

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA  
CAMPUS DI CESENA

---

Scuola di Ingegneria  
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

VALUTAZIONE SPERIMENTALE  
DELL'ADATTAMENTO TRAMITE  
EMPOWERMENT IN ROBOT  
CONTROLLATI DA RETI BOOLEANE

*Elaborato in*  
SISTEMI INTELLIGENTI ROBOTICI

*Relatore*  
Prof. ANDREA ROLI

*Presentata da*  
LORENZO RIZZATO

*Co-relatore*  
Dott. MICHELE BRACCINI

---

Terza Sessione di Laurea  
Anno Accademico 2020 – 2021



## PAROLE CHIAVE

reti booleane

empowerment

plasticità fenotipica

apprendimento online

argos



*In opere esse minime  
magnopereque delectari*



# Indice

<b>Introduzione</b>	<b>xv</b>
<b>1 Stato dell'arte</b>	<b>1</b>
1.1 Reti booleane . . . . .	1
1.1.1 Definizioni . . . . .	2
1.1.2 Attrattori . . . . .	3
1.1.3 Reti critiche . . . . .	4
1.1.4 Adattamento online . . . . .	6
1.1.5 Plasticità fenotipica . . . . .	7
1.2 Empowerment . . . . .	7
1.2.1 Una prospettiva centrata sugli agenti . . . . .	8
1.2.2 Il problema della comunicazione . . . . .	10
1.2.3 Definizione di empowerment . . . . .	11
1.2.4 Applicazioni . . . . .	14
<b>2 Strumenti di simulazione</b>	<b>19</b>
2.1 Il simulatore ARGOS . . . . .	19
2.1.1 Configurazione . . . . .	20
2.2 Foot-bot . . . . .	21
2.2.1 Sensori . . . . .	21
2.2.2 Attuatori . . . . .	23

<b>3</b>	<b>Esperimenti</b>	<b>25</b>
3.1	Obiettivo . . . . .	25
3.2	Metodologia . . . . .	27
3.2.1	Fasi preliminari . . . . .	28
3.2.2	Esecuzione degli esperimenti . . . . .	28
3.2.3	Modalità di esecuzione . . . . .	29
3.2.4	Sensori e adattamento online . . . . .	29
3.2.5	Calcolo delle prestazioni . . . . .	30
3.2.6	Grafici e test statistici . . . . .	31
3.2.7	Descrizione dei risultati . . . . .	35
3.3	Tipi di task . . . . .	35
3.3.1	Fototassi assistita . . . . .	36
3.3.2	Obstacle avoidance . . . . .	38
3.3.3	Path following . . . . .	39
3.4	Fototassi assistita . . . . .	42
3.4.1	Parametri . . . . .	43
3.4.2	Risultati . . . . .	44
3.4.3	Conclusioni . . . . .	46
3.5	Obstacle avoidance . . . . .	48
3.5.1	Parametri . . . . .	48
3.5.2	Risultati . . . . .	50
3.5.3	Conclusioni . . . . .	52
3.6	Path following con partenza fuori percorso . . . . .	54
3.6.1	Parametri . . . . .	54
3.6.2	Risultati . . . . .	56
3.6.3	Conclusioni . . . . .	58
3.7	Path following con partenza sul percorso . . . . .	58



<i>INDICE</i>	ix
3.7.1 Parametri . . . . .	60
3.7.2 Risultati . . . . .	61
3.7.3 Conclusioni . . . . .	63
<b>Conclusioni e lavori futuri</b>	<b>67</b>
<b>Ringraziamenti</b>	<b>69</b>

# Elenco delle figure

1.1	Esempio di rete booleana e delle tabelle di verità. [12] . . . . .	2
1.2	Confine tra regime ordinato e caotico in funzione dei parametri $k$ e $p$ . [13] . . . . .	5
1.3	Rappresentazione del meccanismo di adattamento online in un robot controllato da una rete booleana. [7] . . . . .	6
1.4	Modello dei rapporti causali fra sensori, attuatori e resto del mondo. Qui l'agente prende decisioni esclusivamente in funzione dello stato sensoriale. . . . .	12
1.5	Modello dei rapporti causali fra sensori, attuatori e resto del mondo. L'agente è dotato di "libero arbitrio" e può decidere le azioni da compiere indipendentemente dalla sua percezione del mondo. . . . .	13
1.6	Rappresentazione concettuale dello sciame di veicoli sottomarini durante l'analisi dei fondali marini. . . . .	15
1.7	Modello del ciclo di interazione uomo-macchina per l'ottimizzazione dell' <i>empowerment</i> dell'utente. . . . .	16
2.1	Architettura del simulatore ARGoS. [11] . . . . .	20
2.2	Esempio di <i>footbot</i> . [1] . . . . .	21
2.3	Disposizione e indicizzazione dei sensori di (a) luce (b) prossimità e (c) terreno del <i>foot-bot</i> . . . . .	22
2.4	Disposizione dei cingoli del <i>foot-bot</i> . . . . .	23

3.1	Esempio di <i>box plot</i> che mostra la distribuzione dei valori iniziali e finali di ogni replica. . . . .	31
3.2	Esempio di <i>run-length distribution</i> . . . . .	32
3.3	Esempio di <i>box plot</i> per il calcolo delle soglie di <i>cut-off</i> . . . . .	33
3.4	Esempio di grafico che mostra la traiettoria del robot all'interno dell'arena. . . . .	34
3.5	Ambiente usato per il <i>task</i> di fototassi assistita. . . . .	36
3.6	Ambiente usato per il <i>task</i> di <i>obstacle avoidance</i> . . . . .	38

3.7	Configurazione degli ostacoli nel task di <i>obstacle avoidance</i> . . . . .	40
3.8	Ambiente usato per il task di <i>path following</i> . (a) Partenza fuori dal percorso. (b) Partenza sul percorso. . . . .	41
3.9	Risultato della fase <i>B</i> del task di fototassi assistita. (a) stima della soglia di cut-off. (b) RLD dell' <i>empowerment</i> . . . . .	44
3.10	Route di alcune delle migliori reti pre-adattate nel task di fototassi assistita. . . . .	45
3.11	Confronto fra RLD della funzione obiettivo nel <i>task</i> di fototassi fra reti pre-adattate e non. Soglie: 50%, 60%, 70%. . . . .	46
3.12	Distribuzioni della funzione obiettivo nel task di fototassi assistita.	47
3.13	Risultato della fase <i>B</i> del task di <i>obstacle avoidance</i> . (a) stima della soglia di cut-off. (b) RLD dell' <i>empowerment</i> . . . . .	50
3.14	Route di alcune delle migliori reti pre-adattate nel task di <i>obstacle avoidance</i> . . . . .	51
3.15	Confronto fra RLD della funzione obiettivo nel <i>task</i> di <i>obstacle avoidance</i> fra reti pre-adattate e non. Soglie: 50%, 60%, 70%. . . . .	52
3.16	Distribuzioni della funzione obiettivo nel task di <i>obstacle avoidance</i> . . . . .	53
3.17	Risultato della fase <i>B</i> del task di <i>path following</i> con partenza fuori dal percorso. (a) stima della soglia di cut-off. (b) RLD dell' <i>empowerment</i> . . . . .	55
3.18	Route della migliore rete pre-adattata nel task di <i>path following</i> con partenza fuori dal percorso. . . . .	56

3.19	Confronto fra RLD della funzione obiettivo nel <i>task</i> di <i>path following</i> (fuori percorso) fra reti pre-adattate e non. Soglie: 50%, 60%, 70%. . . . .	57
3.20	Distribuzioni della funzione obiettivo nel <i>task</i> di <i>path following</i> con partenza fuori dal percorso. . . . .	59
3.21	Risultato della fase <i>B</i> del <i>task</i> di <i>path following</i> con partenza sul percorso. (a) stima della soglia di cut-off. (b) RLD dell' <i>empowerment</i> . . . . .	61
3.22	Route della migliore rete pre-adattata nel <i>task</i> di <i>path following</i> con partenza sul percorso. . . . .	62
3.23	Confronto fra RLD della funzione obiettivo nel <i>task</i> di <i>path following</i> (sul percorso) fra reti pre-adattate e non. Soglie: 50%, 60%, 70%. . . . .	63
3.24	Distribuzioni della funzione obiettivo nel <i>task</i> di <i>path following</i> con partenza sul percorso. . . . .	64



# Introduzione

Questo lavoro di tesi si colloca nel campo dei sistemi intelligenti robotici. Nello specifico, vengono trattati alcuni fenomeni riguardanti robot controllati da reti booleane critiche. Un sottoinsieme dei nodi di una rete viene mappato ai sensori e agli attuatori del robot, rendendo il comportamento del robot dipendente dalla dinamica della rete stessa. La scelta di questi nodi avviene tramite un meccanismo di adattamento online: il robot apprende infatti le configurazioni migliori nel corso della sua vita in base a un fattore di merito, senza modificare la struttura interna della rete.

La funzione di merito è un aspetto cruciale di questo meccanismo, in quanto permette di guidare efficacemente il processo di adattamento. Solitamente per ogni *task* viene definita una funzione specifica, ideata e codificata manualmente dal progettista. Alcuni autori hanno cercato di definire una funzione di merito universale, che traesse spunto dal mondo naturale. Essa prende il nome di *empowerment* [10] e permette ad agenti di qualsiasi genere di co-adattare i propri sensori e attuatori a sé stessi e all'ambiente, selezionando quei comportamenti che permettono loro di massimizzare la *percezione* dei cambiamenti ambientali scatenati in risposta alle loro azioni.

Lo scopo di questa tesi è, quindi, valutare l'applicabilità della funzione di *empowerment* come misura di merito in un contesto di adattamento online di robot controllati da reti booleane critiche. I robot sono sottoposti a tre

tipi di *task* classici della robotica, quali fototassi, *obstacle avoidance* e *path following*, per ognuno dei quali viene definita un'adeguata funzione obiettivo. I *task* sono eseguiti da un simulatore per ottenere una maggiore facilità e velocità di riproduzione degli esperimenti. L'adattamento viene condotto utilizzando come funzione obiettivo sia le funzioni di merito specifiche, sia l'*empowerment*. I risultati vengono poi comparati con l'obiettivo di valutare se l'utilizzo dell'*empowerment* come fattore di merito permette di ottenere maggiori prestazioni in termini di funzione obiettivo specifica, utilizzandola sia durante l'intera durata dell'adattamento, sia in un periodo limitato, preliminare all'impiego di una funzione classica.

La tesi è strutturata nel modo seguente. Nel capitolo 1 vengono introdotte le reti booleane e le caratteristiche della funzione di *empowerment*. Nel capitolo 2 viene data una breve descrizione del software di simulazione e del tipo di robot impiegato per gli esperimenti. Nel capitolo 3 vengono descritti gli esperimenti, il tipo di indagini condotte, i risultati ottenuti e alcune conclusioni tratte da essi.



# Capitolo 1

## Stato dell'arte

In questo capitolo verranno riportati brevemente i contributi che hanno ispirato le indagini sperimentali condotte in questo lavoro di tesi.

Nella sezione 1.1 faremo una panoramica sulle reti booleane e sulla loro dinamica, oltre a riassumere un meccanismo di adattamento online in robot controllati da reti booleane. Nella sezione 1.2 sarà spiegato il concetto di *empowerment* e il suo funzionamento, per poi riportare alcuni contributi in cui ne è stato fatto uso.

### 1.1 Reti booleane

In questa sezione daremo una definizione formale delle reti booleane, nonché dei parametri che ne determinano il comportamento osservabile. Inoltre, sarà brevemente descritta una tecnica per utilizzare le reti booleane come software di controllo per robot in un contesto di adattamento *online*.

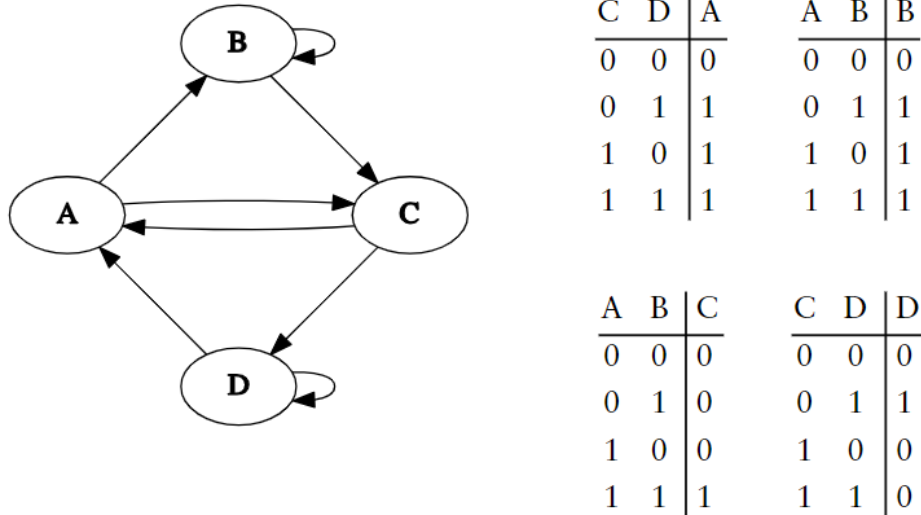


Figura 1.1: Esempio di rete booleana e delle tabelle di verità. [12]

### 1.1.1 Definizioni

Le reti booleane sono un modello originariamente introdotto da Kauffman [9] per descrivere il processo di regolazione genetica che regola il metabolismo cellulare. Una rete booleana è un sistema discreto nel tempo e nello stato, descrivibile topologicamente come un grafo diretto. Formalmente [6] ogni nodo contiene un valore booleano  $x_i, i = 1, \dots, n$ : se la rete contiene  $n$  nodi, il numero di stati possibili della rete corrisponde a  $2^n$  possibili configurazioni. Ogni nodo della rete presenta  $k$  archi in entrata e un arco in uscita. Inoltre, ogni nodo contiene una tabella di verità  $f_i(x_{i_1}, \dots, x_{i_k})$  i cui argomenti rappresentano i valori dei nodi i cui archi in uscita sono connessi al nodo  $i$ ; essa permette di determinare lo stato di ogni nodo all'istante  $t + 1$  unicamente in funzione dei suoi input all'istante  $t$ . Lo stato della rete al tempo  $t$  corrisponde alla lista dei valori dei nodi al tempo  $t$ , che si può definire formalmente come  $s(t) = (x_1(t), \dots, x_n(t))$ . Le sequenze di stati che la rete attraversa durante la sua evoluzione prendono il nome di *traiettorie*. In assenza di perturbazioni

degli stati interni dei nodi, le reti booleane sono a tutti gli effetti un modello completamente deterministico.

Nel caso in cui le connessioni fra nodi e le tabelle di verità vengano generate in modo casuale, queste reti prendono il nome di *reti booleane casuali* (o *RBN*, acronimo di *Random Boolean Networks*).

### 1.1.2 Attrattori

Una rete booleana può presentare uno o più attrattori. Un attrattore non è altro che un sottoinsieme dei possibili stati della rete verso il quale essa tende nel tempo. Una volta raggiunto un attrattore, se il sistema è deterministico e in assenza di perturbazioni, la rete vi resterà per sempre. Nel caso in cui vi siano perturbazioni tali da indurre un “salto” da un attrattore a un altro – ma anche l’arrivo o l’uscita da uno – la rete può attraversare quello che si definisce un *transiente*, ossia una traiettoria che separa i due momenti della dinamica della rete. Gli attrattori ricadono sostanzialmente in due categorie: possono essere *punti fissi*, ossia composti da un singolo stato, o *attrattori ciclici*, se composti da più stati che vengono visitati in ciclo a ogni passo di aggiornamento della rete. In ogni caso, gli attrattori rappresentano dei punti di equilibrio *stabile* e *dinamico*: stabile perché piccole perturbazioni dello stato non influenzano significativamente il comportamento della rete; dinamico poichè l’equilibrio – come avviene nel mondo fisico/chimico – è spesso dovuto all’eguagliarsi di forze o stimoli contrapposti.

Il concetto di attrattore è stato considerato come un forte punto di contatto fra la robotica e la biologia. In particolare, è stato suggerito [12] come i processi di differenziazione cellulare – o più in generale, i meccanismi di regolazione genetica e della sintesi proteica – possano essere descritti come attrattori fra cui la cellula si sposta in seguito a stimoli esterni. Sono stati condotte,

infatti, delle prove empiriche *in vitro* in cui due cellule genotipicamente identiche, se sollecitate con stimoli biochimici distinti, riescono a convergere allo stesso stato stabile dopo aver seguito traiettorie inizialmente divergenti, il che suggerisce la presenza di un attrattore in grado di far tendere i due organismi al medesimo comportamento. Per quanto riguarda il caso di robot controllati da reti booleane, è stato osservato come gli stati delle reti possano seguire un meccanismo analogo alla regolazione genetica, per cui tali robot siano in grado di manifestare comportamenti stabili – sia semplici che composti – raggiungibili tramite perturbazioni ambientali (vedasi 1.1.4). L'idea di fondo è che gli attrattori possano rappresentare i comportamenti di questi robot nello stesso modo in cui possono rappresentare i vari comportamenti delle cellule.

### 1.1.3 Reti critiche

Abbiamo visto come le reti booleane possono essere generate casualmente: dati i parametri  $n$  e  $k$ , infatti, è possibile scegliere per ciascun nodo  $k$  nodi con i quali stabilire un arco entrante, e una tabella di verità di  $2^k$  righe. In particolare, per generare quest'ultima è sufficiente definire un parametro  $p$  – che da qui in poi chiameremo *bias* – che esprime la probabilità di assegnare a una riga della tabella di verità il valore 1; ne segue che la probabilità di assegnarvi il valore 0 sia pari a  $(1 - p)$ .

Detto questo, fissato il numero di nodi  $n$ , le caratteristiche di una RBN dipendono esclusivamente dai parametri  $k$  e  $p$ . Le combinazioni dei loro valori conferiscono, in generale, alle reti due possibili regimi di comportamento, che definiamo *ordinato* e *caotico*. In regime ordinato le reti tendono a essere poco suscettibili alle perturbazioni esterne, la cui propagazione tende a essere breve e di scarso impatto, non riuscendo quindi a deviare la traiettoria della rete dall'attrattore precedente. Al contrario, in regime caotico si osserva una mag-

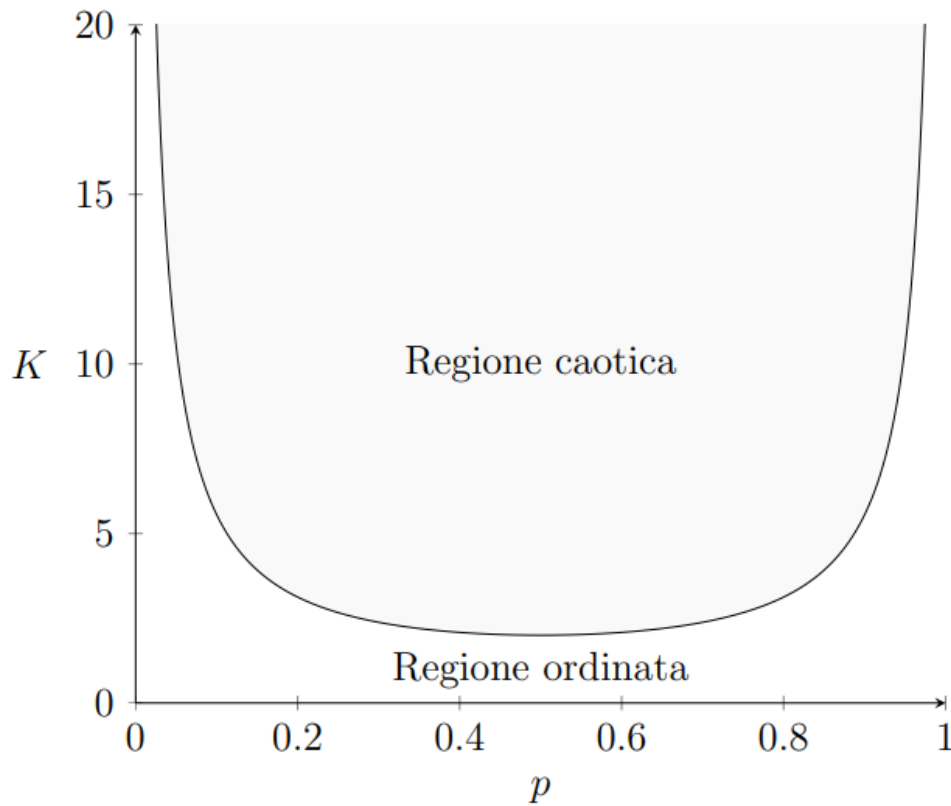


Figura 1.2: Confine tra regime ordinato e caotico in funzione dei parametri  $k$  e  $p$ . [13]

giore propagazione delle perturbazioni, il che rende la traiettoria più instabile e incline a interessare attrattori diversi.

Sulla base dei parametri  $k$  e  $p$  si può definire una formula [3] in grado di descrivere le regioni che caratterizzano i regimi ordinato e caotico:

$$k = \frac{1}{2p(1-p)} \quad (1.1)$$

Il confine tra le due regioni, mostrato in figura 1.2 rappresenta il cosiddetto regime *critico*, che in base a una congettura nota come *criticality hypothesis* definisce reti caratterizzate da un equilibrio tra robustezza e adattabilità [14]. Il confine definito in 1.1 ci permette quindi di trovare i parametri *critici* coi

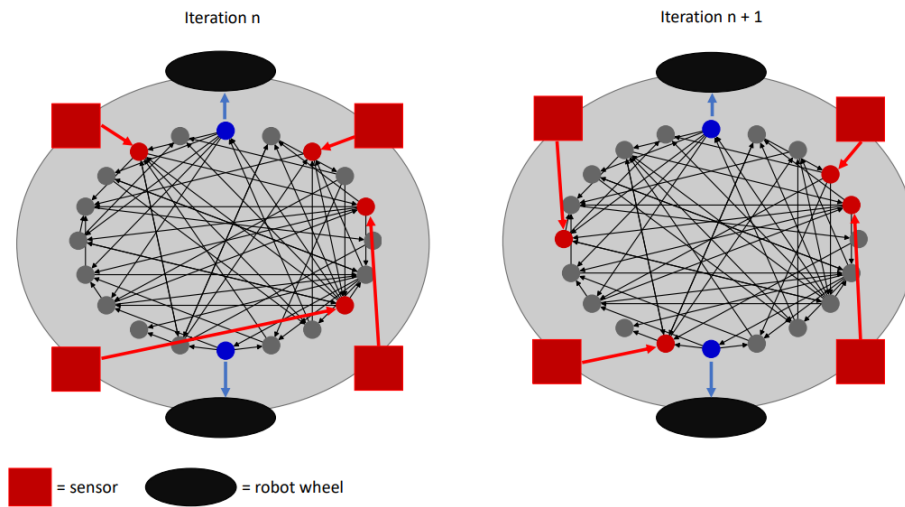


Figura 1.3: Rappresentazione del meccanismo di adattamento online in un robot controllato da una rete booleana. [7]

quali guidare la generazione delle RBN; per esempio, per  $k = 3$  è possibile scegliere i bias 0.21 e 0.79.

#### 1.1.4 Adattamento online

È stato proposto [7] l'utilizzo delle reti booleane come software di controllo nei robot. In questo senso, le reti booleane agiscono in modo analogo alle reti di regolazione genetica nelle cellule: proprio come queste ultime determinano il fenotipo dell'organismo di cui fanno parte, le reti di controllo in un robot ne determinano le caratteristiche e il comportamento osservabile.

In figura 1.3 viene descritto un possibile meccanismo di adattamento *online*. Si noti innanzitutto il meccanismo di controllo: i valori percepiti dai sensori, opportunamente codificati in valori booleani, determinano lo stato interno di alcuni nodi, considerati come *nodi di input*. Il cambiamento dei valori di input causerà nel tempo delle perturbazioni della rete. Analogamente, sono identificati dei *nodi di output* il cui stato, opportunamente decodificato, verrà

usato come segnale per gli attuatori.

Il meccanismo di adattamento *online* proposto prevede che la configurazione della rete venga generata prima dell'esecuzione senza essere mai modificata. Così facendo, l'adattamento consiste esclusivamente nel cambiamento dei collegamenti fra sensori/attuatori e nodi della rete. In questo modo, senza alcun intervento umano, il robot può scegliere autonomamente la configurazione dei suoi input e output in base all'interazione con l'ambiente circostante, optando per collegamenti sempre più favorevoli.

### 1.1.5 Plasticità fenotipica

Questa strategia di adattamento, oltretutto, conferisce ai robot un certo grado di *plasticità fenotipica* [7]. Tale concetto viene inteso in generale come la proprietà degli organismi viventi di esprimere differenti fenotipi a partire da un singolo genotipo, puramente in funzione dell'interazione con ambienti diversi, senza alterare in alcun modo il patrimonio genetico originale. Lo schema di adattamento online precedentemente illustrato permette infatti a robot controllati da reti booleane di mantenere intatto il proprio materiale genetico – la rete booleana di controllo – modulando semplicemente il proprio apparato sensoriale con l'intento di sviluppare comportamenti sempre più consoni alle circostanze ambientali.

## 1.2 Empowerment

In questa sezione verranno discusse le motivazioni che hanno portato alla formulazione dell'*empowerment* come funzione obiettivo *locale* e *universale*; verranno descritti gli strumenti della teoria dell'informazione che ne abilitano il funzionamento; verrà descritto il modello del sistema agente-ambiente che

ne inquadra l'utilizzo; infine verranno riportati alcuni contributi in letteratura che ne hanno fatto impiego.

### 1.2.1 Una prospettiva centrata sugli agenti

Nell'ambito dei sistemi adattativi, è di uso comune definire i compiti da eseguire in termini di funzioni obiettivo. Tali funzioni, ideate dai progettisti e costruite sulla natura specifica degli ambienti e dei *task*, svolgono la doppia funzione di determinare, in termini generali, l'operazione che l'agente deve svolgere e la bontà del comportamento esibito nell'ambiente in cui esso è situato. Guardando al mondo naturale, alcuni autori si sono chiesti [10] se potesse esistere un principio di utilità più generale. In fondo, le funzioni obiettivo sono definite a priori in modo piuttosto specifico, e devono essere costruite traducendo un insieme di proprietà – possibilmente astratte e/o qualitative – in linguaggio matematico, o comunque in una forma adatta a rappresentare la funzione di utilità desiderata. Per questi motivi, ci si è interrogati sull'esistenza di un parallelismo con il mondo naturale, atto a supportare l'uso di una qualche misura di “merito” più astratta e condivisibile da diverse tipologie di agenti, possibilmente immersi in ambienti differenti.

Viene quindi messa in campo l'ipotesi che esista una funzione di utilità a un tempo *locale* e *universale*: locale perché non richiede una conoscenza globale del mondo, nè una memoria infinitamente lunga delle esperienze passate, ma che possa fornire un *feedback* tarato sull'individuo; universale perché adatta a qualunque specie e quindi in grado di adattarsi sia a differenze morfologiche che differenze di nicchie ambientali.

È stato proposto [8] come gli esseri viventi non descrivano il mondo a loro circostante in termini prettamente geometrici, temporali, o meccanici, bensì secondo la consapevolezza di ciò che un essere può percepire e delle azioni



che può compiere. Adottando questa visione dell'adattamento, è facile vedere come gli esseri viventi, in generale, tendano a preferire situazioni nelle quali dispongono di più opzioni fra cui poter scegliere, o comunque circostanze che permettono loro di esercitare – potenzialmente – un maggiore controllo sul mondo. Per esempio, un batterio preferirà trovarsi in punti dove la concentrazione di sostanze nutrienti è più alta, poichè questo gli permetterà maggiori possibilità in termini di sopravvivenza e riproduzione, mentre per l'essere umano disporre di maggiori quantità di denaro significherà maggiori opportunità e possibilità di scelta.

Resta tuttavia un quesito fondamentale: qual è il criterio per misurare la capacità di controllo che un agente possiede? Inoltre, come è possibile distinguere davvero, dall'esterno, se due azioni sono differenti fra loro a parità di risultato? Si pensi a un essere umano che sfoglia un libro: una pagina può essere girata con la mano destra o con la sinistra. Agli occhi di uno spettatore esterno potrà sembrare che il compito di “girare una pagina” venga eseguito correttamente in entrambi i casi; questa valutazione, tuttavia, non tiene conto della *percezione* che l'essere umano ha avuto di sè e delle azioni che ha appena eseguito. Per un destrimano, girare la pagina con la mano sinistra avrà certamente dato l'impressione di esercitare un controllo molto minore rispetto al caso in cui viene usata la mano destra. Si avrà una situazione analoga nel caso di una persona mancina che si trovi ad usare la sua mano destra nello stesso scenario. Certamente uno spettatore esterno potrà avere una visione diversa del livello di controllo esercitato dal soggetto: infatti, monitorando l'attività neurale della persona osservata, noterà come essa riesca effettivamente a controllare i suoi neuroni per attivare i muscoli necessari. Tuttavia, tale percezione sarà molto differente da quella del soggetto, che sarà con ogni probabilità ignara – o quantomeno non cosciente – di questa sua possibilità.

Per questi motivi, le azioni compiute da un agente verranno descritte solamente in termini di ciò che l'agente stesso *percepisce*. In questo modo, basando una funzione obiettivo esclusivamente sui sensori e sugli attuatori di un agente, la valutazione della sua performance risulterà scollegata dalla visione “dall'alto” di un qualsiasi osservatore. Per fare ciò serviranno degli strumenti tratti dalla teoria dell'informazione, che permetteranno di misurare, in ultima analisi, la quantità di controllo che un agente percepisce di possedere.

### 1.2.2 Il problema della comunicazione

Prima di tutto è necessario definire il problema della comunicazione della teoria dell'informazione, assumendo di disporre di un canale di comunicazione *discreto* e privo di memoria.

#### Canale di comunicazione

Consideriamo un mittente e un ricevente. Il mittente invia un segnale, rappresentato dalla variabile aleatoria  $X$ , al ricevente, il quale riceve un messaggio possibilmente diverso da quello inviato, che assegneremo alla variabile aleatoria  $Y$ . Il canale usato per comunicare determina la corrispondenza tra il segnale inviato e quello ricevuto; nel caso di segnali discreti, questo canale può essere descritto dalla distribuzione condizionale  $p(y|x)$ .

#### Mutual Information

È ora possibile definire la *mutual information*, che misura la quantità di informazione contenuta nel segnale ricevuto proveniente dal messaggio inviato. La formula si può definire in funzione della distribuzione caratteristica del

canale  $p(y|x)$  e della distribuzione di probabilità del segnale trasmesso  $p(x)$ :

$$I(X;Y) = \sum_{X,Y} p(y|x)p(x) \log_2 \frac{p(y|x)}{\sum_X p(y|x)p(x)} \quad (1.2)$$

Ricordiamo che la *mutual information* è simmetrica e acausale, non richiedendo la conoscenza delle distribuzioni a priori dei due segnali.

### Capacità di canale

La capacità di canale viene definita come la massima *mutual information* del canale, calcolata su tutte le possibili distribuzioni del segnale trasmesso:

$$C(p(y|x)) = \max_{p(x)} I(X;Y) \quad (1.3)$$

Contrariamente alla *mutual information*, la capacità di canale è asimmetrica e causale, poichè richiede il controllo completo della variabile  $X$ .

La capacità di un canale discreto arbitrario può essere stimata in modo efficiente tramite l'algoritmo iterativo di Blahut-Arimoto [5][4].

### 1.2.3 Definizione di empowerment

Se consideriamo nuovamente il dominio della robotica, possiamo sfruttare gli strumenti teorici precedentemente definiti per stabilire la nozione di *empowerment*.

Prendiamo in esame un agente<sup>1</sup> dotato di un certo numero di sensori e attuatori, immerso in un ambiente. Se applichiamo la classica metafora del loop percezione-azione, sappiamo che le azioni dell'agente in qualche modo riflettono gli input sensoriali, i quali a loro volta possono riflettere le azioni precedenti. Definendo formalmente questi elementi, possiamo indicare con  $S$

---

<sup>1</sup>Qui usiamo volutamente il termine *agente* per includere anche i sistemi simulati – o comunque, non necessariamente fisici.

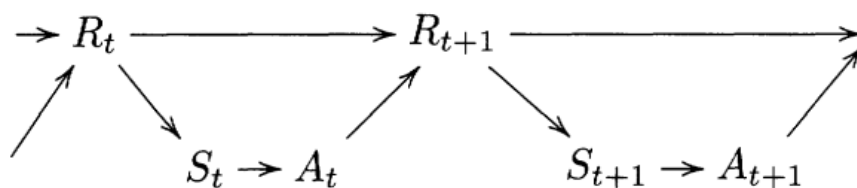


Figura 1.4: Modello dei rapporti causali fra sensori, attuatori e resto del mondo. Qui l'agente prende decisioni esclusivamente in funzione dello stato sensoriale.

i sensori,  $A$  gli attuatori e  $R$  il “resto del sistema”, il quale può comprendere l'ambiente, la morfologia dell'agente, e perturbazioni varie.

Per catturare le dinamiche temporali e causali di questo modello, possiamo effettuare un *unrolling* nel tempo: sia  $t$  un qualsiasi istante temporale, definiamo  $S_t$ ,  $A_t$  e  $R_t$  rispettivamente come lo stato dei sensori, degli attuatori e il resto del sistema all'istante  $t$ . Così facendo è possibile indicare i rapporti fra queste variabili nel tempo tramite una rete Bayesiana (fig. 1.4): al tempo  $t$ , le condizioni ambientali  $R_t$  determinano i valori dei sensori  $S_t$  i quali, a loro volta, determinano lo stato degli attuatori  $A_t$ . Queste azioni potranno quindi avere un impatto sul mondo esterno  $R_{t+1}$ .

Il modello così creato ci permette di interpretare il *loop* percezione-azione nei termini del problema della comunicazione (1.2.2). Difatti, è come se le azioni eseguite dall'agente *iniettassero* informazione prima nell'ambiente, e poi nello stato dei sensori dello stesso agente, in quanto esso sarà in grado di percepire i cambiamenti che le sue stesse decisioni hanno causato in virtù di un certo livello di controllo sul mondo. In questo modo, è come se esistesse un vero e proprio *canale di comunicazione* la cui sorgente è costituita dallo stato degli attuatori, la destinazione dallo stato dei sensori; l'ambiente, invece, è come se

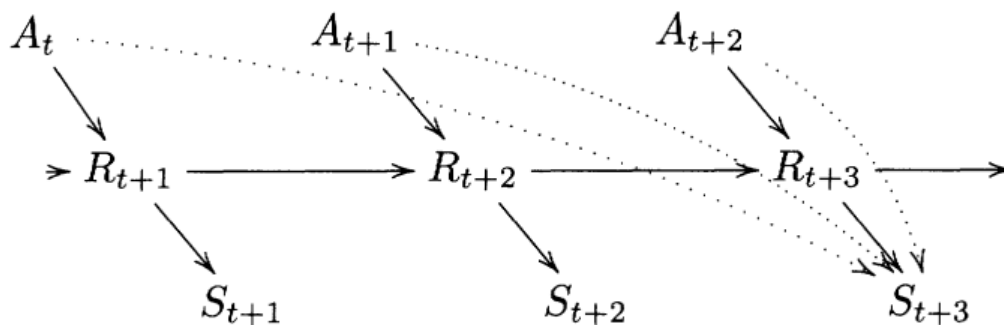


Figura 1.5: Modello dei rapporti causali fra sensori, attuatori e resto del mondo. L'agente è dotato di “libero arbitrio” e può decidere le azioni da compiere indipendentemente dalla sua percezione del mondo.

fungesse da filtro, limitando e/o perturbando le informazioni trasmesse.

Ora è possibile porsi la seguente domanda: se l'agente potesse, all'istante  $t$ , deliberare *arbitrariamente* l'esecuzione di una sequenza di  $n$  azioni, quanta informazione riuscirebbe a trasmettere ai propri sensori al tempo  $t+n$ ? Leggendo questa situazione ancora una volta tramite il problema della comunicazione, possiamo modellare nuovamente i rapporti di causa-effetto che intercorrono tra gli stati dei sensori, degli attuatori e dell'ambiente tramite una rete causale, come mostrato in figura 1.5. Possiamo denotare la sequenza di  $n$  azioni decise al tempo  $t$  con la variabile aleatoria  $A_t^n = (A_t, A_{t+1}, \dots, A_{t+n})$  e i suoi possibili valori con  $a_t^n$ . Scriviamo lo stato dei sensori dopo l'esecuzione delle suddette azioni con la variabile aleatoria  $S_{t+n}$  e i suoi possibili valori con  $s_{t+n}$ . Infine, i rapporti causali tra le variabili in oggetto possono essere catturati tramite la distribuzione condizionale  $p(s_{t+n}|a_t^n)$ , che descrive il canale di comunicazione.

Definiamo quindi l'*empowerment* come la capacità del canale di comunicazione che parte dagli attuatori dell'agente e termina nei suoi sensori:

$$\mathfrak{E}_t = C(p(s_{t+n}|a_t^n)) = \max_{p(a_t^n)} I(a_t^n; S_{t+n}) \quad (1.4)$$

Questa quantità è misurata in bit: si ottiene un valore uguale a 0 se l'agente non esercita alcun controllo percepibile sull'ambiente o, analogamente, se l'agente non riesce a trasmettere informazione dai suoi attuatori ai suoi sensori tramite l'ambiente descritto dal canale di comunicazione.

È bene precisare che non è necessario definire la funzione di *empowerment* tramite una singola lettura dei sensori, ma in via teorica può essere calcolata su una qualsiasi variabile campionata successivamente a  $A_t$ . Per esempio è possibile prendere una sequenza di stati sensoriali  $S_t^n = (S_{t+1}, S_{t+2}, \dots, S_{t+n})$ , oppure una qualche informazione relativa alla memoria dell'agente, sempre ammesso che faccia riferimento a istanti temporali successivi a  $t$ .

In generale, massimizzare l'*empowerment* significa:

1. cercare di raggiungere quelle parti del mondo in cui le azioni dell'agente riescono a determinare al meglio gli stati sensoriali;
2. modificare i sensori e gli attuatori dell'agente in modo da migliorare il controllo dell'*empowerment*.

Mettere in pratica il punto 2. significa, di fatto, co-adattare i sensori e gli attuatori gli uni con gli altri, in modo da adattarli all'ambiente in cui l'agente è immerso.

### 1.2.4 Applicazioni

#### Ispezione di fondali marini

WiMUST [2] è un progetto finanziato dalla Commissione Europea il cui scopo è progettare e testare uno sciame di veicoli sottomarini autonomi in grado di sondare i fondali marini.

In questo progetto, l'*empowerment* è stato adottato inizialmente come euristica per migliorare e rendere più flessibili le reazioni dei veicoli in scenari

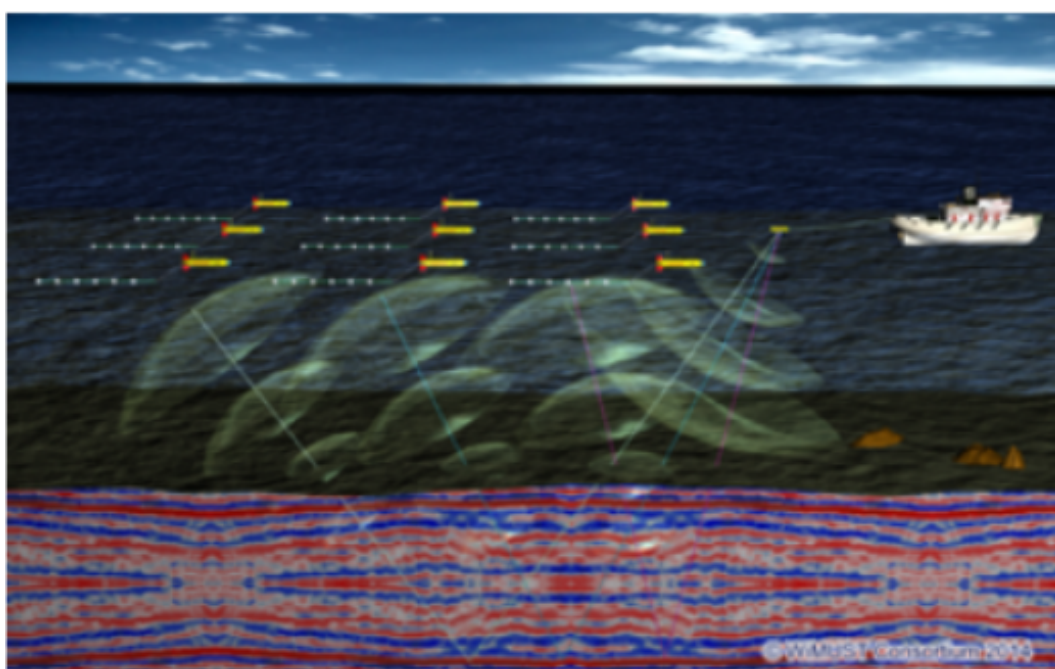


Figura 1.6: Rappresentazione concettuale dello sciame di veicoli sottomarini durante l'analisi dei fondali marini.

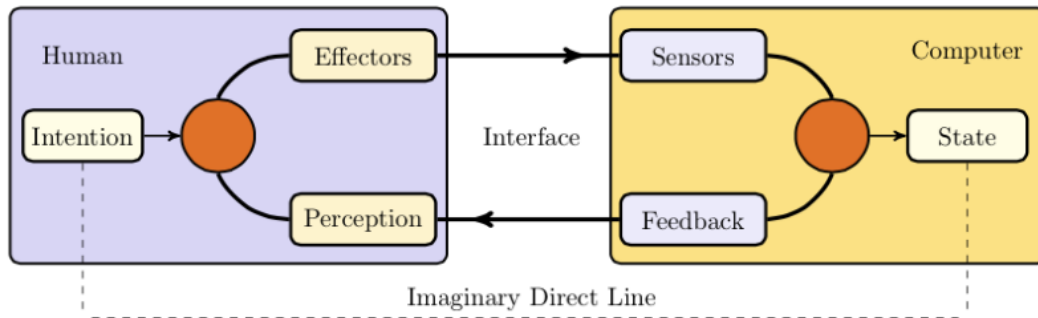


Figura 1.7: Modello del ciclo di interazione uomo-macchina per l'ottimizzazione dell'*empowerment* dell'utente.

critici. Successivamente, ci si è posti l'obiettivo di ideare dei controller basati sull'*empowerment* nell'ottica di allontanarsi il più possibile da algoritmi progettati a mano, in quanto carenti di flessibilità, favorendo una soluzione che potesse proporre reazioni basate sulle singole situazioni da affrontare. Infatti, tale flessibilità si pensa che renda possibile un notevole incremento di auto-sufficienza e *survivability*, in quanto non è più necessario specificare manualmente questo requisito nella definizione della missione.

In figura 1.6 è possibile trovare una stilizzazione del funzionamento delle flotte sottomarine.

## Human-Computer Interaction

Alcuni autori hanno ipotizzato [15] come l'*empowerment* potrebbe essere utilizzato nell'ambito delle interfacce utente per misurare la quantità di controllo percepita da un utilizzatore umano nei confronti della sua esperienza di navigazione e interazione con un'interfaccia.

In figura 1.7 è possibile osservare il meccanismo di interazione uomo-macchina immaginato dagli autori, che prende la forma di un ciclo chiuso di feedback. In questo contesto, un utente si sentirà tanto più in controllo delle sue azioni quan-



to maggiore sarà capace di adattarsi all'interfaccia e ricevere un riscontro delle proprie azioni da essa; nel modello presentato, oltretutto, l'interfaccia stessa viene pensata come un sistema composto da sensori – che ricevono l'input dall'utente – e da degli “attuatori” che corrispondono alle azioni di feedback inviate all'utente.



# Capitolo 2

## Strumenti di simulazione

In questo capitolo presenteremo una breve descrizione del simulatore ARGoS (*Autonomous Robots Go Swarming*) usato per gli esperimenti condotti in questa tesi. Si parlerà inoltre delle caratteristiche principali del *foot-bot*, il tipo di robot usato nelle simulazioni.

### 2.1 Il simulatore ARGoS

Per condurre gli esperimenti è stato usato ARGoS [11], un simulatore *multi-robot* altamente configurabile, sviluppato presso il laboratorio IRIDIA<sup>1</sup> della *Université libre de Bruxelles*. ARGoS è un simulatore efficiente e flessibile: l'efficienza è data dalle alte prestazioni raggiunte come simulatore parallelo; la flessibilità viene raggiunta permettendo all'utente di personalizzare il simulatore stesso al fine di lanciare esperimenti specifici.

In figura 2.1 si può trovare un diagramma che mostra l'architettura di ARGoS. In questa rappresentazione, i rettangoli bianchi corrispondono alle componenti personalizzabili dall'utente.

---

<sup>1</sup><https://code.ulb.ac.be/lab/IRIDIA>

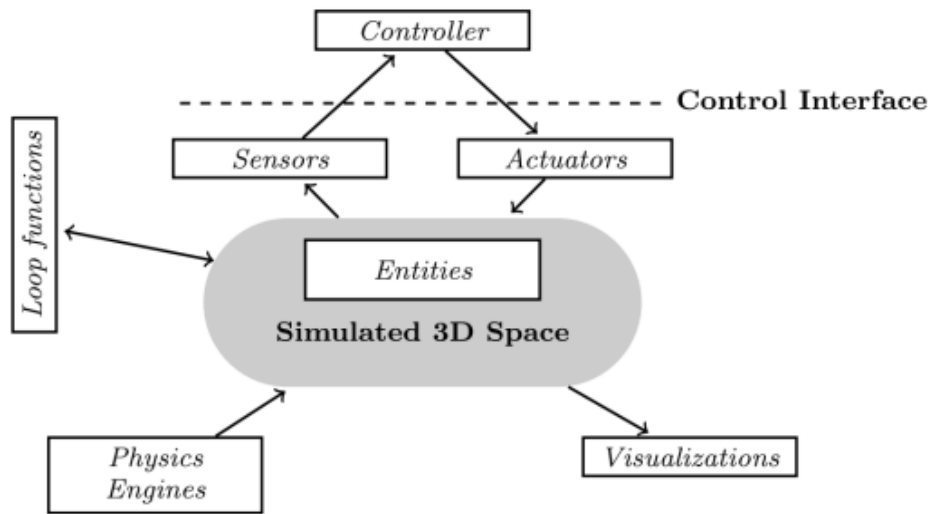


Figura 2.1: Architettura del simulatore ARGoS. [11]

### 2.1.1 Configurazione

In ARGoS, le simulazioni sono definite e configurate tramite un apposito file XML. Esso contiene le informazioni relative a:

- durata dell'esperimento;
- *controller* dei robot;
- sensori e attuatori;
- ambiente di simulazione;
- *engine* fisico;
- visualizzazione.

L'ambiente è ampiamente configurabile, e permette di indicare posizione e dimensioni di muri, ostacoli, luci e pavimento. Per quest'ultimo, inoltre, è possibile definire un'immagine da "proiettare" sul pavimento affinché il robot possa leggerne il colore tramite appositi sensori.

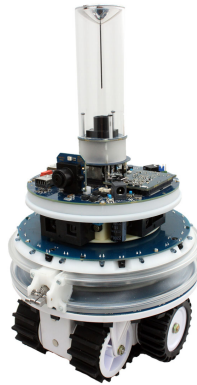


Figura 2.2: Esempio di *footbot*. [1]

## 2.2 Foot-bot

Per gli esperimenti condotti si è scelto di usare un robot denominato *foot-bot*, rappresentato in figura 2.2. Esso è un semplice robot di forma circolare dotato di diversi sensori e attuatori. Il movimento nell'ambiente avviene tramite un paio di cingoli, i quali sono trattati esattamente come un paio di ruote.

### 2.2.1 Sensori

Per gli esperimenti sono stati utilizzati tre tipi di sensori: i sensori di luce, i sensori di prossimità e i sensori di terreno. I sensori di luce (noti come *light*) disposti come in figura 2.3a, sono 24 e seguono la circonferenza del robot. Le letture di ogni sensore sono composte da due valori: un angolo in radianti corrispondente all'orientamento del sensore stesso rispetto al robot, e un valore compreso nell'intervallo  $[0, 1]$ . Se un sensore non percepisce alcuna intensità luminosa, il valore restituito sarà 0. Crescendo l'intensità percepita della luce, questo valore si avvicinerà sempre più a 1, che corrisponde alla massima radiazione percepibile.

I sensori di prossimità (noti come *proximity*) disposti come in figura 2.3b,

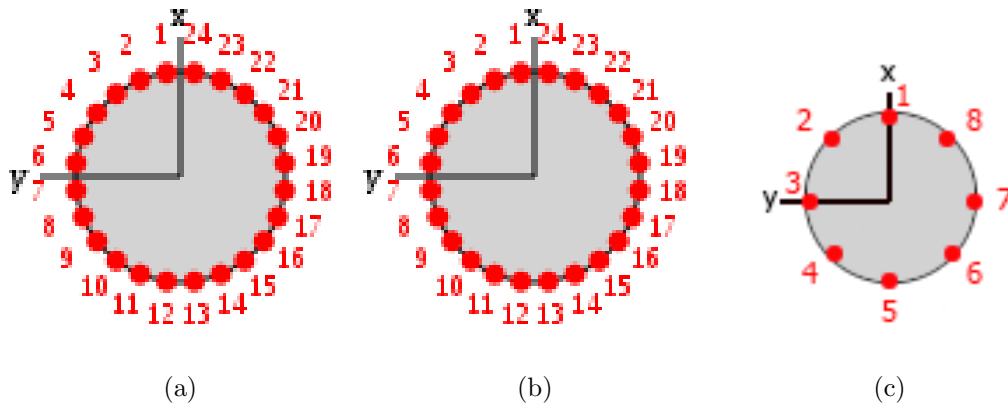


Figura 2.3: Disposizione e indicizzazione dei sensori di (a) luce (b) prossimità e (c) terreno del *foot-bot*.

sono 24 e seguono la circonferenza del robot, in modo analogo ai sensori di luce. Le letture di ogni sensore sono composte da due valori: un angolo in radianti corrispondente all'orientamento del sensore stesso rispetto al robot, e un valore compreso nell'intervallo  $[0, 1]$ . Se un sensore non percepisce alcun ostacolo di fronte a sè, il valore restituito sarà 0. Questo valore sarà tanto più grande quanto sarà ridotta la distanza fra il sensore stesso e l'ostacolo. I sensori di prossimità hanno una portata di 10 centimetri: sopra questa distanza non viene rilevato nulla e viene restituito sempre il valore 0.

I sensori di terreno (noti come *base\_ground*) disposti come in figura 2.3c, sono 8 e anch'essi seguono la circonferenza del robot. Le letture di ogni sensore sono composte da due valori: la posizione in centimetri letta dal sensore, e un valore booleano che corrisponde al colore del terreno: 0 se nero o grigio scuro, 1 se bianco.

Per ragioni di efficienza computazionale, si è deciso di effettuare una fusione dei valori di ciascun tipo di sensore, cercando comunque di mantenerne una buona divisione spaziale. Il tipo di fusione effettuata per ciascun esperimento verrà indicata nella sezione 3.3 dedicata alla definizione dei *task*.

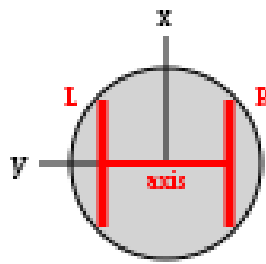


Figura 2.4: Disposizione dei cingoli del *foot-bot*.

### 2.2.2 Attuatori

Ai fini degli esperimenti trattati, gli unici attuatori utilizzati sono stati i cingoli, disposti come in figura 2.4.

È possibile trattare i due cingoli, per semplicità, esattamente come due ruote separate per ognuna delle quali si può specificare una velocità in cm/s.





# Capitolo 3

## Esperimenti

Nel seguente capitolo verrà definita e discussa l'attività sperimentale condotta ai fini di questa tesi.

Saranno definiti gli obiettivi dell'indagine e verrà esposto l'impianto metodologico a supporto dell'esecuzione, della valutazione e dell'interpretazione dei risultati ottenuti. Saranno definite le tipologie dei *task* che i robot dovranno eseguire, oltre alle descrizioni dei relativi ambienti e dei parametri di simulazione. I risultati ottenuti verranno mostrati tramite appositi grafici e, quando possibile, validati attraverso opportuni test statistici, ai quali seguiranno delle brevi conclusioni.

### 3.1 Obiettivo

L'obiettivo di questo studio è stato principalmente quello di valutare, in generale, la fattibilità dell'utilizzo della funzione di *empowerment* come funzione obiettivo in un contesto di adattamento online di robot controllati da reti booleane critiche. Nello specifico, ci si è concentrati sulla possibilità di utilizzare l'*empowerment* per condurre una sorta di *pre-adattamento* delle reti

in un particolare ambiente, cercando di ottenere prestazioni paragonabili all'adattamento con sola funzione obiettivo specifica per il task; o eventualmente, valutare se l'adattamento tramite il solo *empowerment* riesce a generare reti in grado di svolgere un certo compito più efficacemente rispetto a reti addestrate con una classica funzione obiettivo.

L'ipotesi di fondo è che l'*empowerment*, essendo una funzione di utilità locale e universale, se applicata preventivamente a una rete booleana possa aiutare quest'ultima a svolgere un determinato compito, permettendole dapprima di conoscere l'ambiente circostante, adattandovi **autonomamente** i propri sensori e attuatori, per poi essere indirizzata verso una mansione specifica.

Tramite gli esperimenti nelle sezioni successive si cerca quindi di rispondere alle seguenti domande:

1. l'uso dell'*empowerment* come funzione obiettivo riesce a provocare un aumento significativo delle prestazioni di un robot rispetto a un particolare task?
2. l'uso dell'*empowerment* come funzione obiettivo può rendere un robot più efficace nel raggiungere un risultato specifico, rispetto a un robot addestrato esclusivamente tramite una funzione obiettivo progettata esternamente?
3. una rete pre-adattata con *empowerment* e adattata successivamente con una funzione obiettivo specifica presenta prestazioni mediamente migliori di una rete adattata seguendo la sola funzione obiettivo?
4. una rete pre-adattata con *empowerment* e adattata successivamente con una funzione obiettivo può far crescere le performance più velocemente di una rete adattata seguendo la sola funzione obiettivo?

## 3.2 Metodologia

Prima di cercare di dare una risposta soddisfacente alle indagini poste, è necessario definire in modo chiaro la strategia utilizzata nel condurre gli esperimenti, come queste strategie sono state realizzate e descrivere gli strumenti statistici adoperati.

Nel definire ogni tipologia di esperimento, sarà necessario fare chiarezza su una serie di elementi al fine di garantirne al meglio la riproducibilità:

- ambiente sperimentale;
- posizione iniziale del robot;
- numero e tipologia di sensori utilizzati;
- numero e tipologia di attuatori utilizzati;
- funzione obiettivo specifica per il task da valutare.

Ogni fase dell'esperimento verrà configurata tramite una serie di parametri:

- numero di repliche;
- numero di epoche per ogni replica;
- numero di step per ogni epoca;
- numero di nodi della rete;
- numero di connessioni per ogni nodo della rete;
- valore di *bias* con cui generare le tabelle di verità dei nodi;
- numero di step per il calcolo dell'empowerment.

### 3.2.1 Fasi preliminari

Ogni esperimento viene preceduto da due fasi preliminari successive, che nel seguito indicheremo per convenzione con le lettere *A* e *B*.

La fase *A* consiste in un semplice esperimento di adattamento nel quale una rete viene valutata all'interno dell'ambiente predisposto tramite la funzione obiettivo specifica. Questa fase ha lo scopo di creare una *baseline* sulla quale verranno confrontati i risultati della fasi successive. Ad esempio, è utile considerarne i valori della funzione obiettivo, sia in termini di andamento generale (valori iniziali e finali) sia in termini di velocità di convergenza a certe soglie di prestazioni.

La fase *B* consiste nell'adattare una rete booleana esclusivamente seguendo la funzione di *empowerment*, tenendo comunque traccia parallelamente dei valori della funzione obiettivo specifica per il task. Questo sarà utile per tracciare eventuali correlazioni fra i valori raggiunti dell'*empowerment* e della funzione obiettivo, ma principalmente servirà a effettuare una stima del numero medio di epoche necessario per arrivare a una *stagnazione* dell'*empowerment*. Tale soglia si rivelerà utile nelle prime epoche dell'esperimento, dedicate al pre-adattamento della rete.

### 3.2.2 Esecuzione degli esperimenti

Ogni esperimento si compone di due parti consecutive che chiameremo di *pre-adattamento* e di *adattamento*.

Il *pre-adattamento* è un'operazione cruciale di ogni esperimento. Il suo scopo è quello di preparare un robot all'interazione con l'ambiente che lo circonda, co-adattando sensori e attuatori.

La durata del periodo di pre-adattamento viene stimata a partire dai risultati della fase preliminare *B*. I risultati ottenuti indicano il numero di epoche

necessario a causare la stagnazione dei valori di empowerment. La stagnazione è stata definita come il raggiungimento di una determinata soglia, calcolata come una frazione – o percentuale – del valore massimo di *empowerment* raggiunto durante una determinata replica della fase *B*.

Sono state definite tre soglie: 80%, 90% e 100% del valore massimo raggiunto. La durata del periodo di pre-adattamento – qui indicata anche come *cut-off* – è stata definita come la mediana della distribuzione dei valori calcolati con soglia 100%. A tal proposito si è rivelato utile un *box-plot* raffigurante i le distribuzioni dei valori secondo ognuna delle tre soglie.

La fase di *adattamento* consiste invece nell'adattare il robot pre-adattato tramite una classica funzione obiettivo, definita specificamente il *task* da eseguire nell'ambiente dato.

### 3.2.3 Modalità di esecuzione

Gli esperimenti – comprese le fasi preliminari – vengono lanciati più volte in parallelo, con gli stessi parametri ma configurati con *random seed* differenti. In questo testo ci riferiamo al concetto di istanza di un esperimento con il nome di *replica*. Lanciare un numero sufficiente di repliche dello stesso esperimento significa garantire un certo livello di indipendenza statistica dei risultati.

Ogni replica è composta da una serie di *epoche* ognuna contenente lo stesso numero di *step*, che rappresentano le unità atomiche di tempo logico che scandiscono l'evoluzione delle reti di controllo.

### 3.2.4 Sensori e adattamento online

Le reti booleane usate come software di controllo seguono il tipo di integrazione con i robot e le modalità di adattamento *online* esposte nella sezione 1.1.4.

All'inizio dell'esecuzione di ogni replica viene generata una rete booleana *random*, della quale viene selezionato un insieme di nodi da considerare come nodi di input. Ad ogni step di ogni epoca, i valori dei sensori vengono letti e binarizzati, in modo da costituire lo stato interno dei nodi di input.

Alla fine di ogni epoca vengono valutate le prestazioni della rete. A questo punto, a partire dalla rete che ha raggiunto prestazioni massime, viene generata una nuova rete che ne conserva la topologia interna, ma che presenta un diverso insieme di nodi di input. In questo modo il robot risulta in grado di adattarsi all'ambiente circostante dinamicamente nel corso dell'esecuzione.

### 3.2.5 Calcolo delle prestazioni

Il processo di adattamento delle reti booleane è strettamente legato alla modalità con cui vengono calcolate le performance delle stesse. Queste quantità sono calcolate tramite una funzione obiettivo, diversa per ogni *task*, usata per la ricerca della configurazione di ogni rete più adatta a svolgere il compito.

Il valore della funzione obiettivo viene calcolato a ogni step della simulazione. I valori ottenuti vengono sommati nel corso di un'epoca e normalizzati sulla durata in step di quest'ultima. Il risultato aggregato ottenuto alla fine dell'epoca determina il valore di merito della rete; se esso supera il valore massimo registrato sino a quel momento, esso viene considerato come il nuovo massimo e la rete in oggetto viene considerata la rete di riferimento per la ricerca di nuove soluzioni.

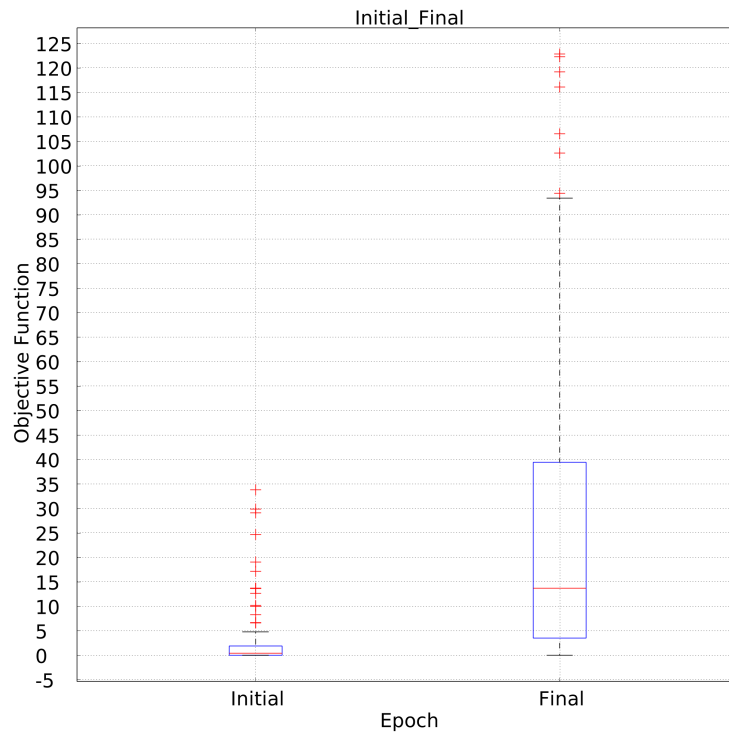


Figura 3.1: Esempio di *box plot* che mostra la distribuzione dei valori iniziali e finali di ogni replica.

### 3.2.6 Grafici e test statistici

#### Distribuzione della funzione obiettivo

Per avere un'idea ben chiara dell'andamento generale delle prestazioni delle reti sono stati utilizzati dei *box-plot* contenenti le distribuzioni dei valori massimi raggiunti da ogni replica dell'esperimento in diversi casi.

Ad esempio, in figura 3.1 si può vedere un esempio di *box-plot* in cui vengono mostrate le distribuzioni dei valori della funzione obiettivo raggiunti dopo la prima e dopo l'ultima epoca di ogni replica.

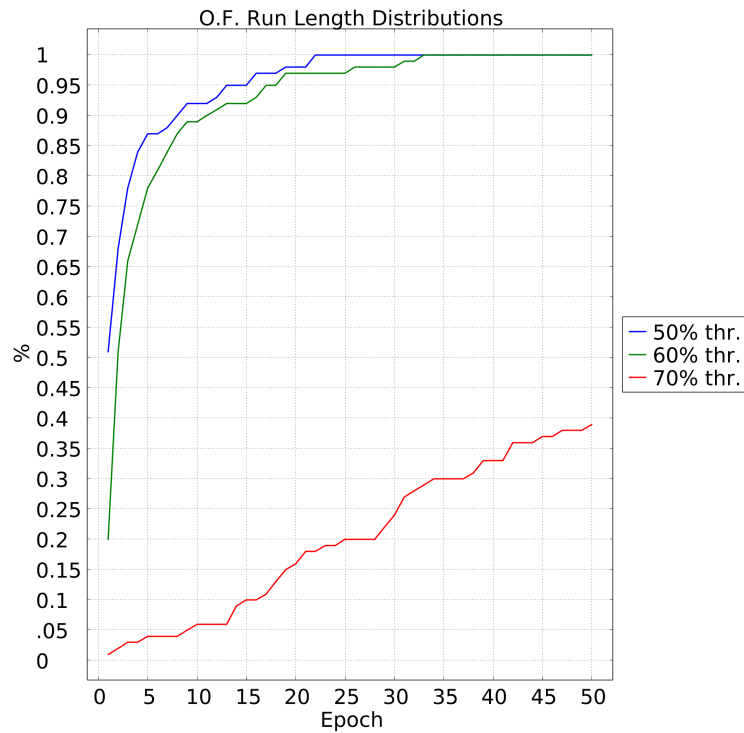


Figura 3.2: Esempio di *run-length distribution*.

## Run Length Distributions

In figura 3.2 si può vedere un esempio di *run-length distribution* (per semplicità da qui in poi useremo l'acronimo RLD). Questo tipo di grafici può dare un'idea generale dei livelli di performance raggiunti dalle repliche dell'esperimento a ogni epoca di simulazione. Le curve mostrate nel grafico rappresentano, per ogni epoca, la frazione di repliche che hanno ottenuto in tale epoca una prestazione uguale o maggiore a una frazione – nell'esempio, 50%, 60% e 70% – del valore massimo raggiunto da tutte le repliche durante la totalità dell'esperimento.



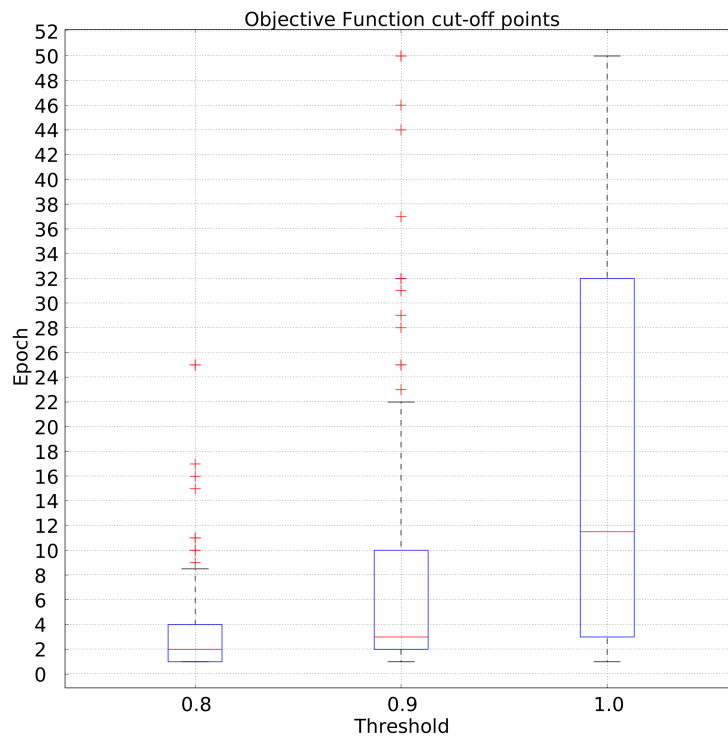


Figura 3.3: Esempio di *box plot* per il calcolo delle soglie di *cut-off*.

### Soglie di *cut-off*

La stima del *cut-off* è essenziale per configurare il processo di pre-adattamento. Per farla si è ricorsi a dei *box plot* contenenti le distribuzioni delle repliche i cui valori di *empowerment* finali superano certe frazioni del valore massimo riscontrato durante le repliche. In figura 3.3 si può vedere un grafico di esempio che mostra le distribuzioni calcolate sulle soglie 0.8, 0.9 e 1.0.

### Traiettoria dei robot

Per poter valutare qualitativamente i comportamenti adottati dai robot nei diversi ambienti sono stati generati dei grafici che ne mostrano gli spostamenti all'interno dell'ambiente.

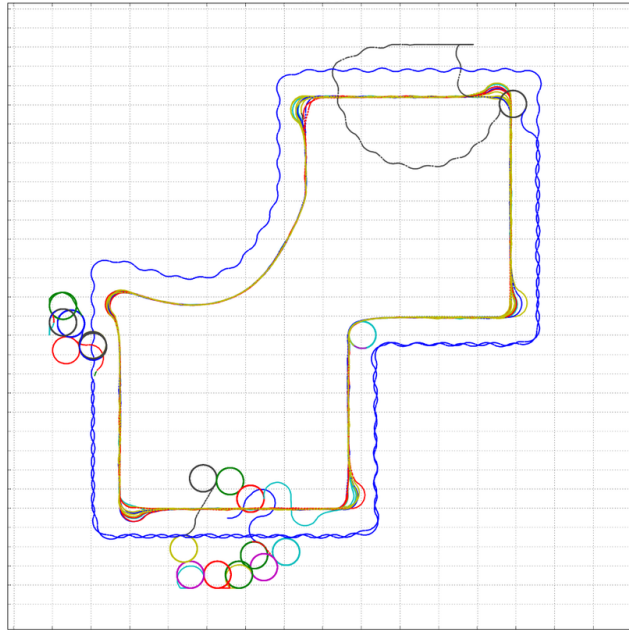


Figura 3.4: Esempio di grafico che mostra la traiettoria del robot all'interno dell'arena.

In figura 3.4 si può trovare un esempio di questo tipo di grafico. È possibile notare come la traiettoria seguita dal robot sia segmentata con diversi colori a seconda dell'epoca di riferimento. Si è volutamente deciso di non mostrare la legenda dei colori di ogni epoca per questioni di ordine, essendosi rivelata utile solo nei momenti di ispezione preliminare dei risultati e per la risoluzione di problemi tecnici.

### Test non-parametrici

Per confermare la veridicità dei risultati ottenuti è necessario fare ricorso a dei test statistici. Non essendoci alcuna assunzione di base sulla natura delle distribuzioni risultanti dagli esperimenti, è necessario fare affidamento a un test non-parametrico noto come test di Wilcoxon-Mann-Whitely. L'ipotesi nulla di questo test afferma che due distribuzioni presentino le stesse mediane

e le stesse forme, mentre l'ipotesi alternativa sostiene che le distribuzioni siano diverse.

Per poter rifiutare l'ipotesi nulla viene considerato il *p-value* generato dal test: se tale valore è più basso di una certa soglia – indicativamente 0.05 – si può confermare l'ipotesi nulla.

### 3.2.7 Descrizione dei risultati

I risultati di ogni esperimento sono presentati tramite due tipi di grafici: *box plot* e RLD.

Queste due tipologie di analisi cercano di dare visioni diverse dello stesso fenomeno. I *box plot* tentano di dare una visione precisa delle prestazioni effettive ottenute nel corso degli esperimenti: le conclusioni che possiamo trarre da essi, coadiuvate dai risultati dei test non parametrici, sono piuttosto affidabili e mai azzardate. Le RLD, invece, ci danno un'informazione molto diversa, in quanto tentano di catturare il numero di “casi” di successo degli esperimenti, in base a soglie ben definite. Infatti, con tali grafici possiamo sempre avere una visione generale di quante repliche dell'esperimento stanno compiendo in modo soddisfacente il task in una data epoca.

## 3.3 Tipi di task

In questa sezione daremo una descrizione dei *task* utilizzati per valutare le prestazioni dei robot. Per ognuno di essi verrà descritto l'ambiente, la sua configurazione spaziale, e la funzione obiettivo specifica per il compito da svolgere.

### 3.3.1 Fototassi assistita

Il primo *task* che è stato simulato è quello della fototassi assistita, che consiste nel cercare di andare verso una luce in corrispondenza della quale, sul pavimento, è posta una macchia nera.

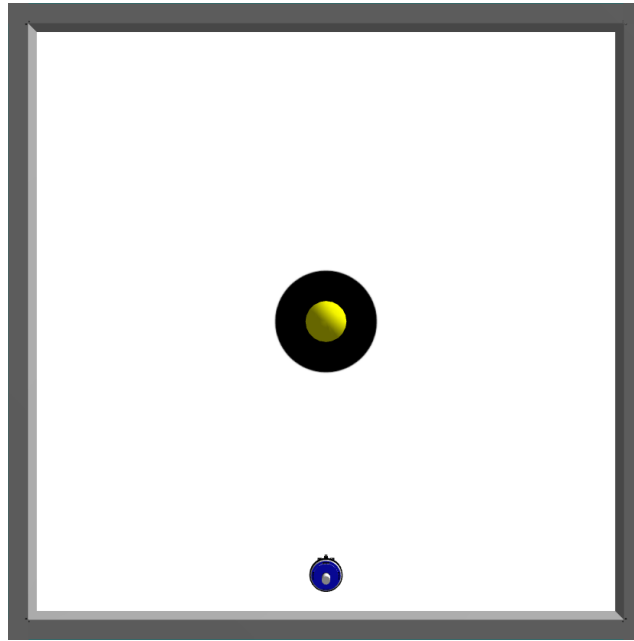


Figura 3.5: Ambiente usato per il *task* di fototassi assistita.

L'arena dell'esperimento (fig. 3.5) è composta da una stanza quadrata di lato  $3\text{ m}$  chiusa da quattro muri perimetrali di dimensioni  $(3, 0.1, 0.25)\text{ m}$ . La loro posizione tiene conto della larghezza del muro stesso, in modo da non togliere spazio all'arena. Per fare ciò, le loro coordinate sono state opportunamente estese aggiungendo un *offset* pari a  $(0.1 / 2)\text{ m} = 0.05\text{ m}$ :

$$\{(0, 1.55, 0), (0, -1.55, 0), (1.55, 0, 0), (-1.55, 0, 0)\}.$$

. Il robot è inizialmente posizionato alle coordinate  $(0, -1.3, 0)$ . La luce ha intensità  $0.5$  ed è posta alle coordinate  $(0, 0, 0.5)$ . Il pavimento è completamente

bianco, fatta eccezione per la macchia circolare di colore nero, posizionata al centro del pavimento, di diametro  $\sim 0.528 m$ .

Il robot è equipaggiato di 24 sensori di luce, le cui letture sono state fuse in quattro gruppi di 6 facendone una media aritmetica. I gruppi sono formati in modo tale da condensare le letture provenienti dalle direzioni “cardinali”: davanti, dietro, a destra e a sinistra del robot. Le letture originali dei sensori sono sottoposte a soglia: i loro valori devono superare il livello prestabilito per non essere considerati nulli.

Inoltre il robot dispone di 4 sensori di terreno, in grado di captare esclusivamente i colori nero e bianco. Essi sono posti complementariamente ai sensori logici di luce, ossia lungo le direzioni intercardinali: nord-ovest, sud-ovest, sud-est e nord-est. Il simulatore ne mette a disposizione 8, ma per ragioni di efficienza si è deciso di selezionare esclusivamente quelli di maggiore interesse.

La funzione obiettivo utilizzata è definita come segue:

$$F_{photo} = \frac{100}{E} \cdot \frac{1}{L + G} \cdot \sum_t l_{max}(t) \cdot L + g(t) \cdot G \quad (3.1)$$

dove:

- $E$  è il numero di step contenuti in un’epoca;
- $L$  è il peso che viene dato alla componente dei sensori di luce (impostato a 4);
- $G$  è il peso che viene dato alla componente dei sensori di terreno (impostato a 1);
- $t$  è il generico step dell’epoca;
- $l_{max}(t)$  è il valore massimo delle letture dei sensori di luce all’istante  $t$ ;
- $g(t)$  è il numero di sensori di terreno attivi all’istante  $t$ .

Questa funzione calcola sostanzialmente una media pesata delle letture dei vari sensori, normalizzandola sul numero di step di ogni epoca per renderla indipendente dalla durata della simulazione.

### 3.3.2 Obstacle avoidance

Il secondo *task* che è stato simulato vede come obiettivo per il robot quello di girare all'interno dell'ambiente evitando il più possibile gli ostacoli che percepisce lungo il suo cammino.

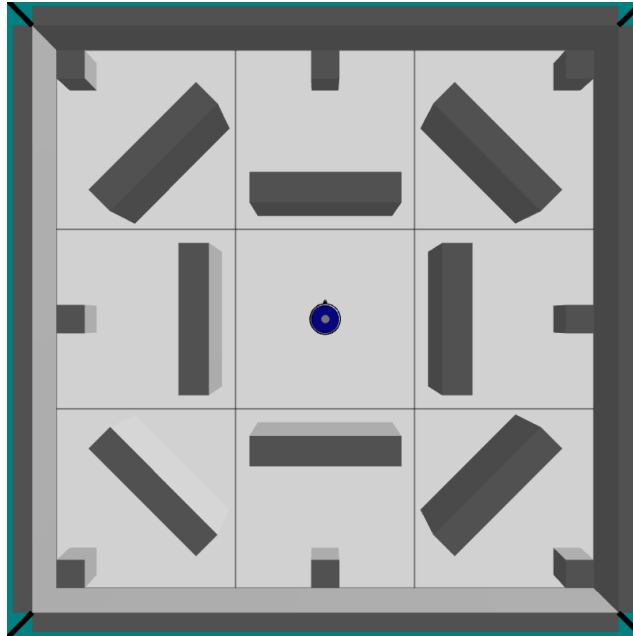


Figura 3.6: Ambiente usato per il *task* di *obstacle avoidance*.

L'arena dell'esperimento (fig. 3.6) è composta da una stanza quadrata di lato  $3\text{ m}$  chiusa da quattro muri perimetrali di dimensioni  $(3, 0.1, 0.25)\text{ m}$ . La loro posizione tiene conto della larghezza del muro stesso, in modo da non togliere spazio all'arena. Per fare ciò, le loro coordinate sono state opportunamente estese aggiungendo un *offset* pari a  $(0.1 / 2)\text{ m} = 0.05\text{ m}$ :

$$\{(0, 1.55, 0), (0, -1.55, 0), (1.55, 0, 0), (-1.55, 0, 0)\}.$$

. Il robot è inizialmente posizionato alle coordinate  $(0, 0, 0)$ . Gli ostacoli presentano le dimensioni e posizioni indicate in figura 3.7.

Il robot è dotato di 12 sensori di prossimità, in grado di captare la vicinanza a eventuali ostacoli in direzione radiale rispetto al robot. Il simulatore mette a disposizione 24 sensori: in questo caso si è deciso di ridurli a gruppi di 2, usando come lettura finale di ognuno il valore massimo della coppia.

La funzione obiettivo utilizzata è definita come segue:

$$F_{obstacle} = \frac{100}{E} \cdot \sum_t (1 - p_{max}(t)) \cdot (1 - \sqrt{l(t) - r(t)}) \cdot \frac{l(t) + r(t)}{2} \quad (3.2)$$

dove:

- $E$  è il numero di step contenuti in un'epoca;
- $t$  è il generico step dell'epoca;
- $p_{max}(t)$  è il massimo valore dei sensori di prossimità calcolato all'istante  $t$ ;
- $l(t)$  indica se il motore sinistro è acceso (1) o spento (0);
- $r(t)$  indica se il motore destro è acceso (1) o spento (0).

Questa funzione cerca di valorizzare un movimento rettilineo del robot che avvenga il più lontano possibile dagli ostacoli presenti nell'ambiente.

### 3.3.3 Path following

Il terzo task consiste nel far seguire al robot un percorso tracciato sul pavimento.

posizione	dimensioni	rotazione
$(-1.35, 1.35, 0)$	$(0.15, 0.15, 0.15)$	$(0, 0, 0)$
$(1.35, 1.35, 0)$	$(0.15, 0.15, 0.15)$	$(0, 0, 0)$
$(-1.35, -1.35, 0)$	$(0.15, 0.15, 0.15)$	$(0, 0, 0)$
$(1.35, -1.35, 0)$	$(0.15, 0.15, 0.15)$	$(0, 0, 0)$
$(0, 1.35, 0)$	$(0.15, 0.15, 0.15)$	$(0, 0, 0)$
$(1.35, 0, 0)$	$(0.15, 0.15, 0.15)$	$(0, 0, 0)$
$(0, -1.35, 0)$	$(0.15, 0.15, 0.15)$	$(0, 0, 0)$
$(-1.35, 0, 0)$	$(0.15, 0.15, 0.15)$	$(0, 0, 0)$

(a)

posizione	dimensioni	rotazione
$(0.85, 0.85, 0)$	$(0.15, 0.75, 0.35)$	$(-45, 0, 0)$
$(0.85, -0.85, 0)$	$(0.15, 0.75, 0.35)$	$(0, 0, 0)$
$(-0.85, 0.85, 0)$	$(0.15, 0.75, 0.35)$	$(45, 0, 0)$
$(-0.85, -0.85, 0)$	$(0.15, 0.75, 0.35)$	$(0, 0, 0)$
$(0, 0.65, 0)$	$(0.15, 0.75, 0.35)$	$(0, 0, 0)$
$(-0.65, 0, 0)$	$(0.15, 0.75, 0.35)$	$(45, 0, 0)$
$(0.65, 0, 0)$	$(0.15, 0.75, 0.35)$	$(0, 0, 0)$
$(0, -0.65, 0)$	$(0.15, 0.75, 0.35)$	$(-45, 0, 0)$

(b)

Figura 3.7: Configurazione degli ostacoli nel task di *obstacle avoidance*.



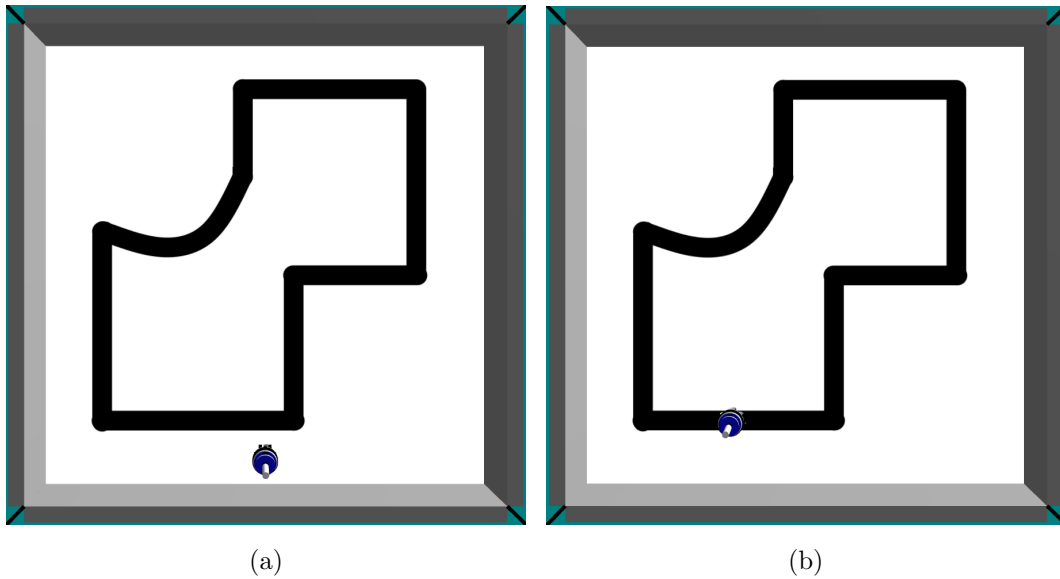


Figura 3.8: Ambiente usato per il task di *path following*. (a) Partenza fuori dal percorso. (b) Partenza sul percorso.

L'arena dell'esperimento (fig. 3.8) è composta da una stanza quadrata di lato  $3m$  chiusa da quattro muri perimetrali di dimensioni  $(3, 0.1, 0.25) m$ . La loro posizione tiene conto della larghezza del muro stesso, in modo da non togliere spazio all'arena. Per fare ciò, le loro coordinate sono state opportunamente estese aggiungendo un *offset* pari a  $(0.1 / 2)m = 0.05 m$ :

$$\{(0, 1.55, 0), (0, -1.55, 0), (1.55, 0, 0), (-1.55, 0, 0)\}.$$

. Il robot è inizialmente posizionato alle coordinate  $(0, -1.3, 0)$  e  $(-0.5, -1.05, 0)$ . Nel primo caso il robot parte fuori dal circuito, mentre nel secondo è già posizionato sopra di esso, orientato verso est in direzione del percorso. Questa duplice configurazione permette di condurre due esperimenti diversi, in modo da capire se le diverse condizioni iniziali possono avere qualche effetto visibile sulle prestazioni dei robot.

Il robot è dotato di 4 sensori di terreno, in grado di captare esclusivamente i colori nero e bianco. In questo caso essi sono posti lungo le direzioni cardinali del robot, per meglio adattarsi alla natura del task. Come nel caso del *task* di fototassi, il simulatore ne mette a disposizione 8, ma per ragioni di efficienza si è deciso di selezionare esclusivamente quelli di maggiore interesse.

La funzione obiettivo utilizzata è definita come segue:

$$F_{path} = \frac{100}{E} \cdot \sum_t \max^*(\overrightarrow{p(0), p(t)}) \cdot g(t) \cdot v(t) \quad (3.3)$$

dove:

- $E$  è il numero di step contenuti in un'epoca;
- $t$  è il generico step dell'epoca;
- $\max^*(\overrightarrow{p(0), p(t)})$  è la massima distanza dal punto di partenza dell'epoca raggiunta dal robot mentre si trova sul percorso;
- $g(t)$  vale 1 se almeno uno dei sensori di terreno sono attivi, 0 altrimenti;
- $v(t)$  vale 1 se almeno una dei due motori è acceso, 0 altrimenti.

Questa funzione cerca non solo di massimizzare il tempo trascorso sul percorso, ma anche di massimizzare la mobilità del robot, premiando maggiormente quelle reti che lo portano a spostarsi il più possibile lontano dal punto di partenza.

### 3.4 Fototassi assistita

In questa sezione presenteremo i parametri di simulazione, i risultati ottenuti e le relative conclusioni riguardanti il *task* di fototassi assistita.

### 3.4.1 Parametri

Le simulazioni sono state configurate con i seguenti parametri:

- 100 repliche;
- 50 epoche per replica;
- 2500 step per epoca;
- 10 step al secondo.

Le reti booleane sono state generate con i seguenti parametri:

- 100 nodi;
- $k = 3$  input per nodo;
- bias 0.79;
- 8 nodi di input;
- 2 nodi di output;
- 3 *swap* massimi dei nodi di input per epoca.

L'*empowerment* è stato calcolato con  $n = 1$  step e con una soglia di convergenza dell'algoritmo di Blahut-Arimoto pari a 5.

#### Binarizzazione della luce

Il sensore di luce utilizzato percepisce valori di intensità continui. Per poter essere usati al fine di perturbare la rete di controllo nei suoi nodi di input, è necessario trasformare questi stimoli in valori booleani. Per farlo è stata tarata e impostata una soglia di binarizzazione pari a 0.2.

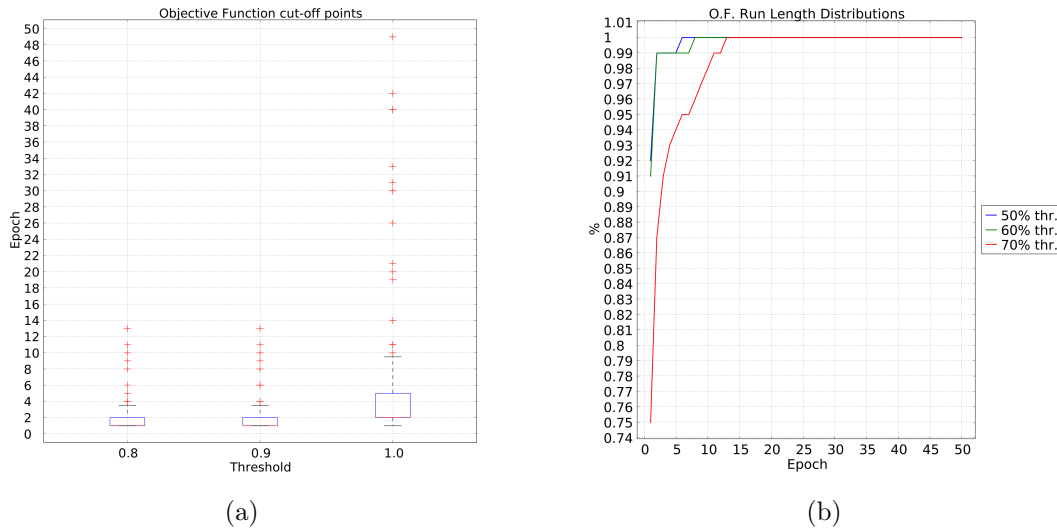


Figura 3.9: Risultato della fase *B* del task di fototassi assistita. (a) stima della soglia di cut-off. (b) RLD dell’*empowerment*.

### Cut-off

Tramite i grafici in fig. 3.9, ottenuti tramite la fase *B* del task è stato possibile stimare la soglia di *cut-off* da utilizzare per l’esperimento. È stato scelto un valore di 2 in quanto mediana della distribuzione calcolata con soglia 1.0. Conseguentemente, le repliche dell’esperimento sono state configurate con un numero di epoche pari a  $50 + 2 = 52$ .

## 3.4.2 Risultati

### Route

In figura 3.10 è possibile vedere le traiettorie dei robot appartenenti a due delle migliori repliche dell’esperimento.

Il task sembra essere eseguito correttamente; se in 3.10a il robot rimane da subito in equilibrio sotto la luce, in 3.10b ciò avviene lo stesso ma con un

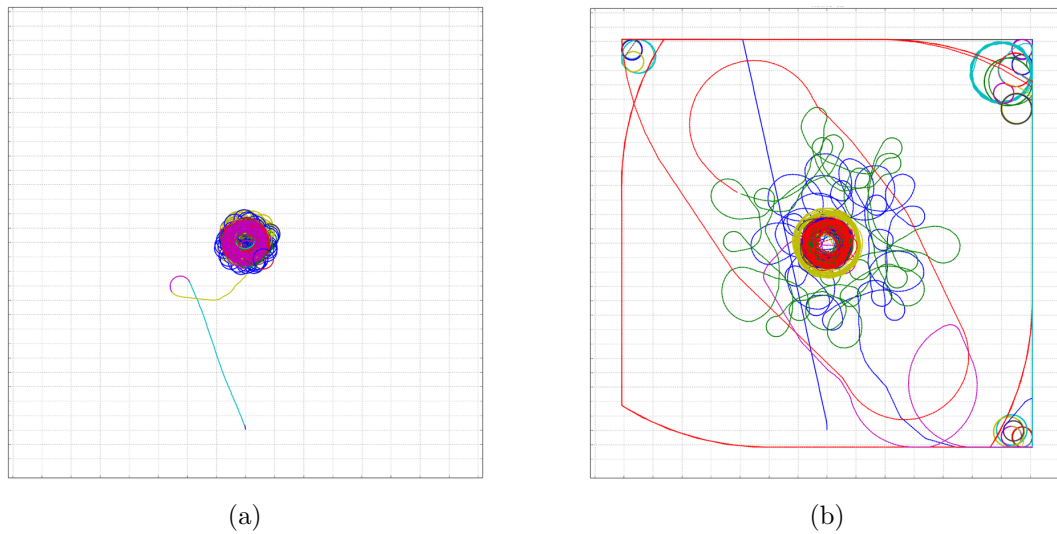


Figura 3.10: Route di alcune delle migliori reti pre-adattate nel task di fototassi assistita.

comportamento tendenzialmente più esplorativo.

### RLD funzione obiettivo

In figura 3.11 è possibile vedere i grafici che mettono a confronto le percentuali delle repliche migliori delle reti pre-adattate con *empowerment* e non. Fatta eccezione per le ultime 10 epoche a soglia 60%, è facile notare come le prestazioni delle reti della fase *A* tendano a dominare quelle delle reti pre-adattate con *empowerment*. Seppur lievemente peggiori, le reti pre-adattate presentano comunque un andamento comparabile con quello delle reti di controllo.

### Distribuzioni e test

In figura 3.12a si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* dopo la prima epoca (rete random) e dopo l'ultima epoca. È facile notare come la distribuzione finale sia nettamente superiore

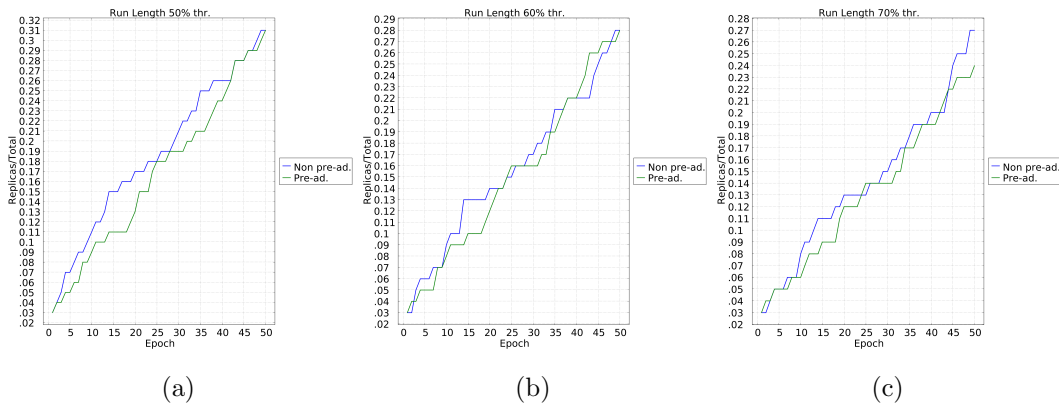


Figura 3.11: Confronto fra RLD della funzione obiettivo nel *task* di fototassi fra reti pre-adattate e non. Soglie: 50%, 60%, 70%.

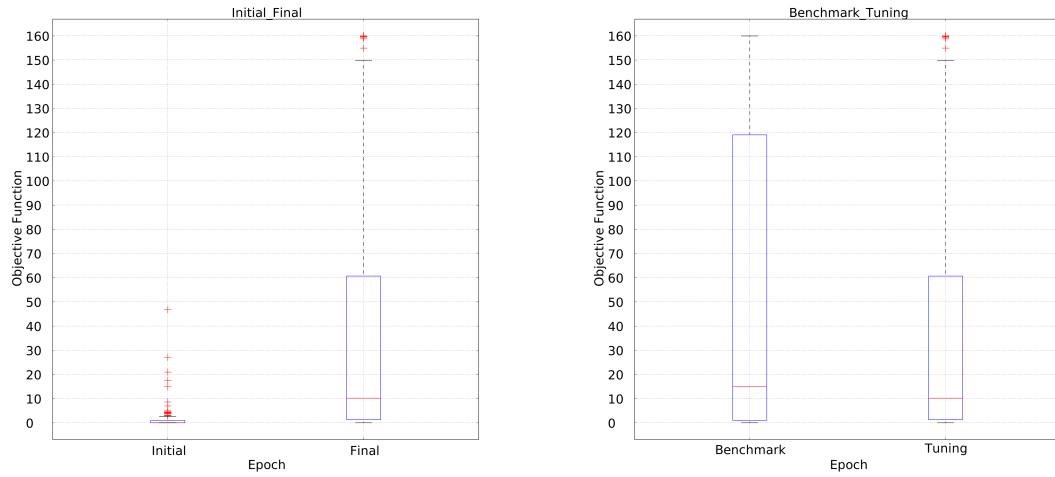
a quella iniziale. Questo ci indica chiaramente che utilizzare l’empowerment come funzione di adattamento per questo task permetta al robot di apprendere in qualche misura anche il task specifico.

In figura 3.12b si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* nell’ultima epoca delle fasi *A* e *B*. A prima vista non sembra che le reti adattate esclusivamente tramite *empowerment* presentino prestazioni maggiori rispetto alle reti di confronto.

In figura 3.12c si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* nell’ultima epoca della fase *A* e dell’esperimento. Non risulta che le reti pre-adattate tramite *empowerment* presentino prestazioni maggiori rispetto alle reti di confronto.

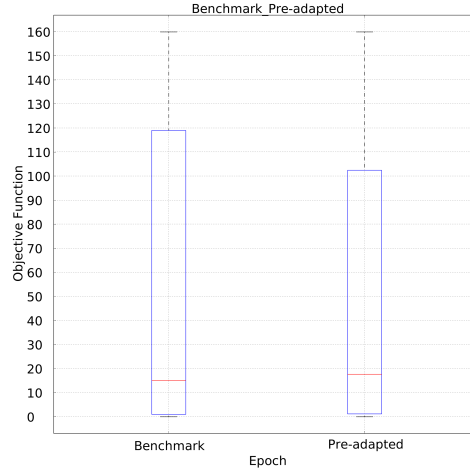
### 3.4.3 Conclusioni

Dai risultati ottenuti non è possibile confermare l’ipotesi secondo cui un pre-adattamento delle reti tramite *empowerment* favorisca le prestazioni delle reti nel task di fototassi, rispetto a reti adattate con una funzione obiettivo



(a) Funzione obiettivo nella fase  $B$  dopo la prima e l'ultima epoca.

(b) Funzione obiettivo dopo l'ultima epoca in fase  $A$  e  $B$ .



(c) Funzione obiettivo dopo l'ultima epoca della fase  $A$  e dell'esperimento.

Figura 3.12: Distribuzioni della funzione obiettivo nel task di fototassi assistita.

<i>confronto valori f.o.</i>	<i>p-value</i>	<i>rifiuta <math>H_0</math></i>
<i>B iniziale vs. B finale</i>	8.644272876859343E-15	sì
<i>A finale vs. B finale</i>	0.623340751865698	no
<i>A finale vs. esperimento finale</i>	0.9396219298232494	no

Tabella 3.1: Risultati del test di Wilcoxon per il task di fototassi assistita.

*task-specific*.

Tuttavia, si può confermare l'ipotesi per cui l'esecuzione di un processo di adattamento esclusivamente tramite *empowerment* causi un aumento delle prestazioni calcolate sullo specifico *task* di fototassi.

Per quanto riguarda le RLD, non risulta in generale che le reti pre-adattate con *empowerment* ottengano un successo maggiore nel tempo rispetto alle reti di confronto.

## 3.5 Obstacle avoidance

In questa sezione presenteremo i parametri di simulazione, i risultati ottenuti e le relative conclusioni riguardanti il *task* di *obstacle avoidance*.

### 3.5.1 Parametri

Le simulazioni sono state configurate con i seguenti parametri:

- 100 repliche;
- 50 epoche per replica;
- 2500 step per epoca;
- 10 step al secondo.



Le reti booleane sono state generate con i seguenti parametri:

- 100 nodi;
- $k = 3$  input per nodo;
- bias 0.79;
- 12 nodi di input;
- 2 nodi di output;
- 3 *swap* massimi dei nodi di input per epoca.

L'*empowerment* è stato calcolato con  $n = 1$  step e con una soglia di convergenza dell'algoritmo di Blahut-Arimoto pari a 5.

### **Binarizzazione dei valori di prossimità**

Come il sensore di luce, anche il sensore di prossimità genera valori continui che devono essere binarizzati per l'uso in una rete booleana. In questo caso non è stato necessario introdurre propriamente una soglia, in quanto il sensore dà valore 0 se non vede alcun ostacolo di fronte a sè; è bastato quindi codificare l'assenza di ostacoli con il valore 0 e la percezione di essi – a distanza massima di 10 cm – con il valore 1.

### **Cut-off**

Tramite i grafici in fig. 3.13, ottenuti tramite la fase *B* è stato possibile stimare la soglia di *cut-off* da utilizzare per l'esperimento. È stato scelto un valore di 2 in quanto mediana della distribuzione calcolata con soglia 1.0. Conseguentemente, le repliche dell'esperimento sono state configurate con un numero di epoche pari a  $50 + 2 = 52$ .

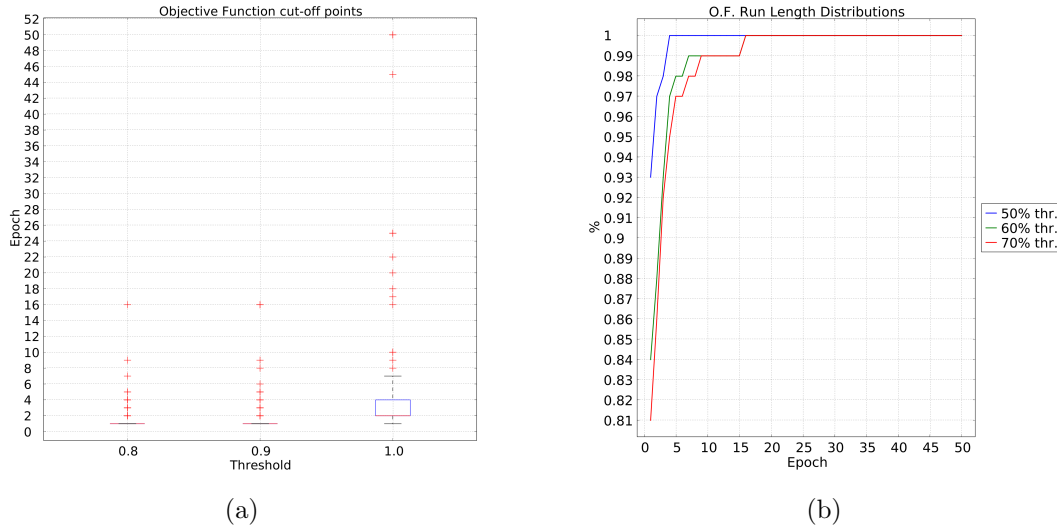


Figura 3.13: Risultato della fase *B* del task di *obstacle avoidance*. (a) stima della soglia di cut-off. (b) RLD dell'*empowerment*.

### 3.5.2 Risultati

#### Route

In figura 3.14 è possibile vedere le traiettorie dei robot appartenenti a due delle migliori repliche della fase dell'esperimento.

Il task sembra essere eseguito correttamente. In 3.14a il robot tende a circumnavigare gli ostacoli senza sbattervi mai contro, e tende a preferire un movimento rettilineo ogni volta che può. In 3.14b si nota invece come il robot tenda a non esplorare l'ambiente, restando stabilmente in mezzo agli ostacoli per quasi tutta la durata della simulazione.

#### RLD funzione obiettivo

In figura 3.15 è possibile vedere i grafici che mettono a confronto le percentuali delle repliche migliori delle reti pre-adattate con *empowerment* e non. Tendenzialmente le prestazioni delle reti di controllo tendono a dominare quel-

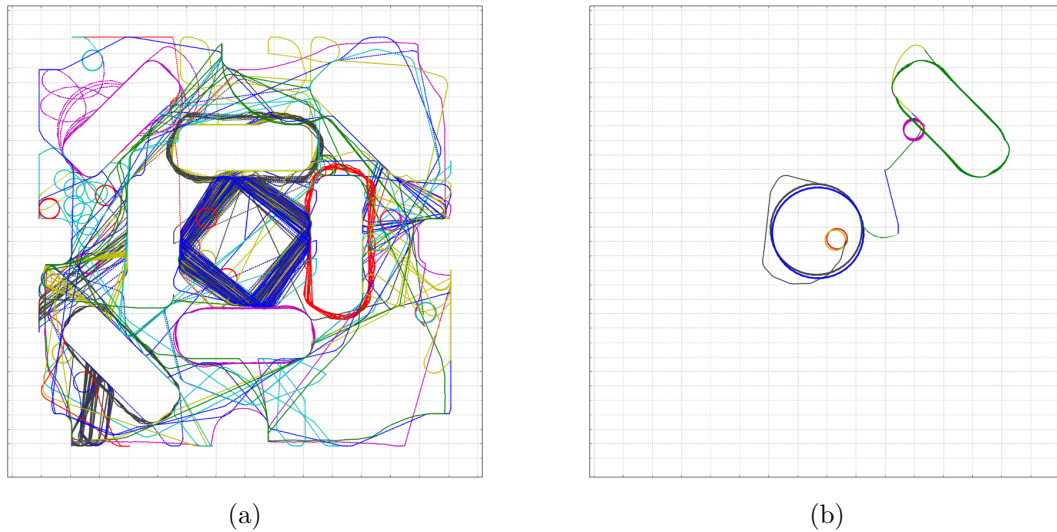


Figura 3.14: Route di alcune delle migliori reti pre-adattate nel task di *obstacle avoidance*.

le delle reti pre-adattate con *empowerment*. Si nota che tale affermazione pare non valere per il caso della soglia a 50%, in cui prevalgono le reti pre-adattate dall'epoca 35 in poi.

### Distribuzioni e test

In figura 3.16a si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* dopo la prima epoca (rete random) e dopo l'ultima epoca. È facile notare come la distribuzione finale sia nettamente superiore a quella iniziale. Questo ci indica chiaramente che utilizzare l'empowerment come funzione di adattamento per questo task permetta al robot di apprendere in qualche misura anche il task specifico.

In figura 3.16b si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* nell'ultima epoca delle fasi *A* e *B*. A prima vista non sembra che le reti adattate esclusivamente tramite *empowerment* presentino

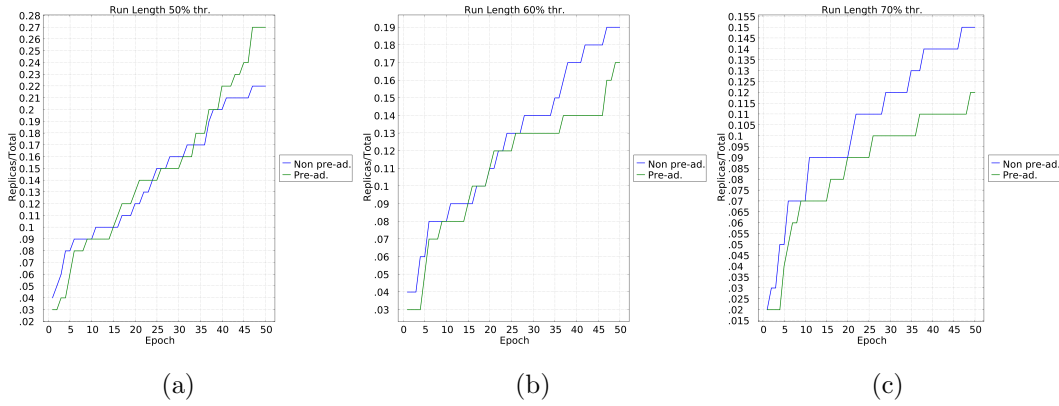


Figura 3.15: Confronto fra RLD della funzione obiettivo nel *task* di *obstacle avoidance* fra reti pre-adattate e non. Soglie: 50%, 60%, 70%.

<i>confronto valori f.o.</i>	<i>p-value</i>	<i>rifiuta <math>H_0</math></i>
<i>B</i> iniziale vs. <i>B</i> finale	5.407840620078627E-21	sì
<i>A</i> finale vs. <i>B</i> finale	0.5526836839704573	no
<i>A</i> finale vs. esperimento finale	0.540492162146446	no

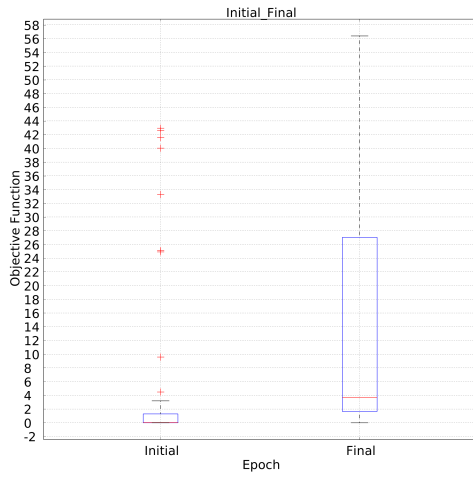
Tabella 3.2: Risultati del test di Wilcoxon per il *task* di *obstacle avoidance*.

prestazioni maggiori rispetto alle reti di confronto.

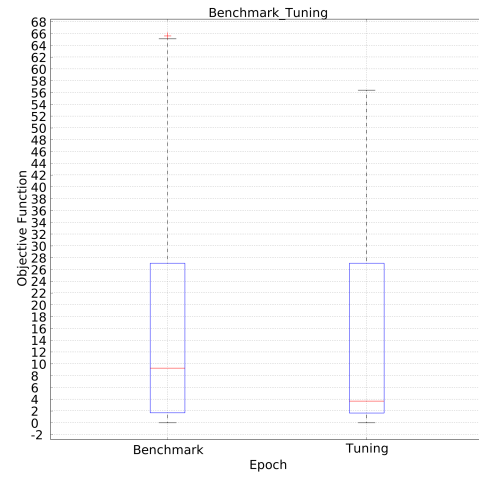
In figura 3.16c si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* nell'ultima epoca delle fasi *A* e *B*. Non risulta che le reti pre-adattate tramite *empowerment* presentino prestazioni maggiori rispetto alle reti di confronto.

### 3.5.3 Conclusioni

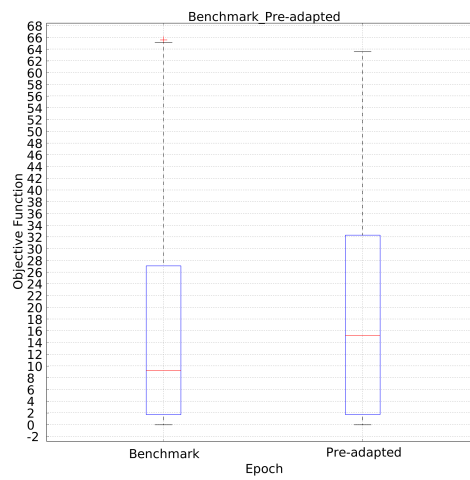
Dai risultati ottenuti non è possibile confermare l'ipotesi secondo cui un pre-adattamento delle reti tramite *empowerment* favorisca le prestazioni delle reti nel *task* di *obstacle avoidance*, rispetto a reti adattate con una funzione



(a) Funzione obiettivo nella fase  $B$  dopo la prima e l'ultima epoca.



(b) Funzione obiettivo dopo l'ultima epoca nelle fasi  $A$  e  $B$ .



(c) Funzione obiettivo dopo l'ultima epoca nella fase  $A$  e dell'esperimento.

Figura 3.16: Distribuzioni della funzione obiettivo nel task di *obstacle avoidance*.

obiettivo *task-specific*.

Tuttavia, si può confermare l'ipotesi per cui l'esecuzione di un processo di adattamento esclusivamente tramite *empowerment* causi un aumento delle prestazioni calcolate sullo specifico *task* di *obstacle avoidance*.

Per quanto riguarda le RLD, non risulta in generale che le reti pre-adattate con *empowerment* ottengano un successo maggiore nel tempo rispetto alle reti di confronto, in particolare considerando le soglie più alte.

## 3.6 Path following con partenza fuori percorso

In questa sezione presenteremo i parametri di simulazione, i risultati ottenuti e le relative conclusioni riguardanti il *task* di *path following* con partenza del robot fuori dal percorso tracciato.

### 3.6.1 Parametri

Le simulazioni sono state configurate con i seguenti parametri:

- 100 repliche;
- 50 epoche per replica;
- 2500 step per epoca;
- 10 step al secondo.

Le reti booleane sono state generate con i seguenti parametri:

- 100 nodi;
- $k = 3$  input per nodo;
- bias 0.79;

- 4 nodi di input;
- 2 nodi di output;
- 3 *swap* massimi dei nodi di input per epoca.

L'*empowerment* è stato calcolato con  $n = 1$  step e con una soglia di convergenza dell'algoritmo di Blahut-Arimoto pari a 5.

### Cut-off

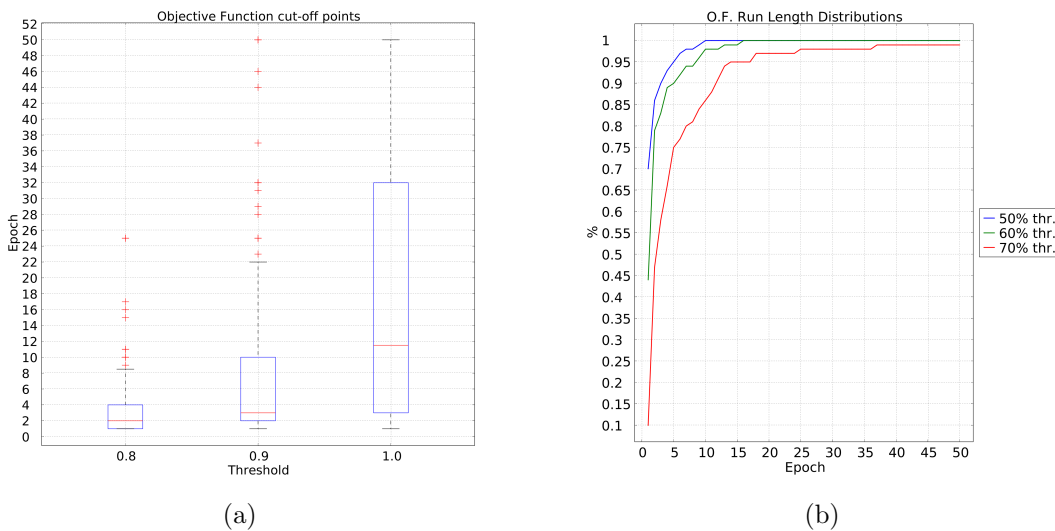


Figura 3.17: Risultato della fase *B* del task di *path following* con partenza fuori dal percorso. (a) stima della soglia di cut-off. (b) RLD dell'*empowerment*.

Tramite i grafici in fig. 3.17, ottenuti tramite la fase *B* è stato possibile stimare la soglia di *cut-off* da utilizzare per l'esperimento. È stato scelto un valore di 12 in quanto mediana della distribuzione calcolata con soglia 1.0. Conseguentemente, le repliche dell'esperimento sono state configurate con un numero di epoche pari a  $50 + 12 = 62$ .

### 3.6.2 Risultati

#### Route

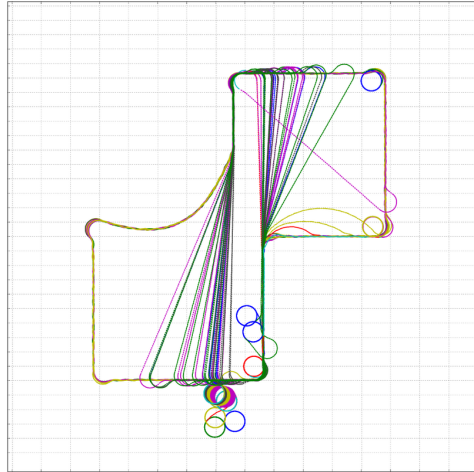


Figura 3.18: Route della migliore rete pre-adattata nel task di *path following* con partenza fuori dal percorso.

In figura 3.18 è possibile vedere la traiettoria del robot appartenente alla migliore replica dell'esperimento.

Il task sembra essere eseguito correttamente seguendo il percorso tracciato sul terreno, anche se ci sono diverse epoche in cui il robot pare “tagliare” il circuito, probabilmente per ottimizzare la massimizzazione della distanza percorsa nell'unità di tempo.

#### RLD funzione obiettivo

In figura 3.19 è possibile vedere i grafici che mettono a confronto le percentuali delle repliche migliori delle reti pre-adattate con *empowerment* e non. È interessante notare come le prestazioni delle reti pre-adattate tendano a convergere prima delle reti di controllo sia a soglia 50% che a soglia 70%, circa 17 epoche dopo.



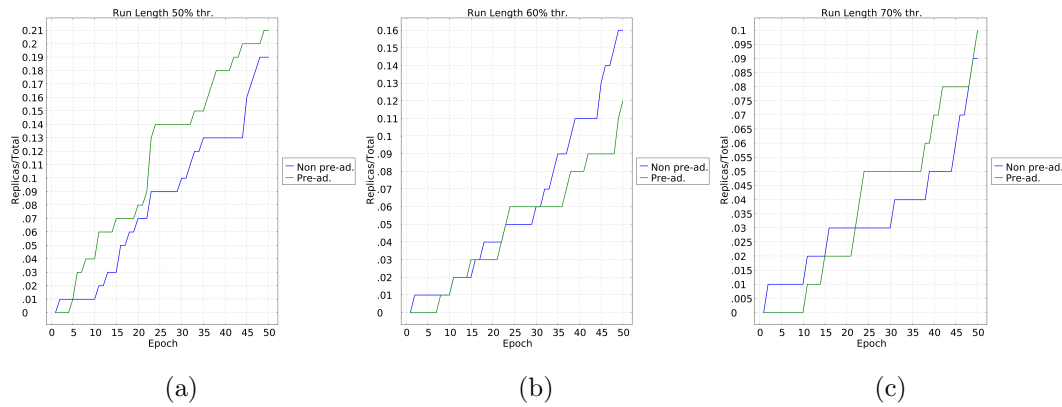


Figura 3.19: Confronto fra RLD della funzione obiettivo nel *task* di *path following* (fuori percorso) fra reti pre-adattate e non. Soglie: 50%, 60%, 70%.

### Distribuzioni e test

In figura 3.20a si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* dopo la prima epoca (rete random) e dopo l'ultima epoca. È facile notare come la distribuzione finale sia nettamente superiore a quella iniziale. Questo ci indica chiaramente che utilizzare l'empowerment come funzione di adattamento per questo task permetta al robot di apprendere in qualche misura anche il task specifico.

In figura 3.20b si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* nell'ultima epoca delle fasi *A* e *B*. A prima vista non sembra che le reti adattate esclusivamente tramite *empowerment* presentino prestazioni maggiori rispetto alle reti di confronto.

In figura 3.20c si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* nell'ultima epoca delle fasi *A* e *B*. A prima vista sembra che le reti pre-adattate tramite *empowerment* presentino prestazioni maggiori rispetto alle reti di confronto. Tuttavia questo risultato non è statisticamente significativo, come mostrato in tabella 3.3, sebbene il p-value ottenuto risulti

<i>confronto valori f.o.</i>	<i>p-value</i>	<i>rifiuta <math>H_0</math></i>
<i>B iniziale vs. B finale</i>	1.148726662389784E-17	sì
<i>A finale vs. B finale</i>	0.2694145543785734	no
<i>A finale vs. esperimento finale</i>	0.3296038307977137	no

Tabella 3.3: Risultati del test di Wilcoxon per il task di *path following* con partenza fuori dal percorso.

comunque più basso rispetto agli stessi risultati degli esperimenti precedenti.

### 3.6.3 Conclusioni

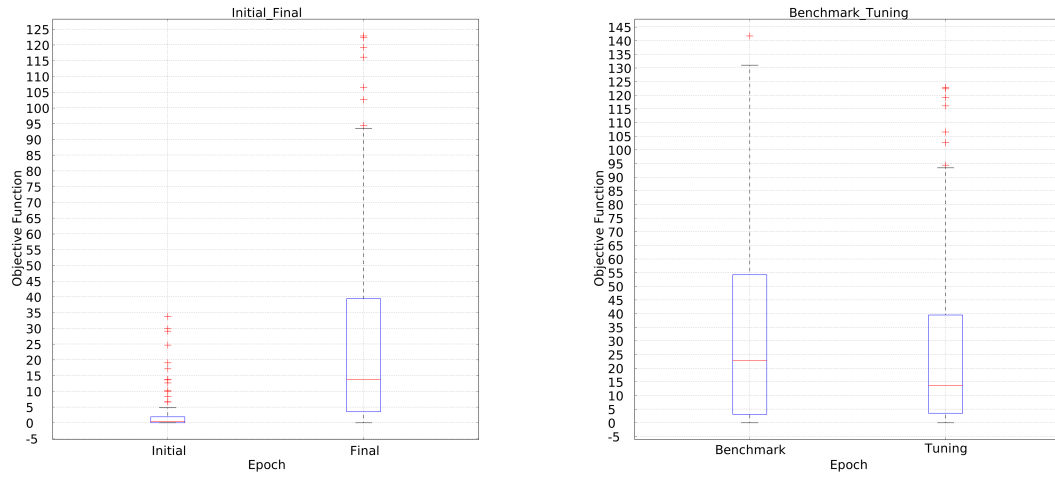
Dai risultati ottenuti non è possibile confermare l'ipotesi secondo cui un pre-adattamento delle reti tramite *empowerment* favorisca le prestazioni delle reti nel task di *path following*, rispetto a reti adattate con una funzione obiettivo *task-specific*.

Tuttavia, si può confermare l'ipotesi per cui l'esecuzione di un processo di adattamento esclusivamente tramite *empowerment* causi un aumento delle prestazioni calcolate sullo specifico *task* di *path following*.

Per quanto riguarda le RLD, è possibile in questo caso che le reti pre-adattate presentino percentuali di successo superiori rispetto alle reti di confronto, senza tuttavia dominarle sistematicamente.

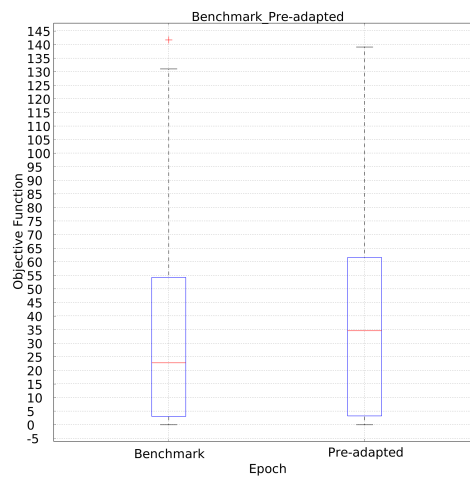
## 3.7 Path following con partenza sul percorso

In questa sezione presenteremo i parametri di simulazione, i risultati ottenuti e le relative conclusioni riguardanti il *task* di *path following* con partenza del robot sul percorso tracciato.



(a) Funzione obiettivo nella fase  $B$  dopo la prima e l'ultima epoca.

(b) Funzione obiettivo dopo l'ultima epoca nelle fasi  $A$  e  $B$ .



(c) Funzione obiettivo dopo l'ultima epoca nella fase  $A$  e dell'esperimento.

Figura 3.20: Distribuzioni della funzione obiettivo nel task di *path following* con partenza fuori dal percorso.

### 3.7.1 Parametri

Le simulazioni sono state configurate con i seguenti parametri:

- 100 repliche;
- 50 epoche per replica;
- 2500 step per epoca;
- 10 step al secondo.

Le reti booleane sono state generate con i seguenti parametri:

- 100 nodi;
- $k = 3$  input per nodo;
- bias 0.79;
- 4 nodi di input;
- 2 nodi di output;
- 3 *swap* massimi dei nodi di input per epoca.

L'*empowerment* è stato calcolato con  $n = 1$  step e con una soglia di convergenza dell'algoritmo di Blahut-Arimoto pari a 5.

#### Cut-off

Tramite i grafici in fig. 3.21, ottenuti tramite la fase *B* del task è stato possibile stimare la soglia di *cut-off* da utilizzare per l'esperimento. È stato scelto un valore di 20 in quanto mediana della distribuzione calcolata con soglia 1.0. Conseguentemente, le repliche dell'esperimento sono state configurate con un numero di epoche pari a  $50 + 20 = 70$ .

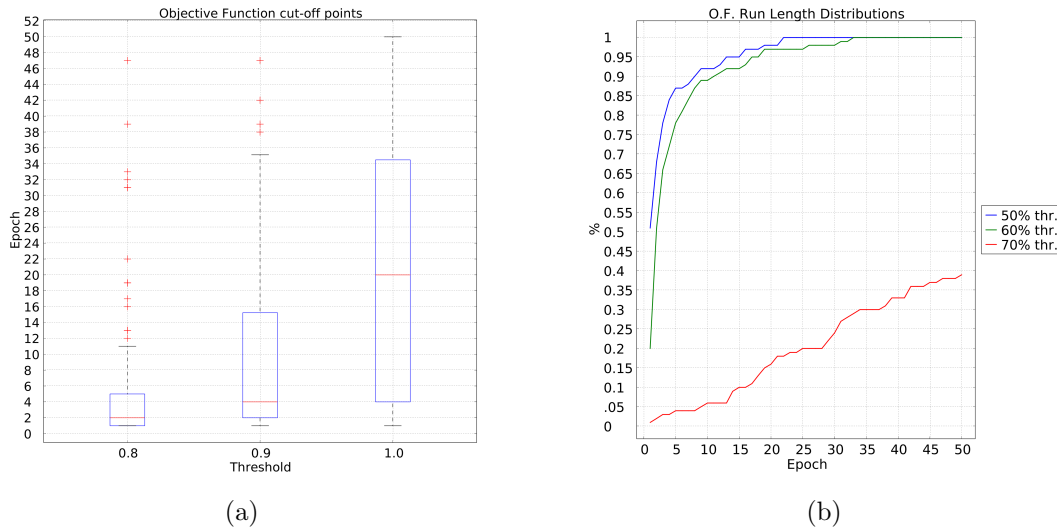


Figura 3.21: Risultato della fase *B* del task di *path following* con partenza sul percorso. (a) stima della soglia di cut-off. (b) RLD dell'*empowerment*.

### 3.7.2 Risultati

#### Route

In figura 3.22 è possibile vedere le traiettorie del robot appartenente alla migliore replica dell'esperimento.

Non è facile valutare qualitativamente la strategia messa in atto dal robot. Tuttavia, è plausibile che l'agente abbia imparato un comportamento dalle caratteristiche simili a quello mostrato nell'esperimento precedente in figura 3.18, per cui la rete potrebbe riuscire a ottimizzare il compito intersecando il percorso in punti opposti – o comunque lontani – massimizzando quindi le distanze percorse a ogni epoca.

#### RLD funzione obiettivo

In figura 3.23 è possibile vedere i grafici che mettono a confronto le percentuali delle repliche migliori delle reti pre-adattate con *empowerment* e non. Si

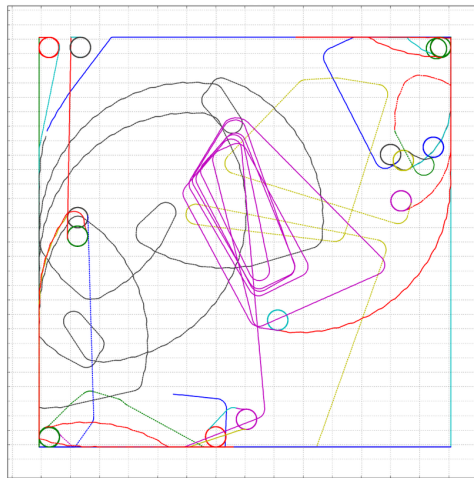


Figura 3.22: Route della migliore rete pre-adattata nel task di *path following* con partenza sul percorso.

può vedere come almeno da metà simulazione le reti convergono più velocemente in caso di pre-adattamento tramite *empowerment*.

### Distribuzioni e test

In figura 3.24a si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* dopo la prima epoca (rete random) e dopo l'ultima epoca. È facile notare come la distribuzione finale sia nettamente superiore a quella iniziale. Questo ci indica chiaramente che utilizzare l'*empowerment* come funzione di adattamento per questo task permetta al robot di apprendere in qualche misura anche il task specifico.

In figura 3.24b si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* nell'ultima epoca delle fasi *A* e *B*. A prima vista non sembra che le reti adattate esclusivamente tramite *empowerment* presentino prestazioni maggiori rispetto alle reti di confronto.

In figura 3.24c si possono vedere le distribuzioni dei valori della funzione obiettivo *task-specific* nell'ultima epoca delle fasi *A* e *B*. Non risulta che le reti

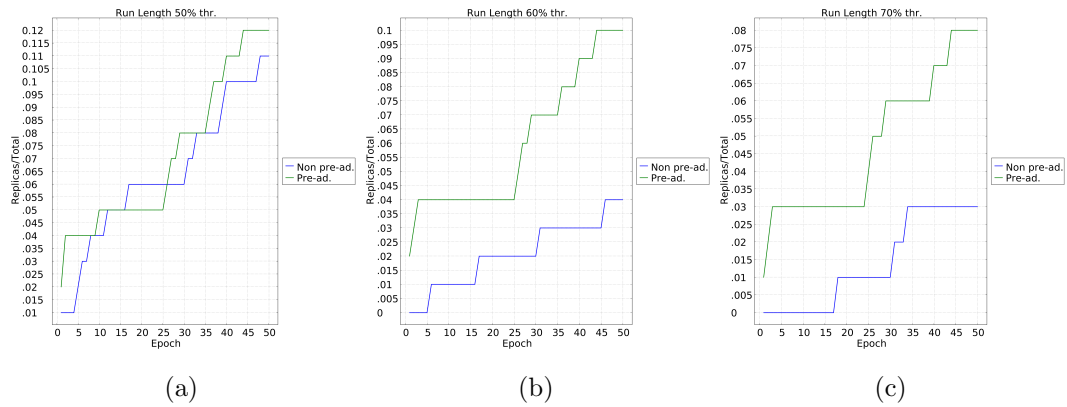


Figura 3.23: Confronto fra RLD della funzione obiettivo nel *task* di *path following* (sul percorso) fra reti pre-adattate e non. Soglie: 50%, 60%, 70%.

<i>confronto valori f.o.</i>	<i>p-value</i>	<i>rifiuta <math>H_0</math></i>
<i>B</i> iniziale vs. <i>B</i> finale	7.626122928016982E-15	sì
<i>A</i> finale vs. <i>B</i> finale	0.500072721112088	no
<i>A</i> finale vs. esperimento finale	0.4732910246070986	no

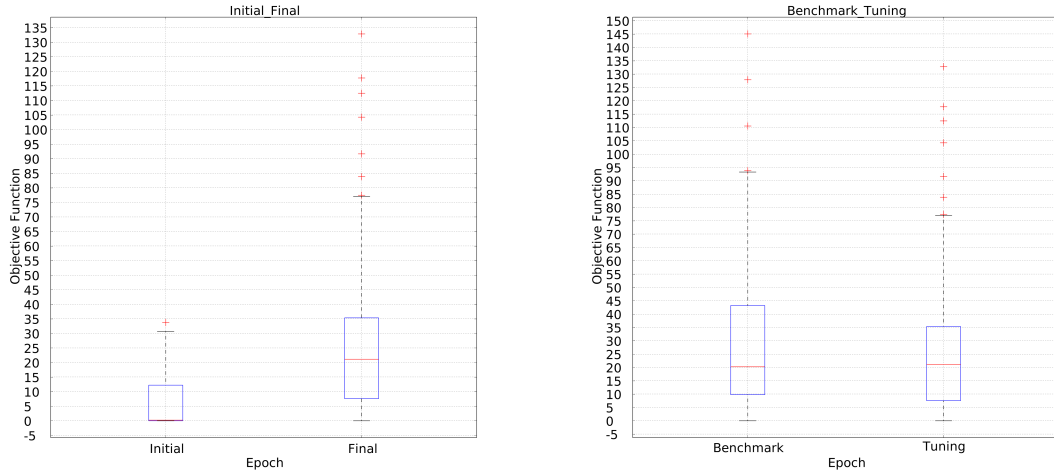
Tabella 3.4: Risultati del test di Wilcoxon per il *task* di *path following* con partenza sul percorso.

pre-adattate tramite *empowerment* presentino prestazioni maggiori rispetto alle reti di confronto.

### 3.7.3 Conclusioni

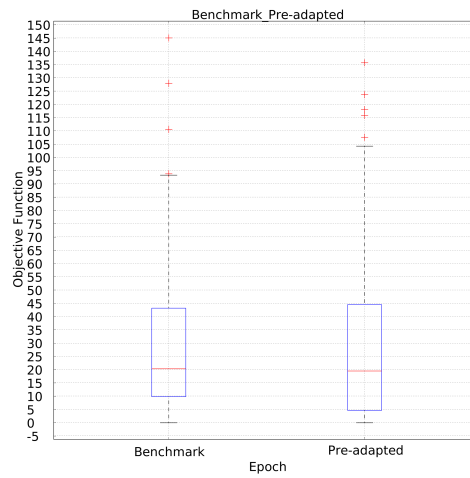
Dai risultati ottenuti non è possibile confermare l'ipotesi secondo cui un pre-adattamento delle reti tramite *empowerment* favorisca le prestazioni delle reti nel *task* di *path following*, rispetto a reti adattate con una funzione obiettivo *task-specific*.

Tuttavia, si può confermare l'ipotesi per cui l'esecuzione di un processo



(a) Funzione obiettivo nella fase  $B$  dopo la prima e l'ultima epoca.

(b) Funzione obiettivo dopo l'ultima epoca nelle fasi  $A$  e  $B$ .



(c) Funzione obiettivo dopo l'ultima epoca nelle fasi  $A$  e  $B$ .

Figura 3.24: Distribuzioni della funzione obiettivo nel task di *path following* con partenza sul percorso.



di adattamento esclusivamente tramite *empowerment* causi un aumento delle prestazioni calcolate sullo specifico *task* di *path following*.

Per quanto riguarda le RLD, è visibile in questo caso come le reti pre-adattate possano ottenere percentuali di successo visibilmente maggiori rispetto alle reti di confronto. Nelle soglie più alte, l'andamento delle reti pre-adattate risulta dominare visibilmente quello delle reti di controllo della fase *A*; ciò lascia presupporre come questo impiego dell'*empowerment* potrebbe effettivamente aiutare i robot nel compimento di questo tipo di task, sicuramente più complessi di quelli precedentemente mostrati. È altresì possibile che il pre-adattamento abbia favorito l'emergere di un comportamenti maggiormente "creativi", come è possibile vedere dal percorso in figura 3.22.



# Conclusioni e lavori futuri

In questo lavoro di tesi sono state svolte delle indagini in merito alla fattibilità dell'utilizzo dell'*empowerment* nell'adattamento online di robot controllati da reti booleane critiche. In particolare, ci si è chiesti se l'utilizzo dell'*empowerment* come fattore di merito possa causare aumenti significativi di una funzione obiettivo specifica per ogni task; se l'utilizzo dell'*empowerment* come fattore di merito possa permettere a un robot di svolgere un certo *task* con più efficacia rispetto a un robot adattato con una funzione obiettivo specifica; se effettuare un pre-adattamento dei robot tramite *empowerment* potesse indurre prestazioni migliori rispetto a robot addestrati esclusivamente tramite una funzione obiettivo specifica.

Sono stati definiti i *task* di fototassi assistita, di *obstacle avoidance* e di *path following*. Su di essi sono state adattati robot controllati da reti booleane critiche con parametri  $\{N = 100, K = 3, p = 0.79\}$ . Per le fasi sperimentali di ogni *task* sono state lanciate 100 repliche dell'esperimento: oltre all'esperimento vero e proprio costituito da una fase di pre-adattamento tramite *empowerment* e successivo adattamento con funzione obiettivo specifica, sono state eseguite due fasi preliminari mirate a ottenere dati con i quali confrontare i risultati dell'esperimento, e al calcolo del periodo di pre-adattamento. I risultati ottenuti sono stati analizzati secondo due modalità principali: *box-plot* e *run length distribution* dei valori delle funzioni obiettivo. Nel primo caso, i risultati sono

stati validati tramite test non-parametrici di Wilcoxon.

Per ogni *task* simulato si è dimostrato che utilizzare l'*empowerment* come funzione obiettivo causa aumenti significativi della funzione obiettivo specifica per il task. Tuttavia, non è stato possibile affermare che i livelli di merito raggiunti superino significativamente i valori risultanti da un adattamento con funzione obiettivo specifica. Non è stato possibile, inoltre, affermare che il processo di pre-adattamento porti a valori di merito superiori a quelli raggiunti da robot non pre-adattati.

Nondimeno, è interessante notare come le reti pre-adattate tendano a mostrare percentuali di successo più alte delle reti non pre-adattate nei *task* di path following. Questo risultato, sebbene non significativo, permette di formulare ipotesi sul fatto che il processo di pre-adattamento potrebbe risultare più efficace nell'esecuzione di compiti più complessi, caratterizzati da interazioni più marcate e specifiche con l'ambiente.

Sicuramente i lavori di ricerca futuri potranno esplorare ulteriormente queste considerazioni: in primis valutando altre combinazioni di parametri critici, ed eventualmente aumentando il numero di repliche eseguite, in modo da ottenere risultati maggiormente significativi. Inoltre, si potrebbe valutare l'impatto dell'*empowerment* in compiti più complessi, o in ambienti dinamici. Infine, sarebbe opportuno misurare l'influenza dell'*empowerment* su alcune misure di complessità, quali *mutual information*, *predictive information* e *transfer entropy*.

# Ringraziamenti

Arrivare fin qui è stato più complicato del previsto. Se ci sono arrivato, se ho superato sfide che sembravano inaffrontabili, se ho risolto problemi all'apparenza intrattabili, lo devo a un grande numero di persone. Di seguito ne riporto solo alcune, quelle a cui sento di dovere qualcosa di più. Una parte sicuramente minoritaria, ma estremamente necessaria; le persone rimanenti, ne sono convinto, sanno comunque di condividere con me qualcosa di speciale.

A Lorenzo, per un'amicizia che dura da una vita;

a Mirko, per avermi insegnato che di fronte a un problema una soluzione si trova sempre;

ad Anita, per avermi insegnato ad essere sincero e a prendermi cura di me;

a Shapour e Ruben<sup>1</sup>, per avermi convinto del fatto che cenare prima delle undici resta tutto sommato una pratica sopravvalutata;

a Perla e Francesca, per non avermi mai fatto mancare un momento di conforto;

a Manuel e Luca, per aver condiviso con me gran parte di questo percorso;

a Giovanni, per la sua guida e per avermi trasmesso una sincera passione per la scienza;

a tutte le persone che hanno vissuto e vivono tuttora S.P.R.I.Te., per avermi sempre fatto sentire a casa, e per avermi permesso di non annoiarmi mai;

---

<sup>1</sup>Nice.

alla mia famiglia, per avermi sempre dato fiducia;  
Vi voglio bene. Grazie.

# Bibliografia

- [1] The swarmanoid project. [https://www.swarmanoid.org/swarmanoid\\_hardware.php.html](https://www.swarmanoid.org/swarmanoid_hardware.php.html).
- [2] Pedro Abreu, Gianluca Antonelli, Filippo Arrichiello, Andrea Caffaz, Andrea Caiti, Giuseppe Casalino, Nicola Catenacci Volpi, Ivan Bielic De Jong, Daniela De Palma, Henrique Duarte, et al. Widely scalable mobile underwater sonar technology: An overview of the h2020 wimust project. *Marine Technology Society Journal*, 50(4):42–53, 2016.
- [3] M Aldana, S Coppersmith, and LP Kadanoff. Perspectives and problems in nonlinear science. *Springer, New York*, 2003.
- [4] Suguru Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.
- [5] Richard Blahut. Computation of channel capacity and rate-distortion functions. *IEEE transactions on Information Theory*, 18(4):460–473, 1972.
- [6] Michele Braccini. Applications of biological cell models in robotics. *arXiv preprint arXiv:1712.02303*, 2017.

- 
- [7] Michele Braccini, Andrea Roli, and Stuart A. Kauffman. Online adaptation in robots as biological development provides phenotypic plasticity, 2020.
- [8] James J Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.
- [9] Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.
- [10] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pages 128–135. IEEE, 2005.
- [11] Carlo Pinciroli, Vito Trianni, Rehan O’Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, et al. Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence*, 6(4):271–295, 2012.
- [12] Andrea Roli and Michele Braccini. Attractor landscape: A bridge between robotics and synthetic biology. *cell differentiation*, 4:5, 2018.
- [13] Andrea Roli, Michele Braccini, and Edoardo Barbieri. Adattamento online di robot controllati da reti booleane.
- [14] Andrea Roli, Marco Villani, Alessandro Filisetti, and Roberto Serra. Dynamical criticality: overview and open questions, 2016.



- [15] Dari Trendafilov, Roderick Murray-Smith, and Daniel Polani. Empowerment as a metric for optimization in hci. 2015.