Group
Fabio Airoldi, Alberto Calloni, Guido Bonomi

Date
31-05-2009

Feasibility Study of "1-2-3 Star!" Game
RoboGame Design -3-

Indice

Prestazioni Meccaniche Robot

E' necessario che il robot sia in grado di muoversi con velocità paragonabili a quelle dei bambini che lo utilizzaranno come compagno di giochi.

Non e' stato tuttavia possibile reperire sul web le specifiche meccaniche relative a spykee. Tuttavia dall'osservazione dei video su AirWiki le prestazioni del robot sembrano adeguate al gioco di 1-2-3 stella.

In particolare si può notare che la velocità di rotazione appare abbastanza comparabile con quella di un bambino, anche grazie al fatto che ruotando in maniera opposta i due cingoli sostanzialmente si raddoppia la velocità di rotazione data dal singolo cingolo.

In ogni caso, se in via sperimentale si dovessero presentare problemi in ambito di velocità di rotazione, si potrebbe comunque pensare a far iniziare la rotazione del robot prima del termine della frase "1, 2, 3, Stella!".

Tale supposizione appare ragionevole per i seguenti due motivi:

1) La rilevazione del movimento da parte del robot diviene comunque attiva solo al termine della frase, perciò non si ha l'effetto di "aiutare" il robot in anticipo nel capire da dove proviene effettivamente il suono, non intaccando le regole di gioco.

2) Anche l'umano, soprattutto visto il target preposto che suppone di avere un interazione con bambini, si volta prima del termine della frase, approssimativamente mentre si pronuncia la parola "Stella!", quindi si renderebbe comunque il comportamento del robot simile a quello dei bambini con cui interagisce.

Per quanto riguarda la velocità e l'accelerazione, dai video appare piuttosto adeguata al tipo di movimento richiesto dal gioco proposto, così come i tempi di arresto del robot.

Hardware di controllo

E' possibile realizzare l'hardware di controllo in 3 modalita':

1. Utilizzare un pc per controllare spykee attraverso la connessione WiFi.
2. Dotare spykee di una board precostruita programmabile (es. Armadillo). Anche in questo caso la comunicazione tra la board e il robot avverrebbe comunque via WiFi (o eventualmente via USB)
3. Programmare direttamente l'hardware di controllo di spykee.

Queste tre opzioni sono in ordine di difficoltà ed eleganza.
In particolare essendo Spykee dotato di una board con processore ARM a 200Mhz su cui gira una versione custom di linux, dovrebbe essere possibile caricare su di

esso dei semplici eseguibili (compilati per architettura arm) ed avviarli direttamente al boot.

Specifiche di Spykee:

- CPU ARM9 @ 200Mhz
- 32Mb SDRAM
- 4Mb NOR Flash
- WiFi client 802.11b/g
- USB host
- 8-bits µC for dual DC motor control and battery control
- 4 leds and 1 flash led
- Mono speaker, 16bit audio out (at 48kHz)
- Microphone 8bit at 16kHz
- QVGA CMOS camera 320 x 240 at 15fps in local mode (optimum conditions) / 4 to 15 fps in distant mode (optimum conditions)
- IR Receptor
- 9,6V NiMH battery charging control -
- JTAG / Serial / i2c port available. Joystick Compliant
- Micro with electret
- Loudspeaker 8 Ohm 2W
- Open source software

Esistono già alcuni esperimenti in tal senso realizzati tuttavia a livello amatoriale.

Per ulteriori approfondimenti si consideri i seguenti link:

http://www.spykeedev.net/index.php?option=com_agora&task=topic&id=52&p=&

http://www.spykeedev.net/index.php?option=com_agora&task=topic&id=51&p=#p543

Interazione WiiMote

Nel gioco proposto l'interazione con i WiiMote si limita al riconoscimento di accelerazioni e alla riproduzione di suoni e vibrazioni.

Queste funzionalita' sono rese possibili dalla libreria opensource per linux *libWiiMote*.

In particolare tale libreria permette:

Inputs:

- Read accelerometer data from the wiimote (0x31, 0x33, 0x35, 0x37).
- Read IR-sensor data (0x32, 0x33, 0x36, 0x37).
- Read key states (0x30-0x37).

- Read battery status (0x20).
- Read nunchuk key, accelerometer and joystick states (0x34-0x37).
- Read classic controller key and joystick states (0x34-0x37).

Outputs:

- Set/unset the wiimote LEDs (0x11).
- Enable/disable force-feedback effect (0x11).
- Play sounds on the built-in speaker (0x18).

Per ulteriori approfondimenti si consideri il link:

http://libwiimote.sourceforge.net/

Sensore ad Infrarossi

Nella parte anteriore del Wiimote e' presente un sensore PixArt, che permette di tracciare la posizione relativa di LED ad infrarossi.
Grazie al funzionamento del sensore, paragonabile a quello di una normale telecamera "pinhole" che però rileva solamente onde elettromagnetiche nel campo dell'infrarosso, è possibile riconoscere sorgenti luminose infrarosse in configurazioni predefinite.
Questo metodo non solo evita di utilizzare una grande banda per trasmettere l'intera immagine rilevata dalla telecamera, ma poiché in condizioni normali i LED risultano l'unica fonte intensa di infrarossi, evita anche di utilizzare una qualunque tecnica per l'identificazione di marker per estrapolare le coordinate sul piano immagine della barra.
Bisogna, tuttavia, notare che, in alcune situazioni particolari, ad esempio inquadrando lampadine ad incandescenza, i LED non risultano più l'unica sorgente di infrarossi rilevata e questo può risultare problematico.
La larghezza di campo è di circa 45 gradi in orizzontale e 35 in verticale, la risoluzione spaziale è di
1024x768, la frequenza di scansione è 100 Hz. Il sensore invia, inoltre, un'informazione (su una scala da 1 a 15) che indica, in modo approssimativo, l'intensità con cui un LED è rilevato.
Queste caratteristiche si prestano bene al tipo di applicazione richiesta per il gioco "1, 2, 3, Stella!", grazie alla capacità di riconoscimento dei marker, e ad una larghezza di visione del campo sufficiente.

Accelerometro

Nel Wiimote è presente un accelerometro a 3 assi ADXL330, con un range minimo di +/- 3g su ogni asse e una sensibilità del 10%.
Questo permette il riconoscimento dei movimenti richiesto dal tipo di gioco proposto, senza particolari problemi di inaccuratezza.

Sensore di posizione

Grazie ai sensori di posizione presenti oggi in commercio, non vi sono particolari problematiche nell'individuazione della posizione del muro e dei giocatori da parte del robot.
La distanza entro cui questi sensori rilevano oggetti è definita portata nominale (o campo sensibile). Alcuni modelli dispongono di un sistema di regolazione per poter calibrare la distanza di rilevazione.

L'assenza di meccanismi d'attuazione meccanica, e di un contatto fisico tra sensore e oggetto, fa sì che questi sensori presentino un'affidabilità elevata.
In particolare, l'affidabilità e la precisione dipendono dalla tipologia di sensori di prossimità utilizzati.

Attualmente, in commercio, si possono trovare proximity sensors realizzati su diversi tipi di tecnologie:

- a sensori induttivi;
- a sensori capacitivi;
- a sensori magnetici;
- a sensori ad ultrasuoni;
- a sensori ottici.

La precisione del sensore richiesto può essere poi appurata per analisi sperimentale.

Riconoscimento suono "1-2-3 stella"

Nel gioco il robot deve poter riconoscere la frase "1-2-3 stella" per individuare gli intervalli di tempo nei quali può muoversi e nei quali deve stare fermo.

Un possibile modo di realizzare il riconoscimento vocale di tale frase e' quello di appoggiarsi alla libreria opensource pocketsphinx.

Tale libreria permette di effettuare riconoscimento vocale continuo indipendente dallo speaker ed e' pensata per il deployment su dispositivi embedded (la piattaforma target per la quale la libreria era stata sviluppata era sharp zaurus con cpu arm @ 209 Mhz e 64Mb sdram).

Con pocketsphinx vengono distribuiti degli script di esempio, uno dei quali (pocketsphinx_tidigits) si occupa del riconoscimento vocale delle cifre decimali (0-9).

Abbiamo effettuato dei test con tale script pronunciando le cifre "*one two three four*" con varie tonalità di voce, inizialmente senza rumore e successivamente con della musica in sottofondo.

E' stato utilizzato un semplice microfono piezoelettrico preamplificato per PC.

Dai test effettuati il riconoscimento sembra abbastanza preciso e tutto sommato resistente al rumore di fondo.

Per migliorare ulteriormente il riconoscimento si potrebbe aggiungere un filtro passa-banda (che per esempio consideri l'intervallo di frequenze comprese tra 1KHz - 4Khz) per filtrare solo le voci umane.

Per ulteriori approfondimenti:

http://cmusphinx.sourceforge.net/html/cmusphinx.php