

UNIVERSITÀ DEGLI STUDI DI MODENA E
REGGIO EMILIA

DIPARTIMENTO DI SCIENZE E METODI DELL'INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA MECCATRONICA

SVILUPPO DI UNA PIATTAFORMA PER IL
CONTROLLO E LA TELEOPERAZIONE DI UN
GRUPPO DI ROBOT MOBILI

Relatore: Prof. Cristian Secchi Laureando: Filippo Bertoncelli
Correlatore: Prof. Lorenzo Sabattini

ANNO ACCADEMICO 2014/2015

*Ai mie genitori
e a Giovanna*

Indice

Introduzione	1
1 Struttura	3
1.1 ROS	3
1.2 Robot Mobili e-Puck	4
1.2.1 Sensori	4
1.2.2 Ruote	5
1.2.3 Controllore	5
1.2.4 Connessioni	5
1.3 Arena	7
2 Controllo dei Robot	9
2.1 Comunicazione	9
2.2 Modellizzazione	9
2.3 Accorgimenti	11
3 Odometria	13
3.1 Calibrazione della fotocamera	13
3.2 Software di riconoscimento: ArUco	15
3.2.1 Markers	15
3.2.2 Boards	15
3.3 Compatibilità con ROS	15
3.3.1 TF	15

4	Controllo in Formazione e Teleoperazione	17
4.1	Formazione	17
4.2	Teleoperazione	17
A	Manuale Tecnico	19
A.1	Setup	19
A.2	Troubleshooting	19
	Bibliografia	21

Introduzione

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

2 paginette giusto per spiegare.

Si noti che questo capitolo viene ottenuto in maniera differente dagli altri. In particolare non viene numerato.

Capitolo 1

Struttura

In questo capitolo verranno descritte le apparecchiature utilizzate per il progetto.

1.1 ROS

ROS, acronimo per Robot Operating System, è un insieme di librerie software e strumenti che facilitano la creazione di applicazioni orientate alla robotica; inoltre è completamente open source in modo da incentivare la collaborazione tra i vari utilizzatori.

ROS è ufficialmente supportato solo sul sistema operativo Ubuntu, ma fornisce anche supporto a svariate piattaforme hardware come manipolatori, robot mobili e robot umanoidi.

In ROS ogni applicazione viene definita nodo; ogni nodo comunica con altri nodi attraverso tre protocolli: Topic, Service e Action.

I Topic sono dei canali di comunicazione con cui i nodi possono scambiarsi messaggi; quando un nodo pubblica un messaggio su un topic tutti i nodi sottoscritti a tale topic vengono notificati della presenza di un messaggio ed eseguono determinate operazioni utilizzando le informazioni fornite dal messaggio.

Il concetto di Service si basa sulla necessità di un sistema distribuito di avere delle interazioni richiesta/risposta; tramite un Service un nodo può richiedere

che un altro nodo esegua un determinato compito e comunichi al richiedente un eventuale risultato.

Il protocollo di Action è fondamentalmente uguale a quello di Service, ma Ã stato esteso in modo da fornire al richiedente la possibilit  di annullare la richiesta nel caso in cui questa impieghi un tempo eccessivo per essere portata a termine.

Questi tre protocolli consentono di realizzare applicazioni per qualunque necessit  in ambito di robotica.

1.2 Robot Mobili e-Puck

I robot mobili e-Puck sono dei robot mobili a due ruote di dimensioni compatte dotati di una grande variert  di sensori e possono essere usati in svariate applicazioni che spaziano dalla robotica mobile a sistemi biologici e di intelligenza artificiale.



Figura 1.1: Robot e-Puck

1.2.1 Sensori

Gli e-Puck sono dotati di 8 sensori di prossimit , un accelerometro 3D, 3 microfoni omni-direzionali, una fotocamera VGA e uno speaker

1.2.2 Ruote

Le ruote sono coassiali e comandate separatamente da due motori passo-passo attraverso un riduttore meccanico con rapporto di trasmissione 1/50.



Figura 1.2: Motore passo-passo con riduttore

1.2.3 Controllore

Il sistema è comandato da un microcontrollore dsPIC con clock a 60MHz e un DSP per processare i segnali.



Figura 1.3: Motore passo-passo con riduttore

1.2.4 Connessioni

Il micro controllore fornisce due connessioni uart; la prima collegata a un chip Bluetooth e la seconda disponibile fisicamente attraverso un connettore micro-match. La connessione Bluetooth avviene attraverso un chip LMX9820A prodotto da National Semiconductor; questo chip fornisce uno stack Bluetooth completo e compatibile con la versione 1.1. Per facilità di utilizzo si è scelto di utilizzare la connessione senza fili per comunicare con i robot, in particolare si è scelto il protocollo RFCOMM, tale protocollo è usato per generare un flusso virtuale di dati seriali, simulando dunque i segnali di controllo dello standard RS232 fornendo una connessione seriale fittizia.

La connessione a lato computer avviene utilizzando due adattatori Bluetooth Logilink compatibili con lo standard 4.0 e retrocompatibili, dunque adatti ad essere utilizzati con i robot scelti. La tecnologia Bluetooth fornisce un indirizzo univoco a 3 bit ad ogni dispositivo collegato alla rete, siccome un dispositivo deve essere denominato master (nel nostro caso gli adattatori) rimangono solo 7 indirizzi disponibili per altri dispositivi, non sufficienti a connettere contemporaneamente tutti e 8 i robot disponibili; per questo motivo si è scelto di utilizzare due adattatori, creando così due sottoreti ed estendendo il numero di connessioni simultanee di robot al computer fino a 14. L'utilizzo di due adattatori fornisce inoltre il vantaggio di eliminare il collo di bottiglia provocato dalla limitata velocità di trasmissione di una singola antenna Bluetooth.



Figura 1.4: Schema delle connessioni Bluetooth

1.3 Arena

Tutte le simulazioni sono effettuate all'interno di un'arena composta da un piano, sul quale i robot si possono muovere, e da una struttura metallica a forma di L.



Figura 1.5: Arena con la struttura di supporto per la webcam

Per sostenere una webcam Logitech QuickCam Pro 9000 dotata di un sensore con risoluzione di 2.0 Megapixel con il compito di acquisire le immagini dei robot in movimento sul piano e consentire al computer di elaborare le immagini per rilevare la posizione dei singoli robot sull'arena. La webcam viene collegata al computer tramite un cavo USB 2.0

Capitolo 2

Controllo dei Robot

Questo capitolo tratta tutte le nozioni e accorgimenti utilizzati per la creazione del nodo di comunicazione con i robot e-Puck

2.1 Comunicazione

Come già anticipato nel capitolo precedente, la comunicazione con i robot avviene tramite una connessione seriale virtuale attraverso la quale vengono scambiati messaggi sotto forma di array di caratteri.

Il firmware precaricato negli e-Puck fornisce una modalità di interazione tramite connessione seriale nella quale è possibile accedere ai valori misurati da tutti i sensori e impartire vari comandi. In particolare inviando “D,*fl*,*fr*\n” è possibile impostare la frequenza di passo dei motori sinistro e destro rispettivamente ai valori di *fl* e *fr*. I valori accettabili per le due frequenze sono i numeri interi compresi tra -1000 e 1000, frequenze alle quali corrisponde una rotazione completa indietro o avanti della ruota in un secondo.

2.2 Modellizzazione

I robot utilizzati possono essere classificati come dei monocicli ad azionamento differenziale, l’assenza di una rotella di supporto non crea problemi in quanto lo

scheletro del robot permette la stabilità e consente il movimento.



Figura 2.1: Modello schematico di un monociclo ad azionamento differenziale

Secondo questo modello il robot può avere due distinte componenti di velocità: una di traslazione v , ortogonale all'asse delle due ruote e parallela al piano di spostamento, e una di rotazione ω ortogonale al piano di spostamento del robot nel punto medio della distanza tra le due ruote.

Da v e ω possono essere stabilite le velocità da assegnare alle singole ruote ωr e ωl tramite le seguenti relazioni in cui r è il raggio delle ruote e d la distanza tra i centri delle ruote:

$$\begin{cases} v = \frac{(\omega r + \omega l)r}{2} \\ \omega = \frac{(\omega r - \omega l)r}{d} \end{cases} \rightarrow \begin{cases} \omega r = \frac{2v + d\omega}{r} \\ \omega l = \frac{2v - d\omega}{r} \end{cases}$$

Per ricavare la frequenza di passo delle ruote si moltiplica i valori ottenuti per $\frac{1000}{2\pi}$ che corrisponde al rapporto tra il numero di passi necessari per compiere una rotazione completa della ruota e una rotazione completa in radanti.

$$\begin{aligned} fr &= \frac{1000}{2\pi} \times \omega r \\ fl &= \frac{1000}{2\pi} \times \omega l \end{aligned}$$

Il nodo realizzato dunque riceve un messaggio tramite topic contenente v e ω e ricava le frequenze di passo da assegnare alle ruote. Questi valori sono inviati al robot tramite connessione seriale utilizzando il messaggio descritto nella sezione precedente. Inoltre se il nodo non riceve messaggi di velocità per due secondi invia al robot un comando per fermare le ruote ovvero imposta la frequenza di passo delle ruote a 0.

2.3 Accorgimenti

Per evitare eventuali problemi di connessione e assicurare che le connessioni siano ripartite correttamente tra i due adattatori Bluetooth è stato creato uno script con linguaggio bash che utilizza i comandi “rfcomm release” e “rfcomm bind” per associare a ogni adattatore quattro robot tramite gli indirizzi MAC dei vari dispositivi. Lo script viene lanciato digitando “./epuckbtset.sh” in un terminale che punta alla cartella contenente il file bash.

Capitolo 3

Odometria

La soluzione utilizzata per rilevare la posizione dei robot all'interno dell'arena include l'utilizzo di metodi di computer vision in grado di rilevare contemporaneamente sia la posizione che l'orientamento di marker grafici attraverso le immagini fornite dalla webcam.

L'acquisizione delle immagini dalla fotocamera avviene tramite il nodo ROS *usb_cam* che comunica con il sensore di immagini utilizzando il protocollo V4L (video for linux) e pubblica i fotogrammi ottenuti tramite *image_transport*, il protocollo standard per la trasmissione delle immagini in ROS.

3.1 Calibrazione della fotocamera

Per riconoscere in maniera ottimale i marker l'immagine proveniente dalla fotocamera deve essere rettificata per bilanciare la distorsione introdotta dalle lenti.

Per ottenere una rettificazione ottimale è necessario generare il file di calibrazione della fotocamera, questo file è ottenuto utilizzando l'applicazione ROS *camera_calibration* che calcola i parametri di distorsione.

La procedura di calibrazione implica l'utilizzo di una scacchiera con dimensioni ben definite

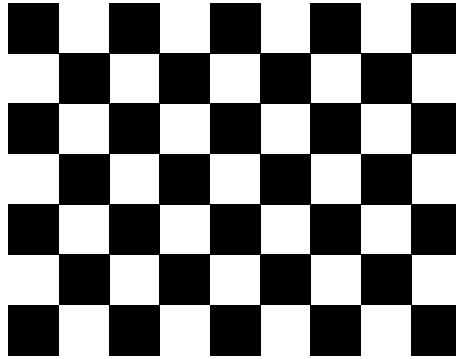


Figura 3.1: Scacchiera 8 x 6

Una volta avviata l'applicazione si procede all'acquisizione dei dati traslando e ruotando la scacchiera lungo l'intero campo visivo della fotocamera fino a quando il software non comunica di aver acquisito sufficienti informazioni per la calibrazione. Il software ora può calcolare i dati di calibrazione e salvarli in un file formato YAML.

La vera rettificazione dell'immagine avviene attraverso il nodo *image_proc* che processa l'immagine utilizzando i dati di calibrazione e restituisce l'immagine rettificata sempre utilizzando il protocollo di trasmissione di immagini di ROS.
inserire immagini di esempio immagini rettificate e non

3.2 Software di riconoscimento: ArUco

3.2.1 Markers

3.2.2 Boards

3.3 Compatibilità con ROS

3.3.1 TF

Capitolo 4

Controllo in Formazione e Teleoperazione

4.1 Formazione

4.2 Teleoperazione

Appendice A

Manuale Tecnico

A.1 Setup

A.2 Troubleshooting

Bibliografia