

**ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI INGEGNERIA E ARCHITETTURA**

**Corso di Laurea in
Ingegneria Elettronica, Informatica e Telecomunicazioni**

Apprendimento e Self-Awareness in robotica

**Elaborato in
Fondamenti di Informatica B**

**Relatore:
ANDREA ROLI**

**Presentato da:
MATTEO DELVECCHIO**

**II Sessione
Anno Accademico 2013/2014**

Introduzione

Al giorno d'oggi le macchine robotiche sono perlopiù strumenti dalle funzionalità prestabilite, codificate in qualche modo al loro interno e immutabili, tanto che nel parlato comune l'espressione "robot" viene spesso utilizzata in modo ironico, in riferimento ad una persona, per indicare rigidità mentale e incapacità di estendere i propri ragionamenti, evidenziando una sorta di limite. Questo limite in effetti è molto spesso presente in queste macchine, le quali peccano di elasticità, adattabilità e capacità di deduzione. Per queste ragioni, è nata la disciplina dell'intelligenza artificiale (*Artificial Intelligence*, AI), che comprende la teoria e le tecniche per lo sviluppo di algoritmi che consentono alle macchine (tipicamente ai calcolatori) di mostrare un'abilità e/o un'attività intelligente (perlomeno in specifici domini), e quindi di svolgere funzioni e ragionamenti tipici della mente umana.

Ovviamente questo campo di studio ha un forte legame con la nozione di apprendimento, di fondamentale importanza al fine di maturare un'intelligenza: l'uomo ad esempio matura le sue capacità cognitive proprio attraverso l'esperienza e la conoscenza continua.

L'elaborato quindi si premura di identificare gli aspetti cardinali dell'apprendimento in robotica, elencando i principali metodi, approcci e requisiti. Successivamente l'attenzione si sposta sul concetto di *consapevolezza di sé*, una tipologia di apprendimento a carattere ontologico, evidenziando le principali tecniche di sintesi dei modelli, e di inferenza delle caratteristiche del robot. Infine vengono esposti concetti derivanti dalla consapevolezza di sé, supportati da esempi applicativi di interesse.

A livello personale ritengo che lo studio di queste dinamiche sia particolarmente interessante poiché unisce i campi di robotica e di informatica con le scienze filosofiche: si va dalla gnoseologia, all'ontologia, fino all'epistemologia, discipline storicamente legate all'uomo, che studiano i fondamenti e i limiti della conoscenza, le relazioni tra soggetto e oggetto della conoscenza, e le condizioni e i metodi per il raggiungimento di una conoscenza scientifica. Il raggiungimento di una sorta di intelligenza robotica, in grado di manifestare una qualche forma di pensiero e di capacità cognitiva è indubbiamente affascinante, e permette anche di apprendere qualcosa in più su noi stessi, sul nostro modo di ragionare e sulla forte importanza che l'apprendimento riveste in quanto fattore di crescita intellettuale.

Indice

Introduzione	i
1 Apprendimento: aspetti chiave	1
1.1 Caratteri generali e definizione formale	2
1.2 Paradigmi e algoritmi di Apprendimento	3
1.3 Approcci	5
1.4 Requisiti dell'apprendimento robotico	9
2 Self-Awareness	11
2.1 Generalità	11
2.2 Il processo di sintesi automatica e l'algoritmo EEA	14
2.2.1 Modelli multipli, e test intelligente	15
2.2.2 I passi dell'algoritmo di stima ed esplorazione	17
2.3 Inferenza Parametrica	19
2.4 Inferenza Topologica	22
2.5 Osservazioni	25
3 Concetti e Applicazioni derivanti dalla Self Awareness	27
3.1 Il <i>Continuous Self Modelling</i> : generalità	27
3.2 L'algoritmo di stima ed esplorazione: autodiagnosi e recupero	28
3.2.1 Metodologia	29
3.2.2 Risultati	31
3.3 Self-Reflection	32
3.3.1 Applicazione in ambito di autodiagnosi e recupero . . .	32

Bibliografia

41

Elenco delle figure

1.1	Struttura di una semplice rete neurale artificiale.	6
2.1	Lo starfish robot utilizzato nella dimostrazione.	12
2.2	Pattern di locomozione creato dal robot (simulato), e pattern di locomozione del robot fisico.	13
2.3	Esempio delle due definizioni complementari di test intelligente.	16
2.4	Morfologia del robot e rete neurale.	20
2.5	Risultato di tre diverse esecuzioni, utilizzando configurazioni di sensori differenti.	24
3.1	Flusso di esecuzione degli algoritmi di esplorazione e di stima, e confronto con il flusso degli algoritmi convenzionali.	29
3.2	Architettura a cervelli annidati.	33
3.3	La catena di cervelli di Minsky.	34
3.4	Diagramma di flusso del metodo che permette di ottenere un robot dal comportamento elastico attraverso autodiagnosi, adattamento e recupero del controllore.	35

Capitolo 1

Apprendimento: aspetti chiave

Il concetto di apprendimento, inteso nella sua forma più generale, consiste nell'acquisizione o nella modifica di conoscenze, comportamenti, abilità, valori o preferenze. È una capacità degli esseri umani, degli animali, delle piante e di alcune macchine (ossia di alcuni robot): è scopo di questo capitolo presentare ed approfondire quest'ultima declinazione.

L'apprendimento robotico è un campo di intersezione tra la robotica e l'apprendimento automatico. Si occupa dello studio delle tecniche che permettono ai robot di acquisire nuove capacità o di adattarsi al loro ambiente attraverso algoritmi di apprendimento.

L'apprendimento automatico (noto come *machine learning*) è un sottocampo dell'informatica e dell'intelligenza artificiale, che si occupa dello studio e della realizzazione di sistemi e algoritmi, i quali si basano su osservazioni come dati per la sintesi di nuova conoscenza, piuttosto che seguire le istruzioni programmate solo in modo esplicito.

Sostanzialmente, l'apprendimento può avvenire catturando caratteristiche di interesse provenienti da esempi, strutture dati, o sensori, al fine di analizzarle e valutare le relazioni tra le variabili osservate.

1.1 Caratteri generali e definizione formale

La creazione di programmi e sistemi che apprendono dall'esperienza permette di far fronte a tutto quell'insieme di problemi per i quali risulta difficile la realizzazione di programmi in grado di risolvere essi in maniera soddisfacente; per di più i programmi potrebbero mancare di generalità, potrebbero non funzionare correttamente con una mole vistosa di dati, e probabilmente necessitano di aggiornamenti continui.

L'approccio *machine learning* consiste essenzialmente nella seguente prospettiva: anziché scrivere un programma manualmente per ogni specifico task, vengono raccolti numerosi esempi che specificano il comportamento corretto: un algoritmo di apprendimento automatico prende in esame questi esempi e ne produce un programma che esegue il comportamento desiderato. Il programma in questo modo funzionerà anche per nuovi casi, inoltre, se cambiano i dati, anche il programma si modifica, apprendendo dai nuovi dati.¹

Nel 1959 Arthur Samuel aveva definito l'apprendimento automatico come

“campo di studio che conferisce ai computer la capacità di imparare senza essere esplicitamente programmato.”²

La definizione formale, postulata da Tom M. Mitchell, include un qualsiasi programma che migliora le sue prestazioni in merito ad un certo compito attraverso l'esperienza. Precisamente:

“un programma apprende da una certa esperienza E nel rispetto di una classe di compiti T , con una misura di prestazioni P , se la prestazione P misurata, nello svolgere il compito T è migliorata dall'esperienza E .”³

¹Cfr. A. Roli, *Robot Learning*, dispense del corso di Sistemi Intelligenti Robotici, a.a. 2013/2014.

²Rif. P. Simon, *Too big to Ignore: Business Case for Big Data*, Wiley, 2013.

³Cit. T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

1.2 Paradigmi e algoritmi di Apprendimento

Gli algoritmi di apprendimento possono essere organizzati in una tassonomia basata sul risultato voluto dall'algoritmo o dal tipo di ingresso disponibile durante l'addestramento della macchina.

Distinguiamo in primo luogo le principali tipologie basate sul tipo di dati in ingresso:

- **Apprendimento supervisionato**

Questo tipo di algoritmo cerca di dedurre e generalizzare una funzione o un mapping di input-output, i quali dovrebbero essere in grado di predire in corretto valore di output per ogni input valido.

L'approccio utilizzato è dunque il seguente: viene fornito un insieme di esempi che rappresentano le relazioni di input-output; questi esempi sono utilizzati per fornire un feedback al processo di apprendimento, il quale genera un modello di stima delle relazioni di input-output non note.

- **Apprendimento non supervisionato**

In questo caso il problema diventa quello di trovare strutture nascoste in strutture dati non classificate in precedenza, dalle quali dunque non si è in grado di valutare una possibile soluzione. In generale quindi i dati di input non hanno i loro corrispondenti dati di output in dotazione, perciò l'obiettivo è quello di apprendere come i dati sono organizzati, scoprendo così il modello e costruendo una rappresentazione di essi.

- **Apprendimento con rinforzo**

Questa tipologia si basa sul presupposto di poter ricevere stimoli esterni a seconda delle scelte dell'algoritmo, per cui i dati in input sono generati dall'interazione del soggetto con l'ambiente; successivamente un *critico* conferisce una ricompensa o una penalità al soggetto al fine di indicare la desiderabilità del comportamento risultante. Il ruolo del *critico* dunque rappresenta il feedback a cui fa riferimento il processo di apprendimento. In ultima analisi lo scopo del soggetto è quello

di selezionare sequenze di azioni tali da massimizzare l'accumularsi di ricompense.⁴

Analizzando invece gli algoritmi in base al modo in cui essi formano la loro "esperienza", sulla base delle quali poi effettuano le scelte, distinguiamo le seguenti tipologie:

- **Esperienza con apprendimento continuo**

Gli algoritmi di questo tipo hanno come presupposto la disposizione di un meccanismo in grado di valutare le scelte dell'algoritmo, e quindi premiano o puniscono quest'ultimo a seconda del risultato; il risultato è quello di potersi adattare anche a modifiche sostanziali dell'ambiente. Questa tipologia di algoritmo viene utilizzata da esempio nei programmi di riconoscimento del parlato o di riconoscimento ottico dei caratteri (*optical character recognition*, OCR), i quali migliorano le proprie prestazioni grazie all'utilizzo dell'algoritmo stesso.

- **Esperienza con addestramento preventivo**

Questa tipologia di algoritmo parte dalla constatazione che l'analisi costante delle azioni dell'algoritmo può essere un processo non automatizzabile oppure molto costoso: in questo caso si istruisce l'algoritmo e, quando il sistema viene ritenuto affidabile, esso viene cristallizzato e reso non più modificabile. Questa strategia è spesso applicata in componenti elettronici che utilizzano reti neurali al loro interno: i pesi sinaptici delle reti non sono modificabili, ma fissati durante la realizzazione circuitale.

⁴Cfr. A. Roli, *Robot Learning*, dispense del corso di Sistemi Intelligenti Robotici, a.a. 2013/2014.

1.3 Approcci

Di seguito vengono elencati i principali approcci all'apprendimento automatico.⁵

- **Albero decisionale**

Un albero di decisione è fondamentalmente un grafo di decisioni e delle loro possibili conseguenze (inclusi costi, risorse e rischi), impiegato per realizzare un *plan* (ossia un “piano di azione”) atto ad un *goal* (scopo). Nell'ambito specifico dell'apprendimento automatico un albero di decisione è un modello predittivo, nel quale ogni nodo interno rappresenta una variabile, ogni arco verso un nodo figlio rappresenta un possibile valore per quella proprietà, e una foglia il valore predetto per la variabile obiettivo a partire dai valori delle altre proprietà, il quale è rappresentato all'interno del grafo dal cammino dal nodo radice alla foglia. Solitamente gli alberi decisionali vengono realizzati utilizzando tecniche di apprendimento a partire dall'insieme dei dati iniziali. Questo insieme (*data set*) può essere diviso in due sottoinsiemi: il *training set*, sulla base del quale viene creata la struttura del grafo, e il *test set*, il quale viene utilizzato per valutare l'accuratezza del modello predittivo creato. Al fine di evitare una crescita eccessiva dell'albero decisionale, che potrebbe causare uno sproporzionato aumento della complessità computazionale rispetto ai benefici di questo approccio, spesso si rivela necessario definire un criterio di arresto, detto anche *criterio di potatura* (*pruning*): tale accorgimento permette di determinare la profondità massima dell'albero.

- **Rete neurale artificiale**

Una rete neurale artificiale è un modello matematico che rappresenta l'interconnessione tra elementi, detti neuroni artificiali, i quali sono costrutti matematici che in qualche misura imitano le proprietà dei neuroni biologici. La rete riceve segnali esterni su uno strato di nodi

⁵Rif. *Apprendimento automatico*, Wikipedia.

(neuroni) d'ingresso, ciascuno dei quali è collegato con numerosi nodi interni, organizzati in più livelli. Ogni nodo elabora i segnali ricevuti e trasmette il risultato ai nodi successivi. Naturalmente questo tipo di rete può essere realizzato sia da programmi software che da hardware dedicato.

Durante la fase di apprendimento, la struttura della rete neurale muta sulla base delle informazioni interne o esterne che scorrono attraverso essa: ciò indica che le reti di questo tipo sono sistemi di tipo adattativo.

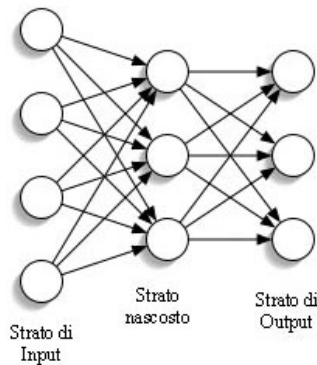


Figura 1.1: Struttura di una semplice rete neurale artificiale

- **Programmazione Genetica**

Questo approccio utilizza gli algoritmi genetici al fine di apprendere. Un algoritmo genetico è un algoritmo euristico che trae la sua ispirazione dal principio della selezione naturale e dell'evoluzione biologica teorizzato da Charles Darwin nel 1859: il nome deriva dal fatto che il modello darwiniano trova spiegazione nella genetica e dal fatto che gli algoritmi genetici hanno meccanismi concettualmente simili a quelli dei processi biochimici scoperti da questa branca.

La ricerca della soluzione del problema inizia con una popolazione di soluzioni iniziale; per mezzo di operazioni quali la mutazione casuale

e il crossover (modellati sui processi di evoluzione biologica), essa dà luogo ad una nuova generazione di popolazione. Ad ogni passaggio, la popolazione di soluzioni attuale viene valutata secondo un criterio di fitness stabilito, per cui le soluzioni più adatte vengono selezionate in maniera probabilistica come candidate per la produzione della generazione successiva.

Gli algoritmi genetici sono stati applicati in maniera efficace ad una vasta gamma di compiti di apprendimento e problemi di ottimizzazione, in particolare vengono utilizzati per ottimizzare la topologia dei parametri di apprendimento per reti neurali artificiali.

- **SVM**

Sviluppate da Vladimir Vapnik negli anni '90, le macchine a vettori di supporto (*Support Vector Machines*, SVM) sono un gruppo di metodi di apprendimento supervisionato per l'analisi, la stima e la classificazione di *pattern*. La tecnica di addestramento SVM presenta un algoritmo efficiente, ed è in grado di rappresentare funzioni non lineari complesse, per cui fa fronte ai principali problemi delle reti neurali; se queste ultime infatti sono ad un solo strato, l'algoritmo è efficiente, ma utile solo nel caso di dati linearmente separabili, se invece esse sono multistrato, possono rappresentare funzioni non lineari, ma diventano difficili da addestrare a causa della dimensione. Dato un set di esempi di addestramento, ciascuno contrassegnato come appartenente a due possibili categorie, un algoritmo di addestramento SVM costruisce un modello in grado di prevedere a quale categoria debba appartenere un nuovo esempio di input.

- **Clustering**

Il clustering (o *cluster analysis*, termine introdotto da Robert Tyrone nel 1939), è un gruppo di tecniche atte alla selezione e al raggruppamento di elementi omogenei in un insieme di dati. Le strategie di clustering si basano sulla misurazione della similarità, o piuttosto della dissimilarità

tra gli elementi: spesso essa è concepita in termini di distanza in uno spazio multidimensionale, motivo per cui la bontà delle analisi ottenute dipende strettamente dalla metrica scelta. L'appartenenza di un elemento ad un insieme dipende dalla distanza tra l'elemento e l'insieme stesso.

Le strategie di raggruppamento si dividono in due categorie principali: dal basso verso l'alto (*Bottom-Up*), metodi aggregativi in cui ogni elemento è considerato come cluster a sé, e poi l'algoritmo si occupa dell'unione dei cluster più vicini, fino a raggiungere il numero di cluster prefissato, oppure dall'altro verso il basso (*Top-Down*), metodi divisivi in cui inizialmente tutti gli elementi sono in un unico cluster, e l'algoritmo si occupa della divisione in tanti cluster di dimensioni inferiori, sulla base dell'omogeneità, fino a raggiungere il numero di cluster prefissato. Il clustering è un metodo di apprendimento non supervisionato.

- **Reti bayesiane**

Le reti bayesiane sono un insieme di grafi probabilistici che rappresentano un gruppo di variabili aleatorie con le loro dipendenze condizionali: per cui i nodi rappresentano le variabili, mentre gli archi rappresentano le relazioni di dipendenza statistica tra le variabili e le distribuzioni locali di probabilità dei nodi foglia rispetto ai valori dei nodi padre. Una rete bayesiana, in ultima analisi, rappresenta la distribuzione della probabilità congiunta di un insieme di variabili.

L'approccio bayesiano dunque si pone come metodo di inferenza probabilistica, basandosi sul presupposto che le quantità di interesse siano governate da distribuzioni di probabilità, e che le decisioni ottimali possano essere prese sulla base dell'analisi di queste probabilità insieme ai dati osservati. La teoria Bayesiana, nell'ambito dell'apprendimento automatico, fornisce un criterio quantitativo per valutare le prove a sostegno delle ipotesi alternative.

1.4 Requisiti dell'apprendimento robotico

benché l'ingegneria e le scienze cognitive siano due campi diversi sotto molti aspetti, essi si trovano ad avere a che fare con problematiche simili quando si tratta di apprendimento mediante l'utilizzo di robot reali: si rivela perciò necessario definire alcuni requisiti per gli algoritmi che devono funzionare nel mondo reale.⁶ I più importanti requisiti per l'apprendimento automatico in robotica sono:

- **Immunità ai disturbi** (*Noise immunity*): questo requisito vale per qualsiasi sistema che opera nel mondo reale; è uno dei motivi per cui spesso si rivela conveniente utilizzare reti neurali robuste, resistenti ai disturbi.
- **Convergenza rapida** (*Fast convergence*): gli algoritmi devono convergere velocemente, poiché le azioni sono necessarie per accumulare esperienza, e verrebbe impiegato troppo tempo nell'esecuzione di un grande numero di azioni.
- **Apprendimento On-line** (*On-line learning*): un algoritmo dovrebbe consentire al robot di imparare ad eseguire il proprio compito. Dal momento che il robot stesso genera “esempi” attraverso meccanismi di coordinamento sensoriale-motorio, ciò gli permette ad di esplorare il suo ambiente circostante molto più velocemente e di produrre esempi sempre migliori, poiché esso migliora continuamente le proprie prestazioni.
- **Apprendimento incrementale** (*Incremental learning*): questo requisito riveste un ruolo fondamentale, poiché l'apprendimento non deve mai arrestarsi. Ciò implica una buona adattatività del robot in quanto, in un mondo che cambia continuamente, esso deve sempre essere pronto a cambiare. Dal momento che i dati non sono noti in anticipo, può non

⁶Cfr. R. Pfeifer C. Scheier, *Understanding Intelligence*, Bradford Books.

esserci distinzione tra un apprendimento e una fase di esecuzione come nel paradigma di apprendimento supervisionato.

- **Trattabilità** (*Tractability*): gli algoritmi di apprendimento dovrebbero essere trattabili a livello computazionale, ossia ogni iterazione deve poter essere effettuata in tempo reale.
- **Fondatezza** (*Groundedness*): le tecniche utilizzate dovrebbero dipendere dalle informazioni acquisite dai sensori, oppure dalle informazioni acquisite dal robot nel tempo. Questo concetto è strettamente collegato alla nozione di contesto: ogni apprendimento dovrebbe infatti essere basato sulla visione del mondo da parte del soggetto (in questo caso il robot), e non sulla visione del progettista o dell'osservatore.

Capitolo 2

Self-Awareness

2.1 Generalità

Il termine *Self-Awareness* letteralmente significa consapevolezza di sé. Rispetto alle tipologie di apprendimento analizzate in precedenza, il concetto di self-awareness trova certamente alcune somiglianze, ma rappresenta perlopiù un indirizzo a sé stante. Nasce dalla necessità di gestire azioni e comportamenti in un ambiente complesso e dinamico: la maggior parte dei sistemi intelligenti robotici è fragile di fronte al cambiamento, poiché è legata ad un dominio ristretto di possibili condizioni e situazioni, le quali difficilmente rispecchiano la totalità dei possibili scenari. Una volta implementati, i sistemi si trovano effettivamente limitati ad un insieme statico di istruzioni, in forma di azioni e comportamenti, che non fanno fronte a alla grande varietà di situazioni offerte da un ambiente aperto, articolato e attivo: non sorprende dunque il frequente fallimento dell'esecuzione. La self-awareness si distingue proprio per la possibilità di eludere queste problematiche: essa infatti permette al robot di prendere coscienza di sé, e di generare i propri modelli interni, senza possedere una conoscenza preesistente, oppure di affinarli attraverso l'esperienza, in modo tale da compensare eventuali fattori destabilizzanti esterni.

Una delle prime diffusioni del concetto di self-awareness in robotica risale

al 2007, sulla celebre piattaforma di divulgazione TED ¹; nella conferenza, assistiamo ad un esempio pratico di consapevolezza di sé, in un piccolo robot quadrupede, simile ad una stella marina, che Hod Lipson descrisse in questo modo:

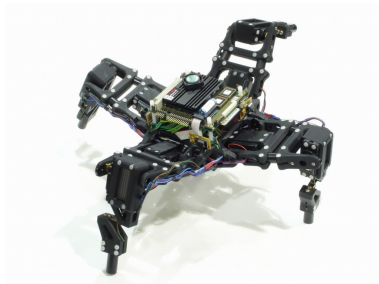


Figura 2.1: Lo starfish robot utilizzato nella dimostrazione

“[...]Ma questa macchina non sa a cosa assomiglia. La guardi e vedi che ha quattro gambe, la macchina non sa se è un serpente, se è un albero, non ha alcuna idea del suo aspetto, ma proverà a scoprirlo. Inizialmente, fa qualche movimento casuale, e poi cerca di capire che aspetto potrebbe avere... e vedete che le passano molti pensieri per la testa, molti auto-modelli che cercano di spiegare le relazioni tra l’agire ed il sentire... e poi prova a fare una seconda azione il più lontano possibile tra le previsioni di questi modelli alternativi, come gli scienziati in laboratorio. Poi fa quell’azione, cerca di spiegarla, e scarta i modelli di sé non validi.

Questo è l’ultimo ciclo, e vedete come abbia capito piuttosto bene a che cosa assomiglia, ed una volta che ha un modello di sé, può usarlo per dedurre un modo di muoversi. State quindi vedendo un paio di macchine...un pattern di locomozione. Speravamo che assumesse una camminata aggressiva, da ragno, ma invece ha

¹Rif. H. Lipson, *Building “self-aware” robots*, TED, 2007.

creato questo modo piuttosto noioso di muoversi.

Ma quando la osservate dovete ricordare che questa macchina non ha fatto alcun test su come muoversi in avanti, nè aveva un modello di sé. Ha cercato di capire a cosa assomigliasse, come muoversi in avanti, e poi in effetti lo ha fatto.”²

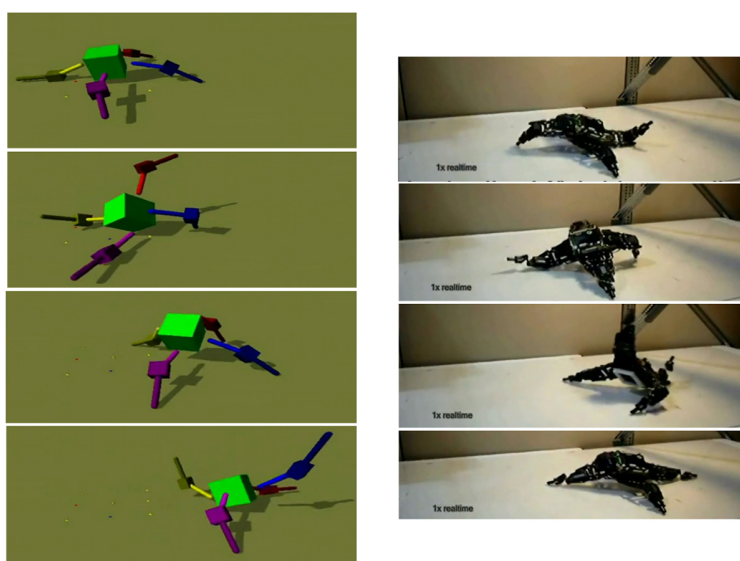


Figura 2.2: (a sinistra) Pattern di locomozione creato dal robot (simulato), (a destra) pattern di locomozione del robot fisico

Volendo riassumere dunque questo caso di studio specifico, possiamo sintetizzare i seguenti passi fondamentali:

1. **primo step:** il robot esegue fisicamente un'azione, la quale inizialmente è casuale, mentre nelle successive ricorrenze di questo step è generata dal terzo step.
2. **secondo step:** il robot genera alcuni modelli di sé, solidali con i dati raccolti dai sensori durante l'esecuzione dell'azione precedente; esso non sa quale modello sia corretto.

²Cit. H. Lipson, *Building self-aware robots*, TED, 2007.

3. **terzo step:** il robot genera alcune azioni possibili, le quali permettono di eliminare le ambiguità presenti tra i modelli. Si torna quindi al primo step.
4. **quarto step:** sintesi del comportamento; dopo un certo numero di cicli dei passi sopra elencati, il modello attualmente migliore viene utilizzato per generare una sequenza di locomozione.
5. **quinto step:** la migliore locomozione viene attuata dal robot fisico
6. **sesto step:** il ciclo continua al secondo step nel caso si voglia affinare il modello, oppure dal quarto step se si vogliono generare nuovi comportamenti.

2.2 Il processo di sintesi automatica e l'algoritmo EEA

I metodi con cui un robot è in grado di sintetizzare lo schema della propria struttura, attraverso l'interazione con il mondo reale, aprono un interessante confronto con il mondo naturale: il modo in cui gli organismi creano i loro schemi strutturali, basandosi sulle interazioni con il mondo reale, al giorno d'oggi sono un problema irrisolto, ma il domandarsi come possa avvenire tale fenomeno fornisce il punto di partenza da cui scaturiscono gli approcci utilizzati in robotica. In particolare, ci sono questioni da affrontare a livello biologico in primis, per poi estenderle al mondo artificiale: nel cervello degli animali superiori i modelli interni sono codificati alla nascita e via via affinati con l'esperienza, o sono sintetizzati nel corso dell'esistenza? E se così fosse, come si sono formati? Gli animali mantengono un unico modello riguardo ad una particolare parte del corpo, o mantengono contemporaneamente modelli concorrenti? Da questi interrogativi discende l'algoritmo di stima ed esplorazione (*Estimation-Exploration Algorithm*, EEA), un algoritmo co-evolutivo, in cui "i modelli generano le azioni, e le azioni sono utilizzate per migliorare

i modelli”³, che consente ai robot di generare automaticamente e mantenere modelli candidati multipli che descrivono la loro morfologia, basandosi solo sui dati registrati dai sensori. Il vantaggio derivante dall'utilizzo di questo algoritmo è duplice, in primo luogo perché vengono generati automaticamente dei modelli predittivi (sostanzialmente dei simulatori) che minimizzano il numero di tentativi effettuati dal robot in fase di apprendimento fisico, riducendo quindi i costi, i tempi, e i rischi che ciò potrebbe comportare, in secondo luogo perché si verifica in maniera continua una evoluzione dei controllori del robot e un adattamento dei modelli interni.

2.2.1 Modelli multipli, e test intelligente

L'identificazione di un sistema implica la costituzione automatica di un modello del sistema stesso, utilizzando unicamente gli input forniti al sistema, e gli output osservati. Dunque un modello accurato è definito come uno che produce dati in uscita simili a quelli del sistema nel momento in cui entrambi vengono dotati dello stesso input. Nel caso di modelli multipli tuttavia lo scenario appare più articolato. L'algoritmo co-evolutivo sintetizza e mantiene modelli multipli, che possono essere utilizzati per la creazione di nuovi controllori (e per controllore intendiamo il sistema di controllo, ossia una traslitterazione del termine *controller* ampiamente utilizzato in informatica): nel caso si voglia dedurre un qualche comportamento voluto in condizioni di incertezza, si utilizzano modelli concorrenti concordi su quale debba essere il comportamento risultante, oppure, nel caso in cui si vogliano determinare azioni che rivelano aspetti nascosti del robot, vengono utilizzati modelli discordanti tra loro, i quali indicano chiaramente la necessità di un'ulteriore sintesi del modello. Queste proprietà dei modelli multipli sono alla base del concetto di test intelligente.

Nel caso dell'identificazione di un sistema, un test intelligente è quello che svela indirettamente componenti interni nascosti del sistema, consentendo in

³Cfr. J. Bongard, H. Lipson, *Automatic Synthesis of Multiple Internal Models Through Active Exploration*, Cornell University.

tal modo la generazione di modelli più accurati. Ciò può essere ottenuto evolvendo un test che provoca il massimo disaccordo all'interno dell'attuale insieme di modelli candidati; l'uscita del sistema sarà tale da fornire una maggiore veridicità ad alcuni modelli, e nel contempo rivelerà l'imprecisione di altri modelli. In altre situazioni, un test intelligente è quello che deduce alcuni comportamenti richiesti dal modello più accurato attualmente a disposizione; se il modello è preciso, allora lo stesso test dovrebbe produrre lo stesso comportamento desiderato nel sistema.⁴

In Figura viene mostrato un esempio pratico delle due definizioni complementari appena menzionate:

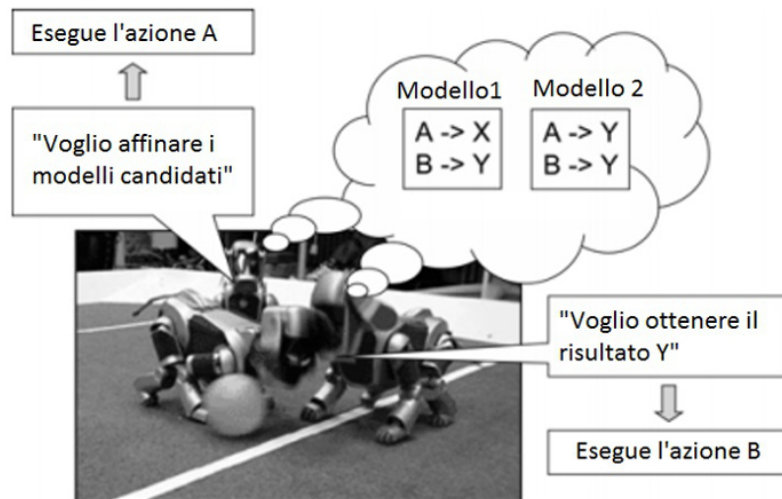


Figura 2.3: Esempio delle due definizioni complementari di test intelligente

in questo esempio, il robot di destra desidera ottenere uno specifico risultato Y, per cui sceglie di eseguire l'azione B poiché entrambi i modelli concordano sul fatto che questa azione darà come risultato Y, benché i modelli siano differenti. Il robot di sinistra, che vuole sapere quale modello sia

⁴Cfr. J. Bongard, H. Lipson, *Automatic Synthesis of Multiple Internal Models Through Active Exploration*, Cornell University.

corretto, sceglie di compiere l'azione A, dal momento che i due modelli sono in disaccordo in merito al risultato di questa azione: in base al risultato ottenuto, uno dei due modelli verrà invalidato.

2.2.2 I passi dell'algoritmo di stima ed esplorazione

Come già menzionato, l'algoritmo EEA si occupa della sintesi automatica di modelli attraverso l'interazione con il mondo reale. La sintesi può essere di due tipologie:

- **Sintesi parametrica:** in questo caso viene fornito un modello approssimato e il robot affina questo modello.
- **Sintesi topologica:** in questo caso il robot sintetizza un modello predittivo della sua struttura, usando i dati raccolti dai sensori (eventualmente utilizzando anche una piccola porzione di conoscenza preesistente).

Indipendentemente dalla tipologia è possibile tuttavia riconoscere un flusso ben determinato di istruzioni da seguire:

1. Inizializzazione

- (a) Se è disponibile un modello approssimato, vai al punto 4.
- (b) Se non è presente alcun modello, genera un test casuale.

2. Esecuzione di un'azione di prova

- (a) Manda il test, evoluto o casuale che sia, a destinazione.
- (b) Registra l'output risultante.

3. Evoluzione dei modelli candidati (*Estimation Phase*)

- (a) Se questa è la prima volta che si attraversa la fase di stima, genera un insieme casuale di modelli candidati.
- (b) Se questa è la seconda volta che si attraversa la fase di stima, o una delle volte successive, aggiorna la popolazione dei modelli con i migliori modelli evoluti finora.

- (c) Fornisci ai modelli evoluti tutti i test e i gli output precedenti, più le nuove coppie di test e output.
 - (d) Fornisci i migliori modelli candidati alla fase di esplorazione.
4. **Evoluzione dei Test Informativi**(*Exploration Phase*)
- (a) Inizia sempre la fase di esplorazione con una popolazione casuale di test.
 - (b) Evolvi un test che causa il maggior disaccordo tra i modelli candidati forniti dalla fase di stima, oppure quello che suscita un qualche comportamento auspicabile, oppure una qualche combinazione di questi due criteri di idoneità (fitness criteria).
 - (c) Torna al punto 2. ⁵

Come si evince dal flusso algoritmico, l'EEA è ciclico, per il fatto che l'evoluzione dei modelli si alterna con l'evoluzione dei test; in particolare i dati provenienti dai test vengono incrementati nell'arco di tutto il tempo di vita dell'esecuzione, e ciò si trova in contrasto con molti metodi di apprendimento automatico, nei quali una grande quantità di dati di addestramento (training data) viene raccolta prima dell'inizio dell'inferenza. L'algoritmo è anche evolutivo, in quanto la fase di stima genera una popolazione di modelli candidati utilizzando un algoritmo evolutivo, e la fase di esplorazione genera una popolazione di test utilizzando un algoritmo evolutivo. Si può parlare in maniera più precisa dunque di co-evoluzione, poiché per definizione essa implica che il progresso evolutivo di una popolazione sia scambievolmente dipendente con il progresso evolutivo dell'altra: nel caso dell'EEA i test sono evoluti utilizzando i modelli correnti più accurati ottenuti dalla fase di stima, e a loro volta i modelli sono evoluti sulla base dei risultati ottenuti dai test in fase di esplorazione.

⁵Cit. J. Bongard, H. Lipson, *Automatic Synthesis of Multiple Internal Models Through Active Exploration*, Cornell Univerity.

2.3 Inferenza Parametrica

In molte applicazioni robotiche lo stato del robot è noto, tuttavia diversi suoi aspetti fisici e l'ambiente circostante sono caratteristiche sconosciute, perciò i comportamenti generati utilizzando un modello non sono propriamente adatti (e quindi trasferibili) al robot fisico. È però possibile utilizzare l'algoritmo di stima ed esplorazione per fornire ed affinare numerosi aspetti fisici non noti al modello, in modo tale da poter trasferire i comportamenti al robot fisico. Nel caso dell'inferenza parametrica dunque, si parte da una conoscenza parziale, per dedurre parametri fisici sconosciuti o, più in generale, affinare i modelli, anche a fronte di nuove e sconosciute variabili fisiche ed ambientali.

Caso di studio: inferenza parametrica dello “Starfish” robot

Per poter applicare l'algoritmo EEA ad un dato problema, sono necessari sei passi fondamentali: caratterizzazione del sistema designato, inizializzazione, fase di stima, fase di esplorazione, terminazione e validazione.

Procediamo con l'esplicazione di questi punti, nel caso di necessità di inferenza parametrica, prendendo come modello esemplificativo ancora una volta lo *Starfish Robot* utilizzato da Lipson:

1. **Caratterizzazione del sistema designato:** il sistema è costituito da un robot quadrupede e articolato, che opera all'interno di uno spazio fisico tridimensionale.
2. **Inizializzazione:** per cominciare, una rete neurale casuale viene generata e fornita al robot: successivamente i dati ricavati dai sensori vengono passati alla fase di stima.

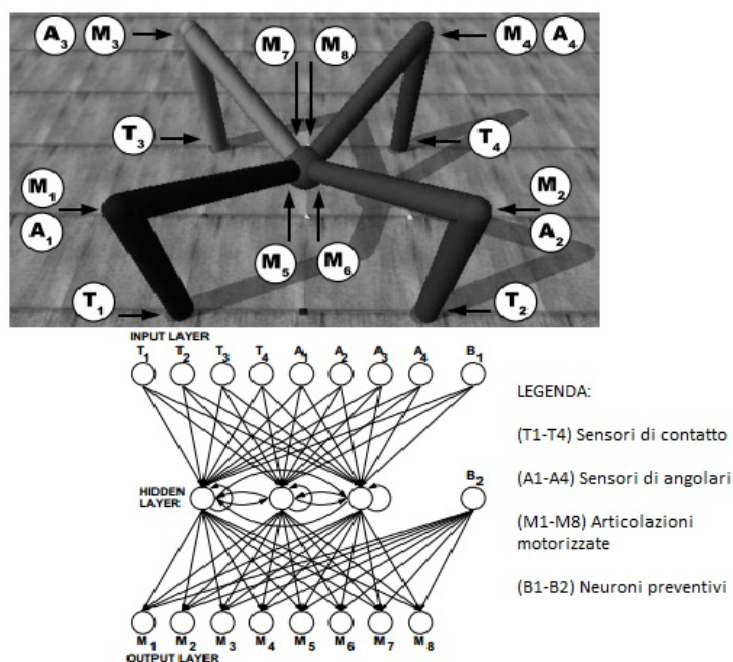


Figura 2.4: Morfologia del robot e rete neurale

3. **Fase di Stima:** questa fase utilizza un algoritmo genetico al fine di evolvere una popolazione di 100 modelli candidati, i quali sono identici al robot fisico in ogni aspetto, tranne che per le masse delle singole parti del corpo e gli intervalli di tempo dei sensori: il genoma che specifica un modello candidato fornisce questi valori mancanti. Dopo aver calcolato l'idoneità di ogni modello, i 50 che si sono rivelati più adatti vengono copiati, mutati e incrociati al fine di produrre 50 nuovi genomi, i quali rimpiazzano i restanti modelli meno adatti.
4. **Fase di Esplorazione:** in questa fase viene evoluta la popolazione dei genomi per un numero considerevole di generazioni (nel caso specifico i 100 genomi sono evoluti per 30 generazioni), utilizzando gli stessi meccanismi di mutazione e di crossover. Tuttavia in questo caso ogni genoma codifica una quantità di *pesi sinaptici* (ossia le intensità tra le connessioni fra i nodi), in modo da specificare un controllore della rete

neurale. Ogni controllore viene applicato al modello fornito dalla fase di stima, e il criterio di idoneità è stabilito (in questo caso specifico) dalla distanza raggiunta dal robot dopo un vasto numero di step nei quali ha utilizzato questo sistema di controllo.

5. **Terminazione:** l'algoritmo termina dopo che un ampio numero di reti neurali è stato collaudato sul robot, ossia quando l'algoritmo di stima ed esplorazione è stato eseguito altrettante volte.
6. **Validazione:** nel caso specifico viene semplicemente valutata la distanza percorsa dal robot come metro di giudizio della bontà dell'algoritmo.⁶

Risultati della Costruzione Parametrica

A causa dell'inaccuratezza del primo modello, la prima rete neurale generata provoca comportamenti diversi tra il modello e il robot. Tuttavia dopo 20 cicli è osservabile una omogeneizzazione tra il modello e il comportamento del robot, dimostrando in questo modo la correttezza del funzionamento dell'algoritmo di stima ed esplorazione.

⁶Rif. J. Bongard, H. Lipson, *Automatic Synthesis of Multiple Internal Models Through Active Exploration*, Cornell University.

2.4 Inferenza Topologica

Se nel caso precedente la maggior parte dei parametri riguardanti il robot era noto, e l'algoritmo di stima ed esplorazione veniva applicato per dedurre soltanto i parametri mancanti, nel caso dell'inferenza topologica si parte dal presupposto di dover risalire alla morfologia di un robot "da zero", sfruttando (a meno di una minimale conoscenza a priori) unicamente i dati provenienti dai sensori.

In generale, la possibilità di apprendere la propria morfologia automaticamente, si rivela essere un sistema valido in molte situazioni, tra cui citiamo la diagnosi dei danni e il recupero (*recover*) in ambienti remoti, e la rapida modellazione di robot complessi.

Inferenza topologica dello "Starfish" robot

Anche in questa occasione occorre effettuare stessi gli step utilizzati per inferenza parametrica; ci riconduciamo al medesimo robot analizzato nel punto precedente per avere, oltre che un esempio concreto, un elemento di confronto con il caso parametrico.

1. **Caratterizzazione del sistema designato:** il sistema viene assunto in maniera molto essenziale come un robot composto di 5 parti cilindriche, dotato di 2 sensori di contatto, e 4 sensori angolari, uno per ogni articolazione. Vengono definiti inoltre dettagli come il peso di ogni parte cilindrica e l'angolo di rotazione delle articolazioni.
2. **Inizializzazione:** come nel caso precedente, viene generato un insieme casuale di pesi sinaptici e caricato sul robot, al quale, nel caso di studio preso in esame, è consentito di muoversi per 20 passi; tuttavia in questo caso è la bravura con cui l'algoritmo riesce a ricostruire la topologia del robot a determinare l'efficacia dell'algoritmo stesso, e non la distanza che il robot riesce a percorrere per merito dell'algoritmo. Ancora una volta, i dati dei sensori vengono inoltrati alla fase di stima.

3. **Fase di Stima:** questa fase tenta di ricostruire la topologia del robot, e non solo alcuni dei suoi parametri fisici. I dati che sono stati specificati in fase di caratterizzazione del sistema designato sono assunti come noti (rientrano quindi nella minimale conoscenza a priori), mentre sono assunti come sconosciuti i seguenti parametri: la lunghezza delle parti del corpo, quale parte sia collegata con l'altra, la posizione relativa e l'orientamento delle giunzioni, e la collocazione dei sensori di contatto. Spetta al genoma la codifica di questi dettagli mancanti per poter generare un modello candidato. Durante il primo passaggio attraverso la fase di stima, viene generata una popolazione di 100 matrici di genomi, sottoposti a selezione e mutazione, la quale è governata da meccanismi tali per cui vengono preservati i genomi più idonei. Questa fase prosegue per 20 generazioni, al termine delle quali il risultato più accurato ottenuto viene fornito alla fase di esplorazione. Nei successivi passaggi attraverso questa fase, la popolazione viene alimentata dalle n matrici di genomi, che corrispondono ai modelli più accurati ottenuti generati durante le n passate precedenti della fase in questione.

4. **Fase di Esplorazione:** questa fase non è abilitata in senso stretto in questo caso; essa semplicemente produce un sistema di controllo di rete neurale casuale. In questa applicazione infatti l'obiettivo non è quello di produrre un controllore ottimale per il robot, bensì quello di produrre un'accurata simulazione del robot automaticamente: in realtà è sufficiente possedere controllori casuali basati sui sensori (*sensor-based*) per suscitare un insieme di comportamenti casuali nel robot, e quindi suscitare in maniera indiretta informazioni morfologiche riguardanti il robot.
La fase si conclude con il caricamento del sistema di controllo sul robot, il quale viene valutato.

5. **Terminazione:** nel caso specifico, dopo 10 cicli l'accuratezza dei modelli non aumenta ulteriormente, perciò viene terminato.

6. **Validazione:** essa consiste nel valutare, a fronte di una metrica di errore oggettivo, quanto sia precisa la distanza tra le singole parti del corpo tra loro e rispetto al centro geometrico; questa stima fornisce una buona approssimazione riguardo alla bontà dell'algoritmo di inferenza.⁷

Risultati della Costruzione Topologica

Come appena osservato, dopo un certo numero di cicli, si ottiene un modello dall'accuratezza costante, tuttavia il livello di accuratezza è subordinato al numero e alla tipologia di sensori utilizzati. La seguente figura mostra tre esecuzioni indipendenti dell'algoritmo per la deduzione topologica: la prima naturalmente è quella analizzata nel caso di studio, le altre sono esecuzioni nelle quali vengono utilizzati sensori di diversa natura, e in numero differente: la seconda esecuzione comporta l'uso di 5 sensori di distanza, la terza ne usa 20.

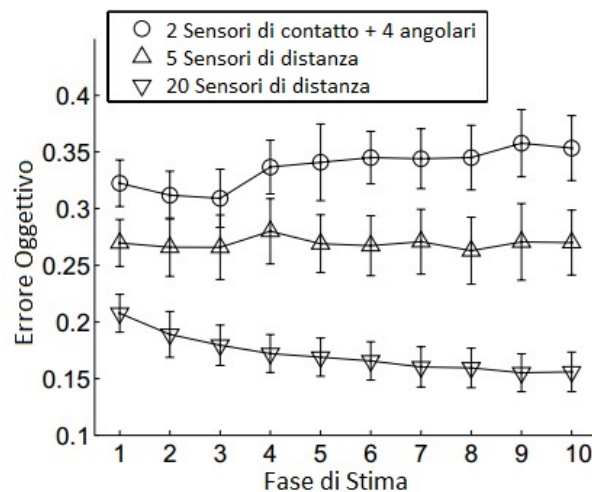


Figura 2.5: Risultato di tre diverse esecuzioni, utilizzando configurazioni di sensori differenti

⁷Rif. J. Bongard, H. Lipson, *Automatic Synthesis of Multiple Internal Models Through Active Exploration*, Cornell University.

Dal grafico si evince con chiarezza che l'utilizzo di sensori di distanza è significativamente migliore. In aggiunta, è possibile notare come la quantità di sensori utilizzati sia secondaria alla tipologia: si osserva infatti che 5 sensori di distanza sono più efficaci di 6 sensori di altra natura (di contatto e angolari).

2.5 Osservazioni

La consapevolezza di sé nell'ambito delle scienze cognitive, come già accennato, è un argomento che suscita molti quesiti, grazie ai quali è stato possibile strutturare approcci e metodi per l'ambito robotico. Uno dei quesiti iniziali consisteva nel chiedersi se gli animali mantengano modelli multipli della loro morfologia, e grazie all'EEA è stato dimostrato che possedere una pluralità di modelli si rivela utile in termini di affermazione e anticipazione: infatti la sintesi di sistemi di controllo che comportano l'accordo dei modelli, aumenta la probabilità che il robot fisico riprodurrà il comportamento desiderato e sintetizzato, mentre la sintesi di sistemi di controllo che causano disaccordo tra i modelli, aumentano la probabilità di ottenere una nuova configurazione, tale da permettere ulteriori raffinamenti dei modelli.

Un'altra domanda tra le più preponderanti all'interno delle scienze cognitive, che è stata posta inizialmente, consisteva nello stabilire se gli animali superiori nascano con modelli già codificati nel cervello, e poi migliorati attraverso l'esperienza, oppure i modelli vengano sintetizzati "da zero" mano a mano che i essi ricavano informazioni sulla loro struttura e sull'ambiente che li circonda; entrambi gli approcci possono essere documentati in ambito robotico, difatti l'identificazione parametrica ha la funzione di miglioramento dei modelli preesistenti, mentre l'identificazione topologia permette di inferire la morfologia del robot dato un insieme pressoché nullo di dati iniziali.

Infine, nelle neuroscienze è stato argomentato che dati sensoriali multimodali sono necessari per creare schemi di un corpo, caratteristica osservabile anche in robotica, dove gli schemi strutturali possono essere costituiti sulla base

delle correlazioni tra segnali provenienti da varie tipologie di sensori, oppure ricavati triangolando sensori di alta qualità al fine di possedere lo stesso dato in maniera ottimale.

Capitolo 3

Concetti e Applicazioni derivanti dalla Self Awareness

Nel capitolo precedente è stato analizzato il concetto di consapevolezza di sé, e dei relativi metodi di inferenza dei dati. Cerchiamo ora di osservare casi di particolare rilevanza che riguardano l'applicazione dei metodi e degli algoritmi identificati.

3.1 Il *Continuous Self Modelling*: generalità

Una caratteristica apprezzabile in riferimento alla self awareness, rispetto ad altri approcci di apprendimento robotico, è sicuramente la flessibilità, che permette al robot di far fronte ad avvenimenti inaspettati di natura ambientale o strutturale: dal momento che è la macchina stessa a generare ed affinare i propri modelli e schemi comportamentali in maniera dinamica, essa ha la possibilità di dedurre sempre nuove soluzioni anche a seguito di cambiamenti improvvisi e imprevisi. Il cuore di questo processo consiste dunque nell'auto-modellazione autonoma e continua, nota anche come *Continuous Self Modelling*: un robot è in grado di dedurre la sua morfologia attraverso l'esplorazione autonoma e successivamente di utilizzare il modello di sé

risultante per generare nuove capacità, inoltre se la topologia o le caratteristiche ambientali mutano improvvisamente, il processo stesso ristrutturata i suoi modelli interni, portando alla generazione, qualitativamente diversa, di un comportamento compensativo.

Anche in questo ambito appare evidente una certa analogia con il mondo naturale: gli animali infatti possiedono la capacità di continuare a “funzionare” anche a seguito di lesioni, adottando comportamenti di compensazione.

3.2 L’algoritmo di stima ed esplorazione: autodiagnosi e recupero

Grazie alla proprietà di auto-modellazione continua, l’EEA si rivela particolarmente adatto a molti scenari: sono infatti copiosi i casi in cui si utilizzano robot in remoto, come ad esempio esplorazioni spaziali oppure operazioni in ambienti pericolosi, in cui un operatore umano non può procedere ad una riparazione manuale oppure ad una compensazione di fallimenti o danni. La possibilità di ripresa in seguito a fail (fallimenti) o a lesioni rappresenta pertanto un grande beneficio.

Tuttavia, di fronte al manifestarsi di un errore, un algoritmo di “generazione e testing” ripetuti non è ottimale: tentativi ripetuti potrebbero amplificare il danno e consumare la limitata quantità di energia, inoltre per poter condurre collaudi ripetuti a livello di hardware è richiesto un lungo arco di tempo. Tutto ciò sembrerebbe in contrasto con quanto detto in precedenza, ma in realtà è dimostrabile che la funzionalità quasi completa del robot può essere ripristinata dopo solo 3 valutazioni (in media) sul robot fisico, contrariamente alle 3000 che verrebbero fatte se il sistema di controllo fosse tutto evoluto sul robot fisico¹: dunque si tratta solo di determinare a che livelli avviene il processo di modellazione continua. Usiamo algoritmi evolutivi per co-evolvere i corpi e i cervelli dei robot, ma utilizzando un processo inverso:

¹Cfr. J. Bongard, H. Lipson, *Automated Damage Diagnosis and Recovery for Remote Robotics*, Cornell University.

3.2 L'algoritmo di stima ed esplorazione: autodiagnosi e recupero 29

anziché evolvere il sistema di controllo a partire dalla morfologia del robot, viene evoluta la morfologia dato un sistema di controllo. Ancora, anziché evolvere al fine di raggiungere un'alta condizione, si evolve verso una bassa condizione osservata (causata da qualche malfunzionamento non noto) come forma di diagnosi.

3.2.1 Metodologia

L'algoritmo di stima ed esplorazione possiede due funzioni: in fase di stima svolge un ruolo di evoluzione delle ipotesi di danno, mentre in fase di esplorazione si occupa dell'evoluzione del sistema di controllo. In questa particolare applicazione dell'algoritmo EEA, esso viene suddiviso in due algoritmi evolutivi separati, l'algoritmo di stima e l'algoritmo di esplorazione. In figura osserviamo il flusso dei due algoritmi, messo a confronto con un convenzionale algoritmo di recovery.

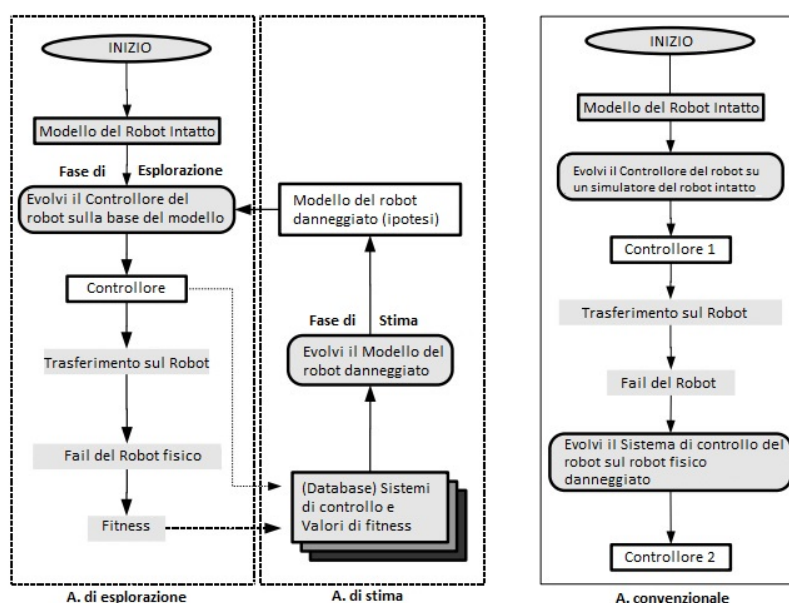


Figura 3.1: Flusso di esecuzione degli algoritmi di esplorazione e di stima, e confronto con il flusso degli algoritmi convenzionali.

- **Fase di Esplorazione: evoluzione del Sistema di controllo.** L'algoritmo di esplorazione viene utilizzato per evolvere il controllore (spesso mediante l'utilizzo di un robot simulato), in modo che sia in grado di eseguire qualche funzione. La prima volta che si attraversa questa fase, viene generato il sistema di controllo del robot intatto, mentre nelle passate successive si cerca di evolvere un controllore di compensazione per il robot danneggiato, usando l'ipotesi di danno più accreditata tra quelle generate fino a quel momento dalla fase di stima. Quando la fase di esplorazione termina, il migliore controllore ottenuto è trasferito al robot fisico e utilizzato.

Fallimento del robot fisico. Il robot utilizza un controllore evoluto per compiere qualche azione: nel caso attuale per semplicità, ipotizziamo che l'azione in questione sia un'andatura in avanti. Si verifica un errore imprevisto sul robot, ed esso registra il suo sbilanciamento (ad esempio uno squilibrio nell'andatura) per un periodo di tempo; successivamente il robot viene fermato, e lo sbilanciamento registrato è inserito in nel database dell'algoritmo, assieme al sistema di controllo a bordo del robot in quel momento: questa coppia permette di applicare il *reverse engineering* sul danno presente nel robot. Durante le passate seguenti dell'algoritmo, il robot danneggiato tenta di funzionare mediante l'utilizzo del sistema di controllo di compensazione prodotto dalla fase di esplorazione stessa.

- **Fase di Stima: evoluzione delle ipotesi di danno.** L'algoritmo di stima viene utilizzato per evolvere un'ipotesi riguardo all'effettivo malfunzionamento riscontrato dal robot. Questa fase usa lo sbilanciamento generato dal robot danneggiato, in unione con il corrispondente controllore attivo in quel momento, al fine di misurare la correttezza di ciascuna delle diagnosi codificate dal genoma della fase di stima stessa. Quando l'algoritmo termina, l'ipotesi che maggiormente rispecchia il danno viene fornita alla fase di esplorazione.

I due algoritmi appena presentati differiscono dunque per alcune caratteristiche, una fra tutte è il fatto che il robot viene arrestato fino a quando non è pronto un controllore in grado di far fronte al malfunzionamento: questa caratteristica conferisce agli algoritmi la denotazione di *offline damage diagnosis and repair algorithm*, che li rende notevolmente più efficienti rispetto agli algoritmi cosiddetti *online*; il problema di eseguire un'operazione online consiste nel dover fare operazioni di testing ripetute sul robot fisico, con tutte le problematiche che ciò, come già illustrato, comporta.

3.2.2 Risultati

Hod Lipson e Josh C. Bongard hanno applicato gli algoritmi proposti all'ormai noto Starfish robot e anche a robot dalla diversa morfologia, come robot "esapedi" (robot dotati di sei arti, *hexapedal robots*), coprendo un ampio numero di possibili scenari: si va da perdite parziali o totali di un arto, fino a fallimenti a livello di sensori. L'unico scenario che non porta ad una corretta ipotesi di malfunzionamento è il caso in cui ci sia un malfunzionamento di un neurone appartenente allo strato nascosto: in questo caso di fallimento imprevisto l'algoritmo realizza un'ipotesi di danno approssimativa.²

In conclusione, questa tipologia di algoritmi è la prima in cui sia presente un flusso di informazioni continuo, automatico e bidirezionale tra la simulazione (intesa come evoluzione delle ipotesi e modellazione) e il robot fisico: difatti il robot fornisce informazioni sul suo stato attuale, utilizzate per aggiornare il modello, mentre il modello costituisce l'elemento per aggiornare i sistemi di controllo del robot. I vantaggi derivanti dall'impiego degli algoritmi sono dunque numerosi. In primo luogo, i tentativi di recupero via hardware sono un numero minimo, e pertanto sono ridotti di ben due ordini di grandezza rispetto agli algoritmi che si evolvono totalmente sul robot fisico. In secondo luogo, l'algoritmo non fa distinzione tra danni morfologici e malfunzionamen-

²Rif. J. Bongard, H. Lipson, *Automated Damage Diagnosis and Recovery for Remote Robotics*, Cornell University.

ti a livello di sistema di controllo, dunque può essere generalizzato per poter rispondere ad entrambe le esigenze. Inoltre, in alcuni casi l'algoritmo è in grado di produrre recupero di danni compositi.

3.3 Self-Reflection

Definiamo la *self-reflection* (o auto riflessione) come la capacità di esercitare introspezione e la predisposizione ad apprendere aspetti riguardanti la propria natura elementare, il proprio scopo e la propria essenza. L'elemento centrale di questa competenza è sicuramente l'introspezione, ossia l'esame della propria coscienza e dei propri pensieri e sentimenti, basandosi sull'osservazione dello stato mentale.

Come si evince dalla definizione, l'auto riflessione è una caratteristica propria dell'uomo; tuttavia è possibile utilizzare l'approccio introspettivo per applicazioni robotiche.

3.3.1 Applicazione in ambito di autodiagnosi e recupero

Rispetto al contesto di adattamento e recupero proposto nel paragrafo precedente, si cerca di attuare un'ulteriore miglioria: ottenere un robot che sia in grado di modellare il proprio sistema di controllo. Ma per quale motivo questa caratteristica dovrebbe rivelarsi utile? Le ragioni sono molteplici: ci sono aspetti del sistema di controllo che non possono essere descritti in maniera esplicita, pur essendo perfettamente nota l'architettura, in aggiunta il sistema di controllo potrebbe mutare in maniere impreviste a causa di fallimenti o cambiamenti ambientali. Modellare un controllore esistente in maniera manuale, senza la possibilità di accesso diretto, richiede inoltre tempo e sforzi (risorse), mentre la manipolazione diretta potrebbe richiedere un inatteso aumento di complessità, sia lato hardware che software. In alcuni casi infine, il sistema di controllo del robot è semplicemente inaccessibile.

La possibilità di modificare le prestazioni di un sistema di controllo esistente senza accedervi direttamente si rivela essere anche una strategia di adattamento sicuro; il sistema di controllo originale non viene mai modificato, dunque il suo comportamento può essere ripristinato in qualsiasi momento. Le conseguenze di questo processo potranno anche far luce sull'evoluzione di sistemi di controllo più intricati, come i sistemi nervosi biologici.

Base teorica

L'approccio si basa sull'assunzione di avere due sistemi di controllo coabitanti, uno "riflettente" sull'altro. Questa architettura è una forma di *Metacognizione*: è l'abilità di riflettere riguardo ad un proprio processo mentale e di autoregolarlo. I processi metacognitivi sono presenti nell'uomo, negli altri primati, e in pochi altri animali.

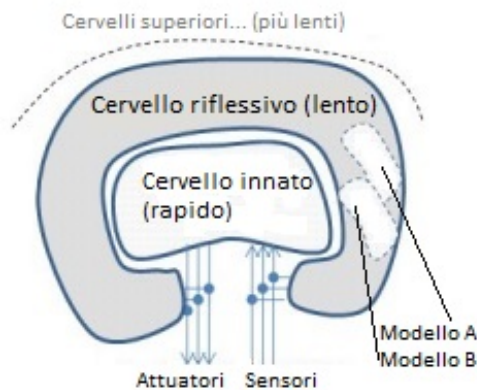


Figura 3.2: Architettura a cervelli annidati

L'informatico e scienziato statunitense Marvin Lee Minsky, specializzato nel campo dell'intelligenza artificiale, evidenzia che un cervello può essere meglio inteso come una "società di menti" (*society of minds*), le quali interagiscono tra loro.³ Egli propose un esperimento mentale che consiste nel dividere un

³Rif. M. Minsky, *The society of mind*, New York, 1988.

cervello in due parti A e B. Mentre le connessioni di input-output della parte A (che chiamiamo cervello A) sono collegate con il mondo esterno, quelle della parte B (o cervello B) sono collegate unicamente al cervello A: pertanto A è l'unico mondo visto da B. Come indicato da Minsky, il cervello B potrebbe dare il suo contributo al cervello A pur non avendo accesso al mondo reale, solo osservando l'attività di costui: Semplici domande, come “Stai ripetendo?”, “Ti senti meglio?” e “Cosa ne pensi?” potrebbero aiutare a produrre uno stato celebrale migliore nel mondo.⁴

È possibile realizzare la suddivisione proposta in un numero di parti superiore, si parla allora di catena di cervelli.⁵

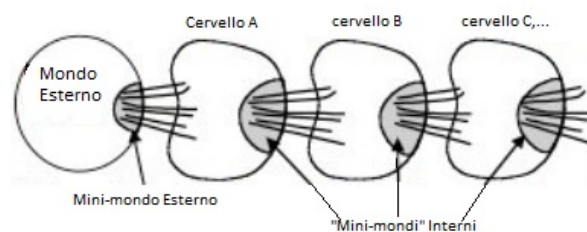


Figura 3.3: La catena di cervelli di Minsky

Come sottolineato da Minsky in uno dei suoi ultimi lavori, devono essere presenti alcuni cervelli che criticano le prestazioni degli altri cervelli o di alcuni loro insiemi. Si potrebbe anche essere in grado di identificare alcuni modi di pensare e riconfigurarli attivando o ricollegando certe aree del cervello.⁶ Per tentare di implementare un sistema come quello presentato da Minsky, è necessario domandarsi come un sistema critico possa identificare e manipolare un certo modo di pensare. Si ipotizza perciò che le menti dovrebbero essere in grado di eseguire una sorta di auto modellazione di altre menti come metodo di confronto e di speculazione sui modelli di attivazione.

⁴Cfr. M. Minsky, *The society of mind*, Simon and Schuster Inc., New York, 1988.

⁵Rif. M. Minsky, *Interior grounding, reflection, and self-consciousness*, Proceedings of International Conference on Brain, Mind and Society, Tohoku University-Japan, 2005.

⁶Cfr. M. Minsky, *The Emotion Machine*, Simon and Schuster Inc., New York, 2007.

Processo applicativo

Hod Lipson e Juan Cristobal Zagal hanno applicato questi concetti e implementato un processo per ottenere un comportamento flessibile attraverso l'autodiagnosi del sistema di controllo, l'adattamento e il recupero dei suoi malfunzionamenti.⁷ In figura è presentato uno schema descrittivo dell'intero processo:

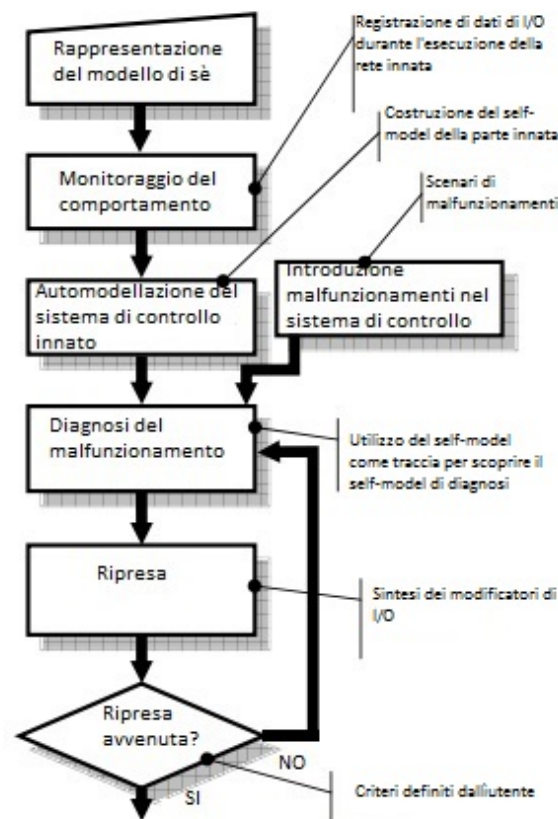


Figura 3.4: Diagramma di flusso del metodo che permette di ottenere un robot dal comportamento elastico attraverso autodiagnosi, adattamento e recupero del controllore

⁷Cfr. J. C. Zagal, H. Lipson, *Resilient Behavior through Controller Self-Diagnosis, Adaptation and Recovery*, Cornell University.

Analizziamo le linee guida della tecnica adottata, evidenziandone i caratteri principali:

- **Descrizione del robot e dell'ambiente:** viene utilizzato un robot dotato di 6 zampe su una superficie piana ricoperta da emettitori di luce rossa e blu in movimento; queste risorse luminose sono disposte inizialmente in maniera casuale sulla superficie e seguono differenti modalità di movimento. Il robot possiede 4 sensori per captare i colori specifici, divisi in coppie e allocati nelle zampe anteriori. Una volta che il robot viene dotato di un comportamento che gli permetta di muoversi in avanti, esso impara a seguire le sorgenti che producono luce blu e ad evitare coloro che producono luce rossa: questo comportamento è stato ottenuto attraverso l'evoluzione dei pesi sinaptici del sistema di controllo formato da una "innata" rete neurale ricorrente.
- **Fase di self-reflection e di automodellazione del sistema di controllo:** in questa fase il sistema di controllo riflessivo acquisisce un modello di sé del sistema di controllo innato; tornando alla formulazione di Minsky, ciò è equivalente a dire che il cervello B chieda al cervello A del suo modo di pensare. Il modello di sé è rappresentato da una rete neurale che ha lo stesso numero di nodi di ingresso-uscita della rete neurale innata, ma non necessariamente lo stesso numero di nodi nascosti. Inoltre vengono registrati i dati di I/O inerenti all'esecuzione "innata". Viene poi misurata la qualità di ciascun modello di controllore candidato ottenuto, valutando la sua abilità di riprodurre gli stessi schemi di input-output di quelli osservati durante l'operato della rete neurale innata (ossia del controllore innato).
- **Introduzione di malfunzionamenti nel controllore:** con il termine malfunzionamento inglobiamo il significato di guasto o danno, intendendo dunque in generale una qualche anomalia. I malfunzionamenti introdotti sono di diverse nature: consistono nella disconnessione di alcuni input dalla rete, nella disconnessione o nello scambio dei col-

legamenti sinaptici all'interno della rete, e anche nell'introduzione di perturbazioni costanti su specifici neuroni.

- **Diagnosi dei malfunzionamenti:** in questa fase il self-model generato dal sistema di controllo riflessivo viene utilizzato per diagnosticare il malfunzionamento introdotto dal sistema di controllo innato. Come già detto, il modello deriva dai dati collezionati durante il normale funzionamento del robot. Introduciamo quindi uno dei malfunzionamenti che abbiamo a disposizione nel controllore innato, lasciando che il robot continui ad operare nel suo ambiente. Naturalmente si crea a questo punto una discrepanza dalle relazioni senso-motorie che vengono registrate in questo momento, e quelle specificate dal self-model posseduto dalla parte riflessiva. Dal momento che il modello di sé fornisce buone spiegazioni comportamentali riguardo alle relazioni senso-motorie innate, ci si aspetta che le piccole deviazioni da questa soluzione abbiano un qualche potere esplicativo del malfunzionamento. Si può ipotizzare inoltre che la distribuzione topologica di queste variazioni possa avere un certo grado di correlazione con la reale sorgente della perturbazione stessa. Viene utilizzata perciò la seguente strategia per produrre nuovi self-model di diagnosi: dati i segnali senso-motori ottenuti durante il funzionamento perturbato, si inizia una nuova ricerca genetica, come quella illustrata nella fase di automodellazione del sistema di controllo. Una prima popolazione di self-model di diagnosi è fortemente costituita dal modello di sé ottimale, la quale viene poi evoluta nell'ottica di minimizzare la differenza tra gli output del modello di sé di diagnosi e quelli osservati durante il funzionamento del sistema di controllo innato danneggiato: se non è presente alcun danno, il modello di sé di diagnosi coinciderà con la popolazione iniziale, altrimenti la ricerca genetica porterà a un self-model di diagnosi tale da produrre i medesimi schemi motori di quelli eseguiti dalla rete neurale innata danneggiata. Una prima stima del grado di malfunzionamento è data dalla misura della variazione dei parametri.

- **Adattamento e recupero:** questo step si occupa del recupero delle funzionalità filtrando gli input e/o gli output del controllore innato, il quale è soggetto a diversi tipi di malfunzionamenti: l'idea è quella di ripristinare le funzionalità innate sintetizzando i modificatori di input/output attraverso il confronto tra il modello di sé ottimale e il modello di sé di diagnosi. La sintesi dei modificatori permette di realizzare la funzione di ripristino, la quale consente di identificare quali siano i pesi sinaptici da modificare e correggere. Questa procedura permette di valutare la qualità dei modificatori senza accedere ulteriormente al controllore innato.

Considerazioni

La procedura illustrata ha messo in evidenza un metodo di incremento dell'elasticità (intesa come capacità di recupero) di un robot utilizzando un processo di self-reflection che comprende autodiagnosi, adattamento e recupero. Tale approccio può essere sfruttato anche per il riutilizzo di hardware già esistente al fine di eseguire nuovi incarichi. Osservando inoltre l'intero processo in maniera più ampia, esso rappresenta un esempio di tecniche di identificazione dei sistemi, con le quali si rivela possibile inferire i parametri di un sistema di controllo: difatti, mentre un controllo adattativo mira a compensare dinamicamente un sistema in cui i parametri sono incerti, in questa situazione ci si trova ad affrontare i problemi derivanti dall'aggiunta di incertezza all'interno del controllore, ma, come già illustrato, permette di individuare anche un corposo insieme di vantaggi. Un ultimo aspetto particolarmente significativo riguarda l'integrazione con i sistemi già esistenti: è infatti possibile sovrapporre un sistema metacognitivo ad un altro sistema, aggiungendo la possibilità di monitorare, controllare ed espandere ulteriormente le funzionalità del sistema preesistente.

Conclusioni

In questo elaborato sono stati mostrati i principali metodi apprendimento e di consapevolezza di sé nell'ambito della robotica, con le relative considerazioni a carattere biologico ed evolutivo, e con esempi applicativi. Gli approcci presentati aprono la strada ad un vasto numero di possibili sviluppi futuri, principalmente in ambito di indagine e sicurezza. Possiamo immaginare scenari come l'esplorazione di fondali marini, dello Spazio, e di un'ampia gamma di luoghi inaccessibili all'uomo, oppure scenari di perlustrazione di luoghi pericolosi e potenzialmente fatali, come edifici colpiti da sisma, infrastrutture soggette a crolli oppure luoghi caratterizzati da particolari agenti atmosferici o ambientali: la possibilità di adattare i propri comportamenti a seguito di cambiamenti ambientali permette al robot di mantenere con continuità il proprio funzionamento; inoltre, a seguito di un eventuale danno o malfunzionamento, esso si rivela essere in grado di ripristinare il comportamento desiderato o perlomeno di effettuare una compensazione del danno, senza richiedere alcun tipo di riparazione o di intervento umano (il quale naturalmente in questo tipo di scenari è impraticabile).

Le possibilità derivanti dall'utilizzo delle tecnologie proposte sono vastissime e di elevato valore, poiché consentiranno di compiere azioni poco attuabili al momento, ma di grande impatto scientifico, tecnologico e sociale.

Bibliografia

A. Roli, *Robot Learning*, dispense del corso di Sistemi Intelligenti Robotici, a.a. 2013/2014.

P. Simon, *Too big to Ignore: Business Case for Big Data*, Wiley, 2013.

T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

Apprendimento automatico, Wikipedia.

R. Pfeifer C. Scheier, *Understanding Intelligence*, Bradford Books.

H. Lipson, *Building “self-aware” robots*, TED, 2007.

J. Bongard, H. Lipson, *Automatic Synthesis of Multiple Internal Models Through Active Exploration*, AAAI Fall Symposium: From Reactive to Anticipatory Cognitive Embodied Systems, November 2005.

J. Bongard, H. Lipson, *Automated Damage Diagnosis and Recovery for Remote Robotics*, IEEE International Conference on Robotics and Automation (ICRA04).

M. Minsky, *The society of mind*, New York, 1988.

M. Minsky, *Interior grounding, reflection, and self-consciousness.*, Proceedings of International Conference on Brain, Mind and Society, Tohoku University-Japan, 2005.

Cfr. M. Minsky, *The Emotion Machine*, Simon and Schuster Inc., New York, 2007.

J. C. Zagal, H. Lipson, *Resilient Behavior through Controller Self-Diagnosis, Adaptation and Recovery*, Performance Metrics for Intelligent Systems Workshop (PerMIS'09), Sept 21-23 2009, National Institute of Standards and Technology, Gaithersburg, Maryland USA.