.25$, il calore fornito dall'elemento riscaldante è $c_1=12$ e la varianza del processo bianco gaussiano $\eta_{1,k}(\cdot)$ è pari a Δt ($\sigma^2=1$). Il fattore di discount γ è pari a 0.95 e $\alpha=0.8$.

L'insieme safe $A_s=1$ è definito come $A_s=Q \times A$, con $A=[17.5,22]$.

Essendo il numero di stanze $n_s=1$, lo spazio di stato continuo sarà \mathbb{R} e ci saranno due modi:

- $q=1$, in cui la stanza è riscaldata;
- $q=2$, in cui la stanza non è riscaldata.

Anche la variabile di controllo u potrà assumere due valori:

- $u=1$, l'elemento riscaldante viene acceso;
- $u=2$, l'elemento riscaldante viene spento.

Per le funzioni peso sullo stato si propongono due scelte alternative: distribuzione gaussiana e distribuzione uniforme.

Distribuzione gaussiana: $\bar{\psi}(x)=N(x;\mu,\sigma)$ con $\mu=\frac{17.5+22}{2}$ e $\sigma=5$. La varianza σ viene scelta in modo che l'intervallo dei valori di x in cui si vuole ottenere un'approssimazione più accurata della value function sia compreso nell'intervallo di ampiezza $\pm 2\sigma$. Nell'esempio in esame, l'intervallo in cui si focalizza l'attenzione è $[13.5,26]$ e si sceglie $\sigma=5$.

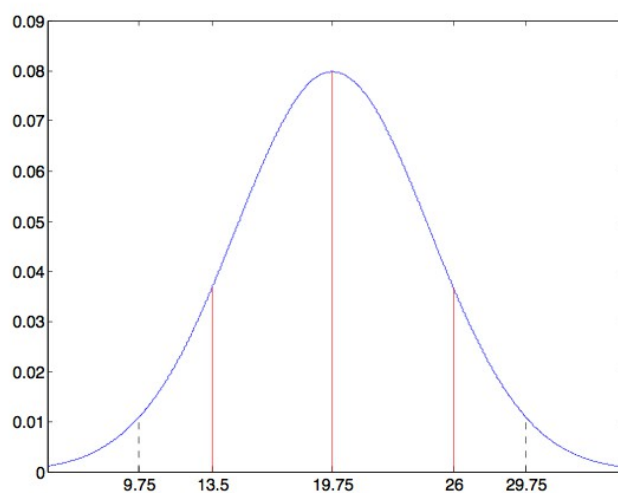


Figura 4.2: Funzione peso gaussiana.

Distribuzione uniforme: $\bar{\psi}(x) = U_{[13.5, 26]}(x)$

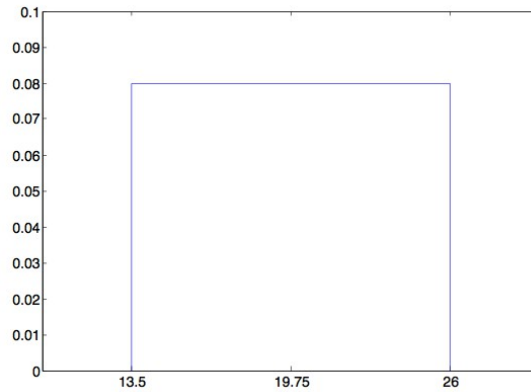


Figura 4.3: Funzione peso uniforme.

Il numero N di valori della temperatura da estrarre in accordo alla distribuzione di probabilità $\bar{\psi}(x)$ è stato scelto fissando come valore del parametro di violazione $\epsilon = 0.01$ e come valore del parametro di confidenza $\beta = 10^{-5}$.

4.3.1 FUNZIONI DI BASE GAUSSIANE

Come prima alternativa per la scelta delle funzioni di base mediante cui costruire l'approssimante della value function ottima, si adottano funzioni gaussiane troncate su A e \bar{A} . Avremo quindi:

$$f_i^A(x) = \frac{1}{\sqrt{(2\pi)\sigma_i}} e^{-\frac{x-\mu_i}{2\sigma_i^2}} \cdot 1_A(x) \quad f_i^{\bar{A}}(x) = \frac{1}{\sqrt{(2\pi)\sigma_i}} e^{-\frac{x-\mu_i}{2\sigma_i^2}} \cdot 1_{\bar{A}}(x)$$

oltre alle funzioni $f_0^A(\cdot) = 1_A(\cdot)$ e $f_0^{\bar{A}}(\cdot) = 1_{\bar{A}}(\cdot)$. Il problema di ottimizzazione permette di determinare i pesi w_{iq} relativi ad ogni funzione di base, ma non la loro media e varianza.

Questi valori andranno quindi fissati a priori. Quello che ci aspettiamo è che la V^w calcolata risolvendo l'HALP (4.9) soddisfi:

$$V^w(s) \geq V^*(s), \forall s \in S,$$

e che sia tanto migliore quanti più sono le funzioni gaussiane adottate.

Nel caso in cui si scelgano poche funzioni di base per ogni modo i valori di media e varianza saranno determinanti per la qualità dell'approssimazione.

Supponiamo di voler costruire un'approssimante del tipo:

$$V^w(q, x) = 1_A(x) \{w_{0q}^A + w_{1q}^A N(x|\mu_{1^A}, \sigma_{1^A})\} + 1_{\bar{A}} \{w_{0q}^{\bar{A}} + w_{1q}^{\bar{A}} N(x|\mu_{1^{\bar{A}}}, \sigma_{1^{\bar{A}}}) + w_{2q}^{\bar{A}} N(x|\mu_{2^{\bar{A}}}, \sigma_{2^{\bar{A}}})\}, \forall q \in Q \quad (4.13)$$

con una gaussiana approssimante la value function all'interno dell'insieme safe e due gaussiane approssimanti la value function all'esterno di esso più la funzione costante. Mostriamo ora come una scelta casuale e “sfortunata” dei valori di media e varianza delle funzioni di base gaussiane generi una cattiva approssimazione. A tale scopo confronteremo la soluzione $V^w(\cdot)$ dell'HALP a scenario con la value function approssimata $\hat{V}(\cdot)$ ottenuta con il metodo della value iteration approssimata (Sezione 3.3, Capitolo 3) avendo scelto $\beta = 10^{-5}$ e una grigliatura dell'insieme safe $A = [17.5, 22]$ ottenuta dividendo tale insieme in $m = 100$ intervalli. L'insieme dello spazio in cui si è effettuata l'approssimazione è tuttavia più ampio dell'insieme A e pari a $[13.5, 26]$.

Nella seguente tabella sono riportati i valori di media e varianza scelti:

Modo		μ		σ	
1	$x \notin A$	17	21	2	3
	$x \in A$	14		0,6	
2	$x \notin A$	18	19	4	2
	$x \in A$	16		1,2	

Tabella 4.1: Valori di media e varianza scelti per l'esempio in Figura 4.4.

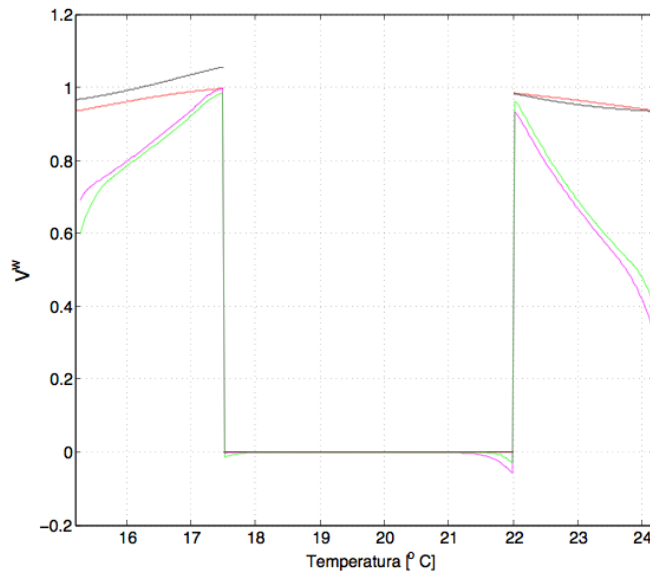


Figura 4.4: Confronto tra value function ottenuta per grigliatura (magenta modo1, verde modo 2) e value function ottenuta mediante HALP (rosso modo 1, nero modo 2) nel caso di scelta causale della media e della varianza delle funzioni approssimanti.

Mostriamo ora cosa accade nel caso di una scelta “opportuna” della media e della varianza delle funzioni gaussiane. Avendo a disposizione i valori della value function $\hat{V}(s), s \in S$ calcolata mediante il metodo della grigliatura, si è effettuato un procedimento di fitting di una funzione del tipo espresso nella (4.13) su tali valori minimizzando la somma degli errori quadratici $|\hat{V}(s) - V^*(s)|^2$ per mezzo della funzione Matlab *fmincon* che ha inoltre permesso di inserire il vincolo $V^w(s) \geq V^*(s), x \in [13.5, 26]$. I valori così ottenuti sono poi stati impostati come media e varianza delle funzioni di base nel problema di ottimizzazione HALP. Nella figura seguente si evince come una scelta opportuna dei valori di media e varianza permetterebbe di ottenere un'approssimazione migliore rispetto al caso precedente di scelta “cieca”. Si nota inoltre una differenza tra quanto ottenuto mediante il procedimento di fitting e la soluzione del problema HALP. Ciò è legato al fatto che nel procedimento di programmazione lineare approssimata l'ottimizzazione dei pesi delle funzioni di base è effettuata contemporaneamente per le funzioni $V^w(q, x), q \in Q$ mentre la procedura di fitting è stata effettuata in modo indipendente per i due modi del sistema.

Modo		μ		σ	
1	$x \notin A$	14,5	19,79	1,89	2,93
	$x \in A$	22,2		0,58	
2	$x \notin A$	14,7	19,7	1,5	2,55
	$x \in A$	23		0,42	

Tabella 4.2: Valori di media e varianza delle funzioni di base ottenute mediante fitting.

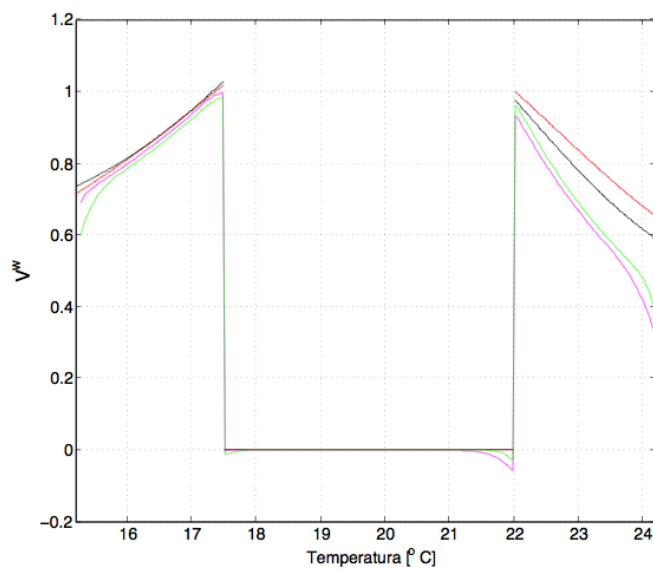


Figura 4.5: Confronto tra value function ottenuta per grigliatura (magenta modo 1, verde modo 2) e value function ottenuta mediante HALP (rosso modo 1, nero modo 2) nel caso si scelgano valori di media e varianza delle funzioni approssimanti pari a quelli ottenuti mediante fitting.

Utilizziamo ora un numero elevato di funzioni di base gaussiane le cui medie sono scelte secondo una griglia uniforme sullo spazio di stato in cui si desidera ottenere una buona approssimazione e con una varianza tale da garantire una sovrapposizione tra gaussiane adiacenti in modo che il vincolo $V^w(q, x) \geq V^*(q, x), x \in [13.5, 26]$ possa essere rispettato anche nelle zone dello spazio di stato comprese tra due gaussiane consecutive. Nella Figura 4.6 viene rappresentata l'approssimazione $V^w(\cdot)$ ottenuta.

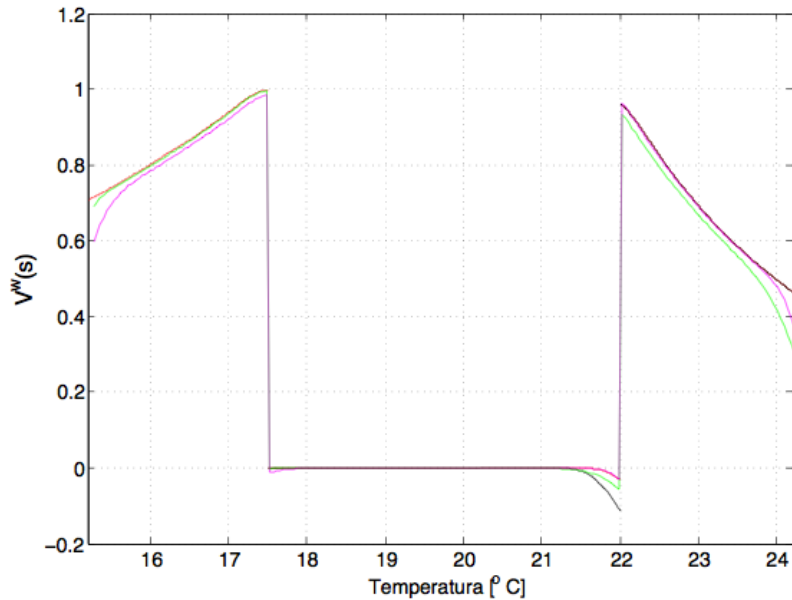


Figura 4.6: Confronto tra value function ottenuta per grigliatura (magenta modo 1, verde modo 2) e value function ottenuta mediante HALP (rosso modo 1, nero modo 2) utilizzando un numero elevato di funzioni di base.

In questo caso si sono utilizzate 20 gaussiane per approssimare la value function all'esterno dell'insieme safe e 5 all'interno di esso per ogni modo. Le gaussiane esterne presentano medie che distano $0.5\text{ }^{\circ}\text{C}$ e una varianza pari a 0.3. La Figura 4.7 chiarisce come la scelta della varianza delle gaussiane sia effettuata in modo da garantire margine di sovrapposizione tra le code.

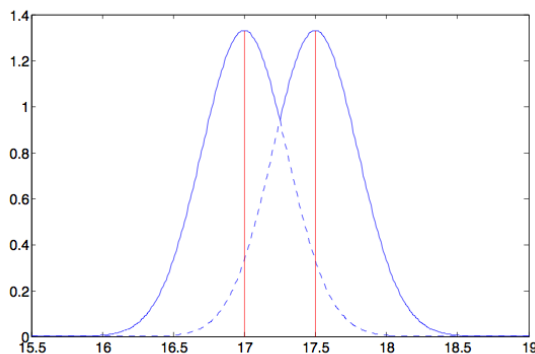


Figura 4.7: Scelta di media e varianza nel caso di approssimazione mediante un numero elevato di funzioni di base gaussiane.

È chiaro che utilizzare un elevato numero di funzioni di base permette di ottenere un'approssimazione molto buona della $V^*(s)$ a discapito dell'efficienza computazionale. Un maggiore numero di funzioni di base, infatti, implica un numero maggiore di variabili di ottimizzazione n_w e conseguentemente un aumento delle dimensioni delle matrici nella (4.11), le cui righe sono determinate dal numero N di campioni da estrarre. Nell'approccio a scenario N in effetti cresce linearmente con n_w . Negli esempi precedenti infatti, per l'approssimazione in Figura 4.5 si ha $n_w=12$ e $N=3269$, mentre per l'approssimazione in Figura 4.8 $n_w=54$ e $N=9104$. Utilizzare questo procedimento di fatto presenta gli stessi svantaggi del metodo della grigliatura: la qualità dell'approssimazione migliora all'infittirsi del numero di gaussiane che coprono S ma il costo computazionale per la soluzione del problema aumenta. Di seguito sono riportati i tempi di calcolo degli esempi precedenti divisi tra quelli richiesti per il calcolo delle matrici F , R ed E del problema (4.11) e quelli richiesti per la soluzione del problema di programmazione lineare. Il codice per il calcolo delle matrici è stato scritto in ambiente Matlab e il toolbox utilizzato per la risoluzione del problema LP è CPLEX mediante interfaccia Matlab. Il calcolatore usato è un Intel Xeon E5-2630 con 64 GB di RAM.

Esempio	n_w	N	Tempo Calcolo Matrici	Tempo Soluzione HALP
2 Gaussiane esterne 1 Gaussiana interna	12	3269	14,21 s	0,0952 s
20 Gaussiane esterne 5 Gaussiane interne	54	9104	261 s	0,501s

Tabella 4.3: Confronto tra i tempi di calcolo per gli esempi in Figura 4.5 e Figura 4.6.

Negli esempi precedenti, la funzione peso sullo stato adottata era anch'essa gaussiana. In particolare:

$$\bar{\psi}(\cdot) = N\left(\cdot; \frac{17.5+22}{2}, 5\right)$$

Nel caso in cui si adotti come una funzione peso la distribuzione uniforme:

$$\bar{\psi}(\cdot) = U_{[13.5,26]}(\cdot),$$

non si riscontrano differenze significative nelle soluzioni ottenute. I grafici seguenti mostrano la soluzione del problema di HALP ottenuta scegliendo una sola funzioni di base per ogni modo oltre a quella costante per costruire l'approssimante all'interno dell'inseme A ed una sola funzione di base oltre a quella costante per costruire l'approssimante all'esterno dell'insieme safe nel caso in cui si adotti una funzione peso gaussiana e in quello in cui si adotti una funzione peso uniforme. Per le funzioni di base all'interno dell'insieme safe si è scelto $\mu=21, \sigma=0.5$ per entrambi i modi mentre per la funzioni di base all'esterno dell'insieme A si è posto $\mu=(17.5+22)/2, \sigma=3.5$. Non si notano differenze significative nei due casi di funzione $\bar{\psi}(\cdot)$ gaussiana (Fig.4.8) e $\bar{\psi}(\cdot)$ uniforme (Fig. 4.9).

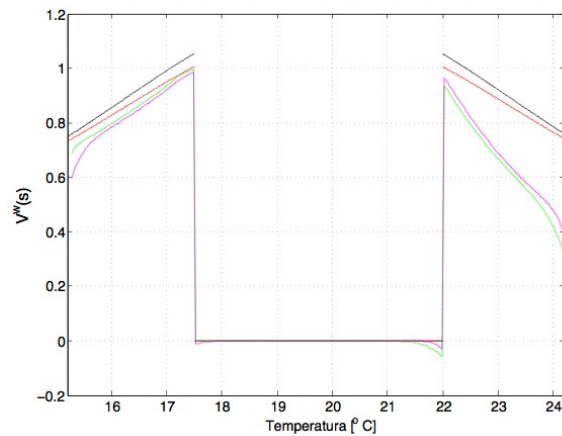


Figura 4.8: Value function approssimata ottenuta con HALP utilizzando una funzione peso gaussiana.

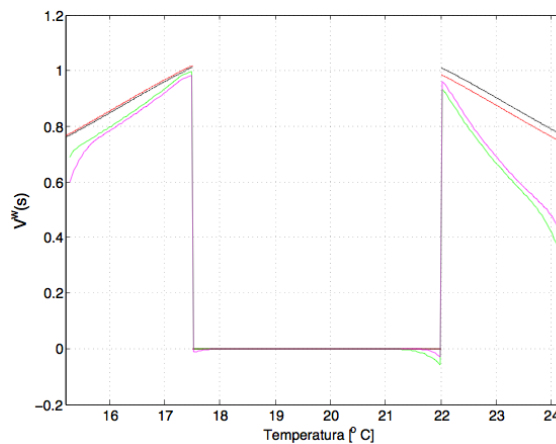


Figura 4.9: Value function approssimata ottenuta con HALP utilizzando una funzione peso uniforme.

4.3.2 FUNZIONI DI BASE POLINOMIALI

Analizziamo ora il caso in cui si scelgano funzioni di base polinomiali. Ciascuna funzione sarà una potenza di x con supporto limitato ad A o \bar{A} :

$$f_i^A(x) = x^n 1_A(x), \quad f_i^{\bar{A}} = x^n 1_{\bar{A}}(x), \quad n \in \mathbb{N}^+$$

con $f_0^A(\cdot) = 1_A(\cdot)$ e $f_0^{\bar{A}}(\cdot) = 1_{\bar{A}}(\cdot)$. La scelta del numero di funzioni da utilizzare equivale a decidere il grado del polinomio in x con cui approssimare la value function ottima.

Rispetto a quanto osservato per le funzioni di base gaussiane, la soluzione del problema HALP nel caso si scelgano funzioni di base polinomiali è influenzata maggiormente dalla scelta della funzione peso sullo stato. La soluzione è più accurata quando si utilizza una funzione peso che sia una distribuzione uniforme sull'intervallo di interesse [13.5,26]. Ciò è dovuto al fatto che le funzioni polinomiali di base su \bar{A} divergono per $|x| \rightarrow \infty$ e di conseguenza l'errore di approssimazione risulta elevato se non si limitano le estrazioni ad una regione limitata. Nelle figure 4.10 e 4.11 si mostra il risultato ottenuto quando:

$$V^w(q, x) = 1_A(x) \{w_{0q}^A + w_{1q}^A x + w_{2q}^A x^2\} + 1_{\bar{A}} \{w_{0q}^{\bar{A}} + w_{1q}^{\bar{A}} x + w_{2q}^{\bar{A}} x^2\}, \quad \forall q \in Q \quad (4.14)$$

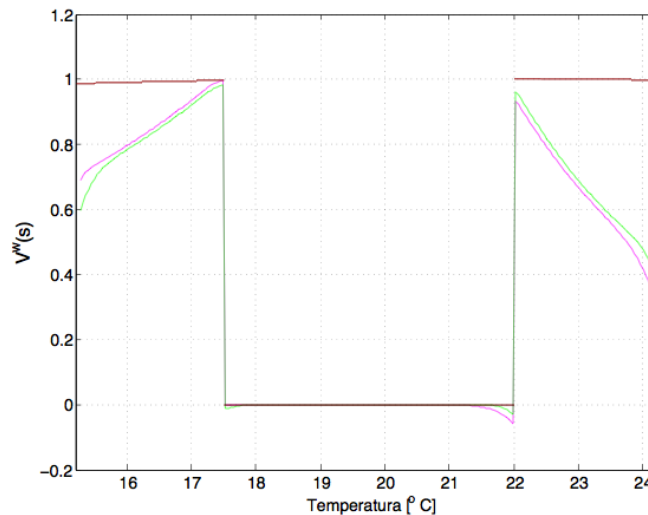


Figura 4.10: Value function $V^w(\cdot)$ ottenuta con HALP utilizzando una funzione peso gaussiana.

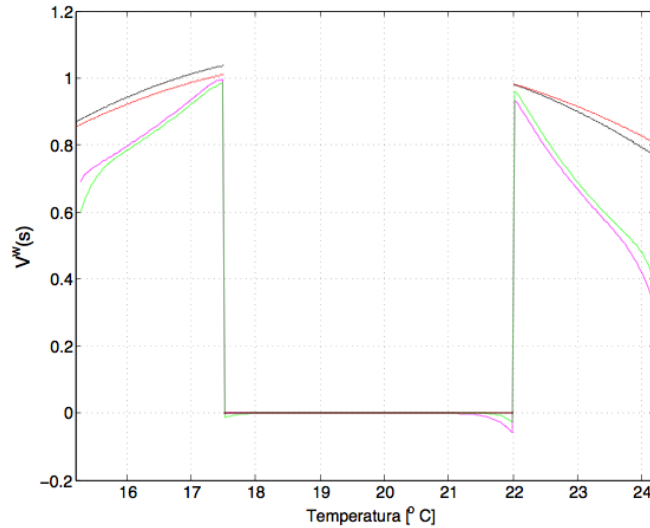


Figura 4.11: Value function $V^w(\cdot)$ ottenuta con HALP utilizzando una funzione peso uniforme.

Nel seguito quindi, si adotterà come funzione peso sullo stato la distribuzione uniforme su $[13.5, 26]$.

L'utilizzo di funzioni di base polinomiali ha il vantaggio, rispetto all'utilizzo di quelle gaussiane, che non occorre fissare opportunamente i valori di parametri che influenzano direttamente la qualità dell'approssimazione (vedi media e varianza delle funzioni di base nel caso esse siano gaussiane) ma soltanto il grado del polinomio approssimante. Più elevato sarà il grado del polinomio, maggiore sarà l'accuratezza della soluzione. Riportiamo ora i risultati in due casi in cui si scelgono come funzioni approssimanti della value function all'esterno dell'insieme A i polinomi in x di grado 3 (Figura 4.12), e 5 (Figura 4.13) rispettivamente. Aumentando il grado del polinomio si riesce ad ottenere una qualità migliore dell'approssimazione.

L'utilizzo delle funzioni polinomiali, rispetto al caso di funzioni gaussiane, ha un costo computazionale maggiore a causa dell'espressione ricorsiva presentata nel Capitolo 3 per il calcolo degli integrali coinvolti nel problema di programmazione lineare approssimata.

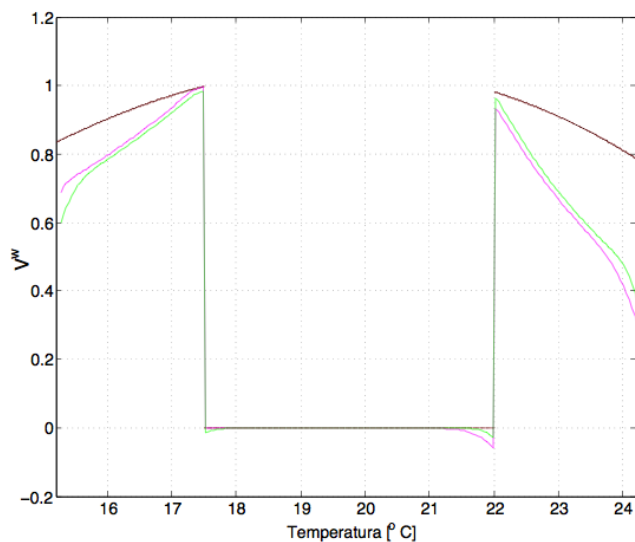


Figura 4.12: Value function $V^w(\cdot)$ ottenuta con HALP utilizzando un polinomio di terzo grado come approssimante all'esterno dell'insieme A .

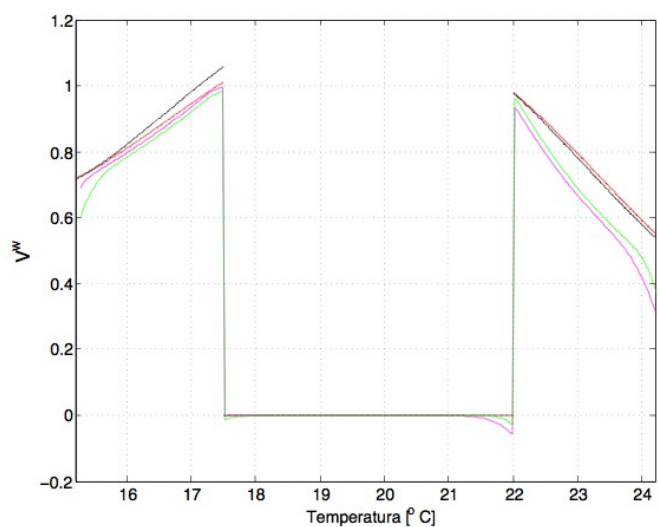


Figura 4.13: Value function $V^w(\cdot)$ ottenuta con HALP utilizzando un polinomio di quinto grado come approssimante all'esterno dell'insieme A .

Nella Tabella 4.4 si riportano dei confronti relativi ai tempi di calcolo per la soluzione del problema suddivisi tra tempi necessari al calcolo delle matrici della (4.11) e tempi di risoluzione del problema di programmazione lineare mediante CPLEX. Dal confronto si evince che l'uso di funzioni di base polinomiali rallenta il calcolo delle matrici (a causa dell'espressione ricorsiva utilizzata per il calcolo degli integrali). In entrambi i casi, sia quello gaussiano che quello polinomiale, i tempi di calcolo delle matrici sono determinati dalla scrittura del codice (numerose cicli for che rallentano l'esecuzione) e in questo lavoro non si è trattato il problema di migliorarne l'efficienza.

Nella sezione successiva, in cui si tratterà il caso in cui le stanze riscaldate siano due e quindi il carico computazionale è ancor maggiore a causa dell'aumento della dimensione dello spazio di stato, si adotteranno funzioni approssimate di tipo gaussiano per non appesantire ulteriormente l'algoritmo e allungare quindi i tempi di risoluzione.

Esempio	n_w	N	Tempo Calcolo Matrici	Tempo Soluzione HALP
3 Gaussiane esterne 3 Gaussiane interne	16	3893	21,91 s	0,0923 s
Polinomio 5° per A Polinomio 1° per \dot{A}	16	3893	261 s	0,143 s

Tabella 4.4: Confronto tra i tempi di calcolo nel caso di funzioni di base gaussiane e nel caso di funzioni di base polinomiali.

4.4 RISULTATI NEL CASO DI DUE STANZE ADIACENTI

Nel caso in cui le stanze riscaldate siano due, ossia $n_s=2$, le equazioni del modello del sistema sono le seguenti:

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + a_{12} \Delta t (x_{2,k} - x_{1,k}) + \Delta t b_1 (x_a - x_{1,k}) + \Delta t c_1 h_{1,k} + \eta_{1,k} \\ x_{2,k+1} &= x_{2,k} + a_{21} \Delta t (x_{1,k} - x_{2,k}) + \Delta t b_2 (x_a - x_{2,k}) + \Delta t c_2 h_{2,k} + \eta_{2,k} \end{aligned} \quad (4.15)$$

dove x_1 e x_2 rappresentano le temperature nella stanza 1 e nella stanza 2 rispettivamente, il passo di integrazione è $\Delta t=1/30$, la temperatura ambiente $x_a=6$, il coefficiente di scambio termico con l'esterno è $b_1=0.25$ per la stanza 1 e $b_2=0.25$ per la stanza 2, il calore fornito dall'elemento riscaldante di ogni stanza è $c_1=12$ e $c_2=14$ per le stanze 1 e 2 rispettivamente, $\eta_{1,k}$ e $\eta_{2,k}$ sono processi gaussiani bianchi indipendenti con media nulla e varianza Δt . Il fattore di discount γ è pari a 0.95 e $\alpha=0.8$. La matrice $A_{scambio}$ è la seguente:

$$A_{scambio} = \begin{bmatrix} 0 & 0.33 \\ 0.33 & 0 \end{bmatrix}$$

L'insieme safe A_s è definito come $A_s = Q \times A$ con:

$$A = [17.5, 22]^2$$

Essendo il numero di stanze $n_s=2$, lo spazio di stato continuo sarà \mathbb{R}^2 e i modi saranno tre:

- $q=1$, in cui la stanza 1 è riscaldata;
- $q=2$, in cui la stanza 2 è riscaldata;
- $q=3$, nessuna stanza è riscaldata.

Anche la variabile di controllo u potrà assumere tre valori:

- $u=1$, l'elemento riscaldante della stanza 1 viene acceso;
- $u=2$, l'elemento riscaldante della stanza 2 viene acceso;
- $u=3$, tutti gli elementi riscaldanti vengono spenti.

In questo esempio numerico si affronta il problema HALP scegliendo una funzione peso sullo stato gaussiana. Nello specifico $\bar{\psi}(\cdot)$ sarà del tipo:

$$\bar{\psi}(\mathbf{x}) = \prod_{i=1}^{n_s} \psi(x_i) = \prod_{i=1}^{n_s} N(x_i; \mu_i, \sigma_i),$$

e quindi, dato che il numero di stanze n_s è pari a due:

$$\bar{\psi}(\mathbf{x}) = N(x_1; \mu_1, \sigma_1) N_2(x_2; \mu_2, \sigma_2)$$

dove le medie μ_i sono centrate sul punto medio dell'intervallo [17.5,22] e le varianze sono pari a 5 come nel caso di una sola stanza.

Anche come funzioni di base si adotteranno funzioni gaussiane. La value function approssimata $V^w(\cdot)$ sarà la combinazione lineare di prodotti di due funzioni gaussiane monodimensionali, le funzioni di base avranno quindi l'espressione:

$$\begin{aligned} f_i^A(x) &= N(x_1; \mu_{i1^A}, \sigma_{i1^A}) N(x_2; \mu_{i2^A}, \sigma_{i2^A}) \cdot 1_A(x) \\ f_i^{\bar{A}}(x) &= N(x_1; \mu_{i1^{\bar{A}}}, \sigma_{i1^{\bar{A}}}) N(x_2; \mu_{i2^{\bar{A}}}, \sigma_{i2^{\bar{A}}}) \cdot 1_{\bar{A}}(x) \end{aligned} \quad (4.16)$$

con $f_0^A(\cdot) = 1_A(\cdot)$ e $f_0^{\bar{A}}(\cdot) = 1_{\bar{A}}(\cdot)$.

Riguardo ai parametri per l'approccio a scenario si è scelto, come nella sezione precedente, $\epsilon = 0.01$ e $\beta = 10^{-5}$. Le N estrazioni sono state effettuate in accordo alla distribuzione di probabilità $\bar{\psi}(\cdot)$.

Nelle figure 4.14 e 4.15 sono riportati i risultati ottenuti mediante il metodo della value iteration approssimata con grigliatura dello spazio di stato continuo in [13]. In particolare la Figura 4.14 riporta l'andamento della value function approssimata $\hat{V}(\cdot)$ (modo 1) e la Figura 4.15 la politica ottima (per i 3 modi) da essa ricavata.

Quanto osservato per il caso monodimensionale, si riscontra anche nel caso in cui la dimensione dello spazio di stato aumenti. La scelta dei valori di media e varianza delle funzioni di base ha un impatto determinante sulla qualità dell'approssimazione.

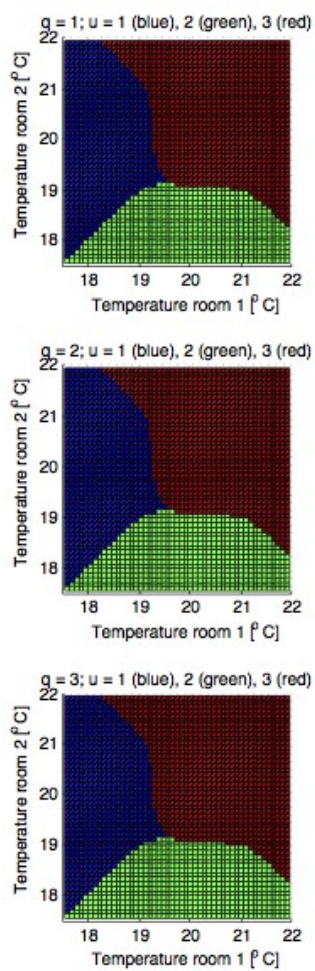


Figura 4.14: Politica ottima ottenuta mediante il metodo della value iteration approssimata con grigliatura della componente continua dello spazio di stato.

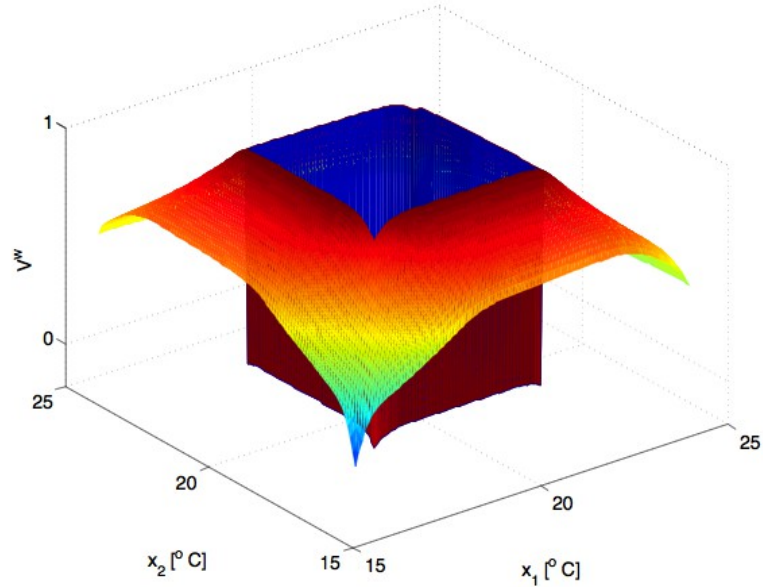


Figura 4.15: Value function $\hat{V}(\cdot)$ approssimata (modo 1) ottenuta mediante il metodo della value iteration approssimata con grigliatura della componente continua dello spazio di stato.

Mostriamo cosa accade alla politica ricavata sulla base della value function approssimata nel caso in cui ci siano due stanze riscaldate. In particolare scegliamo una sola funzione di base del tipo nella (4.16) per costruire l'approssimante sia all'interno che all'esterno dell'insieme A oltre alle funzioni costanti:

$$V^w(q, x) = 1_A(x) \{w_{0q}^A + w_{1q}^A N(x_1; \mu_{11^A}, \sigma_{11^A}) N(x_2; \mu_{12^A}, \sigma_{12^A})\} + 1_{\bar{A}}(x) \{w_{0q}^{\bar{A}} + w_{1q}^{\bar{A}} N(x_1; \mu_{11^{\bar{A}}}, \sigma_{11^{\bar{A}}}) N(x_2; \mu_{12^{\bar{A}}}, \sigma_{12^{\bar{A}}})\}, \forall q \in Q \quad (4.17)$$

Fissiamo le medie delle gaussiane univariate nel punto medio dell'intervallo $[17.5, 22]$. Effettuiamo ora due scelte differenti per i valori delle varianze. Nella Figura 4.16 si è posto $\sigma_{11^A} = \sigma_{12^A} = 10$ e $\sigma_{11^{\bar{A}}} = \sigma_{12^{\bar{A}}} = 1$ per ciascuno dei tre modi. Nella Figura 4.17 si è posto invece $\sigma_{11^A} = \sigma_{12^A} = 15$ e $\sigma_{11^{\bar{A}}} = \sigma_{12^{\bar{A}}} = 1$, $\forall q \in Q$. In quest'ultimo caso si riscontra un'approssimazione migliore in quanto la politica ottenuta si avvicina maggiormente a quella ottenuta per grigliatura della Figura 4.15. Dunque, una scelta casuale dei parametri delle funzioni di base

può portare a risultati più o meno buoni senza che però si abbia un controllo sulla qualità della soluzione ottenuta.

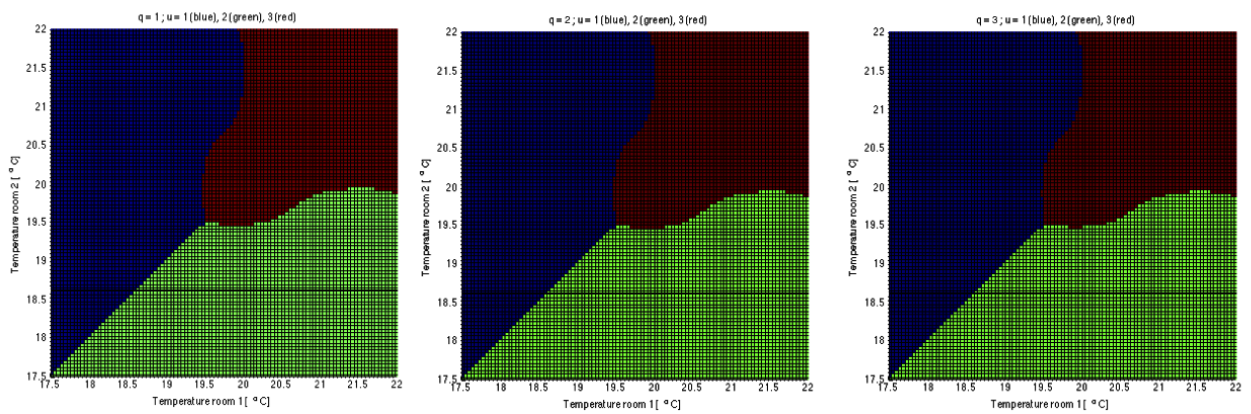


Figura 4.16: Politica ottima ottenuta mediante il metodo della HALP a scenario nel caso $\sigma_{11}^i = \sigma_{12}^i = 10$ e

$$\sigma_{11}^i = \sigma_{12}^i = 1 .$$

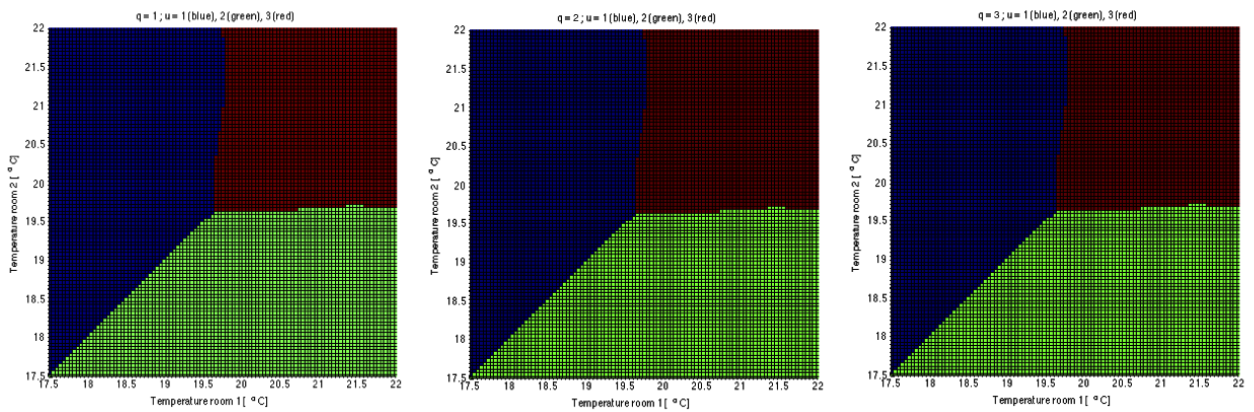


Figura 4.17: Politica ottima ottenuta mediante il metodo della HALP a scenario nel caso $\sigma_{11}^i = \sigma_{12}^i = 15$ e

$$\sigma_{11}^i = \sigma_{12}^i = 1 .$$

Una strada percorribile è quella già introdotta nel caso monodimensionale: adottare un numero elevato di funzioni di base in modo da sfruttare le proprietà di approssimazione locale delle funzioni gaussiane e migliorare la qualità dell'approssimazione ottenuta. In particolare, per il caso in esame, scegliamo delle funzioni di base del tipo (4.16) in cui le medie siano centrate sui vertici di una griglia uniforme di dimensioni $n \times n$ sull'insieme $[13.5, 26]^2$. Le varianze sono scelte secondo l'euristica ([3],[5]):

$$\sigma_i^2 = (|Dom(x_i)| / (n-1))^2,$$

in cui $Dom(x_i)$ rappresenta l'insieme dei valori di interesse della variabile x_i . Nello specifico $|Dom(x_1)| = |Dom(x_2)| = (26 - 13.5)$. La figura seguente esemplifica il caso in cui $n = 3$.

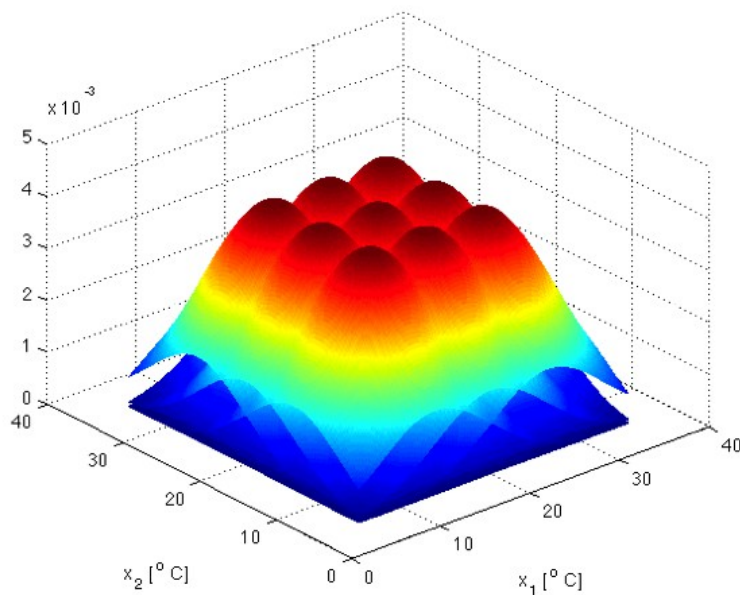


Figura 4.18: Posizione delle gaussiane approssimanti nel caso di una griglia 3×3 .

Riportiamo ora il caso in cui si aumenti progressivamente il valore di n e si ottenga quindi una grigliatura più fitta. Si nota che all'aumentare del valore di n la qualità dell'approssimazione migliora e la value function ottenuta approssima sempre meglio la $\hat{V}(\cdot)$ riportata in Figura 4.15.

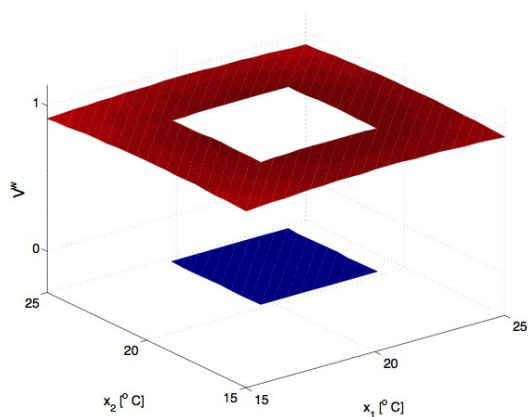


Figura 4.19: Value function approssimata $V^w(\cdot)$ (modo 1) ottenuta con una griglia di funzioni di base 3×3 .

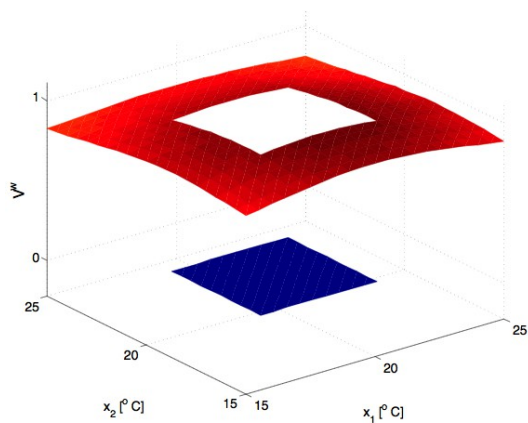


Figura 4.20: Value function approssimata $V^w(\cdot)$ (modo 1) ottenuta con una griglia di funzioni di base 4×4 .

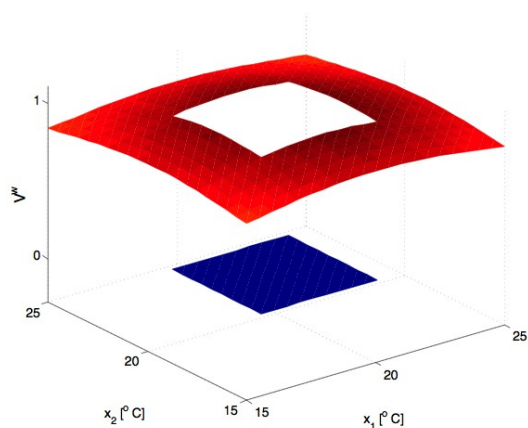


Figura 4.21: Value function approssimata $V^w(\cdot)$ (modo 1) ottenuta con una griglia di funzioni di base 5×5 .

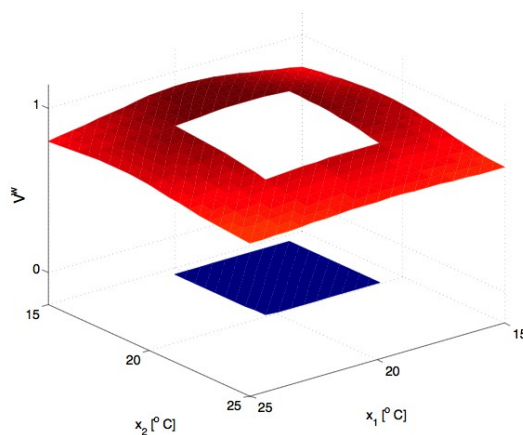


Figura 4.22: Value function approssimata $V^w(\cdot)$ (modo 1) ottenuta con una griglia di funzioni di base 6×6 .

Come per il caso monodimensionale, l'utilizzo di un numero elevato di funzioni di base ha lo svantaggio di richiedere un maggior costo computazionale. Ancora una volta l'aspetto determinante è il tempo richiesto per il calcolo degli elementi delle matrici che compaiono nel problema di programmazione lineare. Riportiamo i tempi necessari alla soluzione ottenuta in Figura 4.19 (griglia di funzioni di base 3×3) e in Figura 4.22 (griglia di funzioni di base 6×6).

Esempio	n_w	N	Tempo Calcolo Matrici	Tempo Soluzione HALP
Griglia 3x3	33	6327	3015,84 s	2,14 s
Griglia 6x6	114	16512	11321,78 s	5,16 s

Tabella 4.5: Tempi di calcolo per le soluzioni in Figura 4.19 e in Figura 4.21.

4.5 RISULTATI NEL CASO DI PIÙ DI DUE STANZE

Il codice Matlab utilizzato per la risoluzione degli esempi nella sezione precedente è stato esteso al caso di un numero generico n_s di stanze. Con l'aumentare delle dimensioni dello spazio di stato continuo non è più possibile ottenere una rappresentazione grafica della value function approssimata ottenuta. Qui ci limitiamo ad analizzare i tempi di calcolo necessari per la soluzione del problema, senza entrare nel merito della valutazione della qualità dell'approssimazione.

Le equazioni che descrivono la dinamica della componente continua dello stato saranno del tipo (4.1) con n_s fissato.

Le funzioni di base scelte sono del tipo:

$$f_i^A(x) = 1_A(x) \cdot \prod_{i=1}^{n_s} N(x_i; \mu_{i^A}, \sigma_{i^A})$$

$$f_i^{\bar{A}}(x) = 1_{\bar{A}}(x) \cdot \prod_{i=1}^{n_s} N(x_i; \mu_{i^{\bar{A}}}, \sigma_{i^{\bar{A}}})$$

e la funzione approssimante:

$$V^w(q, x) = 1_A(x) \{w_{0q}^A + w_{1q}^A f_i^A(x)\} + 1_{\bar{A}}(x) \{w_{0q}^{\bar{A}} + w_{1q}^{\bar{A}} f_i^{\bar{A}}(x)\}, \quad \forall q \in Q$$

Le medie delle singole gaussiane sono centrate sul punto medio dell'intervallo [17.5,22] e le varianze fissate a 15 .

La funzione peso adottata ha la forma seguente:

$$\bar{\psi}(x) = \prod_{i=1}^{n_s} \psi(x_i) = \prod_{i=1}^{n_s} N(x_i; \mu_i, \sigma_i),$$

dove le medie μ_i sono centrate sul punto medio dell'intervallo [17.5,22] e le varianze sono pari a 5 . Per l'approccio a scenario, si è scelto $\epsilon = 0.01, \beta = 10^{-5}$ e le estrazioni sono state effettuate in accordo alla $\bar{\psi}(\cdot)$.

Il fattore di discount γ è pari a 0.95.

- **3 Stanze:** Supponiamo che la matrice $A_{scambio}$ abbia la seguente espressione:

$$A_{scambio} = \begin{bmatrix} 0 & 0.33 & 0.33 \\ 0.33 & 0 & 0 \\ 0.33 & 0 & 0 \end{bmatrix}$$

e che $b_1=b_2=b_3=0.25$, $c_1=12, c_2=14, c_3=13$, $x_a=6$ e $\alpha=0.8$. I processi gaussiani bianchi indipendenti $\eta_{1,k}, \eta_{2,k}, \eta_{3,k}$ hanno media nulla e varianza Δt , con $\Delta t=1/30$.

In questo caso i modi saranno 4 dato che $n_s=3$. La tabella seguente riporta i tempi di calcolo suddivisi tra quelli necessari al calcolo delle matrici e quelli necessari per la risoluzione del problema di programmazione lineare.

Nr. Stanze	n_w	N	Tempo Calcolo Matrici	Tempo Soluzione HALP
3	16	3893	3396,92 s	0,277 s

Tabella 4.6: Tempi di calcolo nel caso in cui $n_s=3$.

Possiamo osservare come il tempo di calcolo per la soluzione del problema di programmazione lineare sia maggiore a quello riscontrato nella sezione precedente e riportato in Tabella 4.4 nonostante il numero di variabili di ottimizzazione e di campioni estratti sia lo stesso. Ciò è dovuto al fatto che in questo caso la cardinalità di Q e di U è più elevata e quindi il numero di vincoli maggiore.

- **4 Stanze :**Supponiamo che le matrici $A_{scambio}$ abbia la seguente espressione:

$$A_{scambio} = \begin{bmatrix} 0 & 0.33 & 0 & 0.33 \\ 0.33 & 0 & 0.33 & 0 \\ 0 & 0.33 & 0 & 0 \\ 0.33 & 0 & 0 & 0 \end{bmatrix}$$

e che $b_1=b_2=b_3=b_4=0.25$, $c_1=12, c_2=14, c_3=13, c_4=13$, $x_a=6$ e $\alpha=0.8$. I processi gaussiani bianchi indipendenti $\eta_{1,k}, \eta_{2,k}, \eta_{3,k}, \eta_{4,k}$ hanno media nulla e varianza Δt , con $\Delta t=1/30$. In questo caso i modi saranno 5 dato che $n_s=4$. La tabella seguente riporta i tempi di calcolo suddivisi tra quelli necessari al calcolo delle matrici e quelli

necessari per la risoluzione del problema di programmazione lineare.

Nr. Stanze	n_w	N	Tempo Calcolo Matrici	Tempo Soluzione HALP
4	20	4491	7345,12 s	0,341 s

Tabella 4.7: Tempi di calcolo nel caso in cui $n_s=4$.

5

CONCLUSIONI E SVILUPPI FUTURI

In questa tesi si è proposto un approccio alla soluzione del problema del controllo ottimo di sistemi ibridi stocastici basato sulla programmazione lineare approssimata ibrida (HALP). Questo approccio utilizza in modo sinergico concetti quali l'approssimazione funzionale e la programmazione lineare per ricondurre il problema del calcolo della value function ottima – soluzione dell'equazione di Bellman che caratterizza il problema di controllo ottimo – ad un problema di programmazione lineare (LP) semi-infinito in cui viene calcolato il migliore approssimante della value function ottima in una certa classe funzionale parametrizzazione linearmente. L'idea sviluppata è stata quella di ricorrere al cosiddetto “approccio a scenario” per ridurre il numero dei vincoli del problema LP semi-infinito a finito, e potere quindi trovare una soluzione attraverso strumenti molto efficienti quali Cplex, fornendo al tempo stesso garanzie probabilistiche sulle proprietà di generalizzazione della soluzione in termini di soddisfacimento anche dei vincoli non considerati per il suo calcolo.

Per valutare la qualità dei risultati ottenibili mediante l'utilizzo combinato del metodo HALP e dell'approccio a scenario si è affrontato un caso di studio e confrontata la soluzione “HALP a scenario” con quella ottenuta applicando un metodo alternativo già noto in letteratura, basato sulla grigliatura della componente continua dello stato.

Lo studio ha evidenziato come la qualità delle soluzioni HALP a scenario sia fortemente condizionata dalla classe funzionale scelta per approssimare la value function ottima. Questo

fatto non è sorprendente e rappresenta un problema ben noto nell'ambito dell'approssimazione funzionale, in cui sono però almeno direttamente disponibili i campioni della funzione da approssimare. Nel nostro contesto, il problema è più complicato perché la value function ottima non è nota nemmeno in modo campionario, e la classe funzionale da adottare deve essere parametrizzata linearmente per ottenere un problema di tipo LP.

Una possibilità che si potrebbe esplorare in studi futuri è quella di utilizzare la logica alla base degli algoritmi a *gradiente residuo* (e cioè la minimizzazione del cosiddetto residuo dell'equazione di Bellman ([10])) per scegliere in successione le funzioni da combinare linearmente nella soluzione HALP.

Un altro aspetto, più di natura computazionale, evidenziato nel caso di studio è che i tempi di calcolo della soluzione HALP a scenario sono determinanti dalla valutazione dei valori attesi coinvolti nella formulazione del problema LP e non dalla sua soluzione mediante Cplex.

Il numero di integrali da calcolare cresce con il numero di vincoli considerati, con il numero dei modi del sistema, con la cardinalità dello spazio di controllo, e con il numero di parametri dell'approssimazione funzionale. Il codice Matlab scritto potrebbe essere ovviamente migliorato, per esempio riducendo il numero di cicli for e optando per una soluzione basata sulla vettorizzazione. Un'idea che appare però più interessante e promettente è quella di investigare una soluzione basata sul "GPU computing", dove i molteplici core della unità di elaborazione grafica vengono utilizzati per eseguire il calcolo in parallelo di questi molteplici integrali.

A

APPENDICE

A.1 CODICE MATLAB PER N_s STANZE CON FUNZIONI DI BASE E FUNZIONE PESO GAUSSIANE

```

clear all
close all

%-----
%  INIZIALIZZAZIONE
%-----
% Parametri orizzonte temporale

dt = 1/30;          % Passo di integrazione
sd = 1*sqrt(dt);   % Deviazione standard del rumore sulla temperatura (°C)

%Numero di stanze:
ns=1; % valore di default: 1 stanza

% Parametri dell'equazione della dinamica di temperatura
%  $x_i(k+1) = x_i(k) + a_{\{i,j\}}*dt*(x_j(k) - x_i(k)) - b_i*dt*(x_i(k)-x_a) +$ 
%  $c_i*dt*(q==i) + n_i(k)$ 
xa    = 6;          % Temperatura esterna  $T_e$  (°C)
b_i=0.25*dt*ones(ns,1); %  $0.1*dt$ ; % Coefficiente di scambio tra  $T_i$  e  $T_e$ 
normalizzato rispetto alla capacità termica ( $b_i*dt =$  adimensionale)
c_i=13*dt*ones(ns,1);
c_i(1:2,1)=dt*[12 14]; % Calore fornito in ingresso alla stanza i
(normalizzato rispetto alla capacità termica) per unità di tempo
A=crea_a(ns);        % Matrice contenente i coefficienti di scambio termico
tra le stanze:  $a_{\{i,j\}}=a_{\{j,i\}}$  coefficiente di scambio tra stanza i e
stanza j, 0 se non c'è scambio.  $a_{\{i,i\}}=0$ . La matrice è simmetrica con
diagonale composta da elementi nulli.

% Estremi dell'insieme safe
xm_S = 17.5;        % Estremo inferiore del safe set (°C)
xM_S = 22;          % Estremo superiore del safe set (°C)

```

```

xmedio=(xM_S+xm_S)/2;% Media dell'intervallo (°C)

nq = ns+1; % Numero stati discreti (q è lo stato dell'elemento riscaldante;
q = r (ON nella stanza r), ns +1 (OFF in tutte le stanze)
nu = ns+1; % Numero ingressi (u = r (accendi nella stanza r e spegni le
altre), ns+1 (spegni tutto))
nx = ns; % Numero di variabili di stato continue

% Funzione di transizione stato discreto: P(q'|x,q,u)
% Nel nostro caso P(q'|x,q,u) = P(q'|q,u) (non dipende da x)
% La definiamo attraverso una matrice Tq(q,q',u)
alfa = 0.8;
Tq = crea_tq(ns);

%-----
%HALP
%-----

%discount factor
gamma=0.95;

%Numero di funzioni approssimanti all'interno del safe set per ogni modo:
ni=ones(nq,1);%valore di default 1 approssimante per ogni modo

%Numero di funzioni approssimanti all'esterno del safe set per ogni modo:
ne=ones(nq,1);%valore di default 1 approssimante per ogni modo

%Psi: peso sullo stato nella funzione da minimizzare:
%Prodotto tra nx funzioni gaussiane di x_j (temperatura nella stanza j)
%psi=PROD_j(N(x_j,media_psi_x_j,sigma_psi_x_j))
sigma_psi=6*eye(nx); %valore di default 6
media_psi=xmedio*ones(1,nx); %valore di default xmedio

%Media e farianza delle funzioni approssimanti (gaussiane) f_j,q,e(x) e
f_j,q,i(x) ad ogni colonna corrisponde
%una variabile di stato e ad ogni riga le medie o le varianze delle due
gaussiane il cui prodotto costituisce
%f_j,q,e(x) o f_j,q,i(x)

medi=cell(nq,1);
mede=cell(nq,1);
sigi=cell(nq,1);
sige=cell(nq,1);

%APPROSSIMANTI INTERNE
%Valori di default media: xmedio; sigma: 1
for i=1:nq
    medi{i}=xmedio*ones(ni(i),nx); %matrice delle medie delle gaussiane per
il modo i interno del SS
    sigi{i}=1*ones(ni(i),nx); %matrice delle varianze delle gaussiane per
il modo i interno del SS
end

%APPROSSIMANTI ESTERNE
%Valori di default media: xmedio; sigma: 4
for i=1:nq
    mede{i}=xmedio*ones(ne(i),nx); %matrice delle medie delle gaussiane per
il modo i esterno del SS
    sige{i}=4*ones(ne(i),nx); %matrice delle varianze delle gaussiane per
il modo i esterno del SS

```

```

end

%-----
%Problema LP
%-----

%Il problema può essere espresso in forma matriciale come come: min S*w
subject to F*w>= R +E*w

%-----
%Funzione da minimizzare nel problema LP:S*w
%-----

%Funzione da minimizzare S*w:

XL=xm_S*ones(1,nx); %Estremi inferiori dell'insieme safe
XU=xM_S*ones(1,nx); %Estremi superiori dell'insieme safe

%Integrale di w_0,q,i*psi su x interno al safe set (ha lo stesso valore per
q=1,q=2 e q=3):

inti=mvncdf(XL,XU,media_psi,sigma_psi);

%Integrale di w_0,q,e*psi su x esterno al safe set (ha lo stesso valore per
q=1,q=2 e q=3):
inte=1-inti;

s0i=inti*ones(1,nq);
s0e=inte*ones(1,nq);

intpsi_fi=[];
int_f_i=cell(nq,1);
for i=1:nq
    ni_q=ni(i);
    medie_f=medi{i};
    sig_f=sigi{i};
    int=zeros(1,ni_q);
    for j=1:ni_q
        pr=zeros(nx,3);
        for k=1:nx
            [mu,sig,c]=prod_gauss(media_psi(1,k),sigma_psi(k,k),medie_f(j,k),sig_f(j,k));
            pr(k,1)=mu;
            pr(k,2)=sig;
            pr(k,3)=c;
        end
        med=pr(:,1)';
        sig=diag(pr(:,2));
        int(1,j)=prod(pr(:,3))*mvncdf(XL,XU,med,sig);
    end
    int_f_i{i}=int;
    intpsi_fqi=int_f_i{i};
    intpsi_fi=[intpsi_fi,intpsi_fqi];
end

intpsi_fe=[];
int_f_e=cell(nq,1);
for i=1:nq
    ne_q=ne(i);
    mede_f=mede{i};
    sig_f=sige{i};
    int=zeros(1,ne_q);

```

```

    for j=1:ne_g
        pr=zeros(nx,3);
        for k=1:nx
            [mu,sig,c]=prod_gauss(media_psi(1,k),sigma_psi(k,k),mede_f(j,k),sig_f(j,k));
            pr(k,1)=mu;
            pr(k,2)=sig;
            pr(k,3)=c;
        end
        med=pr(:,1)';
        sig=diag(pr(:,2));
        int(1,j)=prod(pr(:,3))*(1-mvncdf(XL,XU,med,sig));
    end
    int_f_e{i}=int;
    intpsi_fqe=int_f_e{i};
    intpsi_fe=[intpsi_fe,intpsi_fqe];
end

S=[s0i,s0e,intpsi_fi,intpsi_fe];

%-----
% Teoria a scenario
%-----

% Calcolo degli N campioni secondo la distribuzione gaussiana psi_i(o
psi_e) assegnato
% epsilon (parametro di violazione), beta(parametro di confidenza) e
d(numero di variabili di ottimizzazione).

d=nq*2+(sum(ni(:,1))+sum(ne(:,1)));
eps=0.01;
beta=10^(-5);

%Calcolo degli N campioni in accordo alla Teoria a scenario
N=find_N(eps,beta,d);

%Campionamento della variabile X(temperatura) secondo la distribuzine
gaussiana psi_i
X=zeros(N,nx);
for i=1:nx
    X(:,i)=media_psi(1,i) + sigma_psi(i,i)*randn(N,1);
end

%-----
% Vincoli del problema LP
%-----
% I vincoli vengono espressi in forma matriciale come:
% F*w>=R+E*w;
% Le matrici F ed E sono organizzate a blocchi di nu*nq*d ( e di dimensioni
(N*nu*nq)*d; per ogni blocco è fissato il valore di x estratto e ciascuna
riga è relativa ad una combinazione dei valori della coppia (q,u); in
particolare:
% Riga 1: q=1,u=1;
% Riga 2: q=1,u=2;
% ...
% Riga nu: q=1,u=nu;
% ...
% Riga nq*nu: q=nq,u=nu;
%
% Le d colonne sono relative alle d variabili di ottimizzazione;
% R è un vettore colonna (N*nu*nq)

```

```

%-----
%Calcolo della matrice F
%-----

f0i=zeros(N*nu*nq,nq);
k=1;
r=1;
for i=1:nq
    for j=1:N
        if (X(j,:)<=22)&(X(j,*)>=17.5)
            for m=1:nu
                f0i(k,r)=1;
                k=k+1;
            end
            k=k+(nq-1)*nu;
        else
            k=k+nq*nu;
        end
    end
    r=r+1;
    k=i*nu+1;
end

f0e=zeros(N*nu*nq,nq);
k=1;
r=1;
for i=1:nq
    for j=1:N
        if (X(j,:)<=22)&(X(j,*)>=17.5)
            k=k+nq*nu;
        else
            for m=1:nu
                f0e(k,r)=1;
                k=k+1;
            end
            k=k+(nq-1)*nu;
        end
    end
    r=r+1;
    k=i*nu+1;
end

r=1;
k=1;
fi=zeros(N*nu*nq,sum(ni(:,1)));
for i=1:nq
    mede_f=medi{i};
    sig_f=sigi{i};
    ni_q=ni(i);
    for j=1:ni_q
        k=(i-1)*nu+1;
        for m=1:N
            if (X(m,:)<=22)&(X(m,*)>=17.5)
                for s=1:nu
                    fi(k,r)=mvnpdf(X(m,:),mede_f(j,:),diag(sig_f(j,:)));
                    k=k+1;
                end
                k=k+(nq-1)*nu;
            else
                k=k+nq*nu;
            end
        end
    end
end

```

```

        end
    end
    r=r+1;
end
end
r=1;
k=1;
fe=zeros(N*nu*nq,sum(ne(:,1)));
for i=1:nq
    mede_f=mede{i};
    sig_f=sige{i};
    ne_q=ne(i);
    for j=1:ne_q
        k=(i-1)*nu+1;
        for m=1:N
            if (X(m,:)<=22)&(X(m,*)>=17.5)
                k=k+nq*nu;
            else
                for s=1:nu
                    fe(k,r)=mvnpdf(X(m,:),mede_f(j,:),diag(sig_f(j,:)));
                    k=k+1;
                end
                k=k+(nq-1)*nu;
            end
        end
    end
    r=r+1;
end
end
F=[f0i,f0e,fi,fe];
disp('Ultimato calcolo di F')

%-----
%Calcolo di R
%-----

R=zeros(N*nu*nq,1);
k=1;
for i=1:N
    xu=zeros(nx,1);
    x=zeros(nx,1);
    for j=1:nx
        xu(j)=X(i,j)+A(j,)*(-X(i,j)*(ones(nx,1))+X(i,:))'-
            b_i(j,1)*(X(i,j)-xa)+c_i(j,1);
        x(j)=X(i,j)+A(j,)*(-X(i,j)*(ones(nx,1))+X(i,:))'-b_i(j,1)*(X(i,j)-
            xa);
    end
    sigma=sd*eye(nx);
    media=zeros(1,nx);
    media(1,:)=x;
    if (X(i,*)<=22)&(X(i,*)>=17.5)
        for j=1:nq
            for m=1:nu
                if m==(nx+1)
                    media(1,m)=xu(m,1);
                end
                R(k,1)=-(1-mvncdf(XL,XU,media,sigma));
                k=k+1;
                media(1,:)=x;
            end
        end
    end
end

```

```

        end
    else
        for j=1:nq
            for m=1:nu
                if m~=(nx+1)
                    media(1,m)=xu(m,1);
                end
                R(k,1)=(mvncdf(XL,XU,media,sigma));
                k=k+1;
                media(1,:)=x;
            end
        end
    end
end

disp('Ultimato calcolo di R')

%-----
%Calcolo di E
%-----

e0i=zeros(N*nu*nq,nq);
r=1;
for i=1:nq
    k=1;
    for j=1:N
        xu=zeros(nx,1);
        x=zeros(nx,1);
        for m=1:nx
            xu(m)=X(j,m)+A(m,:)*(-X(j,m)*(ones(nx,1))+X(j,:))'-
                b_i(m,1)*(X(j,m)-xa)+c_i(m,1);
            x(m)=X(j,m)+A(m,:)*(-X(j,m)*(ones(nx,1))+X(j,:))'-
                b_i(m,1)*(X(j,m)-xa);
        end
        sigma=sd*eye(nx);
        media=zeros(1,nx);
        media(1,:)=x;
        for s=1:nq
            for t=1:nu
                if t~=(nx+1)
                    media(t)=xu(t);
                end
                e0i(k,r)=gamma*Tq(s,i,t)*mvncdf(XL,XU,media,sigma);
                k=k+1;
                media(1,:)=x;
            end
        end
    end
    r=r+1;
end

e0e=zeros(N*nu*nq,nq);
r=1;
for i=1:nq
    k=1;
    for j=1:N
        xu=zeros(nx,1);
        x=zeros(nx,1);
        for m=1:nx

```

```

        xu(m)=X(j,m)+A(m,:)*(-X(j,m)*(ones(nx,1))+X(j,:))'-
            b_i(m,1)*(X(j,m)-xa)+c_i(m,1);
        x(m)=X(j,m)+A(m,:)*(-X(j,m)*(ones(nx,1))+X(j,:))'-
            b_i(m,1)*(X(j,m)-xa);
    end
    sigma=sd*eye(nx);
    media=zeros(1,nx);
    media(1,:)=x;
    for s=1:nq
        for t=1:nu
            if t~=(nx+1)
                media(t)=xu(t);
            end
            e0e(k,r)=gamma*Tq(s,i,t)*(1-mvncdf(XL,XU,media,sigma));
            k=k+1;
            media(1,:)=x;
        end
    end
end
r=r+1;
end

ei=zeros(N*nu*nq,sum(ni(:,1)));
r=1;
for i=1:nq
    mede_f=medi{i};
    sig_f=sigi{i};
    ni_q=ni(i,1);
    for j=1:ni_q
        k=1;
        for m=1:N
            xu=zeros(nx,1);
            x=zeros(nx,1);
            pr=zeros(3,nx);
            pru=zeros(3,nx);
            for s=1:nx
                xu(s)=X(m,s)+A(s,:)*(-X(m,s)*(ones(nx,1))+X(m,:))'-
                    b_i(s,1)*(X(m,s)-xa)+c_i(s,1);
                x(s)=X(m,s)+A(s,:)*(-X(m,s)*(ones(nx,1))+X(m,:))'-
                    b_i(s,1)*(X(m,s)-xa);
                [mu sig c]=prod_gauss(x(s),sd,mede_f(j,s),sig_f(j,s));
                [muu sigu cu]=prod_gauss(xu(s),sd,mede_f(j,s),sig_f(j,s));
                pr(1,s)=mu;
                pr(2,s)=sig;
                pr(3,s)=c;
                pru(1,s)=muu;
                pru(2,s)=sigu;
                pru(3,s)=cu;
            end
            med=pr(1,:);
            sig=diag(pr(2,:));
            ci=pr(3,:);
            med_u=pru(1,:);
            sig_u=diag(pru(2,:));
            c_u=pru(3,:);
            for s=1:nq
                for t=1:nu
                    if t~=(nx+1)
                        med(1,t)=pru(1,t);
                        sig(t,t)=pru(2,t);
                        ci(1,t)=pru(3,t);
                    end
                end
            end
        end
    end
end

```

```

end
ei(k,r)=gamma*Tq(s,i,t)*prod(ci)*mvncdf(XL,XU,med,sig);
k=k+1;
ci=pr(3,:);
med=pr(1,:);
sig=diag(pr(2,:));
end
end
end
r=r+1;
end
end
end
ee=zeros(N*nu*nq,sum(ne(:,1)));
r=1;
for i=1:nq
mede_f=mede{i};
sig_f=sige{i};
ne_q=ne(i,1);
for j=1:ne_q
k=1;
for m=1:N
xu=zeros(nx,1);
x=zeros(nx,1);
pr=zeros(3,nx);
pru=zeros(3,nx);
for s=1:nx
xu(s)=X(m,s)+A(s,:)*(-X(m,s)*(ones(nx,1))+X(m,:))'-
b_i(s,1)*(X(m,s)-xa)+c_i(s,1);
x(s)=X(m,s)+A(s,:)*(-X(m,s)*(ones(nx,1))+X(m,:))'-
b_i(s,1)*(X(m,s)-xa);
[mu sig c]=prod_gauss(x(s),sd,mede_f(j,s),sig_f(j,s));
[muu sigu cu]=prod_gauss(xu(s),sd,mede_f(j,s),sig_f(j,s));
pr(1,s)=mu;
pr(2,s)=sig;
pr(3,s)=c;
pru(1,s)=muu;
pru(2,s)=sigu;
pru(3,s)=cu;
end
med=pr(1,:);
sig=diag(pr(2,:));
ci=pr(3,:);
med_u=pru(1,:);
sig_u=diag(pru(2,:));
c_u=pru(3,:);
for s=1:nq
for t=1:nu
if t~=(nx+1)
med(1,t)=pru(1,t);
sig(t,t)=pru(2,t);
ci(1,t)=pru(3,t);
end
end
ee(k,r)=gamma*Tq(s,i,t)*prod(ci)*(1-
mvncdf(XL,XU,med,sig));
k=k+1;
ci=pr(3,:);
med=pr(1,:);
sig=diag(pr(2,:));
end
end
end
end

```

```

        end
        r=r+1;
    end
end
E=[e0i,e0e,ei,ee];
disp('Ultimato calcolo di E')

%-----
%Soluzione LP
%-----

w=cplexlp(S,(E-F),-R,[],[]);

```

A.2 CODICE MATLAB PER N_s STANZE CON FUNZIONI DI BASE POLINOMIALI E FUNZIONE PESO UNIFORME

```

clear all
close all

%-----
%  INIZIALIZZAZIONE
%-----
% Parametri orizzonte temporale

dt = 1/30;          % Passo di integrazione
sd = 1*sqrt(dt);  % Deviazione standard del rumore sulla temperatura (°C)

%Numero di stanze:
ns=1; % valore di default: 1 stanza

% Parametri dell'equazione della dinamica di temperatura
%  $x_i(k+1) = x_i(k) + a_{\{i,j\}}*dt*(x_j(k) - x_i(k)) - b_i*dt*(x_i(k)-x_a) +$ 
%  $c_i*dt*(q==i) + n_i(k)$ 
xa = 6;           % Temperatura esterna  $T_e$  (°C)
b_i=0.25*dt*ones(ns,1); % 0.1*dt; % Coefficiente di scambio tra  $T_i$  e  $T_e$ 
normalizzato rispetto alla capacità termica ( $b_i*dt =$  adimensionale)
c_i=13*dt*ones(ns,1);
c_i(1:2,1)=dt*[12 14]; % Calore fornito in ingresso alla stanza i
(normalizzato rispetto alla capacità termica) per unità di tempo
A=crea_a(ns);      % Matrice contenente i coefficienti di scambio termico
tra le stanze:  $a_{\{i,j\}}=a_{\{j,i\}}$  coefficiente di scambio tra stanza i e
stanza j, 0 se non c'è scambio.  $a_{\{i,i\}}=0$ . La matrice è simmetrica con
diagonale composta da elementi nulli.

% Estremi dell'insieme safe
xm_S = 17.5;      % Estremo inferiore del safe set (°C)
xM_S = 22;       % Estremo superiore del safe set (°C)
xmedio=(xm_S+xM_S)/2;% Media dell'intervallo (°C)

nq = ns+1; % Numero stati discreti (q è lo stato dell'elemento riscaldante;
q = r (ON nella stanza r), ns +1 (OFF in tutte le stanze))

```

```

nu = ns+1; % Numero ingressi (u = r (accendi nella stanza r e spegni le
altre), ns+1 (spegni tutto))
nx = ns; % Numero di variabili di stato continue

% Funzione di transizione stato discreto: P(q'|x,q,u)
% Nel nostro caso P(q'|x,q,u) = P(q'|q,u) (non dipende da x)
% La definiamo attraverso una matrice Tq(q,q',u)
alfa = 0.8;
Tq = crea_tq(ns);

%-----
%HALP
%-----

%discount factor
gamma=0.95;

%PSI: PESO SULLO STATO NELLE FUNZIONE DA MINIMIZZARE
%Prodotto tra nx funzioni uniformi di x_j (temperatura nella stanza j)
%sull'intervallo [a,b]
%Estremi della distribuzione UNIFORME
a=14;
b=26;

%GRADO MASSIMO
gmaxi=3; %grado massimo del polinomio approssimante all'interno del SS
gmaxe=3; %grado massimo del polinomio approssimante all'esterno del SS

%APPROSSIMANTI INTERNE
% Ci matrice dei polinomi di default: uso tutte le combinazioni in cui ho
il prodotto delle stesse potenze: es:
%x_1*x_1,x_1^2*x_2^2...

Ci=zeros(gmaxi,nx);
for i=1:nx
    Ci(:,i)=[1:1:gmaxi];
end
nci=gmaxi;

%APPROSSIMANTI ESTERNE
%Valori di default:
%Matrice dei polinomi di default: uso tutte le combinazioni in cui ho il
prodotto delle stesse potenze: es:
%x_1*x_1,x_1^2*x_2^2...

Ce=zeros(gmaxe,nx);
for i=1:nx
    Ce(:,i)=[1:1:gmaxe];
end
nce=gmaxe;

%-----
%Problema LP
%-----

%Il problema può essere espresso in forma matriciale come come: min s*w
subject to F*w>= R +E*w

```

```

%-----
%Funzione da minimizzare nel problema LP
%-----

%Integrale di  $w_{0,q,i} \psi$  su  $x$  interno al safe set (ha lo stesso valore per
 $q=1, q=2$  e  $q=3$ ):
XL=xM_S*ones(1,nx);
XU=xM_S*ones(1,nx);
inti=(xM_S-xm_S)/(b-a);

%Integrale di  $w_{0,q,e} \psi$  su  $x$  esterno al safe set (ha lo stesso valore per
 $q=1, q=2$  e  $q=3$ ):
inte=1-inti;

s0i=inti*ones(1,nq);
s0e=inte*ones(1,nq);

intpsi_fi=[];
for i=1:nq
    inti=zeros(1,nci);
    for j=1:nci
        pr=zeros(1,nx);
        for k=1:nx
            pr(1,k)=(1/(b-a))*(1/(Ci(j,k)+1))*(xM_S^(Ci(j,k)+1)
                -xm_S^(Ci(j,k)+1));
        end
        inti(1,j)=prod(pr);
    end
    intpsi_fi=[intpsi_fi,inti];
end

intpsi_fe=[];
for i=1:nq
    inte=zeros(1,nce);
    for j=1:nce
        pr=zeros(1,nx);
        for k=1:nx
            pr(1,k)=(1/(b-a))*(1/(Ce(j,k)+1))*(b^(Ce(j,k)+1)-a^(Ce(j,k)+1))
                -(1/(b-a))*(1/(Ce(j,k)+1))*(xM_S^(Ce(j,k)+1)-xm_S^(Ce(j,k)+1));
        end
        inte(1,j)=prod(pr);
    end
    intpsi_fe=[intpsi_fe,inte];
end

S=[s0i,s0e,intpsi_fi,intpsi_fe];

%-----
% Teoria a scenario
%-----

% Calcolo degli N campioni secondo la distribuzione uniforme  $\psi_i$  (o  $\psi_e$ )
assegnato
% epsilon (parametro di accuratezza), beta(parametro di confidenza) e
d(numenro di variabili di ottimizzazione).

d=nq*2+(nci+nce)*nq;
eps=0.01;
beta=10^(-5);

%Calcolo degli N campioni in accordo alla Teoria a scenario
N=find_N(eps,beta,d);

```

```

%Campionamento della variabile X(temperatura) secondo la distribuzine
gaussiana psi_i
X=zeros(N,nx);
for i=1:nx
    X(:,i)=(b-a)*rand(N,1)+a;
end

%-----
% Vincoli del problema LP
%-----
% I vincoli vengono espressi in forma matriciale come:
%  $F*w \geq R + E*w$ ;
% Le matrici F ed E sono organizzate a blocchi di  $nu*nq*d$  ( e di dimensioni
 $(N*nu*nq)*d$  ciascuna riga è relativa ad una
% combinazione dei valori della coppia (q,u); in particolare:
% Riga 1: q=1,u=1;
% Riga 2: q=1,u=2;
% ...
% Riga nu: q=1,u=nu;
% ...
% Riga nq*nu: q=nq,u=nu;
%
% Le d colonne sono relative alle d variabili di ottimizzazione;
% Per ciascun blocco  $nu*nqxd$  si considera lo stesso valore della q-pla X
% estratta
% R è un vettore colonna ( $N*nu*nq$ )

%-----
%Calcolo della matrice F
%-----

f0i=zeros(N*nu*nq,nq);
k=1;
r=1;
for i=1:nq
    for j=1:N
        if (X(j,:)<=22)&(X(j,*)>=17.5)
            for m=1:nu
                f0i(k,r)=1;
                k=k+1;
            end
            k=k+(nq-1)*nu;
        else
            k=k+nq*nu;
        end
    end
    r=r+1;
    k=i*nu+1;
end

f0e=zeros(N*nu*nq,nq);
k=1;
r=1;
for i=1:nq
    for j=1:N
        if (X(j,:)<=22)&(X(j,*)>=17.5)
            k=k+nq*nu;
        else
            for m=1:nu
                f0e(k,r)=1;
            end
        end
    end
end

```

```

                k=k+1;
            end
            k=k+(nq-1)*nu;
        end
        end
        r=r+1;
        k=i*nu+1;
    end

    r=1;
    k=1;
    fi=zeros(N*nu*nq,nci);
    for i=1:nq
        for j=1:nci
            k=(i-1)*nu+1;
            for m=1:N
                if (X(m,:)<=22)&(X(m,*)>=17.5)
                    for s=1:nu
                        pr=zeros(1,nx);
                        for t=1:nx
                            pr(t)=X(m,t)^Ci(j,t);
                        end
                        fi(k,r)=prod(pr);
                        k=k+1;
                    end
                    k=k+(nq-1)*nu;
                else
                    k=k+nq*nu;
                end
            end
            r=r+1;
        end
    end

    r=1;
    k=1;
    fe=zeros(N*nu*nq,nce);
    for i=1:nq
        for j=1:nce
            k=(i-1)*nu+1;
            for m=1:N
                if (X(m,:)<=22)&(X(m,*)>=17.5)
                    k=k+nq*nu;
                else
                    for s=1:nu
                        pr=zeros(1,nx);
                        for t=1:nx
                            pr(t)=X(m,t)^Ce(j,t);
                        end
                        fe(k,r)=prod(pr);
                        k=k+1;
                    end
                    k=k+(nq-1)*nu;
                end
            end
            r=r+1;
        end
    end

    F=[f0i,f0e,fi,fe];
    disp('Ultimato calcolo di F')

```

```

xu=zeros(N,nx);
x=zeros(N,nx);

for j=1:N
    for m=1:nx
        xu(j,m)=X(j,m)+A(m,:)*(-X(j,m)*(ones(nx,1))+X(j,:))'-
            b_i(m,1)*(X(j,m)-xa)+c_i(m,1);
        x(j,m)=X(j,m)+A(m,:)*(-X(j,m)*(ones(nx,1))+X(j,:))'-
            b_i(m,1)*(X(j,m)-xa);
    end
end

%-----
%Calcolo di R
%-----

R=zeros(N*nu*nq,1);
k=1;
for i=1:N
    sigma=sd*eye(nx);
    media=x(i,:);
    if (X(i,:)<=22)&(X(i,*)>=17.5)
        for j=1:nq
            for m=1:nu
                if m==(nx+1)
                    media(m)=xu(i,m);
                end
                R(k,1)=-(1-mvncdf(XL,XU,media,sigma));
                k=k+1;
                media=x(i,:);
            end
        end
    else
        for j=1:nq
            for m=1:nu
                if m==(nx+1)
                    media(m)=xu(i,m);
                end
                R(k,1)=(mvncdf(XL,XU,media,sigma));
                k=k+1;
                media=x(i,:);
            end
        end
    end
end
disp('Ultimato calcolo di R')

%-----
%Calcolo di E
%-----

e0i=zeros(N*nu*nq,nq);
r=1;
for i=1:nq
    k=1;
    for j=1:N
        sigma=sd*eye(nx);
        media=x(j,:);
        for s=1:nq
            for t=1:nu

```



```

        if t~=(nx+1)
            media(t)=xu(j,t);
        end
        e0i(k,r)=gamma*Tq(s,i,t)*mvncdf(XL,XU,media,sigma);
        k=k+1;
        media=x(j,:);
    end
end
end
r=r+1;
end

e0e=zeros(N*nu*nq,nq);
r=1;
for i=1:nq
    k=1;
    for j=1:N
        sigma=sd*eye(nx);
        media=x(j,:);
        for s=1:nq
            for t=1:nu
                if t~=(nx+1)
                    media(t)=xu(j,t);
                end
                e0e(k,r)=gamma*Tq(s,i,t)*(1-mvncdf(XL,XU,media,sigma));
                k=k+1;
                media=x(j,:);
            end
        end
    end
end
r=r+1;
end

ei=zeros(N*nu*nq,nci);
r=1;
for i=1:nq
    for j=1:nci
        k=1;
        for m=1:N
            for s=1:nq
                for t=1:nu
                    xcurr=x(m,:);
                    if t~=(nx+1)
                        xcurr(t)=xu(m,t);
                    end
                    pr=zeros(1,nx);
                    for z=1:nx
                        pr(1,z)=momenti(Ci(j,z),xcurr(z),sd,xm_S,xM_S);
                    end
                    ei(k,r)=gamma*Tq(s,i,t)*prod(pr);
                    k=k+1;
                end
            end
        end
    end
end
r=r+1;
end

ee=zeros(N*nu*nq,nce);
r=1;

```

```

for i=1:nq
  for j=1:nce
    k=1;
    for m=1:N
      for s=1:nq
        for t=1:nu
          xcurr=x(m,:);
          if t==(nx+1)
            xcurr(t)=xu(m,t);
          end
          pr=zeros(1,nx);
          for z=1:nx
            pr(1,z)=momenti(Ce(j,z),xcurr(z),sd,-inf,inf)-
              momenti(Ce(j,z),xcurr(z),sd,xM_S,xM_S);
          end
          ee(k,r)=gamma*Tq(s,i,t)*prod(pr);
          k=k+1;
        end
      end
    end
    r=r+1;
  end
end
toc
E=[e0i,e0e,ei,ee];
disp('Ultimato calcolo di E')

%-----
%Soluzione LP
%-----

w=cplexlp(S,[E-F],-R,[],[]);

```

BIBLIOGRAFIA

- [1] A.Abate, M.Prandini, J.Lygeros, S.Sastry. “ Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems”. *Automatica* 44(11): 2724-2734, 2008.
- [2] A.Fehnker and F.Ivančić. “Benchmarks for hybrid systems verifications”. *Hybrid Systems: Computation and Control, lecture notes in Computer Science 2993*: 326-341, 2004.
- [3] B.Kveton. “*Planning in Hybrid Structured Stochastic Domains*”. M.S., Comenius University, Slovak Republic, 2001.
- [4] C.Cassandras and John Lygeros. “Stochastic Hybrid Systems”. *CRC Press and IEEE Press, Control Engineering Series*, vol. 24, 2006
- [5] C.Guestrin, D.Koller, R.Parr, S.Venkataraman. “Efficient Solution Algorithms for Factored MDPs ”. *Journal of Artificial Intelligence Research* 19: 399-468, 2003.
- [6] D.P.Bertsekas. “Approximate Policy Iteration: a Survey and Some New Methods”. *Journal of Control Theory and Applications* 9.3: 310-335,2011.
- [7] D.P.De Farias and B.Van Roy. “On constraint sampling for the linear programming approach to approximate dynamic programming”. *Mathematics of Operations*

Research, 29(3):462–478, 2004.

- [8] H. Blom and John Lygeros. “Stochastic Hybrid Systems: Theory and Safety Critical Applications”. *Lecture Notes in Control and Information Sciences (LNCIS)*, vol. 337, 2006
- [9] J. Lygeros and M. Prandini. “Stochastic hybrid systems: a powerful framework for complex, large scale applications”. *European Journal of Control, introductory paper to the special issue on Stochastic hybrid systems*, vol. 16(6):583-594, 2010
- [10] L.Baird. “Residual Algorithms: Reinforcement Learning with Function Approximation”. *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.
- [11] M.C.Campi, S.Garatti, M.Prandini. “The Scenario Approach for Systems and Control Design” . *Annual Reviews in Control* 33(2): 149-157, 2009.
- [12] M.L.Puterman and S.L.Brumelle. “On the Convergence of Policy Iteration in Stationary Dynamic Programming”. *Mathematics of Operations Research*, vol. 4, no. 1, 1979.
- [13] M.Prandini and L.Piroddi. “A self-recovery approach to the probabilistic invariance problem for stochastic hybrid systems”. CDC 2012:2096-2101,2012.
- [14] R. Bellman. “*Dynamic Programming*”. Princeton, N.J: Princeton University, 1957.
- [15] R.S.Sutton and A.G.Barto. “*Reinforcement Learning: an Introduction*”. MIT Press, Cambridge, MA, 1998.