



UNIVERSITÀ DI BRESCIA
FACOLTÀ DI INGEGNERIA
Dipartimento di Elettronica per l'Automazione

Laboratorio di Robotica Avanzata **Advanced Robotics Laboratory**

Corso di Robotica
(Prof. Riccardo Cassinis)

Sensori di distanza ad infrarosso per Tobor

Elaborato di esame di:

Stefano Coccoli, Manuel Vega

Consegnato il:

29 luglio 2004

Sommario

L'obiettivo del seguente lavoro è quello di fornire al robot Pioneer 1 dell'ActivMedia Robotics un sistema di rilevazione di ostacoli che possa aggiungersi e magari complementare il sistema di sonar già presente. Tale nuovo sistema consiste di due sensori di distanza ad infrarosso posizionati su una torretta rotante, la cui rotazione è permessa da un servomotore da modellismo. Nella realizzazione si è sfruttata la presenza della scheda di espansione "Experimenter Module" presente su uno dei due robot Pioneer 1 del Laboratorio di Robotica Avanzata della facoltà. È stato, però, necessario modificarne l'hardware originale in modo da adattarlo alle nostre esigenze. Il software è stato interamente scritto in C++ in modo da poter essere utilizzato nell'ambiente di sviluppo Saphira 8.2. Le soluzioni adottate permettono all'utente di visualizzare in modo efficace le letture dei sensori agendo sulla scelta di alcuni parametri ed opzioni.

1. Introduzione

La presente relazione si propone di documentare in modo dettagliato tutte le fasi e i vari problemi riscontrati durante lo sviluppo del nostro lavoro avente come scopo la realizzazione di un sistema ad infrarosso in grado di rilevare ostacoli per il Pioneer 1. L'idea era quella di poter visualizzare i risultati ottenuti attraverso l'ambiente Saphira in modo molto simile a come vengono visualizzate le misurazioni dei sonar, strumenti che fanno parte della dotazione standard del robot.

Ci soffermeremo in particolare su due aspetti differenti:

- software, facendo attenzione alle scelte effettuate durante la stesura del codice e alle ottimizzazioni possibili;
- hardware, descrivendo ciò di cui inizialmente si disponeva e i vari problemi riscontrati con conseguente risoluzione.

Viene qui di seguito proposta una serie di paragrafi atti a far comprendere quali fossero i dispositivi di partenza a nostra disposizione.

1.1. Analisi dei dispositivi del nostro sistema

La prima fase del nostro lavoro è stata l'analisi e il testing dei dispositivi che sarebbero andati a costituire il sistema di rilevazione degli ostacoli:

- Due sensori analogici di distanza ad infrarosso (Sharp GP2Y0A02YK)
- Un servomotore da modellismo (MX-400)

1.1.1 Sensori di distanza ad infrarosso

I sensori da noi utilizzati, i GP2Y0A02YK sono di tipo analogico anche se la loro uscita in realtà assume valori quantizzati. Alimentati a 5 V forniscono un valore di tensione in uscita compreso tra 2,5 V e 0,43 V a seconda che l'ostacolo rilevato sia ad una distanza rispettivamente tra i 20 cm e i 150 cm. La loro tensione di uscita decresce infatti all'aumentare della distanza dell'oggetto che ostacola il raggio emesso dall'emettitore di luce secondo la seguente legge non lineare mostrata in Figura. 1.1. [1].

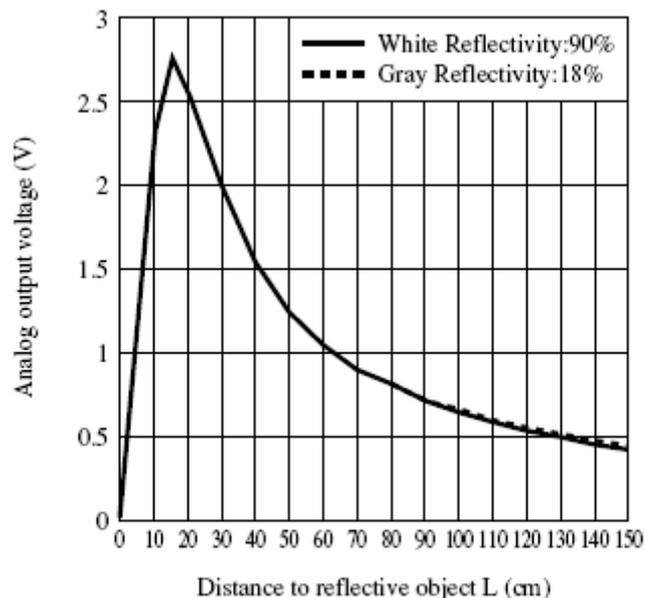


Figura 1.1. Curva caratteristica del sensore IR Sharp GP2Y0A02YK.

1.1.2. Servomotore

Il servomotore da noi utilizzato è un MX-400 (casa costruttrice Acroname) usato spesso nel settore del modellismo. Il suo albero si posiziona a diverse angolazioni a seconda della larghezza dell'impulso del segnale di ingresso che è quello di un PWM standard. Il segnale deve essere generato ogni 20 ms, quindi, ad una frequenza di 50 Hz e deve restare alto per una durata che varia tra i 0,5 e 2,5 ms in base all'angolazione dell'albero che si vuole ottenere. L'angolazione varia tra 0° e 180° e ad ogni aumento di 0,1 ms della larghezza dell'impulso corrisponde una rotazione di 9° . La Figura 1.2. mostra il legame tra larghezza dell'impulso e posizione del servo motore [2].

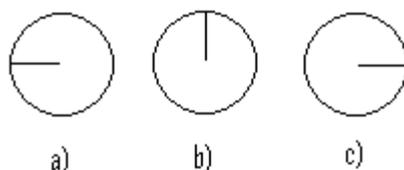


Figura 1.2. Posizione dell'albero del servomotore a cui è stato applicato un segnale di controllo della larghezza di a) 0,5 millisecondi; b) 1,5 millisecondi; c) 2,5 millisecondi

1.2. Caratteristiche tecniche di Tobor

La seconda fase del nostro lavoro si è focalizzata sulla ricerca di documentazioni relative a Tobor, il robot che ci è stato messo a disposizione dal Laboratorio di Robotica Avanzata della facoltà. Tobor è un Pioneer 1 dell'ActivMedia Robotics i cui dispositivi (sonar, motori, porte I/O etc.) sono gestiti da un microcontrollore (MC68HC11). Tobor è stato anche dotato di una scheda di espansione chiamata Experimenter module Figura 1.3. [10] che permette al microcontrollore di gestire un numero di porte I/O superiore a quello standard .

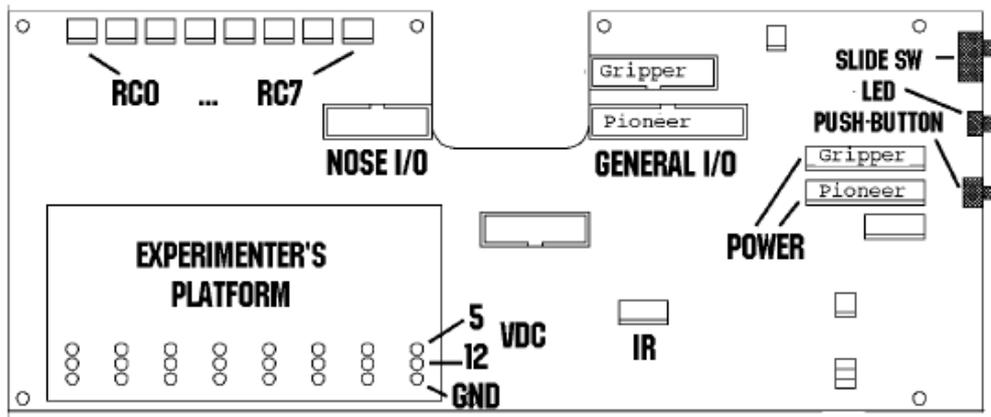


Figura 1.3. Schema semplificato Experimenter module.

Tale espansione permette infatti di moltiplicare l'unico ingresso analogico e l'unica uscita per il controllo del servomotore. Gli ingressi analogici diventano 8 e vengono selezionati da un multiplexer MC74HC4051N, le uscite digitali (anche loro 8) sono abilitate da un demultiplexer 74HCT238N (si veda la Figura 1.4.).

Tutte le linee di input e di output sono collegate al microcontrollore del robot attraverso due connettori: NOSE PORT e GENERAL PORT.

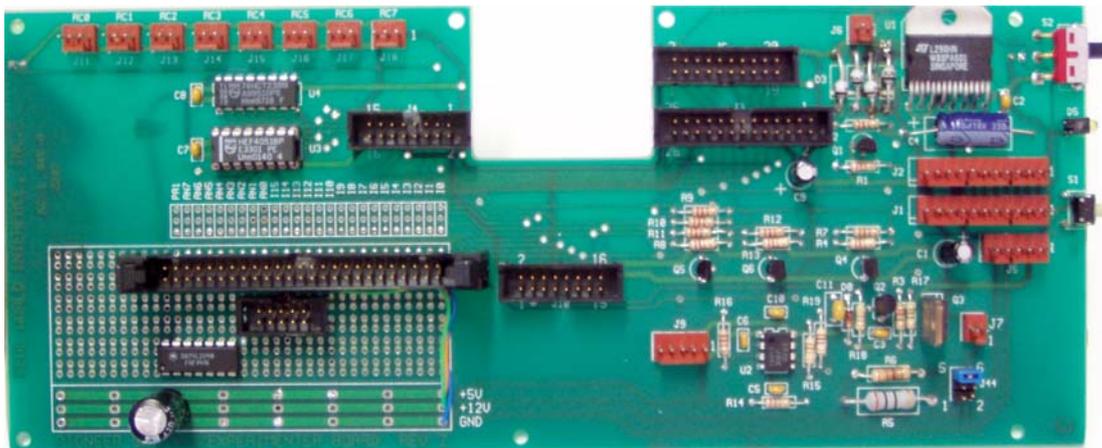


Figura 1.4. Experimenter module.

Il microcontrollore, inoltre, è dotato di un sistema operativo detto Pioneer Server Operating System (PSOS) [9], il quale si occupa della gestione a basso livello di tutti i dispositivi del robot. L'utente può, in questo modo, interagire ad un livello più elevato attraverso una serie di comandi. Due di queste istruzioni sono particolarmente interessanti per il nostro progetto:

- **DIGOUT** → Modifica il valore dei bit delle uscite digitali. Necessita di due byte come argomento. Il più significativo (la maschera) indica quali uscite vengono abilitate ad essere modificate, l'altro parametro è il valore da assegnare alle linee precedentemente selezionate.

Nella Tabella 1.1. sono riportate le uscite digitali modificabili attraverso il comando DIGOUT.

Porte I/O del pioneer		
Etichetta	Pin	Uso
OD0	G17	Direzione motore delle pinze
OD1	G19	Abilitazione del motore delle pinze
OD2	G21	Led per l'utente
OD3	G23	Paddle destro e led
OD4	G25	Paddle sinistro e led
OD5	N13	Speaker
OD6	N11	Servo motori per l'utente
OD7	N9	Servo motori per l'utente

Tabella 1.1. Uscite digitali

Ad esempio se si volessero settare le uscite OD2 e OD4 rispettivamente a 0 e 1, l'istruzione che si dovrebbe utilizzare sarebbe:

```
SfROBOT->com2Bytes(Arcommands::DIGOUT, 0x08, 0x00)
```

che in termini di byte passati come argomento si traduce:

	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
Maschera	0	0	0	1	0	1	0	0
Parametro	0	0	0	1	0	0	0	0

- **PTUPOS** → Serve per il controllo dei servomotori, permette di generare un'onda quadra di larghezza desiderata. Richiede due byte come argomento: il primo è l'indirizzo di un servo motore (0-4); il secondo è la larghezza di impulso (espressa in decine di millisecondi).

Ad esempio l'istruzione:

```
SfROBOT->com2Bytes(Arcommands::PTUPOS, 0, 150)
```

abilitata la porta RC0 e genera un segnale di durata 1500 millisecondi (valore che fa posizionare il nostro servomotore nella posizione di riposo).

Da notare che sebbene sull'Experimenter module ci siano otto porte per il controllo dei servomotori, solo cinque di queste (RC0 ÷ RC4) sono realmente utilizzabili come specificato nel manuale [10]. È utile anche tener presente che senza l'Experimenter Module il PSOS ignora l'indirizzo RC e genera al pin 15 della NOSE PORT (ODT) il segnale specificato dal comando.

Le istruzioni del PSOS non sono state da noi utilizzate direttamente ma sono state richiamate dall'applicazione client Saphira residente sul PC. Tale applicazione comunica con il PSOS usando uno speciale protocollo a pacchetti che però ne limita la velocità ad un comando ogni 100 millisecondi. Esistono metodi designati a tale scopo, uno di questi è: *bool com2Bytes (unsigned char command, char high, char low)* incluso nella classe *ArRobot.h*.

Un altro metodo di nostro interesse della classe *ArRobot.h* è *int getAnalog(void)*, che restituisce il valore digitalizzato letto sulla porta analogica. Il range dei valori è compreso tra 0 e 5 volt e viene convertito in un integer che varia tra 0 e 255.

2. Il problema affrontato

2.1. Una prima proposta

La soluzione più immediata e semplice per affrontare il problema è sembrata subito quella di poter usufruire dei vantaggi offerti dall'Experimenter module. L'idea di avere a disposizione più porte d'ingresso analogiche avrebbe permesso l'utilizzo di più sensori da posizionare sulla torretta ruotante. Il servomotore, poi, sarebbe stato collegato ad una delle otto uscite predisposte per tale scopo.

Il problema, quindi, richiedeva in primo luogo di dover decidere quanti sensori montare sul servomotore stesso. Poiché la corsa massima dell' MX-400 è circa 180 gradi, la scelta migliore è ricaduta sull'utilizzo di due sensori ad infrarosso.

Sfruttando l'Experimenter module, nessuna ulteriore implementazione hardware sarebbe servita poiché tutto il necessario era già incluso sulla scheda. Il microcontrollore come già detto possiede un solo canale di ingresso analogico quindi le letture dei sensori non possono essere simultanee, è necessario quindi commutare il multiplexer presente sull'Experimenter module andando così a selezionare una delle linee di ingresso desiderate. Le linee OD5, OD6 e OD7 della NOSE PORT sono dedicate al controllo di tale multiplexer [10].

Seguendo questa idea abbiamo così creato il nostro primo prototipo: un servomotore collegato alla RC0, due sensori di distanza collegati alle porte AN0, AN1.

2.2. Analisi dei malfunzionamenti

Sono stati riscontrati diversi malfunzionamenti del sistema appena realizzato:

- La rotazione del servomotore non era fluida ma disturbata da vibrazioni causate dalla presenza di rumore nel segnale di controllo del servomotore.
- Le letture dei due sensori erano molti simili anche quando l'ambiente esterno e la disposizione degli ostacoli era molto diversa.
- Il robot si disconnetteva spesso.

Abbiamo provveduto allo smontaggio della scheda Experimenter module notando che il demultiplexer che controlla le porte per i servomotori e il multiplexer degli ingressi analogici avevano le medesime linee di controllo (OD5-OD6-OD7) ed entrambi i chip erano costantemente abilitati. Quindi il servomotore e i sensori non potevano essere utilizzati contemporaneamente.

Neppure l'inserimento di qualche meccanismo di sincronizzazione software tale da produrre il movimento del servomotore e la lettura dei sensori in intervalli di tempo differenti ha portato alla risoluzione dei problemi. Questo perché ci siamo resi conto che l'istruzione di controllo dei servomotori, la PTUPOS, non limita la propria azione solo al ciclo di elaborazione nel quale è stata chiamata, ma da quell'istante in poi impegna costantemente le linee di selezione e l'ingresso del demultiplexer rendendo impossibile l'utilizzo delle medesime per ulteriori scopi. Quindi, il tentativo di utilizzare nello stesso progetto un servomotore ed un ingresso analogico, se non posizionati sullo stesso numero di porta, risulta essere del tutto vano.

Ad ogni ciclo il segnale di selezione del multiplexer analogico veniva cancellato e sovrapposto dal segnale continuo dell'istruzione PTUPOS che settava le tre linee di selezione a zero (000 corrisponde alla porta RC0) costringendo il multiplexer a leggere le misure sempre dalla linea AN0. Inoltre i vani tentativi di commutazione di quest'ultimo provocavano del rumore sul segnale del servomotore che si manifestava sotto forma di vibrazioni indesiderate.

Le continue sconnessioni del Pioneer, invece, erano dovute all'eccessiva richiesta di corrente da parte del nostro sistema.

3. La soluzione adottata

Un intervento diretto sull'Experimenter Module ci è sembrato il rimedio migliore ai problemi precedentemente descritti. Qui di seguito vengono riportati i nostri interventi.

- Abbiamo isolato il multiplexer analogico, recidendo le piste che collegavano le uscite digitali OD5, OD6, OD7 ai suoi piedini di selezione. Necessitando di un solo bit di controllo per selezionare l'ingresso desiderato del multiplexer, abbiamo usato la linea G23 della GENERAL PORT, la quale pilota anche il led sulla pinza. Le continue commutazioni del multiplexer causano, infatti, il lampeggiare di tale led..
- Per quanto riguarda il servomotore, essendo l'unico, si è pensato di comandarlo direttamente con l'uscita PWM generata dal microcontrollore del Pioneer (Pin 15 della NOSE PORT indicato con ODT). L'uscita PWM del microcontrollore (ODT) è però negata rispetto al segnale che ci serve per comandare il servomotore. E' stato, quindi, necessario inserire un invertitore 74LS04N per complementarlo.

Da notare che durante tali modifiche si è bruciato il multiplexer analogico, da noi sostituito con un nuovo multiplexer HE4051 con piedinatura identica a quella del modello precedente. Durante la sostituzione abbiamo inserito anche uno zoccolo sul quale posizionarlo.

Per ultimo, il problema dell'eccessiva richiesta di corrente dovuta sia al servomotore che ai due sensori è stato risolto con l'inserimento di un blocco di alimentazione a 5 V che abbassa e stabilizza i 12 V presi direttamente dalle batterie del Pioneer. Il blocco di alimentazione è formato da un LM7805 [3] e da condensatori di cui due di tipo elettrolitico (da 470 μ F in ingresso e da 100 μ F in uscita) e due ceramici da 100 nF (uno in parallelo all'ingresso e uno all'uscita). Il condensatore elettrolitico ha grande capacità ed elevata induttanza, il condensatore ceramico ha bassa capacità e una minor induttanza. Il loro accoppiamento sia in ingresso che in uscita dello stabilizzatore permette di avere un sistema con grande carica che risponda velocemente alle richieste di energia che la batteria non riesce a soddisfare rapidamente.

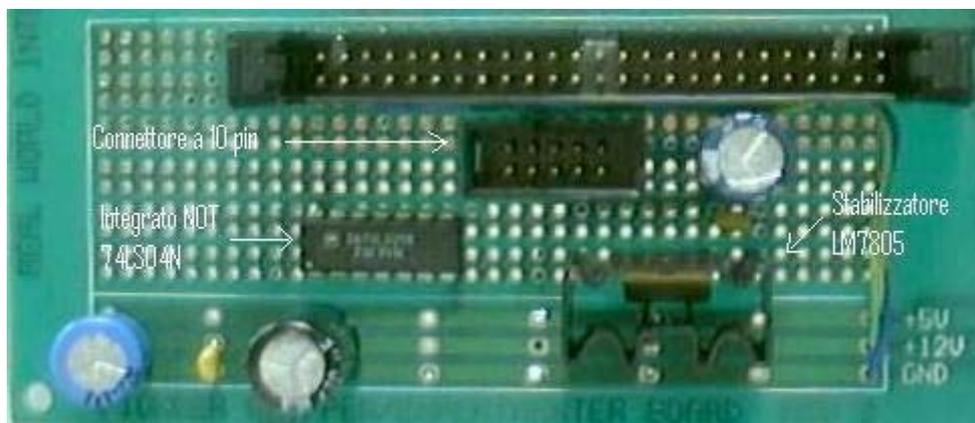


Figura 3.1. Condensatori, stabilizzatore, invertitore, connettore a 10 pin

Per una maggior usabilità e chiarezza, si è pensato di riunire tutti i collegamenti che andavano dall'Experimenter Module al nostro sistema in un unico cavo con prese da 10 pin su entrambe le estremità. Sulla scheda è stato integrata una spina a 10 pin, le cui connessioni sono riportate nella tabella 3.1.

Pin	Uso	Pin	Uso
1	Non connesso	6	Sensore 1 (AN0)
2	Ground	7	Vcc
3	Vcc	8	Servomotore (pin 2 del 74LS04N)
4	Ground	9	Sensore 2 (AN1)
5	Ground	10	Vcc

Tabella 3.1. Piedinatura connettore a 10 Pin

Il servomotore è stato inserito in un contenitore plastico in modo da riunire tutti i fili in una spina a 10 pin collegabile al cavo di cui precedentemente si è parlato (Figura 3.2).



Figura 3.2. Contenitore plastico

3.1. Testing del sistema

Dopo la fase di implementazione hardware e di sviluppo del relativo software ci si è dedicati ad una fase di testing per valutare le prestazioni del sistema appena realizzato. Sono stati effettuati due test:

- Un primo test atto a stabilire la qualità delle misure dei sensori in posizione fissa, disabilitando il servomotore mediante l'apposita funzione.
- Il secondo test, invece, realizzato con il servomotore abilitato, con lo scopo di verificare se il sistema sia in grado di ricostruire l'ambiente circostante in maniera soddisfacente.

Durante la valutazione della qualità delle letture delle misure si è notata una certa imprecisione. Ponendo un oggetto, come una scatola di cartone, ad una certa distanza e tenendo il servomotore disabilitato, il valore letto non rimaneva costante, anzi tendeva a variare ed oscillare.

Analizzando direttamente il segnale uscente da uno dei suoi sensori attraverso un oscilloscopio digitale, ci si è resi conto che risultava essere molto disturbato. La causa di tale disturbo è il modem wireless di cui è equipaggiato il Pioneer ed utilizzato durante la comunicazione con Saphira. I fili che collegano il nostro sistema al corpo del robot erano soggetti a disturbi elettromagnetici e come se non bastasse dalle prove svolte il radiomodem influenzava anche la circuiteria interna dei sensori IR provocando un aumento spropositato nell'errore delle letture. Come visualizzato in figura 3.3, l'ampiezza del disturbo risultava essere superiore ai 3 V rendendo così impossibile una lettura che fosse coerente con il mondo reale esterno.

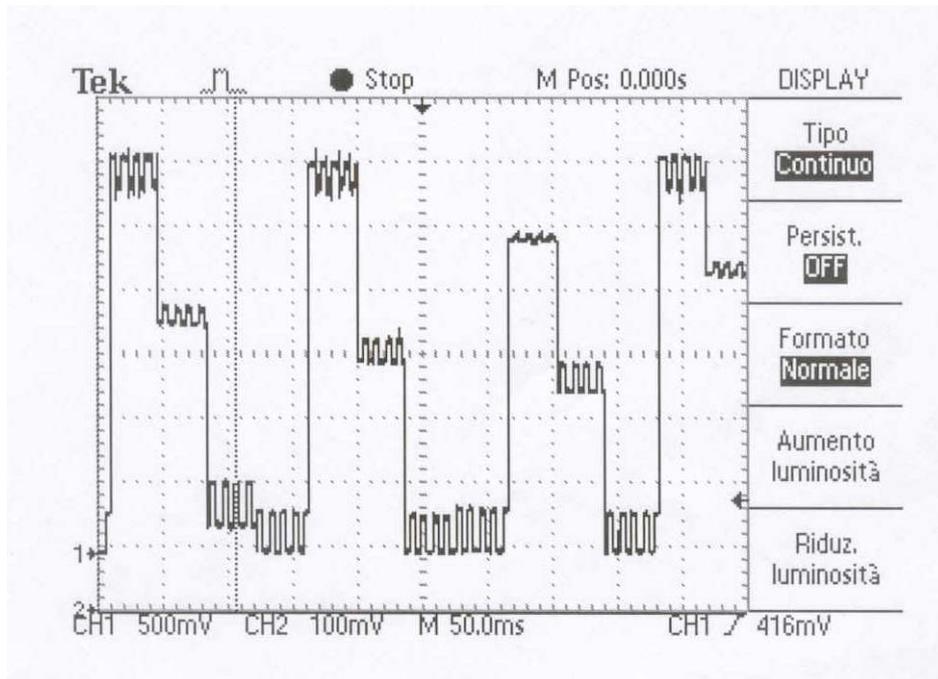


Figura 3.3. Forma d'onda del segnale con da disturbo visualizzato dall'oscilloscopio

Per ridurre tale effetto indesiderato sono stati presi alcuni accorgimenti:

- Si è schermato il cavo di collegamento e lo si è fatto passare il più lontano possibile dal modem.
- Si sono ricoperti con un nastro ramato collegato a massa i sensori ad infrarosso in modo da schermare la loro circuiteria interna.
- All'ingresso dell'alimentazione dei due sensori sono stati posizionati due condensatori, uno elettrolitico e l'altro ceramico, in modo da eliminare i disturbi ad alta frequenza e in modo da aver una corrente d'ingresso il più stabile possibile.
- Per quanto riguarda il segnale, oltre che inserire un condensatore di 470 μF sui contatti del sensore, si è pensato di porre un filtro RC all'ingresso del convertitore analogico del Pioneer in modo da eliminare parte delle frequenze indesiderate. La frequenza di taglio di tale filtro è stata scelta in base a considerazioni sulla dinamica del segnale di ingresso. Poiché ogni sensore può fornire al massimo un valore che varia da circa 0,4 V a circa 2,8 V ogni tre cicli da 100 ms ciascuno e cioè ogni 300 ms, la dinamica del segnale sarà al massimo di 3,34 Hz. Quindi, dato che la maggior parte dell'energia del segnale è contenuta nei 4 Hz, è possibile impostare la frequenza di taglio a circa 4 Hz. Il filtro è stato realizzato con un condensatore da 8 μF e una resistenza da 4700 Ω in serie con l'impedenza resistiva dei sensori che è di circa 200 Ω .

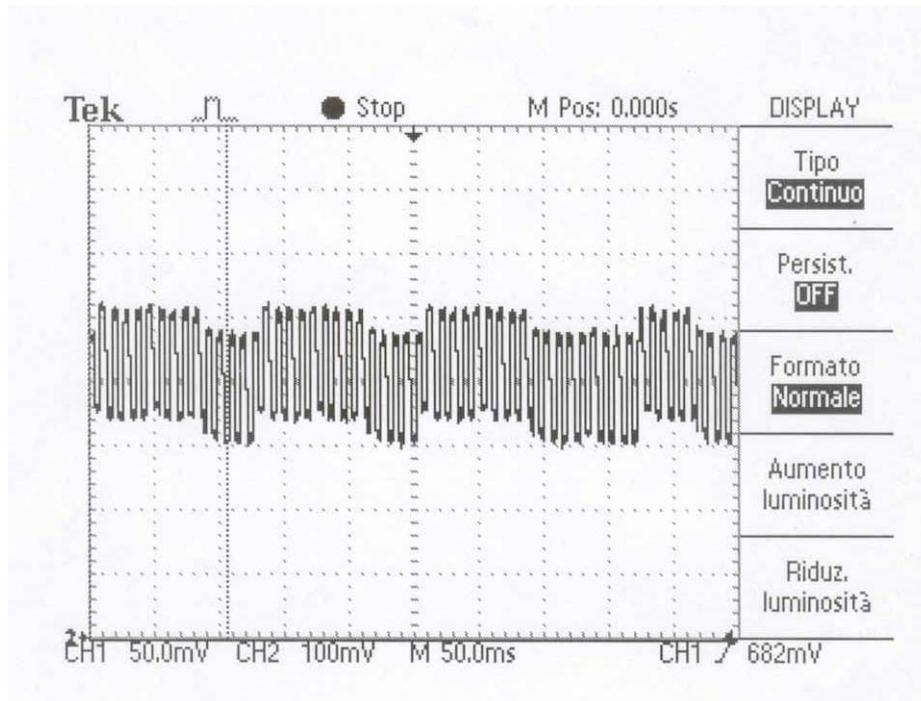


Figura 3.4. Forma d'onda del segnale dopo i filtri e le schermature

In figura 3.4 è riportato l'andamento del segnale all'ingresso del convertitore analogico-digitale dopo tali modifiche. Si può osservare che il disturbo non è stato eliminato completamente, ma ha subito una sostanziale riduzione sufficiente a rendere le letture abbastanza soddisfacenti. In effetti, il valore numerico derivato dalla conversione del segnale analogico oscilla tra più o meno un bit. Tale variazione non arreca grosse imprecisioni per distanze inferiori a 110 cm, dove 10 cm sono mappati su più bit. Mentre per distanze superiori, a causa della non linearità della quantizzazione, la variazione di un bit può corrispondere anche ad un salto di 10 cm.

Abbiamo così eseguito nuovamente i test, effettuando le misure solo tra 20 e 110 cm, seppure il range dei sensori fosse tra 20 e 150 cm. Lasciando disabilitato il servomotore e ponendo una scatola di cartone a diverse distanze dal sistema, abbiamo riportato i valori letti nelle tabelle 3.2 e 3.3. Ogni lettura è accompagnata dall'errore rispetto al valore reale.

Distanza effettiva	Prova 1	% Errore Prova 1	Prova 2	% Errore Prova 2	Prova 3	% Errore Prova 3	Prova 4	% Errore Prova 4	Prova 5	% Errore Prova 5	MEDIA
200	200	0,00	200	0,00	200	0,00	200	0,00	200	0,00	200
300	300	0,00	297	1,00	297	1,00	297	1,00	300	0,00	298
400	400	0,00	400	0,00	408	2,00	400	0,00	408	2,00	403
500	519	3,80	519	3,80	500	0,00	493	1,40	493	1,40	505
600	600	0,00	600	0,00	600	0,00	600	0,00	591	1,50	598
700	712	1,71	712	1,71	712	1,71	723	3,29	712	1,71	714
800	786	1,75	800	0,00	800	0,00	800	0,00	817	2,13	801
900	900	0,00	900	0,00	900	0,00	900	0,00	934	3,78	907

1000	1000	0,00	1000	0,00	967	3,30	1000	0,00	1000	0,00	993
1100	1084	1,45	1084	1,45	1067	3,00	1084	1,45	1067	3,00	1077

Tabella 3.2. Letture del sensore anteriore in posizione statica

Distanza effettiva	Prova 1	% Errore Prova 1	Prova 2	% Errore Prova 2	Prova 3	% Errore Prova 3	Prova 4	% Errore Prova 4	Prova 5	% Errore Prova 5	MEDIA
200	200	0,00	200	0,00	200	0,00	200	0,00	200	0,00	200
300	300	0,00	295	1,67	295	1,67	300	0,00	300	0,00	298
400	419	4,75	407	1,75	419	4,75	419	4,75	419	4,75	417
500	500	0,00	500	0,00	510	2,00	481	3,80	510	2,00	500
600	590	1,67	580	3,33	600	0,00	600	0,00	600	0,00	594
700	700	0,00	717	2,43	731	4,43	684	2,29	684	2,29	703
800	800	0,00	813	1,63	817	2,13	834	4,25	800	0,00	813
900	900	0,00	900	0,00	900	0,00	867	3,67	925	2,78	898
1000	1000	0,00	975	2,50	950	5,00	1000	0,00	975	2,50	980
1100	1100	0,00	1067	3,00	1100	0,00	1100	0,00	1034	6,00	1080

Tabella 3.3. Letture del sensore posteriore in posizione statica

Il secondo test è stato effettuato per valutare la bontà con la quale il sistema riesce a ricostruire il mondo circostante al robot, rilevando la posizione di eventuali ostacoli. Come schematizzato in figura 3.5, sono stati disposti intorno a Tobor tre scatoloni di dimensione e colore differente.

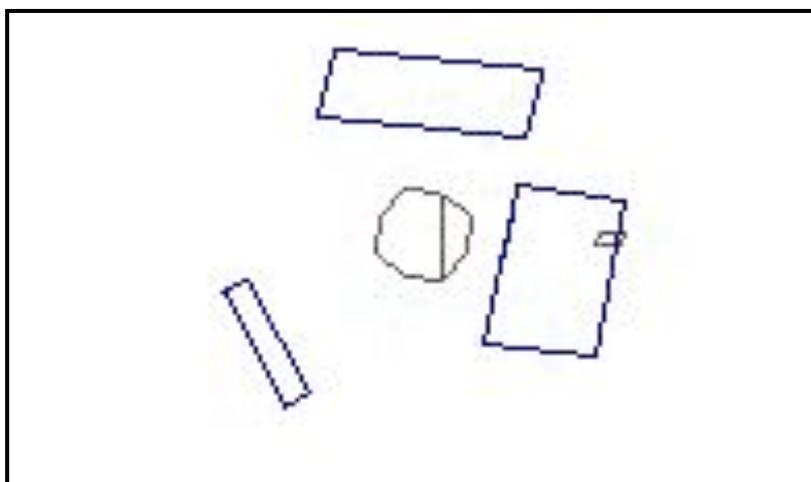


Figura 3.5. Schematizzazione dell' ambiente reale in cui è stato posto Tobor

Una volta avviato il sistema con il servomotore in funzione, tutti e tre gli ostacoli sono stati rilevati e visualizzati in Saphira (Si veda figura 3.6). Per i due scatoloni più prossimi al Pioneer le sagome individuate corrispondono grosso modo alla realtà. Maggior difficoltà è stata invece riscontrata per oggetti più distanti, come ad esempio la scatola in basso a sinistra (Fig.3.5). Le misure, infatti, individuano la presenza di qualcosa sebbene le distanze e la sagoma non siano molto precise.

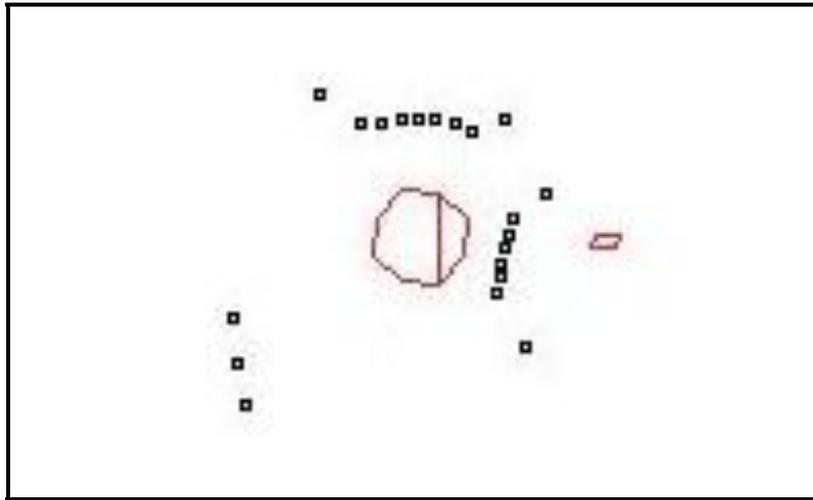


Figura 3.6. Mondo rilevato dal Pioneer e graficato in Saphira

Modalità operative

Segue un'attenta analisi dei requisiti necessari e delle funzionalità offerte dal nostro sistema una volta montato ed installato.

3.2. Componenti necessari

Il corretto funzionamento del sistema richiede da un punto di vista hardware:

- Un robot Pioneer 1 dell'ActivMedia Robotics provvisto dell'espansione Experimenter Module da noi modificata come precedentemente spiegato. Il Robot da noi utilizzato è Tobor, messo a disposizione dal Laboratorio di Robotica Avanzata della facoltà.
- Un servomotore MX-400.
- Due sensori ad infrarosso GP2Y0A02YK.
- Il supporto plastico per contenere il servomotore.
- Un cavo con entrambe le spine a 10 pin.
- Un computer con il quale comunicare attraverso il robot sul quale è installato il sistema operativo Linux. Noi abbiamo utilizzato la versione Mandrake 9.0.

Dal lato software:

- Compilatore gcc versione 2.95.3.

- Il pacchetto Aria versione 1.2.0.
- Il pacchetto Saphira versione 8.2.0.
- Ed infine il software da noi realizzato consistente di due file scritti in C++.

3.3. Modalità di installazione

3.3.1. Installazione Hardware

Nel caso in cui si stesse utilizzando Tobor, quest'ultimo dovrebbe essere già provvisto dell' "Experimenter Module" da noi modificato e sul suo dorso si dovrebbe intravedere un connettore a 10 pin del cavo precedentemente menzionato. Basta, quindi, collegare tale connettore alla porta del supporto plastico contenente l'intero sistema. Il connettore deve essere collegato con la sigla RS rivolta verso il basso. L'altro connettore del cavo deve essere collegato alla presa di 10 pin saldata sulla scheda con la sigla RS rivolta verso il basso, facendo passare il cavo attraverso le fessure del telaio di Tobor.

3.3.2. Installazione Software

L'installazione dei pacchetti relativi ad Aria e Saphira viene rimandata ai rispettivi manuali scaricabili via Web [5] [6]. Per quanto riguarda il software da noi realizzato, esso deve essere copiato in una sottocartella di secondo livello rispetto alla cartella nella quale è installato Saphira. Utilizzando tale directory come corrente, digitare uno dei seguenti comandi:

- **make** compila i file e crea il file shared object sensor.so che viene copiato nella corretta cartella di Saphira. Tale libreria deve essere caricata da Saphira con il comando "loadlib IRsensor". Si utilizzino i comandi "IRstart()" e "IRstop()" per avviare o fermare il sistema.
- **make no_gui** compila il software senza i file corrispondenti all'interfaccia grafica. Al momento dell'avvio non sarà possibile cambiare runtime le impostazioni del sistema attraverso il menu a tendine.
- **make clean** cancella i file object precedentemente creati (i file con estensione .o).

3.4. Funzionalità offerte

Al momento dell'avvio del thread che gestisce l'intero sistema, il servomotore ruota nella posizione iniziale con i due sensori, uno rivolto anteriormente ed uno rivolto posteriormente al robot. Inizia, poi, a ruotare spostandosi di un passo alla volta, in corrispondenza del quale vengono lette le misure fornite da entrambi i sensori. Una volta che il servomotore è giunto a fine corsa (circa 180°), viene invertito il senso di rotazione.

Le misure effettuate vengono visualizzate graficamente attraverso Saphira in modo del tutto simile alla visualizzazione delle misure dei sonar.

Il sistema è, inoltre, dotato di parecchi parametri che possono essere modificati runtime a seconda delle esigenze dell'utente. La modifica può avvenire, oltre che per via software attraverso la chiamata a specifiche procedure, attraverso un'interfaccia grafica attivabile cliccando sulla voce IR Sensor dal menu Sensor. Il frame ha l'aspetto di uno schedario consistente di tre "linguette" come visualizzato in Figura 4.1. Le modifiche vengono rese effettive premendo il tasto "Update" oppure "Ok"

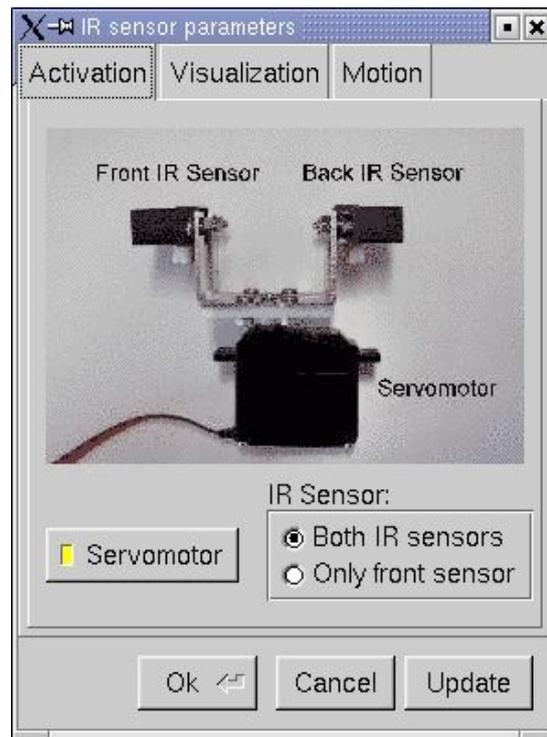


Figura 4.1. Il frame per il controllo del sistema IR Sensor

I parametri modificabili possono essere suddivisi in parametri riguardanti il moto e parametri riguardanti la visualizzazione.

3.4.1. Parametri riguardanti il moto

A causa della relativa lentezza del sistema a compiere una misura totale su 360° attorno al robot, si è pensato di permettere all'utente di configurare il sistema secondo le proprie necessità. La scheda Activation presenta i controlli necessari all'attivazione o disattivazione del servomotore e di uno dei due sensori.

- Disabilitando il tasto Servomotor, si annulla la rotazione del servomotore tenendolo sempre fisso nella posizione di partenza (un sensore rivolto verso il davanti e uno verso il retro del robot). Le misure dei sensori, non più in movimento, risultano più accurate, ma nel complesso l'efficienza del sistema risulta notevolmente degradata, poiché il campo visibile è di soli 10° avanti a sé e 10° dietro.
- Una maggior efficienza si può ottenere attivando solo il sensore anteriore attraverso il radio button IR Sensor e tenendo attivo il servomotore. La maggior parte delle volte si è, infatti, interessati agli ostacoli che il robot può incontrare di fronte a sé piuttosto che posteriormente. Questo permette una maggior velocità di spostamento del servomotore che non deve più attendere la lettura di entrambi i sensori, ma di uno solo.

La scheda Motion del frame, invece, permette la configurazione vera e propria del movimento del servomotore. Come si può vedere dalla Figura 4.2, si agisce su due parametri in particolare: il range che si vuole coprire e il passo di spostamento entrambi in gradi, modificabili attraverso i controlli Motion Setting.

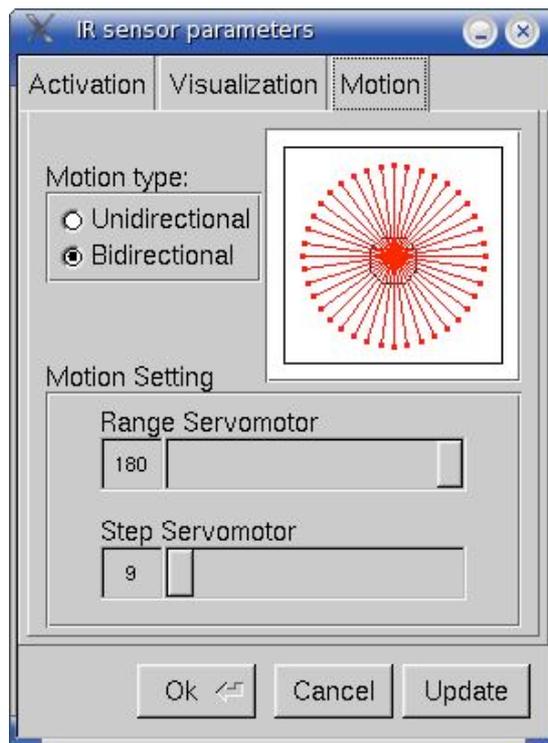


Figura 4.2. Scheda "Motion"

Le modifiche di tali parametri vengono visualizzate in anteprima nell'immagine per dare un'idea di quali saranno i movimenti del servomotore come è illustrato in Figura 4.3 (a) e (b).

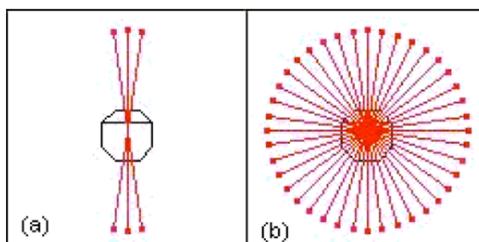


Figura 4.3. (a) Campo visivo minimo (b) Campo visivo massimo

Aumentando lo step e riducendo il range si può aumentare la velocità del sistema nel coprire tutto il campo selezionato a scapito di una minor definizione.

È stato, inoltre, inserito un terzo parametro, probabilmente di scarsa utilità, che setta il comportamento del servo motore una volta giunto ai fine corsa.

- Con One way si intende che il servomotore, una volta arrivato nell'ultima posizione, torna alla prima posizione per poi ricominciare le misure.
- Con Double way si intende che il servomotore, una volta arrivato ai fine corsa, continua le misure invertendo il senso di marcia.

3.4.2. Parametri riguardanti la visualizzazione

Come già detto in precedenza le misure dei sensori ad infrarossi vengono visualizzate in modo simile alle misure dei sonar. Distinguerle può risultare difficile. Si è pensato quindi di dare all'utente anche la possibilità di visualizzare le letture dei sensori ad infrarossi con differenti segni grafici e colori. Dalla

scheda Visualization mostrata in Figura 4.4 è possibile cambiare il colore selezionandone uno tra quelli possibili.

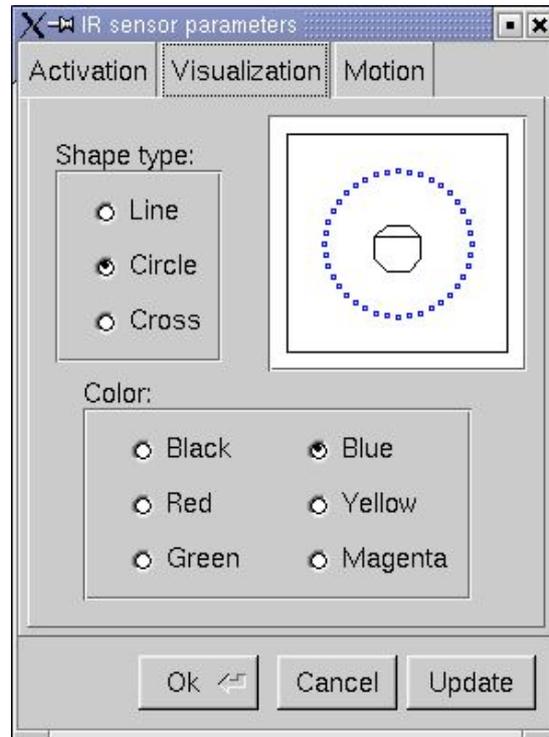


Figura 4.4 Scheda “Visualization”

È inoltre possibile selezionando una delle voci tra Line, Circle, Cross cambiare il segno grafico:

- Circle e Cross disegnano un cerchio o una croce in corrispondenza di ogni posizionamento del servomotore lungo tutto il campo di visibilità ad una distanza proporzionale alla lettura del sensore (Si veda Figura 4.5 (a), (b)).
- Line disegna, invece, una linea dal centro verso unicamente il posizionamento corrente del servomotore (Si veda Fig 4.5 (c)).

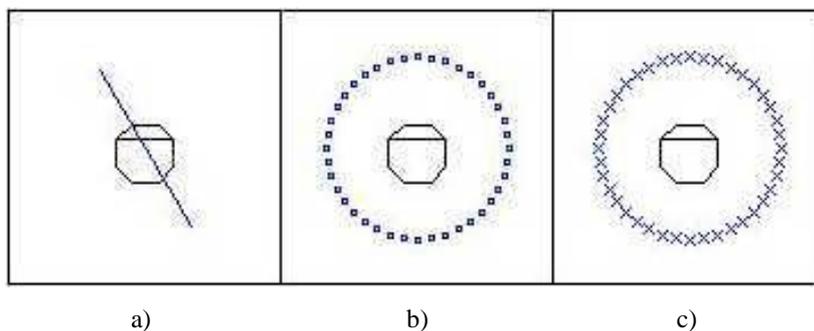


Figura 4.5 Segni grafici che possono essere impostati per le letture dei sensori IR

4. Conclusioni e sviluppi futuri

A causa della lentezza della comunicazione tra Saphira e il PSOS e della conseguente lentezza del movimento del servomotore, le prestazioni del sistema ad infrarosso per la rilevazione di ostacoli non sono troppo elevate. Ciò non è dovuto tanto all'imprecisione delle letture dei sensori, ma quanto al ritardo del sistema nell'aggiornarsi in seguito ad un cambiamento dell'ambiente esterno. Come già accennato in precedenza, per far fronte a tale limitazione il software dà la possibilità all'utente di impostare e attivare alcune proprietà del servomotore e dei sensori in modo da ottimizzarne l'esecuzione a seconda delle proprie esigenze. Un possibile sviluppo futuro potrebbe essere quello di contrastare il ritardo della propagazione dei cambiamenti dell'ambiente circostante via software:

- Si potrebbe far interagire le misure dei sensori ad infrarosso con quelle dei sonar, dispositivi, forse, più imprecisi, ma che forniscono una visione del mondo esterno in modo quasi istantaneo. Questi ultimi, vedendo a distanze maggiori, potrebbero essere utilizzati inizialmente per avere un'idea vaga della locazione degli oggetti circostanti e per guidare il robot verso la destinazione desiderata. I sensori ad infrarosso dovrebbero essere utilizzati per una scansione più precisa.
- Una seconda soluzione potrebbe essere quella di monitorare costantemente il movimento del robot. Sarebbe opportuno, poi, stimare i cambiamenti dell'ambiente esterno e, in base a questi ultimi, aggiornare in modo statistico il vettore contenente le misure delle distanze come fosse una previsione futura di quello che poi vedrebbero in realtà i sensori una volta giunti in posizione.

Nel caso in cui si volessero aggiungere altri dispositivi analogici agli ingressi del multiplexer, è possibile utilizzare come ulteriore bit di controllo il piedino 25 della General Port.

Appendice A

Questa appendice riporta delle considerazioni riguardanti i sensori Sharp GP2Y0A02YK.

Alcune di tali considerazioni ci sono state fornite da altri studenti che in precedenza avevano già analizzato il funzionamento della famiglia dei sensori analogici. Essi, infatti, avevano caratterizzato la funzione non lineare di *tensione in uscita - distanza dall'ostacolo* al variare del tipo di superficie. Tale curva viene riportata in Figura A.1.

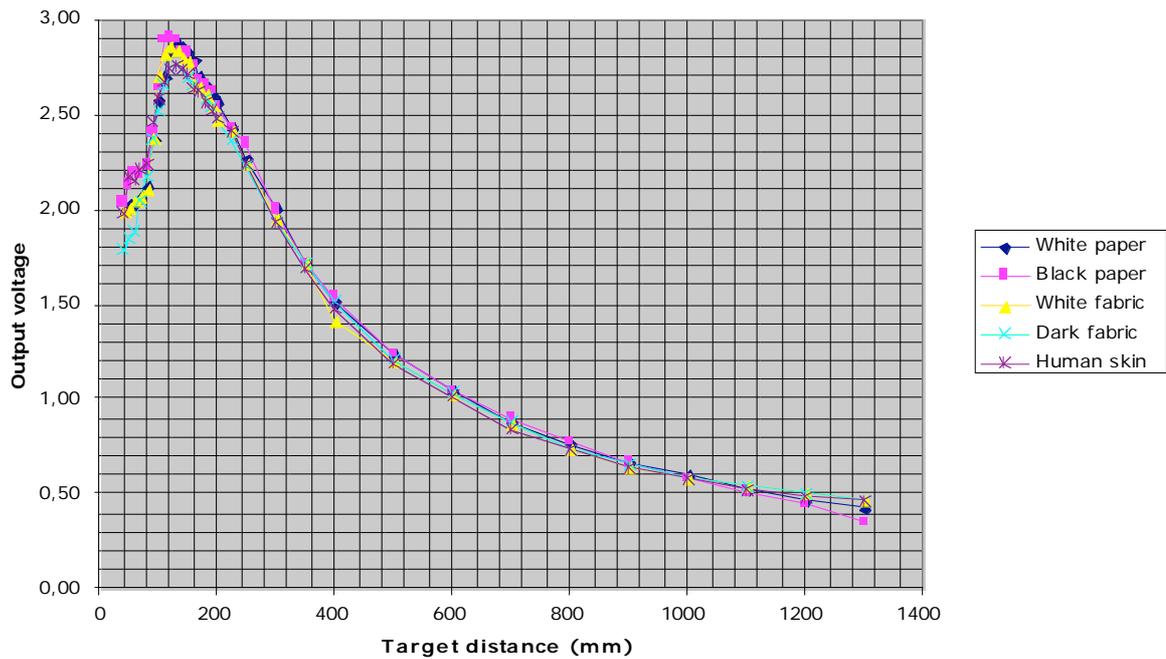


Figura A.1 Grafico tensione di uscita-distanza dall'oggetto.

L'analisi sperimentale ha portato a stabilire che la tensione di uscita del sensore varia a seconda del colore e del tipo di materiale dell'oggetto rilevato in modo trascurabile. Per questo motivo ci siamo limitati ad analizzare il caso di una superficie cartoncata. I risultati ottenuti sono riepilogati in tabella A.1. e rappresentati nel grafico riportato in figura A.2.

Distanza(cm)	Superficie cartonata (V)
150	0,43
140	0,45
130	0,5
120	0,53
110	0,61
100	0,65
90	0,71
80	0,75
70	0,9
60	1
50	1,2
40	1,5
35	1,7
30	1,95
25	2,25
20	2,5

Tabella A.1. Prove sperimentali effettuate con superficie cartonata.

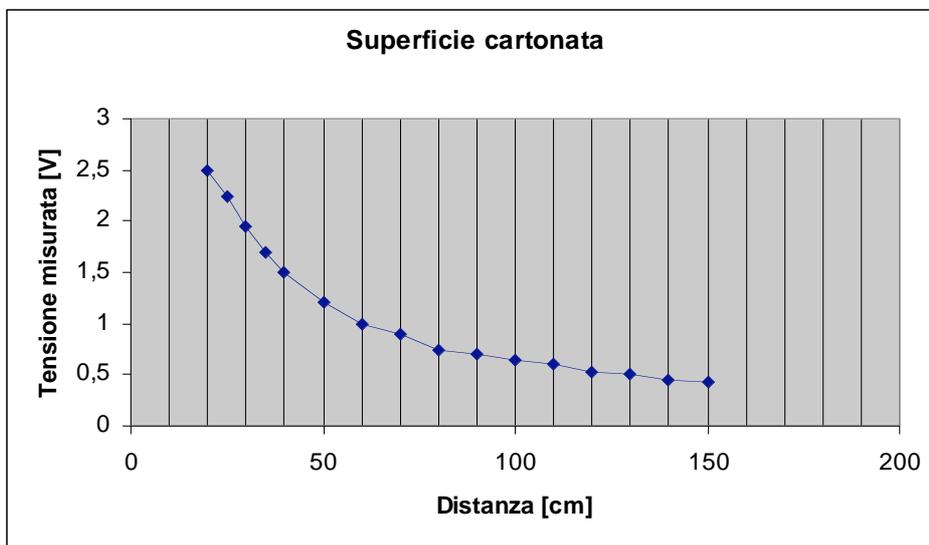


Figura A.2. Grafico riassuntivo tensione di uscita-distanza dall'oggetto per superficie cartonata

Per quanto riguarda altri aspetti del sensore ci siamo soffermati soprattutto sul cono nel quale i sensori riuscivano ad individuare un ostacolo infatti per noi era una caratteristica importante da tenere in considerazione dato che dovevano girare su un servomotore e riuscire a coprire un angolo di 360 gradi per darci una visione del mondo tutto intorno al robot che li avrebbe montati.

Le nostre prove sperimentali hanno condotto ad individuare un cono di circa 10 gradi notando anche che a seconda dell'inclinazione della superficie dell'ostacolo individuato si ha una variazione della misura di distanza rilevata di almeno il 10%.

Il sensore fornisce una lettura stabile della distanza dopo 32 millisecondi.

Appendice B

La seguente appendice è dedicata all'analisi di una parte del codice da noi proposto per il funzionamento del sistema ad infrarosso. Esso è scritto interamente in C++ e, se compilato con la vecchia versione di gcc 2.95.3, è eseguibile come shared object in ambiente Saphira. Il software consiste in più file:

IRSensor.cpp e il suo rispettivo header *IRSensor.h*

IRSensorGui.cpp e il suo rispettivo header *IRSensorGui.h*

In questa sezione ci soffermeremo esclusivamente sul primo file contenente il corpo principale del codice.

Nel file *IRSensor.h* viene definita la classe *SystemProcess* come sottoclasse di *SfUTask* e di *SfArtifact*.

La classe *SfUTask* è la classe di Saphira che si occupa della creazione di un task sincrono, il cui metodo *process* viene richiamato ogni 100 millisecondi. Tale classe permette la creazione di un thread autonomo per la gestione di tutto il nostro sistema. La classe *SfArtifact*, invece, serve per la gestione della grafica di Saphira. Ad ogni ciclo viene richiamato il metodo *draw* e vengono rappresentate attraverso simboli le distanze lette dai sensori.

Qui di seguito sono descritti alcuni metodi di interesse per il funzionamento del sistema.

B.1. Il metodo void process (void)

Tale metodo costituisce il cuore di tutto il software da noi creato. Come già accennato viene richiamato ogni 100 millisecondi e gestisce l'alternarsi delle letture dei sensori ai movimenti del servomotore. La successione delle operazioni viene gestita dalla struttura *switch* basata sul controllo della variabile *opCod*.

```
switch (opCod)
```

```
{
    case 0: ...
        break;
    case 1: ...
break;
    case 2: ...
break;
}
```

Nei casi in cui *opCod* sia 0 o 1 si provvede alla lettura rispettivamente dei sensori anteriore o posteriore. Viene letto il valore presente alla porta analogica e attraverso la funzione *convertDistance(int value)* viene convertito nel valore di distanza. Tale risultato viene, poi, inserito nell'array nel quale sono memorizzate tutte le letture in base all'angolo corrente del servomotore contenuto nella variabile *currentServoAngle*.

```
value = SfROBOT->getAnalog();
currentReadingValue[opCod] = convertDistance(value);
index=(int)((((currentServoAngle- firstServoAngle)/data.degreeIncrement) + (stepNumber opCod));
vectorReadingValue[index]= currentReadingValue[opCod];
```

Infine, a seconda delle impostazioni correnti , viene decisa la successiva operazione:

Si comanda al multiplexer analogico di commutare in modo da poter leggere l'altro sensore il ciclo successivo:

```
SfROBOT->com2Bytes(ArCommands::DIGOUT,0x08,0x08);
```

oppure il servo passando nell'istruzione la posizione desiderata:

```
SfROBOT->com2Bytes(ArCommands::PTUPOS,0,convertCurrentServoAngleToDuty());
```

Nel caso in cui *opCod* sia 2 si provvede ad aggiornare l'angolo corrente del servomotore, in base al range e all'incremento memorizzato nelle impostazioni correnti.

```
if ((currentServoAngle == lastServoAngle) || (currentServoAngle == firstServoAngle)) {
    currentIncrement = (-1)* currentIncrement;
    currentServoAngle += currentIncrement;
} else {
    currentServoAngle += currentIncrement;
}
```

B.2. Le funzioni `convertDistance (int analogInput)` e `int getPointOnLine (float x1, float x2, int y1, int y2, float xp)`

La funzione *convertDistance* si occupa di convertire il valore analogico letto compreso tra 0-255 in un valore di distanza. E' stato necessario costruire un'approssimazione lineare diversa per ciascun sensore, poiché le corrispettive funzioni caratteristiche non erano perfettamente identiche.

La funzione *getPointOnLine* si occupa di trovare l'ordinata del punto di interesse sul tratto di retta che in un determinato range approssima la funzione caratteristica inversa del sensore.

B.3. La funzione `int convertCurrentServoAngleToDuty()`

Tale funzione si occupa della conversione dell'angolo corrente nel valore della durata dell'impulso che deve essere dato in ingresso al servomotore per il corrispondente spostamento.

```
int SystemProcess::convertCurrentServoAngleToDuty()
{
    return (int)(currentServoAngle*(-200/180) + 250);
}
```

Bibliografia

- [1] Data sheet Sensori ad Infrarosso GP2Y0A02YK,
www.makingthings.com/products/acc_datasheets/data_pdfs/acc_ir_dist_001.pdf
- [2] Data sheet Standard Servo R29-MX-400
www.acroname.com/robotics/parts/R29-MX-400.html
- [3] Data sheet Stabilizzatore LM7805
www.fairchildsemi.com/ds/LM/LM7805.pdf
- [4] Konolige, K.: "Saphira Software Manual", Saphira Version 8 (Saphira/Aria integration), 2001
www.ing.unibs.it/~cassinis/Dida/robb/Documentazione/saphira-manual.pdf
- [5] Konolige, K.: "SAPHIRA TUTORIAL", Compiling, Loading and Debugging C++ Files: Unix Systems Software version 8.0 (Saphira/Aria), September 2001
www.ing.unibs.it/~cassinis/Dida/robb/Documentazione/actions.pdf
- [6] Dimitri van Heesch: "Saphira Reference Manual 8.2.0", February 2003
www.ing.unibs.it/~cassinis/Dida/robb/Documentazione/Saphira-Reference.pdf
- [7] Dimitri van Heesch: "Aria Reference Manual 1.2.0", February 2003
<http://www.ing.unibs.it/~cassinis/Dida/robb/Documentazione/Aria-Reference.pdf>
- [8] Konolige, K.: "Colbert User Manual", Software version 8.0a (Saphira/Aria), September 2001
<http://www.ing.unibs.it/~cassinis/Dida/robb/Documentazione/colbert-user.pdf>
- [9] Konolige, K.: "COLBERT: A Language for Reactive Control in Saphira", February 2003
<http://www-user.tu-chemnitz.de/~stj/lehre/colbert.pdf>
- [10] Activ Media Robotics: "Pioneer Mobile Robot Operation Manual, Edition 2", January 1998
<http://www.ing.unibs.it/~cassinis/Dida/robb/Documentazione/opmanv2.pdf>
- [11] Activ Media Robotics: "Gripper & Experimenter's Module Manual"
<http://www.ing.unibs.it/~cassinis/Dida/robb/Documentazione/gripman.pdf>

Indice

SOMMARIO	1
1. INTRODUZIONE	1
1.1. Analisi dei dispositivi del nostro sistema	1
1.1.1. Sensori di distanza ad infrarosso.....	1
1.1.2. Servomotore	2
1.2. Caratteristiche tecniche di Tobor	2
2. IL PROBLEMA AFFRONTATO	5
2.1. Una prima proposta	5
2.2. Analisi dei malfunzionamenti	5
3. LA SOLUZIONE ADOTTATA	6
3.1. Testing del sistema	7
MODALITÀ OPERATIVE	11
3.2. Componenti necessari	11
3.3. Modalità di installazione	12
3.3.1. Installazione Hardware	12
3.3.2. Installazione Software	12
3.4. Funzionalità offerte	12
3.4.1. Parametri riguardanti il moto	13
3.4.2. Parametri riguardanti la visualizzazione	14
4. CONCLUSIONI E SVILUPPI FUTURI.....	16
APPENDICE A	17
APPENDICE B	20
B.1. Il metodo void process (void)	20
B.2. Le funzioni convertDistance (int analogInput) e int getPointOnLine (float x1, float x2, int y1, int y2, float xp)	21
B.3. La funzione int convertCurrentServoAngleToDuty()	21
BIBLIOGRAFIA	22
INDICE	23