ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

SCUOLA DI INGEGNERIA E ARCHITETTURA
Corso di laurea in
Ingegneria Elettronica, Informatica e Telecomunicazioni

Tecniche di fault tolerance in swarm robotics

Elaborata nel corso di Fondamenti di Informatica B

Relatore:
Prof. ANDREA ROLI

Presentata da:
JACOPO LONGHINI

ANNO ACCADEMICO 2013/2014
SESSIONE II

PAROLE CHIAVE

Swarm intelligence
Swarm robotics
Fault tolerance
Fault detection

A mio padre

Indice

Introduzione

La swarm robotics è lo studio di sistemi multi-robot decentralizzati costituiti da gruppi di piccoli robot relativamente semplici, che cooperando tra loro e interagendo con l' ambiente circostante, portano a termine compiti che un singolo componente del gruppo non riuscirebbe a svolgere. Tale studio è in continua evoluzione e progetta sistemi sempre più completi che potranno trovare concreta applicazione nella realtà data la loro alta potenzialità di utilizzo in svariati contesti. Robustezza, auto-organizzazione e adattamento sono alcune delle proprietà fondamentali che si cerca di ottenere in questi sistemi. Sono fonte di ispirazione per la ricerca nella swarm robotics lo studio dei modelli di comportamento emergente negli insetti sociali. Si avverte la necessità di creare sistemi sempre più affidabili e quindi tolleranti ai guasti che si possono presentare, per rendere il loro utilizzo sempre più efficiente e sicuro. In genere si ritiene che gli sciami robotici siano dotati di un' innata robustezza dovuta alla ridondanza delle entità che lo compongono, ma ciò non sempre si può riscontrare quindi è necessario approfondire ulteriormente tale aspetto. Questa tesi presenta e discute le sfide per raggiungere sistemi di sciami robotici affidabili e tolleranti ai guasti e quindi anche metodi per rilevare anomalie in essi, in modo tale che ipotetiche procedure per il recupero possano essere affrontate.
Nel primo capitolo verrà analizzato lo scenario di ricerca nel quale è inserita la problematica affrontata. Nel secondo capitolo verrà illustrato un metodo per avere un sistema di misura qualitativo per valutare l' affidabilità e la tolleranza ai guasti. Nel terzo capitolo verranno mostrati alcuni studi significativi e ricerche riguardanti tecniche che mirano a rendere gli sciami robotici sempre più tolleranti ai guasti. Verrà presentato uno studio di che si ispira al lampeggio delle lucciole per scovare robot guasti all' interno dello sciame, poi verrà preso in esame uno studio che

prende spunto dalla formazione del granuloma nel sistema immunitario ed una sua possibile applicazione alla robotica degli sciami, verrà presentata anche un' architettura che si basa sulla "motivazione" dei robot per organizzare il loro lavoro in maniera efficiente per riuscire a portare a termine un compito nonostante malfunzionamenti. Nel capitolo conclusivo verranno presentati alcuni possibili sviluppi futuri inerenti la tematica qui trattata.

Capitolo 1

Background

1.1 Swarm intelligence

La swarm intelligence è stata introdotta per la prima volta da Beni e Wang [4], la swarm intelligence si può definire come l' intelligenza collettiva emergente da un gruppo di semplici unità. Essa è una nuova branca dello studio dell' intelligenza artificiale; è connotata da un approccio interdisciplinare che si avvale degli apporti dei campi della biologia, robotica e informatica che contribuiscono attivamente per la creazione di sistemi di swarm intelligence. Quando si parla di swarm intelligence si fa riferimento a sistemi di controllo decentralizzati e alla semplicità delle unità che compongono lo sciame composto da una moltitudine di individui, possibilmente identici o con pochi tipi di variazione. Alcuni esempi di sistemi di swarm intelligence presenti in natura sono quelli composti dai cosiddetti insetti sociali come le colonie di formiche, gli stormi di uccelli e i banchi di pesci. In essi possiamo notare che i componenti di ogni gruppo sono omogenei, appartenenti alla stessa specie, e seguono semplici regole. Il comportamento emergente che risulta dall' insieme dei componenti è sorprendente poiché riescono a gestire situazioni che un singolo componente non potrebbe fronteggiare nonostante non sia presente alcun sistema centralizzato e che il comportamento dei singoli dipenda solo da informazioni locali, che hanno in risposta a degli stimoli, senza avere la conoscenza globale del sistema.

1.2 Swarm robotics

La swarm robotics è un' area in pieno sviluppo della robotica, è un approccio che si basa sui principi della swarm intelligence, per coordinare e organizzare sistemi multi-robot composti da robot relativamente semplici. La swarm robotics quindi utilizza un approccio decentralizzato nel quale il comportamento emergente desiderato deriva dall' interazione e comunicazione locale tra i robot e ambiente. Lo scopo di questa area di ricerca è la realizzazione di sciami comprendenti uno svariato numero di robot, autonomi, omogenei o con lievi variazioni, e con meccanismi di comunicazione ed interazione locale. Le caratteristiche principali sono:

- robustezza: che è il grado con il quale un sistema può ancora funzionare in presenza di anomalie o guasti
- flessibilità: la capacità di adattarsi a nuovi e diversi ambienti
- scalabilità: la capacità di espandere un meccanismo auto organizzato per supportare grandi o piccoli gruppi di individui senza modificare notevolmente le prestazioni.

Le motivazioni che spingono allo studio della swarm robotics sono la possibilità di impiegare questi sciami in svariate situazioni reali, ad esempio, per sostituire l' uomo o affiancarlo in alcune attività. Gli sciami robotici hanno molti vantaggi rispetto ai singoli robot, essi possono affrontare compiti più velocemente grazie al loro parallelismo, sono una soluzione economica per complesse applicazioni le quali possono essere affrontate con più robot specializzati piuttosto che un costoso robot multifunzionale, e a volte grazie alla ridondanza hanno un grado maggiore di affidabilità e robustezza; ma spesso gli sciami robotici presentano una varietà di problemi, che non si manifestano nei sistemi composti da un singolo robot. Il funzionamento di questi sistemi è spesso influenzato anche dall' ambiente in cui operano. Durante il loro funzionamento potrebbero verificarsi comportamenti indesiderati o anomalie, che ad esempio possono essere causati da guasti nel sistema o dall' interazione con l'ambiente. Pertanto, la capacità di tollerare i guasti nonché le interferenze dall'ambiente costituiscono un' area di ricerca importante nella swarm robotics.

1.3 Fault tolerance

Qualsiasi sistema perfettamente progettato, testato e con un ambiente operativo ragionevolmente ben definito, potrebbe ancora incorrere in alcuni comportamenti indesiderati a causa di una serie di motivi. La fault tolerance è la capacità del sistema di continuare a operare nonostante siano presenti dei guasti, è quindi necessario esaminare le prestazioni del sistema in modo tale che qualunque deviazione rispetto al comportamento atteso sia individuata e gestita. Come sostiene Parker [11], affinché la progettazione di un sistema multi-robotico sia affidabile e robusto è necessario l' approfondimento di alcune questioni quali:

- Come (o se) intercettare un guasto di un robot;
- Come (o se) diagnosticare e identificare il guasto di un robot;
- Come (o se) comportarsi in caso di guasti in un robot.

Il sistema deve essere progettato tenendo in considerazione le sopraelencate questioni e che i guasti potrebbero accadere inaspettatamente. Ci sono molte ragioni per cui i sistemi potrebbero subire guasti che non sempre sono prevedibili, alcune possono derivare da cause interne dipendenti dall' hardware e o dal software appartenenti a ciascun robot, altre possono derivare da fattori esterni dovute all' ambiente circostante. Alcune cause più comuni sono:

- Malfunzionamenti individuali dei robot: i singoli robot, essendo costituiti da diversi componenti, hanno potenzialmente molti punti deboli, più sono i componenti utilizzati per la loro costruzione più c'è possibilità che essi subiscano dei guasti, le cause di questi guasti possono essere imputate a difetti di progettazione del software, a guasti fisici o a errori derivanti dall' errato controllo umano.
- Comportamenti locali che sono globalmente incoerenti: gli sciami robotici sono sistemi distribuiti composti da entità individuali che seguono comportamenti derivanti da una percezione locale, nessuna entità ha la completa conoscenza dell' intero sistema, azioni intraprese con informazioni locali, se non correttamente gestite, potrebbero portare ad una soluzione globale incoerente (non attesa),

poiché se alcuni approcci al controllo locale sono adatti per certe situazioni potrebbe accadere che eventi inaspettati o progettazioni incomplete portino al fallimento dei meccanismi di coordinazione, quindi il comportamento emergente che ne risulta non è coerente con l' obbiettivo del sistema collettivo.

- Errori software: complessi sistemi software non sempre sono facili da testate, e sicuramente il software non può essere progettato per ogni possibile situazione o ambiente in cui i robot opereranno, quindi per ottenere soluzioni robuste è importante fornire ai robot un' abilità che consenta loro di sapersi rapportare con eventi non modellati.

- Errori nella comunicazione: alcuni sistemi robotici possono aumentare le loro prestazioni scambiando informazioni parziali riguardo il loro stato o ambiente ma devono essere progettati adeguatamente a operare anche in assenza di comunicazioni.

Si è visto che gli sciami robotici possono incappare in diversi tipi di guasti durante il loro impiego, la squadra deve avere quindi alcuni sistemi per compensare i guasti che possono accadere a qualche unità, questo spetta alla progettazione del sistema che deve assicurarsi che lo sciame operi correttamente anche in caso di guasto e che il team possa raggiungere con successo i propri obbiettivi.

Capitolo 2

Valutazione e classificazione degli errori

Per realizzare sciami robotici affidabili è necessario effettuare un' accurata analisi del pericolo, ovvero analizzare condizioni, eventi o circostanze che potrebbero causare o contribuire ad un comportamento non desiderato, quest' analisi non sempre è di immediata realizzazione, poichè non esiste un metodo formale per realizzarla, ma deve essere fatta ipotizzando ogni possibile rischio capace di compromettere, totalmente o parzialmente, le funzionalità dell' entità in analisi. Nel nostro studio questa fase è resa più agevole grazie alle caratteristiche degli sciami robotici che sono costituiti da semplici unità che seguono poche regole. Winfield e Nembrini [13] propongono una tecnica di analisi come la FMEA (Failure Mode and Effect Analysis) [8] per valutare la tolleranza ai guasti (fault tolerance) di un sistema auto organizzante.

2.1 Caso di studio

Tale tecnica viene utilizzata in un caso di studio dove è impiegato uno sciame di robot per un compito di contenimento o di incapsulazione (Fig. 2.1) per ottenere questo comportamento emergente (emergent behaviour) vegono utilizzati dei semplici robot dotati di comunicazione wireless, con raggio di azione limitato, collegata al suo movimento, in modo tale che i robot nello sciame rimangano uniti. L' algoritmo fa si che ogni robot diffonda (broadcast) periodicamente il suo ID e gli ID

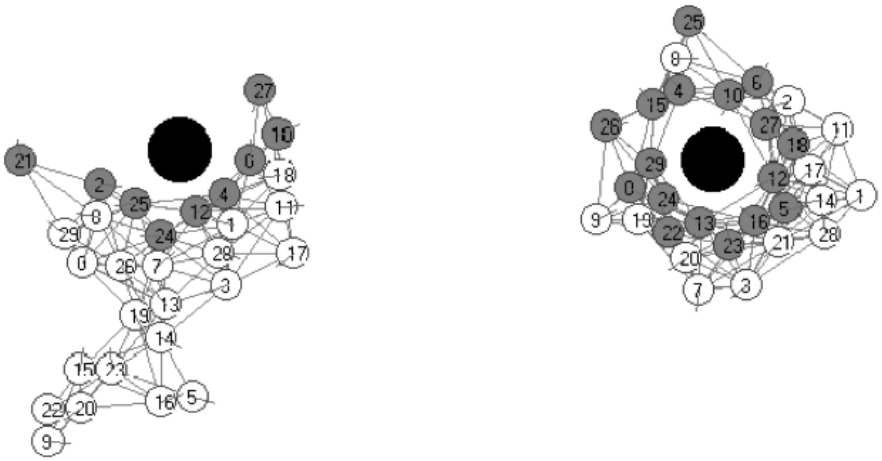

Figura 2.1: Contenimento di un obbiettivo: a sinistra l' avvicinamento, a destra contenimento avvenuto

dei suo più stretti vicini, tale comunicazione verrà ricevuta dai robot in prossimità, se un robot perde la connessione con un altro robot, e perciò non riceve un messaggio con il suo ID, e il numero rimanente dei robot vicini è minore o uguale ad una determinata soglia β , allora si può pensare che il robot si stia allontanando dal gruppo e quindi invertirà la sua rotta di 180°, se invece il suo numero di interconnessioni con altri robot aumenta, il robot va in una nuova direzione casuale. Ogni robot è anche equipaggiato con sensori di prossimità che gli impediscono di effettuare collisioni e con un sensore a lungo raggio in grado di avvertire una fonte di segnale infrarossi (beacon sensor) che una volta avvertita modifica il valore soglia β in modo tale da ottenere un comportamento emergente che guidi lo sciame all' obbiettivo. Questo algoritmo è stato testato tramite simulazioni al computer ed anche da robot fisici come i Linuxbot (Fig. 2.2)

In sintesi tale progettazione dei robot presenta cinque comportamenti emergenti:

- aggregazione dello sciame
- rete ad-hoc coerente

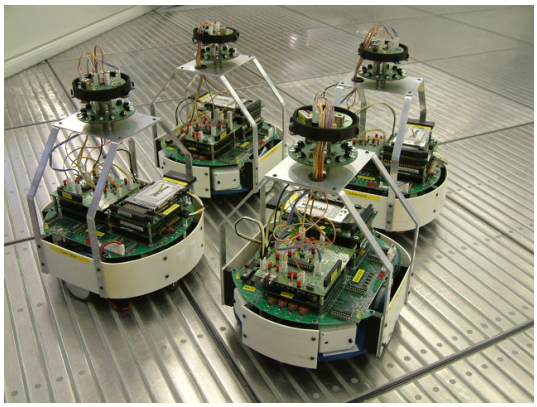

Figura 2.2: Linuxbot

- orientamento verso la fonte (beacon taxis)
- evitare gli ostacoli
- accerchiare l' obbiettivo

La tecnica di analisi (FMEA) è stata utilizzata anche da Bjerknes and Winfield [5] che hanno utilizzato uno sciame di e-puck robot (Fig. 2.3) equipaggiati con equivalenti sensori che seguono un algoritmo simi-

Figura 2.3: E-puck

le a quello sopra descritto, ma eseguono un compito diverso: invece di accerchiare un obbiettivo lo raggiungono.

2.2 Analisi dei guasti

Ora che abbiamo introdotto il caso di studio e le sue principali proprietà possiamo iniziare a valutare gli errori secondo il metodo sopracitato dell' FMEA , secondo cui bisogna identificare tutti i possibili pericoli, noi andremo ad analizzare quelli che vengono definiti interni poiché derivano da guasti del robot o da guasti dei suoi sottosistemi, poi si procede ad analizzare gli effetti di questi pericoli e come essi possono compromettere i comportamenti dello sciame.
I pericoli possono essere sintetizzati in questa tabella:

Pericolo	Descrizione
H_1	Guasto motore
H_2	Guasto comunicazione
H_3	Guasto sensore di prossimità
H_4	Guasto sensore obbiettivo
H_5	Guasto dei sistemi di controllo
H_6	Guasto di tutti i sistemi

Descriviamo di seguito i vari pericoli ed i possibili effetti sul sistema cercando di stabilire anche un livello di pericolosità attraverso un' analisi qualitativa.

Pericolo H_1 : Guasto motore
In questo caso c'è un guasto del motore o del sistema che si occupa del movimento del robot impedendo il moto parzialmente, quando, ad esempio rimane in funzione una ruota sola vincolando il robot a girare su se stesso, o totalmente non facendolo muovere affatto. In queste condizioni tutti gli altri apparati del robot continuano a funzionare e quindi a contribuire ai comportamenti dello sciame, come l' aggregazione e la comunicazione della sua posizione, che potrebbero essere influenzati negativamente da uno o più robot affetti da questo tipo di guasto ancorando di fatto lo sciame in una posizione fissa e impedendo lo svolgimento del

compito. Tutto ciò costituisce un grave pericolo poiché può compromettere seriamente l' accerchiamento dell' obbiettivo, che è il comportamento emergente principale. L' effetto del guasto viene quindi contrassegnato con una lettera maiuscola E_1 per sottolineare la sua gravità.

Pericolo H_2: Guasto comunicazione
Descrive il guasto del sistema di comunicazione wireless di cui sono dotati i robot e impedisce l' invio e la ricezione di messaggi, disconnettendo i robot dallo sciame. I robot affetti da questo tipo di guasto vagheranno all'interno dell' ambiente in maniera casuale non contribuendo più al comportamento emergente dello sciame ma senza arrecare danni, essi saranno considerati dagli altri robot come ostacoli mobili da evitare.
L' effetto di questo guasto non è molto grave e verrà indicato con una lettera minuscola e_2 .

Pericolo H_3 : Guasto sensore di prossimità
Il guasto del sensore di prossimità ha effetti relativamente gravi, poiché il robot che ne è affetto rischierà di collidere con elementi dell' ambiente circostante o con altri robot che hanno lo stesso guasto, ma non compromette il comportamento emergente dello sciame anche se potrebbe collidere con l' obbiettivo piuttosto che accerchiarlo, ma solo in particolari circostanze potrebbe essere considerato grave.
Nel nostro caso di studio verrà indicato con una lettera minuscola e_3 .

Pericolo H_4 : Guasto sensore obbiettivo
Questo guasto ha un effetto quasi trascurabile, al massimo potrebbe rallentare il completamento del compito da parte dello sciame ma non comprometterne lo scopo finale. Per questo pericolo non si riscontrata nessun effetto.

Pericolo H_5 : Guasto sistemi di controllo
Gli effetti di questo guasto possono essere ricondotti agli effetti dei guasti visti in precedenza. La struttura dei componenti dello sciame robotico è spesso semplice di conseguenza è semplice anche il suo sistema di controllo, l' eventuale guasto del sistema di controllo, spesso localizzato a livello software, potrebbe lasciare il robot nello stato di avanzamento, pertanto continuerebbe ad avanzare senza avere interazioni con gli altri.

Si può quindi collocarlo nell' effetto e_2 esaminato in precedenza; se invece il problema riguarda l' apparato motorio si ricade nel caso dell' effetto E_1 definito in precedenza.

Pericolo H_6 : Guasto di tutti i sistemi
Una causa di questo guasto potrebbe essere un problema nell' impianto di alimentazione del robot, che potrebbe impedire l' accensione dello stesso. I robot affetti da tale guasto paradossalmente hanno una minima se non nulla influenza sul comportamento emergente dello sciame, poiché verranno trattati dagli altri componenti come elementi statici e quindi da evitare.

2.3 Considerazioni

Dall' analisi si evince che gli effetti dei vari guasti sono principalmente tre:

- E_1 è il più grave poiché compromette la riuscita del compito, riguarda il guasto della parte relativa al moto del robot,
- e_2 è l' effetto dove c'è una perdita di un robot nell' ambiente circostante,
- e_3, riguarda la collisione di un robot con un ostacolo o con l' obbiettivo (Beacon).

Da questo studio è emerso che non tutti i guasti impediscono il completamento di un compito da parte di uno sciame di robot, in particolare, come possiamo vedere nella tabella riassuntiva, l' effetto piu grave E_1 si manifesta 6 volte su 30 possibili combinazioni, mentre per ben 15 volte su 30 non si verificano effetti nocivi, per le rimanenti 9 riscontriamo solo effetti minori.

Comportamento dello sciame	H_1	H_2	H_3	H_4	H_5	H_6
Aggregazione	-	e_2	-	-	e_2	-
rete ad-hoc	-	e_2	-	-	e_2	-
orientamento obbiettivo	E_1	e_2	-	-	E_1	-
evitare gli ostacoli	E_1	e_2	e_3	-	E_1	-
accerchiare l' obbiettivo	E_1	e_2	e_3	-	E_1	-

In sintesi questa analisi (FMEA) dimostra che lo sciame di robot è abbastanza tollerante a diversi tipi di guasti, sorprendente è il fatto che il guasto di tutti i sistemi di uno o piu robot non impedisca affatto il compimento del lavoro assegnato allo sciame, mentre i casi di guasti parziali, come il guasto del motore, possono compromettere gravemente la realizzazione del compito, contrariamente a quello che si potrebbe pensare senza avere un metodo di analisi. Quindi nello studio della tolleranza ai guasti deve essere anche considerato l' effetto di guasti o errori parziali.

Capitolo 3

Strategie di fault tolerance

3.1 Tecnica delle lucciole

Questa metodologia mira a scovare unità dello sciame danneggiate da un guasto e non più operative mediante gli altri componenti ancora funzionanti che possono quindi intraprendere azioni per prevenire ulteriori danni o se in grado di riparare il robot non più funzionante mettere in atto azioni per rimediare al guasto. In questo modo non è più il robot stesso che si deve preoccupare di segnalare un guasto, che in alcuni casi potrebbe compromettere la maggior parte delle sue funzionalità e quindi impedire azioni di questo genere, ma è il gruppo a cui appartiene che cerca di individuarlo, quindi possiamo parlare di una sorveglianza che viene dall' esterno.
La tecnica che andremo a presentare è stata descritta da Christesen, Grady e Dorigo [3], ricercatori molto attivi nel campo della Swarm intelligence e Swarm robotics , che hanno preso parte a svariati progetti, ad esempio Swarmanoid [2], riguardanti questo argomento. L' approccio di questa metodologia è ispirato al comportamento delle lucciole tropicali del Sud-est asiatico, che riescono a sincronizzare la luce che emettono con il resto dello sciame, quindi è coerente con i parametri dello studio della Swarm intelligence che nasce dall' osservazione di comportamenti collettivi di sistemi auto-organizzanti presenti in natura come ad esempio quelli di insetti sociali quali le api e le formiche.

3.1.1 S-bots

Lo sciame robotico che viene considerato in questo studio è composto da S-bot, robot mobili autonomi in grado di creare una connessione fisica tra di loro. Gli S-bot [9] sono stati presentati nel progetto iniziato nel 2001 Swarm-Bots [1] e nel corso degli anni sono stati utilizzati in diversi studi sull' intelligienza e robotica degli sciami, subendo diverse migliorie. I robot (Fig. 3.1) sono equipaggiati con diversi sensori, con una telecamera omnidirezionale, con un sistema di aggancio capace di afferrare altri S-bot, e con led RGB posizionati tutt' intorno al robot (Fig. 3.2). Il software che controlla l' S-bot lo possiamo descrivere come un sistema ciclico:

- lettura dei dati dei sensori
- elaborazione dei dati rilevati
- esecuzione di azioni in base ai dati raccolti

Il componente più importante in questo caso di studio è la telecamera, che come detto in precedenza, ha un campo visivo omnidirezionale (di 360°) capace di registrare immagini a colori di 640x480 pixel, il processore interno del robot è in grado di elaborare l' immagine catturata e riesce a riconoscere oggetti in base al loro colore, nel nostro caso i led colorati degli altri robot, il raggio di azione per il quale questo processo è possibile è di 50cm.
Oltre ai robot fisici è stato utilizzato un software di simulazione. Il software è capace di ricreare in un ambiente virtuale gli S-bot, in questo modo si possono condurre esperimenti utilizzando svariate unità che non possono essere realmente disponibili, ed è possibile condurre più volte lo stesso esperimento con molta facilità in poco tempo, il software permettere anche uno studio statistico con la moltitudine dei dati raccolti.

3.1.2 Metodo di comunicazione

Il concetto principale di questa tecnica è quello di trasmettere periodicamente un segnale rappresentante lo stato di salute dell' unità che lo genera in modo tale che gli altri componenti dello sciame possano identificare eventuali anomalie quando il periodo di questo segnale non rispetta

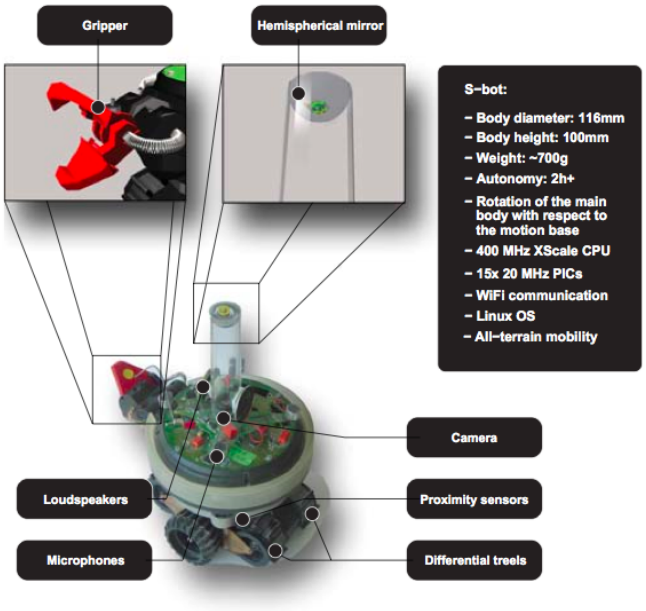

Figura 3.1: S-bot

le linee guida stabilite a priori.
Per mettere in atto questa strategia si possono utilizzare vari mezzi di comunicazione e/o di segnalazione, come ad esempio un segnale sonoro, radio, oppure luminoso. Il segnale sonoro potrebbe essere una soluzione praticabile ma siccome i robot devono essere in grado di localizzare in maniera precisa la posizione del robot guasto, bisognerebbe implementare sul robot microfoni direzionali capaci di misurare la posizione in base alla differenza di tempo con cui ricevono il segnale sonoro ma è un processo abbastanza difficile e a volte poco preciso. Un' altra soluzione potrebbe essere quella del segnale radio, ma anche qui si riscontra il problema della localizzazione dell' unità guasta, per determinare la posizione di una sorgente radio si potrebbero utilizzare antenne direzionali ma non sempre i robot ne sono equipaggiati, ad esempio gli S-bot non ne sono provvisti. Utilizzando i sensori integrati negli S-bot possiamo vedere come un segnale luminoso può essere captato dalla telecamera a bordo e come questa può rilevare implicitamente sia il segnale che la posizione

Figura 3.2: S-bot in funzione, si può notare come sono collocati i led RGB

della sua sorgente.

3.1.3 Protocollo di comunicazione

Determinato il sistema di comunicazione maggiormente sfruttabile nel nostro caso, dobbiamo analizzare come il segnale luminoso deve essere trasmesso. Prendiamo in esame tre protocolli di comunicazione per determinare il più adatto al nostro scopo.

- Lampeggio non sincronizzato: in questo protocollo il robot potrebbe ad esempio, lampeggiare alternando due colori dei led RGB senza interruzioni, un robot che incorre in un guasto smette di lampeggiare oppure i suoi led rimangono accesi senza cambiare colore. Con questo protocollo però abbiamo un grande spreco di risorse, poiché tutti i componenti dello sciame devono continuamente vigilare sui loro compagni, esiste anche un limite fisico, a causa della

costruzione degli S-bot, durante il moto la loro telecamera difficilmente realizzerà immagini stabili e potrebbe creare difficoltà nello stabilire a chi appartengano i led rilevati. Per i motivi esposti il lampeggio non sincronizzato risulta di difficile implementazione nel nostro caso di studio.

- Protocollo Ping-Pong: un robot lancia un segnale, che chiameremo Ping, e i suoi vicini devono rispondere con un contro segnale, che chiameremo Pong, se qualche robot non risponde al segnale inviato potrebbe aver subito un guasto. Rispetto al protocollo precedente il robot non deve continuamente controllare i robot vicini ma soltanto per un momento dopo che ha inviato il segnale Ping, ma anche questo metodo potrebbe avere delle debolezze, ad esempio se un S-bot lancia un segnale Ping ed un altro essendo lontano oppure per un istante coperto da qualche ostacolo non sarà in grado di rispondere con un segnale Pong, ma se il primo che ha inviato il messaggio è in grado di vedere quest' ultimo che non ha risposto penserà che sia incorso in un guasto. Un' altra circostanza che potrebbe verificarsi è qualla in cui più di un robot manda un segnale di Ping e i robot vicini rispondono con un segnale di Pong ma trovadosi a far fronte ad una doppia richiesta manterrano accesso più a lungo i loro LED rendendo cosi difficile la distinzione tra una doppia risposta consecutiva e un guasto che inibisce lo spegnimento degli stessi. Anche il protocollo Ping-Pong non ha le caratteristiche necessarie per essere applicato nel nostro caso di studio.

- Lampeggio sincronizzato: questo protocollo è quello che si ispira al lampeggiamento sincronizzato delle lucciole, tale protocollo può risolvere i problemi riscontrati con i protocolli precedenti, ad esempio il robot deve analizzare solamente un fotogramma quello in cui tutti i robot si illuminano, potendo quindi rilevare i robot guasti che in quell' istante non lampeggeranno affatto o presenteranno un colore diverso. Non ci dobbiamo preoccupare neanche del problema presente nel secondo protocollo quando non si riesce a rilevare la risposta, poiché ora tutti lampeggiando all' unisono e come se mandassero un segnale di Ping e uno di Pong contemporeaneamente.

Il protocollo del lampeggio sincronizzato risulta idoneo ad essere adottato nel nostro caso studio.

3.1.4 Sincronizzazione

La sfida più importante in questa tecnica di tolleranza ai guasti è far sincronizzare il lampeggiamento del segnale che indica lo stato di salute dei robot per poter sfruttare i vantaggi del protocollo di comunicazione del lampeggio sincronizzato. Per realizzare la sincronizzazione seguiremo lo schema degli oscillatori ad impulsi accoppiati (Pulse coupled oscillators), nella quale un oscillatore, nel nostro caso la luce emessa, influenza un altro oscillatore tramite piccoli e periodici impulsi fino a sincronizzarsi, tale metodologia è molto frequente in natura, ad esempio nelle cellule cardiache. Gli oscillatori presentano uno stato interno che cresce linearmente con il tempo; quando questo stato, che chiameremo stato d' attivazione, arriva ad una certa soglia i led lampeggiano per un istante e lo stato d'attivazione torna a zero per poi raggiungere di nuovo la soglia, ma se viene captato un lampeggio da parte di un' altra entità allora il suo stato di attivazione aumenta di un una certa costante in questo modo in un certo lasso di tempo gli stati d' attivazione andranno a pari passo e quindi si sincronizzeranno. Il meccanismo è descritto dal grafico in figura (3.3). Nei sistemi di comunicazione non è sempre cosi semplice ottenere una perfetta sincronizzazione, poiché ci sono molte variabili in gioco, come ad esempio i ritardi che possono intercorrere nella maggior parte delle fasi durante l' esecuzione dell' algoritmo, ma grazie alla semplicità del nostro caso di studio possiamo non considerare molte delle problematiche che in generale si riscontrano.

Nel nostro caso quindi i robot integreranno questo meccanismo, interagendo con gli altri componenti, ed al raggiungimento della soglia accenderanno i LED. Per realizzare questo metodo si è usato un modello matematico discreto, atto ad essere calcolato dai robot.

$$x_i\ (n+1) = x_i\ (n) + \frac{1}{T} + \epsilon \alpha_i(n) h(x_i(n))$$

Nel modello x_i (n) rappresenta lo stato di attivazione del robot i-esimo nel ciclo di computazione n, mentre T rappresenta il periodo tra due lampeggi, l' ultima parte dell' equazione si occupa di sincronizzare

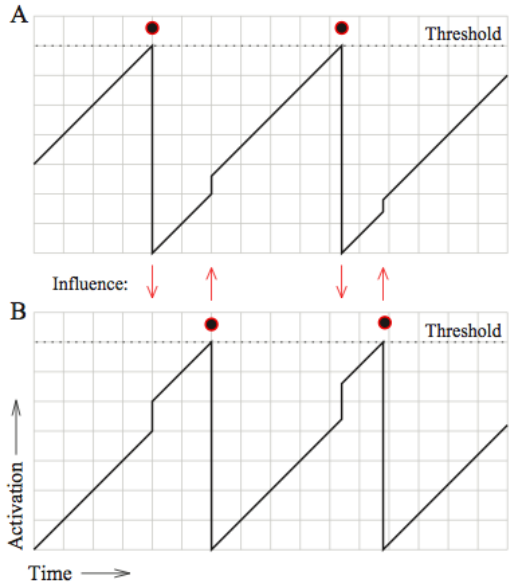

Figura 3.3: Grafico che mostra come avviene la sincronizzazione degli oscillatori, che salgono fino ad un livello soglia (Threshold), e accendono i led, puntino rosso, durante il loro ciclo vengo influenzati da altri oscillatori e se rilevano che in un dato istante è scattato un altro oscillatore allora aumentano il loro stato di attivazione di una certa costante ϵ per arrivare prima alla soglia.

gli impulsi dove ϵ è una costante di accoppiamento α è il numero di robot che hanno lampeggiato nei pressi dell' iesimo, e h(x) è una funzione lineare per gli impulsi accoppiati. Quando x_i (n) supera il valore 1 il robot lampeggia e il suo stato di attivazione torna a zero. Per gli S-bot è stato calcolato un tempo di accensione dei led per una durata di 0.75s.

Un fattore da non trascurare è il tempo di sincronizzazione dei segnali luminosi dello sciame, il quale può variare in base al valore della costante ϵ, alla lunghezza del periodo T, alla densità dello sciame, al numero delle unità e al movimento dello sciame. Grazie al software di simulazione si sono potuti studiare facilmente i comportamenti dello sciame in diverse situazioni ed è stato più agevole ripetere gli esperimenti più volte in mo-

do da trovare tendenze nel tempo di sincronizzazione.
Dalle simulazioni effettuate si è constatato che quando i robot sono in movimento riescono a sincronizzarsi prima rispetto agli esperimenti dove i robot rimangono fermi in una posizione assegnata, poiché questi ultimi sono più esposti alla propagazione ad onda del segnale come si vede in figura (3.4). Tale fenomeno accade a causa del ritardo dalla rilevazione del segnale all' accensione dei led, problema che viene meno quando i robot si muovono poichè essi stessi interrompono questo meccanismo di onda essendo portatori del segnale e quindi muovendosi in maniera casuale riescono ad annullare le distanze tra le varie unità. Ad esempio se un robot si sposta dal S-bot che genera l' onda verso quelli che sono più distanti, muovendosi e quindi accorciando le distanze, fa si che questi accendano i led prima rispetto ad una situazione statica. Il tempo di sincronizzazione è in generale maggiore se sono presenti degli ostacoli.

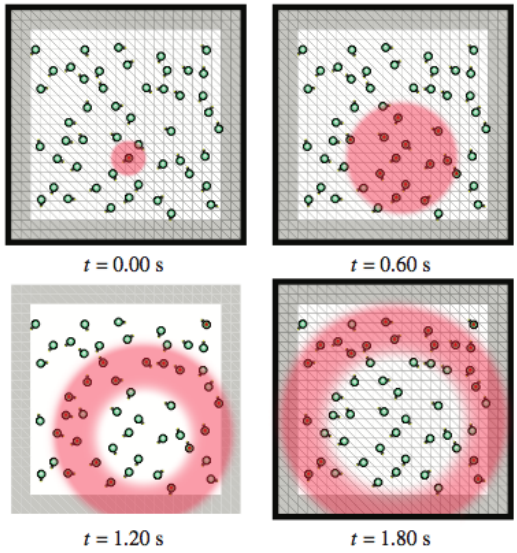

Figura 3.4: L' onda di propagazione del segnale, dal S-Bot centrale (in alto a sinistra) a quelli esterni

Per quanto riguarda le altre variabili che determinano il tempo di sincronizzazione tramite la ripetizione di diverse simulazioni sono stati elaborati dei grafici, si riportano i più significativi.

In entrambi i grafici distinguiamo le due situazioni sopracitate, in rosso sono riportati i risultati condotti mentre lo sciame è in movimento, in verde quelli in cui è statico, si può notare che il risultato è coerente a quello che è stato detto, cioè i robot in movimento hanno sempre un tempo di sincronizzazione minore. Nei grafici è riportata anche la deviazione standard, che è uno modo per esprimere la dispersione dei dati intorno ad ogni media rappresentata nei grafici.

Nel primo grafico (3.5), dove ogni esperimento è stato ripetuto cento volte, vediamo come varia il tempo medio di sincronizzazione in base al numero di unità presenti nello sciame, c'è una relazione lineare che mette in correlazione il numero di unità e il tempo totale di sincronizzazione, la densità dei robot era stata fissata a priori.

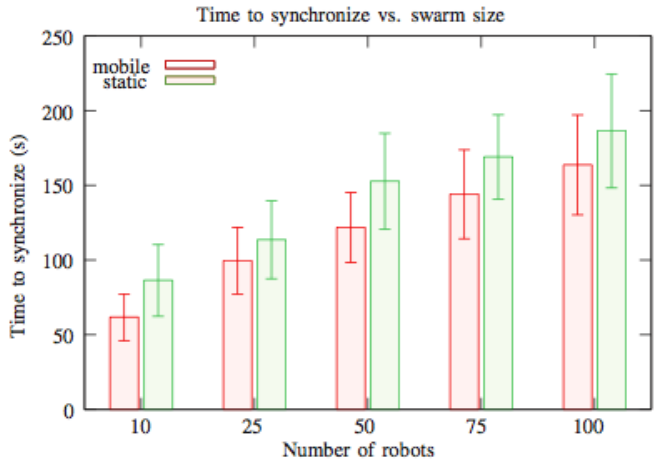

Figura 3.5: Tabella che mette in relazione il tempo di sincronizzazione e il numero di robot

Nel secondo grafico (3.6) osserviamo come varia il tempo medio di sincronizzazione in base alla densità dello sciame, sciami più densi tendono a sincronizzarsi prima, a causa della vicinanza tra le unità che prendono parte al meccanismo.

Questi risultati sperimentali sono stati confermati anche dagli esperimenti condotti con unità concrete, convalidando il modello di sincronizzazione che era stato scelto.

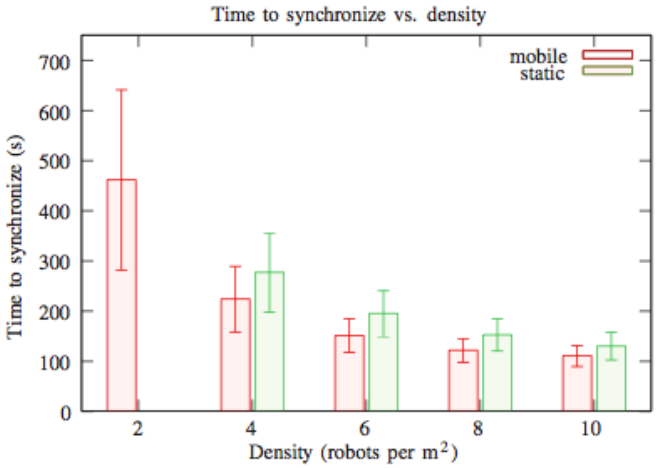

Figura 3.6: Tabella che mette in relazione il tempo di sincronizzazione e la densità di robot

3.1.5 Rilevamento dei guasti

Vediamo ora come il protocollo di comunicazione del lampeggio sincronizzato sia utilizzabile nel rilevamento degli S-Bot che hanno subito un guasto. Un robot può smettere di lampeggiare, e quindi richiedere assistenza, quando si rende conto che è incorso in un guasto, ad esempio un bug nel software, o quando si scollega l' alimentazione del robot e quindi l' intero sistema viene danneggiato. Gli altri componenti dello sciame si accorgono che il robot ha smesso di funzionare e quindi intervengono per rimediare al guasto.

Nella fase iniziale quando i robot sono in procinto di sincronizzarsi, o quando un robot proveniente da un altro ambiente non è ancora sincronizzato con lo sciame del nuovo ambiente, naturalmente i loro stati di attivazione non saranno sin dal primo momento equivalenti e quindi non sono nel complesso sincronizzati. Una strategia per risolvere questo problema potrebbe essere quella con cui un robot che avvista altri robot non sincronizzati, non pensi subito che essi siano incorsi in un problema. I robot "sospettosi", cioè quelli che vedono altri robot non sincronizzati li considerano potenzialmente guasti, e chiameremo questi ultimi candidati. Se il candidato lampeggia prima del sospettoso allora questi due non

sono sincronizzati, ma se il sospettoso lampeggia per due volte prima che il candidato lampeggi, allora il candidato viene considerato come guasto. Si può verificare un caso limite quando i robot sono ancora in fase di sincronizzazione, può accadere che un robot lampeggi due volte prima che un altro robot lampeggi anche se quest' ultimo non è guasto, per non incappare in errori di questo tipo il robot sospettoso lampeggia tre volte e in assenza di un riscontro considera il candidato come guasto. Con questi accorgimenti gli S-bot possono rilevare il guasto anche se il robot candidato non lampeggia affatto ma rimanendo con i led accesi. Nel caso degli S-bot il raggio d' azione della telecamera è limitato a 50cm, e i robot in movimento possono continuamente entrare e uscire da questo campo visivo, perciò i robot che diventano sospettosi, si fermano per qualche istante per determinare se il candidato è guasto o no. L' algoritmo appena descritto potrebbe indebolire le performance dello sciame soprattutto nella fase iniziale, quando ancora nessun robot è sincronizzato con il resto del gruppo e tutti quindi diventano sospettosi bloccando il loro lavoro per averne conferma, per far fronte a questa perdita di performance si può stabilire un tempo di "riscaldamento" nel quale i robot sono inibiti dal diventare sospettosi ma continuano a sincronizzarsi, il tempo utile al riscaldamento può dipendere in base alle variabili viste in precedenza. Questo algoritmo è stato testato, sia dal software di simulazione che da robot veri, i risultati sono concordi alle aspettative. Durante i test con i veri robot, ad uno o più robot veniva immesso un errore, si è constatato che gli altri componenti dello sciame hanno sempre rilevato l' errore, ed il tempo medio di reazione che comprende tutte le fasi sopra descritte per la rilevazione dell' errore e in più l' avvicinamento fisico e l' afferramento del robot, è di circa un minuto.

3.1.6 Considerazioni

Il modello ispirato alle lucciole si dimostra efficace per la rilevazione degli errori. Questa tecnica è distribuita, un notevole vantaggio è quindi la sua potenziale scalabilità, e benchè il tempo di sincronizzazione dipende dal numero di componenti dello sciame, il metodo continua a funzionare anche se la sincronizzazione avviene solo localmente per piccoli gruppi di robot e non globalmente su tutto lo sciame.
A mio avviso, visti i tempi di recupero di un' unità (circa un minuto),

prima di applicare questa metodologia si potrebbe effettuare un' analisi dei pericoli con il metodo della FMEA, proposto nel capitolo precedente, per valutare se riparare un robot durante lo svolgimento di un compito. A volte il recupero di un robot potrebbe comportare un inutile spreco di tempo, sarebbe più utile segnalare e intraprendere azioni solo nel caso in cui un robot impedisca il completamento del compito; per non rallentare i tempi di escuzione del compito si potrebbe non dare assistenza, ad esempio, ad una unità che ha subito un guasto dell' intero sistema che diventa quindi solo un ostacolo ma non nocivo per il completamento del compito.

3.2 Sistema immunitario artificiale

3.2.1 Tecnica del granuloma

Come abbiamo visto la robotica degli sciami per definizione prende ispirazione dalla biologia, in questo sezione vediamo come uno studio sperimentale condotto da Ismail e Timmis [6] ,[7] ipotizza, come soluzione alla tolleranza ai guasti, un procedimento presente nel sistema immunitario quello della formazione di un granuloma. Questo procedimento deriva dagli studi di immuno-ingegneria [12], un nuovo tipo di ingegneria, che può essere usata nello sviluppo di un sistema immunitario artificiale (AIS) basato sulla conoscenza della biologia. Il granuloma è una formazione di cellule immunitarie che cercano di rimuovere sostanze potenzialmente nocive per l' organismo, seguendo questa definizione semplificata vediamo come questo concetto che appartiene alla biologia possa essere mappato in modo distribuito negli sciami robotici per isolare o riparare unità che hanno subito un guasto.

Possiamo applicare questa tecnica al caso di studio trattato nel capitolo precedente, nel quale viene preso in esame uno sciame di robot (Linuxbot) il cui comportamento emergente è quello di muoversi autonomamente sulla base di informazioni locali, pur rimanendo in gruppo, per poi eseguire un compito, che nel caso precedente è quello di circondare un obbiettivo. Per mantenere un certo grado di aggregazione i Linuxbot utilizzano un sistema di comunicazione wireless creando una rete ad-hoc tra robot e robot, abbiamo anche visto, tramite la FMEA quali sono i guasti dei robot, che possono influire negativamente compromettendo

il completamento del compito da parte dell' intero sciame. Lo scopo di questa tecnica, è di implementare una sorta di sistema immunitario, dove i componenti dello sciame dovranno individuare chi tra di loro ha subito un guasto e isolarlo dal resto del gruppo.

3.2.2 Il modello della formazione granuloma

La formazione di un granuloma è un complesso meccanismo capace di incubare e distruggere agenti patogeni, pericolosi per l' organismo ospite. I componenti principali responsabili della formazione del granuloma sono:

- macrofagi
- cellule T
- citochine

I macrofagi si attivano quando un un corpo estraneo potenzialmente dannoso si introduce nell' organismo ospitante, ad esempio, batteri, microbi e virus. Una volta attivati inglobano il batterio che continuerà a riprodursi all' interno del macrofago fino a spezzarlo, ma a questo punto il macrofago infetto manderà segnali ad altri macrofagi che formeranno un muro di contenimento intorno ad esso dando origine alla formazione del granuloma che isolerà l' infezione dalle altre cellule. In figura (3.7) è rappresentato il meccanismo.

La formazione di un granuloma può essere rappresentata anche con un diagramma di stato come quello in figura (3.8). Possiamo notare come un macrofago infettato da un batterio lo inglobi, diventando così un macrofago infetto, che manderà segnali, ad altri macrofagi, da cui verrà circondato affinché non propaghi l' infezione all' intero sistema, e invierà anche segnali alle cellule T che si muoveranno verso il sito dell' infezione isolandolo dalle altre cellule fino a quando la minaccia non sarà stata debellata. Un ruolo importante viene anche svolto dalle citochine che sono responsabili della trasmissione dei segnali tra le cellule del granuloma, del reclutamento di cellule linfoidi e della corretta attivazione dei macrofagi.

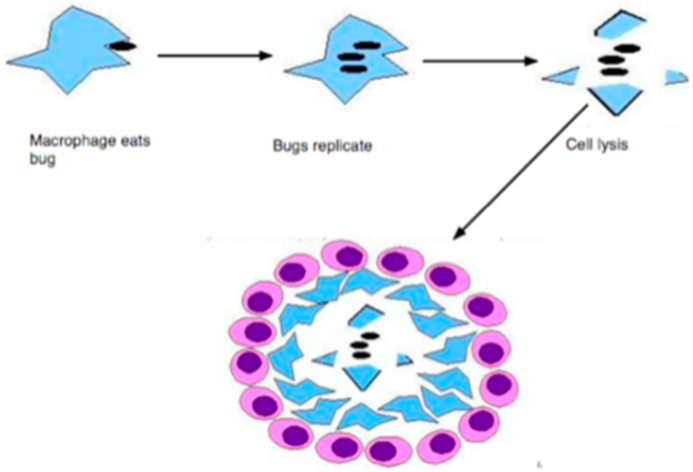

Figura 3.7: Principali passaggi nella formazione del granuloma, in evidenza la funzione dei macrofagi

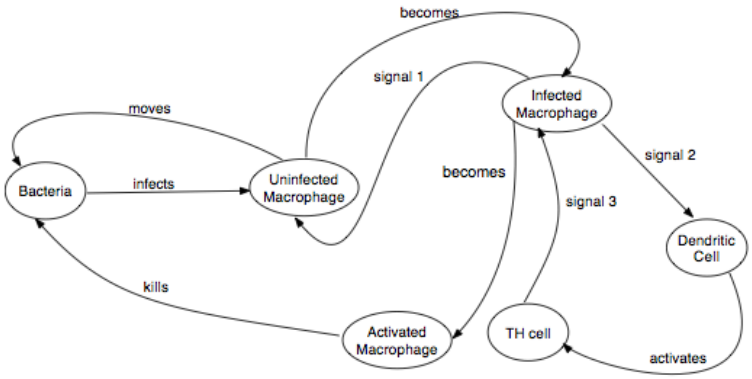

Figura 3.8: Diagramma di stato, formazione granuloma

3.2.3 Applicazione alla robotica

Seguendo il modello della formazione del granuloma, si vuol far in modo che lo sciame circondi i robot guasti, così come fanno le cellule macrofage con i batteri. Isolando i robot guasti o parzialmente guasti, si evita di

compromettere il comportamento dello sciame, rendendo così l' intero sistema più robusto. Ogni qual volta un robot subisce un guasto esso deve segnalare al gruppo il suo malfunzionamento, ad esempio possiamo qui applicare la tecnica delle lucciole vista in precedenza, che seguendo il parallelismo con la formazione del granuloma è la funzione che hanno le cellule citochine, i primi robot che si accorgeranno dell' anomalia avranno la funzione delle cellule - T ovvero quello di attrarre altre unità dello sciame. Con questo procedimento possiamo quindi allontanare dalla sciame quei robot che hanno subito dei guasti e che potrebbero compromettere lo svolgimento del compito. Tale tecnica, ad esempio, potrebbe trovare applicazione nel caso in cui un robot termini la sua scorta di energia, responsabile del corretto funzionamento, allora dei robot muniti con i giusti sistemi potrebbero circondarlo e ricaricarlo, rendendolo capace di riprendere il lavoro che stava svolgendo. Di seguito viene presentato un algoritmo per l' implementazione di questa tecnica

begin
| Immissione dei robot nell' ambiente
| **repeat**
| | Spostamento casuale nell' ambiente
| | Rilevamento di un errore
| | Diffusione segnale che rappresenta il guasto
| | Accerchiamento dell' unità danneggiata
| | Isolamento dell' unita o se possibile riparazione
| **until** per sempre
end

3.3 ALLIANCE

3.3.1 Architettura

ALLIANCE [10] è un' architettura di controllo che facilita la tolleranza ai guasti, l' affidabilità, e la cooperazione adattiva tra piccole e medie squadre di robot eterogenei, che svolgono compiti composti da attività indipendenti, mira a consapevolizzare i robot circa le capacità degli altri robot del loro gruppo. Si basa sull'utilizzo di motivazioni modellate

matematicamente all'interno di ogni robot, come l' "impazienza" e l' "acquiescenza" , che possono essere utilizzate per poi permettere ai robot di determinare se un compito è stato eseguito adeguatamente, o se un robot non è riuscito a completarlo o se ha subito un guasto. ALLIANCE è stato progettata per permettere ai robot di continuare i loro compiti, anche se alcuni robot si guastano o se la comunicazione tra i robot non è più possibile. Questa architettura usa la comunicazione wireless per far trasmettere periodicamente ai robot un messaggio dove indicano il loro stato di salute, l' attività che stanno svolgendo e il loro ID. Quando un robot comunica che sta svolgendo un' attività, altri robot lo controllano per accertarsi che la esegua correttamente, basandosi su dei modelli di aspettativa dove si ipotizzano dei tempi entro il quale un robot deve eseguire il compito, e se queste aspettative non vengono soddisfatte allora un altro robot può intervenire per portare a termine il lavoro. In ALLIANCE, la possibilità per i robot di far fronte a eventi imprevisti e robot guasti, viene definita attraverso l'uso di motivazioni. Possiamo vedere un modello dell' architettura in figura (3.9). I set di comportamento

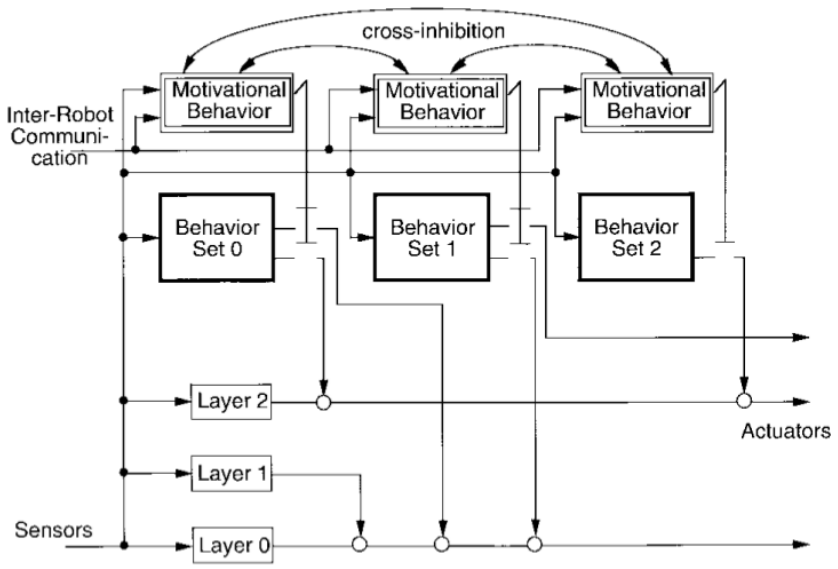

Figura 3.9: Architettura ALLIANCE

(Behavior set) ragguppano comportamenti basilari che sono necessari per l' esecuzione di un particolare compito. Le motivazioni consistono in livelli di impazienza e acquiescenza che possono innalzare o abbassare l' interesse di un robot nell' attivazione di un set di comportamento corrispondente al compito che deve essere eseguito. L' impazienza permette ad un robot di gestire situazioni dove altri robot non riescono a svolgere un determinato compito, l' acquiescenza consente ad un robot di gestire le situazioni in cui, esso stesso, non riesce a svolgere correttamente il suo compito. In ogni momento durante l' esecuzione di un compito, ogni comportamento motivazionale (motivational behavior) riceve input da una serie di fonti, tra cui i sensori, comunicazione tra i robot e le motivazioni interne. Inizialmente la motivazione per eseguire un set di comportamenti è uguale a zero, poi ad ogni periodo di tempo, il livello di motivazione è ricalcolato sulla base di:

- sul precedente livello di motivazione
- sul tasso d' impazienza
- se i sensori indicano che il comportamento è necessario
- se il robot ha già attivo un set di comportamenti
- se un altro robot ha recentemente iniziato a lavorare sul compito
- se il robot vuole rinunciare al compito basandosi sul tempo che sta impiegando per svolgerlo

la motivazione continua a salire con un certo tasso finché una delle sottoelencate situazioni non accade:

- i sensori indicano che il set di comportamento non è più necessario
- un altro set di comportamento è già attivato nel robot
- altri robot hanno appena iniziato il lavoro
- il robot ha accettato il suo compito

in qualsiasi di queste quattro situazioni la motivazione torna a zero, altrimenti la motivazione aumenta fino ad una determinata soglia, a questo punto il corrispondente set di comportamento è attivato e quindi il robot ha selezionato un lavoro da svolgere, allora trasmette un segnale periodico nel quale avvisa gli altri robot che sta svolgendo quel compito, si evita così di assegnare quel lavoro ad un altro robot (cross-inhibition). Un' estensione di questa architettura viene chiamata L-ALLIANCE, fornisce meccanismi che permettono ai robot di aggiornare dinamicamente le impostazioni dei parametri in base alle conoscenze apprese dalle esperienze precedenti e se accade qualche problema durante lo svolgimento del lavoro, i robot possono cambiare i loro compiti per compensare al problema incorso.

3.3.2 Considerazioni

ALLIANCE si è dimostrata affidabile per un sistema multi–robotico composto da poche unità, capace di organizzare in modo ottimale il lavoro del gruppo anche se si riscontrano guasti. Come affermano Dorigo e altri [3], questa architettura risulta di difficile implementazione negli sciami robotici poichè manifesta problemi di scalabilità. Pertanto l' applicazione di questa soluzione deve essere valutata tenendo conto delle dimensioni del sistema in cui deve essere implementata.

Capitolo 4

Sviluppi futuri e concluisoni

4.1 Sviluppi futuri

Attualmente esistono pochi studi relativi alla fault tolerance negli sciami robotici, ramo della robotica sviluppatosi negli ultimi decenni, per cui restano ancora molte questioni aperte da risolvere.

4.1.1 Metodi standard di misura

In futuro sarà necessario trovare metodi standard per la misura della robustezza e per la tolleranza ai guasti negli sciami robotici. Una possibile metodologia potrebbe essere quella esposta (FMEA), che dovrebbe però essere sperimentata, allargando lo studio a diversi tipi di sciami, in modo da ricavare un indice standardizzato per comprendere al meglio l' effettiva tolleranza ai guasti di uno sciame, l' utilizzo di tale indice potrebbe essere uno dei fattori per decidere che tipo di tecnica applicare per aumentare la fault tolerance.

4.1.2 Scalabilità

Bisognerà rendere le tecniche di fault tolerance scalabili, in modo da poterle implementare in sciami che presentano un alto numero di componenti, ad esempio, la tecnica delle lucciole si è dimostrata in linea teorica ottimale per quanto riguarda la scalabilità, ma a causa dei tempi di sincronizzazione in sciami molto numerosi si potrebbero riscontrare dei pro-

blemi o avere una sincronizzazione parziale a gruppi dello sciame, che non dovrebbe compromettere il suo meccanismo, ma tutti questi aspetti necessitano di ulteriori studi ed approfondimenti. Bisognerebbe effettuare studi per applicare tale tecnica, o altre, anche a sciami etereogeneri.

4.1.3 Immuno-ingegneria

Un' altra strada da seguire e esplorare è quella aperta dal nuovo campo dell' immuno-ingegneria, che indagando approfonditamente il sistema tollerante ai guasti per eccellenza, quello che viene "implementato" in natura, che, come visto nell' esempio della formazione del granuloma artificiale, sembra applicabile e affidabile anche negli sciami robotici. Sarà necessario ampliare gli studi interdisciplinari che coinvolgono sia le branche della biologia sia quelle dell' ingegneria e della robotica per trovare soluzioni innovative per l' individuazione di nuove tecniche di fault tolerance.

4.2 Conclusioni

In questa tesi sono state analizzate alcune tecniche per aumentare la fault tolerance negli sciami robotici, e si è sottolineata l' importanza di un' analisi qualitativa dei guasti, che è un indispensabile strumento per giungere ad una completa e chiara comprensione della problematica. Sono state poi analizzate varie tecniche. Per prima è stata esaminata quella delle lucciole, un approccio distribuito che si basa solo su informazioni locali ispirato al lampeggiare sincronizzato di alcune specie di lucciole, per rilevare unità non più funzionati dello sciame, e quindi intervenire per riparare, se possibile, l' unità danneggiata non più funzionante. Tale tecnica si è rivelata essere una strada percorribile per rafforzare la tolleranza ai guasti. La sua caratteristica più importante è la potenziale possibilità di essere utilizzata in sciami di grosse dimensioni, caratteristica che come si è detto nella sezione precedente deve essere ancora testata e perfezionata con sciami reali, ma si sono ottenuti risultati incoraggianti nelle simulazioni; da ciò si deduce che potrebbe essere implementata nei casi concreti con risultati soddisfacenti. In seguito si è visto come lo studio dei sistemi immunitari possa apportare interessanti spunti per creare

nuove tecniche di fault tolerance, come in quella ispirata dalla formazione del granuloma che dà la possibilità allo sciame di isolare unità guaste cosicché non possono più compromettere il comportamento dello sciame e fanno aumentare la sua robustezza. Si è esaminata inoltre l' architettura ALLIANCE e i suoi meccanismi per individuare robot guasti e si sono evidenziate le sue problematiche, ad esempio, la sua difficile applicabilità a grandi sciami, e quindi non idoneo alla scalabilità che spesso è un fattore caratterizzante degli sciami robotici. Gli studi e le tecniche qui presentate potrebbero mantenere un discreto livello di fault tolerance, ma è sicuramente necessario approfondire questa tematica, in modo tale da ottenere meccanismi sempre più efficienti, scalabili e performanti superando le problematiche attuali. La crescente diminuzione dei costi dei componenti elettronici costruiti in larga scala, stimolerà sempre più la costruzione di sciami robotici e quindi anche lo studio relativo alla fault tolarence un campo di ricerca indispensabile e aperto a nuove sfide future.

Ringraziamenti

Ringrazio in particolar modo il Prof. Andrea Roli per la disponibilità e la cordialità dimostrate durante lo svolgimento di questa tesi. Ringrazio inoltre la mia famiglia, i miei amici e tutti i compagni di corso, in particolare Spera, Stefe, Costa, Fioro, Cangio, Mazzo e Mirko, che mi hanno sostenuto nel percorso degli studi.

Bibliografia

[1] Swarm-bots. `http://www.swarm-bots.org/`.

[2] Swarmanoid. `http://www.swarmanoid.org/`.

[3] M. Dorigo A.L. Christensen, R. O'Grady. From fireflies to fault tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4):754–766, 2009.

[4] Gerardo Beni and Jing Wang. Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics?*, pages 703–712. Springer.

[5] J. D. Bjerknes and A. F. Winfield. On fault tolerance and scalability of swarm robotic systems. In *Distributed Autonomous Robotic Systems: The 10th International Symposium. (83) pp. 431 - 444.* Springer, 2013.

[6] Amelia Ritahani Ismail and Jon Timmis. Aggregation of swarms for fault tolerance in swarm robotics using an immuno-engineering approach. *UK Workshop on Computational Intelligence*, 2009.

[7] Amelia Ritahani Ismail and Jon Timmis. Towards self-healing swarm robotic systems inspired by granuloma formation. In *Engineering of Complex Computer Systems (ICECCS), 2010 15th IEEE International Conference on*, pages 313–314. IEEE, 2010.

[8] Dailey K.W. In *The FMEA Handbook*. DW Publishing, 2004.

[9] F. Mondada, G. C. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L.M. Gambardella, and M. Dorigo. Swarm-

bot: a new distributed robotic concept. *Autonomous Robots*, 17(2–3):193–221, 2004.

[10] Lynne E Parker. Alliance: An architecture for fault tolerant multirobot cooperation. *Robotics and Automation, IEEE Transactions on*, 14(2):220–240, 1998.

[11] Lynne E Parker. Reliability and fault tolerance in collective robot systems. *Handbook on Collective Robotics: Fundamentals and Challenges, page To appear. Pan Stanford Publishing*, 2012.

[12] Jon Timmis, Emma Hart, Andy Hone, Mark Neal, Adrian Robins, Susan Stepney, and Andy Tyrrell. Immuno-engineering. In *Biologically-Inspired Collaborative Computing*, pages 3–17. Springer, 2008.

[13] A.F.T. Winfield and J. Nembrini. Safety in numbers: fault-tolerance in robot swarms. In *Int. J. Modelling, Identification and Control, Vol. 1, No. 1, pp.30 – 37*, 2006.