

UNIVERSITÀ DEGLI STUDI DI UDINE

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Tesi di Laurea in Scienze dell'Informazione

METODI PER LA NAVIGAZIONE AUTONOMA
E LA SINTESI AUTOMATICA DI MAPPE DI
AMBIENTI INTERNI IN PRESENZA DI DATI
SENSORI INCERTI

Laureando:

Andrea Fusiello

Relatore:

Prof. Vito Roberto

Correlatore:

Dott. Bruno Caprile

ANNO ACCADEMICO 1992 - 1993

Prefazione

Da un punto di vista molto generale, la tesi che viene qui presentata si inserisce nel tema della rappresentazione astratta di dati sperimentali (segnatamente, sensori), mentre da quello più squisitamente applicativo, l'ambito in cui essa si è svolta è quello della navigazione autonoma di robot mobili. In particolare, il lavoro di ricerca da noi svolto, e di cui il presente elaborato costituisce una descrizione di dettaglio, è consistito nell'affrontare il problema della sistemi automatica di mappe di ambienti interni, dapprima definendone i termini, e quindi proponendo un metodo per la sua soluzione.

L'ambiente viene esplorato da un veicolo mobile autonomo, il quale – dotato di opportune capacità sensorie – sintetizza una rappresentazione che risulta adeguata alla pianificazione del suo proprio moto. Da un punto di vista astratto, il veicolo è un dunque agente, il quale interagisce con il mondo esterno mediante azioni e percezioni e ne “apprende” un modello in termini di relazioni causali – per quanto incerte – intercorrenti fra azioni eseguite e percezioni ricavate. L'ambiente viene rappresentato come un automa a stati finiti: gli stati sono individuati da percezioni sensoriali distinte, mentre gli ingressi sono le azioni motorie. Sia le azioni che le percezioni, possiedono un elevato livello di astrazione e strutturazione, in modo tale che il numero di stati distinguibili e le possibilità di modificare l'ambiente risulti adeguatamente limitato.

Il proposito di operare in un ambiente reale esige tuttavia che vengano adottate tecniche “robuste” per il trattamento dell'incertezza (di misura o di modello) che affligge le percezioni e le azioni. L'incertezza nella stima della posizione del veicolo viene trattata da un punto di vista statistico mediante modellizzazione del processo di localizzazione, conseguentemente utilizzando un formalismo per il trattamento dell'incertezza derivato dalla teoria del filtraggio di Kalman.

In particolare, l'approccio alla rappresentazione dell'ambiente che viene proposto – e che costituisce la parte più originale del lavoro – può essere considerato “ibrido”: sebbene il modello soggiacente sia intrinsecamente qualitativo, dal grafo di transizione dell'automa è possibile estrarre non solo la struttura topologica dell'ambiente, ma anche informazione metrica, relativa cioè, a distanze ed angoli.

Gli esperimenti ed i relativi risultati che vengono presentati e riportati – sia pure tutt'altro che definitivi – sembrano suggerire che l'approccio qui proposto sia sostanzialmente corretto e promettente, e ci incoraggiano quindi a proseguire nell'intento sia di indagare ulteriormente il problema nella sua generalità, quanto di perfezionare ciò che abbiamo avuto modo di realizzare.

Ringraziamenti

Questo lavoro è stato svolto sotto la supervisione di Bruno Caprile nei confronti del quale sono profondamente debitore per aver ispirato l'intero lavoro, per la sua paziente e costante opera di revisione, e per avermi incoraggiato e sostenuto. Sono altresì debitore verso l'intero gruppo di Visione dell'IRST, ed in particolare Marco Aste, Massimo Boninsegna, Roldano Cattoni e Tarcisio Coianiz per i preziosi consigli e l'aiuto fornitomi in molteplici occasioni. Un particolare ringraziamento a Cesare Furlanello per avermi aiutato nella messa a punto degli esperimenti. A tutti indistintamente sono grato per la disponibilità e la gentilezza avuta nei miei confronti.

Ringrazio altresì l'Istituto per la Ricerca Scientifica e Tecnologica di Trento per avermi ospitato e aver messo a disposizione la struttura e le attrezzature.

Infinita gratitudine va ai miei genitori Egle e Vincenzo.

Indice

Prefazione	ii
Ringraziamenti	iv
1 Introduzione	1
1.1 Scopo ed inquadramento del lavoro	2
1.2 Architetture	3
1.3 Sensori esterni/ambientali	5
1.4 Sintesi di mappe ambientali	6
1.5 Localizzazione	6
1.6 Pianificazione del moto	8
2 Lavori collegati	10
2.1 Modelli analogici	12
2.1.1 Griglie cartesiane	12
2.1.2 Modelli adattivi	14
2.2 Modelli simbolici	14
2.2.1 Modelli topologici	14
2.2.2 Mataric	17
2.2.3 Mappe di localizzazione	18
3 Una Proposta	19
3.1 Introduzione	19
3.2 Schema architetturale	20
3.2.1 Descrizione funzionale del Behaviour Level	21
3.2.2 Navigator	23

3.2.3	Learner	23
3.3	Commenti	24
4	Trattamento dell'incertezza	26
4.1	Rappresentazione di parametri incerti	26
4.2	Confronto di misure incerte	28
4.3	Propagazione di misure incerte	29
4.4	Combinazione di misure incerte	30
4.5	Applicazione alla localizzazione di un veicolo mobile	31
4.6	Odometria	32
4.6.1	Modello dell'errore nella posizione	34
5	Modellizzazione dell'ambiente	37
5.1	Rappresentazione dell'ambiente	39
5.2	Ricostruzione dell'automa soggiacente	40
5.2.1	Algoritmo di ricostruzione dell'automa	42
5.3	Applicazione	46
5.3.1	Azioni e firme	47
5.3.2	Matching	47
6	Una griglia cartesiana dai dati sonar	48
6.1	Rappresentazione analogica	48
6.2	Aggiornamento della mappa con la misura sonar	49
6.3	Geometria dell'acquisizione	52
6.4	Possibili sviluppi	54
7	Il sistema di cartografia	57
7.1	Riflessi e percezioni per l'esplorazione	57
7.1.1	Caratteristiche del Robot e dell'Ambiente di Lavoro	58
7.1.2	Riflessi	58
7.1.3	Predicati	62
7.1.4	Percezioni	63
7.2	Navigazione autonoma: il Navigator	65
7.2.1	Procedure di navigazione acquisizione della firma	65

7.2.2	Stima della posizione	67
7.3	Apprendimento ambientale: il Learner	68
7.3.1	Simulazione	70
8	Conclusioni e sviluppi futuri	71
A	Risultati teorici	74
A.1	Ipotesi	74
A.2	Algoritmo di apprendimento	74
A.3	Risultati teorici	75
B	Modello del sensore ultrasonico	79
B.1	La scheda di misura	79
B.2	Modello fisico del trasduttore	81
B.3	Interpretazione dei dati sonar	83
C	Note implementative e codice	86
	Bibliografia	110

Capitolo 1

Introduzione

È spesso difficile apprezzare le difficoltà inerenti al progetto ed alla realizzazione di un veicolo che sappia muoversi autonomamente. Le persone e gli animali non hanno problemi ad agire in un ambiente complesso, a raccogliere, interpretare e rispondere alle più disparate informazioni sensoriali. Dunque perché è così difficile costruire un veicolo autonomo? L'analogia con gli esseri viventi è fuorviante: per quanto primitivo sia un sistema biologico, questo è comunque enormemente più complesso dei veicoli autonomi che si costruiscono attualmente. In effetti anche rispondere alla semplice domanda “dove sono?” in modo veramente automatico è un compito impegnativo per un robot mobile, allo stato attuale della ricerca. Tipicamente i non addetti ai lavori nutrono nei confronti dei robot aspettative che vengono poi fatalmente disattese nella realtà. Ciò in parte avviene perché il termine *robot* viene tipicamente utilizzato in una accezione molto ampia, capace di inglobare gli androidi protagonisti dei romanzi di fantascienza, i manipolatori industriali impiegati nelle catene di montaggio, certi giocattoli che si muovono emettendo luci e suoni. Per quanto riguarderà il presente lavoro un *robot mobile* è un veicolo capace di muoversi autonomamente in un ambiente *non ingegnerizzato* compiendo una certa operazione. Anche se con ciò non pretendiamo certo di aver fornito una definizione in qualche misura soddisfacente, varrà comunque la pena che alcuni dei termini adottati siano meglio precisati:

autonomamente – cioè senza l'intervento umano. Per esempio i veicoli teleguidati non sono autonomi, almeno non nella nostra accezione;

operazione – sequenza di azioni specifiche *finalizzate*, dipendenti cioè dall'applicazione in questione. Alcuni giocattoli si muovono autonomamente nell'ambiente ma non compiono alcunché di utile o finalizzato.

non ingegnerizzato – (*non engineered*) si dirà un ambiente in cui non siano stati predisposti particolari riferimenti, o attuata alcuna modifica *ad hoc* per la navigazione del veicolo mobile. Gli AGV (*Automatic Guided Vehicles*) operanti nelle fabbriche seguono un cavo elettrico inserito nel pavimento e quindi operano in un ambiente ingegnerizzato.

Per soddisfare questi requisiti operativi, si supporrà che un veicolo mobile posseda le seguenti capacità:

- ricavare informazioni sensoriali dall'ambiente esterno (*sensing*);
- elaborare l'informazione sensoriale (*thinking*);
- agire (muoversi) nell'ambiente (*acting*).

Sebbene nel seguito abbiamo deliberatamente evitato di utilizzare termini che, riferiti ad un automa, sarebbero risultati vaghi, inaccurati o inutilmente antropomorfi, la espressività di alcuni termini presi a prestito dalla lingua inglese ci è talvolta sembrata irrinunciabile. Se dunque ricorreremo a parole come *sensing* o *thinking*, si avrà da interpretarle come termini di comodo e non quali riferimenti, per quanto tangenziali, al tema alquanto controverso della “intelligenza” di un agente artificiale.

1.1 Scopo ed inquadramento del lavoro

L'argomento della presente tesi è la sintesi automatica di mappe di ambienti interni da parte di un robot mobile. Una mappa è una collezione di informazioni (o di relazioni spaziali) riguardanti il mondo, opportunamente organizzate e strutturate. Essa deve fornire al robot una parte del supporto conoscitivo che gli permetterà di muoversi nell'ambiente, pianificando le proprie azioni, autolocalizzandosi e localizzando luoghi ed oggetti. Supporremo che il veicolo mobile operi in un ambiente reale e non ingegnerizzato.

Per fornire un'idea di quale sia lo stato dell'arte della sintesi di mappe per la navigazione autonoma, riporteremo un test che Leonard e Durrant-White [74] affermano essere “oltre la capacità di qualsiasi robot esistente al giorno d'oggi”.

Il test del venerdì

- siamo all'interno di un edificio quale quelli che ospitano università od ospedali, e poniamo che siano le 17:30 di venerdì;
- il robot, che non ha alcuna conoscenza a priori dell'ambiente viene attivato e posizionato;
- impartiamo al robot il comando: “Costruisci una mappa dello spazio accessibile muovendoti a piacere, senza danneggiarti o danneggiare l'ambiente”;
- al lunedì mattina, il robot è in grado di eseguire missioni di navigazione usando la mappa che ha sintetizzato durante il fine settimana.

Costruire un sistema capace di superare questo test è il riferimento ideale del presente lavoro.

Nell'affrontare il problema di cui si occupa questa tesi si devono tenere presenti un insieme di problematiche, più o meno collegate, che disegnano l'ambito concettuale e operativo nel quale ci muoveremo. Nel seguito di questo capitolo ci occuperemo di alcune tipiche direzioni di ricerca sulla robotica autonoma, cioè architetture, sensori, mappe, localizzazione, pianificazione del moto. Per una panoramica più ampia sui principali problemi che si devono affrontare in robotica e sulle principali direzioni di indagine che vengono percorse attualmente si veda per esempio[34, 107].

1.2 Architetture

Le componenti funzionali separate di un robot mobile – percezione, elaborazione, azione – vanno organizzate in uno schema architettonico d'insieme.

L'architettura più convenzionale prevede moduli separati per l'acquisizione di dati sensorî, l'elaborazione dell'informazione, il controllo dell'attuazione. Questo, se da un lato permette di individuare con precisione i metodi da applicare in ciascun modulo,

dall'altro presenta lo svantaggio di non permettere la costruzione di moduli in cui le tre componenti siano strettamente accoppiate. Collegata a questo tipo di architettura è la navigazione pianificata [88], in cui il robot pianifica ogni sequenza di azioni da intraprendere, e si basa fundamentalmente su informazioni globali. Per svolgere questa attività il sistema deve necessariamente mantenere una rappresentazione del mondo.

L'approccio opposto è rappresentato dalla *subsumption architecture* di Brooks [19]. L'architettura è strutturata in livelli, corrispondenti ciascuno ad una precisa abilità comportamentale. Ciascun livello assume il ruolo di quelli sottostanti ¹ ed è composto da moduli che possono essere inibiti da quelli del livello superiore. Ogni modulo è sostanzialmente un automa a stati finiti in cui sensori e attuatori sono strettamente accoppiati. L'architettura è dotata di una certa attrattiva per il fatto di non prevedere alcun controllo centralizzato, tuttavia l'interazione tra i singoli moduli crea un comportamento emergente nel quale è difficile separare i contributi di ciascun modulo. Un'idea collegata è quella della navigazione reattiva o riflessiva in cui il veicolo non pianifica il moto ma reagisce a informazioni sensoriali di carattere locale (spaziale e temporale). Il robot non possiede alcuna rappresentazione dell'ambiente.

Vi sono poi architetture ibride che trovano origine dalla constatazione i) della impraticabilità dell'approccio classico alla navigazione pianificata, a causa della complessità computazionale [26] e della inerente imprevedibilità dell'ambiente, ii) della difficoltà per i moduli riflessivi ad interfacciarsi con strutture simboliche. Tra le architetture ibride si colloca il sistema di navigazione del robot mobile di MAIA ² [103, 92, 24, 25], in cui un sottosistema che elabora informazione simbolica e mantiene una rappresentazione globale del mondo si avvale di un apparato senso-motorio puramente reattivo. L'approccio – ispirato a lavori che diversi autori hanno presentato nel recente passato [108, 89] – consiste nel partire da funzionalità molto semplici (riflessive) e quindi definirne altre via via più complesse, finché venga raggiunto un livello di astrazione consistente con l'informazione che il sistema di navigazione scambia con l'esterno. Ne risulta un sistema che è capace di compiere missioni specificate in un

¹significa che il livello superiore realizza un comportamento più complesso, che in qualche modo “contiene” quello realizzato dal livello sottostante. Dal punto di vista del controllo ciò vuole semplicemente dire che esiste un meccanismo di inibizione per cui i livelli superiori hanno la priorità su quelli inferiori.

²MAIA è l'acronimo di Modello Avanzato di Intelligenza Artificiale.

linguaggio di alto livello e di reagire in modo “locale” ad eventi asincroni.

1.3 Sensori esterni/ambientali

Ricavare informazione dall’ambiente (*sensing*) è una delle abilità fondamentali richieste ad un veicolo autonomo ed in particolare lo è poter misurare la distanza dagli oggetti (*range*), poiché questo consente di ricavare informazioni riguardanti la posizione del veicolo, gli spazi liberi e la presenza di ostacoli.

I telemetri ultrasonici o sonar³ sono i sensori più comunemente usati per questo scopo. Il principio di funzionamento si basa sulla misura del tempo di volo impiegato da un impulso sonoro riflesso per raggiungere il sensore. Poiché l’emissione sonora è concentrata in un cono di approssimativamente 30 gradi di apertura, l’eco può provenire da un punto qualsiasi all’interno del cono. Nell’utilizzo del sensore al fine di evitare gli ostacoli (*bumper*), questa ridotta risoluzione angolare si rivela adatta allo scopo ed inoltre ha il vantaggio di consentire una computazione veloce. Quando il sonar venga utilizzato per costruire mappe dell’ambiente [48] o per ottenere informazioni geometriche [66], tuttavia, si rende necessaria una elaborazione più sofisticata. Se è richiesta una maggior precisione, essa può essere ottenuta con i misuratori di distanza laser, i quali si basano sulla misura del tempo di volo di impulsi laser oppure sullo sfasamento di un segnale modulato ed hanno caratteristiche operative migliori di quelli ultrasonici. Si tratta di una tecnologia che si è resa disponibile commercialmente solo negli ultimi anni, a causa di alcune difficoltà legate all’elettronica di misura del tempo di volo (o dello sfasamento). Si veda [58] per una panoramica sulle tecniche di *range finding*.

Analogamente a quanto avviene per gli esseri umani, non è necessario che il robot emetta radiazione sonora o elettromagnetica per percepire l’ambiente; si può infatti pensare di dotarlo di sensori visivi passivi che sfruttino la radiazione luminosa naturalmente presente nell’ambiente (*passive ranging*). Si tratterà di sensori *logici*⁴

³SONAR è l’ acronimo di SOund NAVigation and Ranging, ma viene correntemente usato per indicare un dispositivo che misura la distanza mediante emissione di ultrasuoni (*Ultrasonic Range Finder*).

⁴si tratta di una astrazione del processo fisico di misura dall’informazione fornita dal sensore. Per esempio un misuratore di profondità logico prescinde dal sensore fisico e dal processo di elaborazione associato.

costituiti dalla telecamera(e) e da un algoritmi per l'estrazione dell'informazione di profondità. Tra i metodi più diffusi in robotica è il cosiddetto *depth from motion*, in cui la profondità viene dedotta – noto il moto del veicolo – mediante inseguimento di bersagli e triangolazione o calcolo del flusso ottico (si veda per esempio [99]).

1.4 Sintesi di mappe ambientali

Un veicolo autonomo può necessitare di più mappe contenenti differenti informazioni per diverse applicazioni. Per esempio, una mappa usata per la navigazione può contenere informazioni sulla posizione dei riferimenti ambientali, mentre una mappa utile per la pianificazione del moto contiene informazioni sulla posizione e la forma degli ostacoli (o degli spazi liberi). Naturalmente anche la rappresentazione dell'informazione contenuta nella mappa e le primitive utilizzate, dipendono dall'applicazione. Grosso modo gli approcci alla rappresentazione dell'ambiente sono di due tipi:

1. simbolico: la mappa è (tipicamente) un grafo che rappresenta l'ambiente astraendone alcune proprietà sue intrinseche di carattere geometrico, funzionale, topologico, qualitativo;
2. geometrico/analogico: la mappa è una riproduzione della struttura geometrica dell'ambiente (tipicamente si considera un proiezione bidimensionale). Avremo modo di approfondire gli approcci alla rappresentazione dell'ambiente nel capitolo successivo.

1.5 Localizzazione

La localizzazione, ovvero stima della posizione di un veicolo mobile è legata al più generale problema della navigazione, consistente nell'azione del condurre un veicolo da un luogo all'altro. Possiamo grossolanamente individuare tre approcci alla localizzazione, che può essere:

- assoluta;
- relativa a punti di riferimento (*reference*);

- stimata con l'integrazione della traiettoria (*dead-reckoning*);

La posizione assoluta del veicolo viene calcolata rispetto ai punti di riferimento “visibili” la cui posizione assoluta è nota a priori. Tipicamente questi riferimenti sono artificiali, per esempio sono dei fari (*beacons*), e il loro utilizzo è un metodo efficace per ottenere una precisa localizzazione. I fari più semplici sono quelli attivi, che trasmettono continuamente un segnale acustico od elettromagnetico. Il veicolo, ricevendo il segnale, è in grado di calcolare il rilevamento (*bearing*)⁵ dal faro: dunque tre fari sono sufficienti per fornire la posizione e l'orientazione. I fari attivi sono comunemente usati nella navigazione marina e aerospaziale e in questi ambiti sono stati sviluppati diversi sistemi di navigazione come il LORAN e il GPS (basati su segnali radio). I trasponditori (*transponders*) sono dispositivi che richiedono la trasmissione di un segnale da parte del veicolo. Alla ricezione di quest'ultimo emettono un impulso. Conoscendo la velocità di trasmissione del segnale e il tempo di risposta del trasponditore, il robot calcola il rilevamento e la distanza dal trasponditore. I fari passivi più comunemente usati per applicazioni su veicoli mobili sono i catadiottri. Il robot è equipaggiato con un trasmettitore attivo, per esempio un laser, il quale scansiona continuamente il piano su cui si trovano i catadiottri. Quando un sensore riceve una riflessione luminosa, l'angolo del laser fornisce il rilevamento.

È possibile utilizzare per il posizionamento assoluto il rilevamento di *geometric beacons* naturalmente occorrenti nell'ambiente, come spigoli verticali, angoli o pareti [4] [73], la cui posizione viene dedotta da una mappa dell'ambiente. Quest'ultimo approccio, senz'altro più interessante dal punto di vista scientifico, presenta alcuni problemi di non facile soluzione legati all'interpretazione dei dati sensorî, ovvero al riconoscimento delle primitive geometriche che costituiscono i riferimenti.

Nella localizzazione del secondo tipo (*reference*), la posizione relativa del veicolo viene calcolata in riferimento ad oggetti o luoghi caratteristici (*landmarks*). Questo tipo di localizzazione è più generale e complessa e richiede un modello dell'ambiente [28].

La navigazione stimata (*dead-reckoning*) si basa sulla stima della posizione assoluta mediante un sensore “interno” che consenta la ricostruzione del moto del veicolo.

⁵angolo formato tra la direzione di un oggetto e una direzione fissa.

Un importante inconveniente di questo approccio consiste nell'accumulo degli errori che inevitabilmente intervengono nella misura, con l'effetto che la inaccuratezza nella posizione cresce illimitatamente. I sensori tipicamente utilizzati per la stima del moto sono l'odometro o altri dispositivi di guida inerziale. Questi ultimi sono generalmente costituiti da tre giroscopi e tre accelerometri, montati ortogonalmente, i quali misurano l'orientazione e l'accelerazione. Integrando due volte l'accelerazione si ottiene la posizione. L'odometro, per la semplicità ed economicità, è il sensore più diffuso a bordo dei veicoli mobili. La stima della posizione si ottiene integrando il moto, dedotto dalla lettura di codificatori assiali montati sulle ruote del veicolo [109]. Il procedimento richiede una sola integrazione contro le due richieste da sensori basati su accelerometri, di conseguenza l'errore di posizione cresce come il tempo anziché come il quadrato del tempo. Le fonti di errore per il sensore odometrico sono essenzialmente due, slittamento delle ruote e variazione della loro circonferenza dovuta al carico.

Un problema che ci si trova inevitabilmente ad affrontare nell'ambito della esplorazione di ambienti sconosciuti, è il cosiddetto *Am I There Yet Problem*. Mentre si muove nell'ambiente, il robot deve essere in grado di riconoscere luoghi precedentemente visitati in modo che l'esplorazione proceda in modo consistente. L'idea è che i luoghi debbano essere in qualche modo caratterizzati da una "firma" che consenta al robot di decidere sulla identità dei due luoghi confrontandone le firme. In mancanza di riferimenti ambientali ciò può essere fatto solo utilizzando la posizione stimata, la quale però è imprecisa. Il formalismo che introdurremo nel capitolo 4 consentirà di affrontare il problema in un modo che, oltre ad essere piuttosto elegante, si è anche dimostrato sufficientemente robusto.

1.6 Pianificazione del moto

Una delle attività principali (assieme alla localizzazione) per le quali è indispensabile una mappa è la pianificazione della traiettoria del veicolo. Il problema può essere formulato nel modo seguente: dati lo stato iniziale del veicolo, lo stato finale desiderato e i vincoli sulla traiettoria, trovare un percorso dallo stato iniziale a quello finale che rispetti i vincoli assegnati. Lo stato del veicolo tipicamente è rappresentato da posizione nel piano e orientazione. Spesso si assume che la traiettoria del robot non sia

soggetta a vincoli cinematici o dinamici e che l'unico vincolo sia di evitare le collisioni; quindi il problema si riduce a determinare un percorso nello spazio libero. Sfortunatamente molti robot (sia quelli con ruote sterzanti che quelli a guida differenziale) non sono sistemi olonomi, ovvero non sono in grado di seguire traiettorie arbitrarie, e si dovrà quindi tenere presente che non tutti i percorsi nello spazio libero corrispondono a traiettorie ammissibili.

La rappresentazione adottata per la mappa influenza il modo in cui viene pianificato il percorso. In molti casi la mappa geometrica può essere vista come un grafo (o meglio un tassellamento dello spazio) in cui gli archi rappresentano relazioni geometriche di adiacenza tra regioni di spazio libero, e il percorso può essere ottenuto applicando un qualche algoritmo classico di ricerca di un cammino sul grafo [87] – una volta che adeguate funzioni di costo siano state introdotte. Se la mappa è di tipo qualitativo [69] – ovvero è un grafo in cui i nodi sono i luoghi, e gli archi sono le azioni che consentono di transitare da un luogo all'altro – un cammino sul grafo rappresenta un percorso nello spazio, espresso in forma procedurale. Un approccio alternativo alla pianificazione, in cui la mappa passa in secondo piano, è quello di usare un modello a potenziale di campo (si veda per esempio [94]). Gli ostacoli esercitano una forza repulsiva mentre l'obiettivo attrae il veicolo. La traiettoria è generata localmente scegliendo la direzione lungo la quale il gradiente del potenziale è massimo. La computazione è locale e veloce, ma il metodo presenta lo svantaggio che il veicolo, in un ambiente densamente occupato da ostacoli, può rimanere intrappolato in un minimo locale.

Nel prossimo capitolo presenteremo una panoramica sui principali lavori svolti nell'ambito della sintesi di mappe, entrando poi nei successivi all'illustrazione vera e propria della nostra proposta.

Capitolo 2

Lavori collegati

In questo capitolo presenteremo una insieme di approcci e di lavori nel campo della sintesi di mappe ambientali, divisi per categorie. L'elenco non ha la pretesa di essere esaustivo così come la tassonomia introdotta risponde largamente ad un criterio espositivo e le classi che individua non sono strettamente distinte. Fatte queste premesse, dividiamo gli approcci alla rappresentazione dell'ambiente in *analogici* e *simbolici*. Nelle rappresentazioni del primo tipo la mappa riproduce caratteristiche fisico-geometriche del mondo. I modelli analogici si possono suddividere in due gruppi: i) adattivi, in cui le primitive utilizzate sono derivate dalla natura degli oggetti che essi rappresentano (es. poligoni per rappresentare spazi liberi); ii) rigidi, in cui viene imposta una struttura all'ambiente (come una griglia cartesiana). Molto sinte-

Rif.	Autori	Istituto	Approccio
[86, 49]	Elfes, Moravec	Carnegie-Mellon	occupancy grids
[13, 14]	Borenstein, Koren	Michigan	histogram grids
[10]	Beckerman, Oblow	Oak Ridge	labeled grid
[110]	Zelinsky	Monash	grid-based map
[81]	Malkin, Addanky	IBM Watson	hybrid graph
[69, 70]	Kuipers, Byun	Austin	qualitative map
[40, 8]	Dean, Basye et al.	Brown U.	graph-based map
[83]	Mataric	M.I.T.	distributed graph
[18]	Brooks	M.I.T.	topological map
[102]	Smith et al.	GMR Michigan	stochastic map
[75, 74]	Durrant-White et al.	Oxford	geometric beacons

Figura 1: elenco dei principali lavori nell'ambito della sintesi di mappe.

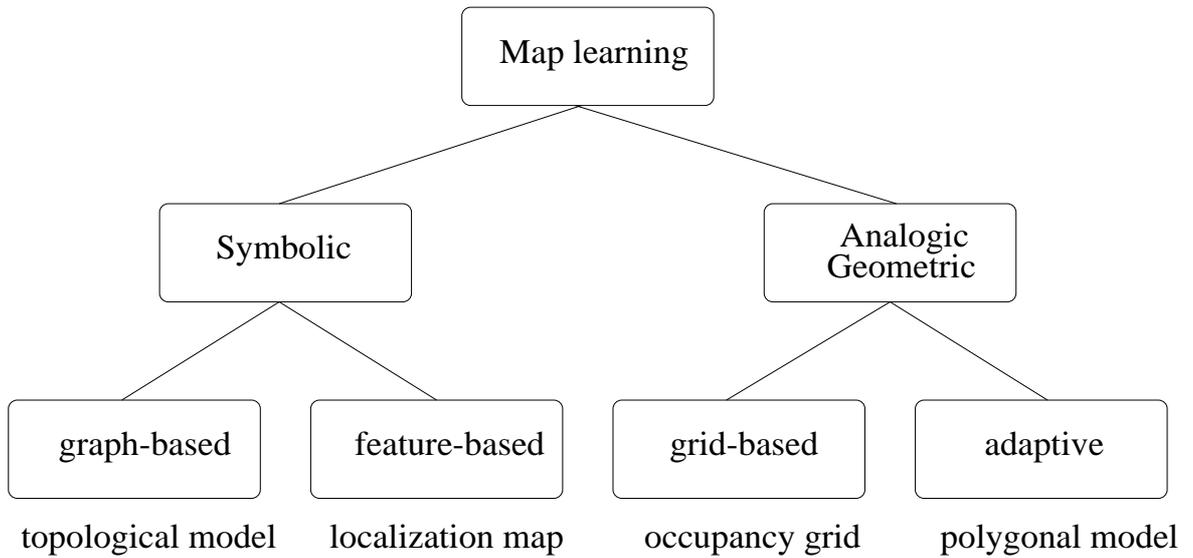


Figura 2: tassonomia degli approcci alla mappatura dell'ambiente.

ticamente possiamo dire che nel primo caso si ha un tassellamento irregolare del piano con tessere la cui forma tenta di riprodurre la forma degli spazi (liberi o occupati), mentre nel secondo caso il tassellamento è regolare e periodico. In entrambi i casi le primitive recano l'informazione della propria posizione assoluta in un sistema di riferimento globale.

La classe delle rappresentazioni simboliche è più composita: una grande quantità di modelli diversi vi ricadono per il fatto di non “riprodurre” la struttura spaziale del mondo ma di astrarne alcune caratteristiche diverse. Classifichiamo tali modelli in mappe di localizzazione (o *feature-based*) e rappresentazioni in forma di grafo (*graph-based*). La mappa di localizzazione è una collezione di posizioni assolute (tipicamente con l'incertezza associata) di riferimenti, naturali o artificiali ed è particolarmente adatta per la navigazione (e auto localizzazione), meno per la pianificazione del moto. Nella costruzione autonoma di queste mappe riveste notevole importanza la stima della posizione (*dead-reckoning*) ovvero l'informazione metrica. I modelli *graph-based* forniscono una rappresentazione qualitativa (invece che analogica) dell'ambiente. In un grafo, i cui nodi rappresentano luoghi “particolari” in base ad un qualche criterio percettivo (la presenza di un *landmark*, una certa configurazione dei dati di profondità), gli archi rappresentano relazioni spaziali o funzionali tra i luoghi. Questi modelli sono anche indicati come *topologici* in quanto la rappresentazione è invariante per trasformazioni continue; l'informazione di natura metrica è assente o comunque

non ha un ruolo importante in essa. Modelli di questo tipo si avvicinano a quello che pare essere il modo in cui gli esseri umani rappresentano il mondo. Le persone, infatti, hanno pessime prestazioni nella stima delle distanze, tali da giustificare la congettura che la parte metrica debba avere un ruolo marginale.

2.1 Modelli analogici

La costruzione di modelli analogici, per sua natura, richiede una accurata modellizzazione del sensore, dalle cui caratteristiche non si può prescindere. In tutti i lavori presentati in questo paragrafo viene impiegato il sonar e quindi la soluzione dei problemi inerenti alla interpretazione dei dati sonar costituisce un punto qualificante di ciascun approccio.

2.1.1 Griglie cartesiane

Molti autori hanno dedicato la loro attenzione alla costruzione di mappe sotto forma di reticoli che tassellano lo spazio bidimensionale (in cui ciascuna cella contiene informazione sulla sua occupazione). I primi a proporre con efficacia questo modello sono stati Elfes e Moravec [86, 49].

Elfes and Moravec

La *Occupancy Grid* rappresenta lo spazio come una matrice 2d o 3d di celle, ciascuna delle quali contiene la stima della probabilità che l'area della cella sia occupata. La mappa viene aggiornata con una regola Bayesiana, interpretando ciascuna lettura sonar in accordo ad un modello (euristico) del sensore. Il sonar misura la distanza radiale dall'oggetto più vicino all'interno del cono di emissione ma non specifica la posizione angolare del bersaglio, il quale può trovarsi su tutto un arco di circonferenza. L'idea alla base del modello è che un bersaglio prossimo all'asse acustico ha più probabilità essere osservato rispetto ad uno lontano dall'asse. In particolare, si assume una distribuzione (angolare) di probabilità Gaussiana centrata sull'asse del sensore. Le celle interne al cono di emissione sono considerate invece libere, con probabilità (radiale) Gaussiana.

I punti di forza di questo modello sono la facilità con cui si possono integrare dati inaccurati provenienti da disparati sensori, e la rappresentazione esplicita degli spazi liberi, comune a tutti i modelli analogici. D'altra parte, in questo tipo di modelli è presente più informazione di quella che veramente serve per la navigazione e questo costringe il robot a elaborare più dati del necessario. Ad esempio la localizzazione basata sul *matching* di griglie di occupazione è computazionalmente onerosa (oltre che non del tutto affidabile). Dal punto di vista concettuale è notevole che il modello non necessiti di assunzioni a priori sulla struttura spaziale del mondo.

Borenstein and Koren

Borenstein and Koren [13, 14] propongono un modello, chiamato *Histogram Grid* analogo alle *Occupancy Grids*, differendo da quest'ultimo solo nel modo in cui le celle vengono aggiornate. Per ogni lettura sonar di distanza viene incrementato il valore di confidenza nell'occupazione della cella che si trova sull'asse del sensore alla distanza misurata, come in un modello *naive* del sonar. In realtà poiché questa operazione viene effettuata con elevata frequenza mentre il veicolo si muove, si ottiene una distribuzione di probabilità come quella rappresentata dalle *Occupancy Grids*. Il metodo da noi utilizzato per la costruzione di mappe analogiche si ispira a questo approccio, come discuteremo nel capitolo 6. Il sistema presentato dagli autori impiega la *Histogram Grid* in combinazione con un algoritmo di pianificazione a potenziale di campo per navigare evitando gli ostacoli in un ambiente sconosciuto.

Beckermann and Oblow

La mappa è ancora una griglia cartesiana [10] bidimensionale ma non vi è la componente probabilistica: ogni cella è etichettata (sin dall'inizio) come *Empty*, *Occupied* o *Unknown* in base ad uno schema di interpretazione della misura sonar che tiene conto della imprecisione angolare. I conflitti che possono insorgere nell'etichettatura vengono risolti in due passi: prima con un metodo euristico che tiene conto ancora delle caratteristiche del sonar e poi imponendo una condizione di etichettatura consistente. Le mappe risultanti non sono molto dettagliate e a nostra conoscenza questo lavoro non ha avuto seguito.

Zelinsky

Anche Zelinsky [110] impiega una rappresentazione dell'ambiente come una griglia cartesiana in cui ciascuna cella è etichettata con *Empty*, *Occupied* o *Unknown*. L'enfasi dell'approccio però è sulla esplorazione attiva: il robot aggancia e insegue segmenti sonar *sonar edges* che estrae da una scansione sonar mediante un algoritmo di *line-fitting* ricorsivo e applica una versione modificata del *sonar barrier test* di Drumheller [42] per disambiguare i dati sonar. Le mappe che l'autore mostra nell'articolo sono molto dettagliate, ma la posizione del veicolo necessita di essere corretta a mano durante l'esplorazione.

2.1.2 Modelli adattivi

Molti autori, specialmente negli anni passati, hanno dedicato attenzione a questo tipo di modello. I singoli approcci differiscono molto tra di loro e risulta difficile delineare con sufficiente chiarezza ciascuno di essi, in poche righe. Ci limitiamo a riportare un elenco di parole chiave cui è associato un riferimento bibliografico. Voronoi Diagrams [23], Generalized Cones [17], Segment Model [36], Convex Polygon Model[28], Polygonal Region Model[84].

2.2 Modelli simbolici

La principale critica che viene portata ai modelli analogici-geometrici, basati sull'accumulo di descrizioni metriche dell'ambiente, è la vulnerabilità rispetto alla inaccuratezza dei dati metrici. Le griglie di occupazione includono gli errori cumulativi originati dai sensori e dagli attuatori. Un approccio più qualitativo o una esplicita rappresentazione dell'incertezza possono superare la fragilità di metodi puramente metrici.

2.2.1 Modelli topologici

La rappresentazione simbolica dell'ambiente più utilizzata in letteratura è il grafo, con alcune differenze nel significato attribuito ad archi e nodi. Il lavoro pionieristico nella rappresentazione qualitativa è quello di Kuipers [67] nel quale egli descrive il

modello TOUR, un metodo per simulare alcuni aspetti delle mappe cognitive umane. A questo sono seguiti altri lavori dello stesso autore e di altri che ne hanno tratto ispirazione. Anche il nostro approccio è ispirato ai lavori di Kuipers e Byun [69, 70].

Kuipers and Byun

La rappresentazione qualitativa dell'ambiente è basata su una gerarchia di rappresentazioni. Al livello sensomotorio vengono identificati alcuni luoghi e percorsi *distinctive*, in base all'interazione delle strategie di controllo del robot e del suo sistema senso-motorio con il mondo. Un *Local Distinctive Place* è definito dal massimo di una adeguata misura di *distinctiveness* definita nelle sue immediate vicinanze e viene raggiunto con una strategia di *hill-climbing*. Un *Distinctive Travel Path* similmente è definito da una appropriata misura e da una strategia di controllo di *path-following*. Il modello topologico viene creato collegando i luoghi (LDP) con i percorsi (DTP), ottenendo così un grafo etichettato. L'informazione metrica viene incrementalmente assimilata in descrizioni geometriche locali di luoghi e percorsi e alla fine le relazioni metriche tra questi vengono sfruttate per costruire una mappa geometrica globale. Contrariamente all'approccio classico, in cui la gerarchia di astrazioni è

sensorio → geometrico → topologico,

gli autori introducono un modello in cui le proprietà qualitative dell'ambiente vengono sfruttate prima di quelle geometriche – afflitte da errori cumulativi – secondo lo schema

[sensomotorio-controllo] → topologico → geometrico.

Al livello sensomotorio gli errori cumulativi di localizzazione sono eliminati dalle strategie di controllo locali e questo legittima l'astrazione da un mondo fisico continuo a una descrizione topologica discreta.

Come discusso nell'introduzione, un veicolo che sia in grado di esplorare l'ambiente mantenendone una rappresentazione simbolica consistente, deve poter riconoscere luoghi precedentemente visitati. La soluzione degli autori – volendo rinunciare, a questo livello, ad informazioni di natura metrica – è la *Rehearsal Procedure*, una procedura che verifica se il luogo corrente sia uno dei luoghi precedentemente visitati. In sintesi, viene pianificato sulla mappa topologica un percorso che parte dal luogo ipotizzato. Se i luoghi visitati nel percorso sono stati correttamente previsti l'identificazione è positiva, altrimenti si tratta di un luogo nuovo. Il modello qualitativo introdotto è

molto interessante, tuttavia ha un punto debole: vengono fatte assunzioni poco realistiche nella simulazione del sensore sonar, immaginato simile ad un *ray-tracer scanner* ideale (si veda appendice B). Non sono affatto chiare le prestazioni di questo modello in un ambiente reale.

Malkin and Addanki

Prendendo spunto dall'approccio qualitativo di Kuipers, Malkin e Addanki [81] propongono un modello di grafo ibrido per la rappresentazione spaziale. Un *LOGnet* è un grafo in cui i nodi sono mappe locali analogiche (*Local Occupancy Grids*) di luoghi *distinctive* e gli archi sono sequenze di azioni che il veicolo effettua per spostarsi da un luogo all'altro. Il problema del riconoscimento dei luoghi (*Am I There Yet*) viene risolto con il matching tra le mappe di occupazione locali. In questo il metodo differisce sostanzialmente da quello di Kuipers, oltre che nell'utilizzo di una mappa di occupazione locale come etichetta del luogo e nel criterio con cui vengono scelti i luoghi. La mappa viene costruita senza una precisa strategia: ogni volta che viene rilevato un ostacolo il robot costruisce una LOG. Se il luogo non combacia con alcuno dei precedentemente visitati viene aggiunto al grafo un nuovo nodo, altrimenti viene aggiunto o aggiornato un arco. Il robot non ha modo di determinare la posizione in un sistema di riferimento globale, quindi nessuna informazione metrica è disponibile. Come gli stessi autori riconoscono, “*If the robot is near any form of corner, the LOGs are more sensitive to translational offset*”. Non è chiara allora la capacità effettiva di identificazione di luoghi che si ha in questo modo.

Dean and Basye

Al lavoro di questi autori ci siamo particolarmente ispirati in questa tesi, in particolare molta influenza ha avuto l'articolo di Basye: “Graph-based mapping by mobile robot” [6] In esso l'autore introduce un algoritmo per la costruzione di un modello in forma di grafo dell'ambiente che sarà oggetto di studio nel capitolo 5. In sintesi Basye, Dean e colleghi hanno rappresentato l'ambiente come un automa a stati finiti, seguendo lo stesso Kuipers, Levitt e altri [71, 76]. Il problema dell'apprendimento di una mappa dell'ambiente viene elegantemente ricondotto all'identificazione di un automa a stati finiti, di cui gli autori hanno considerato alcune variazioni, inserendo funzioni di

transizione e/o di uscita rumorose, per meglio trattare il caso reale, in cui l'incertezza è inevitabilmente presente [40, 8]. Nel resto di questo scritto avremo modo più volte di tornare sull'argomento.

Brooks

L'approccio concettuale alla rappresentazione del mondo è ben sintetizzato da uno dei "dogma" che l'autore espone: [18] "*A representation of the world is not something from which the world needs be reconstructable. Rather a representation of the world is a statement of facts deductable from observations, and ideally includes enough fact that anything deductable from the past observation is also deductable from the representation. A representation is not an analogous structure to the world; it is a collection of facts about the world.*" Partendo da questo Brooks propone una rappresentazione degli spazi liberi come *freeways* (regioni elongate di spazio libero) e *meadows* (regioni convesse di spazio libero) collegati in un grafo che ne rappresenta le relazioni. L'uso di un sistema di riferimento cartesiano assoluto viene deliberatamente evitato, in quanto l'accumularsi degli errori nella localizzazione renderebbe globalmente inadatto il modello che si andrebbe costruendo.

2.2.2 Mataric

L'autrice [83] presenta un metodo per l'apprendimento della struttura dell'ambiente basato su di una architettura distribuita (*subsumption*). Nell'intento di aggirare la rappresentazione esplicita dello spazio, viene presentato un metodo che deduce la struttura dell'ambiente dal moto del robot, basandosi sull'euristica che il movimento avviene, per definizione, attraverso lo spazio libero. L'ambiente è descritto come un insieme di *landmarks*, strutturati in un grafo in cui gli archi indicano adiacenza topologica. Il grafo è parallelo e distribuito: i nodi sono *comportamenti* concorrenti, ciascuno di essi riceve ingresso dai sensori e da un rilevatore di *landmark* e comunica inviando messaggi ai vicini.

2.2.3 Mappe di localizzazione

Leonard e Durrant-Whyte

Gli autori propongono [75, 74] un algoritmo per la costruzione autonoma di una mappa basata su primitive geometriche. La rappresentazione è funzionale alla navigazione mediante inseguimento delle primitive, introdotta dagli stessi autori [73]. Ciascuna primitiva nella mappa è rappresentata da una stima della posizione e da due distinte misure di incertezza: una matrice di covarianza che rappresenta l'incertezza nella posizione della primitiva, e una misura di credibilità che rappresenta la confidenza nella validità della primitiva. Durante ogni ciclo di aggiornamento della posizione, viene usato un modello dei sensori per predire la posizione delle primitive correntemente inseguite. La credibilità viene rafforzata quando la posizione della primitiva viene prevista correttamente, viceversa viene diminuita in presenza di previsioni non supportate dalla misura. Osservazioni impreviste sono usate per inizializzare nuove primitive da inseguire. Il formalismo utilizzato per la propagazione dell'incertezza è il filtro di Kalman. Un particolare rilievo ha inoltre l'approccio al trattamento di dati sonar introdotto dagli autori prendendo spunto dal precedente lavoro di Kuc [65]. I problemi relativi alle riflessioni multiple e false riflessioni vengono risolti con l'introduzione delle *Region of Constant Depth* (si faccia riferimento all'appendice B).

Smith, Self, Cheeseman

Gli autori hanno introdotto una rappresentazione per l'informazione spaziale chiamata *stochastic map* [102], assieme alle procedure per costruirla, per leggere l'informazione contenuta e per aggiornarla incrementalmente man mano che vengono raccolti nuovi dati. La mappa contiene la stima delle relazioni spaziali fra "oggetti" dell'ambiente, che possiamo identificare con le primitive geometriche (*geometric beacons*) di Leonard e Durrant-White [73]. In sostanza la mappa consiste di un vettore che contiene la posizione media (rispetto ad un sistema di riferimento globale) degli oggetti nella mappa, e della matrice di covarianza associata che dà l'incertezza di ciascun punto nella mappa e l'interdipendenza di queste quantità. Il lavoro è sviluppato nell'ambito della teoria della stima dello stato (di sistemi dinamici) ed in particolare, per la procedura di aggiornamento della mappa, viene utilizzato un filtro di Kalman.

Capitolo 3

Una Proposta

3.1 Introduzione

Nel presente capitolo, che possiamo considerare quale vera e propria introduzione al corpo del lavoro di questa tesi, verrà delineata, sia pure in termini propositivi, una architettura per un sistema capace, mediante navigazione automatica, di estrarre informazione sensoriale dall'ambiente, di elaborarla e di rappresentarla in una forma che diremo *cartografica*. Il ricorso al termine “cartografico” è deliberato: intendiamo con esso sottolineare come in ogni mappa di un ambiente interno che sia di utilità generale, elementi di natura metrica convivano con elementi decisamente simbolici, come avviene per le comuni carte topografiche, le quali possono per così dire essere “lette” in modo consistente sia col regolo, sia estraendone relazioni di carattere topologico. L'approccio che adotteremo sarà dunque di tipo simbolico ma non rinuncerà completamente all'informazione di natura metrica. In questo senso il modello cartografico è ibrido.

Sembra pertanto naturale organizzare il sistema in una struttura che rispecchi quanto più efficacemente possibile questa osservazione e che sappia tuttavia incorporare – in forma adeguatamente modulare – le procedure atte sia alla movimentazione, sia alla raccolta dei dati sensori necessari.

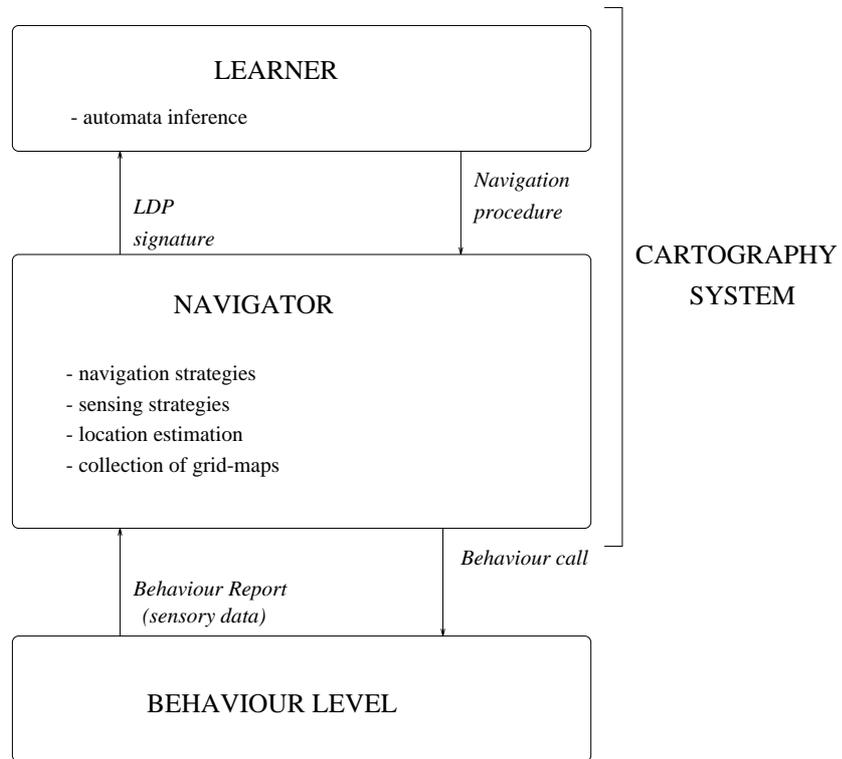


Figura 3: architettura d'insieme del sistema di cartografia.

3.2 Schema architetturale

L'architettura che proponiamo si inquadra tra quelle ibride (capitolo 1), e si articola in tre livelli (con riferimento alla figura 3): il *learner*, il *navigator* e il *behaviour level*. Quest'ultimo fornisce l'interfaccia verso le abilità senso motorie del veicolo e fa parte integrante del sistema di navigazione del robot mobile di MAIA. I livelli di cui ci occuperemo nel dettaglio e che riguardano più da vicino l'apprendimento di mappe ambientali sono il *learner* e il *navigator*, che costituiscono il sistema di cartografia descritto in questa tesi. Il primo guida la strategia di esplorazione e costruisce un modello funzionale/qualitativo dell'ambiente. Il *navigator* traduce le azioni simboliche (*procedure di navigazione*) inviate dal *learner* in una sequenza di azioni eseguibili dal *behaviour level* e mantiene una stima del punto in base a informazioni provenienti da diverse fonti. Esso, inoltre, astrae dai dati sensorî grezzi una rappresentazione simbolica da fornire al *learner* e raccoglie mappe geometriche locali dell'ambiente dai dati sonar (questa attività sarà discussa nel capitolo 6).

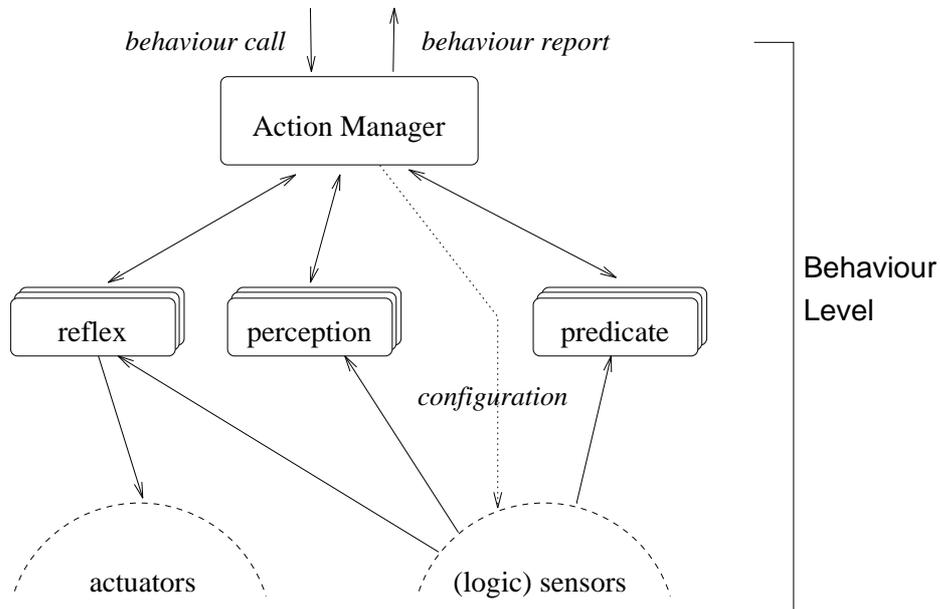


Figura 4: architettura funzionale del Behaviour Level.

Nei prossimi paragrafi illustreremo brevemente ciascun livello del sistema di cartografia, rimandando ai capitoli seguenti per una più approfondita discussione sul funzionamento di ciascuno. Iniziamo dal basso, ovvero dal *behaviour level*. Esso è stato sviluppato precedentemente a questa tesi nell'ambito del progetto MAIA [24, 25].

3.2.1 Descrizione funzionale del Behaviour Level

Il Behaviour Level è composto dalle seguenti entità principali (ci si riferisca alla figura 4):

- *i sensori*: costituiscono la sorgente di informazione per riflessi, percezioni e predicati; in essi sono compresi non soltanto i sensori “fisici” (telecamere, sonar) ma anche sensori “logici” – ovvero moduli che forniscono informazione sensoria già parzialmente elaborata (per esempio un *edge detector* oppure l’odometro che descriveremo nel capitolo 4);
- *gli attuatori*: sono gli apparati motori del robot;
- *i riflessi*: sono moduli finalizzati al controllo motorio del robot: un riflesso mette in corrispondenza un insieme di ingressi sensoriali ed un insieme di comandi di

attuazione. La trasformazione effettuata è caratterizzata dalla località spaziale e temporale. I riflessi vengono inizializzati, attivati e disattivati dall' Action Manager e comunicano con esso solo in situazioni anomale (fallimento). In ogni istante, vi è al più un riflesso attivo;

- *le percezioni*: sono moduli finalizzati alla stima di grandezze fisiche (numeriche) di carattere locale (per esempio, la distanza dalle pareti laterali o frontali, lo scostamento della direzione del robot dalla mezzeria del corridoio, la distanza percorsa). Le percezioni vengono controllate completamente dall'Action Manager: una volta inizializzata ed attivata, ogni percezione continua ad effettuare la stima della grandezza per la quale è stata progettata fino a quando non viene disattivata dall'Action Manager. Le stime di grandezze effettuate dalle percezioni vengono riportate all'Action Manager solo su sua esplicita richiesta – attraverso uno schema sincrono di *query/answer*.
- *i predicati*: sono moduli finalizzati a valutare se certe proposizioni relative all'ambiente siano vere o false (per esempio: “Il robot ha percorso una distanza di almeno 7 metri?” oppure “C'è un corridoio che si apre sulla sinistra?”). L'Action Manager inizializza e quindi attiva un predicato, il quale continua a valutare la condizione a cui è dedicato fino a quando essa non si verifichi: a quel punto il predicato invia un messaggio all'Action Manager segnalando la situazione. L'Action Manager provvede quindi a disattivare il predicato. La comunicazione risulta quindi essere di tipo asincrono.
- *l'Action Manager*: è il modulo dedicato alla gestione e supervisione dell'intero Behaviour Level. Riceve in ingresso da parte del modulo utilizzatore (*Navigation Shell* nel sistema di navigazione, *Navigator* nel sistema di cartografia descritto in questa tesi) un behaviour da mandare in esecuzione: verificata la correttezza sintattica dell'ordine ricevuto, inizializza e attiva gli opportuni moduli (un riflesso, un certo numero di percezioni e predicati) interpretando l'evoluzione globale dell'esecuzione. Provvede a disattivare i moduli al momento opportuno inviando all'utilizzatore un *report* sull'avvenuta esecuzione del *behaviour* richiesto. Eventuali dati sensoriali acquisiti con l'attivazione di una percezione all'interno del behaviour, vengono comunicati assieme al *report*. È opportuno sottolineare che generalmente l'Action Manager non interagisce direttamente

con i sensori e gli attuatori, poiché la comunicazione con queste entità è mediata dai riflessi (per il controllo degli attuatori), dalle percezioni (per la stima di grandezze fisiche) e dai predicati (per la valutazione di condizioni). L'unica eccezione a questo stato di cose è rappresentata dall'operazione di configurazione dei sensori: l'Action Manager, utilizzando comandi diretti sui sensori, può modificare la modalità con cui i dati vengono forniti (per esempio nel caso dei sonar l'operazione di configurazione consiste nello scegliere il sottoinsieme di sensori fisici da utilizzare e la sequenza di attivazione).

La coppia formata da un *riflesso* e da una combinazione logica di *predicati* – che specifica la condizione di terminazione del *riflesso* – prende il nome di *comportamento elementare* o *basic behaviour*.

3.2.2 Navigator

Al *navigator* verrà assegnata tutta la conoscenza procedurale inerente la navigazione e l'acquisizione di dati sensorî (*sensing*) nell'ambiente reale; esso sovrintende alla esecuzione delle *procedure di navigazione* attivando opportune sequenze di *comportamenti elementari* tramite l'Action Manager e gestisce i casi di fallimento in modo che la procedura appaia atomica e semanticamente consistente dal punto di vista del *learner*. Inoltre il *navigator* attua opportune strategie di acquisizione di dati sensoriali al fine di raccogliere i dati che poi interpreta astraendo le informazioni rilevanti per il *learner*. I *riflessi* e le *percezioni* di cui si è ritenuto necessario dotare il robot per l'esplorazione, in termini dei quali vengono realizzate le *procedure di navigazione*, saranno presentati nel capitolo 7. Oltre a questo compito di virtualizzare azioni e percezioni, il *navigator* manterrà una collezione di mappe geometriche locali, ottenute mediante i sonar durante l'esecuzione delle procedure di navigazione; al termine di ognuna di queste un puntatore alla mappa appena raccolta viene passato al *learner*, assieme ad una “firma” del luogo corrente.

3.2.3 Learner

Il *learner* elabora primitive simboliche e costruisce una rappresentazione dell'ambiente in forma di un grafo, o meglio un automa a stati finiti, applicando un algoritmo di

identificazione che sarà oggetto di studio nel capitolo 5. Esso invierà al *navigator* comandi di movimento (*procedure di navigazione*), del tipo “percorri il corridoio verso Nord” e riceverà una *firma* del luogo in cui il veicolo si trova al termine della procedura. Questa contiene in forma sintetica l’informazione sensoriale che identifica il luogo, e in particolare, la posizione odometrica del veicolo con la relativa incertezza e le direzioni non occluse percepite. La rappresentazione del mondo che si ha a questo livello è un grafo orientato etichettato, in cui i nodi sono luoghi rilevanti identificati dalla propria firma e gli archi indicano l’azione che permette di transitare da un luogo all’altro (in analogia al modello TOUR introdotto da Kuipers [67]). Ad ogni arco è associato un puntatore ad una mappa geometrica locale mantenuta dal *navigator*.

3.3 Commenti

La parte del lavoro di questa tesi che riteniamo più originale ricade nell’approccio cartografico alla rappresentazione dell’ambiente, ovvero nella coesistenza e integrazione di informazione geometrica/numerica (la griglia cartesiana) e informazione funzionale/simbolica (il grafo etichettato). Discuteremo più dettagliatamente nel capitolo 7 di come l’informazione geometrica (mantenuta dal *navigator*) e simbolica (elaborata da *learner*) possano venire utilmente integrate in una vera e propria carta dell’ambiente.

Sarà opportuno in conclusione portare all’attenzione alcune tematiche di interesse generale che questa tesi affronta, e che saranno oggetto di discussione nei capitoli successivi:

- trattamento, sia al livello simbolico che a quello geometrico dell’incertezza, la quale è presente inevitabilmente nell’ ambiente reale;
- costruzione di un modello funzionale dell’ambiente attraverso le capacità sensorimotorie di un agente (il quale interagisce con il mondo effettuando opportuni esperimenti);
- integrazione della rappresentazione qualitativa (globalmente consistente ma localmente inaccurata) con la rappresentazione geometrica (globalmente inconsistente ma localmente accurata) del mondo.

In questi primi capitoli abbiamo introdotto e formalizzato il problema di cui ci occupiamo, presentato le soluzioni proposte in letteratura, quindi abbiamo delineato il nostro sistema di cartografia. Nei prossimi capitoli tratteremo dettagliatamente alcuni temi centrali del nostro approccio, segnatamente: i) introdurremo un formalismo per la manipolazione dell'incertezza spaziale che ci consentirà di esplorare un ambiente sconosciuto mantenendone una rappresentazione consistente, ii) presenteremo un algoritmo per l'apprendimento di un modello qualitativo (topologico) dell'ambiente mediante esplorazione, ii) discuteremo sulle tecniche di elaborazione dei dati sonar che consentono di ottenere mappe geometriche (griglie di occupazione) dell'ambiente. Nel capitolo 7 tutti questi argomenti saranno ripresi e verrà descritto il funzionamento globale del sistema di cartografia, assieme alle funzionalità senso motorie (riflessi e percezioni) che consentono al robot di esplorare e rappresentare ¹ un ambiente sconosciuto.

¹torneremo nel seguito su questo argomento, ma notiamo sin d'ora che il modello funzionale che un agente costruisce del mondo dipende in modo cruciale dalle capacità senso motorie di cui è dotato

Capitolo 4

Trattamento dell'incertezza

In questo capitolo introdurremo un formalismo per il trattamento dell'incertezza. Tale strumento ci consentirà di affrontare il problema della localizzazione di un veicolo mobile in presenza di errori.

4.1 Rappresentazione di parametri incerti

Un parametro è rappresentato da un vettore \mathbf{x} di uno spazio di dimensione n . Nel nostro caso siamo interessati alla localizzazione di un robot mobile e quindi il parametro in questione è lo stato del veicolo, ovvero la sua posizione e orientazione in un sistema di riferimento cartesiano. Lo stato è descritto da un vettore

$$\mathbf{x} = (x, y, \theta)^\top,$$

dove x e y sono le coordinate della posizione e θ è l'orientazione, data come una rotazione attorno all'asse z . Gli errori che inevitabilmente intervengono nell'operazione di misura rendono incerta la stima del parametro. Un parametro *incerto* è rappresentato da una distribuzione di probabilità sulle sue componenti, ovvero da una densità di probabilità $\zeta(\mathbf{x})$ che assegna ad ogni valore del parametro una probabilità:

$$P[\mathbf{x} = \mathbf{a}] = \zeta(\mathbf{a})d\mathbf{x}.$$

Una conoscenza dettagliata della distribuzione non è necessaria in gran parte delle applicazioni, ed inoltre spesso non è accessibile a causa della mancanza di un modello così accurato dei sensori. Per queste ragioni, seguendo Smith et al. [102] scegliamo

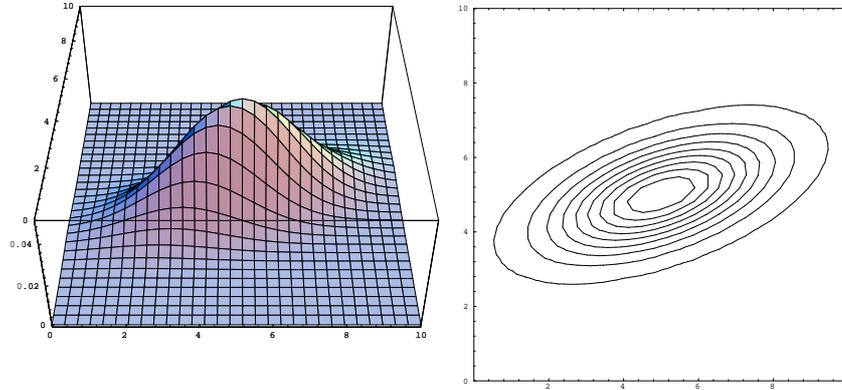


Figura 5: grafico e curve di livello di una distribuzione normale bivariata. Queste ultime sono ellissi che racchiudono regioni di confidenza per la stima della media.

di individuarla stimandone i primi due momenti – la media e la covarianza – definiti come:

$$\bar{\mathbf{x}} = E[\mathbf{x}],$$

$$\mathbf{\Lambda}_x = C[\mathbf{x}] = E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^\top]$$

La stima della media rappresenta il valore del parametro e la covarianza definisce una regione di confidenza per la stima. Nel seguito si scriverà $\bar{\mathbf{x}}$ anche per indicare una stima della media della distribuzione, ovvero si confonderà la media della distribuzione con la media campionaria che ne è una stima. La misura di un parametro viene interpretata come una stima della media.

In alcune circostanze può essere necessario determinare una particolare distribuzione che possieda i momenti stimati. Dati i momenti fino al secondo ordine, dal principio di massima entropia consegue che la distribuzione che assume meno informazione è la normale. Inoltre se la stima del parametro è ottenuta cumulando osservazioni diverse, il teorema del limite centrale garantisce che la risultante distribuzione tende ad una normale (multivariata):

$$\zeta(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Lambda}_x|}} e^{-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^\top \mathbf{\Lambda}_x^{-1} (\mathbf{x} - \bar{\mathbf{x}})}.$$

Può essere utile rappresentare la regione di incertezza di parametro tracciando un contorno che racchiuda una frazione assegnata (tipicamente prossima ad uno) della probabilità della distribuzione. Assumendo una distribuzione normale multivariata con i momenti assegnati, le curve a (densità di) probabilità costante sono iperellissoidi della famiglia

$$(\mathbf{x} - \bar{\mathbf{x}})^\top \mathbf{\Lambda}_x^{-1} (\mathbf{x} - \bar{\mathbf{x}}) = c^2.$$

Gli assi di ciascun iperellissoide sono orientati come gli autovettori di $\mathbf{\Lambda}_x$ e la lunghezza è pari a $2c\sqrt{\hat{\lambda}_i}$ (dove $\hat{\lambda}_i$ è l' i -esimo autovalore di $\mathbf{\Lambda}_x$). Si può dimostrare che [59] :

$$P[(\mathbf{x} - \bar{\mathbf{x}})^\top \mathbf{\Lambda}_x^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \leq u_\alpha] = \alpha$$

dove u_α è il quantile α -esimo di una distribuzione chi-quadrato (con n gradi di libertà, se n è la dimensione del parametro). Ad esempio, un parametro incerto in 3 dimensioni ($\bar{\mathbf{x}}, \mathbf{\Lambda}_x$) cade all'interno dell'ellissoide $c^2 = 7.81$ con probabilità 0.95. È importante sottolineare che non si assume che le distribuzioni soggiacenti siano normali. Stimiamo i momenti ed usiamo la distribuzione normale solo quando è richiesto di calcolare contorni di probabilità o, in generale, quando le circostanze impongono di assumere una particolare distribuzione.

Il formalismo introdotto per la rappresentazione dell'incertezza fornisce una elegante ambito matematico per affrontare tre problemi che ogni modello dell'incertezza si trova a fronteggiare [93], ovvero:

confronto – come valutare se due stime di un parametro sono consistenti;

combinazione – come combinare due stime di un parametro in una sola che ne costituisca un raffinamento;

propagazione – come ottenere stima di una funzione di un parametro incerto, nota una stima di quest'ultimo.

4.2 Confronto di misure incerte

Una operazione ricorrente nella percezione è il confronto o *matching*. I dati reali sono affetti da errori e quindi la prassi comune è di identificare due primitive quando la

loro differenza abbia una norma inferiore ad un valore prefissato. La rappresentazione esplicita dell'incertezza, e in particolare il formalismo introdotto nella sezione precedente, consente di usare l'incertezza come tolleranza nell'operazione di confronto. Questa consiste nel determinare se due stime si riferiscono allo stesso parametro, ovvero se due stime dello stesso parametro sono consistenti. La similarità di due stime, ovvero la confidenza nell'identificazione, è misurata da una distanza normalizzata rispetto all'incertezza delle stime. Introduciamo la seguente forma definita positiva, detta distanza di Mahalanobis:

$$\|\bar{\mathbf{x}}^m - \bar{\mathbf{x}}^o\|_M = (\bar{\mathbf{x}}^m - \bar{\mathbf{x}}^o)^\top (\mathbf{\Lambda}_x^m + \mathbf{\Lambda}_x^o)^{-1} (\bar{\mathbf{x}}^m - \bar{\mathbf{x}}^o)$$

Se $(\bar{\mathbf{x}}^m - \bar{\mathbf{x}}^o)$ ha una distribuzione normale a media nulla, la norma di Mahalanobis ha una distribuzione chi-quadrato con n gradi di libertà, come già osservato precedentemente. Per esempio, se identifichiamo due parametri incerti a 3 dimensioni quando la loro distanza di Mahalanobis è più piccola di 7.81, si ha un intervallo di confidenza con probabilità 0.95. Si osservi che l'utilizzo di questa metrica è naturale in presenza di quantità distribuite normalmente. Infatti, assegnata una distribuzione normale multivariata $(\bar{\mathbf{x}}, \mathbf{\Lambda}_x)$, un iperellissoide a probabilità costante è una palla nella metrica di Mahalanobis:

$$B(c) = \{\mathbf{x} \mid \|\mathbf{x} - \bar{\mathbf{x}}\|_M \leq c\}.$$

4.3 Propagazione di misure incerte

Quando due primitive sono legate da una trasformazione si pone il problema di propagare l'incertezza attraverso la trasformazione. Più precisamente se due parametri \mathbf{x} e \mathbf{y} sono legati da una relazione lineare $\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{b}$ ed è nota una stima $(\bar{\mathbf{x}}, \mathbf{\Lambda}_x)$ del primo, si vuole ottenere una stima del secondo. Usando le definizioni e la linearità dell'operatore di aspettazione, si verifica facilmente che:

$$\begin{aligned} \bar{\mathbf{y}} &= \mathbf{M}\bar{\mathbf{x}} + \mathbf{b}; \\ \mathbf{\Lambda}_y &= \mathbf{M}\mathbf{\Lambda}_x\mathbf{M}^\top. \end{aligned}$$

Se la relazione $\mathbf{y} = f(\mathbf{x})$ che lega \mathbf{x} e \mathbf{y} non è lineare è necessario approssimarla con il suo sviluppo in serie di Taylor attorno al punto $\bar{\mathbf{x}}$ ottenendo:

$$f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \mathbf{J}_f(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}}) + \dots,$$

dove

$$\mathbf{J}_f(\bar{\mathbf{x}}) = \left. \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}}$$

è lo Jacobiano di f valutato in \mathbf{x} . Troncando l'espansione di \mathbf{y} al termine lineare e calcolando il valore di aspettazione si ottiene la stima lineare della media di \mathbf{y} :

$$\bar{\mathbf{y}} = f(\bar{\mathbf{x}}).$$

Analogamente la stima al primo ordine della covarianza è:

$$\mathbf{\Lambda}_y = \mathbf{J}_f \mathbf{\Lambda}_x \mathbf{J}_f^\top.$$

4.4 Combinazione di misure incerte

Una componente fondamentale di un formalismo che tratti l'incertezza è la possibilità di raffinare incrementalmente le stime delle primitive man mano che nuovi dati si presentano. Più precisamente, se è nota una stima $(\bar{\mathbf{x}}^m, \mathbf{\Lambda}_x^m)$ di un parametro \mathbf{x} e una nuova stima $(\bar{\mathbf{y}}^o, \mathbf{\Lambda}_y^o)$ dello stesso parametro viene ottenuta, si vorrebbe un metodo per combinarle che sortisca un raffinamento della stima del parametro. Immaginiamo che l'osservazione di \mathbf{x} sia indiretta, ovvero che venga stimato un parametro \mathbf{y} rappresentato da $(\bar{\mathbf{y}}^o, \mathbf{\Lambda}_y^o)$ e che sussista la relazione $\mathbf{y} = \mathbf{C}\mathbf{x}$. Il raffinamento della stima di \mathbf{x} avviene mediante una tecnica derivata dal *filtro di Kalman* [105]. La stima aggiornata $(\bar{\mathbf{x}}^n, \mathbf{\Lambda}_x^n)$, date la stima corrente $(\bar{\mathbf{x}}^m, \mathbf{\Lambda}_x^m)$ e la nuova osservazione $(\bar{\mathbf{y}}^o, \mathbf{\Lambda}_y^o)$, si calcola con:

$$\begin{aligned} \mathbf{K} &= \mathbf{\Lambda}_x^m \mathbf{C}^\top [\mathbf{C} \mathbf{\Lambda}_x^m \mathbf{C}^\top + \mathbf{\Lambda}_y^o]^{-1} \\ \bar{\mathbf{x}}^n &= \bar{\mathbf{x}}^m + \mathbf{K} (\bar{\mathbf{y}}^o - \mathbf{C} \bar{\mathbf{x}}^m) \\ \mathbf{\Lambda}_x^n &= (\mathbf{I} - \mathbf{K} \mathbf{C}) \mathbf{\Lambda}_x^m. \end{aligned}$$

Ove non si faccia alcuna ipotesi di normalità sulla distribuzione dell'errore di misura, il filtro di Kalman fornisce il migliore stimatore lineare non polarizzato di $\bar{\mathbf{x}}$. Questo significa che tra tutti gli stimatori che si esprimono come combinazione lineare della sequenza (temporale) di misure $\{\bar{\mathbf{y}}\}$, otteniamo quello per cui il valore atteso dell'errore di stima è zero (non polarizzato) e la sua varianza è minima. Si può vedere che la stima corrente del parametro $\bar{\mathbf{x}}^m$ viene corretta di una quantità proporzionale

all'errore, $(\bar{\mathbf{y}}^o - \mathbf{C}\bar{\mathbf{x}}^o)$, chiamata innovazione. Il fattore di proporzionalità, \mathbf{K} , è il cosiddetto guadagno del filtro di Kalman.

Se la relazione $\mathbf{y} = f(\mathbf{x})$ che lega \mathbf{x} e \mathbf{y} non è lineare, per applicare il filtro di Kalman è necessario linearizzarla, approssimandola con il suo sviluppo in serie di Taylor troncato al termine di primo grado. In pratica la matrice \mathbf{C} viene sostituita dallo Jacobiano di f nelle formule precedenti.

Se si dispone direttamente dell'osservazione di \mathbf{x} , $(\bar{\mathbf{x}}^o, \Lambda_x^o)$, si pone $\mathbf{C}=\mathbf{I}$, e si ottiene:

$$\begin{aligned}\mathbf{K} &= \Lambda_x^m [\Lambda_x^m + \Lambda_x^o]^{-1} \\ \bar{\mathbf{x}}^n &= \bar{\mathbf{x}}^m + \mathbf{K}(\bar{\mathbf{x}}^o - \bar{\mathbf{x}}^m) \\ \Lambda_x^n &= (\mathbf{I} - \mathbf{K})\Lambda_x^m.\end{aligned}$$

Questo è il caso che si presenta trattando della localizzazione di un veicolo mobile: il parametro ovvero lo stato del veicolo è ottenibile direttamente dalla lettura di un sensore.

4.5 Applicazione alla localizzazione di un veicolo mobile

La capacità di autolocalizzazione è una delle caratteristiche desiderabili di ogni veicolo mobile autonomo. In particolare, in assenza di riferimenti ambientali noti, risulta importante conoscere la posizione del veicolo in coordinate cartesiane. Mentre il robot si muove, i dati \mathbf{u} provenienti da codificatori ottici solidali con le ruote vengono elaborati (attraverso l'applicazione A che contiene l'informazione sulla geometria e la cinematica del veicolo) per determinare la velocità del robot. Il risultato è integrato per calcolare lo spostamento:

$$\mathbf{p} = \int_{t_0}^t A\left(\frac{\partial \mathbf{u}}{\partial t}\right) dt$$

Il sensore logico che legge gli spostamenti delle ruote forniti dai codificatori e aggiorna la posizione del robot è l'*odometro*. Un problema sorge dal fatto che c'è sempre un errore associato con il movimento di un veicolo, dunque la sua posizione è un parametro incerto. La rappresentazione esplicita dell'incertezza nella posizione si rivela molto utile, specialmente per confrontare informazioni sensoriali acquisite in

posizioni diverse. Nel sistema che abbiamo progettato, gli strumenti introdotti nelle precedenti sezioni vengono usati per per la localizzazione del veicolo mobile in un ambiente sconosciuto e privo di riferimenti artificiali. In particolare risultano utili per i seguenti compiti:

- riconoscere in base alle coordinate spaziali luoghi precedentemente visitati, ovvero risolvere quello che in letteratura viene chiamato *Am I There Yet Problem*. L'identificazione di luoghi viene effettuata in base alla distanza di Mahalanobis tra le rispettive posizioni (intese come parametri incerti). In virtù della rappresentazione esplicita delle regioni di incertezza, questo formalismo richiama il concetto di *uncertainty manifold* proposto da Brooks [18].
- aggiornare la stima della posizione in seguito alla identificazione di un luogo già visitato. Quando due posizioni stimate sono riconosciute come riferite allo stesso luogo fisico, le due stime vengono combinate come visto nella Sezione 4.4. Si ottiene una nuova stima della posizione del luogo in questione, nonché della posizione corrente del veicolo.

Per utilizzare questi strumenti è necessario possedere una stima della posizione del veicolo assieme alla incertezza associata. Un sensore virtuale che produce tali dati sarà presentato nella prossima sezione. Tecniche per la rappresentazione e la manipolazione di informazione geometrica incerta trovano applicazione oltre che nella localizzazione spaziale di un veicolo mobile [28] [18] [45] [102] [74], anche nella analisi di scene 3-D in termini di primitive parametriche [4] [38] [93] [37].

4.6 Odometria

In questa sezione discuteremo dettagliatamente un modello per l'odometria di un veicolo mobile e, seguendo Leonard e Durrant-Whyte [74], introdurremo un modello discreto per il calcolo della matrice di covarianza della posizione. Il veicolo mobile è dotato di due ruote motrici indipendenti, montate agli estremi di un asse di lunghezza L , come si vede in figura 4.6 La sterzata viene effettuata dalla differenza di velocità delle due ruote (come nei veicoli cingolati). Questa configurazione, chiamata guida differenziale (*differential drive*) assicura una maggiore manovrabilità rispetto a quella con ruote sterzanti, dato che il robot può ruotare sul posto. Come già accennato, in

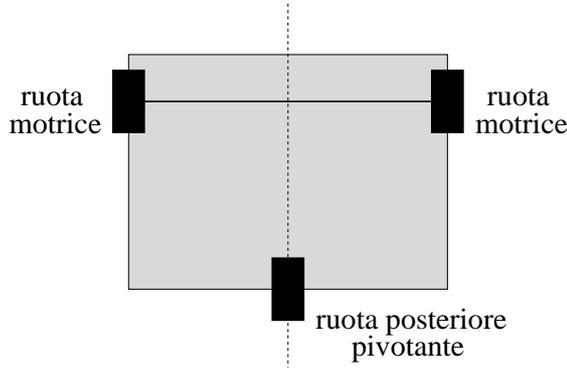


Figura 6: geometria del sistema di guida del veicolo. La sterzata viene effettuata differenziando la velocità delle due ruote motrici (*differential drive*)

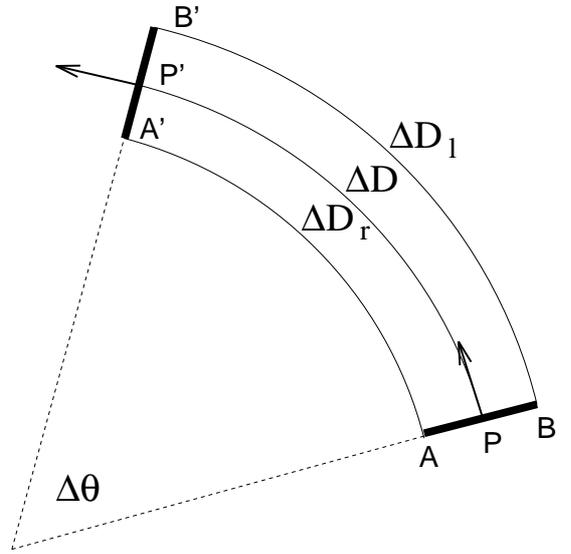


Figura 7: illustrazione del calcolo dell'aggiornamento della posizione. Il veicolo si sposta da P a P' percorrendo la distanza ΔD e variando la sua orientazione di $\Delta\theta$

due dimensioni lo stato del veicolo mobile al tempo (discreto) k è rappresentata dal vettore

$$\mathbf{x}(k) = (x(k), y(k), \theta(k))^T,$$

dove $(x(k), y(k))$ sono le coordinate del punto medio dell'asse delle ruote, in un sistema di riferimento cartesiano globale, e θ è l'angolo formato dall'asse delle x con la retta perpendicolare all'asse delle ruote (direzione di movimento o orientazione). Nella figura 4.6 le posizioni delle due ruote destra e sinistra sono indicate rispettivamente da A e B mentre P , il punto medio dell'asse, è il punto che rappresenta la posizione del robot. In questo esempio il veicolo si muove da P a P' . La distanza percorsa ΔD e la variazione nell'angolo di orientazione $\Delta\theta$, si possono calcolare in base alla misurazione della distanza percorsa dalla ruota destra e da quella sinistra, indicati rispettivamente con $\Delta D_r(k)$ $\Delta D_l(k)$:

$$\Delta D(k) = \frac{\Delta D_r(k) + \Delta D_l(k)}{2}; \quad \Delta\theta(k) = \frac{\Delta D_r(k) - \Delta D_l(k)}{L}.$$

Si vuole determinare lo stato al tempo $k + 1$ conoscendo l'incremento nella distanza percorsa da ciascuna ruota. Semplici considerazioni geometriche ci permettono di

calcolare $\mathbf{x}(k+1)$ nel modo seguente: ¹

$$\begin{aligned}x(k+1) &= x(k) + \frac{\sin(\Delta\theta(k)/2)}{\Delta\theta(k)/2} \Delta D(k) \cos(\theta(k) + \Delta\theta(k)/2) \\y(k+1) &= y(k) + \frac{\sin(\Delta\theta(k)/2)}{\Delta\theta(k)/2} \Delta D(k) \sin(\theta(k) + \Delta\theta(k)/2) \\\theta(k+1) &= \theta(k) + \Delta\theta(k) .\end{aligned}$$

È definita in questo modo la dinamica di un sistema non lineare tempo discreto :

$$\mathbf{x}(k+1) = F(\mathbf{x}(k), \mathbf{u}(k)),$$

in cui l'ingresso è:

$$\mathbf{u}(k) = (\Delta D_r(k), \Delta D_l(k))^\top .$$

4.6.1 Modello dell'errore nella posizione

A causa di una varietà di errori tra cui individuiamo lo slittamento delle ruote, la loro deformazione dovuta al carico e l'imprecisione dell'attuazione, la misura dello spazio percorso fornita dai codificatori è imprecisa e ciò si propaga rendendo imprecisa la posizione del robot calcolata dall'odometro. L'odometro, quindi, fornisce una stima $\hat{\mathbf{x}}$ (affetta da errore) dello stato del robot, e l'incertezza può essere rappresentata mediante la matrice di covarianza $C[\hat{\mathbf{x}}]$

Si assume quindi che l'ingresso sia perturbato da un rumore addittivo $\mathbf{v}(k)$ a media nulla, con matrice di covarianza nota:

$$\mathbf{u}^*(k) = \mathbf{u}(k) + \mathbf{v}(k)$$

$$E[\mathbf{u}^*(k)] = \mathbf{u}(k)$$

$$C[\mathbf{u}^*(k)] = C[\mathbf{v}(k)].$$

Indicando con $\hat{\mathbf{x}}$ la stima dello stato, si ottiene:

$$\hat{\mathbf{x}}(k+1) = F(\hat{\mathbf{x}}(k), \mathbf{u}^*(k)).$$

¹In una approssimazione del primo ordine, il termine $\frac{\sin(\Delta\theta(k)/2)}{\Delta\theta(k)/2}$ non compare. L'approssimazione corrisponde a considerare rettilineo il percorso PP' .

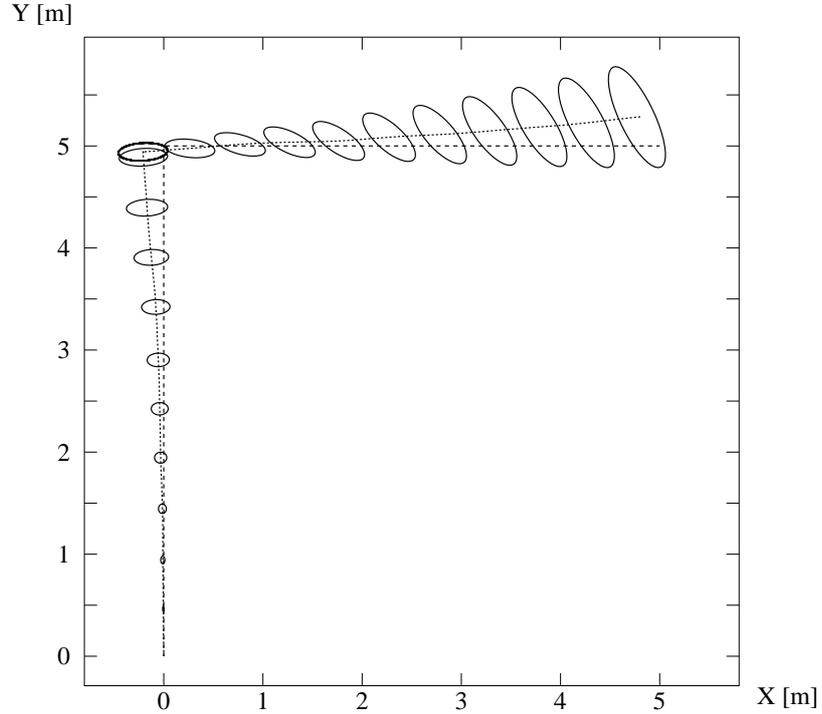


Figura 8: simulazione dell'errore nella posizione. La curva a tratteggio meno fitto è la traiettoria ideale (che coincide con quella reale se non vi sono errori nell'attuazione) ricavata dai comandi di movimento, l'altra curva tratteggiata rappresenta una traiettoria odometrica, con errore relativo simulato dell'1% sulle letture dei codificatori. Le ellissi rappresentano la regione di incertezza associata alla stima della posizione (con una confidenza del 95%).

Per l'aggiornamento della matrice di covarianza è necessario linearizzare il sistema, ottenendo la formula (valida nell'ipotesi che il rumore e lo stato siano scorrelati):

$$C[\hat{\mathbf{x}}(k+1)] = \mathbf{A}(k)C[\hat{\mathbf{x}}(k)]\mathbf{A}(k)^\top + \mathbf{B}(k)C[\mathbf{u}^*(k)]\mathbf{B}(k)^\top$$

dove

$$\mathbf{A}(k) \frac{\partial F}{\partial \mathbf{x}} \Big|_{\hat{\mathbf{x}}(k)}; \quad \mathbf{B}(k) \frac{\partial F}{\partial \mathbf{u}} \Big|_{\mathbf{u}(k)}.$$

Assumendo che:

- gli errori *relativi* sulle misure dei due codificatori siano indipendenti ed identicamente distribuiti, con media 0 e varianza σ^2 (ricavabile sperimentalmente);
- la distribuzione dell'errore assoluto $\Delta D_r(k)$ abbia una varianza proporzionale

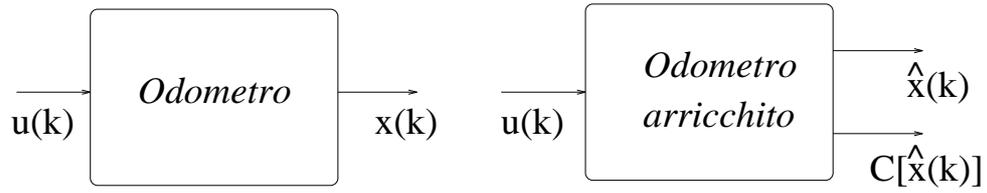


Figura 9: odometria. L'odometro arricchito fornisce anche la covarianza associata alla stima della posizione, permettendo il trattamento esplicito dell'incertezza nella localizzazione.

alla misura stessa, ovvero $\sigma_r^2(k) = \sigma^2 \Delta D_r(k)$ (e analogamente per la ruota sinistra),

si ottiene la matrice di covarianza del rumore di ingresso:

$$C[\mathbf{v}(k)] = \begin{pmatrix} \sigma_r^2(k) & 0 \\ 0 & \sigma_l^2(k) \end{pmatrix}.$$

Il sistema presentato in questa sezione può essere efficacemente interpretato come un sensore logico (che chiameremo *odometro arricchito*) il quale fornisce una stima dello stato del veicolo assieme ad una misura dell'incertezza associata a tale stima (figura 9).

Capitolo 5

Modellizzazione dell'ambiente

In questo capitolo la nostra attenzione si focalizzerà sull'apprendimento della struttura di un ambiente mediante esplorazione. Il problema si inquadra nella più generale tematica dell'*apprendimento dall'ambiente*. Diremo che un agente ha appreso dall'ambiente, quando è in grado di modificare con le sue azioni lo stato dell'ambiente in modo da raggiungere un determinato obiettivo [100]. Introduciamo più precisamente alcuni termini (si faccia riferimento alla figura 10):

l'agente è un sistema autonomo che può compiere un insieme di azioni sull'ambiente e ricevere percezioni dall'ambiente;

l'ambiente è un sistema caratterizzato da uno stato che viene modificato in conseguenza delle azioni dell'agente. Le regole o funzioni che mappano le azioni nello stato non sono note a priori;

le percezioni sono le informazioni che l'agente ottiene sullo stato dell'ambiente. Non necessariamente lo stato è percepito completamente dall'agente;

le azioni sono cambiamenti che occorrono all'interno dell'agente e possono produrre cambiamenti nello stato dell'ambiente. Le azioni sono entità separate dalle conseguenze sull'ambiente. In qualche modo l'azione "esiste" all'interno dell'automa, mentre le conseguenze dipendono dall'ambiente.

Lo schema qui delineato possiede almeno tre interessanti istanziazioni:

- apprendimento di schemi senso-motori; le azioni e le percezioni appartengono al cosiddetto livello senso-motorio, si tratta di comandi di attuazione elementari e



Figura 10: definizione di apprendimento dall'ambiente. L'agente interagisce con l'ambiente attraverso azioni e percezioni e costruisce un modello del legame tra di esse.

di dati sensori grezzi. Verosimilmente è possibile apprendere abilità di carattere “locale” e quindi “reattive”;

- apprendimento della struttura dell'ambiente; le azioni e le percezioni hanno un elevato livello di astrazione e strutturazione in modo da restringere i possibili stati distinguibili dell'ambiente e le possibili azioni che possono modificarlo. La drastica restrizione della capacità di interazione con l'ambiente consente di costruirne una mappa funzionale [6];
- apprendimento di regole di predizione; le azioni e lo stato dell'ambiente sono descritti simbolicamente mediante il “linguaggio mentale” dell'agente; fissato un insieme di obiettivi (stati ambientali) l'apprendimento consiste nel raffinare un insieme di regole del tipo $\langle \text{Condizione Azione Predizione} \rangle$ che consentano all'agente di ottenere “deliberatamente” gli obiettivi [100].

Dal punto di vista della teoria dei sistemi, l'apprendimento dall'ambiente è equivalente al problema dell'identificazione di un sistema dinamico (in generale non lineare) [77]. In questo ambito concettuale, dunque, il problema della sintesi automatica di mappe per la navigazione autonoma può essere adeguatamente modellizzato come l'apprendimento della interazione del veicolo con l'ambiente. Fornito di adeguate capacità percettive, il robot sarà in grado di ripartire l'ambiente circostante in un insieme relativamente piccolo di classi di equivalenza, le quali tassellano lo spazio in regioni omogenee rispetto ad un qualche criterio percettivo. Lo stato dell'ambiente è definito

dalla presenza del veicolo in una certa regione, in questo contesto l'apprendimento della mappa consisterà nel saper prevedere lo stato conseguente ad ogni assegnata sequenza di azioni ammissibili. Tali azioni sono procedure atomiche che consentono al veicolo di navigare attraverso regioni percettivamente omogenee.

Più nel dettaglio esaminiamo il caso di un veicolo mobile dotato di sensori ultrasonici che si muove in ambiente strutturato in corridoi di diverse dimensioni che confluiscono ad angolo retto (come nel caso dell'edificio dell'IRST, in cui abbiamo condotto gli esperimenti). Un corridoio è definito come uno spazio limitato su due lati da due superfici parallele ininterrotte e sugli altri due lati da una giunzione - indicata da un repentino cambiamento in una delle superfici parallele - o da un'altra superficie (corridoio cieco). In un tale ambiente i corridoi possono essere distinti dagli incroci dove due o più corridoi si incontrano, anche usando semplici sensori ultrasonici (o ad infrarossi). Ignorando l'orientazione, tutte le giunzioni dove due (o più) corridoi si incrociano ad angolo retto sono percettualmente equivalenti tra di loro, così come sono equivalenti tra di loro i corridoi, ove non venga considerata la lunghezza.

5.1 Rappresentazione dell'ambiente

Seguendo Kuipers [69], adotteremo una descrizione dell'ambiente di tipo qualitativo, in cui si mettono in evidenza relazioni di adiacenza (o topologiche). Il modello consiste di un grafo i cui nodi rappresentano luoghi localmente distinguibili mediante le abilità percettive di cui il robot è dotato (*Local Distinctive Places*) e gli archi sono percorsi elementari (*Conduits*) che collegano due LDP. Ciascuno di questi è etichettato con la procedura di navigazione che consente di attraversarlo, e ciò dona una connotazione funzionale alla rappresentazione. Ad ogni LDP è associata una etichetta che chiameremo firma (*signature*) che è l'insieme delle informazioni sensoriali disponibili sul luogo, e che possibilmente lo caratterizza. Nel nostro caso identifichiamo i *Conduits* con i corridoi e gli LDP con gli incroci. Volendo affrontare il problema dell'apprendimento ambientale, per come lo abbiamo introdotto in questo capitolo, si passa naturalmente ad un modello di automa a stati finiti [41]. Gli stati dell'ambiente corrispondono all'occupazione di un LDP, gli ingressi alle azioni del robot e le uscite alle informazioni sensoriali accessibili in un dato LDP. Formalmente un automa che rappresenta l'ambiente è una sestupla $M = (Q, A, Y, \zeta, \gamma, q_0)$ tale che:

- Q è un insieme non vuoto di n stati (LDP);
- A è un insieme finito di p ingressi (Procedure di navigazione);
- S è un insieme finito di m uscite (firme);
- $\zeta : Q \times A \rightarrow Q$ è la funzione di transizione; se (LDP1,NP,LDP2) appartiene al grafico della funzione, il LDP2 viene raggiunto eseguendo la procedura NP a partire dal LDP1;
- $\gamma : Q \rightarrow S$ è la funzione di uscita e mappa un LDP nella sua firma;
- q_0 è lo stato iniziale (in questo contesto non ha un particolare significato).

Le funzioni ζ e γ di un ASF possono essere convenientemente rappresentate in forma di grafi o di matrici. La funzione di transizione è assegnata da un insieme di p matrici $n \times n$, una per ogni stato di ingresso, ciascuna definita come segue:

$$\begin{aligned} \mathcal{Z}(\alpha)[i, j] &= 1 \text{ se } \zeta(q_i, \alpha_k) = q_j \\ &= 0 \text{ altrimenti .} \end{aligned}$$

A ciascuna matrice è associato naturalmente un grafo. Similmente la funzione di uscita viene rappresentata mediante una matrice $n \times m$, definita da:

$$\begin{aligned} \mathcal{G}[i, j] &= 1 \text{ se } \gamma(q_i) = \lambda_j \\ &= 0 \text{ altrimenti .} \end{aligned}$$

Ciascuno degli elementi di queste matrici è 1 o 0 e la somma degli elementi di ciascuna colonna è 1. Se la funzione di uscita è biettiva, la matrice \mathcal{G} è una permutazione.

5.2 Ricostruzione dell'automa soggiacente

In questa sezione affronteremo il compito di inferire la struttura dell'automa soggiacente l'ambiente effettuando una sequenza di azioni ed osservando l'uscita ad ogni passo (identificazione). In generale, non è possibile inferire la struttura esatta dell'automa se la funzione di uscita non è iniettiva, ovvero se è tale che

$$\exists q_1, q_2 \in Q : \gamma(q_1) = \gamma(q_2) \wedge q_1 \neq q_2.$$

Infatti possono esistere stati indistinguibili, ovvero tali che nessuna sequenza di osservazioni possa discriminarli. Per esempio se la sola informazione disponibile in un nodo è il suo grado, si vede facilmente che nessun algoritmo può ricostruire il grafo in base ad informazioni locali. Tradotto in pratica ciò significa che un robot dotato di sensori che gli consentano solo di stabilire la forma di una giunzione non può ricostruire la mappa di un ambiente realisticamente complesso. Se viceversa la funzione di uscita è iniettiva, ogni stato possiede una firma unica e l'apprendimento è facile. Tuttavia si dimostra che, anche nel caso in cui la funzione di uscita è ambigua, la struttura dell'automa può essere ugualmente ricostruita se si presenta uno dei due seguenti casi:

- l'automa in questione ha una *distinguishing sequence* ovvero una sequenza di azioni d tale che $\forall q_i, q_j \in Q : q_i \langle d \rangle = q_j \langle d \rangle \implies q_i = q_j$ ($q \langle d \rangle$ è la sequenza di uscite che si osservano eseguendo la sequenza di azioni d a partire dallo stato q) [39];
- il robot possiede un *marker* che può depositare e riprendere [44].

Nel nostro caso l'uso dell'odometro consente di disporre di una firma virtualmente unica per ogni LDP.

In situazioni reali, tuttavia, le cose sono complicate sia dalla imprecisione e dalla difficoltà di interpretazione dei dati sensoriali, sia dalla inaccuratezza della localizzazione. In particolare assumeremo che le transizioni e le osservazioni siano regolate da processi stocastici. L'incertezza nelle transizioni nasce dalla possibilità che il veicolo possa in realtà compiere una azione diversa da quella che avrebbe “inteso” compiere. Per esempio se il veicolo è dotato di una bussola imprecisa e gli viene impartito il comando “*muoviti verso Nord*”, non è detto che un osservatore lo veda andare realmente in quella direzione. Per tenere conto dell'incertezza nella navigazione, introduciamo una funzione di transizione stocastica η la quale, dato lo stato attuale e l'azione (ingresso), specifica le *probabilità* di raggiungere gli altri stati. Dunque η può essere specificata tramite le matrici di probabilità condizionale $\mathcal{N}(\alpha_1) \dots \mathcal{N}(\alpha_p)$, dove ciascuna $\mathcal{N}(\alpha)$ è una matrice stocastica ¹ $n \times n$ tale che

$$\mathcal{N}(\alpha)[i, j] = P[q_j | q_i, \alpha]$$

¹Una matrice si dice stocastica se la somma degli elementi di ciascuna riga è 1.

(la probabilità che l'ingresso α nello stato q_i sortisca la transizione nello stato q_j). Un'altra fonte di incertezza riguarda i sensori e in particolare il risultato dell'estrazione della firma di un LDP, ovvero il suo riconoscimento. Si assume che vi sia una distribuzione di probabilità che governa ciò che il robot osserva in un determinato stato. Analogamente a quanto fatto per la funzione di transizione, si introduce una funzione di uscita stocastica φ , assegnata mediante una matrice stocastica \mathcal{F} definita da

$$\mathcal{F}[i, k] = P[\lambda_k | q_i]$$

(la probabilità di osservare la firma λ_k nello stato q_i). È assegnato in questo modo un *automa stocastico* che è una “approssimazione” di quello deterministico soggiacente. Per evitare situazioni poco significative, si assume che le transizioni e le osservazioni siano corrette il più delle volte; più precisamente:

$$\begin{aligned} \mathcal{Z}(\alpha)[i, j] = 1 &\implies \mathcal{N}(\alpha)[i, j] > 0.5 \\ \mathcal{G}[i, j] = 1 &\implies \mathcal{F}[i, j] > 0.5. \end{aligned}$$

L'idea è che l'automa deterministico viene corrotto dal “rumore”, ma rimane comunque riconoscibile. Inferire la struttura di un automa stocastico effettuando degli esperimenti non è un compito facile. Nella prossima sezione descriveremo un algoritmo dovuto a Basye [6] che risolve il problema.

5.2.1 Algoritmo di ricostruzione dell'automa

L'algoritmo presentato in questa sezione effettua la ricostruzione di un automa in cui la funzione di uscita γ è univoca, in presenza di transizioni ed osservazioni stocastiche. L'algoritmo necessita di alcune ipotesi sulle proprietà strutturali dell'ambiente di cui il robot tenta di costruire una mappa e sulle proprietà delle interazioni tra il veicolo e l'ambiente. Più nel dettaglio si fanno le seguenti assunzioni:

- il veicolo è capace di compiere un insieme di azioni atomiche che restringono la sua interazione con il mondo, in un modo tale che sia ragionevole la costruzione di un modello ad automa. Ovvero esiste un automa che modella l'ambiente ed è quello che l'algoritmo si propone di ricostruire;
- l'ambiente è “reversibile”, ovvero per ogni azione esiste l'azione inversa tale che l'applicazione in sequenza delle due lascia il robot nella posizione di partenza;

- il robot è capace di acquisire la firma di ciascun LDP che raggiunge;
- il robot è in grado di transitare da un LDP ad un altro mediante una procedura di navigazione (azione) atomica;
- il robot è capace di percepire quali azioni, se eseguite, lo lascerebbero nella medesima posizione.

L'algoritmo usa una semplice strategia per esplorare il grafo di transizione e memorizzare per ogni coppia di firme (λ_i, λ_j) e per ogni azione α_k , il numero di volte che l'osservazione di λ_i seguita dall'esecuzione di α_k è risultata nell'osservazione di λ_j . Dopo aver raccolto un numero sufficiente di osservazioni, da questi dati viene estratto un automa. In particolare, se λ_j è la firma più frequentemente osservata dopo l'esecuzione di α_k partendo da λ_i , allora il grafo risultante conterrà la transizione da un certo stato q_i ad uno stato q_j in corrispondenza dell'azione α_k , in altri termini si giunge a concludere che l' LDP q_j la cui firma è λ_j è raggiungibile a partire dall' LDP q_i la cui firma è λ_i compiendo l'azione α_k . Più formalmente, il grafo di transizione dell'automa si può apprendere nel modo seguente:

Algoritmo 5.1

1. *inizializza la lista delle firme L con la firma λ_1 della locazione di partenza;*
2. *per ogni azione α_k costruisci una tabella bidimensionale T_{α_k} indicizzata in ciascuna dimensione da L ;*
3. *per un certo numero di passi,*
 - (a) *per ogni azione α_k che viene percepita lasciare il veicolo nello stato corrente di firma λ_i , incrementa di uno il valore di $T_{\alpha_k}(\lambda_i, \lambda_i)$;*
 - (b) *tra le azioni non considerate al punto precedente, scegli*
 $\alpha_k = \arg \min_{\alpha_l} \sum_j T_{\alpha_l}(\lambda_i, \lambda_j)$ *(è l'azione eseguita il minor numero di volte nello stato corrente);*
 - (c) *esegui l'azione α_k e ottieni la firma λ_j dello stato risultante;*
 - (d) *se la firma non è in L , aggiungila ed estendi ciascuna tabella con una nuova riga e una nuova colonna;*

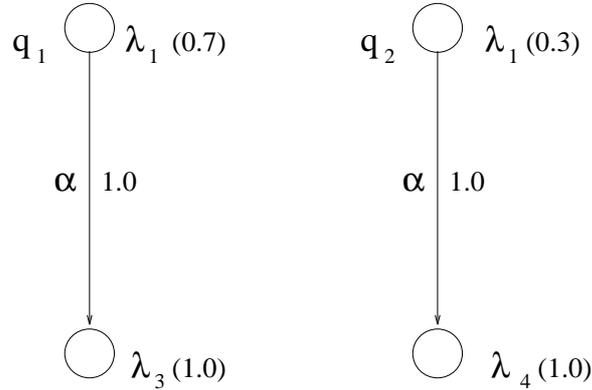


Figura 11: la frequenza di visita deve essere uniforme. Nella situazione prospettata in figura, se q_2 venisse visitato 3 volte più frequentemente che q_1 , la struttura dell'automa verrebbe ricostruita in maniera non corretta.

- (e) se α_k^{-1} è l'inversa della azione α_k appena compiuta, incrementa di uno il valore di $T_{\alpha_k}(\lambda_i, \lambda_i)$ e di $T_{\alpha_k^{-1}}(\lambda_j; \lambda_i)$;

4. costruisci il grafo di transizione di un automa nel modo seguente:

$$\forall \alpha_k, \forall \lambda_m : (q_m, \alpha_k, q_n) \in \zeta \iff \lambda_n = \arg \max_{\lambda_j} T_{\alpha_k}(\lambda_m, \lambda_j).$$

Come si può vedere, l'algoritmo mantiene una stima delle matrici $\mathcal{N}(\alpha)$, ovvero costruisce una stima dell'automa stocastico. Affinché il procedimento sia corretto è necessario che la distribuzione stazionaria delle frequenze di visita degli stati sia uniforme (si veda la figura 11). Sebbene questa condizione non sia formalmente garantita, si può congetturare sulla base di dati empirici che la strategia di esplorazione adottata tenda ad equalizzare le frequenze di visita. La figura 12 mostra l'entropia² dell'istogramma delle frequenze di visita in funzione del numero di passi dell'algoritmo. Si vede che questa tende decisamente verso il minimo, indicando che la distribuzione delle frequenze di visita approssima la uniforme. A nostra conoscenza non vi sono risultati formali riguardanti la correttezza e la complessità di questo algoritmo, tuttavia per una sua versione semplificata si può dimostrare [60] che il risultato prodotto è corretto dopo un numero di passi polinomiale nella dimensione dell'automa. Questo risultato è riportato nell'appendice A.

²Ricordiamo che l'entropia di una distribuzione discreta $p(x_i)$ è definita (a meno del segno) come $\sum p(x_i) \log p(x_i)$. Questa quantità ha il suo (unico) minimo in corrispondenza della distribuzione uniforme e dunque è utile come misura della uniformità di un istogramma.

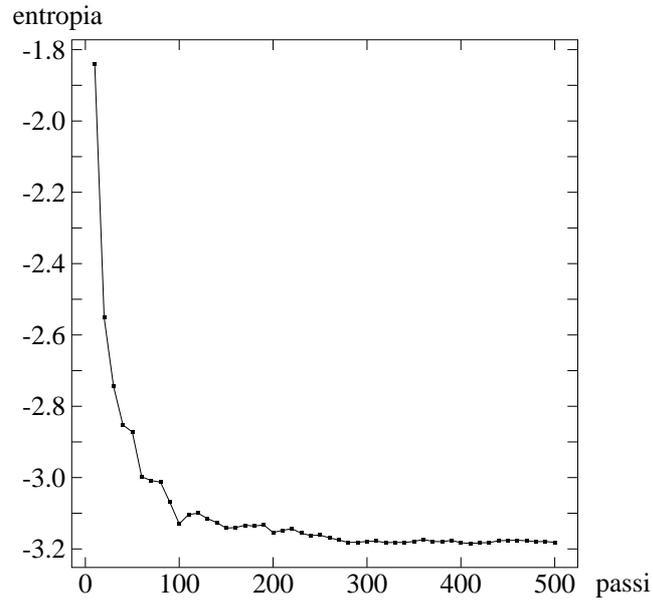


Figura 12: entropia della distribuzione delle frequenze di visita in funzione del numero di passi dell'algoritmo su di un grafo di 24 nodi. Il valore minimo è $\log(1/24) = -3.178$.

Abbiamo osservato che l'algoritmo 5.1, introdotto originalmente da Basye, non consente di trattare il caso in cui vengano osservate firme non corrispondenti ad alcuno stato dell'automa. In sostanza viene fatta l'ipotesi che per l'automa stocastico l'insieme delle firme osservabili coincida con il codominio di γ , ovvero con le firme degli stati dell'automa deterministico. Per estendere l'algoritmo 5.1 al caso in cui vi siano più firme osservabili che stati, si potrebbe allora modificarne il passo 4 nel seguente modo:

α			
	λ_1	λ_2	λ_3
λ_1	0	98	2
λ_2	0	100	0
λ_3	0	0	100

α^{-1}			
	λ_1	λ_2	λ_3
λ_1	100	0	0
λ_2	100	0	0
λ_3	100	0	0

Figura 13: tabelle prodotte dall'algoritmo di esplorazione nel caso di osservazione di una firma *outlier*. Si sfrutta la simmetria delle due per rilevare la situazione anomala

Algoritmo 5.2

.....

4. costruisci il grafo di transizione di un automa nel modo seguente:

$$\forall \alpha_k, \forall \lambda_m : (q_m, \alpha_k, q_n) \in \zeta \iff \lambda_n = \arg \max_{\lambda_j} T_{\alpha_k}(\lambda_m, \lambda_j) \wedge \lambda_m = \arg \max_{\lambda_j} T_{\alpha_k^{-1}}(\lambda_n, \lambda_j).$$

Il vincolo di reversibilità viene opportunamente sfruttato per eliminare le transizioni tali che $(q_m, \alpha_k, q_j) \in \zeta \wedge (q_j, \alpha_k^{-1}, q_m) \notin \zeta$. Queste transizioni sarebbero presenti in un automa costruito secondo l'algoritmo 5.1 come artefatti dell'osservazione sporadica di firme non corrispondenti ad alcun LDP. Per esempio si consideri la situazione in cui λ_1 è stata correttamente osservata tutte le volte, tranne una in cui è stata osservata una firma λ_3 che non corrisponde in realtà ad alcuno stato. Le tabelle prodotte dall'algoritmo in questo caso sono riportate in figura 13. Si vede che la transizione $(q_3, \alpha_0^{-1}, q_0)$ soddisfa la prima condizione ma non la seconda.

Basye riferisce che l'algoritmo originale si è mostrato appropriato per la costruzione della mappa di un piano di un edificio, strutturato in corridoi (maggiori dettagli si trovano in [6]). Alcuni esperimenti preliminari consistenti nel simulare l'azione dell'algoritmo 5.2 su di un grafo riprodotto la struttura dell'istituto verranno illustrati nel capitolo 7.

5.3 Applicazione

In questa sezione delineremo l'applicazione dell'algoritmo alla esplorazione e costruzione di mappe in un ambiente reale. Altri dettagli saranno riportati nel capitolo 7.

5.3.1 Azioni e firme

Le azioni (o *navigation procedures*) di cui è capace il veicolo dal punto di vista dell'algoritmo sono 4, e verranno indicate dalle lettere **N**, **S**, **W**, **E**. Ad esempio l'azione **N** ha come effetto che il veicolo si orienta verso Nord (le direzioni sono puramente convenzionali) e percorre il corridoio che si diparte in quella direzione fino al primo incrocio, nel quale si ferma. La firma di un LDP è costituita da due campi: i) la posizione approssimata del centro dell'incrocio, e la relativa matrice di covarianza (che ne rappresenta l'incertezza), ii) una etichetta che rappresenta la "forma" dell'incrocio come viene percepita dal robot. Per esempio l'etichetta *NSE* si riferisce ad un incrocio in cui incidono 3 corridoi, lungo le direzioni *N,S,E*. Si noti che l'azione **W** lascerebbe il veicolo nella posizione di partenza.

5.3.2 Matching

Il passo (*d*) dell'algoritmo prevede un operazione di confronto (*matching*) della firma corrente con le firme nella lista. La quantità in base alla quale si effettua il confronto è la distanza di Mahalanobis tra le posizioni (x, y) registrate nelle rispettive firme. Se la distanza minima sulla lista corrente è inferiore ad una soglia fissata, la firma corrente viene identificata con quella che realizza il minimo, altrimenti viene riconosciuta come nuova e inserita considerate come stime dello stesso parametro (la posizione della lista). Nel caso di confronto positivo, le due posizioni, centro del LDP) vengono combinate (*merging*) come spiegato nel capitolo 4 e la risultante sostituisce nella lista la vecchia posizione. Il confronto viene effettuato sul sottoinsieme di firme che possiedono le stesse etichette di forma. Anche se una momentanea occlusione dovesse confondere i sensori e la forma del LDP corrente non venisse riconosciuta correttamente, la firma risultante costituirebbe una osservazione sporadica (*outlier*) e verrebbe scartata al passo 4 dell'algoritmo.

Capitolo 6

Una griglia cartesiana dai dati sonar

In questo capitolo sarà introdotto un modello geometrico dell'ambiente ed un metodo che ci consentirà, mediante utilizzo di sensori ad ultrasuoni, di ricavare informazioni sulla sua struttura spaziale.

6.1 Rappresentazione analogica

In un ambiente interno caratterizzato dalla presenza di grandi superfici verticali (le pareti) appare piuttosto naturale rappresentare il mondo mediante la sua proiezione ortogonale sul piano del pavimento. Non diversamente da quanto viene fatto in molte realizzazioni cartografiche (mappe turistiche di città, per esempio), si riesce ad ottenere – a fronte di una perdita di informazione generalmente trascurabile – una rappresentazione compatta e maneggevole. Il modello analogico/geometrico che adotteremo è una immediata rappresentazione delle informazioni percettive “grezze” provenienti dai sensori. Ragioni di semplicità e di costo computazionale ci suggeriscono di rappresentare il mondo come un insieme di cellette elementari. Fissato un sistema di coordinate cartesiane bidimensionale solidale col pavimento su cui si muove il veicolo, una colonna di cellette nello spazio viene proiettata sul piano dando in tal modo luogo ad una griglia bidimensionale. La scelta di celle di lato uguale a 10 cm si è dimostrata un buon compromesso tra accuratezza e carico computazionale risultante. La rappresentazione per le mappe locali segue dunque l'approccio *grid-based*

discusso nel capitolo 2. Idealmente, la griglia di occupazione potrebbe essere a valori binari (una cella è o libera o occupata), ma la necessità di tenere conto dell'imprecisione del sensore fisico fa sì che sia interessante poter attribuire, a ciascuna cella, una *probabilità di occupazione* risultante da un modello esplicito [49] o implicito [14] del sensore.

Il sensore più semplice e più economico attualmente disponibile (e per questo anche quello di gran lunga più popolare) che sia capace di fornire dati relativi alla occupazione di porzioni di spazio è senza dubbio il sonar. Nella prossima sezione vedremo come dati provenienti da un tale sensore, situato a bordo del veicolo mobile, possano essere elaborati al fine di costruire una griglia di occupazione.

6.2 Aggiornamento della mappa con la misura sonar

La modellizzazione del sensore ad ultrasuoni è ampiamente trattata nella Appendice B; ci limiteremo pertanto qui ad esporre brevemente le caratteristiche essenziali del sensore. Il telemetro ultrasonico (o sonar) è un dispositivo in grado sia di emettere che di ricevere onde sonore di frequenza fissata. Il fatto che il dispositivo rilevi un'onda sonora in arrivo viene attribuito al fatto che un qualche oggetto ha in realtà riflesso un'onda che il dispositivo aveva emesso qualche tempo prima. Gli intervalli di tempo intercorrenti fra la emissione delle onde e la ricezione delle eco di ritorno porgono le distanze alle quali sono collocati gli oggetti riflettenti. Il meccanismo appena delineato, tuttavia, funziona fintantoché l'angolo (azimut) fra l'asse del sensore e la normale alla superficie riflettente è contenuto in un certo intervallo piuttosto limitato. Il valore massimo dell'angolo sotto il quale un oggetto può essere "percepito" prende il nome di angolo di visibilità ed è pari circa a $\pm 12^\circ$. Il sensore non ritorna però l'azimut dell'oggetto rilevato, dunque è caratterizzato da una intrinseca incertezza angolare di cui è necessario tenere conto.

Il metodo più diffuso per scansionare l'ambiente è equipaggiare il robot con un certo numero di sonar montati su un anello orizzontale attorno al veicolo, oppure dotarlo di uno o più sensori ruotanti. Il risultato è comunque una scansione circolare. Tipicamente questo metodo di acquisizione prevede che il robot, da fermo, prenda

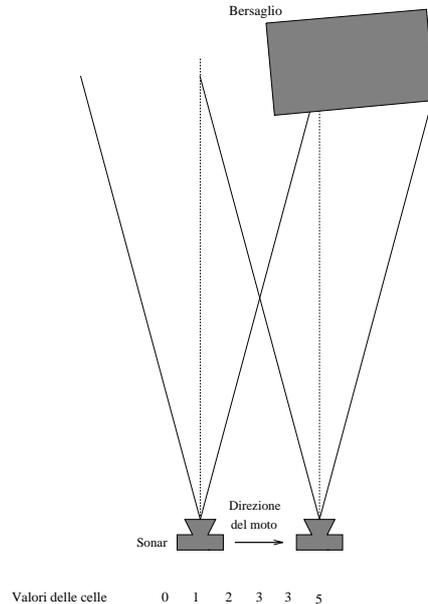


Figura 14: metodo di acquisizione in movimento come campionamento di una distribuzione. Via via che il bersaglio viene visto dal sensore in movimento con un angolo minore, aumenta la frequenza delle osservazioni.

una scansione panoramica, si sposti e ripeta l'operazione [49]. Nel nostro caso, seguendo Borenstein e Koren [14], il veicolo raccoglie dati sonar mentre si muove. Per ogni lettura sonar viene incrementato il valore di occupazione della cella che si trova sull'asse del sensore alla distanza misurata dal sonar (punto TOF). Il procedimento può sembrare ipersemplificato, in realtà se assumiamo che il sensore sonar risponda ad un modello probabilistico, in cui la probabilità di vedere un bersaglio è funzione dell'azimut del bersaglio stesso, un singolo oggetto sarà visibile con probabilità diverse mentre il veicolo si muove, poiché percorrerà da un estremo all'altro tutto l'angolo di visibilità e quindi in realtà il procedimento è assimilabile al campionamento di una distribuzione ignota. Otteniamo così una griglia i cui valori delle singole celle specificano probabilità di occupazione, come nelle *Occupancy Grids*.

Abbiamo individuato alcuni vantaggi in questo approccio:

- velocità
- considerando che il veicolo si muove tipicamente lungo corridoi, e quindi (per come è costruito il riflesso di avanzamento nel corridoio) è orientato parallelamente alle pareti, esse si trovano in pieno angolo di visibilità di un sensore

montato lateralmente sul veicolo.

- non necessita di un modello euristico esplicito del sensore, ma ne sfrutta implicitamente le caratteristiche stesse.

In sostanza quello che facciamo è campionare una variabile spaziale – la densità spaziale di occupazione rilevata dal sensore, risultante dalla convoluzione della densità reale con un nucleo che rappresenta la funzione di trasferimento del sensore – quindi il sonar si deve muovere con velocità uniforme rispetto al bersaglio (in questo caso le pareti dell’edificio), oppure il valore di cui viene incrementata ciascuna cella v’è normalizzato rispetto alla velocità. La velocità con cui un immaginario raggio sonar scandisce la superficie di un oggetto stazionario nell’ambiente è data approssimativamente da :

$$v_s = v + \omega r$$

dove v è la velocità di traslazione del veicolo, ω è la sua velocità di rotazione e r è la distanza dall’oggetto fornita dal sonar (la lunghezza del raggio di scansione). La cella viene incrementata di un valore intero Kv_s , dove K è una costante arbitraria di normalizzazione.

L’informazione che il sonar fornisce, però, non specifica solamente l’occupazione di una cella, ma ci dice anche che tutte le celle lungo l’asse del sensore, fino a quella occupata, sono libere. L’interpretazione di questo dato si traduce nel decrementare ciascuna delle n celle che giacciono lungo l’asse del sonar e che distano meno di r da quest’ultimo, di una quantità Kv_s/n .

Il valore contenuto in una cella specifica una probabilità di occupazione ¹ in una gamma che per comodità fissiamo in 0..255 (per l’utilizzo che si fa della mappa, considerato anche che spesso viene binarizzata, la risoluzione sembra più che sufficiente). Ogni qual volta una cella dovesse superare il valore di 255 o assumere un valore negativo, l’intera mappa verrebbe rinormalizzata. La figura 15 mostra le griglie di occupazione di alcuni corridoi dell’Istituto.

Nel passaggio dalla misura di profondità sonar all’incremento di una cella della griglia cartesiana viene utilizzata la posizione assoluta del veicolo fornita dall’odometro per indirizzare la cella appropriata. Se potessimo localizzare in modo assoluto e preciso il veicolo in ogni istante, le mappe che si otterrebbero sarebbero soddisfacenti.

¹a rigore specifica una frequenza di osservazione dell’ occupazione della cella

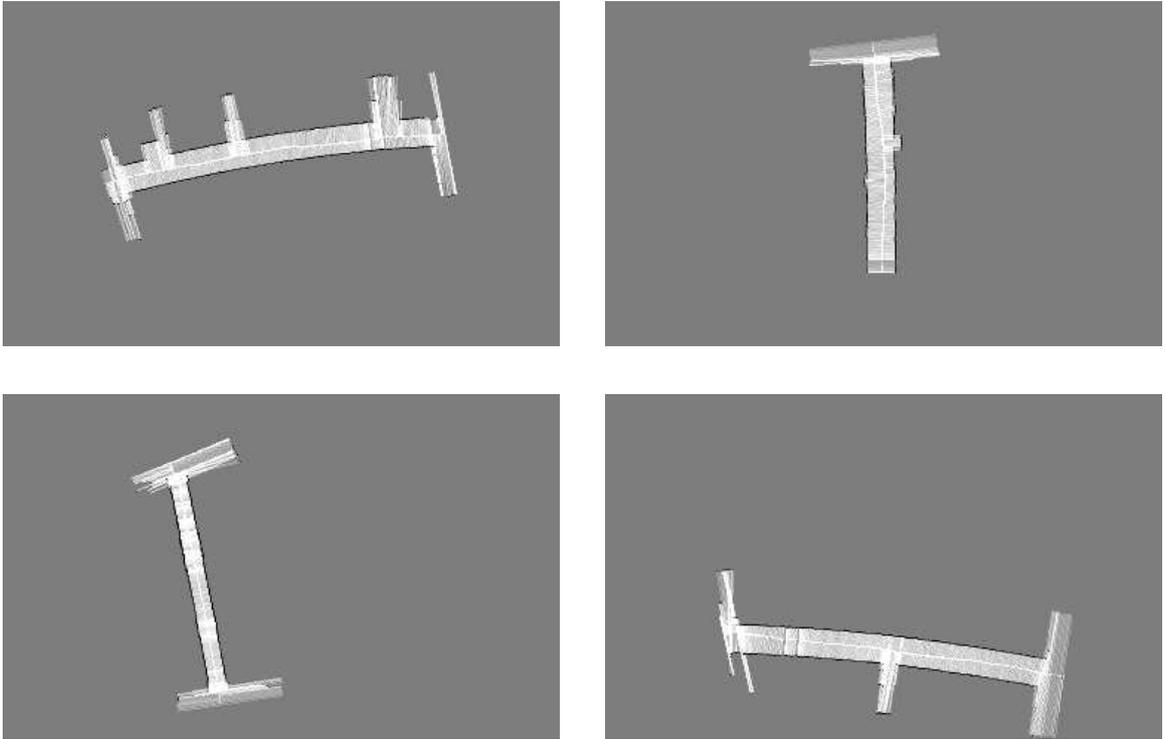


Figura 15: griglie di occupazione di 4 corridoi dell'Istituto. Il livello di grigio rappresenta la probabilità di occupazione : le zone chiare sono probabilmente libere, le zone scure sono probabilmente occupate. Il grigio dello sfondo rappresenta una probabilità di occupazione di 0.5.

Purtroppo l'odometro è affetto da un errore cumulativo che rende inaffidabili e inconsistenti su grande scala, le osservazioni – che pure localmente sono accurate. Per questo motivo non è praticabile l'approccio *naive* alla sintesi di mappe: si rende necessario esplicitare le proprietà intrinseche della struttura dell'ambiente, costruendo un modello topologico, al quale si demanda il mantenimento della consistenza globale.

6.3 Geometria dell'acquisizione

Vediamo più nel dettaglio come viene aggiornata la mappa. La posizione del robot è espressa dal vettore

$$\mathbf{p} = (p_x, p_y, \theta)^\top,$$

dove p_x e p_y rappresentano la sua posizione in un sistema di riferimento globale e θ è l'orientazione, espressa come una rotazione attorno all'asse z – più precisamente è

l'angolo formato dall'asse delle x con lo *heading* del veicolo. Introduciamo un sistema di riferimento solidale con il veicolo, centrato nel punto che ne identifica la posizione e orientato in modo che l'asse x corrisponda con la direzione di moto in avanti del veicolo. In questo modo l'orientazione θ del veicolo è l'angolo di cui è ruotato questo sistema di riferimento rispetto al sistema di riferimento globale. La posizione di un sonar è espressa nelle coordinate del veicolo da un vettore

$$\mathbf{s}' = (s'_x, s'_y, \phi')^\top,$$

in cui ϕ' è l'angolo formato dall'asse x' con l'asse del sonar. È immediato verificare che la posizione e l'orientazione del sonar nel sistema di riferimento globale sono date da:

$$\begin{pmatrix} s_x \\ s_y \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} + \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} s'_x \\ s'_y \end{pmatrix}$$

$$\phi = \phi' + \theta.$$

Il punto TOF corrispondente ad una lettura di profondità r fornita dal sonar \mathbf{s} viene posto alla profondità r lungo l'asse del sonar, dunque ha coordinate cartesiane (globali)

$$\begin{pmatrix} t_x \\ t_y \end{pmatrix} = \begin{pmatrix} s_x \\ s_y \end{pmatrix} + r \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix}$$

Per esempio nel caso del sonar anteriore sinistro del nostro veicolo si ha

$$\mathbf{s}' = (0.5, 0.2, \pi/2)^\top,$$

e dunque

$$\begin{aligned} t_x &= p_x + 0.5 \cos \theta - (0.2 + r) \sin \theta \\ t_y &= p_y + 0.5 \sin \theta + (0.2 + r) \cos \theta. \end{aligned}$$

La cella discreta in cui cade il punto TOF viene incrementata come spiegato in precedenza, mentre le celle intersecate dal segmento che ha per estremi la posizione del sensore e il punto TOF vengono decrementate. Formalmente il segmento è il luogo dei punti tali che:

$$\begin{pmatrix} u_x \\ u_y \end{pmatrix} = \alpha \begin{pmatrix} s_x \\ s_y \end{pmatrix} + (1 - \alpha) \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad \text{con } 0 < \alpha < 1$$

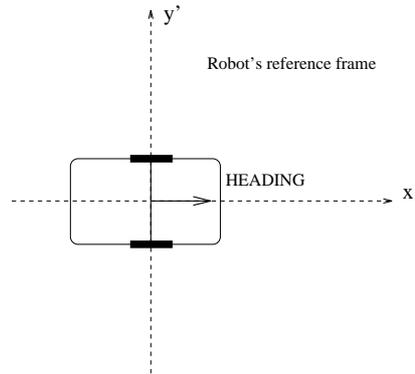


Figura 16: illustrazione del sistema di coordinate solido con il veicolo. L'asse y coincide con l'asse delle ruote motrici e l'asse x con la semiretta lungo la quale avviene il moto in avanti.

6.4 Possibili sviluppi

Abbiamo centrato il nostro lavoro sulla costruzione di una carta dell'ambiente, in cui la mappa topologica guida la costruzione del modello analogico. Classicamente, però, il livello simbolico giace sopra quello analogico e deriva da quest'ultimo attraverso un processo di astrazione. Infatti è possibile condensare una rappresentazione le cui primitive sono coni generalizzati o poligoni convessi di forme prestabilite; l'esplorazione di tale strada ci ha portati a sperimentare il *fitting* di rettangoli sugli spazi liberi della mappa binaria, in modo da descrivere lo spazio libero in forma simbolica come giustapposizione di rettangoli (tra l'altro questa è la rappresentazione che utilizza *Navshell*, lo strato simbolico del sistema di navigazione di MAIA). Il veicolo ha navigato nell'Istituto servendosi di una mappa prodotta in questo modo, tuttavia abbiamo abbandonato questa strada poiché il metodo ci sembrava poco generale e con un eccessivo supporto euristico. La mappa di figura 17(a) è quella costruita con il metodo descritto nella precedente sezione. Si noti che l'errore nell'odometria fa sì che la rappresentazione sia inconsistente su grande scala (il quadrato non si chiude). La 17(d) è stata ottenuta dalla prima nel seguente modo:

1. viene calcolata la linea di mezzeria di ciascun corridoio, mediante un algoritmo di *line-fitting* che minimizza la deviazione assoluta;
2. viene calcolato l'angolo di cui è necessario ruotare ciascun corridoio affinché le mezzerie siano tra loro ortogonali;

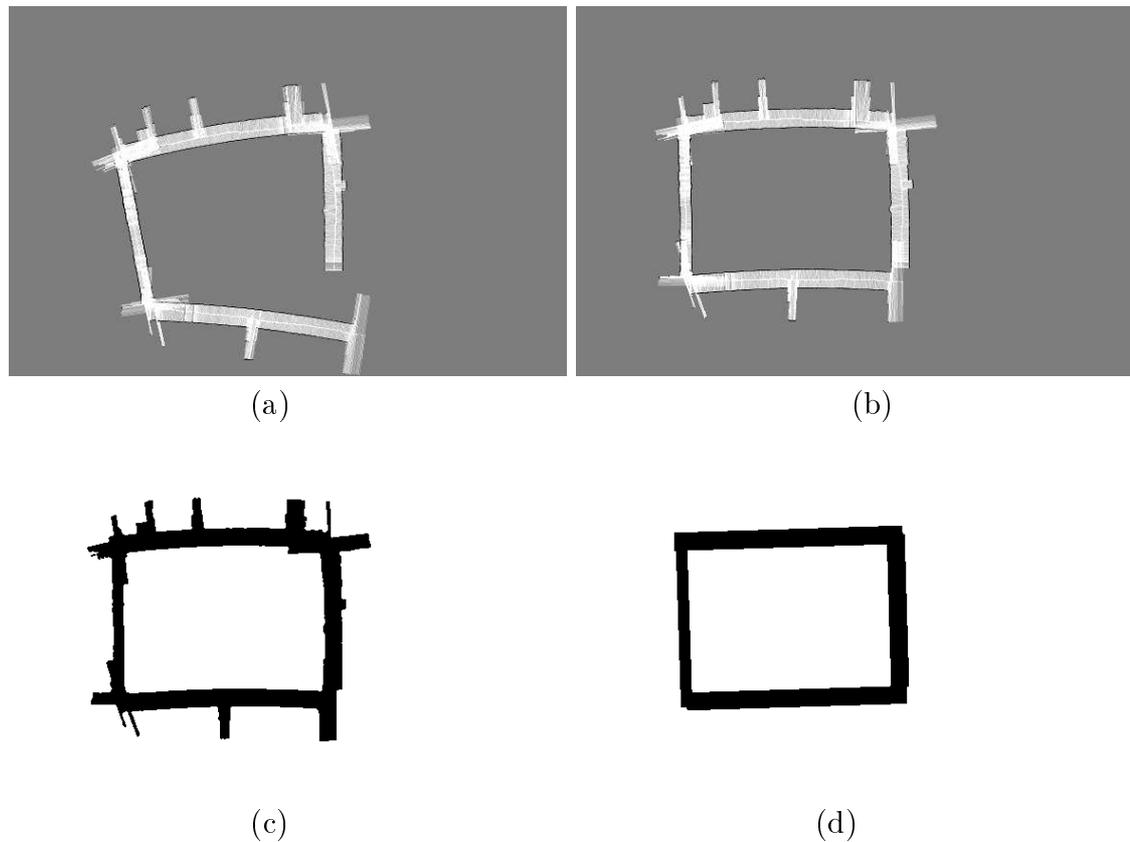


Figura 17: alcuni metodi *ad hoc* per le mappe analogiche. a) griglia di occupazione così come si ottiene dai dati sonar e odometrici; b) l'ortogonalità dei corridoi viene sfruttata per rendere consistente la mappa; c) sogliatura e filtraggio; d) adattamento di un modello a rettangoli.

3. viene applicata tale rotazione all'insieme delle misure sonar del corridoio stesso (figura 17(b));
4. la mappa viene binarizzata e filtrata con un algoritmo di rilassamento per eliminare celle occupate isolate (figura 17(c));
5. viene effettuato l'adattamento di rettangoli a partire dalle linee di mezzera, massimizzando separatamente una funzione di sovrapposizione nelle due dimensioni.

Il procedimento ha prodotto buoni risultati, ma sostanzialmente funziona perché, come si vede chiaramente dalla figura, l'odometro sbaglia molto nel registrare le rotazioni, mentre è più affidabile quando il veicolo trasla, con il risultato che le misure

prese in ciascun corridoio soffrono di un errore sistematico accumulato nella rotazione del veicolo all'imbocco del corridoio. Tale comportamento è un artefatto della meccanica del veicolo, e non è per nulla generale. È vero che la stima della posizione è soggetta ad un errore crescente con il moto del robot, ma in generale non è possibile localizzare nelle rotazioni il contributo principale.

Un altro possibile approccio consiste nell'incorporare esplicitamente nella mappa analogica l'errore (stimato) nella posizione del veicolo. In sostanza avremo mappe più "sfumate" ma non inconsistenti. Il formalismo per il trattamento dell'incertezza introdotto nel capitolo 4 può essere proficuamente applicato alle misure sonar. Vi è una trasformazione non lineare che lega \mathbf{t} a \mathbf{p} data dalle equazioni scritte nella sezione precedente. Poiché la posizione del robot è affetta da incertezza e supponendo che la distanza misurata dal sonar sia corretta, abbiamo un parametro incerto \mathbf{t} che dipende non linearmente da un altro parametro incerto \mathbf{p} di cui conosciamo la matrice di covarianza $\mathbf{\Lambda}_p$. La formula introdotta nel capitolo 4 ci consente di calcolare la matrice di covarianza $\mathbf{\Lambda}_t$ della posizione del punto TOF nel modo seguente:

$$\mathbf{\Lambda}_t = \mathbf{J}_f \mathbf{\Lambda}_p \mathbf{J}_f^\top.$$

In particolare

$$\mathbf{J}_f = \begin{pmatrix} 1 & 0 & -s'_y \cos \theta - (s'_x + r) \sin \theta \\ 0 & 1 & -s'_y \sin \theta - (s'_x + r) \cos \theta \end{pmatrix}$$

Il punto TOF ha una regione di incertezza associata, quindi è rappresentato da una gaussiana di covarianza assegnata piuttosto che da un impulso. L'aggiornamento della mappa interesserà non più una sola cella, ma tutto un suo intorno, in misura determinata da una distribuzione gaussiana. Il risultato è che la mappa diventerà sempre più sfumata con il procedere del veicolo. Non abbiamo risultati riguardanti quest'ultimo approccio.

In questo capitolo abbiamo visto con quali tecniche di acquisizione sia possibile costruire mappe geometriche dell'ambiente. Nel prossimo capitolo descriveremo ad un maggiore livello di dettaglio l'architettura e il funzionamento del sistema di cartografia.

Capitolo 7

Il sistema di cartografia

7.1 Riflessi e percezioni per l'esplorazione

Per costruire le *procedure di navigazione* (mediante le quali il *learner* interagisce con il mondo) si è ritenuto opportuno dotare il veicolo di un certo numero di funzionalità sensomotorie “semplici” che opportunamente strutturate realizzassero la procedura richiesta. In particolare le funzionalità di movimento sono tre: avanzare nel corridoio corrente, ruotare sul posto e muoversi semplicemente in avanti. Queste funzionalità, nel linguaggio introdotto precedentemente, vengono indicate come *riflessi*. I *riflessi* terminano quando determinati eventi vengono rilevati; il veicolo è in grado di rilevare eventi quali la presenza di ostacoli tali da impedire il passaggio del robot, la presenza di un eventuale corridoio laterale. Nella terminologia introdotta precedentemente, la combinazione di *riflessi* con condizioni di terminazione basate sugli eventi rilevati viene detta *comportamento elementare*. È un esempio di comportamento elementare: “Avanzare lungo il corridoio fino all’apertura di un corridoio laterale”.

Le *percezioni* sono indispensabili per l’attività di esplorazione poiché permettono di rilevare la struttura dell’ambiente. Il veicolo è dotato di capacità percettive che gli permettono di conoscere la sua posizione in un sistema di coordinate globale, di misurare la distanza (*range*) di “bersagli” e di rilevare le direzioni lungo le quali vi è spazio libero sufficiente per il suo passaggio.

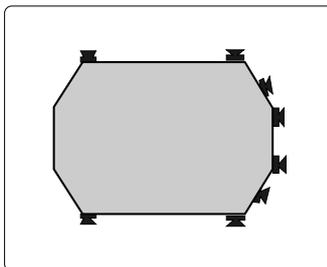


Figura 18: vista in pianta del robot. Si noti la disposizione dei sonar: due coppie laterali per rilevare le pareti del corridoio, gli altri disposti frontalmente per la rilevazione degli ostacoli.

7.1.1 Caratteristiche del Robot e dell'Ambiente di Lavoro

Le caratteristiche del robot e dell'ambiente in cui esso si deve muovere vanno tenute in debito conto, sia nella scelta del tipo di sensori utilizzati e della loro disposizione, sia nel progetto del sistema di controllo. L'ambiente che il robot deve esplorare è costituito dall'insieme dei corridoi dell'I.R.S.T, nei quali vi sono ostacoli fissi – come radiatori, piante, armadi, stampanti – e ostacoli in movimento – le persone o eventuali altri robot.

Il robot è dotato di otto telemetri ultrasonici (*sonar*) in grado di misurare una distanza nella gamma $[0.2m, 4.0m]$ (si veda l'appendice B). I sensori sono posizionati a bordo del robot come in figura 18: la disposizione cerca di conciliare le necessità di garantire la copertura dello spazio frontale al fine di evitare gli ostacoli, e di localizzare le pareti del corridoio in cui il robot si muove. Il robot dispone di un sistema di controllo che consente di pilotare in velocità i motori delle due ruote motrici. L'odometro, in base al numero dei giri dei motori (si veda il capitolo 4), mantiene aggiornata la posizione e direzione del robot in un sistema di riferimento cartesiano assoluto.

7.1.2 Riflessi

Il Riflesso di Avanzamento nel Corridoio

Il Riflesso di Avanzamento nel Corridoio [30] è stato costruito quale composizione di quattro riflessi di livello più elementare, e più precisamente i riflessi di

- Allineamento alle pareti
- Mantenimento della zona centrale

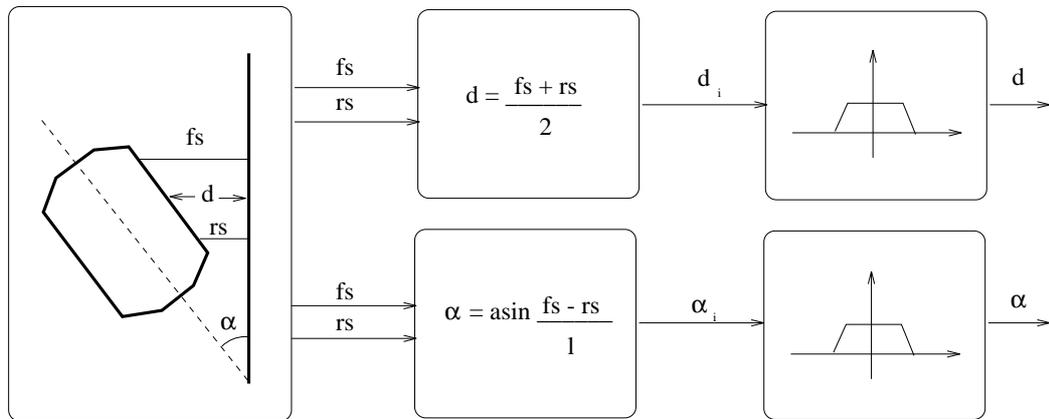


Figura 19: stima della direzione e della distanza della parete laterale. La media delle misure fornite dai due sonar laterali (fs e rs) fornisce una stima della distanza della parete dal robot. L'arcoseno della differenza, sulla distanza tra i due sonar, fornisce la stima dell'angolo della parete, rispetto all'asse del robot.

- Aggiramento di piccoli ostacoli
- Avanzamento

I primi tre riflessi modulano la velocità di rotazione in funzione della velocità di traslazione, l'ultimo la velocità di traslazione.

L'output del Riflesso di Avanzamento nel Corridoio è una combinazione lineare degli output dei quattro riflessi elementari, secondo pesi che sono stati calibrati sperimentalmente sull'ambiente di lavoro. La sintassi del riflesso è: **Follow-corridor**

Allineamento alle pareti

La stima della direzione e distanza delle due pareti del corridoio, rispetto all'asse del robot, viene effettuata sulla base di considerazioni geometriche sulle misure di distanza fornite dai sonar laterali. Tali stime vengono stabilizzate tramite un filtro passa basso che ha il duplice scopo di ridurre gli effetti sia del rumore di misura intrinseco ai sensori utilizzati, sia di oggetti o persone in movimento, che portano necessariamente a stime istantanee erranee della direzione o della distanza della parete (figura 19).

L'angolo massimo fisicamente rilevabile è funzione dell'ampiezza del cono di emissione dei sonar e della distanza reciproca tra le coppie di sonar laterali. Quando

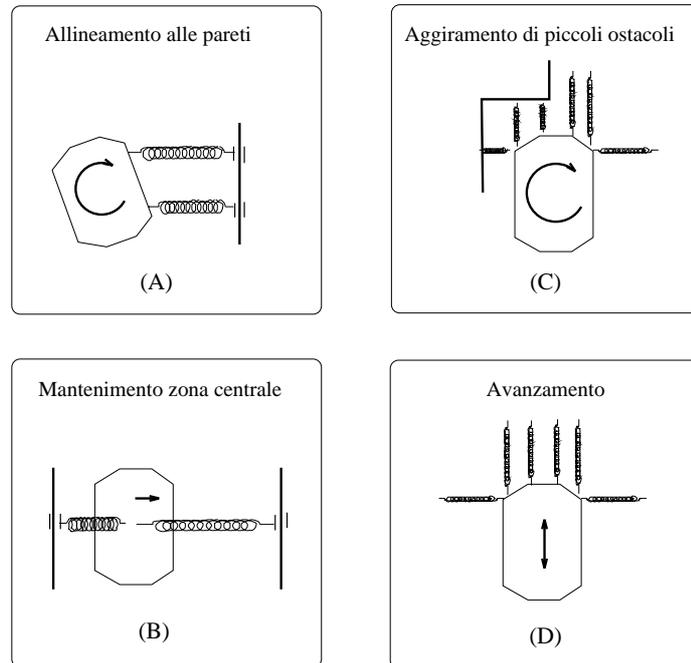


Figura 20: il riflesso di avanzamento nel corridoio è composto di quattro riflessi più elementari: (A) Allineamento alle pareti; (B) Mantenimento della zona centrale; (C) Aggiramento di piccoli ostacoli; (D) Avanzamento.

l'angolo formato dall'asse del robot con la parete supera tale angolo massimo, la direzione e distanza della parete vengono aggiornate solo sulla base del movimento del robot, stimato sulla base dei dati forniti dall'odometro.

La direzione del corridoio viene quindi calcolata come media pesata delle direzioni delle due pareti, con pesi che tengono conto della stabilità temporale delle due stime.

Stimata la direzione del corridoio, il riflesso di allineamento alle pareti tende ad allineare il robot al corridoio imponendo una rotazione di verso opportuno e velocità proporzionale all'angolo di disallineamento per angoli minori di 45° , costante per angoli maggiori.

Detto α l'angolo formato dall'asse del robot con l'asse del corridoio e ω la velocità di rotazione del robot stesso ($\omega = \dot{\alpha}$), si avrà

$$\omega(t) = -k\alpha(t) \quad (1)$$

con k opportuna costante positiva. Si tratta cioè di un sistema del primo ordine: a partire da una posizione iniziale non nulla ($\alpha \neq 0$), il robot tenderà ad allinearsi al corridoio con andamento esponenziale decrescente.

È stata recentemente sviluppata una funzionalità per il computo dell'orientazione relativa del veicolo rispetto alla direzione del corridoio, basata su visione monoculare [31]. Come noto, segmenti che nel mondo 3-d sono paralleli alla direzione del corridoio appaiono in una immagine come convergenti verso un punto: il punto di fuga. Lo scostamento laterale del punto di fuga dal centro ottico dell'immagine determina l'angolo tra l'asse della telecamera (coincidente con l'asse avanti-dietro del veicolo) e la direzione del corridoio.

Mantenimento della zona centrale

Interviene solo quando la distanza di una delle due pareti scende sotto soglia: in tale situazione la velocità di rotazione è inversamente proporzionale alla distanza della parete. Se entrambe le pareti sono sotto soglia la velocità di rotazione viene posta uguale a zero.

Se d è la distanza dalla parete, si avrà quindi

$$\omega(t) = \frac{h}{d(t) - d_s} \quad (2)$$

con h e d_s opportune costanti positive.

Aggiramento di piccoli ostacoli

Si attiva quando si individua un ostacolo, frontalmente o lateralmente, sotto la soglia di intervento. La velocità di rotazione è direttamente proporzionale alla differenza tra la distanza stimata dell'ostacolo più vicino nella parte destra del robot, e quella dell'ostacolo più vicino nella parte sinistra.

Le distanze degli ostacoli lato destro e sinistro vengono valutate quali funzioni lineari delle misure dei diversi sonar.

$$d_{l,r} = \sum_{i=1}^n c_i (s_i)_{l,r} \quad (3)$$

Avanzamento

La velocità di traslazione a regime è proporzionale alla distanza dell'ostacolo più vicino, posto frontalmente o lateralmente al robot. Un ostacolo sotto la soglia di

sicurezza provoca l'arresto del robot.

$$v = k \min_i \{FrontSonar_i\} \quad (4)$$

dove $FrontSonar_i$ rappresenta la distanza misurata dall' i -esimo sonar frontale. Il riflesso di avanzamento è attivabile anche da solo (per esempio per muoversi in spazi aperti come ad esempio atrî od aree comuni). Sintassi: **Move**.

Il riflesso di rotazione

La rotazione viene effettuata sul posto, cioè con velocità di traslazione nulla. I sonar vengono utilizzati per modulare la velocità di rotazione in funzione della distanza dell'ostacolo più vicino, o per arrestare la rotazione, qualora venga rilevato un ostacolo sotto la soglia di pericolo. Sintassi: **Rotate**.

7.1.3 Predicati

Rilevazione dell'angolo di rotazione

Verifica, leggendo l'odometro, se l'orientazione del veicolo è cambiata di un angolo superiore ad un certo valore. Sintassi: **Rotated-angle-gt <float>**

Rilevazione di un'occlusione frontale

L'occlusione viene rilevata mediante i sonar frontali. Per garantire che sia una occlusione che non consente l'aggiramento, tutte le misure di distanza frontale devono essere sotto soglia, inoltre la situazione deve perdurare per un certo tempo fissato. Tipicamente così viene rilevata la parete che chiude un corridoio cieco. Sintassi: **Frontal-wall-detected**

Rilevazione del corridoio laterale

Per la rilevazione del corridoio laterale viene utilizzata la coppia di sonar laterali. Viene individuato il corridoio quando entrambi i sonar laterali misurano una distanza che supera quelle della parete di una quantità minima prefissata (figura 21). Sintassi: **Lateral-corridor-detected**

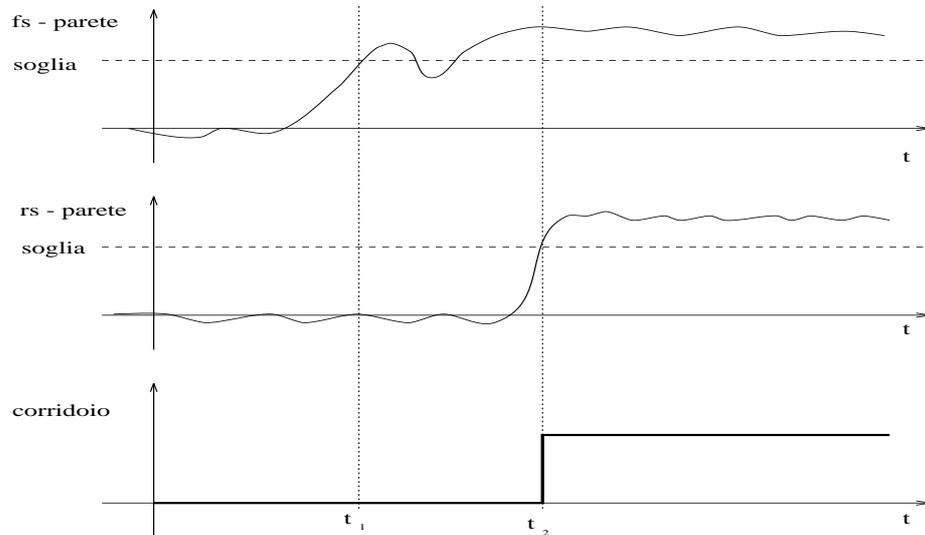


Figura 21: rilevazione del corridoio laterale. Il corridoio viene individuato quando entrambi i sonar laterali misurano una differenza, rispetto alla stima corrente della parete, che supera la soglia di innesco (al tempo t_2 in figura).

Di fatto viene quindi rilevato solo un allontanamento della parete. Se ciò corrisponda effettivamente alla presenza di un corridoio laterale non è evidentemente rilevabile in questo modo: una porta aperta sufficientemente larga o una rientranza del corridoio possono venire rilevati allo stesso modo di un corridoio laterale. Dal punto di vista del modello funzionale dell'ambiente questa distinzione puramente semantica non ha importanza, purché l'osservazione sia ripetibile.

7.1.4 Percezioni

Percezione della posizione assoluta

Riporta la lettura dell'odometro, quindi la posizione assoluta ed orientazione del robot (x, y, θ) in un sistema cartesiano, assieme ad una matrice 3×3 che rappresenta l'incertezza associata alla misura. (È una matrice di covarianza, la quale individua una regione di incertezza ellittica). L'odometro (arricchito) è stato implementato come descritto nel capitolo 4. Sintassi: `Get-position`

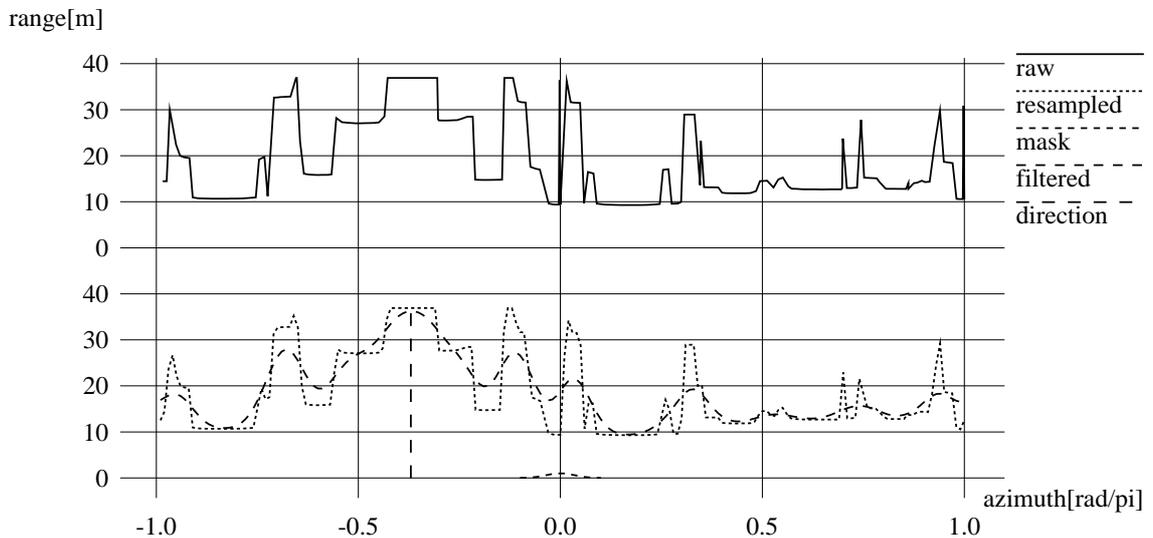


Figura 22: scansione presa in un corridoio in prossimità di una fine cieca. Con la linea continua è tracciato il diagramma delle distanze così come viene acquisito dai sonar. In tratteggio (più fitto) è disegnato lo stesso profilo ricampionato con passo costante e filtrato (meno fitto) con una maschera gaussiana. La linea verticale indica la direzione riconosciuta correttamente come non occlusa.

Percezione delle distanze

Ha lo scopo di registrare continuamente la posizione in coordinate cartesiane globali dei punti TOF rilevati dai sonar. L'elaborazione consiste nel calcolo delle coordinate del punto a partire dalla posizione odometrica e dalla misura del tempo di volo fornita dal sonar, nota la posizione del sensore sul veicolo. Sintassi: `Get-tof-dots`

Percezione degli spazi liberi

Questa percezione ha lo scopo di individuare direzioni in cui non vi siano occlusioni, dunque potenzialmente percorribili dal veicolo. Se attivata in un luogo dove si incrociano più corridoi è supposta fornire le direzioni dei corridoi incidenti. Il robot effettua una rotazione di 180° durante la quale raccoglie misure con i sonar laterali destro e sinistro, componendo così un profilo di profondità locale (in forma polare). I dati grezzi vengono prima ricampionati con passo costante e quindi sottoposti a convoluzione circolare con una maschera gaussiana. I massimi assoluti del diagramma di profondità che si ottiene, rappresentano le direzioni cercate. L'euristica alla base di

questo metodo è consiste nel privilegiare le zone del diagramma di elevata profondità ed estensione angolare: così picchi dovuti al rumore o ad echi sonar multipli, vengono tagliati, mentre i pianori di massima profondità conservano inalterato il valore al centro (si veda la figura 22). Sintassi: `Get-free-direction`

7.2 Navigazione autonoma: il Navigator

Il *navigator* è il modulo che gestisce la navigazione nel mondo reale e l'acquisizione di dati sensoriali in modo funzionale alle necessità del *learner*, oltre che effettuare la costruzione di mappe di occupazione locale ed il mantenimento della localizzazione.

7.2.1 Procedure di navigazione acquisizione della firma

Dopo avere introdotto le funzionalità elementari possiamo spiegare come avvengono le due attività fondamentali per l'esplorazione: l'esecuzione delle *procedure di navigazione* e l'acquisizione di una *firma*. Ricordiamo brevemente (si faccia riferimento al capitolo 3) che il *navigator* dialoga con *Action Manager* effettuando un *behaviour call* e ricevendo in risposta un *behaviour report*. La sintassi della *behaviour call* è la seguente:

(BASIC-BEHAVIOUR

```
:reflex <reflex>
:termination <termination>
:report <perception> )
```

<termination> è una condizione di terminazione espressa come congiunzione di *predicati*, la quale specifica gli eventi la cui rilevazione pone termine al riflesso. I dati forniti dalle percezioni specificate in <perception> vengono restituiti con il *navigation report* al termine dell'esecuzione.

Le procedure di navigazione previste, lo ricordiamo, sono **N**, **S**, **W**, **E**. Vediamo ad esempio come viene espansa dal *navigator* la procedura **N** il cui effetto, partendo da un incrocio, è di far percorrere al robot il corridoio che si diparte verso Nord (se c'è) fino al prossimo incrocio.

- il *navigator* calcola l'angolo α della rotazione che orienta il veicolo lungo la direzione (odometrica) in cui il nord è mappato;

- invia a A.M. la richiesta di esecuzione

```
(BASIC-BEHAVIOUR
  :reflex Rotate
  :termination (Rotated-angle-gt  $\alpha$ ) )
```

- chiede all A.M l'esecuzione di

```
(BASIC-BEHAVIOUR
  :reflex Follow-corridor
  :termination (OR (Lateral-corridor-detected)
                 (Frontal-wall-detected))
  :report (Get-tof-dots))
```

- al ricevimento da A.M. del *behaviour report* notifica al *learner* la terminazione della procedura. I dati sonar vengono invece elaborati e memorizzati dal *navigator* stesso in una mappa locale, nel modo che abbiamo discusso nel capitolo 6.

A questo punto il *learner* tipicamente richiede al *navigator* la *firma* del luogo. Ricordiamo che una firma ha la struttura seguente:

```
object SIGNATURE is
  shape:  SHAPE_TYPE;
  position:  ARRAY[3]{DOUBLE};
  uncertainty:  MATRIX[3]{DOUBLE};
end;
```

Il *navigator* procede all'attivazione, attraverso l'A.M. della percezione che rileva le direzioni di spazio libero, effettuando la chiamata:

```
(BASIC-BEHAVIOUR
  :reflex Rotate
  :termination (Rotated-angle-gt pi)
  :report (Get-free-direction Get-position))
```

Ricevendo il *navigation report*, in *navigator* valuta se le direzioni percepite siano consistenti con il vincolo di ortogonalità dei corridoi e con la causa di terminazione del riflesso di avanzamento. In caso di esito negativo mette in atto una strategia senso motoria che gli consenta di ottenere informazione sensoriale più affidabile; nell'implementazione attuale semplicemente ripete l'operazione. In caso positivo mappa le direzioni (espresse come angoli nel sistema di riferimento odometrico) nell'insieme N,S,W,E e ottiene così una etichetta della forma della giunzione. Ricavando dal report la posizione e la relativa matrice di covarianza, la firma del luogo viene così istanziata in tutti i suoi campi e viene comunicata al *learner*, assieme ad un puntatore (`local-map-tag`) alla mappa locale dell'ultimo corridoio percorso.

7.2.2 Stima della posizione

Uno dei compiti fondamentali di un navigatore è fare il punto. In qualsiasi momento l'attivazione della percezione `Get-position` fornisce la posizione e la matrice di covarianza, ma come discusso precedentemente, se non si riduce in qualche modo l'incertezza questa cresce rapidamente rendendo priva di significato la stima della posizione. Il meccanismo di reset che proponiamo emerge come effetto collaterale dell'esplorazione la quale essendo qualitativa e dunque immune da errori metrici, si suppone proceda in modo consistente. In sostanza ogni qual volta il *navigator* spedisce la firma del luogo attuale al *learner*, riceve indietro da questo la firma che viene inserita nella lista. Questa può essere la firma stessa se il luogo è riconosciuto come mai visitato, e in tal caso non si può fare nulla, oppure può avere la posizione cambiata in accordo all'algoritmo di fusione derivato dal filtro di Kalman (capitolo4). In questo caso il *navigator* prende atto del raffinamento della stima della posizione effettuando una chiamata diretta a `Set-position` che modifica la stima della posizione (e della matrice di covarianza) mantenuta dall'odometro.

Osserviamo per inciso che *navigator* potrebbe, in linea di principio, sfruttare altre fonti di informazione per la localizzazione o addirittura attuare particolari strategie senso motorie atte a ridurre l'incertezza sulla posizione (si pensi per esempio ad un riflesso di allineamento al corridoio basato su sonar e visione; se l'orientazione del corridoio nel sistema assoluto è noto, il veicolo può aggiornare la propria orientazione).

Immaginiamo ora che, terminata la fase di apprendimento della mappa, questa

venga utilizzata da un pianificatore. La pianificazione sul grafo prodotto dal *learner* è banale. Assegnando un costo pari alla lunghezza del corridoio a ciascun arco, dati i luoghi di partenza e quello di arrivo, si tratta semplicemente di trovare il cammino di costo minimo tra due nodi. La sequenza di *navigation procedure* che si ottiene è un piano immediatamente eseguibile dal *navigator*, il quale, continuando a notificare le firme dei luoghi visitati consente al pianificatore di monitorare l'esecuzione del piano. Questo dimostra due fatti: la adeguatezza delle mappe prodotte per la navigazione e la corretta ripartizione delle funzionalità tra i livelli dell'architettura .

7.3 Apprendimento ambientale: il Learner

Il *learner* è il modulo che realizza fisicamente l'algoritmo 5.2. Esso restituisce un grafo etichettato i cui vertici rappresentano luoghi dotati di firma, ed i cui archi rappresentano procedure di navigazione mediante le quali è possibile transitare da un luogo (vertice) ad un'altro. Sebbene il modello soggiacente sia intrinsecamente qualitativo, varrà la pena notare come dal grafo così ottenuto sia possibile estrarre ben di più della mera struttura topologica dell'ambiente: esso contiene infatti anche informazione metrica, relativa cioè, a distanze ed angoli. Ai vertici del grafo è attribuita una posizione assoluta (un campo della firma) e gli archi (ovvero i corridoi che essi rappresentano) possiedono una orientazione la quale corrisponde alla direzione prevalente che il veicolo segue durante la esecuzione della procedura di navigazione (nel nostro caso, Nord, Est, Sud, Ovest). In questo senso, dunque, il grafo viene dotato di una struttura metrica che può essere utilizzata per "guidare", ovvero vincolare, il problema di come le mappe locali dei corridoi possano poi essere incollate a formare un atlante.

L'incertezza relativa ai dati odometrici è trattata, dal *learner* mediante gli strumenti introdotti nel capitolo 4, consentendo pertanto al nostro sistema di esibire elevati livelli di "robustezza" rispetto al rumore di misura. In sintesi, l'informazione metrica è irrinunciabile – come unico modo di rendere non ambigue le firme dei luoghi – ma essendo affetta da errori è opportuno che non abbia un ruolo primario nell'apprendimento dell'ambiente, che deve invece seguire un metodo qualitativo. Dove l'informazione metrica viene usata, la rappresentazione esplicita dell'incertezza ad essa associata garantisce flessibilità e robustezza

Le informazioni, ovvero le firme dei luoghi che sono stati già visitati vengono organizzate in una lista, che viene gestita nel seguente modo:

- ogni firma in ingresso viene confrontata con quelle già presenti nella lista e l'identificazione avviene sulla base sia della forma (orientata) della giunzione, sia in base alla distanza di Mahalanobis tra le posizioni;
- se la firma viene identificata con una già presente nella lista, quest'ultima viene cancellata ed in suo luogo ne viene inserita una di uguale forma, ma la cui posizione viene ottenuta opportunamente mediando le posizioni della "vecchia" e della "nuova" firma. In questo modo, l'incertezza relativa alla posizione delle firme può essere ridotta;
- in caso contrario, cioè quando la firma in ingresso non viene riconosciuta equivalente ad alcuna già presente, essa viene aggiunta alla lista.

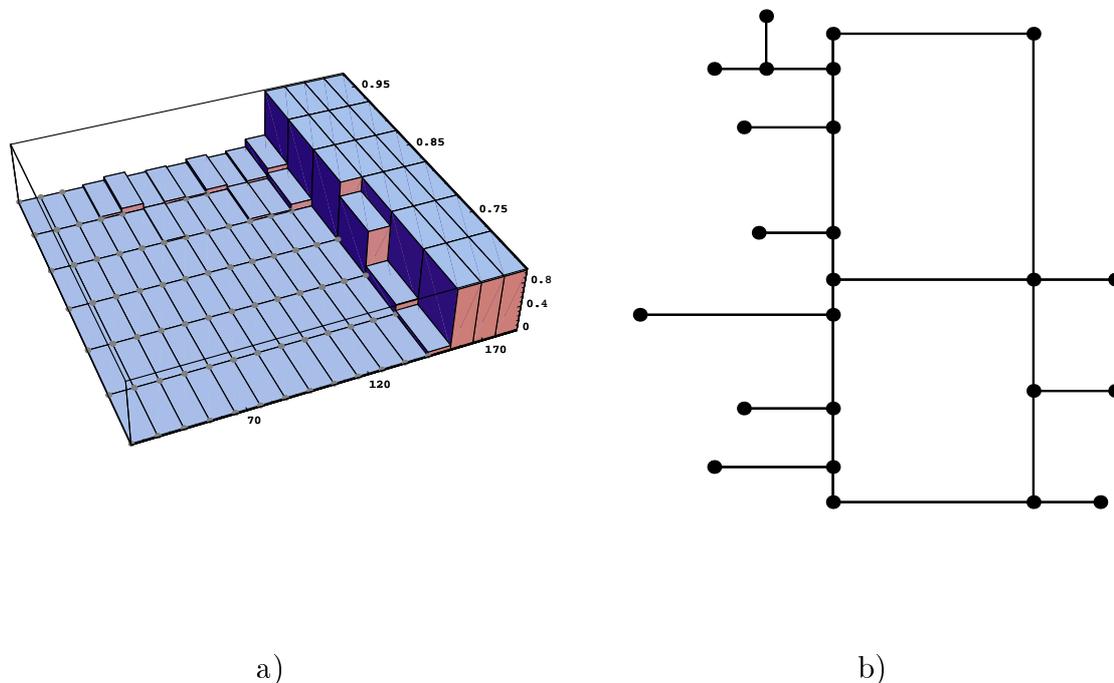


Figura 23: a) risultati dell’algoritmo. L’altezza di ciascuna barra è proporzionale alla frequenza di risultati corretti su 50 prove, con numero di passi e probabilità di errore fissata. b) grafo riproducente la struttura dell’IRST, utilizzato nella simulazione

7.3.1 Simulazione

Abbiamo simulato l’azione del *learner* su un grafo che riproduce la struttura dell’IRST (figura 23b). Il modulo che implementa il *learner* è quello che funzionerà a bordo del robot, mentre il *navigator* è un simulatore che naviga “virtualmente” su di un grafo predefinito e risponde coerentemente ai messaggi del *learner*. È possibile variare la probabilità con cui il simulatore fornisce la firma corretta e la probabilità con la quale esegue l’azione comandata. Nella simulazione le due probabilità sono state poste uguali. Il grafico di figura 23a) mostra le frequenze di risultati corretti su 50 prove al variare della probabilità e del numero di passi. Si vede come siano necessari almeno 150 passi dell’algoritmo per ottenere delle buone prestazioni, corrispondenti ad attraversare ciascun corridoio circa 6 volte.

Capitolo 8

Conclusioni e sviluppi futuri

Lo schema per la sintesi automatica di mappe di ambienti interni mediante esplorazione non guidata che è stato proposto in questa tesi può definirsi ibrido, inquantoché basato su procedure e moduli in grado di ricavare e conseguentemente elaborare informazioni sia di carattere numerico (misure da sonar, odometria), sia di carattere simbolico (modellizzazione dell'ambiente mediante grafo). Dopo aver presentato una panoramica piuttosto completa del lavoro che altri autori hanno recentemente svolto nel campo, abbiamo cominciato a delineare i termini del problema che andavamo ad affrontare e dei metodi dei quali intendevamo avvalerci. In particolare, abbiamo ammesso che *tutte* le informazioni accessibili al sistema – ovvero al robot mobile che materialmente effettua l'esplorazione – siano incerte, ovvero affette da errori di varia natura (errori di misura nel posizionamento del veicolo, nella localizzazione delle superfici riflettenti, nella etichettatura dei *Local Distinctive Place*). Per quanto riguarda la determinazione della posizione odometrica del veicolo, abbiamo quindi fatto ricorso a tecniche per il trattamento dell'incertezza basate su stime iterative della matrice di covarianza, mentre per quel che riguarda l'identificazione in presenza di errori del grafo che rappresenta la struttura topologica dell'ambiente ci siamo basati su di un algoritmo dovuto a Basye al quale abbiamo apportato una modifica che lo rende – riteniamo – più robusto in ambienti reali.

Per quanto riguarda le abilità senso-motorie del veicolo, abbiamo tratto vantaggio non meno che ispirazione dal lavoro di ricerca e di sperimentazione nel campo della navigazione automatica che viene attualmente realizzato all'Istituto per la Ricerca Scientifica e Tecnologica di Trento, dove questa tesi si è svolta. Allo stato attuale del

lavoro, tutti i moduli realizzati quali componenti del sistema descritto, si inseriscono compiutamente nella preesistente architettura funzionale del robot, della quale possiamo pensare essi verranno a far parte una volta sottoposti a verifica sperimentale estensiva.

Tra le difficoltà maggiori che il lavoro ha presentato, dovremo menzionare quella incontrata nell'affrontare il problema della auto-localizzazione del veicolo – questo sia dal punto di vista numerico (errore odometrico) sia dal punto di vista topologico (identificazione di luoghi). Se un odometro perfetto (o un qualunque altro dispositivo di posizionamento assoluto) esistesse, allora il lavoro qui esposto sarebbe quasi totalmente inutile. Tuttavia, un tale strumento non esiste (né è plausibile pensare che esisterà nel futuro ragionevolmente prevedibile) ed è stato quindi importante esplorare quali possibilità un approccio *spurio* quale quello da noi seguito apra alla soluzione del problema. Le modellizzazioni dei dispositivi che ci hanno consentito di trattare l'onnipresente incertezza in modo che riteniamo elegante, ma soprattutto robusto e flessibile, riteniamo ci permetteranno anche di effettuare esperimenti estesi ed esauritivi in un futuro che possiamo prevedere molto prossimo. Recentissimi esperimenti, sia pure molto preliminari[53], ci fanno sperare che il robot potrà esplorare e mappare l'ambiente dell'istituto in un tempo dell'ordine di alcune ore, e – sulla scorta delle informazioni così organizzate – eseguire poi missioni pianificando il movimento sul grafo appreso.

Molte sono le strade che rimangono aperte per ulteriori sviluppi ed indagini. Sul fronte più squisitamente teorico, la correttezza e la complessità dell'algoritmo 5.1 restano da dimostrare, mentre sul fronte empirico una più approfondita analisi sperimentale delle sue prestazioni potrebbe aiutare a delinearne meglio il suo campo di applicabilità e gli eventuali miglioramenti che potrebbero esservi apportati. Per quanto riguarda gli aspetti più applicativi, una grossa sfida risiede certamente nel tentativo di rendere cartograficamente ambienti ben più complessi e strutturati dei corridoi ai quali ci siamo limitati – come, ad esempio, spazi aperti o stanze. A tale scopo, nuove e più adatte procedure di navigazione vanno studiate e realizzate, di pari passo con moduli che realizzino nuove abilità percettive. I principi di modularità a cui l'architettura che abbiamo proposto si ispira, consentono altresì di modificarne i componenti (possibilmente migliorandone le prestazioni !) senza che questo richieda un ripensamento radicale della sua struttura. Un miglior stimatore della posizione o

migliori percezioni, o riflessi, potranno essere via via inseriti una volta che diverranno disponibili.

Appendice A

Risultati teorici

In questa appendice riportiamo alcuni risultati sulla correttezza e la complessità di una versione semplificata dell'algoritmo 5.1, tratti da [60].

A.1 Ipotesi

L'ambiente del robot può essere modellizzato come un automa stati finiti, ovvero come un grafo diretto il cui insieme di nodi è Q e gli archi sono etichettati con elementi dell'insieme di azioni A (si faccia riferimento al capitolo 5 per maggiori dettagli). Ciascun nodo ha grado uscente $|A|$. Ogni nodo ha una unica firma, percepibile dall'agente e **l'insieme delle firme è noto in partenza**. Il rumore nella percezione è modellizzato probabilisticamente: la probabilità che l'agente osservi la firma propria del nodo i trovandosi nel nodo j è data da $\mathcal{F}[i, j]$ (ogni visita ad un nodo è un esperimento indipendente). **Non vi è incertezza nel movimento**.

A.2 Algoritmo di apprendimento

Sotto queste ipotesi il grafo di transizione può essere appreso con il seguente

Algoritmo A.1

1. per ogni azione $\alpha \in A$ costruisci una tabella bidimensionale T_α indicizzata in ciascuna dimensione da Q e inizializzata a 0;
2. per numero N di passi, ripeti:

- (a) scegli in modo casuale una azione $\alpha \in A$ da eseguire ed osserva la firma λ_j dello stato di arrivo;
- (b) se λ_i è la firma dello stato di partenza, incrementa di uno il valore di $T_\alpha(\lambda_i, \lambda_j)$;

3. costruisci un grafo (Q,E) tale che:

$$\forall \alpha, \forall \lambda_m : (q_m, \alpha, q_n) \in E \iff \lambda_n = \arg \max_{\lambda_j} T_\alpha(\lambda_m, \lambda_j) .$$

A.3 Risultati teorici

In questo paragrafo determineremo le specifiche condizioni sotto le quali l'algoritmo A.1 fornisce un risultato corretto, assieme ad una limitazione sul numero di osservazioni che l'agente deve fare al fine di avere una fissata confidenza nella corretta identificazione del grafo.

Indicheremo con $\alpha\lambda_i$ l'evento corrispondente all'osservazione di λ_i dopo l'esecuzione dell'azione α e con $P[\alpha\lambda_j \mid \lambda_i]$ la probabilità di osservare la firma λ_j dopo l'esecuzione di α , essendo λ_i la firma osservata precedentemente. L'algoritmo in sostanza costruisce una stima di queste probabilità.

Il seguente lemma afferma che è sufficiente visitare ciascun nodo un numero di volte polinomiale nella dimensione del grafo perché il risultato dell'algoritmo sia corretto, con una confidenza assegnata.

Definizione A.1 Per α, i, j e k fissati definiamo la separazione delle probabilità come $s_i^\alpha(j, k) = P[\alpha\lambda_j \mid \lambda_i] - P[\alpha\lambda_k \mid \lambda_i]$. Sia $s_i^\alpha = s_i^\alpha(j, k)$ dove λ_j è il più probabile successore di λ_i e λ_k è il secondo più probabile successore, in seguito all'azione α . Sia $s = \min_{i,\alpha} s_i^\alpha$.

Lemma A.1 Il risultato dell'algoritmo è corretto con probabilità almeno $1-\varepsilon$ dopo che ciascun nodo è stato visitato almeno N volte, con N polinomiale in $\frac{1}{s}$, $\frac{1}{\varepsilon}$, $|Q|$, $|A|$.

Dimostrazione. Calcoliamo quante volte è necessario visitare un singolo nodo per ottenere un arco correttamente. Fissiamo l'azione α ed il nodo λ_i . Siano λ_j e λ_k rispettivamente la prima e la seconda firma più probabili dopo l'osservazione di λ_i e

l'esecuzione di α . Consideriamo una variabile casuale Bernoulliana:

$$L_j = \begin{cases} 1 & \text{se } \lambda_j \text{ viene percepita dopo } \lambda_i \text{ come risultato dell'azione } \alpha \\ 0 & \text{altrimenti} \end{cases}$$

e analogamente si definisce L_k . Si consideri ora la variabile casuale (anch'essa Bernoulliana) $D = L_j - L_k$. È immediato notare che la sua media $\mu_D = \mathbf{E}[D]$ è proprio $s_i^\alpha = P[\alpha\lambda_j \mid \lambda_i] - P[\alpha\lambda_k \mid \lambda_i]$.

Una stima di μ_D è la statistica:

$$\hat{D} = \frac{T_\alpha(\lambda_i, \lambda_j) - T_\alpha(\lambda_i, \lambda_k)}{\sum_n T_\alpha(\lambda_i, \lambda_n)}$$

L'algoritmo sceglie correttamente l'arco non appena \hat{D} e μ_D hanno lo stesso segno (la firma risultante con più alta probabilità è anche quella con la più alta frequenza campionaria). Una condizione sufficiente perché abbiano lo stesso segno è

$$|\mu_D - D| < |\mu_D| ;$$

così la probabilità di errore, β , può essere maggiorata con

$$\beta \leq P[|\mu_D - D| \geq |\mu_D|] .$$

Dato che ogni visita è un evento indipendente, possiamo applicare la disuguaglianza di Hoeffding per maggiorare l'errore dopo N visite con

$$\beta \leq P[|\mu_D - D| \geq |\mu_D|] \leq e^{-2N\mu_D^2} \leq e^{-2Ns^2} .$$

Risolviendo rispetto ad N , otteniamo

$$N \leq \frac{-\ln \beta}{2s^2} .$$

Per garantire che l'intero grafo sia corretto con probabilità $1 - \varepsilon$ si deve scegliere $\beta < 1 - (1 - \varepsilon)^{1/(|Q||A|)}$, ottenendo così la maggiorazione finale su N , il numero di visite per azione per firma di partenza, di

$$N \leq \frac{-1}{2s^2} \ln(1 - (1 - \varepsilon)^{1/(|Q||A|)}) ,$$

che è polinomiale in $\frac{1}{s}$, $\frac{1}{\varepsilon}$, $|Q|$, e $|A|$. \square

Il seguente lemma (dovuto a Basye [8]) ci consente di determinare la lunghezza di una passeggiata casuale che garantisca che ogni nodo venga visitato un numero minimo di volte. Per ottenere una lunghezza polinomiale è necessario che il grafo sia tale che per ogni arco (q_i, α, q_j) ne esiste un'altro (q_j, α^{-1}, q_i) ¹ ovvero che in sostanza il grafo sia non diretto. In effetti, in un grafo diretto, un tale cammino ha lunghezza esponenziale nella dimensione del grafo.

Lemma A.2 *Per ogni grafo connesso non diretto G , per ogni $\delta > 0$ e ogni $N \geq 1$, esiste un polinomio $p(|Q|, |A|, N, \frac{1}{\delta})$ tale che con probabilità $1 - \delta$, una passeggiata casuale su G di lunghezza p visita ogni nodo di Q almeno N volte.*

Dimostrazione. Ci si basa su un risultato precedente il quale stabilisce che un cammino di lunghezza $2|A||Q|(|Q| - 1)$ attraversa ogni arco di G con probabilità maggiore di $\frac{1}{2}$, e dunque un tale cammino visita tutti i nodi con probabilità maggiore di $\frac{1}{2}$. Sia k una tale lunghezza e consideriamo ogni cammino di lunghezza k come un singolo esperimento indipendente, la probabilità che ogni nodo sia visitato in r di questi esperimenti è maggiore di $1 - (\frac{1}{2})^r$. Se ora consideriamo ciascuno di questi cammini di lunghezza kr come un esperimento indipendente, la probabilità che in N di questi esperimenti ciascun nodo venga visitato N o più volte è almeno $(1 - (\frac{1}{2})^r)^m$. Questo valore deve essere più grande di $1 - \delta$. Elaborando e risolvendo rispetto ad r , si vede che $r = \log_2(\frac{N}{\delta})$ è sufficiente, dunque l'intero cammino risulta lungo $rkN = 2N|A||Q|(|Q| - 1) \log_2(\frac{N}{\delta})$. \square

Il lemma seguente ci consente di caratterizzare la quantità s in termini di un minorante della probabilità di corretta osservazione $P \leq \min_i \mathcal{F}[i, i]$, il cui significato è più chiaro dal punto di vista sperimentale ed è immediatamente connesso alla modellizzazione della percezione delle firme che abbiamo adottato.

Lemma A.3 *Sia z il rapporto tra la più alta e la più bassa frequenza relativa di visita dei nodi in una passeggiata casuale di una certa lunghezza. Se le probabilità di osservazione scorretta sono riflesive, ovvero $\mathcal{F}[i, j] = \mathcal{F}[j, i]$ allora $\forall \alpha, i$*

1. $s_i > 0$ se $P > \frac{1}{4}(1 - z + \sqrt{z^2 + 6Z + 1})$, e

2. $s_i > \frac{1}{z}P(2P - 1) + P - 1$.

¹Si noti che non si richiede che l'agente sia in grado di reverse le proprie azioni (altrimenti l'apprendimento è banale).

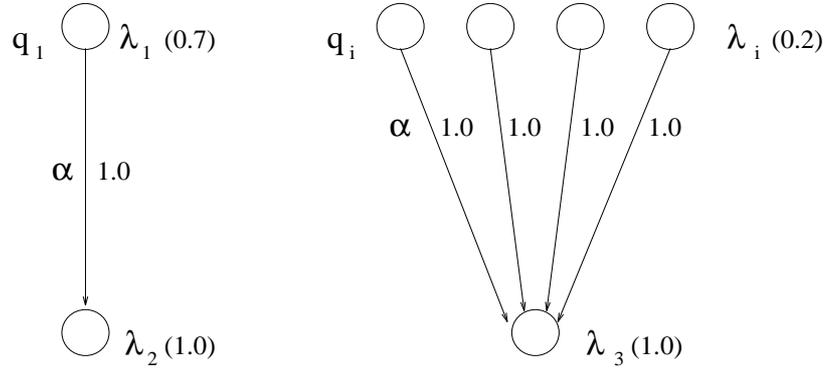


Figura 24: frammento di un automa patologico. Per mostrare la necessità dell'ipotesi $\mathcal{F}[i, j] = \mathcal{F}[j, i]$ supponiamo che $\mathcal{F}[1, 1] = 0.7$, $P[\alpha\lambda_2 | \lambda_1] = 1.0$ e che vi siano n stati q_i per ciascuno dei quali vale $\mathcal{F}[i, 1] = 0.2$, $P[\alpha\lambda_3 | \lambda_i] = 1.0$. È verosimile che per $n > 3$ il robot osservi più frequentemente λ_1 seguito da λ_3 di quanto non osservi λ_1 seguito da λ_2 e che quindi ricostruisca in modo errato la struttura dell'automata.

La dimostrazione di questo lemma è lunga e tediosa ed il risultato è poco generale, preferiamo dunque ometterla e passare direttamente al

Teorema A.1 (Basye et al.) *Il risultato dell'algoritmo A.1 è corretto con probabilità almeno $1 - \varepsilon$ dopo una passeggiata casuale di lunghezza polinomiale in $\frac{1}{z}P(2P - 1) + P - 1, \frac{1}{\varepsilon}, |Q|, |A|$ non appena:*

1. $P > \frac{1}{4}(1 - z + \sqrt{z^2 + 6Z + 1})$,
2. $\forall i, j : \mathcal{F}[i, j] = \mathcal{F}[j, i]$
3. *il grafo sia effettivamente non diretto.*

Dimostrazione. Il teorema segue immediatamente dai lemmi precedenti. \square

In effetti sotto le condizioni richieste da lemma 2, le frequenze di visita stazionarie sono uguali per ciascun nodo, dunque $z = 1$. Questa osservazione riduce il primo requisito del teorema a $P > \frac{\sqrt{2}}{2}$ e rende la lunghezza della passeggiata polinomiale in $1/(P - \frac{\sqrt{2}}{2})$.

Appendice B

Modello del sensore ultrasonico

I telemetri ultrasonici o semplicemente SONAR sono ampiamente usati in robotica per la loro semplicità e per il basso *data rate* che consente di ottenere rapidamente e a basso costo (computazionale ed economico) informazioni sensoriali che, per quanto rozze, possono risultare utili in molte applicazioni. Il più diffuso è il *Polaroid ultrasonic ranging system*, basato sul tempo di volo. Durante il funzionamento viene trasmesso un impulso sonoro verso un bersaglio e l'eco risultante viene rilevata. Il tempo trascorso tra la trasmissione e la ricezione, o tempo di volo (abbreviato TOF da *Time Of Flight*), può quindi essere convertito in una distanza conoscendo la velocità di propagazione del suono. La primitiva utilizzata per descrivere l'informazione fornita da un sensore sonar è il punto TOF (*TOF dot*) che è semplicemente un punto individuato sull'asse del sensore alla distanza ottenuta dal tempo di volo; più formalmente è una coppia ordinata formata dal valore di profondità r e dall'orientazione del sensore α . La figura 25 mostra la rappresentazione cartesiana di un insieme di punti TOF presi con una scansione semicircolare in un corridoio. Discuteremo nel seguito dell'interpretazione di questi dati, dopo avere descritto il sistema di acquisizione ed il principio fisico di funzionamento.

B.1 La scheda di misura

Il telemetro ultrasonico utilizzato nei nostri esperimenti consiste di un trasduttore elettrostatico Polaroid [32] e di una scheda elettronica SN28827 della Texas Instruments, ed è capace di misurare la distanza di oggetti in una gamma che va da $0.2m$

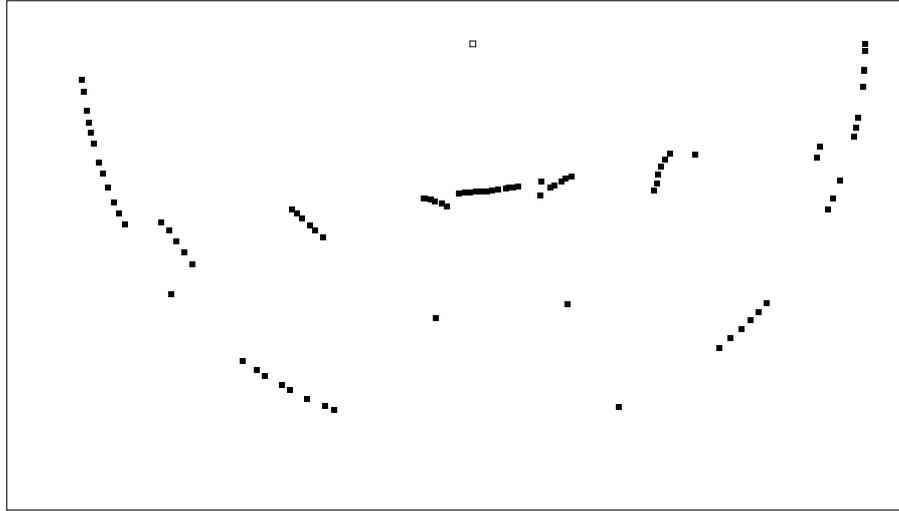


Figura 25: scansione semicircolare in un corridoio. Il quadratino vuoto indica la posizione del robot. Il corridoio è orientato in senso orizzontale.

a $10m$. La figura 26 mostra un diagramma semplificato del sistema. Quando l'unità viene attivata (con l'asserzione del segnale INIT), l'oscillatore invia un treno di 16 impulsi sinusoidali alla frequenza di $50Khz$ al trasduttore, il quale emette un impulso sonoro. Nello stesso istante viene inibito l'amplificatore di ricezione per un tempo pari a $2.38ms$ (da cui dipende la distanza minima misurabile), dopo di che il trasduttore funziona da ricevitore e attende di ricevere l'eco di ritorno dal bersaglio. Alla ricezione dell'eco, il trasduttore converte l'energia da sonora ad elettrica ed il segnale viene applicato ad un amplificatore a guadagno variabile nel tempo (o TVG) per compensare l'attenuazione del suono nell'aria. Il segnale in uscita dell'amplificatore di ricezione viene integrato e rivelato da un circuito a soglia. La misura del tempo di volo è ottenuta dalla differenza tra l'istante in cui la soglia viene superata (per la prima volta) dal segnale di ritorno (asserzione di ECHO) e l'istante di inizio della trasmissione del treno di impulsi (asserzione di INIT).

La scheda utilizzata permette di effettuare misure di distanza utilizzando i due soli segnali di INIT e ECHO. Esistono altri due segnali di controllo, BLNK e BINH utilizzabili rispettivamente per annullare la situazione di echo ricevuta, consentendo la rilevazione degli echi successivi al primo, e per modificare il tempo di inibizione

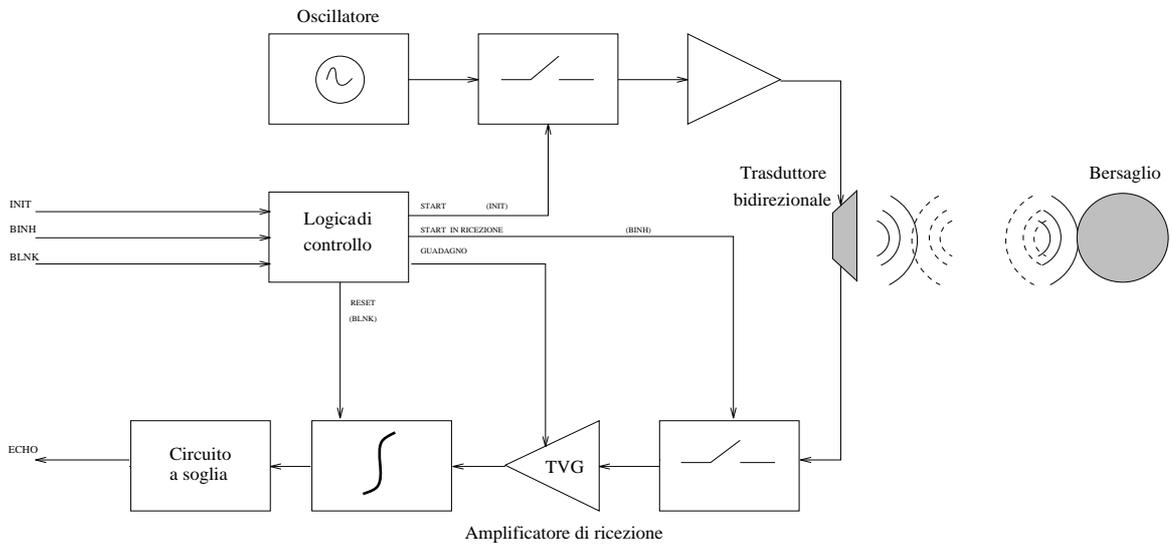


Figura 26: sistema di telemetria ultrasonica. La distanza dal bersaglio viene calcolata misurando il tempo di volo dell'onda acustica emessa dal trasduttore e riflessa indietro dal bersaglio

successivo alla trasmissione, ottenendo una diversa distanza minima misurabile.

B.2 Modello fisico del trasduttore

Il componente principale dell'apparecchio è il trasduttore ultrasonico, che funziona sia da altoparlante (sorgente) che da microfono (rilevatore). Quando il raggio a ($= 19mm$) dell'apertura del trasmettitore è molto più grande della lunghezza d'onda acustica $\lambda = c/f$ ($= 6.84mm$), l'emissione forma un fascio in cui è concentrata l'energia. Modellizzando il trasmettitore come un pistone piatto di raggio a chiuso in una membrana infinita, si ottiene un pattern di emissione che ha due regioni distinte: la zona vicina (zona di *Fresnel*) e la zona lontana. Nella zona vicina, che si estende di fronte al traduttore per una profondità di circa a^2/λ il fascio è contenuto in un cilindro di raggio a . Nella zona lontana il fascio diverge con un angolo che dipende da a e λ . Essendo interessati alla modellizzazione del sensore per l'analisi ambientale, consideriamo le caratteristiche nella zona lontana, poiché gran parte dei bersagli sono in tale regione. Quando la vibrazione acustica avviene ad una singola frequenza, il pattern di radiazione – che descrive la distribuzione radiale della pressione acustica –

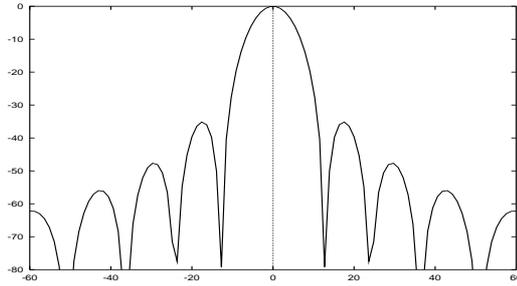


Figura 27: pattern di emissione teorico monocromatico. In ascisse è riportato l'azimut in gradi, in ordinate la potenza acustica (i valori sono in dB, normalizzati rispetto al valore sull'asse)

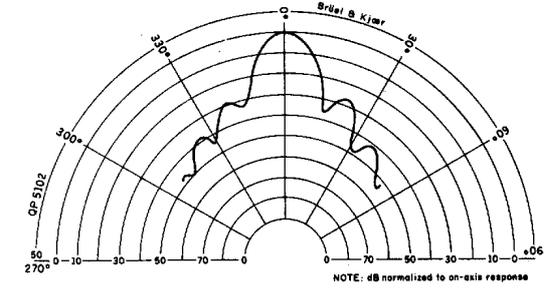


Figura 28: pattern di emissione sperimentale. Si noti che è in accordo con quello teorico solo nei primi due lobi

nella zona lontana è dato da:

$$p(\theta) = 2 \frac{J_1(ka \sin(\theta))}{(ka \sin(\theta))},$$

dove $k = 2\pi/\lambda$, J_1 è la funzione di Bessel del primo tipo, θ è l' azimuth rispetto all'asse del sensore. Una misura della semi-ampiezza radiale del fascio nella zona lontana è fornita dal più piccolo angolo dove l'emissione si annulla, dato da:

$$\theta_0 = \sin^{-1} \frac{0.61\lambda}{a}.$$

Nel nostro caso $\theta_0 = 12.7^\circ$. La figura B.2 mostra la potenza acustica in funzione dell'azimuth, nel caso monocromatico. Una prima approssimazione del sonar, che tuttavia produce buoni risultati [5] consiste nel considerare il solo lobo principale di emissione, ovvero trascurare l'emissione al di fuori del settore $\pm\theta_0$, il quale viene dunque indicato come cono di emissione.

Nel caso del telemetro ultrasonico che stiamo esaminando, l'emissione acustica è un pacchetto d'onde (come abbiamo visto nella sezione precedente), dunque sono presenti una gamma di frequenze, ciascuna delle quali genera il proprio pattern. La somma pesata di queste componenti (che sono funzioni di Bessel) è approssimata, nel lobo principale, da una Gaussiana della forma:

$$p_T(\theta) = e^{-2\theta^2/\theta_0^2}.$$

Poiché lo stesso trasduttore che trasmette funziona anche da ricevitore, il principio di reciprocità afferma che il pattern di sensibilità del ricevitore è identico al pattern

di emissione:

$$p_R(\theta) = p_T(\theta).$$

Il pattern totale del trasduttore bidirezionale è dato dal prodotto:

$$p_{T/R}(\theta) = p_T(\theta)p_R(\theta) = e^{-4\theta^2/\theta_0^2}.$$

Dal pattern totale vogliamo determinare l'ampiezza dell'eco riflesso da un bersaglio. A causa della dispersione radiale del fascio nella zona lontana, l'ampiezza dell'onda di propagazione decresce con la distanza come $1/r$, come dettato dalla conservazione dell'energia. Se A_0 è l'ampiezza osservata lungo l'asse ad una distanza unitaria, l'ampiezza dell'eco riflessa da un bersaglio distante r con azimuth θ è data da:

$$A(\theta, r) = \frac{A_0}{r} e^{-4\theta^2/\theta_0^2}.$$

Per il corretto funzionamento del telemetro è necessario che l'amplificatore di ricezione a guadagno variabile (TVG) sia correttamente tarato, in modo che l'ampiezza del segnale in uscita non dipenda dalla distanza.

B.3 Interpretazione dei dati sonar

Come evidenziato da Durrant-White, spesso in letteratura il sonar viene considerato come una cattiva approssimazione di un sensore ideale (*ray-trace scanner*) il quale ha la proprietà di emettere un fascio sottile e non divergente, passibile di riflessione diffusa. Con questo sensore è possibile misurare la distanza da qualunque bersaglio si trovi sull'asse del sensore indipendentemente dalla sua orientazione e dalla natura della superficie. In realtà un telemetro sonar possiede delle caratteristiche marcatamente diverse; in particolare differisce dal *ray-trace scanner* in due aspetti: apertura radiale del fascio (*beam width*) e riflessione speculare. Infatti come discusso nella precedente sezione, il sonar emette un fascio con una apertura angolare non nulla (25.4°), e la lunghezza d'onda ($6.95mm$) è tale per cui molti bersagli danno origine a riflessione speculare¹. Questo dà luogo ad una serie di problemi:

¹Come noto, un'onda incidente su una superficie è passibile di due tipi di riflessione: diffusa e speculare. La prima ha luogo quando la superficie è liscia rispetto alla lunghezza d'onda dell'onda incidente, la riflessione diffusa, invece, ha luogo quando la superficie riflettente è ruvida rispetto alla lunghezza d'onda del raggio illuminante.

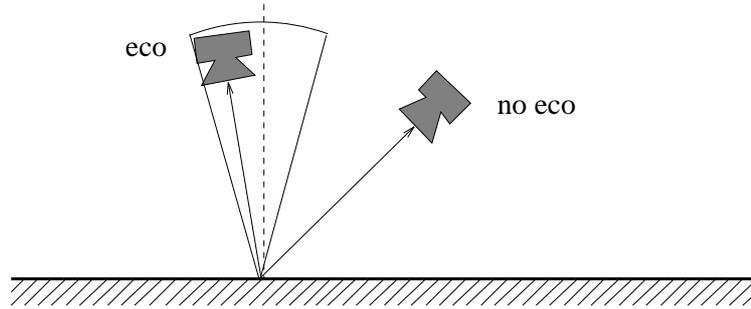


Figura 29: illustrazione dell'angolo di visibilità di un bersaglio. Il trasduttore riceve l'eco di ritorno solo se l'angolo tra il suo asse e la normale alla superficie riflettente è compreso in un determinato intervallo, corrispondente in prima approssimazione con l'estensione angolare del lobo principale di emissione.

- incertezza nella localizzazione del bersaglio. A causa della apertura del fascio l'eco ricevuta può provenire da un qualsiasi punto all'interno del cono di emissione;
- a causa della riflessione speculare è possibile ricevere l'eco solo da superfici ortogonali alla direzione di propagazione. Poiché il fascio emesso possiede una apertura angolare non nulla, si può ricevere l'eco da superfici illuminate con angoli di incidenza diversi da 0, diciamo compresi in $\pm\beta/2$. Questo è il semiangolo di visibilità della superficie, e in prima approssimazione si considera uguale a θ_0 (figura 29);
- un segnale spurio può arrivare al ricevitore a causa di riflessioni speculari multiple e come effetto viene sovrastimata la distanza del bersaglio dal sensore.

È possibile tuttavia interpretare correttamente i dati sonar mediante una corretta modellizzazione del sensore come indicato da Kuc [66] [64] [16] e successivamente da Leonard e Durrant-White [74]. La figura 30 rappresenta la profondità (*range*) in funzione dell'orientazione del sensore per la stessa scansione di figura 25. Vista in questo modo è evidente che le primitive predominanti sono segmenti orizzontali – insiemi di punti TOF adiacenti aventi quasi la stessa profondità. Queste sequenze prendono il nome di *Regions of Constant Depth* e corrispondono ad un arco in coordinate cartesiane. Nella figura 30 si possono riconoscere le RCD dovute alla risposta del muro laterale del corridoio. Quella ottenuta nel lobo principale, si presenta come un pianoro di profondità $1.4m$ – che ben riproduce la distanza del sensore dal muro – ed

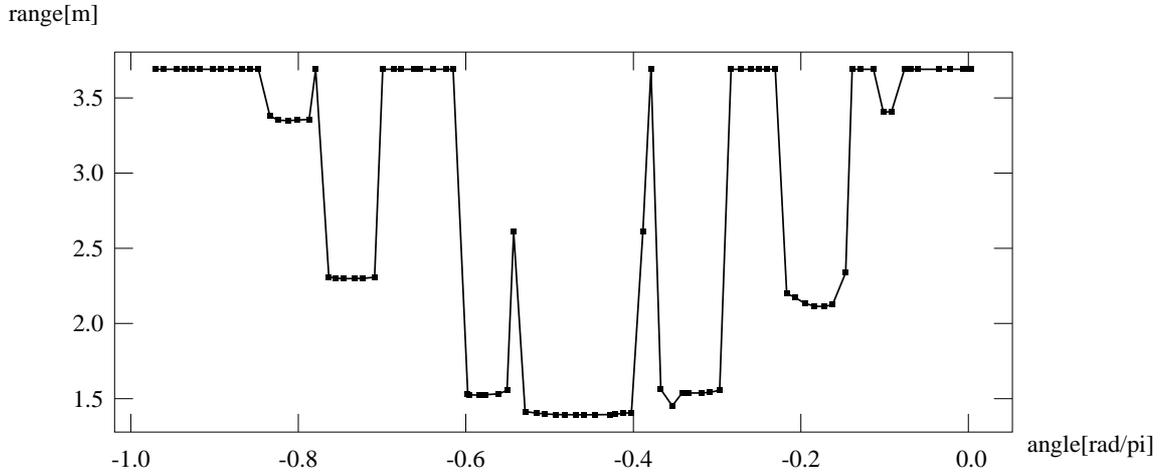


Figura 30: grafico della profondità misurata in funzione dell'orientazione del sonar. Si notino le regioni di profondità costante, corrispondenti alle risposte di ciascuno dei lobi di emissione. Solo la risposta al lobo principale, però, fornisce la distanza corretta.

ha una estensione angolare di 24.5° . I punti TOF di profondità $3.7m$ (corrispondente alla massima profondità misurabile) derivano da mancata ricezione dell'eco, ed a rigore non possono costituire delle RCD. Le altre RCD – grossomodo simmetriche rispetto alla prima – sono dovute alla emissione nei lobi laterali del fascio, sono meno estese in angolo e recano una informazione di profondità tanto più inesatta quanto più bassa è l'energia sonora emessa nella direzione perpendicolare alla superficie. Questo fenomeno è spiegabile ricordando il funzionamento del circuito di ricezione. Quando il segnale di ritorno è piccolo rispetto a quello previsto, il condensatore dell'integratore si carica lentamente, con il risultato che la soglia di rilevazione viene superata con un certo ritardo rispetto all'istante di ricezione del primo impulso del pacchetto. Le RCD ottenute dal lobo principale sono le sole che danno informazione utile e possono essere distinte dalle altre (anche da quelle dovute a riflessioni multiple) grazie alla maggiore estensione angolare. Un algoritmo per l'estrazione e l'interpretazione delle RCD è presentato in [74].

Appendice C

Note implementative e codice

Non riporteremo in questa appendice tutto il codice che è stato scritto, in quanto non ci sembra di estremo interesse e inoltre non ha un grado di assestamento tale da essere riportato qui. Parte del codice infatti è una simulazione del robot reale e parte invece è costruita per girare a bordo del veicolo ed interfacciarsi con Action Manager (attualmente in fase di test). La parte più indipendente dal mondo fisico e più leggibile (per quanto il C possa essere leggibile) e l'implementazione dell'algoritmo 5.1 per la ricostruzione dell'automa.

La comunicazione con il navigator avviene tramite due pipe: una per le azioni e una per le firme. Automata prende come argomenti rispettivamente:

1. il file descriptor della pipe di ingresso;
2. il file descriptor della pipe di uscita;
3. il numero di passi da eseguire;
4. il PID del navigator

Quest'ultimo parametro è necessario solo nella simulazione affinché automata possa far terminare navigator. A regime verranno utilizzati i sockets per la comunicazione tra i processi; questo consentirà ai moduli di girare su macchine differenti (infatti automata girerà su una WS Sun mentre navigator su di un PC a bordo del robot, sotto Linux o.s.)

L'output del programma avviene su di un file '`graph_world.out`', in cui è rappresentato il grafo ottenuto dall'esplorazione nella forma richiesta da Mathematica

[101]. Consiste di n linee, corrispondenti ai nodi del grafo, la linea i -esima ha la forma $i \ x \ y \ v_1 \ v_2 \ \dots \ v_k$, dove il nodo i ha coordinate (x,y) nel piano ed è adiacente ai vertici $v_1 \ \dots \ v_k$. Per esempio ecco un frammento della descrizione del grafo dell'IRST:

```

1    0.000000  0.000000  2  1  1  18
2    0.000000  6.000000  3  19 1  2
3    0.000000 16.000000  4  20 2  3
4    0.000000 32.000000  5  6  3  4
5    0.000000 38.000000  6  5  4  11
6    0.000000 46.000000  7  12 5  6

```

Durante l'esecuzione sullo standard output viene visualizzata la situazione corrente, per esempio:

```

TAKEN ACTION : V
CURRENT LABEL:
TAG: 1
SHAPE: NE
POSITION: 0.000000 0.000000 0.500000

```

```

Updating table #W entry: 1 1
Updating table #S entry: 1 1
TAKEN ACTION : N
CURRENT LABEL:
TAG: 2
SHAPE: NWS
POSITION: 0.000000 6.000000 0.500000
Updating table #N entry: 1 2
Updating table #S entry: 2 1

```

```

Updating table #E entry: 2 2
TAKEN ACTION : N
CURRENT LABEL:

```

```
TAG: 3
SHAPE: NWS
POSITION: 0.000000 16.000000 0.500000
Updating table #N entry: 2 3
Updating table #S entry: 3 2

Updating table #E entry: 3 3
TAKEN ACTION : N
CURRENT LABEL:
TAG: 4
SHAPE: NWS
POSITION: 0.000000 32.000000 0.500000
Updating table #N entry: 3 4
Updating table #S entry: 4 3
```

Riportiamo di seguito il file `automata.c`, che implementa l'algoritmo: 5.1

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/uio.h>
#include <string.h>
#include <signal.h>

#include "matrix.h"
#include "automata.h"
#include "automataP.h"

int main(int argc, char **argv)
{
    static int
        in, /*____FD of input  pipe____*/
        out; /*____FD of output pipe____*/
    static u_int found;
```

```

FILE *graph, *st;
static int
    size, /*__current table size __*/
    maxtag,
    k, /*__index ranging from 1 to ACTIONS__*/
    i, j, min, max, sum, step, maxstep;

TableType Tables;
ActionType action, inverse_action;
TagType prev_signature_tag, signaturemax;
SignatureListType Signatures_List;
NodePtrType signature, lm;
static int visits[DIM+1]; /*_visit frequencies_*/
static int child; /*_navigator PID_____*/

/*__initialization__*/

in = atoi(argv[1]);
out = atoi(argv[2]);
child = atoi(argv[3]);
maxstep = atoi(argv[4]);
st=fopen("visits.xg", "w");

/*__global look-up array__*/

a2a[N]='N';
a2a[W]='W';
a2a[S]='S';
a2a[E]='E';

/*_initialize the list to the signature of the starting location_*/

Signatures_List.head = NewSignature();

```

```

Signatures_List.tail = Signatures_List.head;
signature = Signatures_List.tail;

GetSignature(in,&(signature->info));

signature->tag = 1;
signature->next = NIL;
++(visits[signature->tag]);
maxtag=1;

/*_____for each action construct a 2-d table _____*/
/*   indexed in each dimension by the signatures in the list   */

size=1;
for(k=1; k<=ACTIONS; ++k)
    for (i=1; i<=size; ++i)
        for (j=1; j<=size; ++j)
Tables[k][i][j]=0;

printf("\n\n");
printf("TAKEN ACTION :\tV\n");
printf("CURRENT SIGNATURE:\n");
printf("\tTAG:\t%d\n",signature->tag);
printf("\tSHAPE:\t%s\n",signature->info.shape);
printf("\tPOSITION: %f %f %f\n",
signature->info.position[0],
signature->info.position[1],
signature->info.position[2]);
printf("\n\n");
fflush(stdout);

step=0;
while (step<maxstep)

```

```

    {
step++;

/*___for each action wich is perceived to leave the robot___*/
/*  in the same location increment the entry indexed by the */
/*  current signature in the corresponding table.           */
/*  Choose the least taken action among the feasible ones. */

        action=1;
        min=1000;
        for (k=1; k<=ACTIONS; k++)
            if (!occluded_direction(signature->info.shape, k))
            {
                sum=0;
                for(i=1; i<=size; ++i)
sum = sum + Tables[k][(signature->tag)][i];
                if (sum<min)
            {
                min=sum;
                action=k;
            }
        }
    else
    {
        (Tables[k][signature->tag][signature->tag])++;
        printf("Updating table #%c entry: %d %d\n",
a2a[k],signature->tag,
signature->tag);
        fflush(stdout);
    }

/*_____execute least taken action_____*/

```

```

        if (ExecuteAction(out,action) == NULL)
    {
        fprintf(stderr,"%s: can't execute action",
        argv[0]);
        exit(1) ;
    }
/*-----get the current signature-----*/

        prev_signature_tag=signature->tag;
        signature=NewSignature();
        if (GetSignature(in,&(signature->info))== NULL)
    {
        fprintf(stderr,"can't get signature");
        exit(1) ;
    }
signature->next = NIL;
/*-----matching in the list of signatures-----*/

signature->tag = 0;
lm=Signatures_List.head;
        found=0;
while ((lm != NIL) && (!found))
    {
        if (similar(&(lm->info),&(signature->info)))
    {
        found=1;
        signature->tag = lm->tag ;
    }
        lm=lm->next;
    }
        if (signature->tag == 0)
    {
/*-----push current signature in the list -----*/

```

```

        signature->tag = (Signatures_List.tail->tag) + 1;
        maxtag++;
    Signatures_List.tail->next = signature;
    Signatures_List.tail  = signature;

/*_____extension of tables_____*/

    size++;
        for (k=1; k<=ACTIONS; ++k)
            for (i=1; i<=size; ++ i)
    {
        Tables[k][size][i]=0;
        Tables[k][i][size]=0;
    }
}

/*_____a valid tag is assigned to the current signature_____*/

++(visits[signature->tag]);
if(!(step%10))fprintf(st,"%d %g \n",
    (int)(step/10),
    entropy(visits,maxtag));

/*_____increment by 1 ...just taken action and its inverse_____*/

    inverse_action = (action + 1)%4 +1 ;
    (Tables[action][prev_signature_tag][signature->tag])++;
    (Tables[inverse_action][signature->tag][prev_signature_tag])++;

/*_____show current signature _____*/

printf("TAKEN ACTION : \t%c\n",a2a[action]);
printf("CURRENT SIGNATURE:\n");

```

```

printf("\tTAG:\t%d\n",signature->tag);
printf("\tSHAPE:\t%s\n",signature->info.shape);
printf("\tPOSITION: %f %f %f\n",
      signature->info.position[0],
      signature->info.position[1],
      signature->info.position[2]);
printf("Updating table #%c entry: %d %d\n",
      a2a[action],
      prev_signature_tag,
      signature->tag);
printf("Updating table #%c entry: %d %d\n",
      a2a[inverse_action],
      signature->tag,
      prev_signature_tag);
printf("\n\n");
fflush(stdout);

      } /*end while*/

/*-----build the map -----*/

printf("\n      *****OUTPUT*****\n");
for(k=1; k<=ACTIONS; ++k)
  {
printf("\nAction %c \n",a2a[k]);
for (i=1; i<=size; ++i)
  {
    for (j=1; j<=size; ++j)
      printf("%d\t",Tables[k][i][j]);
printf("\n");
  }
  }
}

```

```

    fflush(stdout);
    graph=fopen("graph_world.out","w");
    lm=Signatures_List.head;
    fprintf(graph,"%d\n",maxtag);

    while (lm!=NIL)
        {
    fprintf(graph,"%d\t%s\t%lf %lf\t",
    lm->tag,
    lm->info.shape,
    lm->info.position[0],
    lm->info.position[1]);
    for(k=1; k<=ACTIONS; ++k)
        {
            max=-1;
            for(j=1; j<=size; ++j)
                if (Tables[k][lm->tag][j]>max)
            {
                max=Tables[k][lm->tag][j];
                signaturemax=j;
            }
            fprintf(graph,"%d\t",signaturemax);
        }
    fprintf(graph,"\n");
    lm=lm->next;
        }

    fclose(graph);
    fclose(st);
    kill(child,SIGKILL);    /*__kill navigator befor exit__*/

} /*-----End of Main-----*/

```

```

NodePtrType NewSignature()
{
    return ((NodePtrType) malloc(sizeof(NodeType)));
}

/*-----interface with NAVIGATOR-----*/

int GetSignature(int in, SignatureType *signature)
{
    return(read(in,(char *)signature,sizeof(SignatureType)));
}

int ExecuteAction(int out, ActionType k)
{
    return(write (out,&(a2a[k]),sizeof(char)));
}

#define SIMTOLL 4.61

int similar (SignatureType *lm, SignatureType *lo)
{
    double mhn_norm(VEC *z, MAT *P2);
    VEC *x;
    MAT *A;

    double mhd;
    int i,j;

    A=get_mat(2,2);
    x=get_vec(2);
    for(i=0;i<2; i++)
        x->ve[i]=lo->position[i] - lm->position[i];
    for(i=0;i<2; i++)

```

```
        for(j=0;j<2; j++)
            A->me[i][j]= lo->confidence[i][j]+lm->confidence[i][j];
mhd=mhn_norm(x,A);
return (mhd < SIMTOLL);
}
```

```
double mhn_norm(VEC *z, MAT *P2)
{
    VEC *t;
    MAT *T;

    T= get_mat(2,2);
    m_inverse(P2,T);
    t=mv_mlt(T,z,VNULL);
    return (v_sum(v_star(t,z,VNULL)));
}
```

```
static double entropy(int v[],int n)
{
    int sum,i;
    double e,a;

    sum=0;
    for(i=1; i<=n; ++i)
        sum=sum+v[i];
    e=0;
    for(i=1; i<=n; ++i)
        {
            a=(double)v[i]/(double)sum;
            if (a>0)
                e=e+a*log(a);
        }
    return (e);
}
```

}

Bibliografia

- [1] Anderson, Madsen, Sorensen, Kirkeby, Jones, and Christensen. Navigation using range images on a mobile robot. *Robotics and Autonomous Systems*, 10:147–160, 1992.
- [2] T. L. Anderson and M. Donath. Animal behaviour as a paradigm for developing robot autonomy. *Robotics and Autonomous Systems*, 6:145–168, 1990.
- [3] R. C. Arkin. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.
- [4] N. Ayache and O. D. Faugeras. Maintaining representations of environment of a mobile robot. Technical Report 789, Unité de recherche INRIA-Roquencourt, Fevrier 1988.
- [5] B. Barshan and R. Kuc. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):560–569, june 1990.
- [6] K. Basye. Graph-based mapping by mobile robots. In Wolfe and Chun, editors, *Mobile Robot VII*, pages 643–649. The International Society for Optical Engineering, November 1992.
- [7] K. Basye and T. Dean. Map learning with indistinguishable locations. In Lemmer and Kanal, editors, *Uncertainty in Artificial Intelligence 5*, pages 331–341. Elsevier, 1990.
- [8] K. Basye, T. Dean, and J. S. Vitter. Coping with uncertainty in map learning. Technical Report CS-89-27, Brown University Department of Computer Science, 1989.

- [9] K. Basye, T. Dean, and J. S. Vitter. Coping with uncertainty in map learning. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 663–668. IJCAI, 1989.
- [10] M. Beckerman and E. M. Obrow. Treatment of systematic errors in the processing of wide-angle sonar sensor data for robotic navigation. *IEEE Transactions on Robotics and Automation*, 6(2):137–145, april 1990.
- [11] M. Betke, R. L. Rivest, and M. Singh. Piecemeal learning of an unknown environment. apparirà in COLT, july 1993.
- [12] J. Borenstein and Y. Koren. Obstacle avoidance with ultrasonic sensors. *IEEE Journal of Robotics and Automation*, 4(2):213–218, april 1988.
- [13] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):1179–1187, october 1989.
- [14] J. Borenstein and Y. Koren. Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 7(4):535–539, august 1991.
- [15] J. Borenstein and Y. Koren. The vector field histogram - fast obstacle avoidance for mobile robot. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, june 1991.
- [16] O. Bozma and R. Kuc. Building a sonar map in a specular environment using a single mobile sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(12):1260–1269, december 1991.
- [17] R. A. Brooks. Solving the find-path problem by good representation of free space. In *Proceedings of the National Conference on Artificial Intelligence*, pages 381–386. AAAI, 1982.
- [18] R. A. Brooks. Visual map making for a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 824–829. IEEE, 1985.

- [19] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, march 1986.
- [20] R. A. Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15, 1990.
- [21] M. K. Brown. Feature extraction techniques for recognizing solid objects with ultrasonic range sensor. *IEEE Journal of Robotics and Automation*, 1(4):191–205, december 1985.
- [22] M. K. Brown. Locating objects surfaces whith an ultrasonic range sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 110–115. IEEE, 1985.
- [23] J. Canny and B. Donald. Simplified voronoi diagrams. *Computational Geometry*, 3(3):219–236, 1988. in Autonomous Robot Vehicles.
- [24] R. Cattoni, T. Coianiz, and B. Caprile. Planning and Reactivity for the Mobile Robot of MAIA. In *Proceedings of the 6th International Conference on Advanced Robotics*, Tokyo, Japan, November 1993.
- [25] R. Cattoni, G. Di Caro, M. Aste, and B. Caprile. Bridging the gap between Planning and Reactivity: a Layered Architecture for Autonomous Indoor Navigation. Technical Report 9403-01, Istituto per la Ricerca Scientifica e Tecnologica, February 1994. Submitted to IROS '94 Conference.
- [26] D. Chapman. Planning for Conjunctive Goals. *Artificial Intelligence*, 32:333–377, 1987.
- [27] J. W. Chapman. Moment, variances, and covariances of sines and cosines of arguments which are subject to random error. *Technometrics*, 12(3):693–694, August 1970.
- [28] R. Chatila and J. P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 138–145. IEEE, 1985.

- [29] P. Cheeseman. In defense of probability. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1002–1009. IJCAI, 1985.
- [30] T. Coianiz. Controllo di un robot mobile in ambienti strutturati mediante l'utilizzo di sensori ad ultrasuoni. Internal Report 9210-12, Istituto per la Ricerca Scientifica e Tecnologica, October 1992.
- [31] T. Coianiz and M. Aste. Improving Robot's Indoor Navigation Capabilities by Integrating Visual, Sonar and Odometric Measurements. In Paul S. Schenker, editor, *Sensor Fusion VI*, volume 2059, pages 225–235. SPIE - The International Society for Optical Engineering, September 1993.
- [32] Polaroid Corporation. Ultrasonic Range Finders, 1982.
- [33] I. J. Cox and J. B. Kruskal. On the congruence of noisy images to line segment model. In *International Conference on Computer Vision*, pages 252–258. IEEE, December 1988.
- [34] I. J. Cox and G. T. Wilfong. *Autonomous Robot Vehicles*. Springer-Verlag, 1990.
- [35] J. L. Crowley. Dynamic world modeling for an intelligent mobile robot using a rotating ultra-sonic ranging device. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 128–135. IEEE, 1985.
- [36] J. L. Crowley. Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, 1(1):31–41, march 1985.
- [37] J. L. Crowley and F. Ramparany. Mathematical tools for representing uncertainty in perception. In A. Kak and S. Chen, editors, *Proceedings of the Workshop on Spatial Reasoning and Multi-sensor Fusion*, pages 293–302. Morgan Kaufmann, 1987.
- [38] J. L. Crowley, P. Stelmaszyk, and C. Discours. Measuring image flow by tracking edge-lines. In *International Conference on Computer Vision*, pages 658–664. IEEE, December 1988.

- [39] T. Dean, D. Angluin, K. Basye, S. Engelson, L. Kaelbling, E. Kokkevis, and O. Maron. Inferring finite automata with stochastic output functions and application to map learning. Technical Report CS-92-27, Brown University Department of Computer Science, 1992.
- [40] T. Dean, K. Basye, R. Chekaluk, S. Hyun, M. Lejter, and M. Randazza. Coping with uncertainty in a control system for navigation and exploration. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1010–1015. AAAI, 1990.
- [41] T. Dean, K. Basye, and L. Kaelbling. Uncertainty in graph-based map learning. In J. Connel and S. Mahadevan, editors, *Robot Learning*. Kluwer Academic, 1993.
- [42] M. Drumheller. Mobile robot localization using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):325–332, march 1987. anche A.I. Memo 826, MIT, january 1985.
- [43] G. Dudek, P. Freedman, and S. Hadjres. Using uncertain sensing data to create reliable maps: an algorithm for exploring/mapping unknown graph-like worlds. In Wolfe and Chun, editors, *Mobile Robot VII*, pages 650–660. The International Society for Optical Engineering, November 1982.
- [44] G. Dudek, E. Miliot, and D. Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7(6):859–865, December 1991.
- [45] C. Durieu, J. Opderbecke, and Gilles Allegre. A data fusion application for location of a mobile robot using an odometer and a panoramic laser telemeter. In Groen, Hirose, and Thorpe, editors, *Intelligent Autonomous Systems*, pages 519–529. IOS Press, 1982.
- [46] H. F. Durrant-Whyte. Sensor model and multisensor integration. In A. Kak and S. Chen, editors, *Proceedings of the Workshop on Spatial Reasoning and Multi-sensor Fusion*, pages 303–312. Morgan Kaufmann, 1987.
- [47] H. F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31, february 1988.

- [48] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, june 1987.
- [49] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, pages 46–57, june 1987.
- [50] A. Elfes and L. Matthies. Sensor integration for robot navigation: Combining sonar and stereo range data in a grid-based representation. In *Proceedings of the 26th Conference on Decision and Control*, pages 1802–1807. IEEE, December 1987. anche in IEEE ICRA 1988.
- [51] P. Freedman and J. Liu. Using uncertain measurements to make plausible perceptual hypothesis. In Groen, Hirose, and Thorpe, editors, *Intelligent Autonomous Systems*, pages 258–265. IOS Press, 1982.
- [52] R. Froom. Approximate maps for high-speed control of a mobile robot. In Wolfe and Chun, editors, *Mobile Robot VII*, pages 619–630. The International Society for Optical Engineering, November 1982.
- [53] A. Fusiello and B. Caprile. Methods for automatic synthesis of maps of indoor environment in presence of uncertainty. Technical Report IRST, 1994. in preparazione.
- [54] E. Gat. On the role of theory in the control of autonomous mobile robots. presented at AAAI Fall Symposium on AI in Autonomous Mobile Robots,, 1992.
- [55] T. T. Good. Blank-map orienteering for a mobile robot using certainty grids. In Wolfe and Chun, editors, *Mobile Robot VII*, pages 631–642. The International Society for Optical Engineering, November 1982.
- [56] G. Hager and H. F. Durrant-Whyte. Information and multi-sensor coordination. In Lemmer and Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 381–395. Elsevier, 1988. anche in *Autonomous Robot Vehicles*.
- [57] T. C. Henderson and R. Grupen. Logical behaviors. *Journal of Robotic Systems*, 7(3):309–336, 1990.

- [58] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):122–139, march 1983.
- [59] Johnson and Wichen. *Applied multivariate statistical analysis*. Pretnice-Hall, 3 edition, 1992.
- [60] Kaelbling, Basye, Dean, Kokkevis, and Maron. Robot map-learning as learning labeled graphs from noisy data. Technical Report CS-92-15, Brown University Department of Computer Science, 1992.
- [61] J. R. Kender and A. Leff. Why direction-giving is hard: the complexity of using landmarks in one-dimensional navigation. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1656–1659, nov/dec 1989.
- [62] D. J. Kriegman, E. Triendl, and T. O. Binford. A mobile robot: Sensing, planning and locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 402–408. IEEE, 1987. in Autonomous Robot Vehicles.
- [63] B. J. A. Krose, K. M. Compagner, and F. C. A. Groen. Accurate estimation of environment parameters from ultrasonic data. In Groen, Hirose, and Thorpe, editors, *Intelligent Autonomuos Systems*, pages 622–630. IOS Press, 1982.
- [64] R. Kuc. A spatial sampling criterion for sonar obstacle detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):686–690, july 1990.
- [65] R. Kuc and M. W. Siegel. Physically based simulation model for acoustic sensor robot navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):766–778, november 1987.
- [66] R. Kuc and V. B. Viard. A phisycally based navigation strategy for sonar-guided vehicles. *International Journal of Robotics Research*, 10(2):75–87, april 1991.
- [67] B. Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2:129–153, 1978.

- [68] B. Kuipers and Y. T. Byun. A qualitative approach to robot exploration and map-learnibg. In A. Kak and S. Chen, editors, *Proceedings of the Workshop on Spatial Reasoning and Multi-sensor Fusion*, pages 390–404. Morgan Kaufmann, 1987.
- [69] B. Kuipers and Y. T. Byun. A robust qualitative method for robot spatial learning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 774–779. AAAI, 1988.
- [70] B. Kuipers and Y. T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.
- [71] B. Kuipers and T. S. Levitt. Navigation and mapping in large-scale space. *AI magazine*, pages 25–43, summer 1988.
- [72] S. Y. T. Lang and A. K. Wong. Building geometric world models with graph syntesis for sensor fusion in mobile robots. *Computational Intelligence*, 6:91–107, 1990.
- [73] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, june 1991.
- [74] J. J. Leonard and H. F. Durrant-Whyte. *Direct Sonar Sensing for Mobile Robot Navigation*. Kluver Academic, 1992.
- [75] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):286–298, august 92.
- [76] T. S. Levitt, D. T. Lawton, D. M. Chelberg, and P. C. Nelson. Qualitative landmark-based path planning and following. In *Proceedings of the National Conference on Artificial Intelligence*, pages 689–694. AAAI, 1987.
- [77] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.

- [78] R. C. Luo and M. G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, 19(5):901–931, sept/oct 1989.
- [79] R. C. Luo, M. Lin, and R. S. Scherp. Dynamic multi-sensor data fusion system for intelligent robots. *IEEE Journal of Robotics and Automation*, 4(4):386–396, august 1988.
- [80] R. Malik and S. Prasad. Robot mapping with proximity detectors. In Wolfe and Chun, editors, *Mobile Robot VII*, pages 614–618. The International Society for Optical Engineering, November 1982.
- [81] P. K. Malkin and S. Addanki. Lognets: A hybrid graph spatial representation for robot navigation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1045–1050. AAAI, 1990.
- [82] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312, june 1992.
- [83] Maja. J. Mataric. A distributed model for mobile robot environment-learning and navigation. Technical Report AI-TR 1228, M.I.T., may 1990. revised version of Master Thesis.
- [84] D. Miller. A spatial representation system for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 122–127. IEEE, 1985.
- [85] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, summer 1988. anche in NATO ASI Series F52.
- [86] H. P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 116–121. IEEE, march 1985.
- [87] N. J. Nilsson. *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill, 1971.

- [88] N. J. Nilsson. Shakey the Robot. Technical Report 323, SRI International, April 1984.
- [89] F. Noreils and R. Chatila. Control of Mobile robot actions. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 701–707, Washington, DC, May 1989. IEEE Computer Society Press.
- [90] O. Patrouix, C. Novales, and B. Jouvencel. Basic aerial telemeter by ultrasonic sonar: Fast extraction of principal range information of a scene. In *Proceedings of the International Conference on Advanced Robotics*, pages 567–572, november 1993.
- [91] L. F. V. Pau. Behavioral knowledge in sensor/data fusion systems. *Journal of Robotic Systems*, 7(3):295–308, 1990.
- [92] T. Poggio and L. Stringa. A Project for an Intelligent System: Vision and Learning. *International Journal of Quantum Chemistry*, 42:727–739, 1992.
- [93] R. Ramparany. An integrated support for fusion perceptual information. In Groen, Hirose, and Thorpe, editors, *Intelligent Autonomuos Systems*, pages 500–508. IOS Press, 1982.
- [94] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, october 1992.
- [95] R. L. Rivest and R. Sloan. Learning complicated concepts reliably and usefully. In *Proceedings of the National Conference on Artificial Intelligence*, pages 635–640. AAAI, 1988.
- [96] Rl L. Rivest and R. E.. Schapire. Inference of finite automata using homing sequences. *Information and Computation*, 103:299–347, 1993.
- [97] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. *IEEE Computer*, pages 85–90, june 1989.
- [98] G. Shaffer, J. Gonzales, and A. Stenz. Comparson of two range-based pose estimators for mobile robot. In Wolfe and Chun, editors, *Mobile Robot VII*,

- pages 661–667. The International Society for Optical Engineering, November 1982.
- [99] C. Shekhar and R. Chellappa. Passive ranging using a moving camera. *Journal of Robotic Systems*, 9(6), 1992.
- [100] W. Shen. Discovery as autonomous learning from the environment. *Machine Learning*, 12:143–165, 1993.
- [101] S. Skiena. *Implementing Discrete Mathematics*. Addison Wesley, 1990.
- [102] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In Lemmer and Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 435–461. Elsevier, 1988. anche in *Autonomous Robot Vehicles*.
- [103] L. Stringa. Il sistema MAIA. *Le Scienze*, 290:86–97, October 1992.
- [104] Y. C. Tang and C. S. George Lee. A geometric feature relation graph formulation for consistent sensor fusion. *IEEE Transactions on Systems, Man and Cybernetics*, 22(1):115–129, January 1992.
- [105] The Analytic Science Corporation Technical Staff. *Applied Optimal Estimation*. The M.I.T. Press, 1974.
- [106] S. C. A. V. Thomopoulos. Sensor integration and data fusion. *Journal of Robotic Systems*, 7(3):337–372, 1990.
- [107] C. E. Thorpe. Mobile robots. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(3):383–397, 1991.
- [108] C. E. Thorpe, M. Hebert, T. Kanade, and S. A. Shafer. Vision and Navigation for the Carnegie-Mellon Navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3):362–373, March 1988.
- [109] C. M. Wang. Location estimation and uncertainty analysis for mobile robot. In Cox and Wilfong, editors, *Autonomous Robot Vehicle*. Springer-Verlag, 1990.
- [110] A. Zelinsky. Environment mapping with a mobile robot using sonar. In *Proceedings of the Australian Joint Artificial Intelligence Conference*, pages 363–378, 1988.

- [111] A. Zelinsky. Mobile robot map making using sonar. *Journal of Robotic Systems*, 8(5):557–577, 1991.
- [112] S. Zhao, A. Schmitt, and D. Meizel. Ultrasonic simulation in a large scope of incident angles and visible environment dynamical modelling for mobile robot using a multi-rings configuration. In *Proceedings of the International Conference on Advanced Robotics*, pages 573–578, november 1993.