

POLITECNICO DI TORINO

III Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica

Relazione di progetto

Automazione Discreta



LUCA BELLUCCINI
PIERGIUSEPPE BETTASSA
MANUELE BOMMARITO
MATTEO CASALINO
DAVIDE FERRI
IVAN LAZZERO

ANNO ACCADEMICO 2008-2009

Indice

Sommario	II
1 Costruzione e meccanica del Robot	1
2 Il package Navigation	2
2.1 La classe LashMotor	2
2.2 L'interfaccia CheckersNavigator	2
2.2.1 La classe SimpleNavigator	3
2.2.2 La classe MathNavigator	4
2.3 La classe ArmController	4

Sommario

Questo è il sommario

Capitolo 1

Costruzione e meccanica del Robot

Capitolo 2

Il package Navigation

Il *package* `it.polito.Navigation` contiene le classi deputate a gestire i movimenti del Robot sulla scacchiera.

In particolare sono stati progettati due controllori: `CheckersNavigator` e `ArmController` che sono responsabili rispettivamente della navigazione nelle due direzioni orizzontali e in quella verticale.

Di seguito si entrerà nel dettaglio delle implementazioni delle classi sopra citate e dei relativi helper.

2.1 La classe LashMotor

2.2 L'interfaccia CheckersNavigator

Questa interfaccia astrae le funzionalità di movimento bidimensionale sulla scacchiera 8x8; i metodi più importanti sono:

```
/** Ritorna la posizione X [0; 7] */  
public int getX();  
/** Ritorna la posizione Y [0; 7] */  
public int getY();  
/** Muove il braccio sulla casella (X, Y) */  
public void goTo(int newX, int newY) throws NotCalibratedException;  
/** Muove il braccio sulla casella "base" */  
public void goHome() throws NotCalibratedException;  
/** Modifica la velocità dei motori */  
public void setSpeed(int speedA, int speedB);  
/** Effettua la calibrazione */  
public void calibrate();
```

Si noti come i metodi che comportano un movimento non possano essere eseguiti se prima non è stata effettuata la calibrazione (eccezione `NotCalibratedException`). Esamineremo ora le implementazioni che sono state progettate per questa interfaccia.

2.2.1 La classe `SimpleNavigator`

Questa prima implementazione si basa su un mapping statico di tutte le caselle relativo ad un punto iniziale su cui il Robot tenta di posizionarsi in fase di calibrazione.

Calibrazione Il Robot, per come è costruito, può ruotare il suo braccio agendo sul motore B, o può spostarlo avanti e indietro agendo sul motore A.

Il primo metodo di calibrazione che è stato pensato è pertanto semplicemente mirato a portare il braccio in una posizione nota (punto rosso in Figura 2.1), in modo da poter usare offset predeterminati (relativi ad essa) per spostarlo su tutte le altre caselle.

Il metodo `calibrate()` non fa altro che ruotare il braccio verso destra finché il

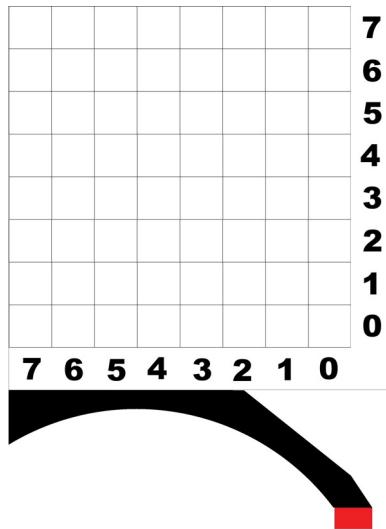


Figura 2.1. Scacchiera `SimpleNavigator`

senso di colore montato su di esso non rileva il colore rosso, a quel punto recupera eventuali giochi dei motori e azzeri i contatori di distanza angolare in essi contenuti.

Navigazione Gli angoli di destinazione cui vengono fatti ruotare i motori sono calcolati semplicemente come¹:

```
destAngleA = offA+posy [newY]+dely [newX];  
destAngleB = offB+posx [newX];
```

I vettori `posx` e `posy` definiscono l'offset in funzione delle rispettive coordinate; il vettore `dely` contiene le correzioni da effettuare sull'asse y in funzione della coordinata x, in modo da recuperare gli scostamenti in verticale dati dal fatto che il braccio si muove su un arco di circonferenza.

Le costanti `offA` e `offB` definiscono invece l'offset necessario a portare il braccio sulla casella (5,0) che costituiva il punto più comodo cui riferire la taratura di tutte le altre caselle.

2.2.2 La classe `MathNavigator`

Calibrazione

Navigazione

2.3 La classe `ArmController`

¹X e Y sono rispettivamente le coordinate di ascissa e di ordinata delle caselle