

**POLITECNICO DI MILANO**  
Corso di Laurea Specialistica in Ingegneria Informatica  
Dipartimento di Elettronica e Informazione



## **Sviluppo di un sistema di controllo basato su odometria per una carrozzina robotica**

**AI & R Lab**  
Laboratorio di Intelligenza Artificiale  
e Robotica del Politecnico di Milano

**Relatore: Ing. Matteo Matteucci**  
**Correlatore: Ing. Davide Migliore**  
**Correlatore: Ing. Simone Ceriani**

**Tesi di Laurea di:**  
**Marco Dalli, matricola 704380**

**Anno Accademico 2007-2008**



*A tutti quelli che avranno la pazienza di leggerla fino in fondo...*



# Sommario

Il presente lavoro si colloca nell'ambito della robotica mobile, disciplina che si occupa di studiare e progettare veicoli autonomi dotati di funzionalità "intelligenti", ovvero di caratteristiche che li rendono in grado di operare senza bisogno d'intervento umano o riducendo i comandi che l'uomo deve impartire. In particolare si sono applicate le tecniche della robotica mobile allo sviluppo di una carrozzella d'ausilio ai disabili motori, estendendo le funzionalità di una carrozzina elettrica al fine di studiare come avvicinarla alle esigenze degli utenti che trovano difficoltà nell'utilizzare i dispositivi esistenti. Oggetto di questo lavoro è stata la riprogettazione e la realizzazione di un circuito che, collegato alla scheda di controllo interna al joystick di comando, comunica al computer la posizione della leva del joystick e permette di controllare il movimento della carrozzina. Per introdurre il dato sulla velocità della carrozzina sono state analizzate varie alternative: dopo aver scartato la soluzione basata su un accelerometro, per via del suo elevato costo, si è preferito optare per una soluzione basata su sensori ottici. Sulla carrozzina sono stati quindi installati degli encoder per misurare la velocità delle ruote e, per estensione, della carrozzina stessa. È stato quindi progettato un secondo circuito, che si occupa di leggere i dati dai sensori e comunicarli al circuito di comando principale. La presenza di sensori odometrici ha poi consentito di estendere le funzionalità del circuito di comando, introducendo un sistema basato su controllori PID per realizzare il controllo in anello chiuso della velocità della singola ruota. A livello più alto, il software della carrozzina è stato modificato per affiancare il dato dell'odometria a quello di un sistema di visione artificiale, per migliorare la localizzazione in ambiente indoor e la ricostruzione della traiettoria della carrozzina. Le funzionalità sono state testate in ambienti semplici e controllati: il prototipo realizzato è risultato ben progettato e corrispondente agli obiettivi proposti.



# Ringraziamenti

Alla fine di un cammino lungo come quello che mi ha portato a scrivere questa tesi è giusto voltarsi indietro e vedere quante sono le persone a cui si deve dire grazie per l'aiuto e il supporto offerto. Innanzitutto devo ringraziare i Proff. Andrea Bonarini e Matteo Matteucci, senza i quali non conoscerei l'AIRLab e non sarei qui a scrivere queste righe. Un grazie speciale va anche all'Ing. Davide Migliore, per aver coordinato i lavori e all'Ing. Simone Ceriani, per aver contribuito allo sviluppo della carrozzina Lurch e per le innumerevoli volte che mi ha tirato fuori dal pantano. Si ringrazia Mario Ceriani, papà di Simone, per aver partecipato attivamente alla realizzazione del disco e dei mozzi. Grazie anche a tutti i ragazzi dell'AIRLab, in particolare all'Ing. Davide "Gigi" Rizzi, per aver sviluppato il codice per la porta seriale del PIC e per i mille consigli che è sempre pronto a dispensare, e all'Ing. Luigi Malagò, l'uomo più impegnato dell'universo assieme a Matteo, ma che nonostante questo riesce sempre a trovare il tempo per darmi una mano. Grazie a Martino, unico baluardo della stirpe degli elettronici in AIRLab e prodigo di consigli utili alla realizzazione di circuito, cosa cui noi poveri informatici siamo poco avvezzi.

Un grazie non può che andare alla mia famiglia, per avermi concesso l'opportunità di svolgere questo ciclo di studi e per avermi sempre fatto fare percorrere la strada che più desideravo. Grazie anche a tutti i miei Amici, quelli con la "A" maiuscola, per essermi sempre stati vicini.

Grazie infine a Leslie Lamport, per avere inventato  $\LaTeX$ , senza il quale questa tesi avrebbe portato via il doppio del tempo per essere scritta e impaginata.

Grazie anche a tutti quelli che ho dimenticato: a voi chiedo scusa, ma sono le 2 di notte e incomincio ad avere sonno. . .

Ora posso spegnere il PC e andare a letto. (Grazie Monkey Island)





# Indice

<b>Sommario</b>	<b>I</b>
<b>Ringraziamenti</b>	<b>III</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 Ausili alla mobilità e tecnologia</b>	<b>7</b>
2.1 Ausili alla mobilità . . . . .	7
2.2 Carrozze elettriche e robotica mobile . . . . .	8
2.2.1 Estensione delle funzionalità delle carrozze elettriche	9
2.2.2 La carrozina elettrica come robot mobile . . . . .	10
2.2.3 Funzionalità offerte . . . . .	10
2.2.4 Dispositivi di comando . . . . .	11
2.3 Progetti esistenti . . . . .	12
<b>3 Cenni di robotica mobile</b>	<b>19</b>
3.1 Localizzazione con odometria . . . . .	19
3.1.1 Errori dell'odometria . . . . .	20
3.2 Encoder . . . . .	21
3.2.1 Encoder ottici . . . . .	22
3.2.2 Encoder in quadratura . . . . .	23
3.3 Cinematica di un robot mobile . . . . .	24
3.3.1 Cinematica per Differential Drive . . . . .	25
3.3.2 Cinematica probabilistica . . . . .	28
3.4 Il filtro di Kalman . . . . .	28
3.4.1 Filtro di Kalman per sistemi lineari . . . . .	29
3.4.2 Filtro di Kalman esteso . . . . .	30
3.4.3 EKF per ricostruzione di traiettorie . . . . .	31
3.5 Regolatori PID . . . . .	33

<b>4</b>	<b>Progetto logico della soluzione del problema</b>	<b>35</b>
4.1	Carrozzina . . . . .	35
4.1.1	Scheda di controllo . . . . .	39
4.1.2	Software di controllo e sensori . . . . .	39
4.2	Analisi dei requisiti . . . . .	40
4.2.1	Obiettivi . . . . .	41
4.3	Encoder . . . . .	42
4.3.1	Sensore . . . . .	42
4.3.2	Disco . . . . .	43
4.4	Scheda di controllo . . . . .	44
4.4.1	Versione finale . . . . .	44
4.4.2	Dimensionamento del filtro passa-basso . . . . .	46
4.4.3	Isolamento tra i componenti . . . . .	47
4.5	Computer di bordo . . . . .	48
4.6	Evoluzione del sistema di controllo . . . . .	50
<b>5</b>	<b>Architettura del sistema</b>	<b>53</b>
5.1	Architettura del sistema . . . . .	53
5.2	Scheda master . . . . .	54
5.2.1	Hardware . . . . .	54
5.2.2	Software . . . . .	57
5.3	Scheda odometry . . . . .	62
5.3.1	Hardware . . . . .	62
5.3.2	Software . . . . .	64
5.4	Protocolli di comunicazione . . . . .	65
5.4.1	Protocollo di comunicazione master/odometry . . . . .	65
5.4.2	Protocollo di comunicazione master/PC . . . . .	65
5.5	Encoder . . . . .	69
5.5.1	Disco e mozzo . . . . .	69
5.5.2	Profilo di sostegno sensori . . . . .	72
5.6	Architettura MRT . . . . .	72
5.6.1	Implementazione preodometria . . . . .	72
5.6.2	Modifiche relative all'odometria . . . . .	74
5.6.3	Altre modifiche . . . . .	74
<b>6</b>	<b>Realizzazioni sperimentali e valutazione</b>	<b>77</b>
6.1	Schede . . . . .	77
6.1.1	Ritardi . . . . .	77
6.2	Encoder . . . . .	80
6.2.1	Quadratura degli encoder . . . . .	80

6.2.2	Calibrazione dell'odometria . . . . .	82
6.2.3	Calibrazione del controllore PID . . . . .	83
6.3	Test sperimentali . . . . .	83
<b>7</b>	<b>Conclusioni e sviluppi futuri</b>	<b>89</b>
7.1	Conclusioni . . . . .	89
7.2	Odometria . . . . .	90
7.3	Joystick . . . . .	91
7.3.1	Joystick come dispositivo di interfaccia . . . . .	91
7.4	Guida in ambienti esterni . . . . .	91
7.5	Controllo della carrozzina tramite BCI . . . . .	92
	<b>Bibliografia</b>	<b>94</b>
	<b>A Datasheet</b>	<b>99</b>



# Capitolo 1

## Introduzione

*“Finché dal mezzo di queste tenebre una luce improvvisa mi illuminò, una luce così brillante e portentosa eppure così semplice: cambiare i poli da positivo a negativo e da negativo a positivo. . . Io solo sono riuscito a scoprire il segreto di infondere la vita, macché, anche di più: io, proprio io, sono divenuto capace di rianimare nuovamente la materia inanimata. . .  
Sì. Può. Fare!”*

Frederick Frankenstein in “Frankenstein Junior”

Il presente lavoro si colloca nell’ambito della robotica mobile, disciplina che si occupa di studiare e progettare veicoli autonomi dotati di funzionalità “intelligenti”, ovvero di caratteristiche che li rendono in grado di operare senza bisogno d’intervento umano o riducendo i comandi che l’uomo deve impartire. In particolare si sono applicate le tecniche della robotica mobile allo sviluppo di una carrozzina d’ausilio ai disabili motori, estendendo le funzionalità di una carrozzina elettrica al fine di studiare come avvicinarla alle esigenze degli utenti che trovano difficoltà nell’utilizzare le normali carrozzine esistenti.

Lo scopo della presente tesi è stato progettare e realizzare un sistema di controllo basato su odometria, al fine di migliorare le funzionalità di una carrozzina robotica precedentemente realizzata in [6].

Le motivazioni di questo lavoro nascono dal desiderio di superare le forti limitazioni di usabilità che le carrozzine elettriche attualmente in commercio presentano nei confronti degli utenti affetti da specifici problemi motori. Alcuni disturbi (quali la spasticità) limitano infatti fortemente la capacità di comandare una carrozzina elettrica tradizionale, e possono dunque precludere ogni possibilità di movimento autonomo a chi ne è affetto, influenzando sullo stato di benessere fisico e psicologico delle persone, che dipende fortemente dalla possibilità di essere autonomi negli spostamenti e nei movimenti, come

illustrato in [35] [38] [16]. Nel corso degli anni sono stati studiati, progettati e commercializzati numerosi ausili che possono aiutare le persone con problematiche motorie a conseguire uno stato di indipendenza e una condizione di autonomia. Tra di essi, l'ausilio che si è dimostrato maggiormente utile alla categoria di persone afflitte da problemi motori è la carrozzina, sia essa a comando manuale o motorizzata. In particolare, la carrozzina elettrica assume un ruolo fondamentale nell'assicurare la mobilità a coloro che sono affetti da disabilità motorie.

Purtroppo, alcune categorie di disabili motori non sono in grado di utilizzare correttamente e agevolmente i dispositivi di comando presenti sulle carrozzine elettriche attualmente in commercio; il presente lavoro vuole contribuire al superamento di questa limitazione. La ricerca scientifica in ambito robotico si è spesso interessata alle problematiche di questa categoria di persone, cercando di sviluppare carrozzine con funzionalità tipiche dei robot mobili. In [32] [35] [40] sono illustrate le caratteristiche salienti di decine di carrozzine dalle funzionalità estese progettate e realizzate dagli anni novanta fino a oggi. Molti di questi prototipi sono dotati di funzionalità evolute e interessanti, tra cui evitamento ostacoli e guida autonoma, come, ad esempio, nei lavori descritti in [25] [9] [30] [5]; altri rivolgono la loro attenzione principalmente all'estensione delle modalità con cui è possibile comandare la carrozzina, utilizzando, ad esempio, sistemi di riconoscimento vocale [24], rilevazione dei movimenti della testa [20] [27] o interfacce Computer-Cervello [14] [17] [1].

Molti di queste carrozzine integrano sistemi di odometria per il calcolo della traiettoria e della posizione nell'ambiente. Questa è una soluzione a basso costo che permette di conoscere sia la velocità che la strada percorsa dalla carrozzina; grazie a queste informazioni il sistema di controllo può trarre molti benefici in termini di fluidità e sicurezza: ad esempio, conoscendo la velocità di avvicinamento a un ostacolo è possibile pianificare una traiettoria alternativa oppure arrestare la carrozzina. È possibile inoltre avere un riscontro dei comandi impartiti ai motori, cosa che si rivela particolarmente utile quando ci si deve interfacciare a un bus proprietario senza avere il controllo diretto sui motori.

I prototipi presentati poc'anzi rappresentano tuttavia vere e proprie piattaforme robotiche mobili adattate al trasporto di un utente. Essi assumono dunque il ruolo di utili prototipi da laboratorio, ma difficilmente sarebbe possibile immaginarne un uso pratico in situazioni reali, a causa di vistosi limiti in termini di praticità e costo. Ad esempio, è facile immaginare che quando l'accesso al sedile di guida della carrozzina è reso difficoltoso dalla presenza di sensori ingombranti o da altri dispositivi, l'uso della stessa ri-

sulta problematico. Inoltre in molti progetti sono stati utilizzati sensori e componenti molto costosi, impedendo di fatto un possibile sviluppo di tali progetti verso prodotti effettivamente commercializzabili.

Per cercare di ovviare a questa situazione di “prototipo” è stata sviluppata, presso il Politecnico di Milano, una carrozzina robotica in grado di offrire funzionalità “intelligenti”, ma dotata di caratteristiche di usabilità e costo comparabili con quelle di una carrozzina elettrica tradizionale. Alla base di questa c’è una carrozzina commerciale, opportunamente modificata per venire incontro alle suddette esigenze.

La carrozzina sviluppata al Politecnico è in grado di compiere movimenti autonomi e semi-autonomi, oltre che localizzarsi in ambienti indoor grazie a un sistema di visione artificiale. Non è tuttavia provvista di odometria e il controllo del moto risulta così operare in anello aperto. Le informazioni di visione artificiale non sono sufficienti per localizzare la carrozzina con precisione, dal momento che questo sistema analizza 7.5 frame al secondo e, inoltre, basandosi su marcatori artificiali, non è in grado di localizzare la carrozzina qualora non vi siano marker visibili.

Oggetto di questo lavoro è stata la riprogettazione e la realizzazione di un circuito che, collegato alla scheda di controllo interna al joystick di comando, comunica al computer la posizione della leva del joystick e permette di controllare il movimento della carrozzina. Per introdurre il dato sulla velocità della carrozzina sono state analizzate varie alternative: dopo aver scartato la soluzione basata su un accelerometro, per via del suo elevato costo, si è preferito optare per una soluzione basata su sensori ottici. Sulla carrozzina sono stati quindi installati degli encoder per misurare la velocità delle ruote e, per estensione, della carrozzina stessa. È stato quindi progettato un secondo circuito, che si occupa di leggere i dati dai sensori e comunicarli al circuito di comando principale.

La presenza di sensori odometrici ha poi consentito di estendere le funzionalità del circuito di comando, introducendo un sistema basato su controllori PID per realizzare il controllo in anello chiuso della velocità della singola ruota. A livello più alto, il software della carrozzina è stato modificato per affiancare il dato dell’odometria a quello di un sistema di visione artificiale, per migliorare la localizzazione in ambiente indoor e la ricostruzione della traiettoria della carrozzina.

Lo sviluppo del sistema è stato condizionato da alcune linee guida:

- Basso costo;
- Portabilità;

- Facilità di installazione.

Lo scopo di questa tesi è infatti quello di produrre un sistema “accessorio” sufficientemente generale da supporre di poterlo applicare a una vasta gamma di carrozzine. Per fare questo è stato necessario contenere i costi dei materiali impiegati nel sistema, pur mantenendo un buon livello di prestazioni. Si è anche sentita la necessità di produrre una soluzione che fosse al tempo stesso semplice da approntare e adattabile al maggior numero possibile di carrozzine.

Il prototipo realizzato ha mostrato che l’aggiunta dell’odometria al sistema di controllo può portare notevoli benefici: il primo luogo la carrozzina fornisce una risposta migliore quando utilizzata in modalità di guida semi-automatica. Nella fase di evitamento ostacoli, infatti, è in grado di prendere delle decisioni migliori sulla base della distanza dall’ostacolo e dalla velocità con cui si sta avvicinando ad esso. Anche la localizzazione è ora più precisa, potendo fare affidamento su due sistemi, quello odometrico e quello di visione, in grado di funzionare anche quando uno dei due è assente.

Grazie al presente lavoro si sono estese le funzionalità di una carrozzina robotica “intelligente”, sulle quali è tuttavia possibile intraprendere ulteriori azioni di affinamento ed estensione, sia considerando la possibilità di aggiungere ulteriori sensori, sia perfezionando le funzionalità già presenti al fine di creare un prototipo sufficientemente evoluto da poter essere sottoposto al giudizio di utenti disabili in ambienti reali.

Parallelamente a questo lavoro è stata implementata anche un’interfaccia Brain Computer Interface (BCI) [1] che permette di controllare il moto della carrozzina tramite l’analisi di segnali elettroencefalografici, consentendo quindi a un numero ancora maggiore di utenti disabili di aumentare le loro capacità e possibilità di muoversi autonomamente. Va però detto che questa interfaccia è da considerarsi rivolta a quei disabili impossibilitati a operare la carrozzina in altri modi, ad esempio tramite joystick, comandi vocali o movimento degli occhi. Dove possibile, infatti, risulta più semplice manovrare utilizzando il joystick o la voce.

La struttura con cui è organizzata questa tesi è la seguente:

- Nel Capitolo 2 viene illustrata l’importanza degli ausili alla mobilità e vengono presentate le caratteristiche salienti di alcune soluzioni proposte in letteratura per estendere le funzionalità delle carrozzine elettriche.
- Nel Capitolo 3 viene presentata una panoramica riguardante i sen-



sori per odometria e le tecniche della robotica mobile utilizzate per localizzare un robot nell'ambiente integrando più dati assieme.

- Nel Capitolo 4 viene descritto il progetto della carrozzina elettrica oggetto di questo lavoro, identificando gli obiettivi da raggiungere e le componenti utilizzate per il loro perseguimento.
- Nel Capitolo 5 viene presentata l'architettura del sistema illustrandone tutte le componenti, sia hardware che software, presentandone la struttura e le funzioni.
- Nel Capitolo 6 sono analizzati e commentati i risultati sperimentali ottenuti, sia dal punto di vista della realizzazione del prototipo che dal punto di vista del comportamento del controllore nello svolgere i compiti di assistenza alla guida e di guida autonoma.
- Nel Capitolo 7 sono tracciate le conclusioni del lavoro svolto e ne vengono proposte le linee di sviluppo, sia in termini di miglioramento delle funzionalità già realizzate che di identificazione di nuove funzionalità aggiuntive.



## Capitolo 2

# Ausili alla mobilità e tecnologia

*“Burns: Sai Smithers, credo che donerò un milione di dollari all’orfanotrofio. Quando voleranno i maiali.*

[Ridono entrambi, mentre un maiale sfreccia davanti alla finestra]

*Smithers: Donerà quel milione di dollari, signore?*

*Burns: No, ci ho ripensato, preferisco di no!”*

I Simpson - Ep 3F03 - “Lisa la vegetariana”

In questo capitolo si affronta il problema degli ausili alla mobilità necessari ad alcune categorie di persone. Vengono presentate le principali carrozzine elettriche intelligenti realizzate in ambito di ricerca, le linee in cui le funzionalità di questi ausili sono state estese e le loro caratteristiche salienti.

### 2.1 Ausili alla mobilità

Per gli esseri umani, la possibilità di muoversi autonomamente e in modo agevole è essenziale per il conseguimento di uno stato di benessere fisico e psicologico, come evidenziato da numerosi studi [35]. Ad esempio, nello sviluppo cognitivo dei bambini il movimento favorisce l’apprendimento dei concetti di relazione spaziale, senza i quali i normali processi di apprendimento e sviluppo delle facoltà intellettive possono risultare rallentati e limitati [38]. Negli anziani, invece, la difficoltà crescente di compiere movimenti induce un processo involutivo che peggiora le condizioni psico-fisiche della persona.

Dal punto di vista fisico le maggiori difficoltà di movimento limitano il numero e la durata delle attività che una persona riesce a svolgere, fino

a indurla a rinunciare a espletare i bisogni fisiologici e non soddisfare le necessità nutrizionali, con conseguente peggioramento della sua salute. Dal punto di vista psicologico l'impossibilità di muoversi agevolmente comporta difficoltà di socializzazione, accrescendo lo stato di solitudine e favorendo l'insorgere di stati d'ansia e depressione. In [16] si evidenzia che il 31% delle persone con difficoltà motorie soffrono di crisi d'ansia e stati depressivi, contro il 4% delle persone senza difficoltà motorie.

L'uso di carrozzine manuali o motorizzate aumenta le possibilità di movimento e favorisce lo sviluppo di circoli virtuosi che portano le persone con impedimenti fisici a un sostanziale stato di benessere psico-fisico. Le carrozzine motorizzate si guidano solitamente con un joystick e sono rivolte soprattutto alle persone che non sono in grado di imprimere alle ruote la forza necessaria per muovere una carrozzina manuale o per le quali questo compito risulterebbe troppo gravoso.

Un discreto numero di persone che potrebbero usufruire dei benefici degli ausili alla mobilità non è però in grado di utilizzare al meglio le normali carrozzine presenti in commercio. Possiamo, ad esempio, individuare in questa categoria spastici, distonici, parkinsoniani e, in generale, le persone affette da disturbi che alterano la capacità di compiere movimenti in modo accurato o che causano movimenti involontari. Inoltre risulta rischioso affidare veicoli semoventi a ipovedenti e a coloro che, oltre a problemi fisici, presentano deficit mentali.

Al fine di aumentare il grado di autonomia di queste persone, alleggerendo inoltre il compito di chi li assiste, fin dai primi anni '80 sono state studiate in ambito accademico numerose soluzioni per approntare delle carrozzine con più funzionalità rispetto a quelle normalmente presenti sul mercato. Tali funzionalità non sono da considerarsi migliorative in senso assoluto, ma solo in relazione alla particolare categoria di persone che abbiamo considerato. Per le persone che riescono agevolmente a utilizzare in sicurezza le normali carrozzine elettriche, le funzionalità aggiuntive possono risultare oltre che superflue anche scomode.

## 2.2 Carrozzine elettriche e robotica mobile

Nel corso degli anni sono state progettate e realizzate numerose carrozzine elettriche con funzionalità estese. Le modifiche sono state apportate in diverse direzioni, con diversi scopi e con funzionalità offerte differenti. Nel progettare carrozzine con potenzialità superiori rispetto a quelle normalmente presenti sul mercato, è possibile seguire tre vie:

- Creare un robot mobile con un sedile che possa ospitare una persona (e.g. Véhicule Autonome pour Handicapé Moteur (VAHM) [30] [5], Mister Ed [9]).
- Utilizzare una carrozzina motorizzata commerciale come base e cambiare i dispositivi elettronici di controllo dei motori con un apparato progettato *ad hoc* (e.g. NavChair [24], Office wheelchair with high Maneuverability and Navigational Intelligence (Omni) [4], Mobility Aid for elderly and disabled people (MAid) [29], SENARIO [19], MIT [13]).
- Utilizzare una carrozzina motorizzata commerciale come base creando dei dispositivi che si interfacciano con i bus proprietari (e.g. Smart Wheelchair Assistance Module (SPAM) [36], Hephaestus [37], Tin-Man [28], Siamo [27], TDS-PWC [15], Lurch [1]).

### 2.2.1 Estensione delle funzionalità delle carrozze elettriche

L'aggiunta di un sedile a un robot mobile è stata una delle prime strade percorse, ma risulta dispendiosa e richiede un notevole sforzo per la progettazione e realizzazione della struttura meccanica e della motorizzazione.

L'uso di una carrozzina commerciale come base permette di mantenere tutte le funzionalità che non sono legate direttamente al movimento della carrozzina, ma che la rendono differente da una "sedia a motore". Le carrozze elettriche presenti sul mercato sono studiate per servire al meglio la persona disabile che le occuperà. Il sedile è oggetto di studi di ergonomia, in quanto deve garantire comfort anche per utilizzi prolungati nel tempo. Su alcuni modelli il sedile è a sua volta dotato di attuatori che permettono, con l'uso dello stesso joystick di guida, di adattarlo alla postura della persona che lo occupa senza necessità di intervento manuale. L'uso di una base commerciale è la strada più praticata e le modifiche riguardano la sostituzione del sistema di controllo dei motori con uno creato *ad hoc*. Questa soluzione facilita il compito di aggiungere un sistema basato su odometria per conoscere la velocità delle ruote. Il joystick originale della carrozzina viene tipicamente estromesso dal controllo nella guida, oppure il suo utilizzo è mutuamente esclusivo con l'uso delle funzionalità aggiuntive previste, che saranno accessibili solo mediante un diverso dispositivo di comando.

La progettazione di dispositivi che si interfacciano con i bus dati già presenti sulla carrozzina permette un grado elevato di integrazione con le funzionalità di base disponibili e permette di realizzare un sistema che è compatibile con tutte le carrozze basate sullo stesso tipo di bus dati. Le difficoltà sono legate alla conoscenza delle specifiche elettriche del bus dati, delle infor-

mazioni che si trasmettono su di esso e del protocollo che governa lo scambio dei messaggi. Spesso i bus dati sono proprietari e le specifiche non accessibili, con conseguente necessità di effettuare operazioni di “reverse-engineering” per ricostruire le informazioni necessarie a interfacciare apparati esterni con il bus.

Intercettando i segnali analogici prodotti dalla leva del joystick prima che essi siano campionati e codificati per essere immessi sul bus dati è possibile interfacciarsi al bus dati in modo indiretto. Lo studio di questi segnali e la produzione di circuiti d’interfaccia è solitamente più semplice dell’analisi del protocollo del bus dati, anche se lega strettamente il circuito di interfaccia al particolare modello di joystick utilizzato.

### 2.2.2 La carrozzina elettrica come robot mobile

Lo scopo primario della costruzione di una carrozzina con funzionalità aumentate sta nel fornire ausili alla mobilità più efficienti e disponibili a un pubblico più vasto per diminuire le barriere e superare gli ostacoli che si presentano nel loro percorso di integrazione sociale. Dal punto di vista prettamente scientifico, però, una carrozzina elettrica risulta essere un buon veicolo adatto ad applicazioni e studi di robotica mobile. Una carrozzina elettrica vista come robot mobile dovrebbe avere le seguenti caratteristiche:

- essere progettata per muoversi in ambienti reali, interni o esterni;
- sviluppare velocità di spostamento sufficientemente elevate;
- essere agile nei percorsi tortuosi e permettere manovre agevoli in spazi stretti;
- essere dotata di una struttura robusta sulla quale poter montare dispositivi e sensori;
- avere un’alimentazione a batterie ben dimensionata da poter essere sfruttata per alimentare gli accessori che si vogliono aggiungere alla carrozzina.

### 2.2.3 Funzionalità offerte

Nel panorama delle carrozzine elettriche con funzionalità estese si possono individuare due grosse categorie: carrozzine semiautonome e carrozzine autonome.

Le carrozze semiautonome hanno come obiettivo principale quello di assistere l'utente nella guida, implementando funzioni come l'evitamento ostacoli, il passaggio attraverso porte o l'avvicinamento a oggetti. Il percorso da seguire è completamente a discrezione dell'utente, che lo comunica attraverso il dispositivo di comando (o i dispositivi) disponibile sulla carrozina.

Le carrozze autonome offrono funzionalità molto simili a quelle solitamente implementate nei robot autonomi. L'utente è chiamato a specificare una destinazione e il sistema di controllo si preoccupa di pianificare il percorso ed eseguirlo. Per svolgere questo tipo di attività è necessario conoscere e rappresentare la mappa del luogo in cui si opera o disporre di un sistema di riconoscimento dei luoghi raggiunti, ad esempio, basandosi su landmark attivi o passivi. Nell'eseguire il percorso è auspicabile che il sistema di evitamento ostacoli sia attivo. Considerando la difficoltà di integrare le informazioni conosciute sull'ambiente con quelle che si acquisiscono mano a mano (come gli ostacoli imprevisti), è raro che nelle funzionalità offerte sia prevista la possibilità di ripianificare percorsi integrando le nuove informazioni sull'ambiente.

Come è facile immaginare, è possibile far interagire la modalità semiautonomo con quella autonoma, lasciando all'utente la scelta di quale utilizzare o inserendo dei metodi automatici di cambio modalità.

In questo lavoro non saranno trattate le possibili estensioni delle carrozze dal punto di vista meccanico, come, ad esempio, la costruzione di apparati supplementari che diano alla carrozina la possibilità di salire le scale o l'installazione di sospensioni attive che garantiscano stabilità anche su terreni accidentati, analizzate in [11].

#### 2.2.4 Dispositivi di comando

I dispositivi di comando tradizionali di una carrozina elettrica sono il joystick e una serie di pulsanti direzionali. A questi sono stati aggiunti in ambito di ricerca altre interfacce di comando come

- Comando vocale, grazie alla disponibilità di hardware e software sul mercato (e.g., SENARIO [19] e carrozina MIT [13]).
- Tracciamento dell'attività oculare tramite l'uso di elettrodi o telecamere (e.g., Siamo [27]).
- Riconoscimento dell'orientamento e posizione della testa (e.g., Madarasz wheelchair [25], Intelligent Wheelchair System dell'università

di Osaka [20], Wheelchair for Museum Visit dell'Università di Saitama [34]).

- Tracciamento dell'attività cerebrale mediante Interfaccia Cervello Computer (*Brain Computer Interface*, BCI) (e.g., Università di Saragozza [17], BCW [31], MAIA [14] [39] e Lurch [1]).
- Uso della lingua come sistema di guida (e.g., TDS-PWC [15]).

Si intuisce facilmente che questi metodi di comando non sono precisi e rapidi come quello attuabile tramite un joystick. Ad esempio, con un'interfaccia di riconoscimento vocale, sarà possibile eseguire comandi a intervalli di tempo dell'ordine dei secondi; inoltre, bisogna porre molta attenzione ai casi in cui i comandi non vengono riconosciuti correttamente o vengono del tutto ignorati.

I metodi e i dispositivi di comando alternativi devono dunque essere accoppiati con sistemi di identificazione degli ostacoli che garantiscano la sicurezza dell'utente dalle conseguenze di comandi non riconosciuti o interpretati male (e.g., NavChair [24]).

### 2.3 Progetti esistenti

I primi prototipi di carrozzine elettriche con funzionalità estese risalgono ai primi anni '80. Nel corso degli anni ne sono state sviluppate molte con caratteristiche molto differenti per quanto riguarda il tipo di sensori utilizzati, le interfacce di comando per la guida e le funzionalità offerte. Di seguito sono brevemente elencati alcuni progetti e le loro peculiarità, una lista più completa si può trovare in [32], [35] e [40].

**SMART ALEC** (Stanford, USA, 1980-1990) è la prima carrozzina semiautonoma. È basata su una struttura commerciale modificata, equipaggiata con encoder sulle ruote e sonar per il rilevamento di ostacoli. Il sistema di guida è basato sul rilevamento della posizione della testa dell'utente, effettuato sempre con l'uso dei sonar. Le funzionalità offerte sono l'evitamento delle collisioni, il mantenimento di distanze prefissate da un muro, utile per lo spostamento nei corridoi, e inseguimento di obiettivi.

**Madarasz wheelchair** (Università dell'Arizona, USA, 1986) [25] è la prima carrozzina autonoma. È equipaggiata con sonar e sistemi di visione artificiale per il riconoscimento di landmark artificiali. I movimenti da eseguire sono specificati in un linguaggio apposito. La carrozzina



adotta un sistema di odometria basato su encoder ottici montati sulle ruote motrici, con una risoluzione di 1 pollice in distanza percorsa, anche se questa informazione non viene utilizzata per la localizzazione del robot.

**Mister Ed** (IBM, USA, 1990) [9] è un robot mobile con un sedile su di esso. Il software è basato su architettura subsumption con comportamenti primitivi come l'attraversamento di porte, mantenimento della distanza da un muro e inseguimento di un target.

**VAHM** (Università di Metz, Francia, 1992-2004) [30] [5], acronimo di Véhicule Autonome pour Handicapé Moteur, è basato su un robot mobile con un sedile. L'architettura di controllo è a tre livelli e rende possibile la navigazione autonoma basata su mappa interna, evitamento ostacoli e mantenimento della distanza costante da un muro. La scelta della modalità operativa è lasciata all'utente. La mappa interna è multipla (metrica e topologica). Per la pianificazione del percorso da seguire utilizza fari a luce infrarossa. È stato realizzato un secondo prototipo, con le stesse funzionalità, basato su una carrozzina commerciale modificata. Sensori odometrici e ultrasonici sono utilizzati per la localizzazione.

**NavChair** (Università del Michigan, USA, 1993-2002) [24] si basa su una carrozzina commerciale con modifiche al sistema di controllo dei motori. Offre funzionalità di evitamento collisioni e un gruppo di comportamenti per compiti specifici, come il passaggio per porte o il mantenimento della distanza costante da un muro. I comandi possono essere impartiti grazie a un sistema di riconoscimento vocale. Sui motori sono installati dei sensori di rotazione per migliorare l'accuratezza della mappa.

**TinMan** (Kipr, USA, 1994-1999) [28] rappresenta una serie di prototipi basati su carrozzine elettriche. Il primo prototipo utilizzava un dispositivo meccanico per muovere la leva del joystick, i successivi sono stati sviluppati modificando il sistema elettronico di controllo dei motori di carrozzine commerciali. Le funzionalità offerte sono l'evitamento di collisioni e la navigazione autonoma. Entrambe le versioni montano encoder sui motori, in grado di fornire un'accuratezza di 6.725 tic per pollice percorso.

**CCPWNS** (Università di Notre Dame, USA, 1994-2000) [10], acronimo di Computer Controlled Power Wheelchair Navigation System, permette

di riprodurre percorsi precedentemente appresi dal sistema. Il sistema di visione identifica landmark. Non prevede nessun tipo di evitamento ostacoli.

**SENARIO** (Finlandia, 1995-1998) [19] è una carrozzina commerciale modificata. Permette l'interazione nella guida dell'utente con il sistema di evitamento ostacoli. La navigazione autonoma è gestita con una mappa interna. L'architettura software è di tipo distribuito e la localizzazione, basata su una rete neurale, è affidata a sensori infrarossi ed encoder sulle ruote motrici.

**OMNI** (Università di Hagen, Germania, 1995-1999) [4], acronimo di Office wheelchair with high Maneuverability and Navigational Intelligence, è una carrozzina elettrica omnidirezionale commerciale modificata. Le funzionalità sono organizzate in forma gerarchica: evitamento ostacoli semplice, modi operativi per compiti specifici (distanza costante da muri, attraversamento porte) e navigazione autonoma. La velocità delle ruote è ottenuta attraverso encoder ottici montati sull'asse del motore.

**Rolland I e II** (Università di Bremen, Germania, 1997-2002) [21] [22] [23] sono state oggetto di numerose evoluzioni e studi. Sono basate su una carrozzina commerciale (Meyra Genius 1.522) che dispone già di interfacce per il comando in velocità e la misura della velocità con encoder sulle ruote. Il primo prototipo prevedeva navigazione autonoma basata su landmark artificiali e odometria abbinata all'evitamento di collisioni con sonar, IR e bumpers. Le modalità operative erano numerose (distanza costante da un muro, attraversamento di porte) ed era possibile eseguire percorsi appresi.

Nel secondo prototipo sono presenti solo sonar, ma l'algoritmo di evitamento ostacoli è più sofisticato. I comportamenti base con cui insegnare le traiettorie sono la rotazione sul posto e il mantenimento della distanza costante da un muro. Nella modalità semiautonoma la carrozzina modifica la sua velocità in base alla presenza degli ostacoli.

**MAid** (Germania, 1998-2003) [29], acronimo di Mobility Aid for elderly and disabled people, è una carrozzina commerciale modificata nella parte di controllo dei motori. Ha due modalità operative: Narrow-Area Navigation (NAN) e Wide-Area Navigation (WAN). Nella modalità NAN permette di navigare da un punto di partenza a un goal, nella modalità WAN è in grado anche di identificare ed evitare oggetti mo-

bili nell'ambiente. Successivamente è stata aggiunta la possibilità di inseguire oggetti in movimento. La parte di odometria è operata mediante un sensore giroscopico e una coppia di encoder installati sulle ruote.

**Intelligent Wheelchair System** (Osaka University, Giappone, 1998-2003) [20] interpreta i movimenti della testa con l'uso di una telecamera rivolta verso il sedile. Una seconda telecamera, rivolta verso l'esterno, permette di inseguire un obiettivo e di interpretare i comandi anche quando l'utente non è a bordo della carrozzina. La volontà dell'utente viene confrontata con la possibilità di muoversi in una certa direzione, basandosi sia su una mappa topologica sia sulle rilevazioni dei sonar. La localizzazione è affidata a un sistema di integrazione dei dati odometrici. Grazie all'uso della telecamera esterna il sistema è in grado di riconoscere la presenza dei pedoni e muovere la carrozzina in modo da evitare le collisioni.

**Hephaestus** (TRAC Labs, USA, 1999-2002) [37] prevede comportamenti di evitamento ostacoli. È compatibile con più modelli di carrozzine anche di diverse case costruttrici e non richiede nessuna modifica al sistema di controllo dei motori.

**Siamo** (Università di Alcalá, Spagna, 1999-2003) [27] è stata utilizzata con vari dispositivi di input: controllo vocale, riconoscimento di movimenti della testa, movimento oculare rilevato con elettrodi. Prevede un sistema di evitamento ostacoli con scanner laser e sensori a infrarossi in grado di riconoscere anche avvallamenti o pendii ripidi. La localizzazione è effettuata mediante encoder sulle ruote motrici.

**Smart Wheelchair** (Università di Kanazawa, Giappone, 2000) [33] è in grado di localizzarsi utilizzando dei radiofari. Grazie alla conoscenza della posizione permette la navigazione automatica. Può apprendere e riprodurre percorsi grazie all'uso di reti neurali. Non implementa nessun sistema di identificazione e evitamento ostacoli, ma è dotata di odometria sulle ruote.

**SPAM** (Sciences, USA, 2003-2004) [36], acronimo di Smart Wheelchair Assistance Module, è sviluppata come accessorio per motorizzare carrozzine manuali. È compatibile con numerosi modelli di carrozzine manuali. La funzionalità offerta è l'evitamento delle collisioni. L'architettura software è basata su regole di comportamento.

**Rolland III** (Università di Bremen, Germania, 2005) [26] ha due scanner laser posizionati al livello del terreno, uno in direzione frontale e uno rivolto all'indietro. Le ruote sono dotate di encoder incrementali per la conoscenza dell'odometria. Il sistema di visione si basa su una telecamera omnidirezionale per la ricerca di punti salienti nell'ambiente.

**BCW** (National University of Singapore, 2007) [31], acronimo di Brain Controlled Wheelchair, è basata su una carrozzina commerciale Yamaha JW-I. È dotata di encoder ottici montati su speciali rotelle, un sensore di prossimità installato sulla parte frontale e un lettore di codici a barre per la localizzazione. Il controllo avviene tramite una interfaccia BCI.

**Carrozzina MIT** (Massachusetts Institute of Technology, 2008) [13] è una carrozzina a interfaccia vocale che si appoggia a reti Wi-Fi per la navigazione indoor, mentre l'evitamento ostacoli è assicurato da un sensore SICK affiancato da due scanner laser Hokuyo.

**SILLA** (Università di Saragozza, 2008) [17] è una carrozzina comandata col cervello tramite interfaccia BCI. Dispone di encoder sulle ruote per l'odometria, mentre per l'evitamento ostacoli si affida a uno scanner SICK posto tra le gambe dell'utente.

**TDS-PWC** (Georgia Tech University, 2008) [15], acrononimo di *Tongue Drive System-Powered WheelChair*, sfrutta un sistema noto come Tongue Drive System per guidare una carrozzina commerciale interfacciandosi al bus proprietario. Il sistema prevede l'installazione di un magnete sulla lingua, i cui spostamenti vengono rilevati da alcuni sensori a effetto Hall posti su un apposito casco.

**iBOT** è una carrozzina commercializzata da Independence Technology<sup>1</sup>. Questa carrozzina è dotata di quattro ruote motrici in grado di svolgere numerose funzioni. iBOT può infatti funzionare come una comune carrozzina con due ruote motrici, andare con trazione a quattro ruote oppure ergersi in equilibrio sulle ruote posteriori, per portare il disabile alla stessa altezza degli accompagnatori. È anche in grado di salire e scendere le scale, operazione per cui è però richiesto all'utente di tenersi a un corrimano.

---

<sup>1</sup><http://www.ibotnow.com/>



*Figura 2.1: Carrozzine con funzionalità estese realizzate (in ordine da sinistra a destra e dall'alto in basso): TinMan II, Intelligent Wheelchair System, TDS-PWC, OMNI, Rolland II e MIT*

La Figura 2.1 mostra alcune delle carrozzine citate in questo Capitolo. Si notino la carrozzina Rolland II e la cintura di sonar che ne avvolge l'intera struttura e la carrozzina del Massachusetts Institute of Technology con lo scanner laser SICK sul poggia piedi.



## Capitolo 3

# Cenni di robotica mobile

*“Ogni tecnologia sufficientemente avanzata è indistinguibile dalla magia.”*

Arthur C. Clarke “Profiles of the Future” (1973)

In questo capitolo vengono presentati i sensori e le tecniche tipiche della robotica mobile che meglio si adattano all'estensione delle funzionalità di una carrozzina elettrica. Sviluppare in questa sede un'analisi completa dello stato dell'arte della robotica mobile sarebbe alquanto complesso e lungo, di conseguenza in questa sezione sono descritte solo le tecniche e i sensori che sono risultati utili o che hanno ispirato lo sviluppo del presente lavoro. Nello specifico, verranno presentati i sensori per odometria e i problemi a essa legati, proponendo una soluzione per migliorarne le prestazioni. Viene inoltre fatto un accenno alla cinematica della carrozzina.

### 3.1 Localizzazione con odometria

Per sapere in quale posizione ci si trova è possibile percorrere sostanzialmente due strade: si può “leggere” la propria posizione in un sistema di riferimento assoluto (i.e., con un sistema di localizzazione come il GPS o con landmark artificiali) oppure si può calcolare quanta strada è stata percorsa dall'ultima posizione nota. Quest'ultima tecnica, nota come “dead reckoning” e già in uso ai tempi della spedizioni per mare, consente di calcolare la posizione attuale partendo da quella precedente e conoscendo la velocità e la rotta percorsa. Quando si ha a che fare con robot mobili basati su ruote, la misura della velocità è facilmente calcolabile conoscendo la velocità di rotazione delle ruote. Per un approfondimento sul calcolo delle velocità si rimanda alla Sezione 3.3.1.

### 3.1.1 Errori dell'odometria

Il calcolo della velocità basato su odometria e la conseguente distanza percorsa si basa sull'assunzione che le ruote girino in maniera lineare rispetto al terreno. Questa affermazione è vera fintanto che non si considerano alcuni disturbi esterni che ne possono alterare anche significativamente la misura. Si consideri ad esempio il caso di una ruota sollevata da terra: essa girerà come se fosse appoggiata al suolo, ma se ci si dovesse basare sulle sue rotazioni per stabilire il movimento di un robot, essa indicherebbe che il robot si è spostato di una certa misura, ma questo non corrisponderebbe alla realtà dei fatti.

Assieme a questo problema ce ne sono anche molti altri noti in letteratura [3] che inficiano la misura dell'odometria. In generale è possibile distinguere due categorie: errori sistematici ed errori non sistematici. Nella prima categoria ricadono gli errori dovuti ai seguenti fattori:

- Il diametro delle ruote è diverso da una ruota all'altra e può variare nel tempo a seconda delle condizioni climatiche;
- Le ruote possono essere disallineate;
- I dispositivi di rilevamento hanno una risoluzione e una frequenza di campionamento finite.

Gli errori non sistematici sono invece dovuti ad agenti esogeni; tra essi si annoverano:

- Percorso su pavimenti non perfettamente piani;
- Passaggio sopra oggetti inaspettati;
- Scivolamento delle ruote dovuto a:
  - suolo sdruciolevole,
  - accelerazioni troppo elevate,
  - curve troppo veloci,
  - interazione con forze esterne,
  - forze interne, come ad esempio ruote caster,
  - contatto ruota-terra non puntiforme.

È importante distinguere tra errori sistematici e non sistematici per riuscire a minimizzarne gli effetti sul calcolo dell'odometria. Gli errori sistematici, ad esempio, sono particolarmente gravi in quanto tendono ad accumularsi



costantemente nel tempo, portando a un errore sempre maggiore durante il funzionamento. Questo è particolarmente vero se si prende in considerazione il caso di funzionamento in ambienti indoor, dove il pavimento è solitamente piano e privo di disconnessioni e gli errori sistematici rappresentano la componente principale dell'errore. Se tuttavia ci si sposta in ambienti esterni dove il suolo presenta molte asperità, sono gli errori non sistematici a pesare di più. Va anche sottolineato che, se da un lato gli errori sistematici possono in qualche modo essere compensati, per quelli non sistematici il compito è ben più arduo, in quanto si presentano in maniera casuale e non prevedibile. Nella Sezione 3.4 si introdurrà un possibile metodo per compensare questi errori fornendo una localizzazione più accurata.

## 3.2 Encoder

La misura della velocità rotazionale di un asse e, per estensione, di una ruota, è effettuabile mediante un dispositivo chiamato “encoder”. Di encoder ne esistono di svariati tipi, generalmente raggruppabili in due macro categorie:

- Encoder assoluti
- Encoder incrementali

### Encoder assoluti

Gli encoder assoluti sono in grado di rilevare la posizione assoluta dell'asse in movimento. Ciò significa che ne conoscono la posizione anche all'istante dell'accensione. Per questo sono particolarmente indicati per applicazioni di grande precisione e in generale laddove sia richiesta una conoscenza assoluta della posizione (si pensi ad esempio al problema di sapere in quale posizione si trova un braccio robotico per poterlo spostare).

### Encoder incrementali

Gli encoder incrementali (anche detti “relativi”) sono invece in grado di calcolare la variazione di spostamento, senza però sapere nulla sulla posizione iniziale. Questo tipo di encoder è inoltre più semplice da realizzare e interfacciare con un circuito di controllo, oltre a essere particolarmente indicato per il calcolo della velocità.

La classificazione degli encoder può anche essere fatta in base alla tecnologia con cui sono realizzati. Qui di seguito elenchiamo alcuni tipi di encoder:

- Potenzimetri
- Magnetici
- Induttivi
- Capacitivi
- Ottici

Nei modelli potenziometrici, si sfrutta la capacità di un potenziometro di emettere un segnale elettrico proporzionale alla posizione che assume il suo rotore. Questi tipi di encoder sono pertanto solo di tipo assoluto. I modelli magnetici, induttivi, capacitivi ed ottici possono invece essere sia di tipo assoluto che incrementale. Data la complessità delle soluzioni assolute, ci concentreremo solo sugli encoder di tipo relativo, e in particolare su quelli ottici.

### 3.2.1 Encoder ottici

Gli encoder ottici sono costituiti essenzialmente di due parti: un emettitore e un ricevitore. L'emettitore emette della luce, che può essere bianca o infrarossa, la quale interagisce con un disco dentato e viene catturata dal ricevitore. Il rilevamento avviene sul fronte tra un settore e quello attiguo, di conseguenza un disco con  $n$  settori avrà  $2 \times n$  fronti rilevabili, spesso chiamati col nome inglese di *tic*.

Gli encoder ottici si possono distinguere principalmente in due categorie:

- a riflessione
- a forcella

Per gli encoder a riflessione, emettitore e ricevitore sono posti sullo stesso piano, rivolti verso un pattern stampato. In genere il pattern è composto da una serie di settori bianchi e neri, o comunque tali da assorbire o riflettere la luce proveniente dall'emettitore. Il ricevitore cattura invece l'eventuale luce riflessa.

Gli encoder a forcella, al contrario, sono costruiti in modo che emettitore e ricevitore siano posti sullo stesso asse e rivolti l'uno verso l'altro. La luce emessa dall'emettitore passa attraverso il disco dentato e viene catturata dal ricevitore. In questo caso il pattern deve essere composto di materiale trasparente e opaco, alternato. In Figura 3.1 è visibile lo schema di un encoder: si noti anche la presenza di un circuito raddrizzatore, il cui scopo

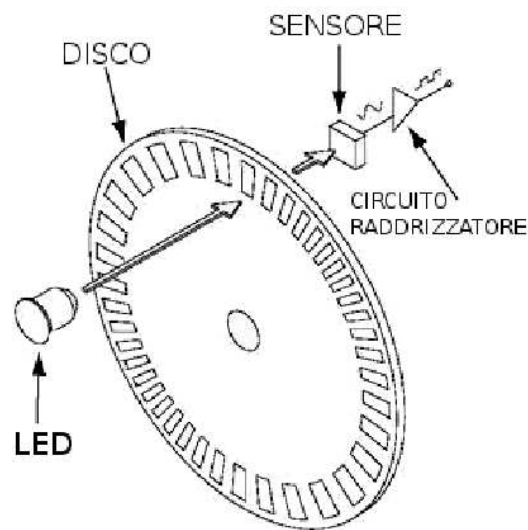


Figura 3.1: Schema di funzionamento di un encoder ottico

è quello di portare il segnale analogico in uscita dal sensore a un segnale digitale pronto per essere usato con altri circuiti integrati.

### 3.2.2 Encoder in quadratura

L'uso di un singolo encoder pone un grosso limite al sistema: se è infatti possibile stabilire il numero di tic effettuati, non è però dato sapere in che direzione si sta muovendo il disco. Per ovviare a questo limite si può ricorrere a una soluzione che prevede l'uso di due sensori per disco, che in questa configurazione prendono il nome di encoder in "quadratura". I due sensori vanno posizionati in modo che le onde risultanti siano sfasate di  $\pi/2$ , come mostrato in Figura 3.2; a livello pratico, i due sensori vanno posizionati di modo che uno sia a metà di un dente, mentre il secondo dev'essere esattamente su un fronte. Ipotizzando un sistema composto da due sensori A e B, posti l'uno a sinistra e l'altro a destra e analizzando i segnali in uscita dai due sensori è infatti possibile stabilire se il disco si sta muovendo dal sensore A a quello B o viceversa. Nello specifico, nel caso di movimento orario l'onda risultante sarà simile a quella mostrata in Figura 3.2, dove l'onda prodotta dal sensore B è "in anticipo" rispetto a quella del sensore A. Il viceversa indica una rotazione antioraria.

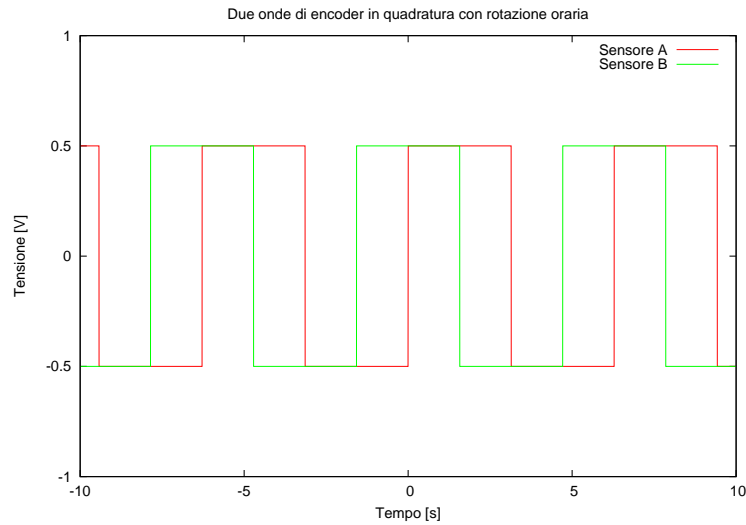


Figura 3.2: Segnale di uscita di due encoder in quadratura con disco in movimento orario

### 3.3 Cinematica di un robot mobile

La cinematica è quella branca della fisica che si occupa di studiare il moto dei corpi in relazione al tempo. In robotica ha diversi ambiti di utilizzo, anche se in questo contesto verrà trattata solo l'applicazione a robot mobili. Dovendo studiare il moto di un robot mobile, è necessario innanzitutto conoscere quali sono le tecniche utilizzate per muoverlo. Il movimento di un robot può infatti essere realizzato con zampe o ruote. Nel campo dei robot a ruote si distinguono diversi sistemi di guida:

- Differential Drive;
- Synchronous Drive;
- Omnidirectional Drive;
- Cingoli.

Il sistema di guida noto come Differential Drive è il più semplice sistema che si possa realizzare ed è costituito da due ruote motrici installate su un asse comune e in grado di muoversi avanti o indietro indipendentemente l'una dall'altra. È inoltre presente una terza ruota passiva che permette di mantenere in equilibrio il robot. I movimenti rotatori sono eseguiti facendo

muovere le due ruote con velocità e direzioni diverse. Per una trattazione più approfondita si rimanda al Sezione 3.3.1.

Nei sistemi Synchronous Drive il movimento viene invece realizzato per mezzo di ruote attuate e sterzanti controllate assieme. Ciò implica che tutte le ruote sterzano e ruotano all'unisono, permettendo quindi al robot di ruotare attorno al proprio centro. Generalmente questa configurazione prevede l'utilizzo di tre ruote, ma sono possibili anche configurazioni con un numero maggiore.

I sistemi Omnidirectional Drive, invece, prevedono l'uso di ruote attuate e sterzanti indipendenti. Questa soluzione conferisce al robot una grande libertà di movimento, potendo ruotare e traslare nello stesso istante.

I robot cingolati si possono accostare, dal punto di vista cinematico, ai robot Differential Drive. In questo caso, ovviamente, non c'è una terza ruota passiva, in quanto i cingoli sono sufficienti a garantire l'equilibrio.

Esiste poi un'altra categoria, in cui due ruote motrici sono attuate tramite un unico asse e la rotazione è garantita da una o due ruote non motrici che vengono sterzate. Si pensi, ad esempio, a un'automobile con due ruote motrici.

### 3.3.1 Cinematica per Differential Drive

Come abbiamo visto nel paragrafo precedente, il sistema Differential Drive prevede l'uso di due ruote motrici indipendenti e almeno una ruota passiva per garantire l'equilibrio. In questa configurazione la rotazione avviene attorno a un punto determinato dall'intersezione delle rette passanti per gli assi delle ruote, comprese quelle non motrici. Questo punto viene chiamato in letteratura Centro di Istantanea Rotazione (o CIR). In Figura 3.3 è illustrato il CIR per un robot con differential drive e ruote anteriori "caster". Si nota inoltre la larghezza del robot  $L$  e le velocità delle ruote  $V_L$  e  $V_R$ .

Le velocità delle due ruote sono calcolabili dall'odometria come

$$V_L = \frac{\#_{tic_L}}{\#_{tic_{giro}} \cdot \Delta t} \cdot 2\pi \cdot r_L \quad (3.1)$$

$$V_R = \frac{\#_{tic_R}}{\#_{tic_{giro}} \cdot \Delta t} \cdot 2\pi \cdot r_R \quad (3.2)$$

dove  $\#_{tic}$  è il numero di tic effettuati in un intervallo di tempo  $\Delta t$  dalla ruota e  $r_L$  e  $r_R$  sono i raggi delle ruote, sinistra e destra rispettivamente.

Conoscendo  $V_L$  e  $V_R$  otteniamo la velocità del robot, espressa come

$$v = \frac{V_L + V_R}{2} \quad (3.3)$$

$$\omega = \frac{V_R - V_L}{L} \quad (3.4)$$

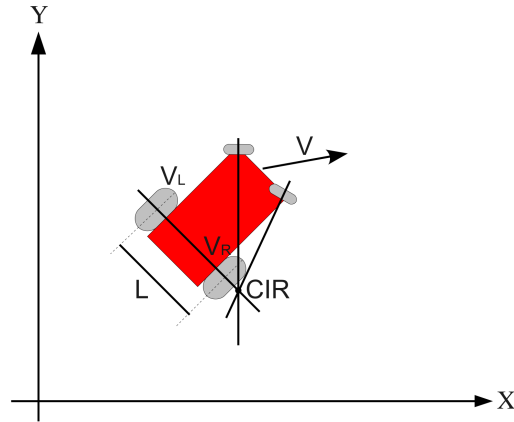


Figura 3.3: Centro di istantanea rotazione (CIR)

Per convertire la velocità del robot espressa come  $v$  e  $\omega$  in velocità delle singole ruote si procede come segue

$$V_L = \omega \left( R - \frac{L}{2} \right) \quad (3.5)$$

$$V_R = \omega \left( R + \frac{L}{2} \right) \quad (3.6)$$

dove  $R$  è il raggio di curvatura calcolabile come

$$R = \frac{L}{2} \cdot \frac{V_R + V_L}{V_R - V_L}.$$

Le Equazioni 3.5 e 3.6 sono valide solo nel caso in cui il robot proceda con velocità angolare  $\omega \neq 0$ .

Nel caso in cui  $\omega = 0$ , invece, le velocità delle ruote sono calcolabili come:

$$V_L = V_R = v$$

Per il calcolo del CIR, infine, si ricorre alla seguente espressione:

$$CIR = \begin{bmatrix} x + R \cdot \cos(\theta + \pi/2) \\ y + R \cdot \sin(\theta + \pi/2) \end{bmatrix} = \begin{bmatrix} x - R \cdot \sin(\theta) \\ y + R \cdot \cos(\theta) \end{bmatrix}$$

Se supponiamo il robot posizionato in un ambiente bidimensionale, è possi-

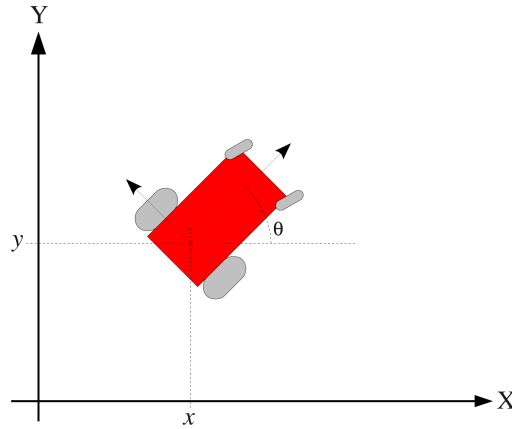


Figura 3.4: Pose del robot nel sistema di riferimento mondo

bile calcolarne posizione e orientamento tramite la postura (in inglese *pose*), definita come

$$P = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

dove  $x$  e  $y$  rappresentano la posizione in coordinate cartesiane e  $\theta$  è l'angolo rispetto all'asse  $X$  con cui il robot è orientato nello spazio. In Figura 3.4 è visibile un esempio di orientamento di un robot nel sistema di riferimento mondo, la cui *pose*  $P$  è calcolabile come:

$$\theta_t = \theta_{t-1} + \omega \cdot dt \quad (3.7)$$

$$x_t = x_{t-1} + v \cdot \cos(\theta_t) \cdot dt \quad (3.8)$$

$$y_t = y_{t-1} + v \cdot \sin(\theta_t) \cdot dt \quad (3.9)$$

La carrozzina Otto Bock Rabbit utilizzata per questo lavoro è dotata di due ruote motrici posteriori e due anteriori “caster” indipendenti. Le ruote motrici sono attuate da due motori in maniera differenziale, per questo la si può assumere come un sistema differential drive. In realtà, data la presenza di due ruote caster girevoli, lo studio della cinematica si rivela piuttosto complesso, dato che la traiettoria del moto dipende anche dalla posizione in cui le ruote anteriori si trovano all'istante iniziale. Si è però visto sperimentalmente che le leggi valide per i sistemi differential drive sono applicabili anche in questo caso, a meno di un errore che non giustifica né l'adozione nel sistema di controllo di leggi ben più complesse, né la sensorizzazione delle ruote anteriori necessaria per conoscerne la posizione.

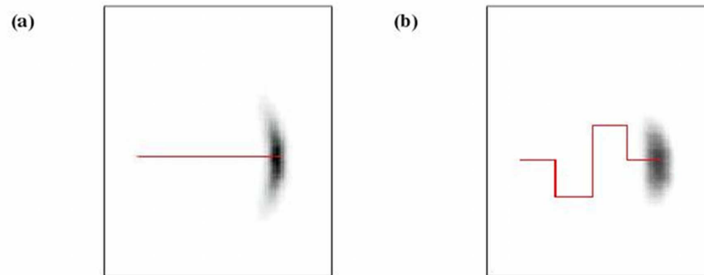


Figura 3.5: Modello di cinematica probabilistica. Distribuzione della postura del robot dopo aver eseguito la traiettoria mostrata in rosso. Più scura è la zona a maggiore sarà la probabilità che il robot si trovi in quella postura.

### 3.3.2 Cinematica probabilistica

Il modello cinematico visto finora funziona se si assume che il mondo esterno sia perfettamente predicibile; in questo caso la pose finale di un robot che compie delle azioni prestabilite sarà ottenuta in maniera deterministica utilizzando le leggi della cinematica presentate. Nella realtà il robot è però disturbato da forze esterne (e.g., l'attrito, terreno non perfettamente liscio, ecc...) che ne alterano la traiettoria. Il modello della cinematica probabilistica consente di calcolare la pose del robot non come un punto preciso, bensì come una distribuzione attorno a un punto. In Figura 3.5 sono illustrati due esempi di modello cinematico per un robot mobile rigido operante in un ambiente planare. In entrambi i casi la posizione iniziale è  $x_{t-1}$ . L'area ombreggiata di grigio rappresenta la distribuzione  $p(x_t|u_t, x_{t-1})$ , dove  $u_t$  è il vettore di comando e  $x_t$  la pose finale del robot. I punti più scuri rappresentano una maggiore probabilità che il robot si trovi in quella pose; per chiarezza, in figura è stata omessa la coordinata  $\theta$  di orientamento del robot, riproiettando la pose nello spazio cartesiano  $xy$ . La Figura 3.5a mostra un esempio di una semplice traiettoria lineare, durante la quale il robot può accumulare degli errori rotazionali e traslazionali. In Figura 3.5b è invece mostrato un esempio di traiettoria più complessa, che porta a un'incertezza maggiore.

## 3.4 Il filtro di Kalman

Come si è visto nella Sezione 3.1.1, l'odometria non è in grado di fornire con precisione una stima della posizione del robot, essendo soggetta a errori. Si cerca quindi di affiancare a essa un sistema che possa restituire la posi-



zione assoluta del robot nel sistema di riferimento mondo, con una buona precisione. Una possibile soluzione per ambienti interni prevede l'uso di un sistema di visione artificiale in grado di localizzarsi mediante il rilevamento di marcatori artificiali di cui è nota a priori la posizione. Per l'uso in ambienti esterni, dove non è possibile posizionare dei marcatori, si potrebbe invece utilizzare un sistema di posizionamento satellitare GPS. Rimane però il problema di integrare i dati provenienti dall'odometria e quelli del posizionamento assoluto, di modo che si possa sempre utilizzare l'odometria per il calcolo della traiettoria del robot, compensando gli errori con i rilevamenti della visione o del GPS. In questa tesi verrà trattato il solo caso della visione artificiale, in quanto già sviluppato sul software di controllo della carrozzina.

### 3.4.1 Filtro di Kalman per sistemi lineari

Per risolvere il problema dell'integrazione di odometria e visione viene in aiuto il filtro di Kalman, inventato negli anni '50 da Rudolph Emil Kalman [18] come tecnica di filtraggio e predizione per sistemi lineari. L'uscita del filtro al tempo  $t$  è uno stato probabilistico  $x_t$ , rappresentato come media  $\mu_t$  e covarianza  $\Sigma_t$ . La probabilità di trovarsi in uno stato successivo è espressa dall'equazione

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \quad (3.10)$$

dove  $x_t$  e  $x_{t-1}$  sono vettori di stato,  $u_t$  è il vettore di controllo al tempo  $t$  e  $A_t$  e  $B_t$  sono matrici. La quantità  $\varepsilon_t \sim \mathcal{N}(0, R_t)$  è invece una componente aleatoria che modella la casualità della transizione di stato, con media nulla e matrice di covarianza  $R_t$ . Si noti inoltre che la matrice  $R_t$  è variabile nel tempo, permettendo di modificare la covarianza in base a fattori significativi (e.g., se la velocità rilevata è alta, maggiore sarà la probabilità di errore).

L'algoritmo per questo filtro è illustrato nel listato dell'Algoritmo 1 e può essere idealmente diviso in due parti, una di predizione e una di aggiornamento. L'algoritmo restituisce la media e la covarianza dello stato  $x_t$ . La parte di predizione si occupa di calcolare la postura al passo attuale prendendo in ingresso la postura al passo precedente e i dati di controllo. Nell'aggiornamento, la postura calcolata in predizione viene corretta con i dati assoluti ricevuti dalla misura.

Le Linee 1 e 2, in particolare, rappresentano la parte di predizione e permettono di calcolare media e covarianza al passo successivo, prima di incorporare la misura  $z_t$ , qualora questa fosse disponibile.

La matrice  $K_t$  calcolata alla Linea 4 è anche nota come guadagno di Kalman

**Algoritmo 1 Kalman filter** ( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ )

---

```

1:  $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$ 
2:  $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$ 
3: if update then
4:    $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$ 
5:    $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$ 
6:    $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$ 
7:   return  $\mu_t, \Sigma_t$ 
8: else
9:   return  $\bar{\mu}_t, \bar{\Sigma}_t$ 
10: end if

```

---

e specifica a che livello la misura di aggiornamento vada incorporata nella media. I passi successivi integrano l'incertezza dello stato  $x_t$  con la misura  $z_t$ .

Si noti infine che i due sistemi di *prediction* e *update* non operano necessariamente alla stessa frequenza, in quanto la parte di *update* viene eseguita solo quando c'è effettivamente un dato disponibile (come si può notare dalla Linea 3).

### 3.4.2 Filtro di Kalman esteso

Il filtro di Kalman, per come è stato presentato nel paragrafo precedente, funziona solo se si assumono transizioni di stato e misure lineari con rumore Gaussiano aggiunto. Nella pratica queste condizioni sono però raramente soddisfatte: per esempio, un robot in movimento con velocità traslazionale e rotazionale costanti procederà tipicamente lungo una traiettoria circolare che non può essere descritta con transizioni di stato lineari.

Per poter utilizzare dunque il filtro di Kalman anche su sistemi robotici è necessario quantomeno risolvere la questione della linearità. Per questo è stato sviluppato il filtro di Kalman esteso (*Extended Kalman Filter*, EKF) in cui le probabilità dello stato e della misura sono governate da funzioni non lineari  $g$  e  $h$ :

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \quad (3.11)$$

$$z_t = h(x_t) + \delta_t \quad (3.12)$$

In particolare, la funzione  $g$  sostituisce la matrici  $A_t$  e  $B_t$ , mentre  $C_t$  è sostituita dalla funzione  $h$ . Utilizzando funzioni arbitrarie  $g$  e  $h$ , l'incertezza non è più Gaussiana. Il filtro di Kalman esteso calcola perciò un'approssimazione

**Algoritmo 2 Extended\_Kalman\_filter**  $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$ 


---

```

1:  $\bar{\mu}_t = g(u_t, x_{t-1})$ 
2:  $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$ 
3: if update then
4:    $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$ 
5:    $\mu_t = \mu_t + K_t (z_t - h(\bar{\mu}_t))$ 
6:    $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$ 
7:   return  $\mu_t, \Sigma_t$ 
8: else
9:   return  $\bar{\mu}_t, \bar{\Sigma}_t$ 
10: end if

```

---

della incertezza, rappresentandola tramite una Gaussiana. Questa incertezza continua a essere rappresentata mediante media  $\mu_t$  e covarianza  $\Sigma_t$ , ma si tratta in questo caso di un'approssimazione. Le principali differenze si possono riscontrare nell'Algoritmo 2 alla Linea 1 nel calcolo della predizione della media e alla Linea 5 nella predizione della misura. Inoltre il filtro di Kalman esteso utilizza, al posto delle matrici  $A_t$  e  $B_t$ , lo Jacobiano  $G_t$  calcolato come  $\frac{\partial g(u_t, x)}{\partial x}$  e valutato in ogni istante di tempo in  $(u_t, \bar{\mu}_{t-1})$ . Al posto della matrice  $C_t$  si utilizza infine lo Jacobiano  $H_t$ , calcolato similmente a  $G_t$  come derivata parziale della funzione  $h$  e valutato in  $(\bar{\mu}_t)$ .

### 3.4.3 EKF per ricostruzione di traiettorie

Per poter applicare il filtro di Kalman esteso a un robot mobile, come ad esempio la carrozzina oggetto di questa tesi, è necessario innanzitutto specificare come calcolare lo spostamento del robot da una posizione a un'altra. Con riferimento alla Figura 3.6, si suppone il robot nella postura iniziale  $X_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})^T$ . In seguito a uno spostamento, il robot si troverà al tempo  $t$  nella postura  $X_t = (x_t, y_t, \theta_t)^T$ . Non conoscendo l'effettiva traiettoria seguita dal robot, possiamo semplificare lo spostamento supponendolo diviso in tre punti:

- una rotazione di  $\theta_{rot1}$  per orientare il robot lungo la direzione dello spostamento,
- una traslazione di lunghezza  $\rho$ ,
- una rotazione di  $\theta_{rot2}$  per orientare il robot nella direzione finale di angolo  $\theta_t$ .

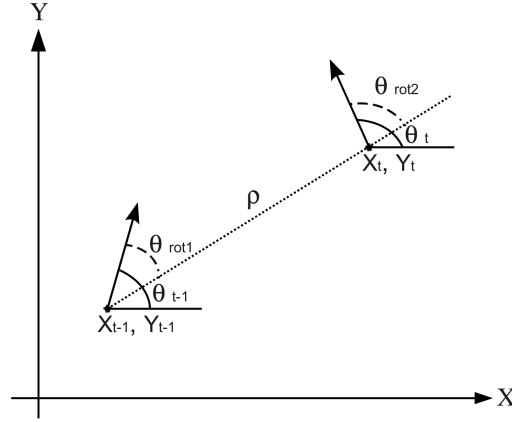


Figura 3.6: Spostamento da  $(x_{t-1}, y_{t-1})$  a  $(x_t, y_t)$  nel sistema di riferimento mondo

Conoscendo le pose  $X_t$  e  $X_{t-1}$ , è possibile calcolare le rotazioni come:

$$\theta_{rot1_t} = \arctan(y_t - y_{t-1}, x_t - x_{t-1}) - \theta_{t-1} \quad (3.13)$$

$$\theta_{rot2_t} = \theta_t - \theta_{t-1} - \theta_{rot1_t} \quad (3.14)$$

mentre la traslazione è ricavabile applicando il Teorema di Pitagora:

$$\rho_t = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (3.15)$$

Questi tre valori sono sufficienti per indicare lo spostamento del robot nell'intervallo di tempo  $(t-1, t)$  e possono pertanto essere utilizzati come variabile di controllo per il filtro di Kalman

$$u_t = \begin{pmatrix} \rho_t \\ \theta_{rot1_t} \\ \theta_{rot2_t} \end{pmatrix}$$

Lo stato  $\mu_t$  corrisponde invece al vettore pose  $(x_t, y_t, \theta_t)^T$ ; dello stesso tipo è anche il vettore delle osservazioni  $z_t$ .

Le matrici di covarianza  $\Sigma_t$  associata allo stato  $\mu_t$  e  $R_t$  associata al valore di controllo  $u_t$  sono matrici quadrate  $3 \times 3$  variabili nel tempo e scelte con i seguenti valori:

$$\begin{bmatrix} \sigma_x^2 & Cov(x, y) & Cov(x, \theta) \\ Cov(x, y) & \sigma_y^2 & Cov(y, \theta) \\ Cov(x, \theta) & Cov(y, \theta) & \sigma_\theta^2 \end{bmatrix} \quad (3.16)$$

La matrice  $Q$  associata alla misura  $z_t$ , i cui valori sono calcolati nello stesso modo della matrice 3.16, è una sempre una matrice  $3 \times 3$ , ma viene utilizzata

come matrice costante, in quanto si suppone che la pose della visione abbia una incertezza non variabile nel tempo.

La funzione  $g$  ha lo scopo di aggiornare lo stato prima di integrare la misura ed è quindi definita come segue:

$$g(u_t, \mu_{t-1}) := \begin{cases} x_{t-1} + \rho \cos(\theta_{t-1} + \theta_{rot1}) \\ y_{t-1} + \rho \sin(\theta_{t-1} + \theta_{rot1}) \\ \theta_{t-1} + \theta_{rot1} + \theta_{rot2} \end{cases} \quad (3.17)$$

La matrice  $G_t$  è invece definita come:

$$G_t := \frac{\partial g(u_t, \mathbf{x})}{\partial \mathbf{x}}(u_t, \mu_{t-1}) = \begin{bmatrix} 1 & 0 & -\rho \sin(\theta_{t-1} + \theta_{rot1}) \\ 0 & 1 & \rho \cos(\theta_{t-1} + \theta_{rot1}) \\ 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

La funzione  $h$  serve invece per “trasformare” il dato di aggiornamento in un dato di stato; nel caso visto sinora, però, la visione restituisce un dato in coordinate  $(x_t, y_t, \theta_t)^T$ , pertanto la funzione  $h$  sarà:

$$h(\bar{\mu}_t) := \begin{cases} \bar{x}_t \\ \bar{y}_t \\ \bar{\theta}_t \end{cases} \quad (3.19)$$

e la matrice associata  $H_t$  è una identità, in quanto:

$$H_t := \frac{\partial h(\mathbf{x}_t)}{\partial \mathbf{x}_t}(\bar{\mu}_t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

### 3.5 Regolatori PID

I PID [2], o regolatori ad azione Proporzionale, Integrale, Derivativa sono certamente tra i regolatori lineari più usati in ambito industriale. Tradizionalmente la struttura dei PID viene introdotta in base a considerazioni empiriche secondo le quali è opportuno che la variabile di controllo  $u$  sia generata come la somma di tre contributi. Il primo, di significato intuitivo, è proporzionale all'errore  $e$  tra il segnale di riferimento  $w$  e la variabile di uscita  $y$  del sistema sotto controllo. Il secondo contributo, proporzionale all'integrale di  $e$  (i.e., al suo valore medio), è richiesto per imporre che l'errore si annulli asintoticamente a fronte di segnali di riferimento o disturbi additivi costanti. Infine il terzo contributo, proporzionale alla derivata di  $e$ , ha lo scopo di tentare di “anticipare” l'andamento dell'errore negli istanti futuri:

se ad esempio l'andamento della derivata dell'errore è positiva, è opportuno aumentare  $u$  per provocare un aumento di  $y$  e quindi una diminuzione di  $e$ .

L'andamento nel tempo della variabile di controllo è quindi

$$u(t) = K_P e(t) + K_I \int_{t_0}^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (3.21)$$

dove  $K_P$ ,  $K_I$  e  $K_D$  sono costanti positive o nulle (nell'ipotesi che il guadagno del processo sia positivo). Il coefficiente  $K_P$  è detto coefficiente dell'azione proporzionale, mentre  $K_I$  e  $K_D$  sono rispettivamente il coefficiente dell'azione integrale e il coefficiente dell'azione derivativa.

## Capitolo 4

# Progetto logico della soluzione del problema

*“La logica è solo l’anticamera della saggezza, non il suo epilogo.”*

Spock in “Star Trek VI: Rotta verso l’ignoto”

In questo capitolo si illustra la fase di progettazione del sistema odometrico. Partendo dall’analisi dei requisiti e degli obiettivi da raggiungere, vengono presentate le scelte fatte, i sensori e la componentistica hardware utilizzati, lo studio del montaggio e posizionamento degli stessi a bordo della carrozzina, la reingegnerizzazione del circuito di interfaccia tra il sistema di controllo della carrozzina e il computer, oltre alle caratteristiche richieste al software di controllo. Vengono inoltre illustrate le varie versioni del sistema che si sono succedute nel corso dello svolgimento della tesi e le migliorie apportate al software di controllo in seguito all’adozione del sistema odometrico.

### 4.1 Carrozzina

La carrozzina commerciale a disposizione per lo sviluppo del sistema è il modello Rabbit, prodotto dalla tedesca Otto Bock<sup>1</sup>, cui è stato dato nome “Lurch”, acronimo di *Let Unleashed Robots Crawl the House*. Come si può notare dalla Figura 4.1, si tratta di una carrozzina adatta sia ad ambienti indoor che outdoor, in quanto presenta due ruote posteriori di dimensioni generose con pneumatici tassellati. La trazione è realizzata con due motori indipendenti che agiscono su ciascuna delle ruote posteriori, mentre le ruote anteriori sono libere e non comandabili. I motori assorbono una potenza di

---

<sup>1</sup><http://www.ottobock.com>



Figura 4.1: La carrozzina Otto Bock Rabbit utilizzata per questo lavoro

circa 200W ciascuno, quando utilizzati a piena potenza. L'alimentazione è fornita da due batterie da 12V e 70Ah collegate in serie, per una tensione complessiva di 24V. Le batterie sono poste sotto il sedile di guida. La carrozzina della Otto Bock implementa, per la parte di controllo, una soluzione commerciale che va sotto il nome di DX System, realizzata dalla neozelandese Dynamic Controls<sup>2</sup>. Lo schema di questa soluzione è illustrato in Figura 4.2. Il cuore del sistema è costituito da una centralina (Power Module) alimentata direttamente dalle batterie della carrozzina. Alla centralina, nella configurazione della Rabbit, sono collegati un modulo per il controllo delle luci e un Remote Control. Il collegamento avviene mediante un bus proprietario denominato DXBUS, che non è altro che un'implementazione del bus CAN, uno standard usato nel settore automotive. Il DXBUS è composto da quattro poli, due per i segnali differenziali (chiamati CAN H e CAN L) e due di alimentazione (massa e 24V). Le periferiche collegate alla centralina vengono quindi alimentate utilizzando i 24V portati dal questo bus. È inoltre possibile collegare altre periferiche alla centralina, a seconda delle necessità dei singoli utenti.

L'interfaccia utente della carrozzina è affidata a un Remote Control chiamato DX Dolphin Master Remote (DX-REM34), le cui funzioni vanno dal semplice comando del movimento della carrozzina al controllo di luci, stato delle batterie e, se il sistema lo prevede, gestione del sedile elettrico. Per la parte che pertiene a questa tesi, il Remote Control è composto da un joystick a due assi a effetto hall e da una scheda elettronica che campiona i segnali in uscita dalla leva, li elabora mediante un processore Philips P80CE598

---

<sup>2</sup><http://www.dynamiccontrols.com>



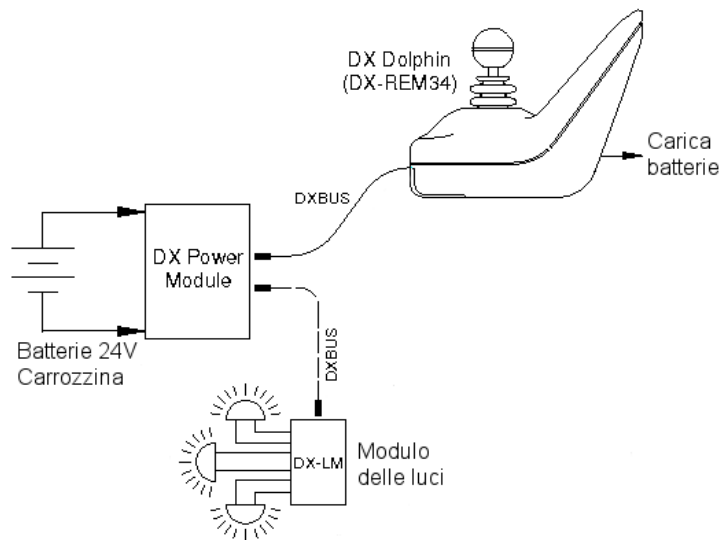


Figura 4.2: Sistema di controllo della carrozzina implementato da Dynamic Controls

e invia i dati attraverso il DXBUS. Dato che il protocollo è proprietario e non ci è stato possibile ottenerne le specifiche, si è deciso di procedere sulla stessa strada seguita nel precedente progetto, ovvero prelevare i segnali in uscita dalla leva e generare quelli necessari al controllo da collegare poi al circuito interno del Dolphin, realizzando una sorta di “man in the middle” tra leva ed elettronica di controllo. La leva del joystick è infatti collegata al circuito con un connettore a 7 poli, come illustrato in Figura 4.3(b). Di questi poli, due sono dedicati all'alimentazione della leva (5V e massa), uno è non connesso, due sono per i movimenti sull'asse avanti-indietro e due per quelli sull'asse destra-sinistra. I canali di movimento sono realizzati in configurazione differenziale, con segnali centrati su 2.5V e compresi nell'intervallo [1.25V, 3.75V], dove le tensioni (2.5V, 2.5V) indicano velocità nulla, (3.75V, 1.25V) massima velocità avanti (o destra) e, viceversa, (1.25V, 3.75V) massima velocità indietro (o sinistra). Sul joystick si nota inoltre un integrato Texas TLC274C, un amplificatore operazionale, a valle del quale è stato posto un filtro passa-basso per ogni canale. Un secondo filtro è stato inserito, sempre per ogni canale, in ingresso agli ADC del processore Philips sulla scheda del Dolphin, assieme a un partitore di tensione che mantiene la tensione a 2.5V in fase di accensione. In Figura 4.4 sono schematizzati i componenti interni al joystick.

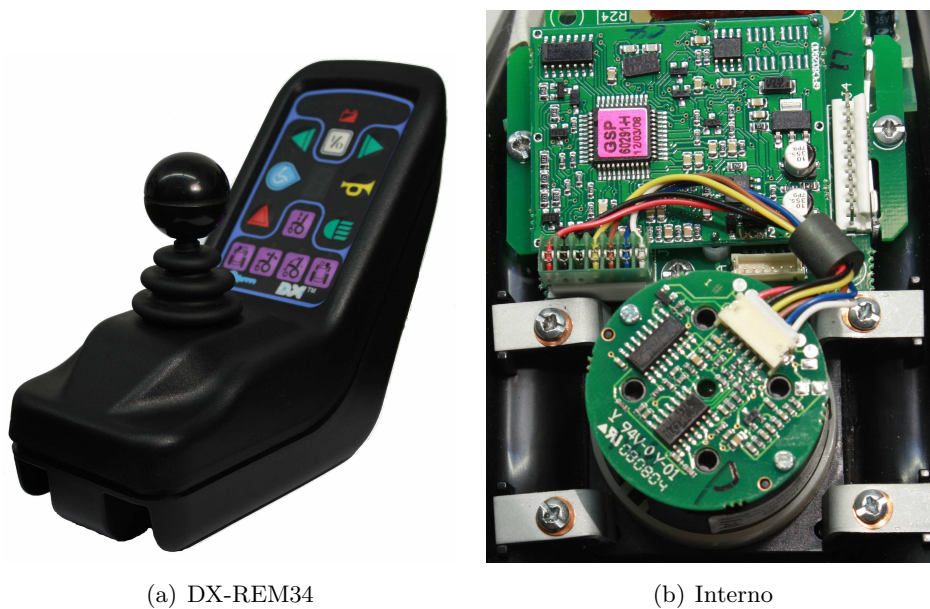


Figura 4.3: Il comando DX Dolphin Master Remote (DX-REM34) di Dynamic Controls, con un particolare dell'interno

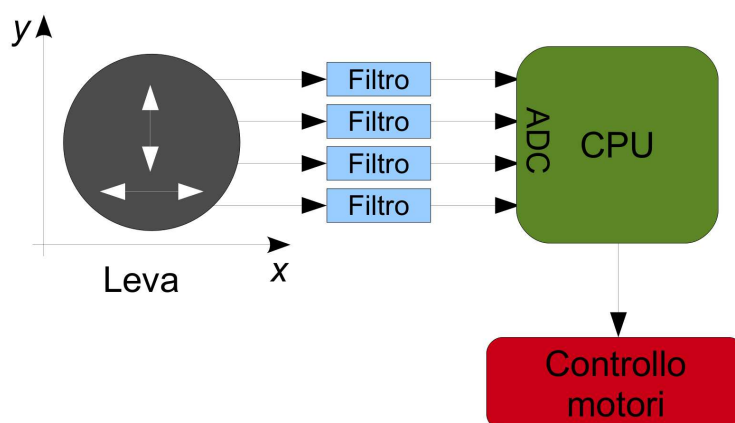


Figura 4.4: Sistema del Joystick prodotto da Dynamic Controls

### 4.1.1 Scheda di controllo

Nell'ambito del corso "Laboratorio di Intelligenza Artificiale e Robotica" è stata sviluppata una scheda di controllo per interfacciare la carrozzina a un computer [8] [7]. Questa scheda era basata su di un microprocessore Microchip PIC18F in esecuzione a 4MHz, affiancato da due convertitori Digitale/Analogico (DAC) Microchip MCP4822 e un amplificatore operativo LM324N in ingresso ai convertitori Analogico/Digitale (ADC) integrati nel PIC. La comunicazione con il PC di bordo avveniva tramite RS-232 con un integrato Maxim MAX232, sostituito poi da una scheda wireless XBee per ovviare a strani comportamenti dovuti probabilmente alla fluttuazione della massa, per via dell'elevato disturbo introdotto dai motori. Il tutto era alimentato direttamente dal joystick a 5V, andando però a gravare sui convertitori di tensione interni al Dolphin.

Il software a bordo della scheda poteva funzionare in due modalità diverse: manuale e automatica. In modalità manuale venivano letti i valori di tensione della leva dagli ADC e riprodotti in uscita sui DAC, operando di fatto in configurazione *loopback*. La modalità automatica prevedeva invece di inviare i valori letti dagli ADC al computer e riprodurre sui DAC i valori proposti dal software di controllo presente sul PC. Per passare dalla modalità manuale a quella automatica era necessario l'intervento dell'utente, premendo un pulsante; il passaggio inverso poteva invece avvenire per richiesta del PC, per interruzione della comunicazione tra PC e scheda o per intervento dell'utente, sempre mediante pressione di un pulsante. Accanto al pulsante erano stati installati anche due LED di segnalazione per indicare la modalità di funzionamento.

### 4.1.2 Software di controllo e sensori<sup>3</sup>

Il software di controllo della carrozzina è basato sull'architettura dei robot calciatori del Milan Robocup Team<sup>4</sup> (MRT), i cui moduli, chiamati "esper-ti", comunicano tra di loro scambiandosi messaggi tramite il Device Communities Development Toolkit (DCDT). La carrozzina è stata inoltre dotata di due sensori laser (Hokuyo URG-LX04) per il rilevamento ostacoli e di una telecamera rivolta verso l'alto che, unita a una serie di landmark artificiali passivi, permette il posizionamento nell'ambiente. Queste caratteristiche abilitano due tipi di funzionalità estese: la guida assistita e la guida autonoma. La guida assistita permette di evitare collisioni con ostacoli, mentre la

---

<sup>3</sup>Per un approfondimento sul software di controllo e sulla sensoristica di bordo si rimanda al lavoro svolto da Ceriani [6]

<sup>4</sup><http://robocup.elet.polimi.it/MRT/>

guida autonoma consente alla carrozzina di pianificare ed eseguire percorsi, localizzandosi nell'ambiente.

La comunicazione tra PC e circuito avveniva tramite il membro *MotorExpert*, il quale si occupava di leggere i dati di posizione della leva inviati dal circuito e comunicarli a *BrianExpert*. Questo esperto, il vero cuore del sistema, è basato su *Mr. Brian*, un sistema di controllo a comportamenti fuzzy. In *BrianExpert* confluivano anche i dati di posizione dal membro *PoseExpert* e l'informazione sugli ostacoli da *HokuyoExpert*. In particolare, se il sistema rilevava un ostacolo nelle vicinanze, modificava l'azione di comando, deviando la traiettoria o fermando la carrozzina. Dal membro *JoypadExpert* arrivavano invece i comandi proposti dal joypad Logitech Rumblepad 2, utilizzato per la guida da remoto. Questi comandi venivano ignorati se erano già presenti dei comandi dalla leva del joystick, altrimenti venivano utilizzati per l'azione di comando. *BrianExpert*, una volta elaborata l'azione opportuna, la inoltrava al membro *MotorExpert*, il quale si preoccupava poi di inviarla al circuito per l'effettiva attuazione del comando. In questa configurazione, non avendo a disposizione alcun dato riguardante la velocità, l'azione di comando veniva elaborata in posizione della leva, espressa nelle due coordinate  $x$  e  $y$ . L'intero sistema funzionava inoltre in anello aperto, dal momento che i comandi inviati alla scheda di controllo non avevano un riscontro sensoriale della loro effettiva esecuzione.

## 4.2 Analisi dei requisiti

Lo scopo di questa tesi è quello di migliorare le funzionalità offerte da "Lurch" e aumentarne parallelamente l'affidabilità. Nello specifico, i punti deboli individuati nel progetto precedente sono principalmente due: da un lato, la presenza di un controllore in anello aperto non garantiva l'effettiva esecuzione di un comando trasmesso alla scheda di controllo, dall'altro si erano verificati alcuni problemi di natura elettrica dovuti al collegamento seriale tra il PC e la scheda, parzialmente risolti con l'introduzione di una seriale wireless (XBee di MaxStream, basata sul protocollo IEEE 802.15.4) al posto della seriale RS-232.

Lurch era dotato in precedenza anche di un sensore XSens MTi in grado di leggere sia la velocità rotazionale lungo i tre assi che l'accelerazione lungo gli stessi, ma questa soluzione si è rivelata inefficace per stimare accuratamente l'effettiva velocità del robot, oltre che estremamente costosa.

### 4.2.1 Obiettivi

Per poter migliorare le funzionalità offerte da Lurch è quindi necessario operare su due fronti: da un lato si è palesata la necessità di retroazionare l'anello di controllo con la velocità della carrozzina, istante per istante, utilizzando un controllore Proporzionale Integrativo e Derivativo (PID) per il controllo del sistema. Dall'altro, i problemi riscontrati con il circuito di controllo hanno imposto una reingegnerizzazione dello stesso, al fine di renderlo più sicuro e affidabile.

Come si è visto nel precedente paragrafo, era già stato fatto un tentativo di introdurre la velocità del robot mediante una unità inerziale XSens, soluzione però scartata a causa del costo elevato. Per questa tesi si è quindi deciso di procedere con un approccio più tradizionale, introducendo un sistema di odometria sulle due ruote motrici. Per poter integrare il controllo degli encoder è stato necessario riprogettare la scheda di controllo della carrozzina, dal momento che la versione precedente non era dotata di interfacce adeguate. Allo stesso tempo si è cercato di aumentare anche l'affidabilità del progetto esistente.

Per realizzare quando appena esposto è stato quindi necessario suddividere il lavoro in due parti, una relativa all'odometria, l'altra dedicata al circuito di controllo. Per quanto riguarda l'odometria si è dovuto, nell'ordine:

- Selezionare il tipo di odometria più appropriato, sulla base di alcuni criteri che rendessero il sistema semplice, economico e riutilizzabile anche su altri modelli;
- Sviluppare un circuito per interfacciare gli encoder con il circuito di controllo.

Per quanto riguarda il circuito di controllo, invece, è stato necessario:

- Gestire i dati provenienti dai sensori odometrici;
- Integrare questi dati nel software di controllo;
- Introdurre un controllore PID grazie al dato della velocità fornito dall'odometria
- Mutare da un sistema di controllo in posizione della leva a un sistema di controllo in velocità.



Figura 4.5: Il sensore Honeywell HOA0961

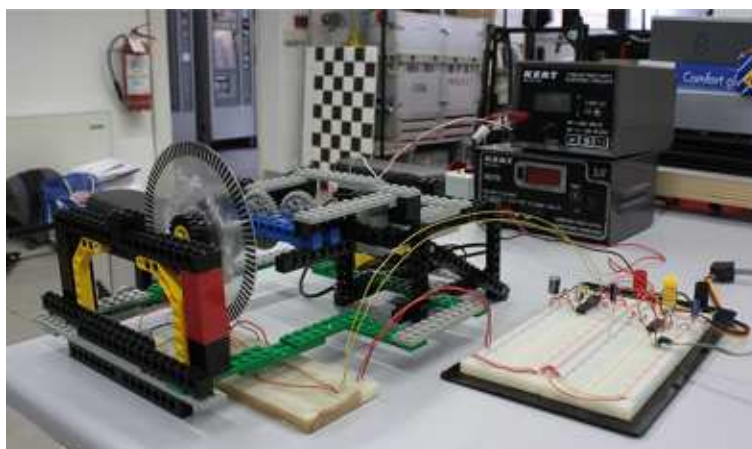
## 4.3 Encoder

### 4.3.1 Sensore

Come già illustrato nel capitolo precedente, gli encoder possono essere di vario tipo. La soluzione ottimale in questo caso sarebbe stata quella di montare un encoder meccanico sull'asse di ciascun motore. Questa ipotesi è però stata abbandonata presto a causa della presenza di un freno magnetico sull'asse che impediva l'installazione di un qualsiasi altro dispositivo esterno. Si è perciò optato per una soluzione ottica che rilevasse la rotazione delle ruote, tenendo ben presente che la vicinanza col terreno avrebbe contribuito ad accumulare sporco sul sensore, diminuendone quindi l'affidabilità. Scartata pertanto la soluzione dell'encoder a riflessione per il problema appena citato, si è iniziato a considerare la possibilità di un encoder a forcella montato all'interno della ruota. All'interno della forcella sarebbe dovuto passare un disco, solidale con l'asse della ruota, che avrebbe fornito l'indicazione di velocità grazie a una serie di settori trasparenti e opachi ai raggi infrarossi.

La scelta del sensore è ricaduta su di un modello prodotto dalla Honeywell, l'HOA0961, illustrato in Figura 4.5. Si tratta di un sensore a forcella composto da un emettitore a infrarossi e un rilevatore Optoschmitt posti uno di fronte all'altro. L'uscita del rilevatore è un segnale binario compatibile con lo standard TTL, direttamente collegabile con un integrato compatibile.

I due sensori ottici necessari per ogni ruota devono poter essere messi in quadratura una volta montati, di conseguenza si è dovuto progettare una piattina che permettesse di agganciare il sistema al telaio della carrozzina,



*Figura 4.6: Banco di prova per gli encoder: il disco è realizzato con CD trasparenti e il supporto con pezzi di LEGO MINDSTORMS*

ma al contempo consentisse di regolare la posizione dei sensori rispetto al disco. La posizione ideale è stata individuata sopra al mozzo, sfruttando l'attacco del motore al telaio. A tale scopo si è progettato un profilo in alluminio a forma di "U", sui cui lati verticali posizionare i due sensori, uno opposto all'altro.

### 4.3.2 Disco

La scelta del materiale per il disco è ricaduta inizialmente sul policarbonato, con due dischi identici a formare un "sandwich" con un terzo disco di acetato con stampato il pattern per l'encoder. Questa soluzione risultava però troppo spessa per un encoder a forcella, andando a toccare il sensore in caso di vibrazioni. Per quanto concerne invece la dimensione del disco, si era inizialmente optato per una circonferenza di 120mm di raggio, per garantire il maggior numero possibile di settori. Questa scelta creava però dei problemi di installazione all'interno del cerchione della ruota, e si è quindi optato per un raggio di 100mm, in grado comunque di garantire una risoluzione di 180 tic per giro, pari a 4 gradi per ogni settore.

I test effettuati col policarbonato (vedi Figura 4.6) hanno però evidenziato come un possibile accumulo di polvere sulla parte trasparente del disco potesse diminuirne la precisione e si è quindi deciso di realizzare i dischi in alluminio. La scelta di questo materiale ha inoltre permesso di utilizzare un solo disco per ruota, ricavando la parte trasparente agli infrarossi intagliando la circonferenza del disco per una profondità di 6mm.

Per affrancare il disco all'asse della ruota, di modo da rendere solidali i

due pezzi, si è progettato un mozzo in alluminio. L'accoppiamento del moto della ruota con il disco è garantito dalla presenza di grani di fissaggio del mozzo all'asse.

## 4.4 Scheda di controllo

Per far fronte alla maggiore potenza di calcolo richiesta dal controllore PID, la scelta iniziale per il microprocessore è ricaduta sulla famiglia dsPIC30F di Microchip, dispositivi a 16bit capaci di effettuare 30 milioni di operazioni al secondo (MIPS). Un integrato della Avago, HCTL2032 si sarebbe occupato di gestire gli encoder in quadratura e inviare il numero di tic al PIC. Questa strada poneva però dei problemi di minimizzazione delle dimensioni del circuito stampato, in quanto l'HCTL2032 è un processore a 32bit con 40 piedini. Fortunatamente, dopo qualche settimana dall'inizio dei nostri test, Microchip ha rilasciato alcuni nuovi processori della famiglia dsPIC33F con integrata una doppia interfaccia per encoder in quadratura.

I dsPIC33F sono microprocessori a 16bit da 40 MIPS che richiedono però una tensione di alimentazione di 3.3V. Questa specifica ha imposto l'utilizzo di un regolatore di tensione, in quanto non era più possibile utilizzare i 5V forniti dal joystick; per la stessa ragione si è dovuto fare affidamento su ADC esterni, anziché su quelli integrati. Questi cambiamenti ci hanno spinto ad alimentare il circuito utilizzando direttamente le batterie della carrozzina, per sgravare il più possibile il compito dei convertitori interni al joystick. La tensione generata dal joystick sarebbe dovuta servire come puro riferimento per i DAC e per accendere il circuito nello stesso istante in cui l'utente accende la carrozzina. Per realizzare questa operazione si è utilizzata la funzione di "Inhibit" del convertitore di tensione da 24V a 5V, ponendo anche particolare cura a isolare tra di loro i vari componenti.

### 4.4.1 Versione finale

Il circuito descritto pocanzi si è però dimostrato poco affidabile, in parte per il livello di complessità troppo elevato per il compito che doveva svolgere, in parte per le soluzioni adottate al fine di permettere di accendere il circuito all'accensione del joystick. In uno dei primi test, infatti, il circuito ha danneggiato il joystick senza una motivazione evidente. In seguito a questo incidente si è quindi optato per un ridimensionamento dei componenti utilizzati, suddividendo il progetto in due differenti circuiti: uno principale dedicato alla comunicazione con joystick e PC e l'altro dedicato all'odometria. Questa nuova versione è illustrata in Figura 4.7. Il circuito principale,



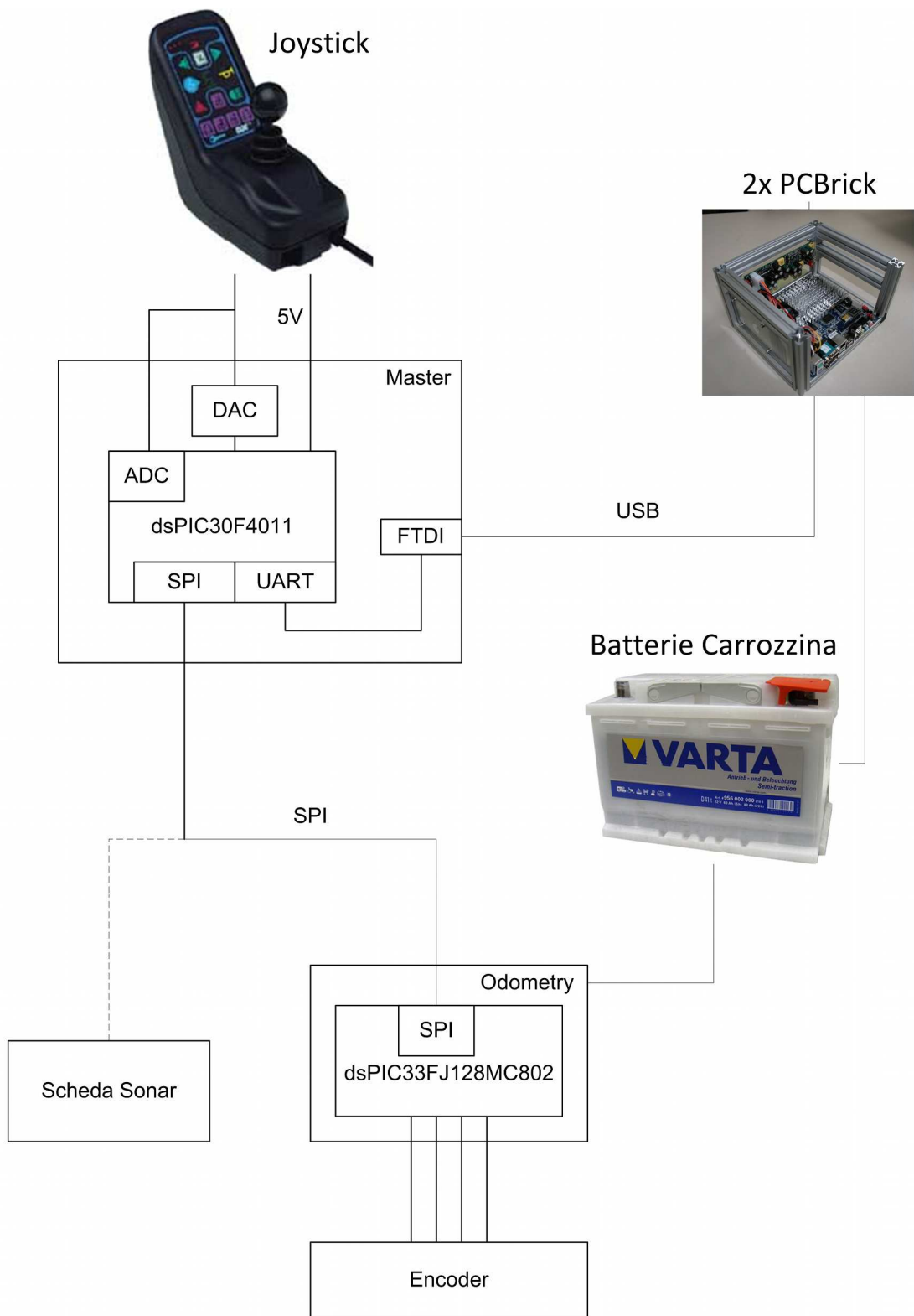


Figura 4.7: Schema della versione finale del sistema

che chiameremo “master”, è sostanzialmente un aggiornamento della vecchia scheda di controllo, avendo però sostituito il PIC18F con un dsPIC30F e la seriale wireless con una più pratica interfaccia USB.

A valle dei convertitori digitali/analogici si è deciso di mettere un filtro passa basso del primo ordine con un buffer in configurazione di inseguitore: in questo modo si “smussa” il segnale digitale, che per conformazione è “scalettato”, portandolo a un segnale più “morbido”, tipico del mondo analogico.

Il circuito “master” viene alimentato utilizzando i 5V del joystick e si occupa di leggere i dati dalla leva, inviare al joystick il comando e comunicare col PC.

Il circuito che invece si occupa dell’odometria legge i dati degli encoder e li invia poi al “master”. In questo caso l’alimentazione viene prelevata dalla scatola di derivazione, per essere poi trasformata nella tensione necessaria per alimentare il dsPIC33F.

La comunicazione tra i due circuiti avviene mediante il bus Serial Peripheral Interface (SPI); il “master” effettua la richiesta all’odometria la quale in ritorno invia il numero di tic effettuati dagli encoder, per entrambe le ruote. Questo dato viene poi utilizzato dal circuito principale all’interno dei controllori PID, o inviato su richiesta al PC.

#### 4.4.2 Dimensionamento del filtro passa-basso

Nel calcolo del filtro passa-basso si è deciso di optare per una frequenza di taglio pari a 10Hz, ben al di sopra della naturali capacità di movimento di un essere umano. Sapendo poi che

$$f_c = \frac{1}{2\pi \cdot \tau}$$

abbiamo che

$$\tau = \frac{1}{2\pi \cdot f_c} = 15.9 \cdot 10^{-3},$$

ma  $\tau = R \cdot C$ , quindi bisogna scegliere due valori per  $R$  e per  $C$  il cui prodotto dia come risultato  $15.9 \cdot 10^{-3}$ . Due valori ammissibili sono quindi  $R = 150\text{k}\Omega$  e  $C = 100\text{nF}$ , il cui prodotto è  $15 \cdot 10^{-3}$ .

Questo genere di filtro introduce anche un ritardo nell’uscita, calcolabile come

$$\Delta t = \frac{\arctan\left(\frac{f}{f_0}\right)}{2\pi \cdot f}$$

Nel nostro caso,  $f_0 = 10\text{Hz}$ , pertanto il ritardo introdotto dal filtro è riassunto nel grafico di Figura 4.8 per valori compresi tra 0Hz e 5Hz. Come

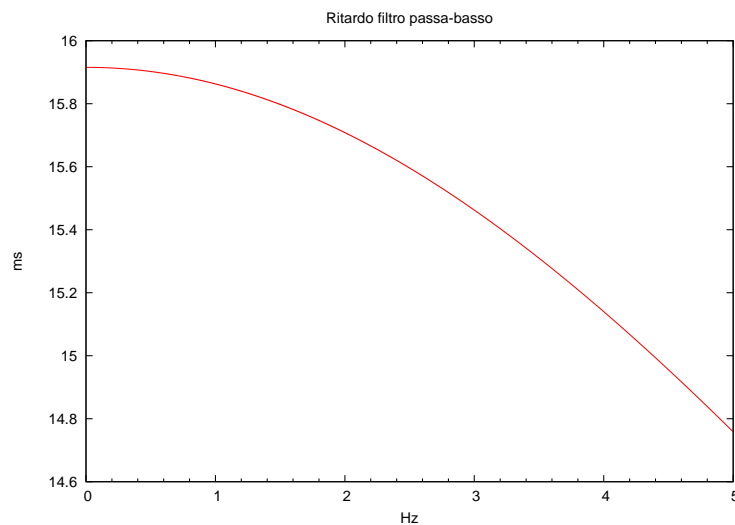


Figura 4.8: Ritardo introdotto dal filtro passa-basso

si può notare, nel caso pessimo (frequenze vicine allo zero) il ritardo è comunque inferiore a un tempo di ciclo, attestandosi a circa 15.9ms. A questo ritardo va poi sommato il ritardo introdotto dal campionamento, rielaborazione e successiva conversione. Nel caso pessimo e per frequenze molto basse può capitare che il fronte d'onda cambi un istante dopo il campionamento, quindi l'aggiornamento avverrà solo al ciclo successivo (20ms dopo). Nel caso "critico" di un ingresso a scalino, mostrato in Figura 4.9, si nota come l'uscita dal filtro impieghi  $5\tau$  per raggiungere il valore massimo, ovvero circa  $5 \cdot 15 \cdot 10^{-3} = 75\text{ms}$ . Considerato che si potrebbe verificare uno scalino soltanto in caso di errore del circuito, una risposta così lenta del filtro è da ritenersi estremamente auspicabile. È infatti impossibile che una persona riesca a produrre un'escursione così rapida utilizzando il joystick della carrozzina manualmente.

#### 4.4.3 Isolamento tra i componenti

Il sistema di controllo è composto sostanzialmente da quattro componenti: un PC, una scheda per gli encoder, una principale ed il joystick. Fatto salvo per joystick e scheda "master", che condividono la stessa alimentazione, gli altri componenti sono tutti alimentati con tensioni diverse. In questa configurazione può accadere che un componente è acceso mentre gli altri sono spenti. Per natura degli integrati selezionati, una tensione in ingresso su un piedino qualsiasi produce un'uscita dai piedini di alimentazione, andando ad alimentare anche il resto del circuito. Per ovviare a questo inconveniente si

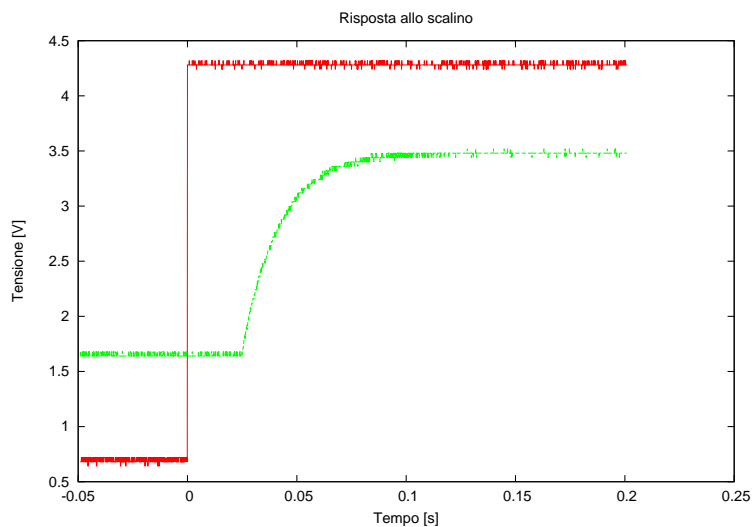


Figura 4.9: Risposta allo scalino

è reso necessario isolare elettricamente i componenti tra di loro operando, dove necessario, anche una traslazione di livello per passare da una tensione a un'altra. Nel dettaglio, è stato necessario isolare l'USB del PC dai piedini di trasmissione e ricezione della UART sul PIC; in questo caso non è però necessario effettuare una traslazione di livello, in quanto entrambi i segnali sono compresi tra 0V e 5V. Traslazione che invece è consigliata tra la scheda "master" e quella "odometria", essendo la prima alimentata a 5V e la seconda a 3.3V. In realtà, sfruttando il fatto che alcuni degli ingressi del dsPIC33F tollerano tensioni di 5V (i.e., operano correttamente anche con segnali logici 0-5V), il sistema funziona correttamente anche senza traslazione, ma si è preferito comunque effettuarla onde evitare possibili malfunzionamenti. Discorso leggermente diverso invece per gli encoder, che possono operare, ma fuori specifiche, anche a tensioni di 3.3V. È però preferibile alimentarli a 5V, cosa che comporta obbligatoriamente l'introduzione di un traslatore di livello tra encoder e PIC, pena l'errato riconoscimento dei livelli "alto" e "basso" dell'uscita TTL.

## 4.5 Computer di bordo

Per raccogliere ed elaborare i dati sensoriali e produrre l'azione di controllo di alto livello per comandare la carrozzina è necessario disporre di un'unità di calcolo. Dovendo montare il computer a bordo della carrozzina particolare attenzione deve essere posta nella scelta dell'hardware, sia per quanto



*Figura 4.10: Il computer di bordo "PCBrick", montato in una struttura in profilati di alluminio*

riguarda le dimensioni che il consumo di corrente. Per il progetto RAWSEEDS<sup>5</sup> il Politecnico di Milano ha assemblato un computer x86 compatibile a cui è stato dato nome PCBrick (Figura 4.10). È basato su una scheda VIA EN15000 che rispetta lo standard mini-ITX (170x170mm) e, grazie a un ampio dissipatore metallico, necessita solo di una piccola ventola per il raffreddamento che risulta assolutamente silenziosa. Le interfacce messe a disposizione da questa scheda madre sono numerose, tra cui: 6 porte USB 2.0, supporto per una porta FireWire, scheda audio a 6 canali integrata, connettore Gigabit LAN Ethernet, video VGA e due porte PS/2. Il processore utilizzato è un VIA C7 con frequenza di clock da 1.5GHz. Alla scheda madre sono collegati 1GB di RAM DDR2 e un hard disk da 2.5" da 80GB a 7200 giri al minuto. Il PCBrick è alimentato in corrente continua e, grazie a una scheda DC/DC, accetta tensioni di alimentazioni comprese tra 6V e 24V. La potenza massima richiesta in condizioni di massimo carico del processore è di 25W. La scheda madre e tutti i componenti citati sono montati in una struttura di profilati di alluminio di dimensioni 225x195x135mm. Il sistema operativo installato sui PCBrick è Linux, nella distribuzione Xubuntu 8.10 con interfaccia grafica XFCE. Questo Window Manager consente di avere a disposizione un ambiente grafico potente e gradevole, pur rimanendo parco di risorse hardware, cosa indispensabile quando si ha a che fare con schede a basse prestazioni come le VIA EPIA.

Nell'ambito del progetto RAWSEEDS è stato inoltre dimostrato che è

---

<sup>5</sup><http://www.rawseeds.org>

semplice e molto efficace utilizzare più PCBrick collegati in rete per gestire il calcolo in parallelo. La versione precedente del progetto Lurch prevedeva un solo computer montato sulla carrozzina, ma si è visto come in questo caso la CPU fosse spesso completamente occupata durante l'acquisizione di frame dalla telecamera, ritardando quindi l'esecuzione dei processi legati alla guida. Dal momento che un comportamento simile non può essere accettato per evidenti motivi di sicurezza, si è deciso di installare un secondo PCBrick, montato al fianco del primo.

I due computer sono collegati tra di loro mediante la rete Ethernet e permettono di eseguire il software di controllo in modo distribuito, grazie all'adozione del framework DCDT, mutuato dal software di controllo distribuito dei robot di MRT. Per evitare di utilizzare un cavo incrociato si è deciso di installare a bordo della carrozzina anche un router wireless D-Link DI-624. Questa soluzione consente anche di potersi collegare ai due PCBrick tramite rete Wi-Fi, sfruttando una shell SSH. Sfruttando poi la presa WAN del router è ovviamente possibile collegare i due computer a un'altra rete.

A uno dei due PCBrick è inoltre collegato un monitor touchscreen da 7 pollici prodotto da Xenarc. Il modello 700TSV qui utilizzato ha una risoluzione di 800x480 pixel, con rapporto d'aspetto di 16 : 10.

## 4.6 Evoluzione del sistema di controllo

Prima dell'introduzione dell'odometria il controllo del moto era in anello aperto: l'unica informazione riguardante il moto era la velocità angolare proveniente dal sensore XSens MTi, non sufficiente per garantire la chiusura dell'anello di controllo. Il sensore sarebbe anche stato in grado di fornire una stima della velocità del robot a partire dall'accelerazione lungo gli assi, ma questa misura si è rivelata ben presto inaccurata e affetta da una significativa deriva che ne peggiorava drasticamente la qualità. Sarebbe stato possibile, sviluppando algoritmi migliori, garantire una buona affidabilità di questa misura; va però considerato che sul sensore sono montati tre accelerometri, tre giroscopi e tre magnetometri, di cui si sarebbe utilizzato un solo giroscopio, lasciando inutilizzati gli altri. Visto il costo del sensore decisamente elevato si è preferito optare per la soluzione odometrica.

La localizzazione della carrozzina era inoltre fornita esclusivamente da un sistema di visione basato su marcatori artificiali, rilevati mediante una telecamera. La frequenza con cui veniva rilevata la posizione era quindi variabile in funzione del numero di fotogrammi "utili" acquisiti in un secondo dalla telecamera e dal successivo tempo di elaborazione, il cui valore massimo era in ogni caso di 15Hz.

Nell'eventualità in cui i marcatori sul soffitto non fossero stati sufficienti per localizzare il robot (si pensi al caso in cui non vi sono marcatori visibili in una zona) *PoseExpert* inviava un messaggio d'errore anziché il dato sulla posizione. Si può quindi dire che la posizione del robot era fornita in maniera binaria (i.e., o la si conosce o non la si conosce), senza offrire pertanto un parametro che ne stabilisse la precisione. Il dato fornito dalla XSens era utilizzato invece solo a livello fuzzy, per decidere quando fornire più potenza ai motori in fase di rotazione.

La presenza dell'odometria ha consentito di apportare alcune migliorie al sistema di controllo. Innanzitutto, il dato riguardante la velocità del robot ha permesso di chiudere l'anello di controllo, con notevole risparmio in termini economici grazie all'eliminazione del sensore XSens MTi e del relativo esperto. L'anello viene inoltre chiuso direttamente a livello di controllo, permettendo di semplificare le regole fuzzy, che ora si possono occupare dei set point espressi in velocità  $v$  e  $\omega$  demandando al controllore il compito di attuare il comando.

Il membro *PoseExpert* è stato inoltre esteso integrando al suo interno un algoritmo di localizzazione basato sul filtro di Kalman. I dati provenienti dall'odometria vengono trattati in maniera differenziale, ovvero viene utilizzata la differenza di posizione in cui si è mossa la carrozzina tra un intervallo di lettura e il successivo. Questi dati vengono rilevati con una frequenza pre-stabilita di 50Hz e ben si prestano quindi a essere utilizzati come *prediction* nel filtro. La parte di visione, al contrario, rappresenta una misura assoluta, ma viene rilevata a una frequenza più bassa e si possono verificare casi in cui non vi siano marcatori visibili; nel filtro di Kalman vengono quindi usati, quando disponibili, come *update* per aggiornare la posizione e fornire una stima della qualità di questa misura, grazie alla disponibilità di una matrice di covarianza associata allo stato del filtro. Nel caso in cui non siano visibili marcatori per un determinato lasso di tempo, infatti, questo parametro aumenta fino a raggiungere un valore di qualità pessima, per poi diminuire nel momento in cui il robot torna in una zona coperta dai marcatori.





## Capitolo 5

# Architettura del sistema

*“Luke Skywalker: Maestro, spostare delle pietre è una cosa: questo è del tutto diverso!”*

*Yoda: No! Non diverso! Solo diverso in tua mente. Devi disimparare ciò che hai imparato.*

*Luke Skywalker: D'accordo, ci proverò.*

*Yoda: No! Provare no! Fare. O non fare. Non c'è provare!”*

L'Impero colpisce ancora

In questo capitolo viene illustrata l'architettura del sistema, software e hardware, sia della scheda master che di quella deputata all'odometria. Vengono inoltre presentate le scelte effettuate per la parte meccanica concernente il disco e il mozzo di supporto, nonché il profilo di sostegno per i sensori ottici. Si presenta infine l'architettura del sistema di controllo sul computer, mostrando le differenze tra la versione precedente e quella successiva all'introduzione dell'odometria, illustrando quali sono le modifiche dovute a questa introduzione.

### 5.1 Architettura del sistema

Come già evidenziato nel capitolo precedente, la parte elettronica del sistema è suddivisa su due schede. Una di queste, denominata “master”, si occupa di gestire la comunicazione con il computer e con l'altra scheda, o con eventuali altre schede. Su di essa risiede il software di controllo ed è programmata anche per funzionare da sola. La seconda scheda, chiamata “odometry”, si occupa invece di raccogliere i dati forniti dai sensori odometrici e ritrasmetterli, su richiesta, alla scheda “master”.

Questa suddivisione permette anche una certa modularità del sistema. Qualora infatti si dovesse decidere di modificare il metodo di acquisizione dei

dati odometrici, implementando magari una tecnologia differente, basterà semplicemente riprogettare la parte di “odometry”, lasciando invariata la scheda “master”, almeno dal punto di vista hardware. La modularità permette anche di realizzare un sistema composto esclusivamente dalla scheda “master”; ciò sarebbe utile in casi in cui non sia possibile installare un sistema di odometria o qualora non sia necessario per il funzionamento della carrozzina (si pensi, ad esempio, a un caso in cui sia solamente necessario filtrare il segnale della leva per pulirlo da eventuali tremori o spasmi).

## 5.2 Scheda master

### 5.2.1 Hardware

In Figura 5.1 è illustrato lo schematico del circuito della scheda “master”. I componenti principali del circuito sono tre: un microprocessore Microchip dsPIC30F4011, un convertitore digitale/analogico Analog Devices AD5666, un buffer Texas Instruments TLC274 e un convertitore USB-seriale FT-DI Chip UB232R. L'alimentazione arriva direttamente dal joystick, nello schematico segnata come +5V e GND. Questa serve ad alimentare tutti i componenti presenti, ad eccezione del convertitore USB-seriale che è autoalimentato da VCCPC e GNDPC. Vi è poi una terza alimentazione, contraddistinta da +3.3V e GNDODO proveniente dal circuito dell'odometria. Essendoci quindi tre tensioni diverse in gioco, si è reso assolutamente necessario introdurre degli isolatori tra la scheda “master” e quella “odometry” e tra il PIC e il convertitore USB-seriale. Due integrati prodotti dalla Analog Digital sono preposti a questo scopo; nello specifico, l'ADUM1401 è un dispositivo a tre ingressi e un'uscita progettato e qui utilizzato per isolare l'SPI tra “master” e “odometry”, mentre l'ADUM1201 ha un ingresso e un'uscita per la UART.

Per la conversione analogico/digitale si sfruttano gli ADC a 10bit integrati nel PIC, utilizzando come riferimento di tensione i +5V e GND del joystick. La connessione è diretta dall'ingresso sul circuito al processore, in quanto si è notato, analizzando il joystick, che sulla leva è già presente un buffer con filtro passa-basso di primo ordine e un eventuale secondo filtro può essere realizzato via software.

La parte di conversione digitale/analogico, invece, è stata realizzata cercando di renderla il più simile possibile all'implementazione presente sulla leva. Per questo motivo è stato realizzato un filtro passa-basso di primo ordine a valle del quale si è posto un amplificatore operativo Texas Instruments TLC274, lo stesso modello impiegato anche sulla leva prodotta dalla

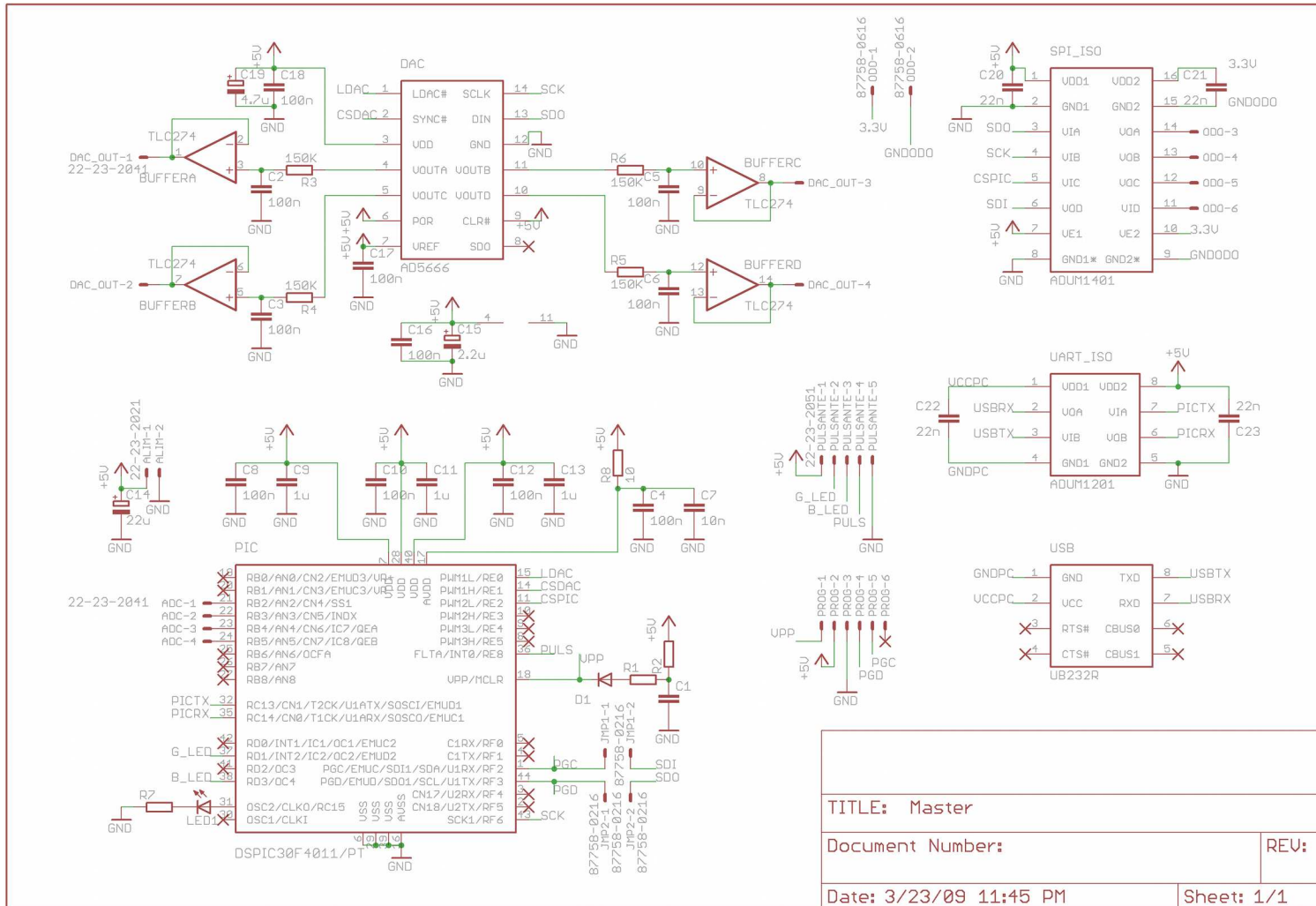


Figura 5.1: Circuito di interfaccia tra joystick della carrozzina e computer

Dynamic Controls. Il DAC utilizzato è invece un Analog Digital AD5666, un modello a 16bit controllabile attraverso il bus SPI che permette inoltre di essere impostato per avere una tensione di reset pari a metà del riferimento. Questa opzione risulta molto utile in quanto il segnale differenziale è effettivamente centrato su 2.5V; viene così garantito che, anche in caso di un reset accidentale, il DAC presenti in uscita una tensione che equivale a quella di “stasi” della carrozzina (2.5V su entrambi gli assi).

Per segnalare all’utente lo stato del sistema, sono stati adottati due LED di segnalazione collocati su di un circuito esterno assieme al pulsante per il cambio di modalità, come in Figura 5.2, di modo da poter installare questo circuito in una posizione visibile e facilmente accessibile dall’utente, senza avere la necessità di tenerli sul circuito principale. Come per la versione precedente, il LED verde indica la modalità di funzionamento manuale, quello blu la modalità automatica, mentre entrambi i LED accesi indicano uno stato di sicurezza, denominato “safe”, da cui è possibile uscire solo mediante intervento umano e in cui la carrozzina rimane ferma. Il sistema può infatti passare da manuale ad automatico solo su esplicita richiesta dell’utente (i.e., tramite la pressione di un pulsante), mentre il contrario può avvenire anche su richiesta del sistema o per timeout e prevede il passaggio attraverso lo stato *safe*. La scelta di questo stato è fatta per evitare che un utente si ritrovi senza più l’ausilio del computer senza preavviso, o peggio ancora con la leva del joystick in posizione non “sicura”. Supponendo ad esempio che un disabile stia utilizzando il sistema di evitamento ostacoli e guidando di conseguenza, se per un qualsiasi errore il sistema dovesse cessare di operare correttamente l’utente si troverebbe inconsapevolmente senza più protezioni, rischiando di farsi del male o danneggiare la carrozzina.

La comunicazione con il PC è invece gestita da un integrato FTDI Chip UB232R, un circuito stampato che include un processore FT232R e un connettore micro USB. Il dispositivo è autoalimentato sfruttando i 500mA forniti dall’USB, fattore che si rivela molto utile nell’ottica di risparmio energetico volto a minimizzare l’impatto del circuito sul joystick della carrozzina.

Come si nota dallo schematico, infine, manca un quarzo esterno: questo perché si è visto dagli esperimenti fatti che l’oscillatore interno al PIC svolge un compito più che sufficiente a garantire l’affidabilità di cui c’è bisogno, operando a una frequenza di 29.48MHz. Inoltre l’assenza del quarzo esterno porta a un risparmio sia in termini di spazio occupato sul PCB che di corrente consumata, anche se quest’ultimo fattore non si può certo definire rilevante.

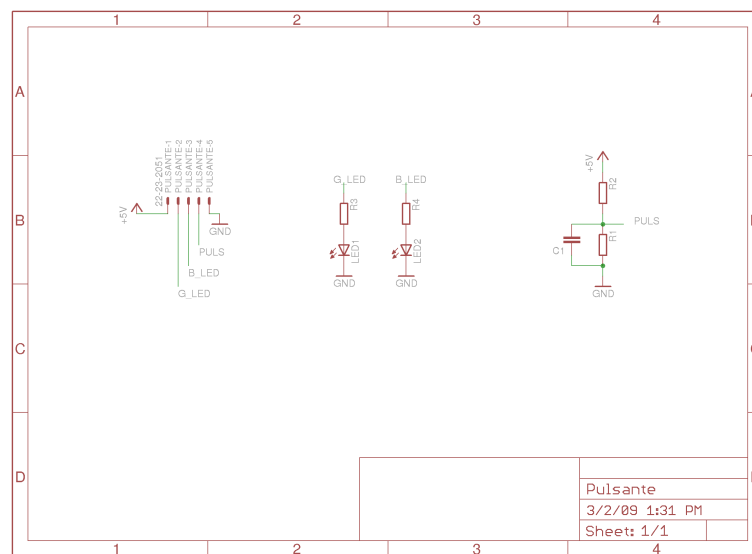


Figura 5.2: Circuito per i LED di segnalazione e il pulsante

### 5.2.2 Software

Il software di controllo della scheda master è diviso in vari blocchi, dedicati ognuno a una funzione specifica:

- **converters**, che gestisce DAC e ADC;
- **eeprom**, che si occupa di salvare sulla EEPROM integrata nel PIC i parametri di configurazione;
- **mapping**, per convertire i valori della posizione della leva in velocità del robot, la velocità del robot in velocità delle singole ruote e viceversa;
- **modes**, che gestisce il passaggio tra le varie modalità di funzionamento: manuale, automatica e *safe*;
- **odometry**, che si occupa sia di richiedere i valori dell'odometria alla scheda "odometry", sia di inviare gli stessi al PC;
- **parser**, per interpretare i comandi che arrivano dal PC tramite seriale;
- **pid**, che contiene il controllore PID: riceve in ingresso il set point e la velocità attuale e attua il comando appropriato;
- **run**, che contiene la sequenza di operazioni che permettono di far funzionare il sistema;

- **serial**, che si occupa della comunicazione seriale con il PC;
- **setup**, che contiene i parametri di configurazione del PIC, come la durata dei vari timer, la configurazione delle porte, ecc. . .

Più nel dettaglio, il blocco *converters* contiene i metodi per leggere dai convertitori analogico/digitali integrati e scrivere su quelli digitale/analogici. La funzione *readADC* si occupa di campionare gli ADC integrati nel PIC. I segnali così acquisiti rappresentano però i valori assoluti di tensione dei quattro poli della leva del joystick. Per tornare al segnale differenziale originale, la funzione esegue anche una sottrazione tra il valore alto e il valore basso, per ognuno dei due canali, di modo da poter lavorare con due soli valori, il primo per l'asse  $x$  e il secondo per l'asse  $y$ . Qualora questi valori siano al di fuori di un intervallo prestabilito, vengono impostati al valore più prossimo dell'intervallo. Nel nostro caso, sappiamo che il segnale differenziale massimo che può produrre la leva è di 2.5V; se per qualche ragione il valore ottenuto dopo il campionamento degli ADC fosse maggiore in valore assoluto, questo verrebbe impostato a 2.5V (o -2.5V nel caso di valore negativo). I valori di massimo e minimo possono comunque essere impostati dall'utente in fase di configurazione del sistema. La funzione *write2DAC*, invece, invia tramite il bus SPI i comandi di configurazione e i valori da scrivere al DAC Analog Digital. Essendo il valore ottenuto dagli ADC un numero a 10bit e avendo i DAC una risoluzione di 16bit, vengono impostati a 0 gli ultimi 6bit da inviare al DAC. I valori passati a questa funzione sono già da intendersi come i valori di tensione dei quattro fili della leva.

Il blocco *eeprom* è invece deputato a gestire la memoria EEPROM integrata nel PIC. Questa viene utilizzata per mantenere salvati in memoria non volatile i dati di configurazione del PIC, onde evitare di doverli inviare da PC ogni qualvolta si accende il sistema. Nel blocco sono presenti le funzioni che servono per inizializzare la EEPROM, scrivere in memoria i parametri di configurazione e caricarli al riavvio successivo.

Nel blocco *mapping* sono invece contenuti i metodi per convertire tra di loro i diversi tipi di velocità utilizzati all'interno del software. Dall'odometria vengono infatti lette le velocità delle singole ruote, del tipo *Vel\_ruote*, espresse in  $v_l$  per la ruota sinistra e  $v_r$  per la ruota destra. La posizione della leva del joystick viene tradotta in *Vel\_joy*, nelle componenti  $v_x$  per l'asse  $x$  e  $v_y$  per l'asse  $y$ . Di fondamentale importanza è anche la velocità del robot, espressa da *Vel\_robot* in velocità tangenziale  $v$  e velocità angolare  $\omega$ .

Il blocco *modes* si occupa invece del cambio della modalità di funzionamento tra automatico, manuale e *safe*. La funzione *modeChange*, in particolare, verifica lo stato attuale di funzionamento e passa alla modalità corretta,

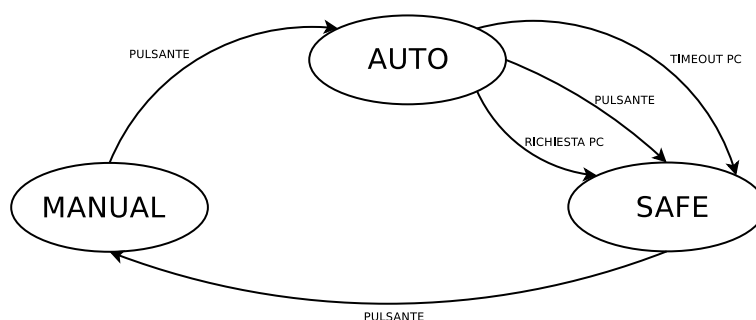


Figura 5.3: Possibili stati di funzionamento e modalità di passaggio da uno stato all'altro

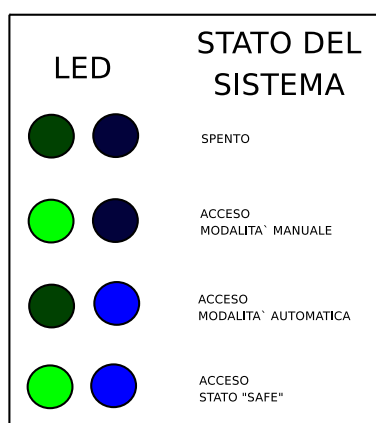


Figura 5.4: Stato dei LED con significato delle varie configurazioni

seguendo lo schema illustrato in Figura 5.3. Il cambio di modalità implica anche l'aggiornamento dei LED di segnalazione posti sulla scheda pulsante, come nella configurazione illustrata in Figura 5.4.

La gestione della parte relativa all'odometria è invece affidata al blocco *odometry*; in esso si trovano il metodo che si occupa della richiesta e della ricezione dell'odometria alla scheda secondaria e quello che inoltra i dati raccolti al PC.

Nel blocco *parser* viene gestito l'intero processo di configurazione dei parametri. I parametri sono contenuti in una struct denominata *Settings* contenente i seguenti campi:

- **Kpl, Kil, Kdl, Kpr, Kir e Kdr**: rappresentano la P, I e D e del controllore PID, per la ruota sinistra (l) e destra (r);
- **l**: rappresenta la distanza tra le ruote del robot, espressa in [m];

- **tic**: rappresenta il numero tic al giro per ogni ruota;
- **circl** e **circr**: rappresentano la misura della circonferenza delle ruote, sinistra (l) e destra (r), espresse in [m];
- **numPesi** e **pesi**: rappresentano, rispettivamente, quanti valori utilizzare e con che peso per effettuare la media mobile della velocità rilevata dalle ruote per ottenere un valore medio da utilizzare come retroazione del controllo;
- **maxV\_FW** e **maxV\_RW**: rappresentano la massima velocità raggiungibile dal robot, nella direzione avanti (FW) e indietro (RW), espresse in [m]/[s];
- **maxW**: rappresenta la massima velocità angolare raggiungibile dal robot, espressa in [rad]/[s];
- **maxt1**, **center1**, **mint1** e **maxt2**, **center2**, **mint2**: rappresentano i valori massimi, centrali e minimi che possono assumere le tensioni in uscita sul primo (1) e secondo (2) canale. Questi valori permettono il funzionamento del circuito anche con altri modelli di joystick, dando all'utente la possibilità di impostare i valori di tensione specifici di ogni modello;
- **minDiff** e **maxDiff**: rappresentano la minima e massima differenza ammissibile in ingresso agli ADC;
- **controller**: viene settato a P se il controllore PID è attivo, a N altrimenti;
- **setbit**: ogni qualvolta un parametro di configurazione viene correttamente ricevuto, viene settato il corrispettivo bit in questa variabile. Questa operazione viene effettuata perché per poter far partire il software è necessario che prima siano stati ricevuti alcuni parametri, diversi a seconda che si scelga di attivare o meno il controllore PID;
- **nullo**: un carattere vuoto, serve solo a portare la dimensione della struct a un numero pari.

Il blocco *pid*, come suggerisce anche il nome, contiene le funzioni necessarie al controllore PID.

Il blocco *run* è diversificato a seconda che sia attivo il controllore PID o meno e contiene la sequenza di operazioni necessarie a far funzionare la carrozzina. Nello specifico, le operazioni sono le seguenti:



1. Leggere dagli ADC la posizione della leva
2. Se il sistema è in modalità automatica, proporre i valori inviati dal PC, altrimenti ricopiare quelli della leva in modalità manuale o ignorarli se in modalità *safe*
3. Se è attivo il controllore PID passargli questi valori
4. Scrivere sul DAC i valori di controllo

Della comunicazione seriale se ne occupa invece il blocco *serial*: la RS-232 è configurata per la velocità di 38400 baud.

Per concludere, il blocco *setup* si occupa della configurazione del PIC e delle sue periferiche. In esso si trovano i metodi delegati al setup degli ADC, del bus SPI e dei timer. Riguardo i timer, nel sistema ne sono presenti due: uno che temporizza il ciclo di esecuzione a 50Hz e uno per gestire il timeout di ricezione dati dal PC. La temporizzazione a 50Hz è stata introdotta per due motivi: da un lato l'elettronica della carrozzina è stata pensata per interagire con esseri umani, i cui movimenti della mano difficilmente raggiungono i 10Hz; dall'altro si è cercato di non sovraccaricare di lavoro il PC, già impegnato a svolgere gli altri compiti di controllo. Per il teorema di Nyquist-Shannon, infatti, il massimo segnale campionabile è di 25Hz, ben maggiore dei 10Hz che sono stati supposti come massimo valore per una persona. Il timeout di ricezione dati dal PC serve invece per dichiarare inattivo il PC e passare quindi in modalità *safe*, se ci si trovava in modalità automatica. Attualmente questo timer è impostato a 2 secondi, ma è possibile modificarlo a piacimento, a seconda delle proprie necessità. Chiaramente, impostare un valore troppo basso rischia di diventare controproducente, dal momento che per un banale ritardo di trasmissione si può perdere la modalità automatica, pur essendo il PC perfettamente funzionante.

L'esecuzione del programma parte con la configurazione delle varie periferiche, come nel blocco *setup*. Successivamente vengono caricati dalla EEPROM i parametri di configurazione. Nel caso questi risultassero nulli, il sistema si pone in attesa degli stessi da parte dell'utente, altrimenti viene eseguito il codice del blocco *run*, verificando se è stato selezionato il controllore PID. In Figura 5.5 si può vedere il ciclo di esecuzione del software sulla scheda "master" durante il funzionamento in modalità manuale.

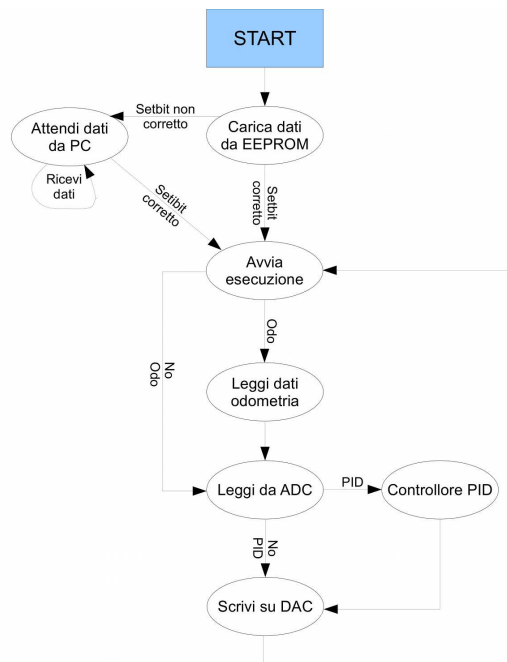


Figura 5.5: Ciclo di esecuzione del programma sulla scheda master

## 5.3 Scheda odometry

### 5.3.1 Hardware

Il circuito realizzato per la scheda “odometry”, come si può notare dalla Figura 5.6, è decisamente più snello rispetto al “master”, in quanto deve svolgere un compito molto più semplice. Il cuore del sistema è costituito da un microprocessore Microchip dsPIC33FJ128MC802, alimentato a 3.3V. L'alimentazione proviene dalla scatola di derivazione alimentata a sua volta dai 24V delle batterie, che vengono poi trasformati in 5V da un convertitore DC-DC. Questi 5V servono per alimentare direttamente gli emettitori e i fotorilevatori degli encoder, oltre che per essere trasformati nella tensione richiesta dal PIC (3.3V) da un regolatore low dropout (LDO) Microchip MCP1702. Questo regolatore è in grado di erogare un massimo di 250mA, comunque più che sufficienti per alimentare il circuito. Durante i primi esperimenti gli encoder erano alimentati a 3.3V, consentendo pertanto un collegamento diretto con il PIC. Avendo però deciso di alimentarli, come da specifiche, a 5V, si è imposto l'uso di un traslatore di livello Texas Instruments TXB0104. Questo componente consente di “shiftare” il segnale di uscita dai 5V degli encoder ai 3.3V richiesti dal PIC, garantendo così il

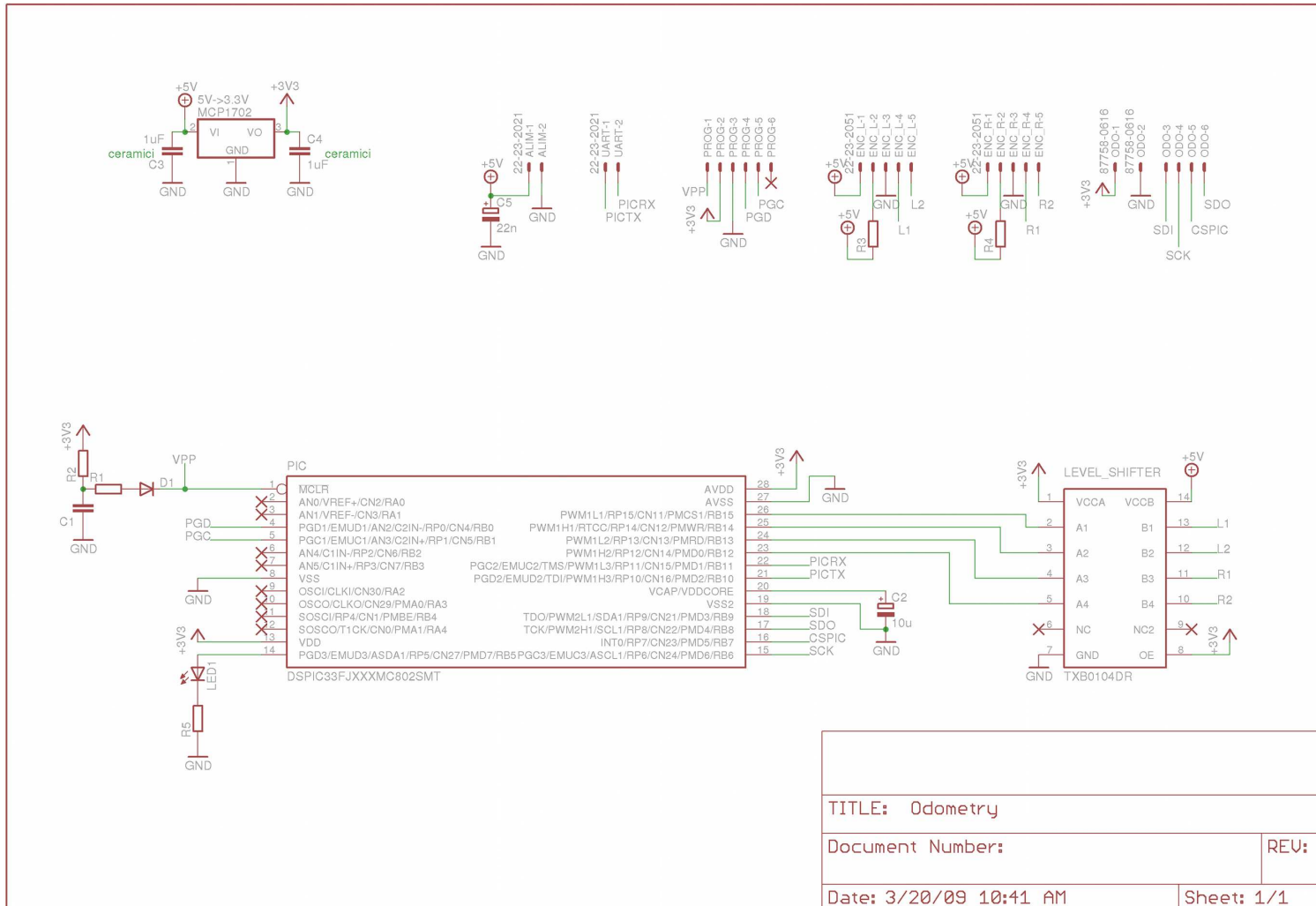


Figura 5.6: Circuito per l'acquisizione dei dati dell'odometria

corretto funzionamento del sistema.

La connessione verso gli encoder è realizzata con due connettori molex da 5 pin l'uno, un connettore per ogni ruota (ricordiamo che su ogni ruota sono installati due encoder). I connettori portano agli encoder una tensione a 5V necessaria per il fototransistor, l'alimentazione del fotodiode e una massa. I restanti due pin sono dedicati al canale di ritorno, uno per encoder, con il segnale TTL che esce dagli encoder.

Gli altri connettori presenti sulla scheda sono per l'alimentazione, la programmazione on circuit e l'SPI verso la scheda "master". In particolare, quest'ultimo, oltre ai classici segnali di SDI, SDO, clock e chip select, porta anche una tensione da 3.3V con relativa massa, necessaria ad alimentare l'isolatore lato "master". Si nota, infine, un molex per la connessione a un *transceiver* seriale, utilizzata solo per debug.

### 5.3.2 Software

Anche la struttura software riflette l'estrema semplicità della scheda dedicata agli encoder. Il software è suddiviso essenzialmente in tre blocchi:

- **odometry**, che si occupa di gestire l'odometria;
- **serial**, comune alla scheda "master", che si occupa di gestire la seriale;
- **setup**, che gestisce la configurazione del PIC.

Entrando nel dettaglio, in *odometry* sono presenti i metodi per leggere il numero di tic dalle interfacce per encoder in quadratura del PIC (Quadrature Encoder Interface, QEI) e per inviarli alla scheda "master" quando richiesti. Lo stesso metodo invia i dati anche alla seriale, di modo da poterli utilizzare in fase di debug. Per la descrizione del codice contenuto in *serial* si rimanda alla sezione precedente.

Anche il blocco *setup* è sostanzialmente mutuato dalla scheda "master", ma in questo caso le uniche periferiche da configurare sono l'SPI e quella per gli encoder. Viene inoltre riproposto il timer da 20ms che garantisce così una temporizzazione a 50Hz. Non si tratta però di una temporizzazione necessaria al funzionamento del circuito, ma è utilizzata esclusivamente per la segnalazione del funzionamento della scheda tramite un LED lampeggiante.

Il flusso di esecuzione prevede che il programma main esegua la fase di *setup*, a seguito della quale si pone in un ciclo infinito. La comunicazione con la scheda "master" è gestita tramite interrupt. Nel momento in cui viene richiesto un dato, si genera un interrupt che avvia la ricezione dei dati sul

bus SPI. Viene quindi controllato il carattere inviato dal “master” e spedito il dato adeguato. Per una trattazione più approfondita del protocollo di comunicazione tra le due schede, si rimanda alla Sezione 5.4.1.

## 5.4 Protocolli di comunicazione

### 5.4.1 Protocollo di comunicazione master/odometry

Il protocollo di comunicazione, come illustrato anche in Figura 5.7, prevede che il master invii all’odometria il carattere `0x00FF` per il dato riguardante la ruota sinistra e `0x0AB0` per la ruota destra. I due caratteri sono stati scelti di modo da avere una bassa probabilità che possano essere confusi in seguito a errori di trasmissione. A seguito di questa richiesta, il circuito di odometria invia al master i dati riguardanti la ruota selezionata, in formato esadecimale a 16bit. I due registri *POS1CNT* e *POS2CNT* rappresentano rispettivamente il contatore degli encoder per la ruota sinistra e per la ruota destra.

La scelta di inviare i dati solo su esplicita richiesta è dovuta principalmente al fatto che l’interfaccia SPI è un protocollo di tipo master/slave e abilita la trasmissione solo in presenza di un clock sulla linea *SCK*. Questo clock può essere generato solo dal dispositivo master, pertanto abilitare una trasmissione continua dall’“odometry” al “master” avrebbe significato impostare il primo come “master” SPI e il secondo come “slave”. Così facendo però la scheda “master” non sarebbe più stata in grado di comunicare con i DAC, anch’essi gestiti via SPI, se non attivando una doppia interfaccia SPI. Vista la complessità della cosa si è pertanto preferito optare per una soluzione di invio dati “a richiesta”.

### 5.4.2 Protocollo di comunicazione master/PC

Ben più complesso e articolato è invece il protocollo di comunicazione tra master e PC. I messaggi possono essere suddivisi in tre macrocategorie:

- **messaggi di configurazione**, inviati dal PC al circuito;
- **messaggi di comando**, inviati da PC a circuito e viceversa;
- **messaggi di servizio**, inviati da circuito a PC.

Un possibile scambio di messaggi è esemplificato nello schema di Figura 5.8. Il comando è formato dalla parola chiave *set*, seguita dal nome della variabile e il valore che si vuole impostare. Tutti i comandi devono essere terminati

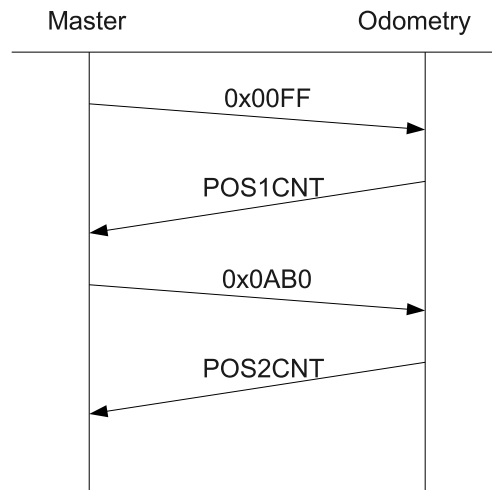


Figura 5.7: Diagramma dei messaggi scambiati tra scheda master e odometry

da un *line feed*. Se il comando è stato ricevuto correttamente e la variabile impostata, il circuito risponde al PC con un *OK*, altrimenti *ERRORE*. Il PC può anche decidere se ricevere o meno i dati dell'odometria, col comando *get odo [y|n]*, che abiliterà l'invio automatico dei dati relativi all'odometria assieme agli altri valori.

### Lista dei comandi

Di seguito vengono elencati i messaggi di configurazione ammissibili con i relativi valori:

- **set l v1**, larghezza del robot espressa in metri (e.g. set l 0.523);
- **set g v1**, numero di tic al giro per ruota (e.g. set g 180);
- **set c v1 v2**, circonferenza della ruota sinistra (v1) e di quella destra (v2) espresse in metri (e.g. set c 0.98 0.97);
- **set p v1 v2 v3 v4 v5 v6**, valori del PID ( $K_p$ ,  $K_i$ ,  $K_d$ ) per la ruota sinistra (v1, v2, v3) e destra (v4, v5, v6) (e.g. set p 0.6 0.005 0.0625 0.6 0.005 0.0625);
- **set m v1 v2 v3 ...**, numero di pesi (v1) per la media mobile e rispettivi valori; la somma dei pesi deve essere pari a 100 (e.g. set m 4 25 25 25 25);

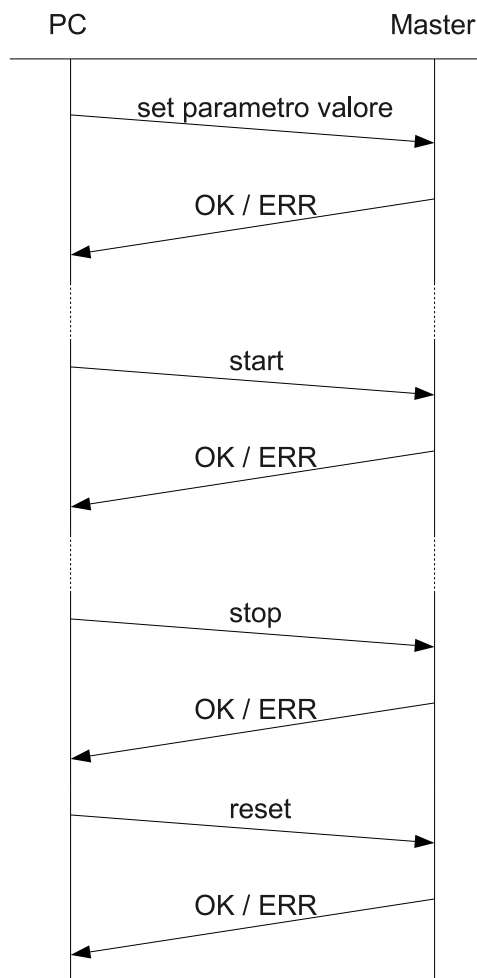


Figura 5.8: Diagramma dei messaggi scambiati tra scheda master e odometry

- **set v v1 v2**, massima velocità raggiungibile nella direzione avanti (v1) e indietro (v2) espresse in [m]/[s] (e.g. set v 1.0 0.6);
- **set w v1**, massima velocità angolare espressa in [rad]/[s] (e.g. set w 1);
- **set dac v1 v2 v3 v4 v5 v6**, valori max, center e min per la ruota sinistra (v1, v2, v3) e destra (v4, v5, v6), espressi in Volt (e.g. set dac 3.75 2.50 1.30 3.75 2.50 1.30);
- **set adc v1 v2**, differenza minima (v1) e massima (v2) per le tensioni differenziali della leva del joystick, espressa in Volt (e.g. set adc -2.50 2.50);
- **set ctrl pid|none**, abilita o disabilita il controllore PID;
- **go**, fa partire l'esecuzione del software di controllo.

Dopo avere fatto partire il software, sono inoltre ammissibili i seguenti messaggi di comando:

- **stop**, ferma l'esecuzione del software di controllo;
- **reset**, resetta i parametri di configurazione, necessita di uno *stop* per essere eseguito;
- **speed v1 v2**, valido solo se sono attivi i controllori PID, imposta il comando in velocità del robot, velocità tangenziale (v1) e velocità angolare (v2);
- **stick v1 v2**, valido solo se non sono attivi i controllori PID, imposta il comando in velocità del joystick, asse y (v1) e x (v2);
- **m auto|manual**, imposta la modalità di funzionamento, tra automatica e manuale.

Dal circuito al PC sono invece previsti i seguenti messaggi “informativi”, inviati in automatico durante il normale funzionamento del software:

- **P #pacc**, indica un numero progressivo a 16bit usato per rappresentare il numero di cicli;
- **O #left #right**, attivo solo se *get\_odo y*, indica i tic per la ruota sinistra e destra;
- **J y x**, indica il valore differenziale letto dagli ADC sugli assi *y* e *x*;



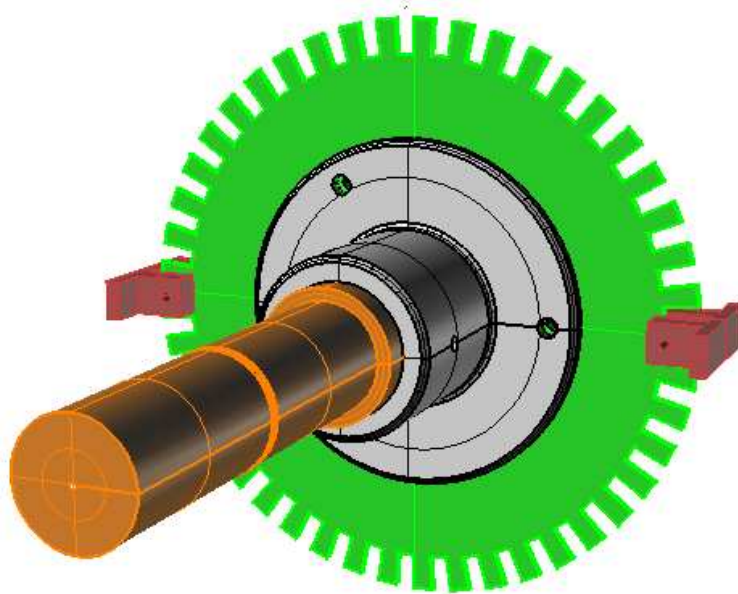


Figura 5.9: Rendering tridimensionale del sistema montato sull'asse

- **C left right**, attivo solo se *set ctrl pid*, indica il comando proposto dai controllori PID;
- **M m|a**, indica il modo di funzionamento della carrozzina, se (m)anuale o (a)utomatico.

## 5.5 Encoder

Il progetto per la realizzazione del sistema di odometria può idealmente essere diviso in due parti: da un lato, la parte che comprende disco e mozzo, solidali con l'asse della ruota; dall'altra, la parte solidale al telaio della carrozzina con i sensori a forcella e il profilo per sorreggerli.

### 5.5.1 Disco e mozzo

Una volta scelto di utilizzare il disco, si è posto il problema di rendere solidali disco e asse delle ruote, di modo che la misura sul disco fosse una misura precisa della rotazione dell'asse. In particolare, ci si è posti l'obiettivo di evitare qualsiasi tipo di slittamento. Questo problema è stato agevolato notevolmente dal fatto che l'asse ruota a velocità piuttosto contenute (da una

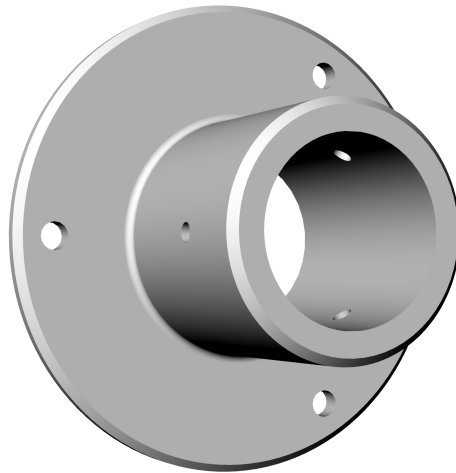
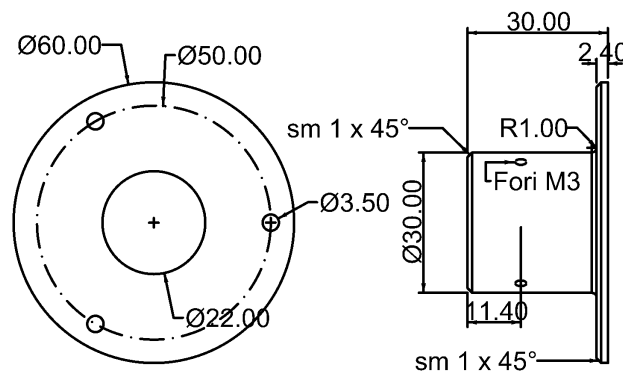


Figura 5.10: Progetto e rendering tridimensionale del mozzo

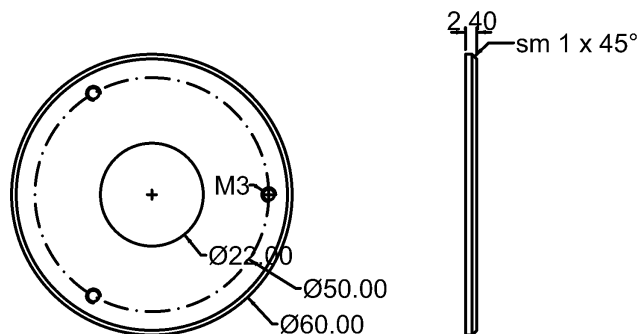


Figura 5.11: Progetto della rondella di fissaggio

stima empirica, la velocità massima raggiungibile a carrozzina sollevata da terra è di circa 120 giri al minuto, che corrispondono pressappoco a 7km/h). Il disco inoltre è piuttosto sottile e leggero, altro punto che rende più agevole il compito, anche se pone il problema di come affrancarlo all'asse. Per questo si è deciso di realizzare un mozzo in alluminio, illustrato in Figura 5.10, da infilare sull'asse e a esso affrancarlo mediante tre grani. Il disco viene invece fissato al mozzo utilizzando una rondella (Figura 5.11) d'acciaio e delle viti. Il disco (il cui progetto è illustrato in Figura 5.12) è anch'esso di alluminio spesso 1mm e di diametro esterno 100mm. I denti, realizzati lungo la sua circonferenza, sono profondi 6mm, per evitare che si possano piegare facilmente durante l'uso. Il numero di questi denti è stato inoltre scelto di modo da poter realizzare comunque una struttura solida, ma al contempo garantire buona affidabilità anche a bassi regimi di rotazione. Nello specifico, sono stati realizzati 45 denti, che danno origine a 45 fronti di salita e 45 di discesa, raddoppiati per l'adozione di un sistema in quadratura, per un totale di 180 tic per giro. In questa configurazione, considerando la temporizzazione di 20ms del circuito, la velocità minima rilevabile “in continua” (i.e. un tic per ogni ciclo di campionamento) è di:

$$\frac{\Delta tic}{\Delta t \cdot tic_{giro} \cdot circ}$$

Sostituendo, otteniamo che:

$$\frac{1}{0.02 \cdot 180 \cdot 0.98} = 0.27 \text{m/s}$$

In realtà è possibile rilevare anche velocità inferiori, interpolando i dati e facendo la media delle velocità rilevate nei precedenti istanti di tempo, motivo per cui si è deciso di calcolare la media mobile delle velocità.

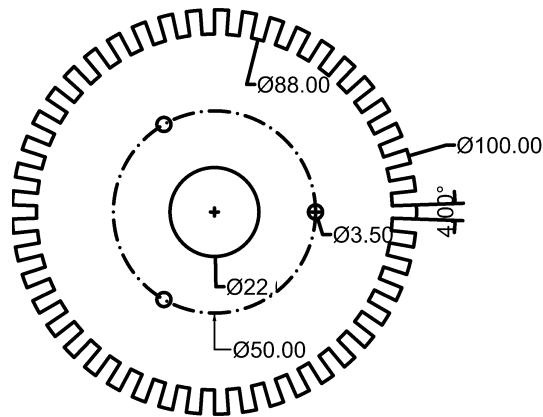


Figura 5.12: Progetto del disco dell'encoder

### 5.5.2 Profilo di sostegno sensori

Il profilo di sostegno per i sensori ottici è realizzato anch'esso in alluminio. Viene agganciato al telaio della carrozzina sfruttando le viti che fissano il motore al telaio stesso e ha una forma a "U", sui cui lati verticali sono posizionati i sensori. Questi rilevatori sono saldati su un circuito millefori e fissati al profilo in alluminio con viti e dadi; il foro per la vite ha forma di asola per permettere la messa in quadratura del sensore. Questa operazione va svolta servendosi di un oscilloscopio, analizzando le forme d'onda in uscita dai sensori di una ruota mentre la ruota gira. Si considerano in quadratura quando le due onde quadre sono sfasate circa di  $90^\circ$ .

## 5.6 Architettura MRT

### 5.6.1 Implementazione preodometria

Nella versione del precedente sistema di controllo, il cui schema è illustrato in Figura 5.13, i membri *VisionExpert* e *MTiExpert* inviavano rispettivamente i messaggi di `MSG_FROM_IMU` e `MSG_FROM_VISION_POS` al membro *PoseExpert*, il quale dialogava coi membri *SequencerExpert* e *BrianExpert*. I messaggi scambiati tra *Pose* e *Sequencer* riguardavano la posizione corrente (`MSG_FROM_POSE_CURRENT_POS`) oppure l'eventuale assenza, dovuta all'assenza di marcatori (`MSG_FROM_POSE_UNKNOWN_CURRENT_POS`). Similmente, i messaggi inviati dal *Pose* a *Brian* riguardavano l'attuale velocità (`MSG_FROM_POSE_CURRENT_SPEED`) o un errore nella sua rilevazione (`MSG_FROM_POSE_UNKNOWN_CURRENT_SPEED`).

Una volta elaborata l'azione di comando, il membro *BrianExpert* in-

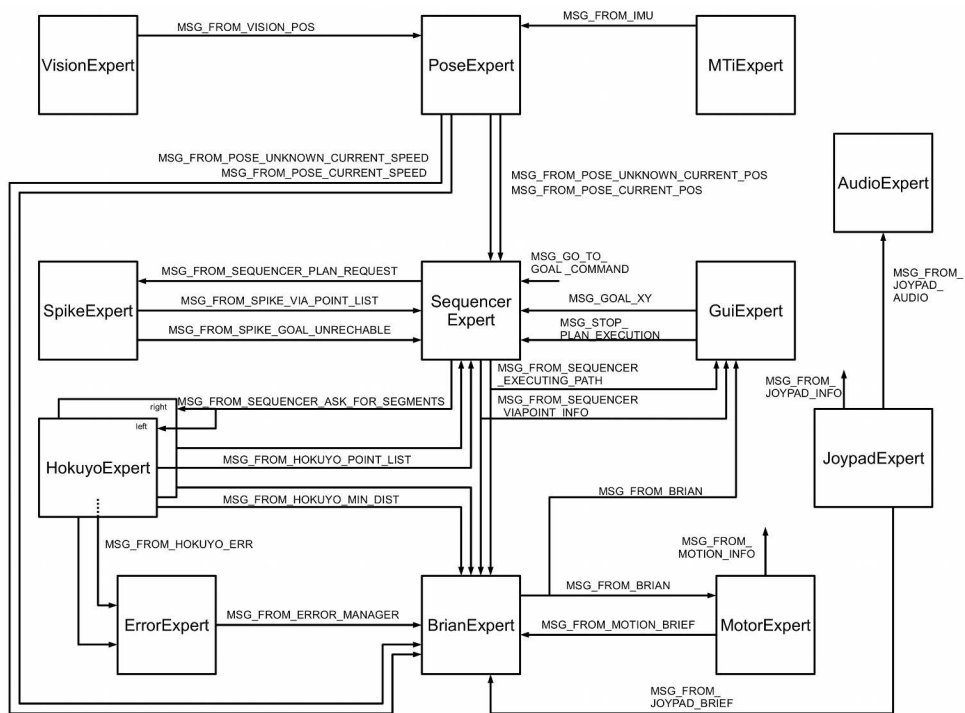


Figura 5.13: Architettura del sistema di controllo prima dell'introduzione dell'odometria

viava un messaggio di comando al *MotorExpert* (MSG\_FROM\_BRIAN), il quale provvedeva a comunicare con il circuito per eseguire l'azione proposta da Brian, senza però avere un riscontro sull'effettiva esecuzione della stessa. Come già detto, una stima sulla velocità del robot arrivava dal membro *MTiExpert*, poi abbandonata in seguito ai problemi esposti nel capitolo precedente.

### 5.6.2 Modifiche relative all'odometria

L'introduzione dell'odometria ha portato alcuni cambiamenti all'architettura del software di controllo, la cui nuova versione è illustrata in Figura 5.14. In primo luogo la rimozione del sensore XSens ha portato all'eliminazione anche del relativo membro *MTiExpert*. I dati relativi all'odometria provenienti dal circuito sono gestiti dal membro *MotorExpert*, esteso per questa funzionalità. È poi quest'ultimo a comunicare i messaggi MSG\_FROM\_ODO\_SPEED e MSG\_FROM\_ODO\_POS al membro *PoseExpert*, in cui vengono inviati rispettivamente la velocità del robot e la sua pose calcolata con l'odometria. Viene reso disponibile anche il dato riguardante la velocità delle singole ruote (MSG\_FROM\_ODO\_WHEELS\_SPEED), anche se non vi è alcun membro al momento che lo utilizzi.

Anche il membro *PoseExpert* ha subito delle variazioni. È stato infatti implementato un filtro di Kalman al suo interno per integrare i dati provenienti dall'odometria con quelli della visione, fornendo così una misura più accurata della posizione attuale del robot. Come si nota anche dallo schema, sono stati rimossi i messaggi di posizione e velocità sconosciuti; al loro posto si è preferito implementare un sistema di qualità gestito a livello fuzzy. Questo sistema prevede che assieme alla posizione venga inviato anche un dato che rappresenta la covarianza del filtro di Kalman o, più semplicemente, con quale intervallo di precisione è nota la posizione del robot. Più basso è questo valore e maggiore sarà la qualità della posizione. Come spiegato nella Sezione 4.6, la qualità della posizione è limitata da un upper bound fissato che viene raggiunto nel caso in cui manchino gli aggiornamenti dal membro *VisionExpert* o il circuito dell'odometria cessi di inviare dati al membro *MotorExpert*.

### 5.6.3 Altre modifiche

Al sistema di controllo sono state applicate anche alcune modifiche che non dipendono direttamente dall'introduzione dell'odometria e che non riguardano il presente lavoro; sono qui riportate solo per completezza. Come si può notare dallo schema, infatti, è stato rimosso il modulo *ErrorExpert* che

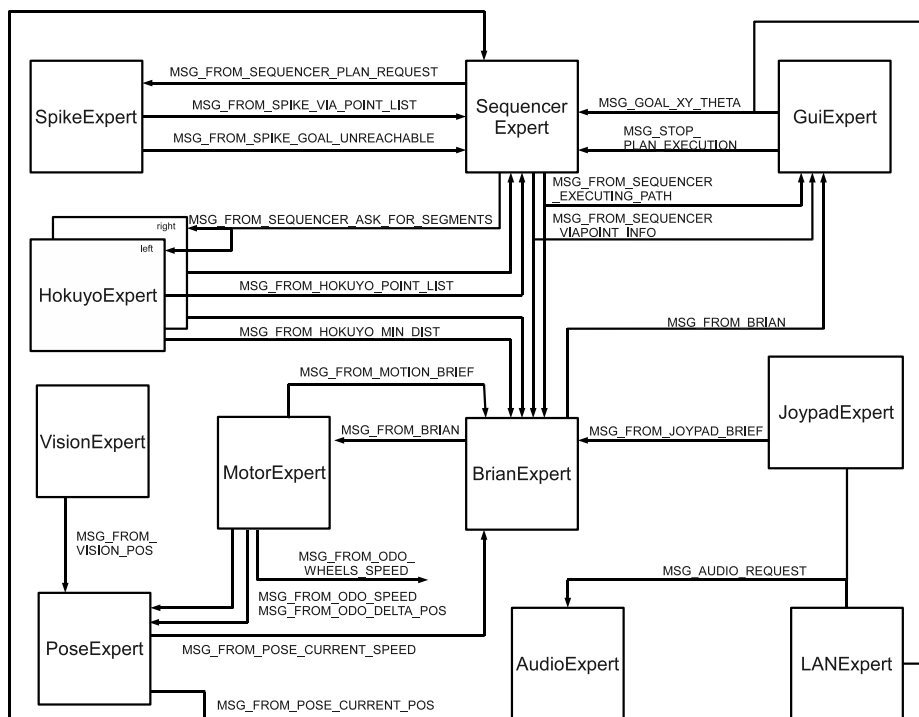


Figura 5.14: Architettura del sistema di controllo dopo l'introduzione dell'odometria

si occupava della gestione degli errori dei sensori laser. Anche per gli scanner Hokuyo, infatti, si è deciso di introdurre il sistema di qualità gestito a livello fuzzy, nelle stesse modalità utilizzate per la posizione e la velocità.

È stato invece introdotto un nuovo esperto, denominato *LANExpert*, preposto allo scambio di messaggi attraverso una connessione Ethernet. Questo modulo è stato implementato al fine di permettere l'uso di un'interfaccia cervello-computer (Brain Computer Interface, BCI) per comandare la carrozzina attraverso di essa. Il modulo può tuttavia essere usato anche per qualsiasi altro sistema posto in rete che necessiti di un collegamento remoto.



## Capitolo 6

# Realizzazioni sperimentali e valutazione

*“Luke Skywalker: Tu vuoi l'impossibile!  
[Yoda riesce a tirare fuori la navicella]  
Luke Skywalker: Non posso crederci!  
Yoda: Ecco perché hai fallito.”*

L'Impero colpisce ancora

In questo capitolo sono analizzati i risultati ottenuti, sia dal punto di vista della realizzazione e montaggio del sistema di odometria basato su encoder che dal punto di vista del comportamento del controllore nel realizzare il controllo in velocità della carrozzina.

### 6.1 Schede

Per i test illustrati in questo Capitolo è stato realizzato un circuito di prova su *bread board*, visibile in Figura 6.1. La realizzazione finale su circuito stampato è in fase di realizzazione al momento della scrittura. Alcuni test preliminari sono inoltre stati condotti utilizzando una copia della scheda odometry realizzata su circuito millefori (vedi Figura 6.2) collegata tramite porta seriale al computer; in questo caso la parte di controllo è stata svolta dal precedente circuito collegato anch'esso al PC tramite seriale.

#### 6.1.1 Ritardi

Il ritardo dovuto all'elaborazione software e al DAC è stato rilevato fornendo in ingresso agli ADC un'onda quadra prodotta dallo stesso PIC e

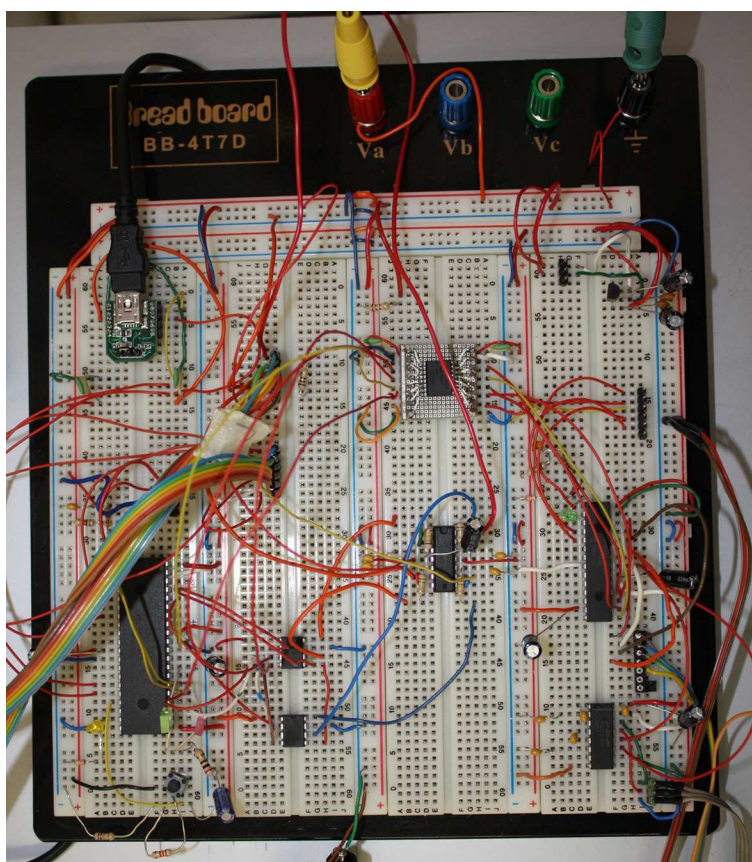


Figura 6.1: Circuito provvisorio contenente la parte "master" (a sinistra) e quella "odometry" (a destra)

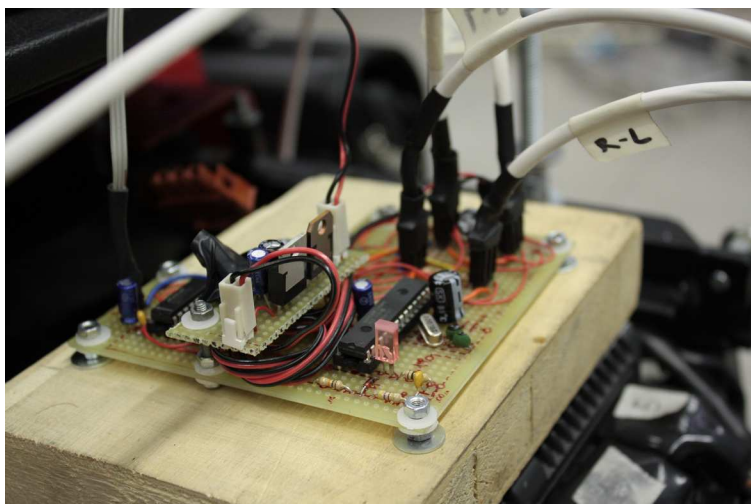


Figura 6.2: Circuito provvisorio della scheda odometry

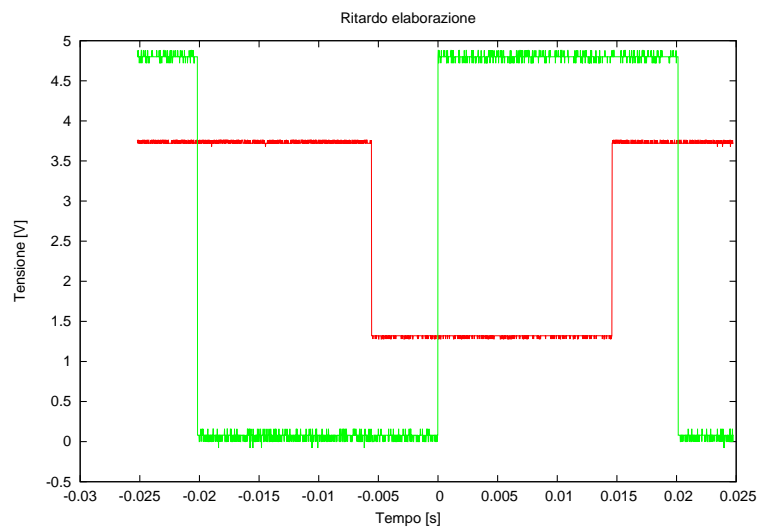


Figura 6.3: Ritardo introdotto dall'elaborazione software e dal DAC.

misurando con un oscilloscopio l'uscita dei DAC, a monte del filtro. Il grafico ottenuto è visibile in Figura 6.3, dove l'onda rossa rappresenta l'ingresso, mentre quella verde l'uscita. Si noti come, giustamente, l'uscita sia compresa nell'intervallo  $[1.25\text{V}, 3.75\text{V}]$ , nonostante l'ingresso vari nell'intervallo  $[0\text{V}, 5\text{V}]$ . Misurando la distanza tra le due curve dopo varie prove, il ritardo è risultato essere costante e pari a circa 15ms. Utilizzando invece un generatore di forme d'onda è stato testato il ritardo generato dal filtro passa-basso assieme all'amplificatore operazionale, utilizzando come campione un'onda sinusoidale di frequenza 2Hz, come da Figura 6.4. Tale frequenza è stata scelta come caso "limite", in quanto è poco probabile, anche se non impossibile, che una persona controlli la carrozzina muovendo la leva a una simile velocità. È invece un buon test nel caso di controllo automatico da PC. Il ritardo generato in questo caso è nell'ordine dei 20ms.

Si noti poi il taglio a 4V, dovuto all'operazionale Texas TLC274. Da specifiche questo integrato è infatti in grado di fornire una tensione minima tipica di 0V, ma la minima tensione massima è di 3.2V, con un valore tipico che si assesta a 3.8V. Questa scelta si rivela un po' "azzardata", dato che in alcuni casi l'operazionale può non essere in grado di fornire la tensione richiesta dalla leva (che può arrivare fino a 3.75V). Tuttavia questo operazionale era installato sulla leva del joystick, pertanto si è deciso di utilizzarlo anche per il circuito, sapendo però che è sempre possibile sostituirlo, essendo pin-to-pin compatibile con diversi altri modelli rail-to-rail, come ad esempio il Microchip MCP6024.

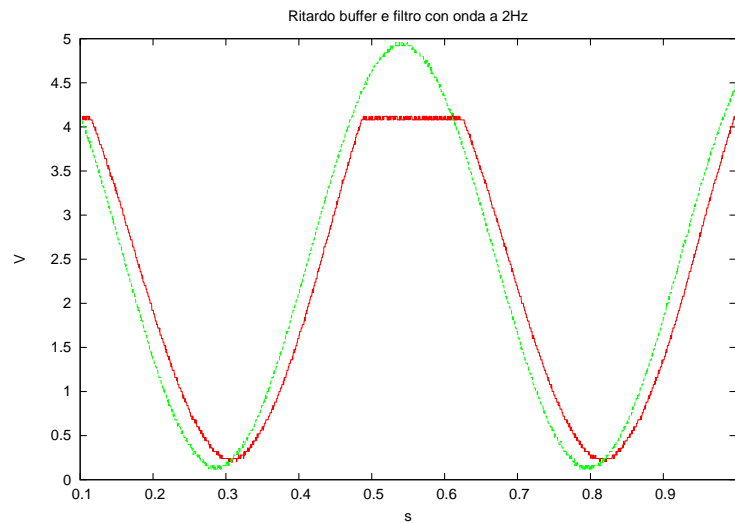


Figura 6.4: Ritardo introdotto dall'opamp e dal filtro passa-basso

## 6.2 Encoder

Per la realizzazione del sistema di odometria sono stati prodotti in officina meccanica i dischi dentati e i mozzi di supporto. Il pezzo di sostegno per i sensori ottici è stato invece realizzato a mano piegando una piattina di alluminio; questa è stata fissata a due supporti, sempre in alluminio, a loro volta affrancati al telaio della carrozzina per mezzo delle stesse viti che sostengono i motori. I sensori ottici sono stati saldati su di un circuito millefori, fissato al sostegno per mezzo di una vite e in grado di scorrere lungo il sostegno stesso. Questo accorgimento si è rivelato indispensabile per poter mettere in quadratura di due sensori. In Figura 6.5 è possibile vedere la realizzazione finale del sistema di odometria installato sulla ruota sinistra (per la ruota destra la realizzazione è identica).

### 6.2.1 Quadratura degli encoder

Come visto nella Sezione 3.2.2, la messa in quadratura degli encoder è un'operazione fondamentale per il loro corretto funzionamento. Oltre a posizionare i sensori come indicato nel paragrafo suddetto, si è provveduto a raffinare il setup facendo girare le ruote della carrozzina a varie velocità e spostando i sensori lungo la piattina di sostegno misurando i cambiamenti con un oscilloscopio. In Figura 6.6 è raffigurato un particolare della piattina di sostegno dei sensori ottici, mentre in Figura 6.7 si mostra il risultato della quadratura ottenuto dopo la messa a punto. Si nota che le due onde

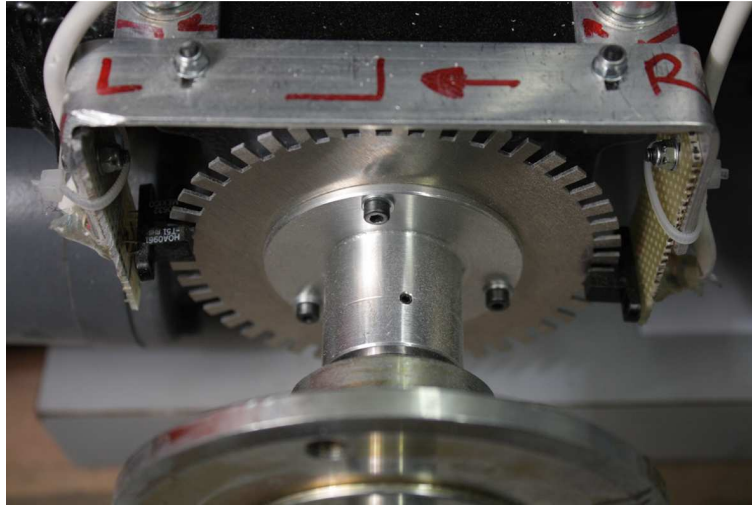


Figura 6.5: Realizzazione del sistema di odometria



Figura 6.6: Particolare della piattina di sostegno dell'encoder a forcella

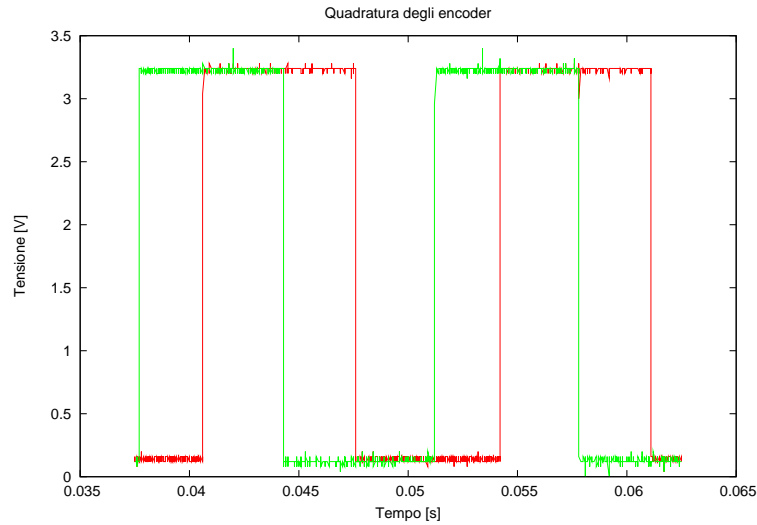


Figura 6.7: Quadratura degli encoder per la ruota sinistra

non sono perfettamente sfasate di  $\pi/2$ , ma già in questa configurazione lo sfasamento tra le due onde è sufficiente per renderle distinguibili e, di conseguenza, per rilevare correttamente la velocità e il senso di rotazione della ruota.

### 6.2.2 Calibrazione dell'odometria

Per poter calcolare la cinematica della carrozzina utilizzando i dati forniti dall'odometria occorre sapere la larghezza del robot e la circonferenza delle ruote, come visto nelle Sezioni 3.3.1 e 5.5.1. La larghezza del robot è stata misurata empiricamente ed è risultata essere di 50cm. La circonferenza delle ruote dipende invece dalle condizioni ambientali (i.e., temperatura, pressione, ecc. . .) e dal carico a bordo della carrozzina. Per poterla misurare in condizioni reali si è quindi percorsa una distanza nota con la carrozzina, misurando i tic contati dall'odometria e dividendo per il numero di tic di un giro moltiplicato per la distanza percorsa. In formule,

$$circ = \frac{\#tic}{\#tic_{giro} \cdot dist}$$

Questa misura è stata effettuata 10 volte e poi mediata; il risultato così ottenuto è stato di 98cm, contro 1m misurato a mano.

In letteratura esistono molti sistemi di calibrazione per l'odometria più complessi e precisi di quello utilizzato, ma si è ritenuto più che sufficiente questo approccio ai fini del controllo in velocità della carrozzina.

Tabella 6.1: Calibrazione del controllore PID secondo il metodo Ziegler-Nichols

	$K_P$	$K_I$	$K_D$
P	$0.5 \cdot \bar{K}_P$	-	-
PI	$0.45 \cdot \bar{K}_P$	$0.8 \cdot \bar{T} \cdot dt$	-
PID	$0.6 \cdot \bar{K}_P$	$0.5 \cdot \bar{T} \cdot dt$	$0.125 \cdot \bar{T}$

### 6.2.3 Calibrazione del controllore PID

Un controllore PID necessita di essere calibrato per poter funzionare correttamente. Nello specifico, è necessario fornire dei valori corretti per le costanti proporzionale, integrativa e derivativa; in caso contrario il sistema risulterà instabile e poco utilizzabile. Non conoscendo il modello della carrozzina, né quello dei suoi motori, è stato utilizzato un metodo di calibrazione on-line in anello chiuso, noto come metodo di Ziegler-Nichols. Il metodo si suddivide in una serie di passi: preliminarmente si attiva la sola azione proporzionale, innalzando il coefficiente  $K_P$  fino a portare il sistema al limite della stabilità, ovvero quando l'uscita è in oscillazione permanente con periodo  $\bar{T}$ . In questo caso l'uscita del sistema è stata misurata analizzando la velocità delle ruote rilevata con l'odometria, il cui grafico è illustrato in Figura 6.8. Il valore così ottenuto viene detto *guadagno critico*  $\bar{K}_P$ . I valori delle costanti vengono quindi calcolati a seconda che si desideri utilizzare un sistema solo proporzionale, proporzionale e integrativo o PID completo, come indicato nella Tabella 6.1.

In Figura 6.9 è invece visibile la risposta del sistema con controllore PID calibrato. Dato il valore precedente di  $\bar{K}_P = 1$ , i valori ottenuti secondo la Tabella 6.1 sono:  $K_P = 0.6$ ,  $K_I = 0.01383$  e  $K_D = 0.06917$  per la ruota sinistra,  $K_P = 0.6$ ,  $K_I = 0.014$  e  $K_D = 0.07$  per la ruota destra, dove  $dt = 0.05$  e  $\bar{T}_L = 0.55$  per la ruota sinistra e  $\bar{T}_R = 0.56$  per la ruota destra.

## 6.3 Test sperimentali

Al fine di valutare il comportamento del sistema e apprezzare i miglioramenti dovuti all'introduzione del sistema di controllo basato su odometria oggetto di questa tesi, è stata svolta una serie di test in ambiente controllato i cui risultati vengono proposti di seguito. In prima istanza sono stati messi in quadratura gli encoder, a cui è seguita l'operazione di calibrazione dell'odometria e dei PID, come illustrato nei Paragrafi precedenti.

Per verificare la bontà del controllore PID sono stati impartiti da joypad

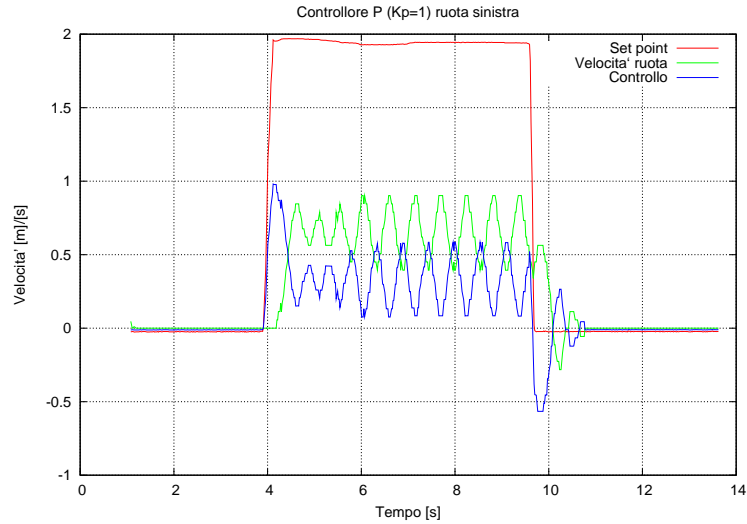


Figura 6.8: Calibrazione del controllore PID con guadagno critico  $\bar{K}_P = 1$

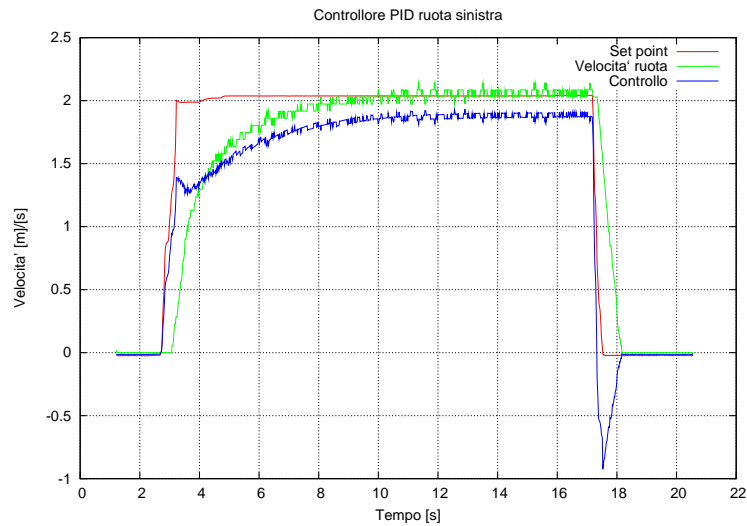


Figura 6.9: Controllore PID calibrato



alcuni setpoint e si è poi analizzata la risposta della carrozzina. In Figura 6.10 è possibile notare il comportamento del sistema con odometria nel raggiungere vari setpoint richiesti: la linea rossa rappresenta il setpoint desiderato, mentre la curva blu e la curva verde rappresentano, rispettivamente, il comando in velocità proposto dal controllore PID e la velocità letta dagli encoder. Come si può notare, il valore di controllo varia in funzione della velocità effettivamente rilevata; nel caso di una brusca frenata, ad esempio, il valore di controllo proposto sarà molto più basso dell'effettivo valore da raggiungere, per ridurre il tempo di frenata (nel grafico il punto di frenata è indicato dai fronti di discesa del setpoint). La velocità riportata dall'odometria è stata ottenuta con un filtro a media mobile di cinque campioni sul segnale in uscita dagli encoder. Questo accorgimento è stato preso dopo aver notato come valori troppo "spigolosi" della velocità portassero il controllore a produrre un'uscita poco lineare e di conseguenza destabilizzare tutto il sistema.

Dopo aver appurato il corretto funzionamento del controllore, si è proceduto a percorrere con il robot alcune traiettorie casuali in un ambiente parzialmente coperto da marcatori artificiali, di modo da poter analizzare il comportamento del sistema anche in assenza di informazioni relative alla visione. In Figura 6.11 è mostrata la traiettoria percorsa della carrozzina, in cui la posizione è ottenuta integrando con un filtro di Kalman Esteso i dati provenienti dalla visione artificiale e quelli dell'odometria. La legenda colore raffigurata a destra rappresenta la qualità della pose, ottenuta come descritto nel Paragrafo 5.6.2: il nero indica qualità ottima, mentre il giallo indica qualità pessima, fissata in questo caso a un errore di 3m. Come si può notare dalla Figura 6.11(a), i dati della vision risultano radi e, in alcune zone, non essendoci marcatori artificiali, addirittura assenti. In questi casi la traiettoria viene calcolata considerando esclusivamente l'odometria, anche se, come si vede, la qualità peggiora con l'allontanarsi dall'ultima posizione "coperta" dalla visione. Quando la carrozzina rientra in un'area coperta (Figura 6.11(b)), il sistema corregge la deriva dell'odometria con la misura assoluta della visione. In questo caso l'errore effettivo è di circa 0.8m, ben al di sotto della stima di qualità che indicava come errore massimo 3m.

Questo test mette anche in luce l'impossibilità di calcolare la traiettoria della carrozzina utilizzando esclusivamente uno solo dei due sistemi. Se, da un lato, la vision non è sempre disponibile e rappresenta un dato poco fitto, dall'altro l'odometria, se lasciata senza aggiornamenti, porta a una significativa deriva. Questo concetto viene illustrato meglio in Figura 6.12, dove si mostrano le traiettorie seguite dalla carrozzina, calcolate utilizzando i dati relativi all'odometria (in blu) e alla vision (in rosso). Per facilitare la

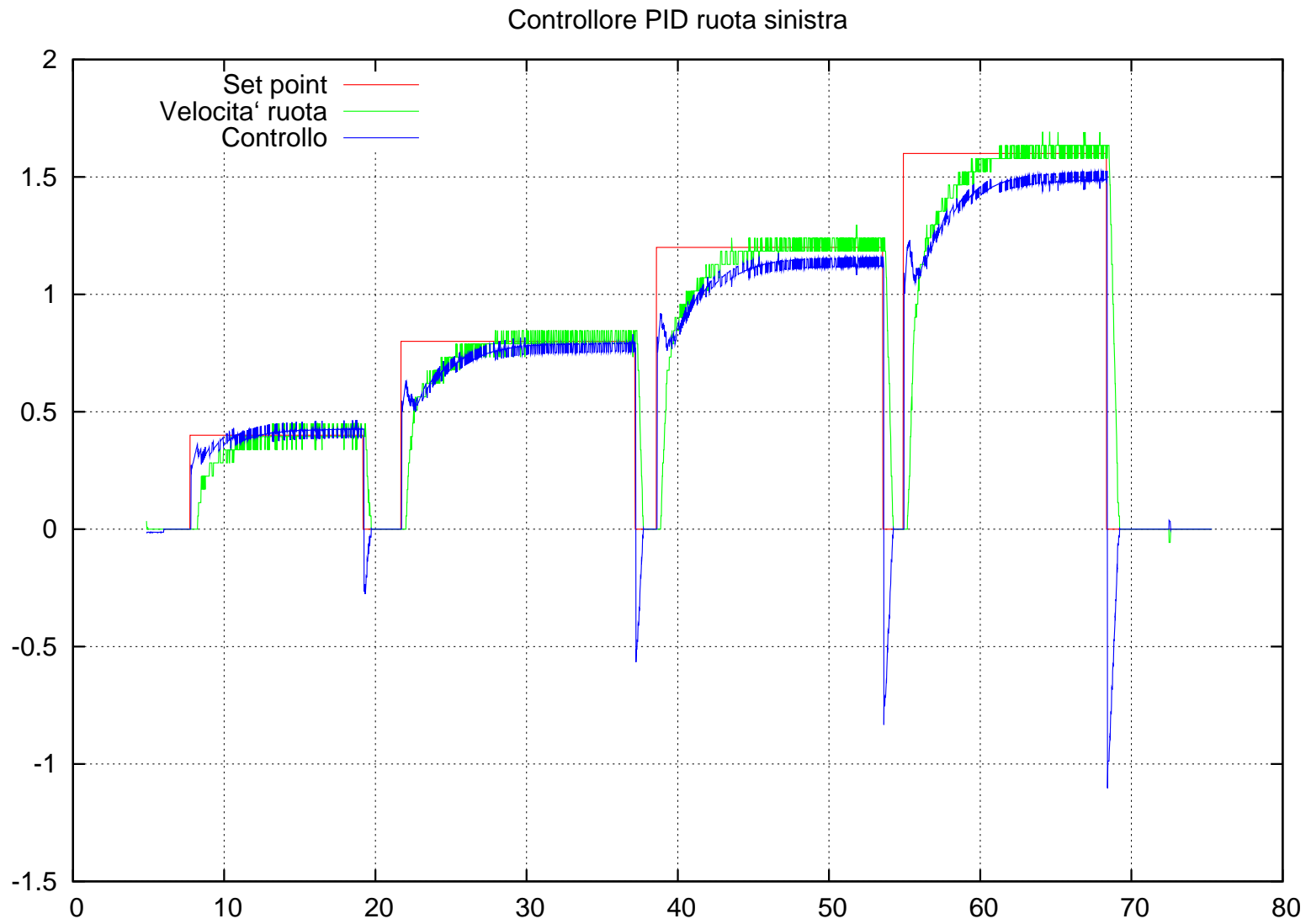


Figura 6.10: Controllo della carrozzina con controllori PID

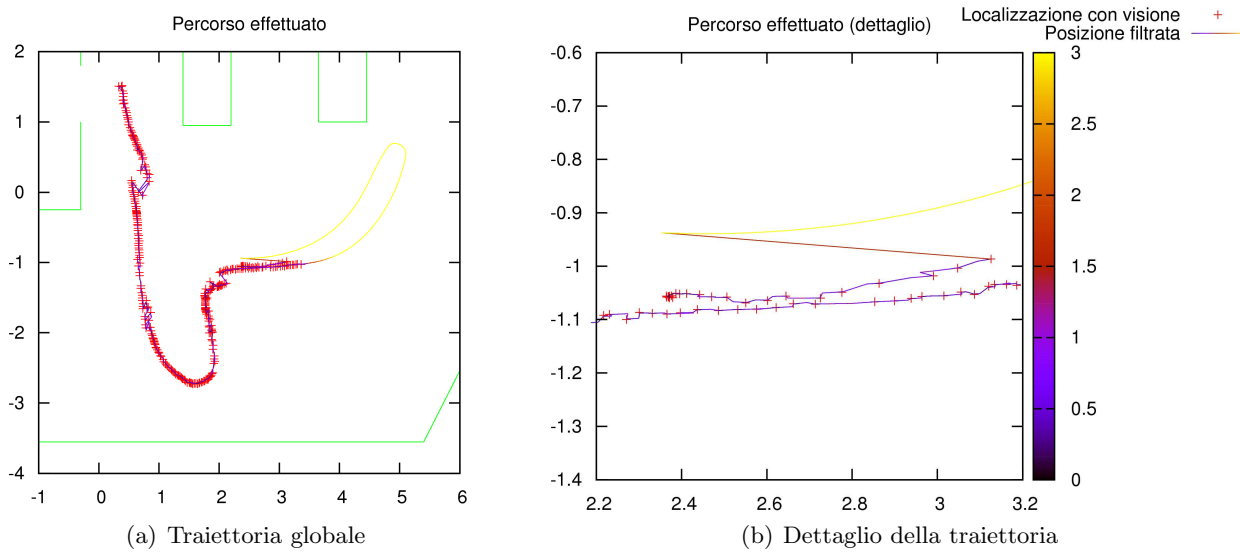


Figura 6.11: Traiettoria del robot filtrata con Kalman

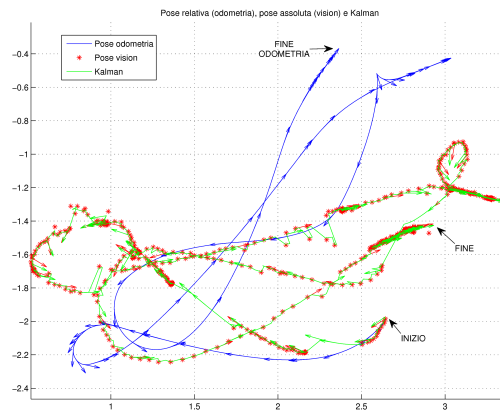


Figura 6.12: Traiettoria stimata con pose odometria e pose vision

comprensione, i dati relativi all'odometria sono stati riproiettati sul punto di origine della vision. Si nota che la posizione fornita dall'odometria risulta errata fin dai primi metri, per terminare nel punto finale con un errore di posizionamento di circa 1m. Utilizzando invece un filtro di Kalman per integrare i due dati (mostrato in verde), la traiettoria risulta più fitta e precisa.

## Capitolo 7

# Conclusioni e sviluppi futuri

*“Quando la sorte t’è contraria  
e mancato t’è il successo  
smetti di far castelli in aria  
e va’ a piangere sul...”*

Igor in “Frankenstein Junior”

In questo capitolo si svolge l’analisi dei punti di forza e si presentano possibili miglioramenti del sistema realizzato, basandosi sui risultati ottenuti (illustrati nel Capitolo precedente). Durante l’analisi sono proposte alcune soluzioni che possono rafforzare i punti critici.

### 7.1 Conclusioni

Lo scopo di questa tesi è stato realizzare un sistema di controllo basato su odometria che estendesse le funzionalità di una carrozzina robotica per disabili motori sviluppata presso il Politecnico di Milano. Durante questo lavoro si sono susseguite le seguenti fasi:

- Scelta dei sensori odometrici e dei componenti hardware per il sistema di controllo;
- Progettazione e realizzazione di una struttura di supporto per i sensori odometrici;
- Sviluppo di un sistema di controllo basato su microprocessore PIC in grado di interfacciarsi con il joystick della carrozzina;
- Test del sistema realizzato in ambienti controllati.

Grazie a questa soluzione di controllo il sistema è ora più fluido nei movimenti e reattivo ai comandi. La chiusura dell'anello di controllo ha permesso infatti di rendere più stabile il sistema, oltre che più sicuro. La funzione di evitamento ostacoli ha infatti tratto benefici dall'odometria, in quanto la conoscenza della velocità consente di avere regole di evitamento più semplici e dettagliate.

Sebbene i miglioramenti dovuti a queste estensioni siano stati significativi, rimangono ancora molte possibilità di sviluppo, sia per estendere il sistema con nuove funzionalità che per migliorare quelle esistenti.

## 7.2 Odometria

Il sistema odometrico è stato realizzato con sensori ottici che rilevano il movimento dell'asse della ruota tramite un disco dentato. Questo disco è stato realizzato in modo da fornire 180 tic di risoluzione al giro, un valore che consente di rilevare velocità minime di 0.27m/s, come visto nella Sezione 5.5.1. Alla massima velocità, invece, non sono stati riscontrati problemi nel rilevare tutti i tic. Volendo però raggiungere una maggiore precisione alle basse velocità, si potrebbe raddoppiare il numero di denti, portando così il numero di tic a 360, pari a 0.13m/s. In questo caso sarebbe però da curare l'aspetto meccanico, in quanto denti così sottili rischierebbero di spezzarsi o piegarsi facilmente, andando a danneggiare l'intero sistema.

Un'altra soluzione per aumentare la risoluzione alle basse velocità potrebbe consistere nell'installare un encoder meccanico sull'asse del motore, a monte del riduttore. Ciò comporterebbe però la modifica del freno magnetico attualmente montato sull'asse, come si è visto nella Sezione 4.3.

La soluzione scelta ha fornito risultati più che soddisfacenti nei test condotti in ambienti controllati, come ad esempio il laboratorio di sviluppo. Nei brevi test condotti in ambienti esterni (i.e., sui marciapiedi e per strada) si sono evidenziati alcuni limiti di questo sistema intrinseci alla sua natura. Quella che viene rilevata è infatti la velocità delle singole ruote, che in alcuni casi può non corrispondere all'effettiva velocità del robot. Avvallamenti, buche e in generale le varie asperità dell'asfalto hanno infatti inficiato la misura dell'odometria, tanto da suggerirne una diversa implementazione qualora si decidesse di sviluppare un sistema in grado di muoversi anche in ambienti esterni. In questo caso sarebbe possibile integrare la misura dell'odometria con informazioni differenti, derivate, ad esempio, da sistemi di visione o da scanner laser e calcolare la velocità basandosi su di esse.

## 7.3 Joystick

Il presente lavoro è stato sviluppato sulla base del joystick in dotazione alla carrozzina, il modello DX REM-34 prodotto da Dynamic Controls. Altri joystick, sia dello stesso che di altri produttori, potrebbero avere un funzionamento differente; la scheda di controllo è infatti realizzata per interagire con dispositivi a quattro canali in configurazione differenziale e con tensioni comprese tra 0 e 5V. Se è possibile agire via software per implementare piccole variazioni (e.g., differenti configurazioni differenziali o valori di tensione leggermente diversi - anche se sempre compresi nell'intervallo precedente), non è però possibile interfacciarsi a joystick che utilizzino sistemi a più canali o con tensioni maggiori. Per questo motivo sarebbe auspicabile potersi interfacciare direttamente col protocollo utilizzato per lo scambio di messaggi tra joystick e centralina dei motori, in quanto questo protocollo è sostanzialmente identico per tutti i joystick di un produttore e i maggiori *player* nel mercato sono soltanto due: Dynamic Controls e PG Drives Technology. Una connessione diretta consentirebbe anche di migliorare il controllo del sistema, inviando comandi più precisi, senza necessità di tradurli da un sistema di riferimento a un altro.

### 7.3.1 Joystick come dispositivo di interfaccia

Il joystick che normalmente viene utilizzato per guidare la carrozzina può essere impiegato anche come dispositivo per interfacciarsi a un computer o uno smartphone. In questo modo l'utente avrebbe la facoltà di interagire con sistemi che altrimenti sarebbero difficilmente raggiungibili o risulterebbero scomodi da utilizzare. Per operare in questa modalità sarebbe sufficiente progettare un quarto stato di controllo in cui i comandi della leva vengono ignorati ai fini del controllo del moto e usati invece come "mouse" virtuale. Il collegamento tra la scheda di controllo e il dispositivo da controllare potrebbe avvenire per mezzo di interfacce radio come ad esempio il Bluetooth, divenuto ormai di serie in molti smartphone e computer portatili.

## 7.4 Guida in ambienti esterni

Per come è stata sviluppata finora, la carrozzina è in grado di muoversi autonomamente soltanto in ambienti interni. Dovendone estendere le capacità di movimento anche ad ambienti esterni si dovrebbero prendere in considerazione vari punti, legati sia al sistema di localizzazione che alla struttura stessa del robot.

Partendo dal sistema di localizzazione, i marcatori artificiali che permettono alla carrozzina di localizzarsi nell'ambiente non sono utilizzabili anche all'esterno. Per questo si dovrebbe implementare un sistema di localizzazione basato su GPS, anche se le versioni economiche non garantiscono sufficiente precisione e quelle più affidabili (e.g., quelle utilizzate in marina o aeronautica) presentano dei costi elevati.

In ambienti esterni si dovrebbero prendere in considerazione anche gli elementi atmosferici, come ad esempio la pioggia: la struttura della carrozzina non è pensata per essere resistente all'acqua, pertanto si dovrebbero impermeabilizzare sia i sensori posti sulle ruote (per via dell'acqua raccolta dalle pozzanghere) sia il circuito di controllo e gli altri dispositivi (i.e., i computer) al fine di garantirne il corretto funzionamento anche in caso di precipitazioni atmosferiche. Il modello Rabbit alla base di Lurch non è pensato per funzionare sotto la pioggia, essendo sprovvisto di una cupola di protezione per il passeggero, tuttavia potrebbe verificarsi una situazione di emergenza che rischierebbe di danneggiare l'intero sistema.

## 7.5 Controllo della carrozzina tramite BCI

Uno sviluppo della carrozzina parallelo a questo tesi ha visto l'introduzione di un modulo BCI (*Brain-Computer Interface*) per comandare il moto della carrozzina [1]. Un modulo BCI rappresenta un'interfaccia diretta tra le attività cerebrali e un computer. Le tecniche di monitoraggio delle attività cerebrali sono numerose, tra cui la più semplice, meno invasiva ed economica è basata sull'uso di un elettroencefalografo e piccoli elettrodi posizionati sullo cuoio capelluto. Numerosi studi in questo ambito sono al vaglio della comunità scientifica [12] e permettono di identificare con diverse tecniche i comandi impartiti dall'utente. Innanzitutto è necessario distinguere tra tecniche basate sulla rilevazione di attività volontarie del cervello o su attività involontarie, chiamate anche *potenziali evocati*. Le attività volontarie del cervello possono essere sfruttate per realizzare un'interfaccia BCI in quanto esse sono modulabili dalla persona dopo un periodo di addestramento. Utilizzando tecniche basate sul riconoscimento dei potenziali evocati è stato possibile, infatti, proporre all'utente un'interfaccia grafica su cui sono visualizzate delle destinazioni predefinite nell'ambiente che vengono illuminate in ordine casuale. Quando l'utente si concentra su una delle frecce, la sua attività cerebrale involontaria viene stimolata dal lampeggio della destinazione considerata. Tale attività viene rilevata con l'uso dell'elettroencefalografo e portata come informazione di comando al computer che attiva la funzionalità di guida autonoma della carrozzina per raggiungere la destinazione.



È facile immaginare che la frequenza con cui possono essere impartiti i comandi tramite l'uso di interfacce BCI è molto più bassa rispetto a un sistema di guida tradizionale. Per questo motivo è notevole importanza il sistema di evitamento ostacoli, che permette di eseguire in sicurezza comandi di durata temporale medio-lunga.

Ausili di questo genere sono particolarmente utili per quei disabili che non sono in grado di comandare una carrozzina con un joystick, piuttosto che con la voce o con il movimento degli occhi o della testa. Nonostante i buoni risultati ottenuti con l'interfaccia BCI, infatti, i sistemi descritti poc'anzi permettono una maggiore potenza e flessibilità nel controllo che una BCI non riesce a raggiungere all'attuale stadio di sviluppo. Ci si augura tuttavia che futuri miglioramenti ne possano aumentare la potenza per renderla una strada percorribile da molte più persone.



# Bibliografia

- [1] Rossella Blatt, Simone Ceriani, Bernardo Dal Seno, Giulio Fontana, Matteo Matteucci, Marco Dalli, Davide Rizzi, and Davide Migliore. An Autonomous Wheelchair Driven by Event-Related Potentials. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009. IEEE. Accepted.
- [2] Paolo Bolzern, Riccardo Scattolini, and Nicola Schiavoni. *Fondamenti di controlli automatici*. McGraw-Hill, 1998.
- [3] J. Borenstein, H. R. Everett, L. Feng, S. W. Lee, R., and H. Byrne. *Where am I? Sensors and Methods for Mobile Robot Positioning*. 1996.
- [4] U. Borgolte, H. Hoyer, C. Bühler, H. Heck, and R. Hoelper. Architectural Concepts of a Semi-autonomous Wheelchair. *Journal of Intelligent and Robotic Systems*, 22:233–253, 1998.
- [5] G. Bourhis, K. Moumen, P. Pino, S. Rohmer, and A. Pruski. Assisted navigation for a powered wheelchair. In *International conference on systems, man and cybernetics*, 1993.
- [6] Simone Ceriani. Sviluppo di una carrozzina autonoma d’ausilio ai disabili motori. Master’s thesis, Politecnico di Milano, 2008.
- [7] Simone Ceriani and Marco Dalli. Guida all’installazione dell’interfaccia per PC. Laboratorio di Intelligenza Artificiale e Robotica A.A. 2006-2007, 2006.
- [8] Simone Ceriani and Marco Dalli. Robotizzazione carrozzina elettrica per disabili. Laboratorio di Intelligenza Artificiale e Robotica A.A. 2006-2007, 2006.
- [9] J. Connell and P. Viola. Cooperative control of a semi-autonomous mobile robot. In *Proceedings of the IEEE International Conference on Robotics And Automation*, 1990.

- 
- [10] Guillermo Del Castilloa, Steven Skaara, Antonio Cardenasb, and Linda Fehr. A sonar approach to obstacle detection for a vision-based autonomous wheelchair. *Robotics and Autonomous Systems*, 54:967–981, 2006.
- [11] D. Ding and R. A. Cooper. Electric powered wheelchairs. *IEEE Control Systems Magazine*, 25:22–34, 2005.
- [12] Guido Dornhege. *Increasing Information Transfer Rates for Brain-Computer Interfacing*. phdthesis, University of Potsdam, 2006.
- [13] F. Doshi and N. Roy. Spoken language interaction with model uncertainty: an adaptive human-robot interaction system. *Connection Science*, 20(4):299 – 318, December 2008.
- [14] Ferran Galán, Marnix Nuttin, Eileen Lew, Pierre W. Ferrez, Gerolf Vanacker, Johan Philips, Hendrik Van Brussel, and José del R. Millán. An Asynchronous and Non-Invasive Brain-Actuated Wheelchair. November 2007.
- [15] Xueliang Huo, Jia Wang, and Maysam Ghovanloo. Wireless control of powered wheelchairs with tongue motion using tongue drive assistive technology. pages 4199–4202, Aug. 2008.
- [16] L. Iezzoni, E. McCarthy, R. Davis, and H. Siebens. Mobility difficulties are not only a problem of old age. *Journal of General Internal Medicine*, 16(4):235–243, 2001.
- [17] I. Iturrate, M. Antelis, J. Minguez, and A. Kuebler. Non-Invasive Brain-Actuated Wheelchair based on a P300 Neurophysiological Protocol and Automated Navigation. 2009.
- [18] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [19] N.I. Katevas, N.M. Sgouros, S.G. Tzafestas, G. Papakonstantinou, P. Beattie, J.M. Bishop, P. Tsanakas, and D. Koutsouris. The autonomous mobile robot SENARIO: a sensor aided intelligent navigation system for powered wheelchairs. *IEEE Robotics & Automation Magazine*, 4:60–70, 1997.
- [20] Y. Kuno, N. Shimada, and Y. Shirai. Look where you’re going [robotic wheelchair]. *IEEE Robotics & Automation Magazine*, 10:26–34, 2003.

- 
- [21] A. Lankenau, O. Meyer, and B. Krieg-Brückner. Safety in robotics: the Bremen Autonomous Wheelchair. *Advanced Motion Control*, pages 524–529, 1998.
- [22] A. Lankenau and T. Röfer. A versatile and safe mobility assistant. *IEEE Robotics & Automation Magazine*, 8:29–37, 2001.
- [23] A. Lankenau, T. Röfer, and B. Krieg-Brückner. Self-Localization in Large-Scale Environments for the Bremen Autonomous Wheelchair. In Freksa C., Brauer W., Habel C., and Wender K.F., editors, *Spatial Cognition III: Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning*, volume 2685 of *Lecture Notes in Computer Science*, pages 34–61. Springer, 2003.
- [24] S.P. Levine, D.A. Bell, L.A. Jaros, R.C. Simpson, Y. Koren, and J. Borenstein. The NavChair Assistive Wheelchair Navigation System. *IEEE Transactions on Rehabilitation Engineering*, 7:443–451, 1999.
- [25] R. Madarasz, L. Heiny, R. Crompt, and N. Mazur. The Design of an Autonomous Vehicle for the Disabled. *IEEE Journal of Robotics and Automation*, RA-2:117–126, 1986.
- [26] C. Mandel, K. Huebner, and T. Vierhuff. Towards an autonomous wheelchair: Cognitive aspects in service robotics. In *Proceedings of Towards Autonomous Robotic Systems*, 2005.
- [27] M. Mazo. An integral system for assisted mobility. *IEEE Robotics & Automation Magazine*, 8:46–56, 2001.
- [28] D. P. Miller and M. G. Slack. Design and testing of a low-cost robotic wheelchair prototype. *Autonomous Robots*, 2:77–88, 1995.
- [29] E. Prassler, J. Scholz, and P. Fiorini. A robotics wheelchair for crowded public environment. *IEEE Robotics & Automation Magazine*, 8:38–45, 2001.
- [30] A. Pruski and G. Bourhis. The VAHM project: a cooperation between an autonomous mobileplatform and a disabled person. In *IEEE International Conference on Robotics and Automation*, 1992.
- [31] B. Rebsamen, E. Burdet, Cuntai Guan, Chee Leong Teo, Qiang Zeng, M. Ang, and C. Laugier. Controlling a wheelchair using a BCI with low information transferrate. pages 1003–1008, June 2007.

- 
- [32] Sven Rönnbäck. *On Methods for Assistive Mobile Robots*. PhD thesis, Luleå University of Technology, 2006.
- [33] H. Seki, S. Kobayashi, Y. Kamiya, M. Hikizu, and H. Nomura. Autonomous/semi-autonomous navigation system of a wheelchair by active ultrasonic beacons. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00.*, 2000.
- [34] T. Shibusawa, Y. Kobayashi, and Y. Kuno. Robotic wheelchair for museum visit. pages 2827–2830, Aug. 2008.
- [35] R. Simpson. Smart wheelchairs: A literature review. *Journal of Rehabilitation Research & Development*, 42(4):423–438, 2005.
- [36] R. Simpson, E. Lopresti, S. Hayashi, I. Nourbakhsh, and D. Miller. The smart wheelchair component system. *Journal Rehabil Res Dev*, 41:429–42, 2004.
- [37] R. Simpson, D. Poirot, and F. Baxter. The Hephaestus Smart Wheelchair system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10:118–122, 2002.
- [38] D. Tefft, P. Guerette, and J. Furumasu. Cognitive predictors of young childrens readiness for powered mobility. *Developmental Medicine & Child Neurology*, 41(10):665–670, 1999.
- [39] Gerolf Vanacker, José del R. Millán, and Eileen Lew. Context-Based Filtering for Assisted Brain-Actuated Wheelchair Driving. *Computational Intelligence and Neuroscience*, 2007:12, 2007.
- [40] H. Yanco. *Integrating Robotic Research: A Survey of Robotic Wheelchair Development*, 1998.

# Appendice A

## Datasheet

Si riportano in seguito i Datasheet dei principali integrati utilizzati sulle schede “master” e “odometry”.



## Quad, 16-Bit DAC with 5 ppm/°C On-Chip Reference in 14-Lead TSSOP

# AD5666

### FEATURES

**Low power quad 16-bit DAC**  
**14-lead TSSOP**  
**On-chip 1.25 V/2.5 V, 5 ppm/°C reference**  
**Power down to 400 nA @ 5 V, 200 nA @ 3 V**  
**2.7 V to 5.5 V power supply**  
**Guaranteed monotonic by design**  
**Power-on reset to zero scale or midscale**  
**3 power-down functions**  
**Hardware LDAC with LDAC override function**  
**CLR function to programmable code**  
**SDO daisy-chaining option**  
**Rail-to-rail operation**

### APPLICATIONS

**Process control**  
**Data acquisition systems**  
**Portable battery-powered instruments**  
**Digital gain and offset adjustment**  
**Programmable voltage and current sources**  
**Programmable attenuators**

### GENERAL DESCRIPTION

The AD5666 is a low power, quad, 16-bit, buffered voltage-output DAC. The part operates from a single 2.7 V to 5.5 V supply and is guaranteed monotonic by design.

The AD5666 has an on-chip reference with an internal gain of 2. The AD5666-1 has a 1.25 V 5 ppm/°C reference, giving a full-scale output of 2.5 V; the AD5666-2 has a 2.5 V 5 ppm/°C reference, giving a full-scale output of 5 V. The on-board reference is off at power-up, allowing the use of an external reference. The internal reference is turned on by writing to the DAC.

The part incorporates a power-on reset circuit that ensures that the DAC output powers up to 0 V (POR pin low) or to midscale (POR pin high) and remains powered up at this level until a valid write takes place. The part contains a power-down feature that reduces the current consumption of the device to 400 nA at 5 V and provides software-selectable output loads while in power-down mode for any or all DAC channels.

### FUNCTIONAL BLOCK DIAGRAM

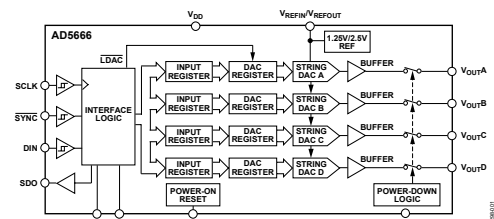


Figure 1.

The outputs of all DACs can be updated simultaneously using the LDAC function, with the added functionality of user-selectable DAC channels to simultaneously update. There is also an asynchronous CLR that clears all DACs to a software-selectable code—0 V, midscale, or full scale.

The AD5666 utilizes a versatile 3-wire serial interface that operates at clock rates of up to 50 MHz and is compatible with standard SPI™, QSPI™, MICROWIRE™, and DSP interface standards. The on-chip precision output amplifier enables rail-to-rail output swing.

### PRODUCT HIGHLIGHTS

1. Quad, 16-bit DAC.
2. On-chip 1.25 V/2.5 V, 5 ppm/°C reference.
3. Available in 14-lead TSSOP.
4. Selectable power-on reset to 0 V or midscale.
5. Power-down capability. When powered down, the DAC typically consumes 200 nA at 3 V and 400 nA at 5 V.

### Rev. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
 Tel: 781.329.4700 [www.analog.com](http://www.analog.com)  
 Fax: 781.461.3113 © 2005 Analog Devices, Inc. All rights reserved.





**dsPIC33FJ32MC302/304,  
dsPIC33FJ64MCX02/X04, AND  
dsPIC33FJ128MCX02/X04**

**High-Performance, 16-bit Digital Signal Controllers**

**Operating Range:**

- Up to 40 MIPS operation (at 3.0-3.6V):
  - Industrial temperature range (-40°C to +85°C)
  - Extended temperature range (-40°C to +125°C)

**High-Performance DSC CPU:**

- Modified Harvard architecture
- C compiler optimized instruction set
- 16-bit wide data path
- 24-bit wide instructions
- Linear program memory addressing up to 4M instruction words
- Linear data memory addressing up to 64 Kbytes
- 83 base instructions: mostly 1 word/1 cycle
- Two 40-bit accumulators with rounding and saturation options
- Flexible and powerful addressing modes:
  - Indirect
  - Modulo
  - Bit-Reversed
- Software stack
- 16 x 16 fractional/integer multiply operations
- 32/16 and 16/16 divide operations
- Single-cycle multiply and accumulate:
  - Accumulator write back for DSP operations
  - Dual data fetch
- Up to ±16-bit shifts for up to 40-bit data

**Direct Memory Access (DMA):**

- 8-channel hardware DMA
- Up to 2 Kbytes dual ported DMA buffer area (DMA RAM) to store data transferred via DMA:
  - Allows data transfer between RAM and a peripheral while CPU is executing code (no cycle stealing)
- Most peripherals support DMA

**Timers/Capture/Compare/PWM:**

- Timer/Counters, up to five 16-bit timers:
  - Can pair up to make two 32-bit timers
  - One timer runs as a Real-Time Clock with an external 32.768 kHz oscillator
  - Programmable prescaler
- Input Capture (up to four channels):
  - Capture on up, down or both edges
  - 16-bit capture input functions
  - 4-deep FIFO on each capture
- Output Compare (up to four channels):
  - Single or Dual 16-bit Compare mode
  - 16-bit Glitchless PWM mode
- Hardware Real-Time Clock/Calendar (RTCC):
  - Provides clock, calendar, and alarm functions

**Interrupt Controller:**

- 5-cycle latency
- 118 interrupt vectors
- Up to 53 available interrupt sources
- Up to three external interrupts
- Seven programmable priority levels
- Five processor exceptions

**Digital I/O:**

- Peripheral pin Select functionality
- Up to 35 programmable digital I/O pins
- Wake-up/Interrupt-on-Change for up to 21 pins
- Output pins can drive from 3.0V to 3.6V
- Up to 5V output with open drain configuration
- All digital input pins are 5V tolerant
- 4 mA sink on all I/O pins

**On-Chip Flash and SRAM:**

- Flash program memory (up to 128 Kbytes)
- Data SRAM (up to 16 Kbytes)
- Boot, Secure, and General Security for program Flash

---

**dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04**


---

**System Management:**

- Flexible clock options:
  - External, crystal, resonator, internal RC
  - Fully integrated Phase-Locked Loop (PLL)
  - Extremely low jitter PLL
- Power-up Timer
- Oscillator Start-up Timer/Stabilizer
- Watchdog Timer with its own RC oscillator
- Fail-Safe Clock Monitor
- Reset by multiple sources

**Power Management:**

- On-chip 2.5V voltage regulator
- Switch between clock sources in real time
- Idle, Sleep, and Doze modes with fast wake-up

**Analog-to-Digital Converters (ADCs):**

- 10-bit, 1.1 Msps or 12-bit, 500 Ksps conversion:
  - Two and four simultaneous samples (10-bit ADC)
  - Up to nine input channels with auto-scanning
  - Conversion start can be manual or synchronized with one of four trigger sources
  - Conversion possible in Sleep mode
  - $\pm 2$  LSB max integral nonlinearity
  - $\pm 1$  LSB max differential nonlinearity

**Audio Digital-to-Analog Converter (DAC):**

- 16-bit Dual Channel DAC module
- 100 Ksps maximum sampling rate
- Second-Order Digital Delta-Sigma Modulator

**Comparator Module:**

- Two analog comparators with programmable input/output configuration

**CMOS Flash Technology:**

- Low-power, high-speed Flash technology
- Fully static design
- 3.3V ( $\pm 10\%$ ) operating voltage
- Industrial and Extended temperature
- Low power consumption

**Motor Control Peripherals:**

- 6-channel 16-bit Motor Control PWM:
  - Three duty cycle generators
  - Independent or Complementary mode
  - Programmable dead time and output polarity
  - Edge-aligned or center-aligned
  - Manual output override control
  - One Fault input
  - Trigger for ADC conversions
  - PWM frequency for 16-bit resolution (@ 40 MIPS) = 1220 Hz for Edge-Aligned mode, 610 Hz for Center-Aligned mode
  - PWM frequency for 11-bit resolution (@ 40 MIPS) = 39.1 kHz for Edge-Aligned mode, 19.55 kHz for Center-Aligned mode
- 2-channel 16-bit Motor Control PWM:
  - One duty cycle generator
  - Independent or Complementary mode
  - Programmable dead time and output polarity
  - Edge-aligned or center-aligned
  - Manual output override control
  - One Fault input
  - Trigger for ADC conversions
  - PWM frequency for 16-bit resolution (@ 40 MIPS) = 1220 Hz for Edge-Aligned mode, 610 Hz for Center-Aligned mode
  - PWM frequency for 11-bit resolution (@ 40 MIPS) = 39.1 kHz for Edge-Aligned mode, 19.55 kHz for Center-Aligned mode
- 2-Quadrature Encoder Interface module:
  - Phase A, Phase B, and index pulse input
  - 16-bit up/down position counter
  - Count direction status
  - Position Measurement (x2 and x4) mode
  - Programmable digital noise filters on inputs
  - Alternate 16-bit Timer/Counter mode
  - Interrupt on position counter rollover/underflow

---

**dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04**


---

**Communication Modules:**

- 4-wire SPI (up to two modules):
  - Framing supports I/O interface to simple codecs
  - Supports 8-bit and 16-bit data
  - Supports all serial clock formats and sampling modes
- I<sup>2</sup>C™:
  - Full Multi-Master Slave mode support
  - 7-bit and 10-bit addressing
  - Bus collision detection and arbitration
  - Integrated signal conditioning
  - Slave address masking
- UART (up to two modules):
  - Interrupt on address bit detect
  - Interrupt on UART error
  - Wake-up on Start bit from Sleep mode
  - 4-character TX and RX FIFO buffers
  - LIN bus support
  - IrDA® encoding and decoding in hardware
  - High-Speed Baud mode
  - Hardware Flow Control with CTS and RTS
- Enhanced CAN (ECAN™ module) 2.0B active:
  - Up to eight transmit and up to 32 receive buffers
  - 16 receive filters and three masks
  - Loopback, Listen Only and Listen All
  - Messages modes for diagnostics and bus monitoring
  - Wake-up on CAN message
  - Automatic processing of Remote Transmission Requests
  - FIFO mode using DMA
  - DeviceNet™ addressing support
- Parallel Master Slave Port (PMP/EPSP):
  - Supports 8-bit or 16-bit data
  - Supports 16 address lines
- Programmable Cyclic Redundancy Check (CRC):
  - Programmable bit length for the CRC generator polynomial (up to 16-bit length)
  - 8-deep, 16-bit or 16-deep, 8-bit FIFO for data input

**Packaging:**

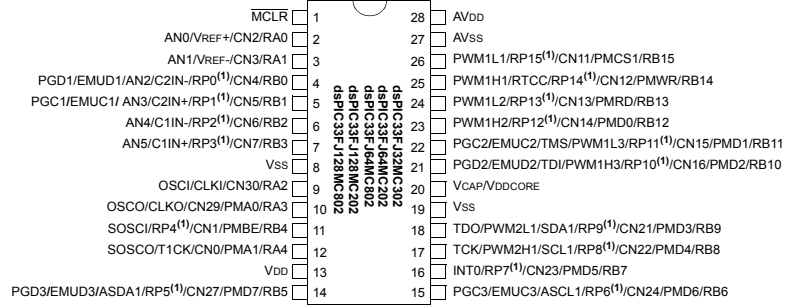
- 28-pin SDIP/SOIC/QFN-S
- 44-pin TQFP/QFN

<p><b>Note:</b> See the device variant table for exact peripheral features per device.</p>
--

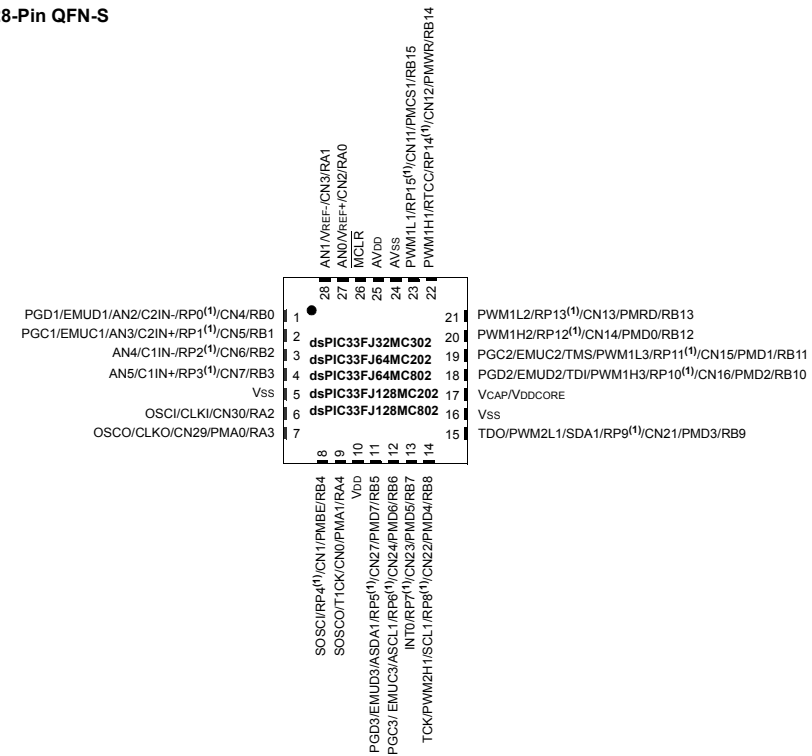
**dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, AND dsPIC33FJ128MCX02/X04**

**Pin Diagrams**

**28-Pin SDIP, SOIC**



**28-Pin QFN-S**



**Note 1:** The RPx pins can be used by any remappable peripheral. See the table “dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04, and dsPIC33FJ128MCX02/X04 Controller Families” in this section for the list of available peripherals.



# dsPIC30F4011/4012

## dsPIC30F4011/4012 Enhanced Flash 16-Bit Digital Signal Controller

**Note:** This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "dsPIC30F Family Reference Manual" (DS70046). For more information on the device instruction set and programming, refer to the "dsPIC30F/33F Programmer's Reference Manual" (DS70157).

### High-Performance, Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture with flexible addressing modes
- 83 base instructions
- 24-bit wide instructions, 16-bit wide data path
- 48 Kbytes on-chip Flash program space (16K instruction words)
- 2 Kbytes of on-chip data RAM
- 1 Kbyte of nonvolatile data EEPROM
- Up to 30 MIPS operation:
  - DC to 40 MHz external clock input
  - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 30 interrupt sources:
  - 3 external interrupt sources
  - 8 user-selectable priority levels for each interrupt source
  - 4 processor trap sources
- 16 x 16-bit working register array

### DSP Engine Features:

- Dual data fetch
- Accumulator write-back for DSP operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single-cycle hardware fractional/integer multiplier
- All DSP instructions are single cycle
- $\pm 16$ -bit, single-cycle shift

### Peripheral Features:

- High-current sink/source I/O pins: 25 mA/25 mA
- Timer module with programmable prescaler:
  - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture input functions
- 16-bit Compare/PWM output functions
- 3-wire SPI modules (supports 4 Frame modes)
- I<sup>2</sup>C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- 2 UART modules with FIFO Buffers
- 1 CAN module, 2.0B compliant

### Motor Control PWM Module Features:

- 6 PWM output channels:
  - Complementary or Independent Output modes
  - Edge and Center-Aligned modes
- 3 duty cycle generators
- Dedicated time base
- Programmable output polarity
- Dead-time control for Complementary mode
- Manual output control
- Trigger for A/D conversions

### Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-Bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

## dsPIC30F4011/4012

### Analog Features:

- 10-Bit Analog-to-Digital Converter (A/D) with 4 S/H inputs:
  - 1 Msps conversion rate
  - 9 input channels
  - Conversion available during Sleep and Idle
- Programmable Brown-out Reset

### Special Digital Signal Controller Features:

- Enhanced Flash program memory:
  - 10,000 erase/write cycle (min.) for industrial temperature range, 100K (typical)
- Data EEPROM memory:
  - 100,000 erase/write cycle (min.) for industrial temperature range, 1M (typical)
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)

### Special Digital Signal Controller Features (Cont.):

- Flexible Watchdog Timer (WDT) with on-chip, low-power RC oscillator for reliable operation
- Fail-Safe Clock Monitor operation detects clock failure and switches to on-chip, low-power RC oscillator
- Programmable code protection
- In-Circuit Serial Programming™ (ICSP™)
- Selectable Power Management modes:
  - Sleep, Idle and Alternate Clock modes

### CMOS Technology:

- Low-power, high-speed Flash technology
- Wide operating voltage range (2.5V to 5.5V)
- Industrial and Extended temperature ranges
- Low-power consumption

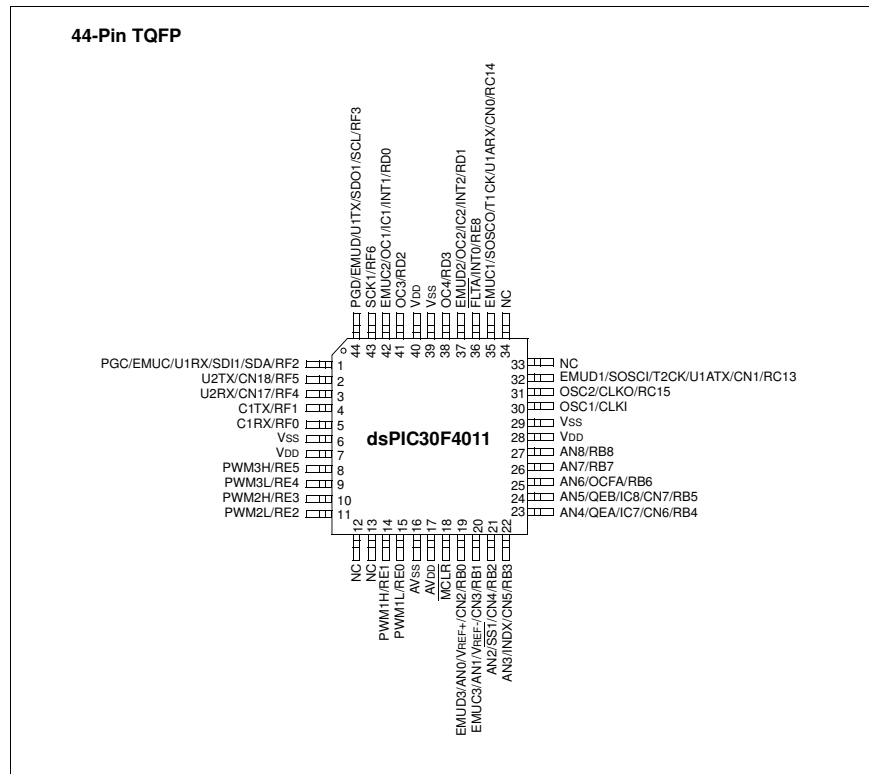
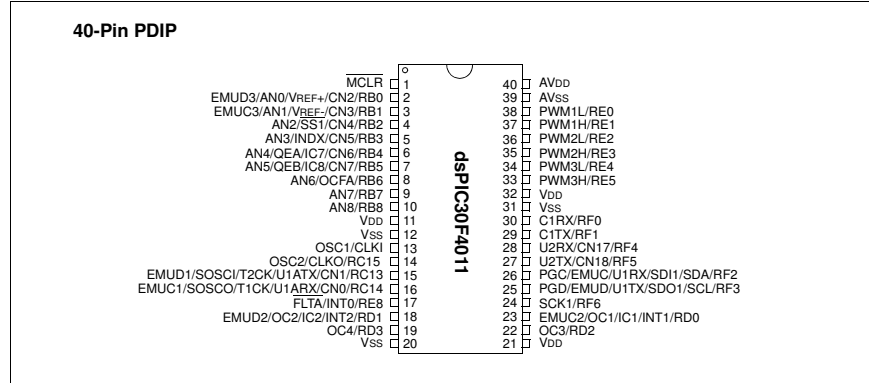
### dsPIC30F Motor Control and Power Conversion Family\*

Device	Pins	Program Mem. Bytes/Instructions	SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	Motor Control PWM	10-Bit A/D 1 Msps	Quad Enc	UART	SPI	I <sup>2</sup> C™	CAN
dsPIC30F2010	28	12K/4K	512	1024	3	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F3010	28	24K/8K	1024	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	-
dsPIC30F4012	28	48K/16K	2048	1024	5	4	2	6 ch	6 ch	Yes	1	1	1	1
dsPIC30F3011	40/44	24K/8K	1024	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	-
dsPIC30F4011	40/44	48K/16K	2048	1024	5	4	4	6 ch	9 ch	Yes	2	1	1	1
dsPIC30F5015	64	66K/22K	2048	1024	5	4	4	8 ch	16 ch	Yes	1	2	1	1
dsPIC30F6010	80	144K/48K	8192	4096	5	8	8	8 ch	16 ch	Yes	2	2	1	2

\* This table provides a summary of the dsPIC30F6010 peripheral features. Other available devices in the dsPIC30F Motor Control and Power Conversion Family are shown for feature comparison.

# dsPIC30F4011/4012

## Pin Diagrams





## Dual-Channel Digital Isolators

### ADuM1200/ADuM1201

#### FEATURES

**Narrow body, RoHS-compliant, SOIC 8-lead package**

**Low power operation**

**5 V operation**

1.1 mA per channel maximum @ 0 Mbps to 2 Mbps

3.7 mA per channel maximum @ 10 Mbps

8.2 mA per channel maximum @ 25 Mbps

**3 V operation**

0.8 mA per channel maximum @ 0 Mbps to 2 Mbps

2.2 mA per channel maximum @ 10 Mbps

4.8 mA per channel maximum @ 25 Mbps

**Bidirectional communication**

**3 V/5 V level translation**

**High temperature operation: 125°C**

**High data rate: dc to 25 Mbps (NRZ)**

**Precise timing characteristics**

3 ns maximum pulse width distortion

3 ns maximum channel-to-channel matching

**High common-mode transient immunity: >25 kV/μs**

**Automotive versions qualified per AEC-Q100**

**Safety and regulatory approvals**

**UL recognition**

2500 V rms for 1 minute per UL 1577

**CSA Component Acceptance Notice #5A**

**VDE Certificate of Conformity**

DIN V VDE V 0884-10 (VDE V 0884-10): 2006-12

V<sub>ORM</sub> = 560 V peak

#### APPLICATIONS

**Size-critical multichannel isolation**

**SPI interface/data converter isolation**

**RS-232/RS-422/RS-485 transceiver isolation**

**Digital field bus isolation**

**Hybrid electric vehicles, battery monitor, and motor drive**

#### GENERAL DESCRIPTION

The ADuM120x<sup>1</sup> are dual-channel, digital isolators based on the Analog Devices, Inc., *iCoupler*<sup>®</sup> technology. Combining high speed CMOS and monolithic transformer technologies, these isolation components provide outstanding performance characteristics superior to alternatives, such as optocouplers.

By avoiding the use of LEDs and photodiodes, *iCoupler* devices remove the design difficulties commonly associated

with optocouplers. The typical optocoupler concerns regarding uncertain current transfer ratios, nonlinear transfer functions, and temperature and lifetime effects are eliminated with the simple *iCoupler* digital interfaces and stable performance characteristics. The need for external drivers and other discrete components is eliminated with these *iCoupler* products. Furthermore, *iCoupler* devices consume one-tenth to one-sixth the power of optocouplers at comparable signal data rates.

The ADuM120x isolators provide two independent isolation channels in a variety of channel configurations and data rates (see the Ordering Guide). Both parts operate with the supply voltage on either side ranging from 2.7 V to 5.5 V, providing compatibility with lower voltage systems as well as enabling a voltage translation functionality across the isolation barrier. In addition, the ADuM120x provide low pulse width distortion (<3 ns for CR grade) and tight channel-to-channel matching (<3 ns for CR grade). Unlike other optocoupler alternatives, the ADuM120x isolators have a patented refresh feature that ensures dc correctness in the absence of input logic transitions and during power-up/power-down conditions.

ADuM1200W and ADuM1201W are automotive grade versions qualified for 125°C operation per AEC-Q100. See the Automotive Products section for more details.

#### FUNCTIONAL BLOCK DIAGRAMS

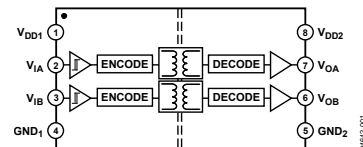


Figure 1. ADuM1200 Functional Block Diagram

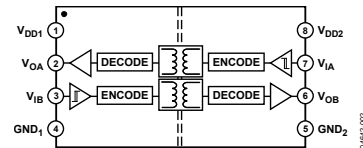


Figure 2. ADuM1201 Functional Block Diagram

<sup>1</sup> Protected by U.S. Patents 5,952,849; 6,873,065; 6,903,578; and 7,075,329. Other patents are pending.

#### Rev. H

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781.329.4700 [www.analog.com](http://www.analog.com)  
Fax: 781.461.3113 ©2004–2009 Analog Devices, Inc. All rights reserved.





## Quad-Channel Digital Isolators

### ADuM1400/ADuM1401/ADuM1402

#### FEATURES

Automotive versions qualified per AEC-Q100

#### Low power operation

##### 5 V operation

1.0 mA per channel maximum @ 0 Mbps to 2 Mbps

3.5 mA per channel maximum @ 10 Mbps

31 mA per channel maximum @ 90 Mbps

##### 3 V operation

0.7 mA per channel maximum @ 0 Mbps to 2 Mbps

2.1 mA per channel maximum @ 10 Mbps

20 mA per channel maximum @ 90 Mbps

#### Bidirectional communication

#### 3 V/5 V level translation

High temperature operation: 125°C

High data rate: dc to 90 Mbps (NRZ)

#### Precise timing characteristics

2 ns maximum pulse width distortion

2 ns maximum channel-to-channel matching

High common-mode transient immunity: >25 kV/μs

#### Output enable function

#### 16-lead SOIC wide body package

#### RoHS-compliant models available

#### Safety and regulatory approvals

UL recognition: 2500 V rms for 1 minute per UL 1577

CSA Component Acceptance Notice #5A

VDE Certificate of Conformity

DIN V VDE V 0884-10 (VDE V 0884-10):2006-12

V<sub>ORM</sub> = 560 V peak

TÜV approval: IEC/EN/UL/CSA 61010-1

#### APPLICATIONS

General-purpose multichannel isolation

SPI interface/data converter isolation

RS-232/RS-422/RS-485 transceivers

Industrial field bus isolation

Automotive systems

#### GENERAL DESCRIPTION

The ADuM140x<sup>1</sup> are quad-channel digital isolators based on Analog Devices, Inc., *iCoupler*<sup>®</sup> technology. Combining high speed CMOS and monolithic air core transformer technology, these isolation components provide outstanding performance characteristics superior to alternatives, such as optocoupler devices.

By avoiding the use of LEDs and photodiodes, *iCoupler* devices remove the design difficulties commonly associated with optocouplers. The typical optocoupler concerns regarding uncertain current transfer ratios, nonlinear transfer functions, and temperature and lifetime effects are eliminated with the simple *iCoupler* digital interfaces and stable performance characteristics.

The need for external drivers and other discrete components is eliminated with these *iCoupler* products. Furthermore, *iCoupler* devices consume one-tenth to one-sixth of the power of optocouplers at comparable signal data rates.

The ADuM140x isolators provide four independent isolation channels in a variety of channel configurations and data rates (see the Ordering Guide). All models operate with the supply voltage on either side ranging from 2.7 V to 5.5 V, providing compatibility with lower voltage systems as well as enabling a voltage translation functionality across the isolation barrier. In addition, the ADuM140x provides low pulse width distortion (<2 ns for CRW grade) and tight channel-to-channel matching (<2 ns for CRW grade). Unlike other optocoupler alternatives, the ADuM140x isolators have a patented refresh feature that ensures dc correctness in the absence of input logic transitions and when power is not applied to one of the supplies.

<sup>1</sup> Protected by U.S. Patents 5,952,849; 6,873,065; 6,903,578; and 7,075,329. Other patents are pending.

#### FUNCTIONAL BLOCK DIAGRAMS

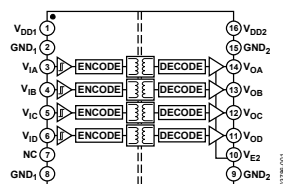


Figure 1. ADuM1400

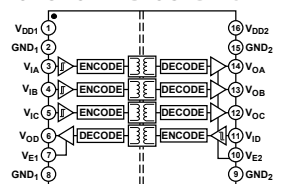


Figure 2. ADuM1401

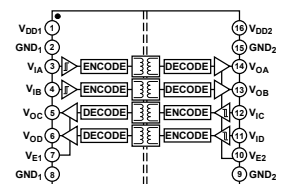


Figure 3. ADuM1402

#### Rev. G

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781.329.4700 [www.analog.com](http://www.analog.com)  
Fax: 781.461.3113 ©2003–2008 Analog Devices, Inc. All rights reserved.

# HOA096X/097X

## Transmissive Optoschmitt Sensor

**FEATURES**

- Direct TTL interface
- Accurate position sensing
- Four mounting configurations
- Buffer or inverting logic available
- Choice of detector aperture
- 0.125 in. (3.18 mm) slot width

**DESCRIPTION**

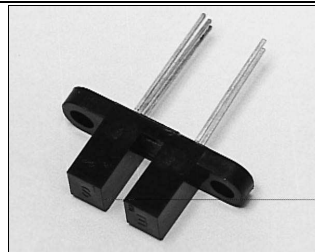
The HOA096X/097X series consists of an infrared emitting diode facing an Optoschmitt detector encased in a black thermoplastic housing. Detector switching takes place whenever an opaque object passes through the slot between emitter and detector. The photodetector consists of a photodiode, amplifier, voltage regulator, Schmitt trigger and an NPN output transistor with 10 kΩ (nominal) pull-up resistor. The user can choose from available options: (1) detector aperture size, (2) mounting tab configuration, and (3) housing material.

The HOA096X series utilizes an IR transmissive polysulfone housing which features smooth optical faces without external aperture openings; this feature is desirable when aperture blockage from airborne contaminants is a possibility. The HOA097X series employs an opaque polysulfone housing with aperture openings for use in applications where maximum rejection of ambient light is important and in situations in which maximum position resolution is desired. The HOA096X/097X series employs plastic molded components. For additional component information see SEP8506 and SDP8600.

Housing material is polysulfone. Housings are soluble in chlorinated hydrocarbons and ketones. Recommended cleaning agents are methanol and isopropanol.

Device Polarity:  
 Buffer - Output is LO when optical path is blocked.  
 Inverter - Output is HI when optical path is blocked.

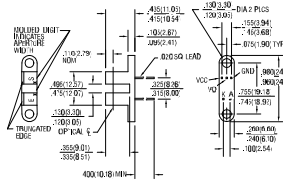
To specify the complete product characteristics, see PART NUMBER GUIDE.



INRA-261 F

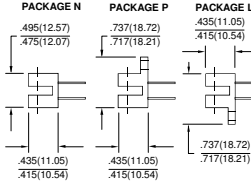
**OUTLINE DIMENSIONS** in inches (mm)  
 Tolerance 3 plc decimals ±0.010(0.25)  
 2 plc decimals ±0.020(0.51)

**Package T**



DIM\_066a.cdr

**Packages N/P/L**



DIM\_41b.dwg



# MCP1702

## 250 mA Low Quiescent Current LDO Regulator

### Features

- 2.0  $\mu$ A Quiescent Current (typical)
- Input Operating Voltage Range: 2.7V to 13.2V
- 250 mA Output Current for Output Voltages  $\geq$  2.5V
- 200 mA Output Current for Output Voltages  $<$  2.5V
- Low Dropout (LDO) voltage
  - 625 mV typical @ 250 mA ( $V_{OUT} = 2.8V$ )
- 0.4% Typical Output Voltage Tolerance
- Standard Output Voltage Options:
  - 1.2V, 1.5V, 1.8V, 2.5V, 2.8V, 3.0V, 3.3V, 4.0V, 5.0V
- Output voltage range 1.2V to 5.5V in 0.1V Increments (50 mV increments available upon request)
- Stable with 1.0  $\mu$ F to 22  $\mu$ F Output Capacitor
- Short-Circuit Protection
- Overtemperature Protection

### Applications

- Battery-powered Devices
- Battery-powered Alarm Circuits
- Smoke Detectors
- CO<sup>2</sup> Detectors
- Pagers and Cellular Phones
- Smart Battery Packs
- Low Quiescent Current Voltage Reference
- PDAs
- Digital Cameras
- Microcontroller Power
- Solar-Powered Instruments
- Consumer Products
- Battery Powered Data Loggers

### Related Literature

- AN765, "Using Microchip's Micropower LDOs", DS00765, Microchip Technology Inc., 2002
- AN766, "Pin-Compatible CMOS Upgrades to BiPolar LDOs", DS00766, Microchip Technology Inc., 2002
- AN792, "A Method to Determine How Much Power a SOT-23 Can Dissipate in an Application", DS00792, Microchip Technology Inc., 2001

### Description

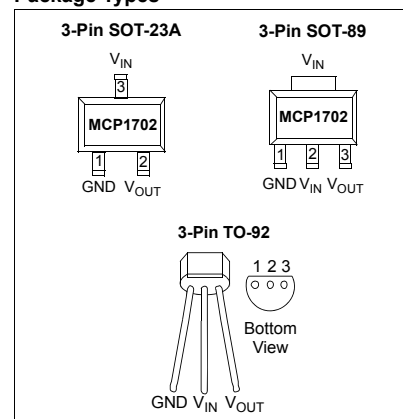
The MCP1702 is a family of CMOS low dropout (LDO) voltage regulators that can deliver up to 250 mA of current while consuming only 2.0  $\mu$ A of quiescent current (typical). The input operating range is specified from 2.7V to 13.2V, making it an ideal choice for two to six primary cell battery-powered applications, 9V alkaline and one or two cell Li-Ion-powered applications.

The MCP1702 is capable of delivering 250 mA with only 625 mV (typical) of input to output voltage differential ( $V_{OUT} = 2.8V$ ). The output voltage tolerance of the MCP1702 is typically  $\pm 0.4\%$  at  $+25^{\circ}\text{C}$  and  $\pm 3\%$  maximum over the operating junction temperature range of  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$ . Line regulation is  $\pm 0.1\%$  typical at  $+25^{\circ}\text{C}$ .

Output voltages available for the MCP1702 range from 1.2V to 5.0V. The LDO output is stable when using only 1  $\mu$ F of output capacitance. Ceramic, tantalum or aluminum electrolytic capacitors can all be used for input and output. Overcurrent limit and overtemperature shutdown provide a robust solution for any application.

Package options include the SOT-23A, SOT-89-3, and TO-92.

### Package Types

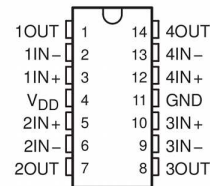


## TLC274, TLC274A, TLC274B, TLC274Y, TLC279 LinCMOS™ PRECISION QUAD OPERATIONAL AMPLIFIERS

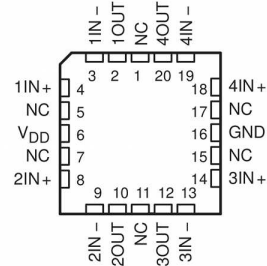
SLOS092D – SEPTEMBER 1987 – REVISED MARCH 2001

- **Trimmed Offset Voltage:**  
TLC279 . . . 900  $\mu\text{V}$  Max at 25°C,  
 $V_{\text{DD}} = 5\text{ V}$
- **Input Offset Voltage Drift . . . Typically**  
0.1  $\mu\text{V}/\text{Month}$ , Including the First 30 Days
- **Wide Range of Supply Voltages Over Specified Temperature Range:**  
0°C to 70°C . . . 3 V to 16 V  
–40°C to 85°C . . . 4 V to 16 V  
–55°C to 125°C . . . 4 V to 16 V
- **Single-Supply Operation**
- **Common-Mode Input Voltage Range Extends Below the Negative Rail (C-Suffix and I-Suffix Versions)**
- **Low Noise . . . Typically 25 nV/ $\sqrt{\text{Hz}}$  at  $f = 1\text{ kHz}$**
- **Output Voltage Range Includes Negative Rail**
- **High Input Impedance . . .  $10^{12}\ \Omega$  Typ**
- **ESD-Protection Circuitry**
- **Small-Outline Package Option Also Available in Tape and Reel**
- **Designed-In Latch-Up Immunity**

D, J, N, OR PW PACKAGE  
(TOP VIEW)



FK PACKAGE  
(TOP VIEW)



NC – No internal connection

### description

The TLC274 and TLC279 quad operational amplifiers combine a wide range of input offset voltage grades with low offset voltage drift, high input impedance, low noise, and speeds approaching that of general-purpose BiFET devices.

These devices use Texas Instruments silicon-gate LinCMOS™ technology, which provides offset voltage stability far exceeding the stability available with conventional metal-gate processes.

The extremely high input impedance, low bias currents, and high slew rates make these cost-effective devices ideal for applications which have previously been reserved for BiFET and NFET products. Four offset voltage grades are available (C-suffix and I-suffix types), ranging from the low-cost TLC274 (10 mV) to the high-precision TLC279 (900  $\mu\text{V}$ ). These advantages, in combination with good common-mode rejection and supply voltage rejection, make these devices a good choice for new state-of-the-art designs as well as for upgrading existing designs.

LinCMOS is a trademark of Texas Instruments.

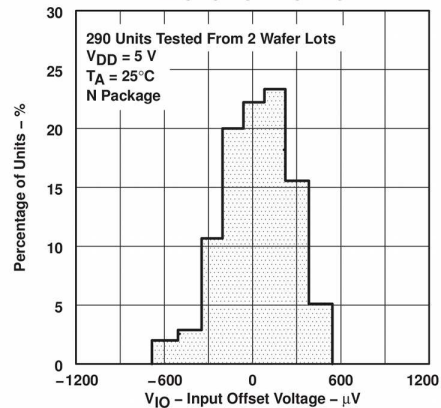
PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2001, Texas Instruments Incorporated

TEXAS  
INSTRUMENTS

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

DISTRIBUTION OF TLC279  
INPUT OFFSET VOLTAGE





Document Reference No.: FT\_000055  
 UB232R USB Mini-B FT232R Evaluation Module Datasheet Version 1.00  
 Clearance No.: FTDI# 43

## 1 Introduction

The UB232R is the smallest USB - serial development module in the FTDI product range. To minimise the size of the module, the UB232R uses a standard USB mini-B connector. It is ideal for new development purposes as well as a platform for adding a USB interface to existing product designs.

The UB232R is based on the FTDI FT232RQ USB to Serial UART IC which handles all the USB signalling and protocols. The FT232R datasheet, **DS\_FT232R**, is available at <http://www.ftdichip.com>,

The UB232R supports RTS/CTS hardware handshaking and is USB powered. The module supports data transfer rates from 300 baud to 3 Mbaud (RS422, RS485, RS232 and at TTL levels). In addition, two of the configurable CBUS pins of the FT232R are made available on the UB232R connector, allowing the UB232R to provide clock signals to external logic or they can be used to drive signal traffic indicator LEDs.

The UB232R is supplied with two 1x4 turned pin board headers with a standard 0.1" pitch. This aids rapid prototyping and development

The UB232R requires USB drivers, available free from <http://www.ftdichip.com>, which are used to make the FT232R on the UB232R appear as a virtual COM port (VCP). This then allows the user to communicate with the USB interface via a standard PC serial emulation port (TTY). Another FTDI USB driver, the D2XX driver, can also be used with application software to directly access the FT232R on the UB232R through a DLL. This is illustrated in the Figure 1.1

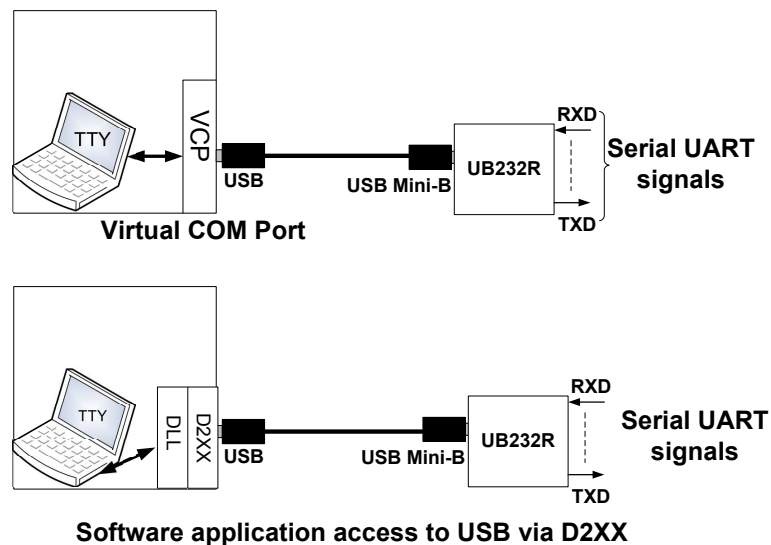


Figure 1.1 Using the UB232R

Drivers for the UB232R are available free from <http://www.ftdichip.com>.