SENSORIZZAZIONE DI UN VEICOLO PER GUIDA AUTONOMA

Candidato: Salvatore Campolo

Relatore: Mauro Velardocchia Co-relatore: Antonio Tota



Politecnico di Torino

Dipartimento di Ingegneria Meccanica ed Aerospaziale Laurea Magistrale in Ingegneria Meccanica A.A. 2018/2019

Ringraziamenti

Vorrei ringraziare innanzitutto la mia famiglia, che mi ha sempre supportato nella mia scelta di studiare lontano da casa, che mi ha sostenuto nei momenti difficili e ha gioito con me nei momenti più belli.

Un ringraziamento a tutti gli amici che ho incontrato durante questi due anni, che hanno reso più leggera questa esperienza importante ed impegnativa che è la laurea magistrale e hanno saputo comprendere la mia determinazione e il mio impegno per raggiungere questo traguardo.

Vorrei ringraziare il professor Mauro Velardocchia per avermi dato la possibilità di esplorare il mondo della guida autonoma, il quale presto diverrà la normalità anche se ad oggi sembra distante, e di avermi permesso di lavorare a ritmi sostenuti per raggiungere i miei obiettivi.

Un ringraziamento speciale va ad Antonio Tota, che mi ha seguito in ogni aspetto di questa ricerca e aiutato ogni qual volta ne avessi bisogno sempre fornendo più del massimo della disponibilità ed interessandosi al mio lavoro.

Un ringraziamento va anche ai proff. Paolo Dabove e Marco Piras del DIATI, per la loro disponibilità a fornire strumentazione e chiarimenti.

Indice

1	Intr	oduzione	12
	1.1	Background sulla sensoristica più utilizzata	12
	1.2	L'obiettivo della tesi	13
n	G4:-	none l'angele di beeding	14
4	50H	Ctime maliente fluere mentice	14
	2.1	Stima mediante nusso magnetico	14
	2.2	Problematiche legate all'uso di magnetometri	15
3	\mathbf{Sen}	sori e schede	17
	3.1	Unità inerziale SBG Ekinox D	17
		3.1.1 Collegare la Ekinox	19
		3.1.2 La porta CAN	20
	32	Arduino e sensori	$\frac{-}{22}$
	0	3.2.1 Scheda Arduino	${22}$
		3.2.2 Accelerometro e giroscopio MPU 6050	23
		3.2.3 Magnetometro HMC5883L	$\frac{20}{24}$
		3.2.4 Breakout board CV 87	24 25
		2.2.5 Modulo Mouson DAC4021	20
	• •	S.2.5 Modulo Mouser DAC4921	20 97
	J.J	2.2.1 Congono Elvinov	21
		3.3.1 Sensore Ekiliox	21
	9.4	3.3.2 Sensori MP 00050+HMC5883L	21
	3.4		28
		3.4.1 Idea di base	28
		3.4.2 Sensori esterni di Arduino	28
4	\mathbf{Filt}	ro di Kalman per il calcolo dell'heading angle	33
	4.1	Filtro di Kalman	33
	4.2	Scelta del modello	35
		4.2.1 Precisazioni sul significato degli angoli	35
	4.3	Altri lavori	36
		4.3.1 Basi teoriche	36
		4.3.2 Lavori di riferimento	36
	4.4	Considerazioni ed ipotesi iniziali	36
	4.5	Angolo di heading e <i>auro bias</i>	37
		4.5.1 I parametri del modello	38
	4.6	Modifiche al modello	38
	_		
5	I ru	mori Q ed R	39
	5.1	Analisi di Allan	39
	5.2	Misura della varianza di Allan del giroscopio	41
	5.3	Simulazioni Matlab	42
		5.3.1 Riempimento delle matrici	43

		5.3.2	Esecuzione delle simulazioni	43
6	Imp	lemen	tazione del filtro su scheda elettronica Arduino	49
	6.1	Impost	tazione del progetto Arduino	49
	6.2	Collega	amenti da effettuare	50
	6.3	Acquis	sire i dati	51
	6.4	Valuta	zione delle prestazioni del filtro di Kalman	52
		6.4.1	Modello con misura $z = \psi_{mag} \dots \dots \dots \dots \dots \dots \dots \dots$	52
		6.4.2	Modello con misura $z = [\psi_{mag} \omega_z]$	54
	6.5	Conclu	isioni	56
7	Moo	difiche	al filtro	57
	7.1	Ipotesi	adottata per la gestione dei disturbi	57
	7.2	Compe	ensazione degli effetti dinamici	57
		7.2.1	Accelerazioni agenti sul sistema	57
		722	Metodo per escludere le accelerazioni dovute alla dinamica	58
		723	Verifica e calibrazione dei parametri	50
	73	T.2.5 Influor	verifica e canorazione dei parametri	64
	1.0	791	Implementarione di un metodo threshold based	65
	74	(.3.1 Comm	Implementazione di un metodo <i>unesnota-basea</i>	66
	1.4	Compe	Verifies dell'electric dovuti a disturbi magnetici	00 67
		(.4.1 C 1		07
	7.5	Conclu	Isioni e precisazioni	69
8	Des	crizion	e del controllore	71
	8.1	I comp	oonenti utilizzati	71
		8.1.1	NI PXI 1031	71
		8.1.2	NI PXI 8110	72
		8.1.3	NI PXI 6123	72
		8.1.4	NI PXI 8513	73
9	La r	ete C	AN	74
0	91	Archit	ettura	75
	0.1	Datab	ase CAN	75
	5.2	Datab		10
10	I so	ftware	National Instruments	79
	10.1	NI Me	asurement and Automation eXplorer	79
		10.1.1	Aggiungere una scheda DAQ	80
		10.1.2	Aggiungere una scheda CAN	81
	10.2	NI Ver	iStand	82
		10.2.1	Introduzione	82
		10.2.2	L'ambiente VeriStand e definizione di un progetto	82
		10.2.3	Il Workspace	94
11	Cre	azione	del progetto di acquisizione dati	96
	11.1	Definiz	zione delle schede da utilizzare	96
	11.2	Interfa	ccia tra Arduino e PXI	96
	11.4	11 9 1	I pin analogici di Arduino	97
		11.2.1 11.9.9	Soluzioni nossibili	07
	11 9	Collog	are Arduine al PXI	91
	11.0	11 9 1	Modifishe al programme Arduire	90 100
		11.3.1		100

12 Il Workspace in VeriStand	101
12.1 Ulteriori informazioni	105
12.2 Un primo confronto	105
13 Prove sperimentali per la valutazione dell'algoritmo	107
13.1 Setup sperimentale	107
13.2 Luoghi delle prove	108
13.2.1 Variazione dei parametri del filtro	109
13.3 Prove interne	110
13.3.1 Prova su percorso interno	110
13.3.2 Valutazione del disturbo	114
13.3.3 Disturbo derivante dalla rotazione del motore	114
13.3.4 Disturbo derivante dal sistema di fissaggio	117
13.4 Valutazione del filtraggio del segnale delle accelerazioni	119
13.4.1 Implementazione in Simulink del filtro	119
13.4.2 Considerazioni aggiuntive	120
13.4.3 Risultati del filtraggio	121
13.5 Prove esterne	123
13.5.1 Prima prova	123
13.5.2 Seconda prova \ldots	128
13.6 Conclusioni sulle prove sperimentali	133
14 Valutazioni per un'applicazione speciale	135
14.1 Il mezzo	135
14.2 Specifiche tecniche	136
15 Prove sperimentali per veicolo speciale	137
15.1 Analisi in frequenza tramite Matlab	138
15.2 Valutazione dell'ambiente di prova	139
15.3 Valutazione della rumorosità del mezzo	140
15.4 Valutazione delle vibrazioni del mezzo	144
15.5 Valutazione del transitorio spento-acceso-spento	146
15.6 Valutazione del posizionamento esterno del sensore	148
15.7 Valutazione del rumore esterno a veicolo acceso	149
15.8 Comportamento del veicolo in moto	152
15.9 Conclusioni sulle prove sperimentali	157
15.10Sviluppi futuri	158

16 Conclusioni

159

Elenco delle figure

2.1	Definizione dell'angolo di heading.	15
3.1	L'unità Ekinox D.	17
3.2	Pannello status globale.	18
3.3	Pannello status globale.	19
3.4	Configurazione inziale Ethernet.	20
3.5	Impostazione IP fisso PC.	20
3.6	Lo Split Box.	21
3.7	Le schede Arduino.	22
3.8	Arduino IDE	23
3.9	MPU 6050	24
3.10	Immagine al microscopio elettronico dell'interno di un accelerometro, da [25].	24
3.11	GY-87	25
3.12	DAC 4921.	26
3.13	Calibrazione dell'MPU 6050.	29
3.14	Effetti dovuti alle diverse sorgenti di disturbo. Fonte: SBG Systems	29
3.15	Figura che mostra il luogo delle misurazioni prima (a destra) e dopo (a	
	sinistra) la calibrazione. Da [24].	30
3.16	http://www.magnetic-declination.com/	31
3.17	MagViewer e i dati correttamente calibrati.	31
3.18	http://www.magnetic-declination.com/	32
5.1	Espressione del procedimento per il calcolo della varianza, da [27]	40
5.2	Tratto del diagramma di Allan che identifica l'ARW, da [27]	41
5.3	Tratto del diagramma di Allan che identifica l'RRW, da [27]	41
5.4	Diagramma di Allan del giroscopio	42
5.5	Simulazione di uno spostamento del sensore	44
5.6	Simulazione del filtro su asse x	45
5.7	Simulazione del filtro su asse y	45
5.8	Simulazione del filtro su asse z	46
5.9	Simulazione del filtro su asse x	46
5.10	Simulazione del filtro su asse y	47
5.11	Simulazione del filtro su asse z	47
6.1	Schermata principale del programma Arduino.	50
6.2	Le schede Arduino.	51
6.3	Prova 1	52
6.4	Prova 2	53
6.5	Prova 3	53
6.6	Prova 1	54
6.7	Prova 2	54
6.8	Prova 3	55
0.0		00

6.9	Visualizzazione del fenomeno di attenuazione	55
$\begin{array}{c} 7.1 \\ 7.2 \\ 7.3 \\ 7.4 \\ 7.5 \\ 7.6 \\ 7.7 \\ 7.8 \\ 7.9 \\ 7.10 \\ 7.11 \\ 7.12 \\ 7.13 \\ 7.14 \\ 7.15 \end{array}$	Roll non compensato.	60 60 61 62 62 63 63 64 65 66 68 69 69
8.1 8.2 8.3 8.4	tarato.	70 72 72 73 73
9.1 9.2 9.3 9.4 9.6	Resa grafica architettura CAN, da [9]	74 75 76 77 78
$10.1 \\ 10.2 \\ 10.3 \\ 10.4 \\ 10.5 \\ 10.6 \\ 10.7 \\ 10.8 \\ 10.9 \\ 10.10$	Schermata principale di NI MAX.	 79 80 81 81 82 83 84 84 85
10.12 10.12 10.12 10.12 10.14 10.14 10.14 10.14	1Definizione scheda di acquisizione	86 86 86 87 87 87 88 88 89
10.18 10.19 10.20 10.21	8Esempio di Calculated channel.	89 90 91 91

10.22Schermata di compilazione.10.23Messaggio di avvenuta compilazione.10.24Inserimento di un modello nel progetto.10.25Configurazione delle grandezze del modello.10.26Esempio di Workspace.10.27Impostazione di un grafico semplice.10.28Impostazione Logging Control.	. 92 . 92 . 93 . 93 . 94 . 94 . 95
11.1 Rappresentazione del PWM in Arduino.11.2 Pin del DAC4921.11.3 Connessione del chip MCP4921.11.4 Circuito per la connessione di Arduino alla NI DAQ.	. 97 . 99 . 99 . 99 . 100
12.1 Workspace: finestra IMU.12.2 Workspace: finestra Dual GPS.12.3 Workspace: finestra per filtro passa basso.12.4 Prova di confronto.	. 102 . 103 . 104 . 106
 13.1 Il setup sperimentale. 13.2 Disposizione dei sensori sul veicolo. 13.3 Percorso interno, Google Maps. 13.5 Output accelerometri, prova interna. 13.6 Output giroscopi, prova interna. 13.7 Output magnetometri, prova interna. La media di Ekinox è calcolata sul- 	. 107 . 108 . 109 . 110 . 111
l'intera prova, mentre la media di Arduino è calcolata secondo l'algoritmo mostrato nel paragrafo 7.4	. 111 . 112 . 113
13.10Differenza tra neading stimati dai due sensori. 13.11Evidenza tra neading stimati dai due sensori. 13.11Evidenza del fenomeno di amplificazione in moto. 13.12Differenza nelle accelerazioni rilevate da Ekinox. 13.13Differenza nelle accelerazioni rilevate da Arduino. 13.13Differenza nelle accelerazioni rilevate da Arduino.	. 113 . 114 . 115 . 115
 13.14A sinistra: FFT eseguita sui segnali a veicolo spinto (max scala -30 dB circa). A destra: FFT eseguita sui segnali a veicolo spinto dal motore (max scala -20 dB circa). 13.15Diagrammi della prova di confronto dentro/fuori. 	. 116 . 117
 13.16A sinistra: FFT eseguita sui segnali a sensore esterno. A destra: FFT eseguita sui segnali a sensore interno	. 118 . 120 . 120
 13.19Output del filtro per diverse frequenze	. 121 e.121 . 122
13.22Segnale originario soggetto a filtraggio. 13.23Andamento nel tempo della prova. 13.23Andamento nel tempo delle velocità angolari. 13.24Andamento nel tempo delle velocità angolari. 13.25Confronto delle stime degli angoli di orientamento	. 122 . 123 . 124 . 124
13.26Confronto degli angoli di heading stimati. 13.27Scostamento degli angoli di heading stimati. 13.28Valori di campo magnetico e variazioni rispetto al valore medio.	. 124 . 125 . 125 . 126

13.29Spettrogrammi eseguiti nel tratto centrale della prova, tra 40 e 140 secondi,	
utilizzando finestre temporali di 1 secondo con 95% di overlap. \ldots .	. 126
13.30Spettrogramma accelerazione z , prova 1	. 127
13.31Confronto tra segnali grezzi e filtrati di Ekinox	. 127
13.32Andamento nel tempo della prova	. 128
13.33Andamento nel tempo delle velocità angolari	. 129
13.34Confronto delle stime degli angoli di orientamento, con il nuovo valore di	
R(1,1) per il pitch angle misurato	. 129
13.35Confronto degli angoli di heading stimati.	. 130
13.36Scostamento degli angoli di heading stimati	. 130
13.37 Valori del fattore di adattamento di R	. 130
13.38Andamento delle misure magnetiche e scostamenti. Le logiche per il calcolo	1
del valore medio di riferimento sono quelle illustrate per la prova preceden	te.131
13.39Fenomeno di drift in Ekinox.	. 131
13.41Spettrogramma accelerazione z, prova 2. Spettrogrammi eseguiti tra 90 e	:
160 secondi della prova, con finestre di 1 secondo e overlap del 95%	. 132
13.42Confronto tra accelerazioni filtrate e grezze di Ekinox, seconda prova esterr	ıa.133
13.43 Riduzione del rumore nella misura dell'angolo θ .	. 133
14.1 Il BRT-AATV	. 136
15.1 Danticolano delle latture degli appelanemetri. Anomale le latture dell'agge e	
15.1 Particolare delle letture degli accelerometri. Anomala la lettura dell'asse y	190
	. 139
15.2 Distribuzione dene letture dei giroscopi.	. 140
15.3 Variazioni delle misure magneticne	. 140
15.4 Confronto esterno-interno	. 141
15.5 Particolare delle letture degli accelerometri.	. 142
15.6 Distribuzione delle letture dei giroscopi.	. 142
15.7 Variazioni delle misure magnetiche	. 143
15.8 Analisi in frequenza, prova 2	. 143
15.9 Output degli accelerometri.	. 144
15.10Distribuzione delle letture dei giroscopi.	. 144
15.11 Variazioni delle misure magnetiche	. 145
15.12Analisi in frequenza, prova 3	. 145
15.13Confronto spento-acceso	. 146
15.14Sequenza di prova 4	. 146
15.15Evidente l'effetto della distorsione magnetica.	. 147
15.16Prova interna, ciclo accensione-spegnimento	. 147
15.18Particolare delle letture degli accelerometri	. 148
15.19Distribuzione delle letture dei giroscopi.	. 149
15.20 Variazioni delle misure magnetiche	. 149
15.21 Accelerazioni prova 7	. 150
15.22Grandezze magnetiche prova 7	. 150
15.23Andamento degli angoli di orientamento prova 7	. 151
15.24Confronto tra le FFT a motore acceso	. 151
15.25Analisi in frequenza della prova a motore acceso: confronto	. 152
15.26Istanti di cambiamento condizioni prova 5	. 153
15.27Andamento orientamento prova 5	. 153
15.28 Valutazione adattatività R prova 5	. 154
15.29Andamento delle tre componenti magnetiche rilevate da Arduino	. 154
15.30Scostamento tra heading di Ekinox ed heading Arduino	. 155
15.31 Differenza di comportamento a seconda del valore di sogli a th impostato.	. 155

15.33Spettrogramma accelerazione $z. \ldots \ldots \ldots \ldots \ldots$. 156
15.34Andamento delle accelerazioni, prova 5, con filtro e senza filtro.			. 157
15.35 Filtraggio di a_x , prova 5			. 157

Elenco delle tabelle

1.1	Sensori: caratteristiche richieste
$5.1 \\ 5.2$	Risultati della analisi di Allan
	di accelerometro e magnetometro
6.1	Collegamenti Arduino-IMU
7.1	Notazione per algoritmo
$11.1 \\ 11.2$	Connessioni Arduino-DAC
	dei collegamenti comuni
11.3	Connessioni Arduino-NI 6123
12.1	Tabella per finestra Workspace 12.1
12.2	Tabella per finestra Workspace 12.2. 101
12.3	Tabella per finestra Workspace 12.3
13.1	Disposizione assi
13.2	Valori utilizzati per lo studio del filtro
14.1	Specifiche BRT
15.1	Prove eseguite. Legenda: Amb = valutazione ambiente, $I = interno, E =$
	esterno, $A =$ veicolo acceso, $S =$ veicolo spento, $SAS =$ serie/spegnimento
15 2	Disposizione assi 138
10.2	

Sommario

In questa tesi viene presentato uno studio sulle grandezze legate alla navigazione autonoma. Più nel dettaglio, viene mostrato come stimare alcune quantità di rilevanza cruciale, almeno per quanto riguarda l'impostazione del progetto che si svilupperà su più anni.

In una prima parte verranno introdotti tutti i sistemi coinvolti nella sensorizzazione del veicolo: unità di navigazione inerziale di alta fascia e altre soluzioni *low cost*, insieme ad una breve introduzione al problema della guida autonoma e del perchè vengono utilizzati proprio questi strumenti.

Nella seconda parte viene illustrato come si intende, nell'ambito di questa tesi, ricavare la grandezza di interesse ovvero l'heading angle.

Nella terza parte vengono sviluppati gli algoritmi per la stima delle grandezze di interesse ed i programmi utilizzati per la costruzione degli stessi.

Nella quarta parte verranno mostrati degli accorgimenti per migliorare l'algoritmo sviluppato nella parte precedente.

Nella quinta parte viene descritto il sistema di acquisizione dei dati utilizzato per la ricezione dei segnali delle unità inerziali, tramite lo sviluppo di un ambiente di lavoro in un software dedicato che verrà illustrato nel dettaglio.

Nella sesta parte vengono mostrati i risultati delle prove sperimentali e i confronti opportuni tra i vari sensori utilizzati.

Questa tesi vuole essere una guida soprattutto per iniziare i colleghi che successivamente si avvicineranno al tema della programmazione delle unità inerziali.

Capitolo 1

Introduzione

Lo scopo di questa tesi è lo sviluppo di un sistema di sensoristica per la navigazione autonoma di un qualsiasi tipo di veicolo.

Il tema della guida autonoma è ormai da molti anni nelle agende dei produttori di autoveicoli e altri mezzi stradali, quindi la strumentazione degli stessi non è un problema che costituisce una novità. Tuttavia, la modernità e novità dell'argomento rappresenta di per sé uno spunto interessante per lo studio di questa tecnologia.

Le caratteristiche che, almeno idealmente, un mezzo autonomo dovrà avere sono:

- la capacità di superare gli ostacoli comunemente disposti sui percorsi stradali;
- un sistema di riconoscimento della posizione e dello spazio circostante preciso e pronto, senza utilizzare algoritmi complessi e lenti;
- la possibilità di riconoscere eventuali guasti ed applicare delle logiche di sicurezza volte alla protezione sia del mezzo stesso sia degli occupanti.

Dalla complessità del problema si può evincere come questo richieda uno studio su più anni; lo scopo di questo lavoro di tesi è quindi quello di gettare le basi per successivi sviluppi, consegnando un sistema di sensorizzazione ed attuazione funzionante e in grado di svolgere semplici manovre.

1.1 Background sulla sensoristica più utilizzata

In letteratura esistono delle soluzioni che si affidano ai sensori più comuni per la stima delle grandezze di interesse.

Diverse pubblicazioni vedono nell'utilizzo di unità inerziali e sensori di visione la tecnologia più efficace per permettere al veicolo di muoversi in autonomia.

In [15], l'impiego di telecamere consente di modellare attraverso un rendering 3D eseguito in tempo reale l'ambiente circostante; in aggiunta a delle mappe con modelli di altitudine da usare con un posizionamento GPS, questa è una delle soluzioni più complete ma anche più dispendiose dal punto di vista della realizzazione.

L'utilizzo di sensori di visione può risolvere alcuni problemi legati alla mancanza di segnale GPS in determinati momenti dell'attività del mezzo. In [11] viene proposto un metodo senza telecamere che sfrutta dei sensori di velocità angolare delle ruote, i cui segnali vengono fusi in un EKF¹ che decide se utilizzare o meno i nuovi segnali GPS. Per gli scopi di

¹Extended Kalman Filter.

questo lavoro non è possibile utilizzare questo tipo di sensori ma è interessante l'approccio alla risoluzione del segnale GPS mancante. Il sistema consente di ottenere precisioni dell'ordine dei 5 cm nel posizionamento del veicolo.

Anche in [2] viene utilizzato un sensore sulle ruote; la differenza dal metodo precedente sta nell'implementazione di un modello cinematico che si svincola quindi dal tipo di veicolo. In più viene installato un sensore ottico per la stima della velocità longitudinale, poiché questa applicazione è pensata per il settore urbano.

Gli accelerometri sono presenti in tutte le pubblicazioni trovate, ma in [10] vengono utilizzati per lo studio delle vibrazioni e la caratterizzazione del veicolo alle basse frequenze, quelle di interesse anche per l'applicazione di questo lavoro di tesi.

In [18] vengono utilizzati degli LVDT per la misura del *displacement* delle sospensioni del veicolo; questo può rappresentare un input interessante anche per il veicolo oggetto di questo studio, ma in questa fase iniziale sarà tralasciato.

In definitiva, i sensori imprescindibili per poter ottenere delle informazioni utili alla navigazione autonoma, ovvero le accelerazioni che il veicolo subisce e gli angoli di rollio e imbardata, chiamati nel seguito **roll** e **yaw**, il quale viene spesso chiamato anche angolo di **heading**, sono le unità di misurazione inerziale o *IMU*, un magnetometro che funga da *compasso elettronico*, due sensori *GPS*, per poter ottenere una stima più precisa dell'angolo di heading e, anche se non oggetto di questa tesi, un *profilometro*. Quest'ultimo strumento consente di costruire degli array di punti che descrivono la conformazione dello spazio circostante il veicolo.

In tab. 1.1 vengono mostrate le caratteristiche che si pensa possano restituire dei buoni risultati per l'applicazione di guida autonoma.

Sensore	Frequenza di lavoro [Hz]	Altre caratt.
IMU e magnetometro	50-200	Triassiali
Wheel Speed Sensor	50-200	non necessario
GPS/GNSS	1-10	più vicini ai 10 Hz possibile
Profilometro	10-100	

Tabella 1.1: Sensori: caratteristiche richieste.

1.2 L'obiettivo della tesi

Tra le grandezze di interesse per la guida autonoma, la capacità di riconoscere la direzione di marcia del veicolo è senza dubbio una caratteristica di fondamentale importanza per consentire ad un veicolo di orientarsi nello spazio.

Tra gli strumenti mostrati in 1.1, si è scelto di utilizzare delle unità inerziali già disponibili presso i laboratori dell'Ateneo, con lo scopo di sviluppare degli algoritmi per la stima dell'angolo di *heading*.

In particolare, si farà uso di unità a 9 g.d.l., ovvero dotate di tre sensori triassiali accelerometro, giroscopio e magnetometro. Come precedentemente indicato, infatti, questi sono ormai disponibili in molte varianti e fasce di prezzo abbordabili e possono costituire, con i dovuti accorgimenti, delle buone soluzioni.

Nel seguito verranno mostrati tutti i sistemi utilizzati e le modalità per programmarli, nonché i risultati di alcune prove sperimentali per giudicarne l'efficacia.

Capitolo 2

Stimare l'angolo di heading

La conoscenza dell'orientamento di un veicolo è uno dei fattori fondamentali per le applicazioni di guida autonoma. Oltre a conoscere la posizione, è infatti necessario fornire una precisa indicazione su dove effettivamente un veicolo sia diretto sia per questioni di navigazione, sia per questioni di sicurezza; non si può escludere, in caso di slittamento o fenomeni analoghi, che la direzione in cui il veicolo è diretto e il suo orientamento sul piano orizzontale diano la stessa informazione.

Ad oggi esistono diversi metodi per la stima dell'orientamento o heading di un veicolo..

Il primo metodo è quello più "tradizionale", ovvero quello che sfrutta la conoscenza del campo magnetico per poter individuare in ogni istante il nord. In sostanza, si potrà definire come è orientato il veicolo in un piano $x, y \in z$ assimilabile ad un sistema NED¹ o NWU² fisso e solidale con il terreno. Questo metodo riprende il funzionamento delle classiche bussole.

Il secondo metodo è basato su un sistema denominato *Dual GPS*. Sfruttando il segnale di posizione fornito da due antenne GPS montate su un veicolo è possibile calcolare un vettore che simuli l'orientamento del veicolo sulla superficie terrestre e, da altre misure fornite dal GPS stesso, valutarne l'orientamento sul piano orizzontale. Questo metodo non verrà sviluppato nel corso di questo lavoro.

Entrambi i metodi presentano vantaggi e svantaggi.

Il metodo basato sul campo magnetico risulta essere inaffidabile se attorno al sensore sono presenti troppi oggetti ferrosi, i quali distorcono il campo stesso generando delle misure non attendibili. Il campo magnetico terrestre è però sempre rilevabile, a differenza dei segnali GPS che possono talvolta essere non ricevuti dalle antenne a causa di condizioni meteo non favorevoli o passaggio in ambienti chiusi e quindi non fornire l'informazione per un tempo più o meno lungo. I segnali GPS non sono però influenzati da materiali ferrosi, rendendo la loro applicazione molto utile in ambito automotive.

2.1 Stima mediante flusso magnetico

Il campo magnetico della Terra ha una componente parallela alla superficie terreste che punta sempre verso il nord magnetico. Questo è il concetto alla base di tutti i compassi³. La Terra è approssimabile con un dipolo, del quale le già citate componenti parallele alla

¹Nord East Down

²North West Up

³Termine con cui vengono indicate le bussole "digitali".

superficie vengono utilizzate per determinare la direzione; la componente verticale, che dipenderebbe anche dall'inclinazione con cui il flusso magnetico interseca la superficie, viene ignorata.

Si differenziano nord magnetico e nord geografico o vero. In ogni punto della superficie questa distanza angolare tra i due nord varia e prende il nome di *declinazione magnetica*, la quale si può calcolare tramite delle tabelle disponibili in diverse fonti.

Il valore dell'heading o yaw si può ottenere considerando la fig. 2.1. Date le componenti sul piano parallelo alla superficie $H_x \in H_y$, la somma in quadratura fornisce l'intensità del campo in direzione nord H_{north} , mentre il seguente rapporto

$$\alpha = \tan^{-1} \left(\frac{H_y}{H_x} \right) \tag{2.1}$$

fornisce l'angolo che la direzione x, coincidente con la mezzeria di un veicolo in prova, forma con la direzione nord stessa.



Figura 2.1: Definizione dell'angolo di heading.

La (2.1) è valida soltanto se la misurazione avviene sempre sul piano. Nel caso il compasso venga ruotato, occorrenza che si ha in molte applicazioni, è necessario compensare gli errori introdotti dalla nuova inclinazione del magnetometro. Con gli strumenti della trigonometria si possono riportare le tre componenti del flusso di campo magnetico sul piano orizzontale alla superficie. Detti $\phi \in \theta$ rispettivamente gli angoli di rollio e beccheggio, si può applicare la seguente trasformazione

$$\begin{cases} H_{xc} = H_x \cos \phi + H_y \sin \theta \sin \phi + H_z \cos \theta \sin \phi \\ H_{yc} = H_y \cos \theta - H_z \sin \theta \end{cases}$$
(2.2)

dove con il pedice csono indicate le grandezze compensate. La formula in $\left(2.1\right)$ diventa allora

$$\alpha = \tan^{-1} \left(\frac{H_{yc}}{H_{xc}} \right)$$

2.2 Problematiche legate all'uso di magnetometri

Utilizzare il campo magnetico terrestre per ottenere l'orientamento è un buon metodo nel caso sia possibile effettuare delle rilevazioni esenti da disturbi. Tuttavia, in ambienti urbani o indoor, la presenza di materiali ferromagnetici o di sorgenti di campi magnetici possono degradare la qualità della misura dell'angolo.

In generale, queste distorsioni nel campo magnetico terrestre sono di varia entità e non è

sempre possibile distinguerle con accuratezza.

Nell'ambito di questa tesi verranno introdotti dei metodi per elaborare e filtrare questi disturbi come mostrato in diverse pubblicazioni 4 .

Negli ambiti civili, come l'utilizzo di telefoni cellulari o apparecchiature elettroniche simili, questi disturbi non vengono compensati in quanto non è scopo primario di questi dispositivi il calcolo delle grandezze di orientamento, come i tre angoli di Eulero, in modo accurato [23]. Anche nell'ambito automotive, spesso nelle vetture è disponibile una bussola a otto direzioni, ma non un sensore che restituisca la misura esatta dell'angolo di heading. Per la navigazione autonoma, anche per sopperire all'improvvisa mancanza di un segnale GPS, è invece di fondamentale importanza avere una stima precisa.

Il magnetometro che verrà utilizzato nell'ambito di questa tesi, mostrato più avanti, non è esente dalle problematiche sopra esposte. Il suo prezzo competitivo, rispetto ad altri sistemi $AHRS^5$, ne consente l'utilizzo in linea con gli scopi esplorativi di questo lavoro di tesi.

 $^{{}^{4}}_{[1]}, [31], [20], [3] \text{ et al.}$

⁵Attitude and Heading Reference System

Capitolo 3

Sensori e schede

In questo capitolo vengono mostrati i sensori che verranno utilizzati nelle varie prove sperimentali per la determinazione delle grandezze di interesse del veicolo, ovvero le accelerazioni, l'angolo di heading e di rollio e la posizione del veicolo stesso.

Vengono prese in esame essenzialmente due soluzioni, una di elevate prestazioni e costo con la quale si generano dei segnali che verranno presi come riferimento, un'altra basata su sistemi Arduino, caratterizzata dal costo estremamente contenuto e dalla sfida intrinseca di programmazione per ottenere dei risultati paragonabili a quelli di riferimento in termini di prontezza e precisione.

3.1 Unità inerziale SBG Ekinox D

L'unità di misurazione inerziale (INS) **Ekinox D** è basata sulla tecnologia MEMS¹ ed è in grado di ottenere un'elevata precisione ed un'elevata accuratezza in un pacchetto compatto. Include una unità di misura inerziale (IMU) ed ha un algoritmo di Extended Kalman Filter incluso che consente di effettuare la cosiddetta *sensor fusion* per migliorare le prestazioni. Il sensore può montare anche due antenne GPS per ottenere la massima precisione di posizionamento e di stima delle grandezze di navigazione come l'heading angle.



Figura 3.1: L'unità Ekinox D.

All'unità è possibile interfacciare anche sensori esterni per aumentare il campo di applicazioni specifiche.

La tecnologia MEMS consente di avere dei design miniaturizzati che restituiscono dispositivi leggeri ed affidabili in quanto si sostituiscono quasi del tutto parti meccaniche.

¹Micro Electronics Measurement System.

Questo consente anche di ottenere una maggiore resistenza agli urti.

Gli *accelerometri*, uno per asse, sono di tipo capacitivo e filtrati internamente in modo da ottenere una misura estremamente stabile e non influenzata da rumore.

I tregiroscopimono
assiali restituiscono un segnale esente da rumore dovuto alle vibrazioni.

Tre magnetometri magnetoresistivi assicurano performance elevate anche grazie ai diversi filtri integrati nell'unità.

Il filtro di Kalman integrato lavora in modo da restituire sempre la massima precisione possibile e si divide in diversi livelli di calcolo, come mostrato nel manuale [30]. In breve, la maggiore precisione sarà ottenuta con due antenne GPS, mentre la minore precisione si otterrà utilizzando il solo segnale del giroscopio.

Per l'heading angle, ad esempio, l'unità Ekinox dapprima utilizza il solo giroscopio ed eventualmente un singolo segnale GPS; appena disponibile, verrà utilizzato il segnale del magnetometro per poter avere una stima anche in stazionario; se in movimento, l'Ekinox sfrutterà il segnale GPS e le accelerazioni; infine, se un sistema dual antenna è disponibile, questo sarà sempre preferito per la stima.

La caratteristica principale di questa unità è la sua facile programmabilità attraverso l'interfaccia Web.

	Main Screen Global Status Information Rawy	values	
General	Solution	Aiding Inputs	
Main Power 🗸	Solution mode Nav position	Velocity Heading Positi	ion UTC
Imu Power 🗸	Quality	GPS 1 🗸 🕺 🗸	~
GPS Power 🗸	Position	GPS 2 🗶 🖌 🖌	×
Settings 🗸	Velocity	Odo. 🗡	
Temperature 🗸	Attitude	Mag. 🗸	
Data Logger 🗸	Heading	DVL X	
		USBL X	
IMU	Accepted inputs	EM 🗡	
General	GPS1 Position	Depth 🖌	
Communication 🗸	GPS1 Velocity 🗸	User 🗙 🗶 🗴	
Built In Test	GPS1 Course X		
	GPS1 True Head.	Interfaces	
Sensors	GPS2 Position 🖌	Opened Receive T	ransmit
x y z In Range	GPS2 Velocity X	Com A V V	1
Accelero	GPS2 Course 🕺	Com B 🖌 🖌	<u> </u>
Gyrn V V V	GPS2 True Head.	Com C X X	2
0110	User Position X	Com D 🗸 🗸	
GPS 1	User Velocity X	Com E 🖌 🗸	
	User Heading	Eth 0 🗸	
Position Status Solution computed	Odometer x	Eth 1 🗸	
Position Type Single Point	Magnetometer V	Eth 2 🖌	
Velocity Status Solution computed	USPI	Eth 3 💉	
Velocity Type Doppler	USDL ×	Eth 4 🖌	
Heading Status Insufficient obs.	Heave		
GPS LILZLS		Clock	
GLONASS LI L2	Valid V	Input	
GPS 2	Velocity compens. V	Input Clock	1
		Clock Aligoment	Walid
Position Status Insufficient obs.	Magnetometer		+ 840
Position Type No Solution	и и т . То Озород	UTC	
Velocity Status Insufficient obs.	x y z in kange	UTC synchro	~
Velocity Type No Solution	Magnetometer V V V	UTC info	Valid
Heading Status Insufficient obs.	Accelerometer V V V		
GPS L1 L2 L5			

Figura 3.2: Pannello status globale.

	Main Screen Global	Status Information Raw values
Device details		Firmware Details
Product Code :	EKINOX-D-G4A2-PS	Firmware version
Device ID :	023000023	Current version: 1.0.3.0
Hardware revision :	1.2.0.0	Upload new firmware
IMU serial :	030000041	
IMU revision :	1.2.2.0	Activated options Upload new license
Calibration date :	11/07/13	Master GPS Generic hardware Unlimited
Calibration version :	1.0.0.0	Rever GPS Generic hardware Unlimited
Network		Storage
FTP access :	ftp://192.168.1.138:21	Data logger disabled
MAC Address :	98:5C:93:00:00:38	3.5 GB free
IP :	192.168.1.138	
Mask :	255.255.255.0	Begin a new session

Figura 3.3: Pannello status globale.

Nella fig. 3.3 è mostrato il pannello di informazioni utili riguardo l'unità. La più importante è l'indirizzo IP col quale è possibile selezionare la rete Ethernet su cui impostare la scheda Ethernet del PC al quale viene collegata la Ekinox.

3.1.1 Collegare la Ekinox

La Ekinox viene fornita insieme ad un elemento aggiuntivo chiamato Split Box (in fig. 3.6), il quale consente di collegare l'unità a diversi dispositivi tramite cinque porte seriali, una porta CAN e una porta Ethernet. Il collegamento fisico tra Ekinox e Split Box avviene tramite tre cavi, mentre tra Ekinox e PC avviene tramite un cavo crossover con attacchi M12-RJ45.

Per potersi interfacciare con la Ekinox, è necessario impostare la propria scheda di rete Ethernet in modo da ottenere un indirizzo IP assegnato automaticamente, come in fig. 3.4. Successivamente aprire un browser (Chrome o Firefox) ed inserire il seguente indirizzo

http://ekinox_02000001.local.

dove il numero è il seriale che si trova sull'etichetta dell'unità. Una volta che si è ottenuto l'accesso all'interfaccia Web, si può controllare l'indirizzo IP come in fig. 3.3 e reimpostare la propria scheda Ethernet (cliccando su Protocollo Internet Versione 4) in modo da avere un indirizzo IP fisso² e non modificare quello dell'unità, come mostrato in fig. 3.5.

Nota: Dopo aver ottenuto l'IP sarà possibile accedere dal browser anche nel seguente modo, supponendo di utilizzare l'IP di fig. 3.5

http://192.168.1.120

 $^{^{2}}$ Se questa impostazione viene modificata per altri utilizzi, per poter ricollegare la Ekinox senza modificare il suo indirizzo IP occorre reimpostare manualmente l'IP della scheda.

~	→ \vee ↑ 隆 > Pannello di controllo > Rete e Internet > Connessioni di rete									
O.	Pro	prietà - Ethernet	Econui diagnosi del X	connessione	Rinomina connessione	Cambia impostazioni connession				
1	Rete	Condivisione								
2	Conne	etti tramite:		eros AR9485WB-E						
	7	Realtek PCIe GBE Family Controller								
			Configura							
	La co	nnessione utilizza gli elementi seguenti:								
		Utilità di pianificazione pacchetti QoS	^							
		Protocollo Internet versione 4 (TCP/IP) Protocollo Microsoft Network Adapter N	V4) Multiplexor							
		Driver protocollo LLDP Microsoft	-							
		Risponditore individuazione topologia li	veli di collegamento							
	•	L Driver di I/O del mapping di individuazi	one topologia livelli c 🗸							
	<		2							
		Installa Disinstalla	Proprietà							
	Descrizione TCP/IP. Protocollo predefinito per le WAN che permette la									
	CO	municazione tra diverse reti interconnesse.								

Figura 3.4: Configurazione inziale Ethernet.

Proprietà - Protocollo Internet version	ne 4 (TCP/IPv4) ×
Generale	
È possibile ottenere l'assegnazione au rete supporta tale caratteristica. In ca richiedere all'amministratore di rete le i	tomatica delle impostazioni IP se la so contrario, sarà necessario mpostazioni IP corrette.
Ottieni automaticamente un indiri	zzo IP
• Utilizza il seguente indirizzo IP: -	
Indirizzo IP:	192 . 168 . 1 . 130
Subnet mask:	255 . 255 . 255 . 0
Gateway predefinito:	
Ottieni indirizzo server DNS autor	naticamente
Utilizza i seguenti indirizzi server l	DNS:
Server DNS preferito:	
Server DNS alternativo:	
Convalida impostazioni all'uscita	Avanzate
	OK Annulla

Figura 3.5: Impostazione IP fisso PC.

3.1.2 La porta CAN

Questa unità presenta sullo Split Box una porta CAN, che facilita enormemente l'interfaccia con il controllore PXI, mostrato in fig. 3.6. Per il funzionamento è necessario un cavo CAN con attacco DB-9 da 60 Ω per terminazione presente in laboratorio.



Figura 3.6: Lo Split Box.

Ulteriori informazioni sulla unità si possono trovare sul manuale [30], disponibile presso i laboratori del DIATI.

3.2 Arduino e sensori

Le schede Arduino, presenti sul mercato da ormai dieci anni, sono un perfetto compromesso tra la prototipazione rapida low cost e soluzioni integrate particolarmente prestazionali grazie all'elevata compatibilità con numerosi sensori di terze parti. La programmazione, che avviene in un ambiente di sviluppo dedicato, facilita la compilazione dei programmi scritti in un linguaggio derivato dal C++.

Nel seguito verranno brevemente discussi i componenti che sono stati utilizzati nel lavoro di tesi. Questi sono:

- Keyestudio Mega 2560 (una scheda equivalente Arduino Mega);
- GY-87 breakout board;
- Mouser DAC MCP4921.

3.2.1 Scheda Arduino

La scheda Arduino è il cuore di ogni progetto Arduino. Viene chiamato *controller* visto che è la parte dell'assieme che riceve, calcola e invia i dati che arrivano da/a tutti gli altri componenti.

È sempre formato da un microprocessore prodotto da Atmel, una porta seriale o interfaccia per la comunicazione col computer³ o altri componenti, un connettore per l'alimentazione e diversi pin analogici e digitali I/O per la comunicazione con i dispositivi esterni.

Può essere alimentato tramite la porta USB o con le batterie, che possono essere collegate all'alimentazione o ai pin 5 V o 3.3 V, i quali funzionano sia come input sia output.

Esistono diversi cloni di Arduino, dal momento che il *blueprint* del circuito è *open source*. Quindi ogni scheda funziona come le originali.



(b) Keyestudio Mega 2560.

(a) Arduino Uno R3.

Figura 3.7: Le schede Arduino.

Le schede Arduino si possono programmare tramite l' \mathbf{IDE}^4 ufficiale, scaricabile dal sito, o con qualsiasi IDE che supporti il linguaggio C. Il programma ufficiale è molto utile in quanto supporta l'indentazione automatica, controllo delle parentesi, highlight della sintassi e un compilatore integrato, chiamato **avrdude**. Permette inoltre il *deploy* degli *sketch*,

 $^{^3}$ In realtà, la comunicazione tra Arduino e PC è tramite USB. Sulla scheda sono montati dei chip USB-to-Serial che traducono i segnali tra queste due parti.

⁴Integrated Development Environment.



Figura 3.8: Arduino IDE.

come vengono chiamati i programmi, in maniera diretta; ha inoltre un visualizzatore incorporato, detto *Serial Monitor*.

Nel corso di questa tesi si considerano già note le basi di programmazione Arduino. In rete è possibile trovare innumerevoli guide e lezioni nel caso si programmi per la prima volta una scheda Arduino.

3.2.2 Accelerometro e giroscopio MPU 6050

La board MPU 6050 è composta di un accelerometro e un giroscopio entrambi MEMS. L'uso combinato di questi due chip permette di ottenere un device per il motion tracking a sei gradi di libertà (6-DOF).

L'unità è costituita da una coppia di tre convertitori ADC⁵ da 16 bit per accelerometro e giroscopio. L'utente può programmare la sensibilità dei due chip su diversi massimi, ovvero 250/500/1000/2000 gradi al secondo per il giroscopio e 2/4/8/16 g^6 per l'accelerometro.

L'MPU 6050 utilizza il protocollo I^2C . Questo è un protocollo seriale che utilizza una linea per inviare i dati (la **SDA** o *Serial DAta line*) e una per la sincronizzazione di questi bit di dati (la **SCL** o *Serial CLock line*).

 $^{^{5}}$ Analog-to-digital

 $^{^{6}\}mathrm{L'accelerazione}$ è espressa in termini di gravità.



Figura 3.9: MPU 6050.

Per estrarre i dati da questa unità, si può programmare Arduino in modo da accedere direttamente ai registri mostrati nel datasheet del sensore o utilizzare delle librerie specifiche, ovvero la I2Cdev.h e la MPU6050.h, in cui sono scritte delle funzioni più "user friendly" per accedere a questi registri e quindi alle informazioni richieste.

Per completezza, si riporta in breve il funzionamento di un accelerometro, con l'ausilio di fig. 3.10. Vengono utilizzate diverse masse per ciascuno degli assi. L'accelerazione lungo un certo asse induce uno spostamento della corrispondente massa, e dei sensori capacitivi rilevano questo spostamento in maniera differenziale. Concettualmente, l'accelerometro si comporta come un sistema massa-molla-smorzatore.





Per maggiori spiegazioni sul funzionamento degli accelerometri, si vedano [23] e [25].

Un giroscopio rileva le rotazioni su tre assi indipendenti tramite dei sistemi vibranti MEMS. L'idea generale è che, quando l'unità viene ruotata attorno ad un asse, l'accelerazione di Coriolis genera una vibrazione che viene rilevata da un sensore di tipo capacitivo. Questo segnale è poi amplificato, demodulato e filtrato per produrre una tensione proporzionale alla velocità di rotazione.

3.2.3 Magnetometro HMC5883L

Il sensore **HMC5883L** è un modulo progettato per la rilevazione di campi magnetici. Costituito da diversi chip, questi sono costruiti allo stato solido con una *cross-axis sensitivity* molto contenuta per la misura del campo magnetico terrestre sia in direzione che intensità, da pochi milligauss fino a 8 gauss.

Il circuito magnetoresistivo del sensore è costituito da tre circuiti più piccoli e circuiti specifici di supporto per la misura del campo magnetico. Quando alimentato, il sensore

converte qualsiasi campo magnetico incidente sugli assi di misura in una tensione. In presenza di un campo, un cambio nel ponte resistivo genera quindi un cambiamento di tensione sugli output del ponte.

Anche l'HMC5883L utilizza il protocollo I^2C ed è possibile accedere direttamente ai registri manualmente o utilizzando la libreria HMC5883L.h.

Ci sono molti aspetti da tenere in considerazione quando si utilizza un magnetometro: la presenza di oggetti di ferro nelle vicinanze modifica molto il campo magnetico e quindi il segnale registrato; è necessaria una calibrazione accurata (mostrata più avanti) e una correzione data dalla differenza tra nord vero e magnetico [7].

Un altro aspetto cruciale di questo sensore è la dipendenza dalla temperatura: le letture sono influenzate dal valore di quest'ultima e algoritmi di compensazione devono essere implementati per correggere l'output, se l'applicazione lo richiede.

3.2.4 Breakout board GY-87

La **GY-87** è una MPU a 10-DOF (contando anche il sensore di pressione integrato, che però non verrà utilizzato) che incorpora la MPU 6050 e l'HMC5883L. Il vantaggio di utilizzare questa board è dato dal minor numero di cavi richiesto per la comunicazione.



Figura 3.11: GY-87.

3.2.5 Modulo Mouser DAC4921

Questo chip integrato è un convertitore D/A che può essere utilizzato per la conversione di un segnale in uscita da un pin dell'interfaccia seriale di Arduino in un segnale analogico. È programmabile attraverso una libreria apposita, il cui scopo è quello di utilizzare il pin selezionato come un normale pin analogico. In breve, il circuito interno al DAC opera sul contenuto in bit del messaggio scritto sull'interfaccia seriale ed opera una conversione per generare un valore di tensione proporzionale a questo.



Figura 3.12: DAC 4921.

Questo verrà utilizzato per l'interfaccia della scheda Arduino con la scheda Multifunction I/O disponibile sul PXI. Collegando infatti un pin PWM ad un ingresso digitale (o anche analogico) di detta scheda, verrà istantaneamente acquisito il livello di tensione massimo o minimo e non un segnale continuo.

Nella sezione *Prove real-time e confronti* verrà mostrato più in dettaglio l'utilizzo di questo componente.

3.3 Caratteristiche dei sensori

In questa sezione vengono brevemente riassunte le caratteristiche dei sensori utilizzati.

3.3.1 Sensore Ekinox

- Accelerometro
 - Scala: 2 g⁷;
 - Rumore: 90 μ g (tra 1 e 25 Hz);
 - Larghezza di banda: 100 Hz (attenuazione $\leq 3dB$);
 - Risoluzione: 0.0001 g;
- Giroscopio
 - Scala: 62 $\frac{rad}{s}$;
 - Rumore: 0.3 $\frac{rad}{s}$ (tra 1 e 25 Hz);
 - Larghezza di banda: 100 Hz (attenuazione $\leq 3dB$);
 - Risoluzione: 0.03 $\frac{rad}{s}$;
- Magnetometro
 - Scala: 6 G;
 - Rumore: 50 $\mu {\rm G}$ (tra 1 e 25 Hz);
 - Larghezza di banda: 50 Hz (attenuazione $\leq 3dB$);
 - Risoluzione: 120 μ G;

3.3.2 Sensori MPU6050+HMC5883L

- Giroscopio
 - Scala: 250 $\frac{deg}{s}$;
 - Rumore: 0.0707 $\frac{deg}{s}$;
 - Risoluzione: 0.008 $\frac{deg}{s}$;
- Accelerometro
 - Scala: 2 g;
 - Rumore: 127 μ g;
 - Risoluzione: 0.00006g
- Magnetometro
 - Scala: 1.3 G;
 - Risoluzione: 0.92 mG;
 - Rumore: 2 mG;

⁷Gravità.

3.4 La calibrazione

Tutti i dispositivi, anche quelli di fascia più elevata, necessitano di una procedura di calibrazione più o meno complessa o che necessita di uno o più passi. Questo perché ogni sensore può essere affetto da tipi diversi di problemi, come il disallineamento degli assi (gli assi segnati sulla board potrebbero non coincidere con quelli effettivi del sensore), errori di offset, di orientamento, di sensibilità, di avvio.

In questo paragrafo viene mostrato come calibrare i vari sensori utilizzati in questo lavoro.

3.4.1 Idea di base

La calibrazione può essere eseguita in diversi modi. L'idea generale è che il dato grezzo sia trasposto attraverso la moltiplicazione con una matrice 3x3 che indica il disallineamento tra gli assi del sensore e della board su cui è montato e decurtato dell'offset, ovvero la distanza tra le origini nelle tre direzioni. I valori di questi due array possono essere ottenuti tramite la media di diverse letture effettuate sui tre assi.

Dal punto di vista matematico, si può scrivere come

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix}_{cal} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{11} & c_{12} & c_{13} \\ c_{11} & c_{12} & c_{13} \end{pmatrix} \begin{pmatrix} a_{raw} - off_a \\ b_{raw} - off_b \\ c_{raw} - off_c \end{pmatrix}$$
(3.1)

Maggiori indicazioni si possono reperire in [19]. È possibile anche utilizzare dei metodi disponibili in letteratura che facilitano il processo di calibrazione stesso, alcuni dei quali verranno mostrati nel seguito.

3.4.2 Sensori esterni di Arduino

Arduino in sé non necessita calibrazione, mentre la IMU e il magnetometro prevedono delle procedure per correggere gli errori sopra esposti.

Un problema comune con i primi due sensori è dovuto al fatto che, anche se fermo e in piano, il sensore rilevi delle quantità diverse da zero. Questo è un problema facilmente rilevabile nell'accelerometro: se lasciato su un piano si rilevano (oltre al rumore) delle accelerazioni anche sugli assi $x \in y$, il sensore necessita di una calibrazione. Anche nel magnetometro, dopo aver eseguito alcuni passaggi di elaborazione dei dati [4], se il sensore punta in una direzione diversa dal nord si può riscontrare questo problema.

Accelerometro

L'MPU 6050 può essere calibrato tramite un algoritmo disponibile sul sito della I2Cdev, una fondazione che si è occupata della scrittura delle librerie per questo sensore. È un tool molto compatto ed esegue la calibrazione automaticamente dopo aver fornito unicamente dei valori di offset di primo tentativo dai quali l'algoritmo parte per effettuare le misure, insieme ad altri parametri come la sensibilità richiesta e la velocità di comunicazione seriale per la visualizzazione dei progressi. In fig. 3.13 è mostrato l'output del programma. Alla fine del processo di calibrazione, i risultati saranno mostrati in una breve lista e i valori inseriti nei propri programmi. 💿 COM12 (Arduino/Genuino Mega or Mega 2560)

L									
Initializing I2C devices									
Testing	esting device connections								
MPU6050	connecti	lon	succe	essful	L				
Average	reading	of	7861	with	x	accel	offset	of	-1179
Average	reading	of	6939	with	x	accel	offset	of	-1279
Average	reading	of	6016	with	x	accel	offset	of	-1379
Average	reading	of	5089	with	x	accel	offset	of	-1479
Average	reading	of	4164	with	x	accel	offset	of	-1579
Average	reading	of	3238	with	x	accel	offset	of	-1679
Average	reading	of	2313	with	x	accel	offset	of	-1779

Figura 3.13: Calibrazione dell'MPU 6050.

Magnetometro

F.

La calibrazione del magnetometro è la parte più critica di un compasso elettronico. Generalmente, questi dispositivi *low cost* non sono pronti all'uso e richiedono una fase di calibrazione prima di poter estrarre dal sensore delle informazioni utili.

In questa sottoparte verrà messo in luce il procedimento alla base della calibrazione ed evidenziato un indicatore che consente di comprendere la correttezza della procedura svolta.



Figura 3.14: Effetti dovuti alle diverse sorgenti di disturbo. Fonte: SBG Systems.

Come mostrato in [24], in un magnetometro vengono rilevati, oltre al campo magnetico terrestre, delle componenti distorsive, comunemente chiamate hard-iron e soft-iron offsets, mostrate in fig. 3.14. Queste influenzano le letture del magnetometro H and and o rispettivamente a spostare l'origine degli assi di misura del magnetometro le prime e modificando il guadagno dei tre assi le seconde. In termini matematici si può scrivere

$$H_{read} = W_{soft} R H_{earth} + V_{hard} \tag{3.2}$$

dove con R si intende una matrice 3x3 di rotazione che tiene conto dell'orientamento del sensore, con W_{soft} una matrice 3x3 con le proprietà degli effetti soft-iron e V_{hard} il vettore degli offset, mentre H_{earth} è il vettore delle letture del magnetometro espresso in un riferimento fisso che punta a nord. Se dunque con un processo di calibrazione si riescono a stimare correttamente V_{hc} e W_{sc} , per una qualsiasi rotazione del sensore dovrà valere la seguente

$$H_{cal} = W_{sc}^{-1}(H_{read} - V_{hc}) = W_{sc}^{-1}(W_{soft}RH_{earth} + (V_{hard} - V_{hc}))$$

Dunque, a procedura ultimata, si otterrà che $V_{hc} = V_{hard}$ e $W_{sc} = W_{soft}$. Ricordando che la matrice di rotazione R è una matrice ortogonale, si otterrà che il luogo geometrico dei punti descritti dalle letture del magnetometro sarà una sfera di raggio pari all'intensità del campo misurato, idealmente quello terrestre

$H_{cal}^T H_{cal} = (RH_{earth})^T RH_{earth} = H^2$

dove il vettore $H_{earth} = [H \cos \delta \ 0 \ H \sin \delta] \ \text{con } \delta$ angolo di inclinazione magnetica⁸.



Figura 3.15: Figura che mostra il luogo delle misurazioni prima (a destra) e dopo (a sinistra) la calibrazione. Da [24].

I programmi per la calibrazione

Piuttosto che implementare una procedura di calibrazione, nell'ambito di questa tesi è stato utilizzato un software chiamato **Magneto 1.2** (fig. 3.16), il quale implementa la matematica descritta in [24]. In sostanza, dato un file di letture grezze del magnetometro, ottenuto orientando il sensore nel maggior numero di direzioni possibile, il programma cerca di risolvere il problema di fitting descritto in precedenza fornendo poi la matrice di scaling da un generico ellissoide a una sfera di raggio B e il vettore V.

Questi possono poi essere implementati in un codice o funzione che consente di ricavare i valori calibrati ad ogni chiamata della funzione stessa a partire dai dati grezzi.

⁸L'angolo formato dal vettore campo magnetico terrestre con un piano parallelo alla superficie terrestre. In ogni punto della Terra ha un valore diverso ma costante nel tempo.

Norm of Magnetic or Gravitational field (same units as the raw measurements	d: 47.279					
Raw magnetic measurements (h) file:	aw magnetic measurements (h) file:					
C:\Users\salva\Downloads\cal.txt						
	Calibrate					
Combined bias (b):	3.663645	-3.966539	-7.525354			
Correction for combined scale factors,	0.973346	0.012204	0.012745			
misalignments and soft iron (A $^{-1}$.	0.012204	0.950111	0.007585			
	0.012745	0.007585	1.141409			
h _{cal} = A ⁻¹ *(h-b)						
Combined scale factors, misalignments	1.027697	-0.013110	-0.011388			
and soft iron (A):	-0.013110	1.052731	-0.006849			
(for comparison to MagCal)	-0.011388	-0.006849	0.876283			
From: www.sailboatinstruments.blogspot	t.com		Quit			

Figura 3.16: http://www.magnetic-declination.com/

Con altri tool è possibile visualizzare la sfera di calibrazione. **MagViewer** (fig. 3.17) traduce i dati inviati sulla porta seriale da una scheda Arduino su cui è stato scritto esclusivamente il codice che estrae i dati calibrati a partire da quelli grezzi. In questo modo è possibile capire se il sensore è stato correttamente calibrato.



Figura 3.17: MagViewer e i dati correttamente calibrati.

Da ultimo, c'è da tenere in considerazione che in ogni luogo della terra esiste uno scostamento tra il polo nord magnetico e il polo nord geografico. Se si vuole ottenere una misura che punti verso il nord geografico, all'equazione (2.1) va aggiunto algebricamente un termine circa costante, se non ci si sposta considerevolmente durante le misure, detto *declinazione*. Questo è mappato per ogni punto della Terra ed è ricavabile facilmente sul web come in fig.3.18.



Figura 3.18: http://www.magnetic-declination.com/

Capitolo 4

Filtro di Kalman per il calcolo dell'heading angle

4.1 Filtro di Kalman

Il filtro di Kalman è uno strumento che consente di ottenere la migliore stima possibile dello stato di un sistema, il quale può essere o non essere osservabile direttamente. Questo deriva dalla linearizzazione della rappresentazione nello spazio degli stati di un sistema dinamico o cinematico, ovvero

$$\begin{cases} \dot{x} = Fx + Bu\\ z = Hx + Du \end{cases}$$
(4.1)

La discretizzazione di (4.1) porta alla (4.2)

$$\begin{cases} x_t = Fx_{t-1} + Bu_t + w_t \\ z_t = Hx_t + Du_t + v_t \end{cases}$$
(4.2)

dove:

- x_t è la stima degli stati del sistema al tempo corrente;
- x_{t-1} è la stima al tempo precedente;
- z_t è la misurazione, un dato o una serie di dati *noti e misurabili* del sistema in esame, al tempo corrente;
- u_t è l'input, una quantità nota e rilevabile, al tempo corrente;
- $w_t \in v_t$ sono rispettivamente i rumori di processo e di misura;
- F è la state transition matrix, ovvero l'operatore che lega lo stato precedente al successivo;
- *B* è la *control input matrix* ovvero la matrice che lega la stima agli input del sistema;
- $H \in D$ sono le matrici che legano rispettivamente le stime e gli input alle misurazioni; ad esempio, se le misurazioni sono velocità e le stime posizioni, H dice come legare queste quantità fisiche diverse.

Una ipotesi fondamentale del filtro di Kalman è che i rumori v_t e w_t siano rumori bianchi con distribuzione gaussiana a media nulla. In particolare, per comprendere il significato di questi termini, si può ragionare come segue. L'acquisizione di un segnale da un sensore è un processo caratterizzato da diversi tipi di rumore, legati alle caratteristiche costruttive del sensore stesso, alla tipologia di segnale che esso invia e alla conversione di quest'ultimo. Tutti questi contributi possono essere inclusi in v_t . Generalmente, si può assegnare a questo termine il significato di varianza delle misure, come dimostrato in [28].

In un modello cinematico o dinamico, il rumore w_t indica l'indeterminazione o l'incertezza associata alla stima della grandezza di interesse. È un valore più difficile da selezionare in quanto non misurabile, e di solito viene tarato per migliorare le caratteristiche del filtro.

Per poter stimare gli stati del sistema queste equazioni non sono sufficienti. Nel filtro di Kalman si hanno cinque passaggi, che consentono ogni volta di comprendere quanto la misura sia vicina o lontana dalle quantità vere, che possono solo essere stimate statisticamente. In [12] è possibile vedere come, considerando la misura e la stima come due Gaussiane, sia possibile risalire alla formulazione matriciale esposta in seguito.

I passaggi prevedono la stima di uno stato detto *a priori*, ovvero il risultato che si otterrebbe da una stima perfetta, che poi verrà corretto dalla misura in maniera tanto maggiore quanto più è inaffidabile la stima. Si hanno due macro passaggi, il *time update* e il *measurement update*. Nel *time update* si hanno:

1. stima dello stato a priori:

$$x_{t_{-}} = Fx_{t-1} + Bu_t + w_t \tag{4.3}$$

2. calcolo della matrice di covarianza $P_{t_{-}}$, che indica la varianza e la covarianza degli stati del sistema:

$$P_{t_{-}} = F P_{t-1} F^T + Q (4.4)$$

dove Q è la matrice della covarianza di processo, legata al rumore presente nelle stime;

mentre nel measurement update:

1. calcolo del guadagno di Kalman:

$$K = P_{t_{-}}H^{T}(HP_{t_{-}}H^{T} + R)^{-1}$$
(4.5)

dove R è la matrice (o lo scalare) di covarianza delle misurazioni;

2. calcolo dello stato a posteriori ovvero la nuova stima:

$$x_t = x_{t_-} + K(z_t - Hx_{t_-}) \tag{4.6}$$

3. calcolo della nuova matrice delle covarianze:

$$P_t = (I - KH)P_{t_{-}} (4.7)$$

La (4.4) dice che la varianza dello stato al nuovo istante è legata a quella dello stato precedente più l'incertezza dovuta al modello. Da (4.5) vediamo come più la misura è inaffidabile più ci si affiderà al modello, ovvero

$$\lim_{R \to \infty} K = 0$$

e quindi in (4.6) si ha la ovvia semplificazione, mentre se la varianza del modello aumenta, ci si affiderà alla misura

 $\lim_{P_{t_-}\longrightarrow\infty}K=1$

L'algoritmo del filtro di Kalman è pensato in modo da minimizzare nel modo più efficiente possibile la matrice delle covarianze P, ovvero rendere la stima più possibile vicina al valore vero. Nonostante non sia noto il valore iniziale di questa matrice, in pochi passaggi la stima tenderà a convergere al valore cercato e la matrice P al valore minimo, come mostrato in [14].

4.2 Scelta del modello

Lo scopo di questo capitolo è la programmazione di un algoritmo che consenta di stimare l'angolo di heading di un veicolo autonomo.

Per poter ottenere un risultato che sia soddisfacente sia dal punto di vista economico, sia dal punto di vista della stima della quantità di interesse, la scelta ricade sulla programmazione di una scheda Arduino in grado di elaborare i segnali provenienti da diversi sensori. Nello studio in oggetto vengono utilizzati i sensori presentati in precedenza.

Combinando i segnali provenienti dai diversi sensori è infatti possibile ottenere diversi benefici, riducendo i problemi di ciascuno di essi.

L'accelerometro è stabile sul "lungo periodo", vale a dire che il valore *medio* da questo fornito in output rimane invariato se lasciato in quiete; tuttavia, l'accelerometro è un sensore dotato di un evidente rumore e ciò lo rende inaffidabile nel breve periodo, che si può far corrispondere a qualche campione di misura.

Il giroscopio è molto meno influenzato da rumore, ma è affetto da *deriva*. Con un sensore a basso costo come quello della MPU6050 (ovvero il GY 581) si possono ottenere degli errori sull'angolo di output, ottenuto attraverso dei processi di integrazione nel tempo, anche di decine di gradi al minuto.

Il magnetometro è uno strumento non affetto da deriva significativa, con un rumore molto contenuto ma con un output dipendente dall'orientazione del sensore stesso. Questo infatti misura il campo magnetico terrestre in corrispondenza di alcune facce perpendicolari ai tre assi x, y, z di un riferimento cartesiano destrorso con x parallelo alla direzione di marcia del veicolo e z che punta verso l'alto. Se x e z puntassero in direzioni diverse, magari a causa di un *tilt* del dispositivo, si otterrebbe una misura falsata.

L'idea di fondo di una soluzione che preveda l'utilizzo di questi tre sensori è quindi quella di utilizzare le informazioni provenienti dai primi due per dare una prima stima dell'orientamento **del sensore**, calcolare correttamente il flusso magnetico in un riferimento con z sempre ortogonale alla superficie e poi proseguire con la stima più accurata tramite filtro di Kalman dei tre angoli di **roll**, **pitch** e **yaw**, che per basse dinamiche si può confondere con l'**heading**.

4.2.1 Precisazioni sul significato degli angoli

In questo lavoro, gli angoli di roll e pitch vengono tradotti in italiano con rollio e beccheggio. La definizione è diversa da quella intesa nella dinamica del veicolo, nella quale questi due angoli indicano l'inclinazione della vettura rispetto alla normale uscente dalla strada. Nel caso di questa tesi, si deve pensare a questi due angoli come quelli rilevabili da un *inclinometro*, ovvero l'orientamento in un sistema di riferimento fisso normale alla superficie
terrestre.

Qualora l'auto proceda in piano, le due definizioni coincidono. Tuttavia, non va perso di vista l'obbiettivo di questo studio, ovvero la stima dell'heading angle, che può avvenire correttamente solo se l'orientamento *rispetto ad un riferimento fisso* del sensore è conosciuto.

4.3 Altri lavori

4.3.1 Basi teoriche

In letteratura è possibile trovare diversi lavori che giustificano le scelte di carattere matematico che verranno utilizzate nel corso della trattazione.

Secondo [7] e [6], la compensazione dell'orientamento dell'oggetto è necessaria, qualora si voglia stimare il vero angolo di *heading*, proprio per le motivazioni legate alla costruzione di un compasso magnetico a basso costo quale l'HMC5883L utilizzato per questa sperimentazione.

Questo fa sì che l'angolo di heading dipenda anche dal valore degli altri angoli, con la possibilità che si verifichino dei fenomeni di interferenza.

In [28], [32] e altri viene mostrato come utilizzare dei valori per i rumori di processo e misura coerentemente legati alle prestazioni effettive dei sensori utilizzati, nonché del modello implementato, consenta di ottenere delle prestazioni soddisfacenti del filtro.

4.3.2 Lavori di riferimento

In [26] è presente un algoritmo che consente di stimare l'angolo di inclinazione di una piattaforma conoscendo soltanto la misura del giroscopio sull'asse di interesse e utilizzando come stati del sistema l'angolo di inclinazione e il *bias* del giroscopio. La semplicità del modello, la possibilità di applicarlo separatamente ai tre angoli da stimare e i buoni risultati ottenuti dagli autori lo rendono un valido riferimento per lo studio in oggetto. In particolare, vengono messi a confronto il filtro di Kalman e il filtro complementare. Quest'ultimo, già oggetto di [4], è molto meno oneroso dal punto di vista computazionale ma ha due principali problemi: risente in misura troppo grande delle misure di accelerazione e l'output subisce un *delay* significativo se si utilizza un coefficiente volto a valorizzare le misure dovute al giroscopio.

In [17] viene invece mostrato come dal punto di vista computazionale sia possibile ottenere dei livelli di performance soddisfacenti anche su schede elettroniche indipendenti, ovvero senza la necessità di processare i dati su un computer. Il documento mette in luce il fatto che molte soluzioni che adottano filtri di Kalman utilizzino i computer per l'elaborazione. Adottando però degli accorgimenti, come non variare nel tempo le matrici di covarianza legate al rumore di processo e misura, è possibile ridurre notevolmente l'onere computazionale. Questo ultimo fatto è anche confermato da [28].

4.4 Considerazioni ed ipotesi iniziali

Il filtro di Kalman risulta particolarmente efficace quando lo si utilizza per la stima di grandezze non direttamente osservabili. Ad esempio un modello di dinamica del veicolo può essere utilizzato per misurare l'angolo di deriva degli pneumatici o l'angolo d'assetto partendo da misure fornite da odometri o unità inerziali, che non restituiscono direttamente la grandezza d'interesse.

Nel suo utilizzo più classico, il filtro può essere utilizzato per rimuovere il rumore presente da un segnale di una misura andando a confrontare la misura stessa con il valore *atteso* o stimato della misura stessa.

Questo porta a considerare l'utilizzo del filtro di Kalman come un buon punto di partenza per compensare le problematiche dei tre sensori citati in precedenza.

Data la limitata disponibilità di sensori in questa fase investigativa iniziale, la scelta ricade sui tre sensori interfacciabili con Arduino citati in precedenza. Questo comporta una notevole limitazione sui modelli matematici sfruttabili e sulla complessità del modello stesso.

Dal momento che in questa tesi si cerca di sviluppare un modello che permetta di stimare l'angolo di heading senza considerare una particolare tipologia di veicolo, nel seguito verrà preso in considerazione un modello che sia il più generale possibile.

Per rendere la trattazione più semplice, si introducono anche le seguenti ipotesi:

- il veicolo si muove in piano;
- il veicolo è caratterizzato da una bassa dinamica.

4.5 Angolo di heading e gyro bias

Il modello adottato per la stima degli angoli di roll, pitch e yaw in questa prima soluzione è il seguente¹

$$\begin{cases} \psi_t = \psi_{t-1} + \omega_t \Delta t - \beta_t \Delta t \\ \beta_t = \beta_{t-1} \end{cases}$$
(4.8)

ovvero l'angolo allo step corrente è dato dall'integrazione con il metodo di Eulero meno l'errore dovuto al bias β del giroscopio il quale fornisce il segnale ω_z di velocità angolare.

In forma matriciale si ha:

$$\begin{bmatrix} \psi \\ \beta \end{bmatrix}_{t} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \psi \\ \beta \end{bmatrix}_{t-1} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \omega_{z}$$
(4.9)

Da (4.9) ricaviamo, per confronto di posizioni con (4.3), l'equazione per il time update considerando anche la presenza di un errore di processo w_t

$$x_{t_{-}} = Fx_{t-1} + Bu_t + w_t \tag{4.10}$$

Per quanto riguarda le misurazioni, l'unica quantità osservabile è l'angolo di interesse ψ (ovvero il roll, il pitch o l'heading angle); non essendo il bias osservabile, otteniamo facilmente la matrice H che consente di ottenere le misure z

$$z_t = (1 \quad 0) x_t + v_t = H x_t + v_t$$

Essendo la matrice F una 2x2, anche la matrice delle covarianze P sarà una 2x2 e di conseguenza anche la matrice delle covarianze di processo Q sarà una 2x2.

Dallo step 2. del *measurement update* si può ricavare che K è un vettore 2x1; essendo z_t uno scalare, anche il prodotto Hx_t fornirà uno scalare.

¹Dato che l'obiettivo è il calcolo dell'angolo di heading o yaw, verrà utilizzata la lettera ψ per la scrittura delle formule.

4.5.1 I parametri del modello

Il vettore degli stati $(\psi \ \beta)^T$ in (4.3) e (4.6) contiene l'angolo di interesse e il bias del giroscopio sull'asse corrispondente.

Le misurazioni z in (4.6), per questo modello, sono quelle date dalle elaborazioni dell'accelerometro o magnetometro a seconda che si stimi rollio/beccheggio o imbardata. Le equazioni seguenti hanno validità se le accelerazioni sono molto contenute o il mezzo viaggia a velocità costante. Sono state comunque scelte per la loro semplicità e la possibilità di valutare fattori correttivi nel seguito.

È noto, da [23] e [25], che l'angolo di heading è calcolabile tramite (2.1), mentre gli angoli di roll ϕ e pitch θ sono ricavati dalle misure di accelerazione secondo:

$$\phi = \tan^{-1} \left(\frac{a_y}{\operatorname{sign}(a_z)\sqrt{a_z^2 + \mu a_x^2}} \right)$$
(4.11)

$$\theta = \tan^{-1} \left(\frac{-a_x}{\sqrt{a_z^2 + a_y^2}} \right) \tag{4.12}$$

dove in (4.11) μ è un fattore correttivo per tenere conto dell'indeterminazione che si raggiunge nella formula effettiva, derivata dallo sviluppo degli angoli di Eulero [25], in modo da evitare eventuali indeterminazioni del valore di ϕ

$$\phi = \tan^{-1}\left(\frac{a_y}{a_z}\right)$$

4.6 Modifiche al modello

Il modello precedentemente esposto presenta come unica misura z l'angolo ottenuto dalle (4.11), (4.12) e (2.1). Questo significa che l'input u, che è la misura del giroscopio, viene usata solo relativamente al modello, quando invece è a tutti gli effetti una quantità misurabile del sistema in esame.

Risulta ragionevole investigare cosa avviene se alla singola misura si sostituisce il vettore $z_t = [\psi \ \omega]$, ovvero annoverando tra le misure anche quella del giroscopio.

La (4.6) può essere modificata per tenere conto della presenza della misura del giroscopio come segue:

$$x_t = x_{t_-} + K(z_t - (Hx_{t_-} + Du_t))$$
(4.13)

dove K moltiplica termine tra parentesi a cui viene dato il nome di *innovazione*.

In questo caso è evidente come si modifichino le dimensioni dei vari componenti di (4.13).

Il vettore D è introdotto per correlare l'input, ovvero ω , alla misura z_t . Dal momento che z_t è un vettore 2x1 e x_t allo stesso modo, il prodotto Du_t sarà un vettore 2x1 con componenti [0 1]. Questo causa delle modifiche anche su R, che non sarà più uno scalare ma una matrice diagonale 2x2.

H diverrà allora una 2x2, come pure il guadagno di Kalman K. In particolare,

$$H = \begin{bmatrix} 1 & 0\\ 0 & 0 \end{bmatrix} \tag{4.14}$$

Nel prossimo capitolo verrà illustrato come calcolare in maniera sistematica i valori da inserire nelle matrici Q ed R.

Capitolo 5

I rumori Q ed R

Il problema principale che riguarda l'applicazione del filtro di Kalman è il calcolo delle matrici Q ed R. Se si conoscono le varianze delle misure e degli stati del sistema, sfruttando il modello stesso è possibile calcolarle esattamente, sia nel caso le varianze siano costanti sia che siano variabili nel tempo, come viene suggerito da [12].

Sempre da quest'ultimo e da [17], viene suggerito di eseguire una stima offline di queste grandezze per poi implementare come valori costanti Q ed R nel caso si preveda l'implementazione su microcontrollori come Arduino. Questo è ciò che dal punto di vista computazionale risulta essere più leggero, anche se è possibile che la stima ne possa risentire.

Dai datasheet dei sensori si possono leggere i valori misurati del rumore del sensore e utilizzare questi come primo tentativo, in quanto sono intrinsecamente vicini al valore effettivo del rumore presente nei segnali acquisiti.

Tuttavia, è ben noto che i valori dei datasheet si riferiscono a condizioni di prova¹ e ad impostazioni e di calibrazione di fabbrica. È possibile che il sensore subisca delle degradazioni o perda la calibrazione originale con l'utilizzo o il tempo.

È necessario quindi ricavare un metodo per caratterizzare il rumore presente nei sensori. In particolare, nella prossima sezione viene esposto un metodo particolarmente adatto per sensori come accelerometri e giroscopi, ovvero la misura della *varianza di Allan*.

Ulteriori considerazioni

In generale, si utilizzano matrici Q ed R diagonali. Nelle applicazioni in cui non è possibile risalire ai valori di Q^2 si può considerare che rivestono più importanza i valori dei rapporti tra gli elementi diagonali omologhi delle matrici stesse che le due matrici intese singolarmente. Inoltre, i valori di Q possono essere scelti più o meno elevati a seconda di quanta importanza si vuole dare a una misura oppure all'altra.

5.1 Analisi di Allan

L'analisi di Allan, conosciuta come varianza di Allan o deviazione di Allan, è una metodologia di analisi nel dominio del tempo sviluppata per determinare la stabilità in frequenza degli oscillatori. Concettualmente è possibile l'applicazione allo studio del rumore presente nei sensori per ricavarne le diverse componenti.

¹Sempre indicate insieme al valore indicato.

 $^{^{2}}$ Il rumore dei sensori, e quindi delle quantità misurate, è in generale sempre rilevabile attraverso calibrazioni e caratterizzazioni.

Dati N campioni di un sensore con tempo di campionamento pari a $\frac{1}{f_s}$ con f_s frequenza di campionamento, si può suddividere l'intero insieme di campioni in intervalli chiamati cluster con durate multiple del tempo di campionamento, τ , di durata $n\tau$ con n intero e crescente.

La varianza di Allan si ricava calcolando le medie dei valori presenti all'interno dei cluster in funzione della lunghezza temporale di questi ultimi. Con riferimento alla fig.5.1, il procedimento si esprime con

$$\sigma_A^2(\tau) = \frac{1}{2(K-1)} \sum_{t=1}^{K-1} (\Omega_{t+1} - \Omega_t)^2$$
(5.1)

dove

- K è il numero di cluster presenti nell'intervallo di dati;
- τ è la dimensione del cluster in secondi;
- Ω_t è la media dei valori presenti nell'*i* esimo cluster.

Se si vuole fare riferimento alla deviazione di Allan, parametro più comunemente adottato, si estrae la radice di (5.1).



Figura 5.1: Espressione del procedimento per il calcolo della varianza, da [27].

Tipicamente il diagramma risultante, ovver
o $\sigma_A^2(\tau)$ è rappresentato in scala doppio logaritmica.

Dal diagramma di Allan si possono ricavare diversi parametri. Secondo [32], in un modello come quello adottato in questo lavoro è utile ricavare i due valori dell'**Angle Random Walk** e del **Rate Random Walk**.

Angle Random Walk

L'Angle Random Walk è il processo col quale un giroscopio produce dei segnali che integrati producono un valore di spostamento angolare anche se il sensore viene lasciato in quiete. In sostanza, questo parametro è legato al drift presente nel giroscopio e dunque rappresenta la componente aleatoria dell'errore dovuta al rumore bianco del sensore.



Figura 5.2: Tratto del diagramma di Allan che identifica l'ARW, da [27].

Il tratto dove si può rilevare questo parametro è quello mostrato in fig. 5.2. In particolare, il parametro N è quello letto in ordinata per $\tau = 1$. La varianza ad esso associata si calcola, data la frequenza di campionamento

$$\sigma_w^2 = \frac{N^2}{\Delta T} \tag{5.2}$$

Rate Random Walk

Questo fenomeno è dovuto a processi di origine incerta ed è rilevabile con buona accuratezza solo con misurazioni di lunga durata. Nella curva di Allan si può identificare in un tratto a pendenza positiva per tempi τ elevati.



Figura 5.3: Tratto del diagramma di Allan che identifica l'RRW, da [27].

Il valore del coefficiente Kassociato è rilevabile in corrispondenza di $\tau=3$ e permette di definire la varianza del rumore di bias come

$$\sigma_b^2 = \frac{K^2}{\Delta T} \tag{5.3}$$

5.2 Misura della varianza di Allan del giroscopio

Per calcolare i parametri necessari è stata eseguita un'analisi della durata di un'ora ed acquisiti i segnali del giroscopio sui tre assi. In questo modo si è ricavato il diagramma seguente



Figura 5.4: Diagramma di Allan del giroscopio.

Da questo è possibile ricavare i valori dei due parametri di interesse e le varianze ad esso associati tramite le (5.2) e (5.3), listati in tab.5.1

Asse	N $\left(\frac{\circ/s}{\sqrt{Hz}}\right)$	K $\left(\frac{\circ/s}{\sqrt{Hz}}\right)$	$\sigma_A^2 \ (\circ/s)^2$	$\sigma_b^2 \ (\circ/s)^2$
g_x	0.02292	0.00035	0.0244	$5.698 \ 10^-6$
g_y	0.03387	0.00018	0.0534	$1.570 \ 10^-6$
g_z	0.02325	0.00009	0.0251	$3.761 \ 10^-6$

Tabella 5.1: Risultati della analisi di Allan.

Sempre secondo [32] e [28] è possibile utilizzare il valore della varianza, ottenuto dopo un'acquisizione sufficientemente lunga come per la varianza di Allan, per gli elementi della matrice R. Per le misurazioni degli angoli sono stati ricavati i rumori associati ai valori degli angoli (4.11), (4.12) e (2.1) e del giroscopio, di cui si possono vedere i valori in tab. 5.2, ricavati elaborando lo stesso set di dati utilizzato per il calcolo dei valori di 5.1.

Asse	ϕ (o)	θ (o)	ψ (o)	$g_x \ (\circ/s)$	$g_y \ (\circ/s)$	$g_z \ (\circ/s)$
$\sigma^2 \; (\circ/s)^2$	0.0335	0.0408	0.6393	0.0029	0.0014	0.0023

Tabella 5.2: Risultati della stima della varianza. Gli angoli sono dati dalle misure dirette di accelerometro e magnetometro.

5.3 Simulazioni Matlab

L'analisi di Allan è stata eseguita per trovare il rumore da associare al drift, legato all'ARW, e al bias, legato al RRW. In questo modo è possibile dare un fondamento alla scelta dei valori della matrice Q.

Per poter testare la bontà dei modelli si passa allo sviluppo di uno script in Matlab.

Questo script prende in ingresso un set di dati e, basandosi sulle sole informazioni ricavate dal giroscopio e dagli angoli calcolati con le accelerazioni e magnetometro, eseguono i passaggi da (4.3) a (4.7) per stimare gli angoli ψ , $\theta \in \phi$ con cui in questo studio sono chiamati roll, pitch e yaw risultanti dal filtro.

Il segnale di spostamento e di velocità di rotazione viene costruito in Matlab e simula uno spostamento angolare di 90° e ritorno alla posizione iniziale.

5.3.1 Riempimento delle matrici

Dal momento che l'algoritmo stima separatamente i tre angoli sono state definite tre matrici, con pedice corrispondente all'asse di riferimento. La forma generica è quella indicata da [32], ovvero

$$Q = \begin{bmatrix} \sigma_A^2 & 0\\ 0 & \sigma_b^2 \end{bmatrix}$$
(5.4)

Le matrici diventano allora

$$Q_x = \begin{bmatrix} 0.00244 & 0\\ 0 & 5.98 & 10^-6 \end{bmatrix}$$
(5.5)

$$Q_y = \begin{bmatrix} 0.0534 & 0\\ 0 & 1.57 & 10^-6 \end{bmatrix}$$
(5.6)

$$Q_z = \begin{bmatrix} 0.0251 & 0\\ 0 & 3.761 & 10^{-6} \end{bmatrix}$$
(5.7)

Dal momento che il modello è lo stesso nei due casi, le matrici Q sopra definite sono applicate anche nel secondo filtro.

Nel secondo filtro, le matrici R vengono costruite coi valori di 5.2. Il valore di R utilizzato nel primo filtro è solo il primo elemento della diagonale delle matrici successive. La forma generica è

$$R = \begin{bmatrix} \sigma_{ang}^2 & 0\\ 0 & \sigma_{gyro}^2 \end{bmatrix}$$
(5.8)

dunque

$$R_x = \begin{bmatrix} 0.0335 & 0\\ 0 & 0.0029 \end{bmatrix}$$
(5.9)

$$R_y = \begin{bmatrix} 0.0408 & 0\\ 0 & 0.0014 \end{bmatrix}$$
(5.10)

$$R_z = \begin{bmatrix} 0.6393 & 0\\ 0 & 0.0023 \end{bmatrix}$$
(5.11)

5.3.2 Esecuzione delle simulazioni

Per poter creare un set di dati misurati è stata acquisita una storia temporale dalla scheda Arduino, hardware dove verrà implementato il codice finale, e costruita una legge di spostamento e velocità in Matlab, mostrata in fig. 5.5, su detta base tempi. Questa è stata poi corrotta con un rumore bianco con varianza corrispondente a quella contenuta nelle (5.9), (5.10) e (5.11).



Figura 5.5: Simulazione di uno spostamento del sensore.

La scelta di corrompere con rumore bianco la legge di spostamento deriva dall'ipotesi sul rumore del filtro di Kalman illustrata in precedenza.

Nel seguito vengono mostrati i risultati della simulazione del filtro sui tre assi del giroscopio, con i parametri delle tabb. 5.1 e 5.2.

Per tutti e tre i casi si vede come il filtro riesca a ridurre il rumore presente nella misura in maniera considerevole, fino a meno di mezzo grado. La scelta dei parametri è quindi coerente con le ipotesi adottate; tuttavia è possibile che sia necessario controllare l'efficacia dei parametri inseriti nelle matrici Q ed R poiché l'ipotesi di rumore bianco con distribuzione gaussiana non è in grado di rappresentare tutte le sorgenti di rumore presenti nel segnale reale.



Figura 5.6: Simulazione del filtro su asse x.



Figura 5.7: Simulazione del filtro su asse y.



Figura 5.8: Simulazione del filtro su asse z.

Per poter eseguire un confronto tra i modelli con le equazioni per la misura (4.6) e (4.13) si utilizza la stessa legge di spostamento simulata del primo caso, in modo che il rumore sia lo stesso.



Figura 5.9: Simulazione del filtro su asse x.



Figura 5.10: Simulazione del filtro su asse y.



Figura 5.11: Simulazione del filtro su asse z.

Dalle figg. 5.9, 5.10 e 5.11 si può vedere come, col sensore in quiete, i risultati siano gli stessi. Tuttavia, l'inserimento della matrice D garantisce una maggiore prontezza del

filtro durante le variazioni.

In definitiva, se la dinamica del sistema è molto contenuta, la modifica non comporta significativi vantaggi al costo di un maggior onere computazionale.

Nel sistema reale, la presenza di accelerazioni o di spostamenti improvvisi del sensore può però essere rilevata più prontamente.

Tenendo conto del fatto che, come mostrato in (2.2), la stima dell'angolo di heading è influenzata dalla stima dell'angolo di roll e pitch, si deve valutare questo fattore attentamente con prove sperimentali.

Nelle prove su cui verrà utilizzato il filtro di Kalman si ipotizza di restare sempre sul piano, dunque i valori di questi angoli restano sempre prossimi allo zero: in questo modo, la stima dell'heading non dovrebbe risentire in maniera marcata delle loro variazioni.

Capitolo 6

Implementazione del filtro su scheda elettronica Arduino

Dopo aver verificato il funzionamento del modello in Matlab, è necessario effettuare le opportune modifiche per poter implementare l'algoritmo del filtro di Kalman su scheda Arduino.

Il passaggio non è immediato in quanto nel Wiring, il linguaggio basato su C con cui si scrivono i programmi per Arduino, non implementa nativamente il calcolo matriciale. Sebbene esistano librerie apposite, per alleggerire l'onere computazionale richiesto alla scheda è consigliato scrivere delle funzioni manualmente che svolgano i calcoli matriciali.

Utilizzando il Symbolic Math Toolbox sono state scritte le operazioni matriciali delle operazioni da (4.3) a (4.7). Il concetto di base è che le operazioni come somma e moltiplicazione già esplicitate riducono il numero di operazioni richiesto al processore Atmel della scheda. Inoltre, si ricorda che l'assegnazione non comporta onere computazionale aggiuntivo. Inoltre, l'inclusione in un progetto Arduino di troppe librerie può ridurre lo spazio disponibile per altre operazioni.

Un'altra considerazione da fare è legata alla struttura portante dei programmi Arduino, che si divide in due parti chiamate setup e loop. Queste sono due funzioni che devono sempre essere presenti; la prima viene eseguita solo all'avvio della scheda, mentre la seconda viene eseguita ripetutamente fino allo spegnimento della scheda stessa.

L'ultima considerazione da fare è che in Arduino non è possibile eseguire assegnazioni a vettori e matrici in maniera immediata come in Matlab. Questo modifica il modo di operare su questi elementi.

6.1 Impostazione del progetto Arduino

Ogni progetto Arduino comprende una finestra principale, in cui si impostano tutte le variabili globali, le librerie da utilizzare ed eventuali oggetti appartenenti a classi specifiche, si eseguono le operazioni di **setup** e si impostano le operazioni da eseguire ciclicamente nel **loop**.

Non è necessario che tutte le operazioni siano definite nel loop; queste possono essere definite con altre funzioni scritte fuori dal loop ed invocate nell'ordine richiesto.

Per questo progetto sono state create quattro schede:

• una finestra principale, dove sono state definite tutte le variabili globali, utilizzabili cioè da tutte le funzioni presenti nelle varie schede, le librerie utilizzate e le funzioni

setup e loop. È la finestra che dà il nome al progetto Arduino;

- una finestra dedicata all'implementazione del filtro di Kalman, dove sono state definite anche le operazioni matriciali con funzioni apposite;
- una finestra per l'elaborazione dei dati ricevuti dai sensori, ovvero dedicata al calcolo degli angoli come in (4.11), (4.12) e (2.1), al calcolo dei valori calibrati del magnetometro e ulteriori compensazioni;
- una finestra per la scrittura delle impostazioni di base dei vari sensori e delle inizializzazioni delle variabili.

Le finestre principali, di elaborazione e impostazione sono uguali (ma non comuni) per i due programmi; la differenza sostanziale sta nella finestra dedicata al filtro di Kalman.



Figura 6.1: Schermata principale del programma Arduino.

6.2 Collegamenti da effettuare

I collegamenti da eseguire tra scheda Arduino e GY 87 sono elencati in tab. 6.1. Per poter leggere i dati che la scheda Arduino processa è sufficiente collegare la scheda stessa ad una porta USB del pc tramite un cavo.

Pin Arduino	Pin GY 87		
5 V	VCC		
GND	GND		
SDA	SDA		
SCL	SCL		

Tabella 6.1: Collegamenti Arduino-IMU.

6.3 Acquisire i dati

Per acquisire i dati su un Arduino esistono diversi modi. I più utilizzati sono l'utilizzo di un modulo SD, come mostrato in [4], o di un modulo Wi-Fi, che invia direttamente i dati sul computer. Entrambi presentano svantaggi notevoli. La scrittura su scheda SD rallenta notevolmente l'esecuzione dei programmi Arduino. Per molte applicazioni non è necessaria una velocità di esecuzione elevata, cosa invece richiesta da questa applicazione. L'utilizzo di un modulo Wi-Fi è legato alla disponibilità di una connessione Internet.

Il monitor seriale di Arduino (accessibile tramite la combianzione Ctrl+Shift+M) non consente di salvare le informazioni che Arduino scrive sulla linea seriale del cavo USB. Tuttavia esistono monitor seriali terzi che dispongono di molte più funzionalità del monitor di Arduino.

In questo lavoro è stato utilizzato *PuTTY*. Senza entrare nel dettaglio delle sue funzioni di programmazione, che esulano dagli scopi di questo lavoro, questo applicativo incorpora un emulatore di terminale, attraverso questo, di leggere e scrivere su diversi tipi di rete, tra cui le linee seriali delle porte COM del PC. Per gli scopi di queste prove, è stata utilizzata la sua funzione di *logging* della sessione di terminale, ovvero è stato possibile scrivere su file i dati elaborati da Arduino e inviati da questo sulla porta COM corrispondente del PC.

Questo metodo ha consentito un notevole risparmio in termini di programmazione Arduino, *payload* del codice stesso e utilizzo di componenti.



(a) Schermata di avvio di PuTTY.

Figura 6.2: Le schede Arduino.

Per effettuare il log di una sessione, basta cliccare su $Logging~({\rm fig.~6.2a})$ ed impostare dove salvare il file.

6.4 Valutazione delle prestazioni del filtro di Kalman

Per valutare le prestazioni del filtro di Kalman nelle sue due varianti presentate al capitolo precedente, sono state eseguite alcune prove per la verifica dei parametri impostati nelle matrici da (5.5) a (5.11).

Per entrambi gli algoritmi sono state svolte le seguenti prove per valutare la prontezza della stima *dell'angolo di heading*:

- 1. rotazione del sensore in piano con sequenza $0^{\circ}/90^{\circ}/0^{\circ}/-90^{\circ}/0^{\circ};$
- 2. rotazione del sensore in piano con sequenza $0^{\circ}/180^{\circ}/0^{\circ}/-180^{\circ}/0^{\circ};$
- 3. cambio veloce di orientamento in piano.

Nel seguito sono mostrati i risultati di dette prove.

6.4.1 Modello con misura $z = \psi_{mag}$



Figura 6.3: Prova 1.



Figura 6.4: Prova 2.



Figura 6.5: Prova 3.

Da queste figure si vede come il rumore venga correttamente attenuato e, come visto pure nelle simulazioni, la stima subisca un fenomeno di rallentamento. Questo fenomeno sembra presentarsi ogni qual volta la velocità di rotazione si azzera.

6.4.2 Modello con misura $z = [\psi_{mag} \ \omega_z]$











Figura 6.8: Prova 3.

La modifica nell'equazione della misura sembra risolvere il problema evidenziato nelle figg. 6.3, 6.4 e 6.5. Anche con una velocità di rotazione più elevata, la stima si avvicina alla misura in maniera più marcata.



Figura 6.9: Visualizzazione del fenomeno di attenuazione.

6.5 Conclusioni

In questo capitolo è stato sviluppato un filtro di Kalman basato su un modello cinematico in modo da poter adattare la stima dell'angolo di heading ad un qualsiasi veicolo, considerando quest'ultimo un unico corpo rigido.

L'analisi di Allan per la determinazione dei parametri delle matrici Q ed R si è dimostrata un valido strumento per la caratterizzazione dei rumori presenti sia nelle misure, sia nel modello. È importante ricordare che è stato possibile utilizzare questo metodo dato il particolare modello adottato; in generale valgono le considerazioni espresse in *Ulteriori* considerazioni nel caso dell'utilizzo di modelli più complessi.

La necessità primaria del modello sviluppato è il calcolo dell'angolo di heading. La stima degli altri angoli poteva essere portata avanti utilizzando un metodo diverso, ma la possibilità di adattare l'algoritmo ad un qualsiasi angolo ha permesso di riutilizzare gli strumenti che già necessitavano di sviluppo.

Si ricorda che gli angoli di rollio e beccheggio vengono calcolati con lo scopo di compensare l'orientamento rispetto ad un riferimento fisso locale del sensore e dunque del veicolo.

La modifica del vettore z delle misure ha portato un beneficio in termini di prontezza del filtro stesso, dunque nel seguito si farà riferimento solo a questa variante. Dal punto di vista computazionale viene richiesto alla scheda un maggior impiego di memoria. L'utilizzo di una scheda Arduino Mega è dunque necessario, in quanto la scheda Uno R3 verrebbe sicuramente saturata se introdotte ulteriori modifiche.

La possibilità di utilizzare una matrice R diagonale, i cui elementi sono le varianze delle misure di angolo, dato da accelerometro e/o magnetometro, e di velocità angolare, permette di fare ulteriori considerazioni, le quali verranno approfondite nel prossimo capitolo.

Capitolo 7

Modifiche al filtro

Il modello sviluppato nel capitolo precedente ha dato prova di buone capacità di filtraggio della grandezza di interesse in caso stazionario o dinamica molto contenuta. Questo è anche in accordo alle ipotesi adottate, coerenti con i primi test da eseguire con un veicolo autonomo - accelerazione e frenata in rettilineo, curve a velocità ridotta, raggiro di un ostacolo.

Tuttavia, sebbene si possano prendere tutti gli accorgimenti per eseguire le prove in un ambiente il più possibile rispettoso delle ipotesi introdotte, si può notare che ci sono degli altri aspetti da tenere in considerazione quando il sensore per la stima dell'heading verrà montato su un veicolo.

7.1 Ipotesi adottata per la gestione dei disturbi

In letteratura è possibile trovare diversi esempi che mostrano come sia possibile adattare alcuni dei parametri del filtro di Kalman per migliorare la stima in presenza di particolari condizioni che il modello non è in grado di gestire analiticamente.

I procedimenti che consentono di raggiungere i migliori risultati comprendono metodi che lavorano su modelli d'errore, come quelli mostrati in [1], [21] e [16]. Tuttavia, la complessità di questi modelli introduce matrici di dimensioni notevoli che poco si accordano con la necessità di questo lavoro di programmare una scheda Arduino per il processamento dei dati e l'esecuzione del filtro.

Per non dover ricorrere a modelli più complessi che, data la sensoristica disponibile durante lo svolgimento di questo lavoro, non sarebbero di facile applicazione, si fa d'ora in avanti l'ipotesi che i vari disturbi che vengono incontrati durante l'esecuzione del filtro vengano considerati al pari di rumori aggiuntivi. In questo modo sarà possibile investigare la possibilità di ottenere dei risultati soddisfacenti con un costo computazionale ridotto. In letteratura è possibile trovare diversi esempi, come quelli mostrati in [32], [8] e [3], che consentono di giustificare l'ipotesi introdotta.

7.2 Compensazione degli effetti dinamici

7.2.1 Accelerazioni agenti sul sistema

Da (4.11) e (4.12) si intuisce come gli angoli di roll e pitch verranno stimati correttamente solo se l'unica forza effettivamente agente sul sensore è la gravità. Non appena interverrà una forza aggiuntiva, che sia una forza di inerzia in frenata e accelerazione o un trasferimento di carico, questi eventi influenzeranno in maniera significativa la stima. Senza prendere gli opportuni accorgimenti, il modello potrebbe stimare erroneamente il valore degli angoli e, ricordando che valgono le (2.2), si correrebbe il rischio di ottenere una stima falsata dell'angolo di heading.

Anche se si riuscisse a condurre il veicolo in condizioni quasi stazionarie, ovvero accelerando, frenando e curvando con una frequenza talmente bassa da praticamente non agire sul sistema, non sarebbe comunque possibile eliminare completamente le cause di stima errata. Le irregolarità della strada possono influenzare allo stesso modo la stima di roll e pitch.

È noto infatti che, anche considerando un modello molto semplice di veicolo mono-sospensione, esiste una funzione di trasferimento tra terreno e massa sospesa, alla quale il sensore è solidale, esprimibile nel dominio della frequenza con

$$F(s) = \frac{s^2 z}{h} \tag{7.1}$$

dove con hsi intende il profilo stradale
e $s^2 z$ l'accelerazione trasmessa da detto profilo alla massa.

Questa accelerazione, che per semplicità si può pensare diretta unicamente sulla direzione verticale del mezzo, entrerà a tutti gli effetti nelle (4.11) e (4.12) modificando il valore effettivo degli angoli misurati e quindi la stima.

Come enunciato in precedenza, la stima dell'orientamento del sensore nello spazio può avvenire solo se si trascurano, o meglio compensano, gli effetti dinamici.

Un modo di procedere può essere rappresentato dalla caratterizzazione del profilo stradale o attraverso una curva di densità di potenza oppure caratterizzando attraverso diverse prove sperimentali il rumore indotto sul veicolo dalla strada di prova.

Entrambi i metodi richiedono molto tempo ed onere computazionale per la determinazione e l'implementazione nel modello sviluppato. Inoltre, il modello è stato selezionato proprio per la sua adattabilità a diversi tipi di veicolo: un approccio volto a caratterizzare il percorso di prova comporterebbe per ogni campagna di acquisizione dati la caratterizzazione del tracciato, andando ad inficiare sulla generalità ricercata.

Tra i diversi metodi, risulta particolarmente adatto all'applicazione sull'algoritmo sviluppato in questo lavoro il correttore proposto in [32].

7.2.2 Metodo per escludere le accelerazioni dovute alla dinamica

Come esposto all'inizio di questa sezione, se non si ha dinamica è possibile risalire agli angoli di roll e pitch tramite la misura delle accelerazioni. Data la natura del filtro di Kalman e la possibilità di trattare separatamente il rumore dovuto alla misura dell'angolo e della velocità angolare, si propone il seguente metodo *threshold-based*¹.

Per poter verificare la presenza di una dinamica è necessario confrontare la situazione istante per istante con la situazione che si avrebbe se sul sensore agisse solo la gravità. Ciò può essere fatto costruendo la seguente quantità scalare

$$\alpha = ||\vec{a}_0 - \vec{g}|| \tag{7.2}$$

ovvero confrontando il valore delle accelerazioni misurate nel riferimento inerziale con il valore del vettore gravità. Se il sensore è stazionario o quasi stazionario, α sarà un valore molto contenuto; se invece $\alpha \gg 0$, si può adattare la matrice R come mostrato di seguito

¹Basato su un valore di soglia

$$R_{comp} = \begin{bmatrix} \sigma_{ang}^2 + k\alpha^2 & 0\\ 0 & \sigma_{qyro}^2 \end{bmatrix}$$
(7.3)

dove k è un parametro moltiplicativo che può essere calibrato a seconda della sensibilità richiesta dall'applicazione.

Per scegliere il valore di soglia che deve attivare la compensazione si può partire dall'ipotesi introdotta ad inizio capitolo. Dal momento che si vuole distinguere il rumore dovuto alla misura dal rumore introdotto dai disturbi, appare naturale porre

$$\alpha_{threshold} = \sigma_{accel}$$

dove σ_{accel} è la deviazione standard ricavabile dalle misure rilevate in fase di calibrazione.

Per poter confrontare le accelerazioni del sensore con la gravità occorre riportare le prime nel sistema di riferimento fisso tramite la trasformazione indicata in [4]

$$\vec{a}_0 = A_i^0 \vec{a}_i \tag{7.4}$$

dove

$$A_{i}^{0} = \begin{bmatrix} \cos\theta\cos\psi & -\cos\theta\sin\psi & \sin\theta\\ \cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi - \sin\phi\sin\theta\sin\psi & -\cos\theta\sin\phi\\ \sin\phi\sin\psi - \cos\phi\sin\theta\cos\psi & \sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix} (7.5)$$

L'algoritmo adattativo con soglia sarà quindi esprimibile tramite

$$\begin{cases} \text{if} \quad \alpha < \sigma_{accel} & R \ ok \\ \text{else} \quad R(1,1) = R(1,1) + k\alpha^2 \end{cases}$$
(7.6)

Sebbene in questo modo la determinazione della condizione di moto dipenda dalla stima degli angoli di interesse, nelle prove di applicazione pratica in cui l'algoritmo verrà testato ci si aspetta che roll e pitch non superino le decine di gradi. In questo modo si possono ridurre eventuali effetti di correlazione e loop, rendendo valida l'applicazione di (7.5).

7.2.3 Verifica e calibrazione dei parametri

L'implementazione in Arduino di questa funzione è particolarmente semplice.

Di seguito sono mostrati dei grafici di prove eseguite su un piano per valutare l'efficacia di (7.3). Le prove sono state eseguite in piano muovendo il sensore su una guida in alluminio in moto alternato, mantenendone l'orientamento circa costante.

Dai confronti delle figg. 7.1 e 7.2 e 7.3 e 7.4 si vede come la modifica sia particolarmente efficace. Si va a ridurre anche l'influenza sull'angolo di heading, ricordando che valgono le (2.1), migliorando ancora la qualità della stima². Nel seguito, con *stima* si intende l'output del filtro di Kalman, mentre con *misura* il valore di angolo ricavato da (4.11) o da (4.12).

²Eventuali variazioni del valore sono da additare alla imprecisione nell'esecuzione della prova



Figura 7.1: Roll non compensato.



Figura 7.2: Roll compensato.



Figura 7.3: Pitch non compensato.



Figura 7.4: Pitch compensato.

Per valutare se il metodo adattativo funziona anche come filtraggio delle asperità del terreno, è stato allestito un veicolo del dipartimento del DIMEAS con una barra di alluminio posta sul tettuccio. Si è effettuata la calibrazione del sensore nel luogo di installazione e rilevata la correttezza della procedura tramite la visualizzazione in MagViewer.



(a) Veicolo.

(b) Montaggio IMU.

Figura 7.5: Veicolo per le prove preliminari.



(a) Posizionamento IMU.

(b) Setup per l'acquisizione.



La grande quantità di materiale ferromagnetico presente nelle vicinanze non può permettere di formulare giudizi assoluti sulla stima dell'angolo di heading, nonostante il sensore sia stato calibrato sul tragitto da percorrere, ma permetterà di valutare il filtraggio sugli altri due angoli e la successiva influenza sulla stima dell'heading. Il tracciato di prova è mostrato in fig. 7.9 e la prova consiste nella percorrenza di un breve rettilineo, purtroppo non esente da ostacoli.



Figura 7.7: Prova di movimento 1.



Figura 7.8: Prova di movimento 2.



Figura 7.9: Percorso di prova.

Dalle prove eseguite si nota come la variazione di roll e pitch angle sia contenuta ma non sia avvenuta una completa attenuazione; il tracciato su cui la prova è stata eseguita, mostrato in 7.9, presenta un manto non uniforme e non perfettamente piano. È verosimile che le piccole variazioni di angolo siano dovute all'effettiva inclinazione del veicolo in alcuni momenti. Considerazioni analoghe possono essere fatte sull'heading angle, in quanto sul tracciato è consentito il transito di pedoni avvenuto anche durante la prova.

Si conclude che l'algoritmo adattativo su ${\cal R}$ porta ai risultati attesi.

7.3 Influenza dei disturbi magnetici sulla misura di heading

Nei modelli proposti per la stima dell'angolo di heading si nota come vi sia una forte dipendenza dalla misura del campo magnetico. Tutte le prove statiche hanno mostrato come, in seguito ad una corretta calibrazione nel luogo dove le prove sono state effettuate, la stima dell'angolo di interesse avviene correttamente.

La presenza di materiali ferrosi o campi magnetici dovuti a motori elettrici ed altri dispositivi elettronici possono però influenzare, anche fortemente, la performance dello stimatore al punto che si possono generare errori anche di decine di gradi, a seconda dell'intensità del campo magnetico disturbatore.

Una corretta calibrazione del magnetometro fa sì che ogni lettura rispetti la seguente

$$(H - H_{off})(H - H_{off})^T = B^2$$
(7.7)

ovvero che la norma del vettore delle misurazioni del campo magnetico H, corretto dell'offset dovuto agli effetti hard-iron H_{off} , sia pari al campo magnetico B effettivamente agente sul sensore. In [24] è dimostrato questo risultato.



(a) Componenti campo magnetico indisturbato. (b) Distorsioni nelle componenti magnetiche.

Figura 7.10: Effetti delle distorsioni, da [20].

Tenendo conto di questo fatto, è possibile pensare che la presenza di disturbi, generati da elementi ferromagnetici, distorca la stima della sfera risultato della corretta calibrazione e, in definitiva, il valore del campo magnetico rilevato dal sensore.

In letteratura esistono diverse testimonianze di metodi per la compensazione dei disturbi. In [1] viene indicato come, in un campo magnetico indisturbato, il valore dell'intensità misurata è pari a quella del campo magnetico terrestre e l'angolo di inclinazione o dipsia costante e pari al valore fornito dagli osservatori globali [13] [5]. Inoltre, se il campo magnetico è indisturbato, si ottiene la relazione ||m|| = 1, ovvero la normalizzazione delle tre componenti restituisce un vettore di lunghezza unitaria.

La conoscenza del valore del campo magnetico terrestre introduce delle complessità non facilmente superabili nell'implementazione su una scheda elettronica come Arduino, prevista per lavorare in modalità *standalone*.

Altri autori [3] propongono diverse soluzioni per compensare la presenza di campi con metodi *threshold based* o *model based*. Come già esposto in precedenza, su un processore come quello di una scheda elettronica embedded è consigliato lavorare con il minor numero possibile di variabili.

7.3.1 Implementazione di un metodo threshold-based

Nell'implementazione di un metodo *threshold-based* efficace è di grande importanza il valore che viene utilizzato per lo scarto delle misurazioni disturbate dalla presenza di campi magnetici temporanei. Il valore deve essere infatti impostato sufficientemente alto in modo da non scartare le naturali fluttuazioni della misura del campo dovute al rumore presente nel sensore e alla conformazione dell'ambiente circostante, ma basso a sufficienza da non includere le distorsioni.

In [31] viene indicato che la distorsione del campo magnetico indisturbato può raggiungere anche valori di centinaia di milligauss quando il sensore viene utilizzato in ambienti con materiali ferrosi. Viene quindi implementato un metodo con valore di soglia con rimozione delle misure magnetiche quando la differenza tra il campo misurato nel sistema di riferimento del sensore e quello misurato nel riferimento fisso terrestre supera il valore di soglia.

Nel magnetometro utilizzato per il lavoro di questa tesi non è prevista la misurazione contemporanea in due sistemi di riferimento diversi. Tuttavia, è in qualche modo possibile scartare le misure che si discostano troppo da un valore di riferimento.

In [8] viene utilizzato un metodo basato sulla probabilità che la covarianza associata alla nuova misurazione sia in linea con la covarianza che viene predetta allo stato precedente. Questa predizione viene poi utilizzata in una funzione densità di probabilità che è il selettore per i diversi filtri di Kalman implementati contemporaneamente³.

Dato questo background teorico, nel seguito verrà sviluppato un metodo volto alla correzione della varianza di misura dell'angolo di heading, il quale è dei tre il più affetto da problemi di disturbi magnetici.

7.4 Compensazione degli effetti dovuti a disturbi magnetici

Nel seguito verrà implementato un metodo che permette di avere un adattamento continuo del valore della varianza σ_{ang}^2 e un valore di soglia per scartare le misure affette da disturbi troppo elevati.

Per applicare questo metodo si fa l'ipotesi che le prove verranno eseguite più volte sullo stesso percorso e nello stesso ambiente, cosicché le distorsioni introducano sempre lo stesso disturbo negli stessi punti. La calibrazione deve essere eseguita per ognuno dei diversi ambienti in cui vengono eseguite le prove, in modo che (7.7) sia valida.

Per poter eseguire un controllo sulla bontà della misurazione *i*-esima, si è scelta una forma gaussiana in quanto consente una variazione continua del valore di R(1,1)

$$f(m_i) = \frac{k}{\sigma_{mag}\sqrt{2\pi}} \exp\left(-\frac{(m_i - m_{mean})^2}{2\sigma_{mag}^2}\right)$$
(7.8)

dove k è un parametro di calibrazione che riduce la velocità di tendenza a 0 della funzione densità di probabilità, σ_{mag} è la deviazione standard delle misure di intensità di campo magnetico, m_i la misura *i*-esima e m_{mean} la media delle misurazioni.



Figura 7.11: Esempio di funzione densità di probabilità o curva gaussiana.

In questo modo si può affermare che se la misura *i*-esima è una misura che ha una probabilità molto bassa di accadere, la varianza delle misure magnetiche verrà impostata ad un valore molto basso tramite la seguente

$$R(1,1) = \frac{R(1,1)}{f(m_i)}$$

Se invece la misura si discosta poco dal valore medio, $f(m_i) \to 1$ e la varianza rimarrà invariata.

³Viene utilizzata una Multimodel Adaptive Estimation o **MMAE**.

Il problema di questo metodo è il calcolo del valore medio da usare nella (7.8). È necessario scartare le misure che eccedono un certo valore di soglia per non ottenere una media falsata. In [31] viene evidenziato come possono considerarsi forti disturbi magnetici quelli che modificano l'intensità del campo rilevato di 10 μT ovvero 100 mG. Inoltre, impostare una media fissa potrebbe accentuare i problemi legati alla rumorosità del sensore e far scartare misure che invece possono essere affidabili.

Per poter aggiornare la media scartando i valori che superano la soglia in maniera efficiente, si utilizza la seguente formula ricorsiva, dove con c vengono indicate le misure effettuate

$$m_i^{mean} = \frac{(c-1)m_{i-1}^{mean} + m_i}{c}$$
(7.9)

ovvero la media al passo i si ottiene dalla media precedente pesandola con le misure effettuate più il nuovo valore. L'algoritmo si può allora sviluppare impostando un valore di soglia come in [31], col funzionamento illustrato di seguito.

- Se $|m_i m_{mean}| < th$ -c = c + 1 $-m_i^{mean} = \frac{(c-1)m_{i-1}^{mean} + m_i}{c}$ $-diff = diff + (m_i - m_{mean})^2$
- Se c == maxCount

$$- dev = \sqrt{\frac{difj}{c}}$$
$$- diff = 0$$
$$- c = 0$$

dove è stata usata la notazione mostrata in tab. 7.1. In questo modo si riescono ad accettare le misure che rientrano nella rumorosità del magnetometro, scartando i disturbi.

7.4.1 Verifica dell'algoritmo

Per verificare che l'algoritmo proposto sia efficace vengono eseguite diverse prove.

In un primo set di prove viene lasciato il sensore in quiete sia internamente sia esternamente al veicolo mostrato in fig. 7.5a. In fig. 7.12a viene mostrato come, indifferentemente dal luogo in cui questo viene posizionato, l'intervallo di oscillazione massima rimanga sempre entro lo stesso limite. In fig. 7.12b viene invece illustrato come nonostante sia differente il valore dell'intensità del campo magnetico rilevato, la rumorosità sia paragonabile. Ciò consente di affermare che le ipotesi di base del metodo proposto sono valide.

Vengono poi eseguite due prove per valutare l'efficacia dell'adattamento del valore di R. Nella prima viene lasciato il sensore in quiete ed avvicinato un oggetto ferromagnetico nelle direzioni dei tre assi. Nella seconda si gira il sensore in piano con regolarità e si avvicina un oggetto ferromagnetico.

Dalla prova statica di fig. 7.13 si vede che i disturbi magnetici più elevati vengono filtrati in maniera soddisfacente, riducendo i gradi di errore a circa 2-3. Si evidenzia nella stessa prova un problema nell'algoritmo di aggiornamento della media, in particolare a 4, 9 e 11 secondi.

Dalla prova di fig. 7.14 invece è possibile notare che i disturbi di intensità minore vengono solo parzialmente corretti, mentre vengono ignorati i disturbi di intensità maggiore.

Si può quindi dedurre che il problema principale di questo metodo è il corretto settaggio del valore di soglia th.

Simbolo	Significato
m_i	Misura del campo magnetico corrente
m_{mean}	Media del campo magnetico
th	Valore di soglia
c	Variabile contatore, numero intero
diff	Differenza tra valore corrente e media al quadrato
dev	deviazione standard aggiornata dopo c misure

Tabella 7.1: Notazione per algoritmo.





(b) Confronto tra i valori rilevati.

Figura 7.12: Prove statiche.



Figura 7.13: Analisi dell'influenza del disturbo.



Figura 7.14: Analisi dell'influenza del disturbo in rotazione.

7.5 Conclusioni e precisazioni

In questo capitolo sono stati sviluppati due metodi per la gestione dei disturbi indesiderati nei segnali che possono inficiare le prestazioni del filtro di Kalman sviluppato nel capitolo

precedente.

Dalle prove eseguite è possibile dedurre che la strada intrapresa è quella corretta nel caso della gestione delle misure accelerometriche, portando a risultati soddisfacenti. Avendo eseguito le prove in piano ci si aspettava che gli angoli $\phi \in \theta$ fossero prossimi a zero in ogni condizione e così è stato riscontrato.

Maggiore difficoltà riveste l'interpretazione delle prove legate alla compensazione dei disturbi magnetici. Affidandosi ai dati trovati in letteratura è stato possibile sviluppare un algoritmo che fornisce dei risultati soddisfacenti solo in maniera relativa. Inoltre, il *tuning* dei parametri riveste un'importanza fondamentale per la riuscita del filtraggio. In questo secondo caso sarà quindi necessario approfondire e valutare ulteriormente in sede di prove sperimentali effettuate con un sensore di riferimento come la Ekinox.

La scelta della funzione gaussiana è stata presa cercando uno strumento matematico che consentisse di avere una variazione piuttosto graduale della varianza da associare alle misure magnetiche. La necessità di non scartare misure che rientrino nella rumorosità del sensore ha portato alla definizione di media mobile. Si è notato dalle prove eseguite come sia necessaria un'attenta calibrazione del valore di th e come la convergenza arrivi dopo alcune iterazioni soltanto (fig.7.15).



Figura 7.15: Visualizzazione della media mobile rispetto all'intensità rilevata della prova di fig.7.14. Si evidenzia come il valore di soglia debba essere attentamente tarato.

Dal momento che il manto stradale su cui le prove sono state eseguite è diverso da quello usuale, per le successive prove sperimentali si utilizzeranno le stesse impostazioni per poter effettuare un confronto più diretto.

Capitolo 8

Descrizione del controllore

Il PXI è un controllore programmabile per sistemi di automazione e misura.

È una piattaforma a prestazioni elevate utilizzata laddove sia necessario testare componenti e sistemi in ambito automotive, aeronautico ed industriale, come processi di produzione.

Un sistema PXI include almeno questi quattro componenti: chassis, controller, moduli e software.

Lo *chassis* è il componente principale del sistema PXI e fornisce l'alimentazione, il raffreddamento e i bus di comunicazione tra il controller e i moduli e tra i componenti del controller stesso.

Il *controller* integrato ad elevate prestazioni consente di lavorare con sistemi operativi Windows o real-time come NI LabVIEW.

I *moduli* consentono la customizzazione del sistema PXI e ne esistono di diverso tipi: schede I/O multifunzione, schede CAN, schede seriali, ecc.

Il software offerto da NI è **VeriStand**, una interfaccia avanzata e user friendly basata su LabVIEW, attraverso la quale si possono programmare i vari canali di comunicazione con le schede, preparare delle interfacce grafiche per la visualizzazione delle grandezze di interesse e eseguire algoritmi per elaborare i dati.

8.1 I componenti utilizzati

8.1.1 NI PXI 1031

Lo chassis utilizzato nelle prove è lo NI PXI 1031 che presenta quattro slot. Dà alimentazione a tutto il sistema e fornisce i collegamenti necessari ai moduli e al controller.


Figura 8.1: Chassis.

8.1.2 NI PXI 8110

 $\grave{\mathbf{E}}$ un controller a prestazioni avanzate per l'utilizzo su sistemi NI. Negli chassis è sempre montato nello slot più a sinistra.

I controller integrati consentono di ottenere un sistema completo e relativamente compatto contenuto all'interno del PXI, senza il bisogno di un PC per l'elaborazione dei dati. In generale, i controller integrati sono dotati di caratteristiche standard come una CPU, hard disk, RAM, Ethernet, attraverso la quale possono comunicare con il PC host, e possibilità di collegare periferiche audio e video.



Figura 8.2: Controllore.

8.1.3 NI PXI 6123

La scheda *multifunction NI PXI 6123* è una scheda dotata di 8 input analogici, 8 pin digitali I/O, trigger, due porte di counter e 8 interfacce programmabili denominate PFI. Viene utilizzata per collegare diversi dispositivi analogici o digitali al PXI come encoder, potenziometri e altri strumenti. È programmabile attraverso VeriStand.



Figura 8.3: Scheda multifunction.

8.1.4 NI PXI 8513

La *NI PXI 8513* è una scheda dotata di una porta CAN. Utilizzando NI-XNET driver è possibile creare applicazioni che richiedono l'elaborazione di frame e segnali CAN ad alta velocità. È possibile importare, modificare e utilizzare i segnali di file DBC chiamati *database*. I segnali che viaggiano su rete CAN sono visualizzabili sul PC host con latenza ridotta grazie ad un sistema privo di interrupt della CPU del PXI che garantisce uno stream a tempo costante.



Figura 8.4: Scheda CAN.

Capitolo 9

La rete CAN

Il *Controller Area Network* è un protocollo seriale utilizzato principalmente in ambiente automotive, introdotto negli anni Ottanta dalla Bosch per snellire i collegamenti delle parti elettroniche delle auto.

Nella rete CAN sono necessari solo due cavi per tutto il sistema, come si può vedere da fig.9.1: si ha dunque un notevole risparmio di peso e costi.



Figura 9.1: Resa grafica architettura CAN, da [9].

A differenza della comunicazione seriale, dove si distinguono nodi *master* e *slave*, quando un nodo CAN è pronto a trasmettere i dati, verifica solo se il bus è occupato o meno; poi il *frame* viene semplicemente inviato sulla rete. Ogni *frame* è etichettato da un ID unico, che tutti i nodi possono leggere e, in base a questo, decidere se accettarlo. Una volta che il messaggio viene inviato sulla rete, rimane sempre disponibile: ciò vuol dire che le informazioni sono sempre accessibili da tutti i nodi simultaneamente.

Nell'ambito di questa tesi, verrà utilizzato un singolo dispositivo da interfacciare con la scheda CAN presente sul PXI; non si presenteranno dunque problemi di questa natura. Potrebbe essere necessario tenere conto di questo fatto in fasi avanzate del progetto, quando potrebbe presentarsi la necessità di collegarsi alla rete CAN del veicolo da sensorizzare.

9.1 Architettura

Il bus CAN è composto da due cavi, CAN High e CAN Low, e la differenza di potenziale tra questi due canali determina il segnale di output. Ogni estremità presenta una resistenza pari a 60 Ω che consente il funzionamento del protocollo. La tensione dei due cavi può variare a seconda dell'applicazione, ma normalmente si assume High intorno ai 3.5 V e Low intorno a 1.5 V.



Figura 9.2: Tensione sul bus CAN, da [9].

Un cavo CAN utilizza un connettore standard DB9, ovvero lo stesso impiegato per le comunicazioni seriali, ma sfrutta solo 3 pin, ovvero il 2, il 3 e il 7 collegati rispettivamente a Low, GND e High.

La scheda impiegata in laboratorio è la **NI PXI 8513** che monta una singola porta CAN e due porte per il *triggering* e il *clocking* esterno dei segnali da scrivere sul controllore. Queste ultime non verranno però utilizzate.

9.2 Database CAN

Il database CAN è un file che contiene la descrizione del protocollo e la struttura dei messaggi. Un messaggio del database si struttura nel modo seguente:

- nome del segnale;
- bit iniziale e numero di bit per ciascun segnale;
- ordine dei byte;
- tipo di dato: intero con segno o senza segno, a virgola mobile;
- scala ed unità di misura;
- range di valori permessi;
- valore di default;
- commento.

Queste informazioni vengono fornite in modo che i byte possano essere facilmente convertiti in dati con un significato fisico e più comprensibile. Per poter interfacciare la unità Ekinox con il PXI è stato molto conveniente l'utilizzo di un *database* fornito da SBG Systems. L'unità inerziale è infatti in grado di fornire moltissimi segnali in uscita e la costruzione di un database sarebbe stata molto lunga e complessa. Sulla costruzione esistono già altri lavori di tesi [9] i quali sono molto utili alla comprensione della struttura dei segnali.

Il software utilizzato per la visualizzazione del database è *NI XNET*. Nella figura 9.3 è possibile vedere la grande quantità di segnali forniti dalla Ekinox e come segnali di uno stesso tipo siano raggruppati in uno stesso frame. Per ogni frame è indicato il numero di *byte* che lo compongono, il nome, l'ID, il *timing* (ovvero se il dato viene fornito ogni volta che è disponibile una nuova informazione, ed in questo caso si chiama *Event data*, o con altre modalità, ad esempio *Cyclic data*) e il tipo di input/output.



Figura 9.3: Visualizzazione di un *frame* in XNET.

In fig. 9.4 viene invece mostrato un singolo segnale, con le informazioni di cui esso è costituito. In particolare distinguiamo:

- nome;
- tipo di segnale (singolo o instradato in un *multiplexer*);
- fattore di scala, offset, massimo, minimo, default ed unità di misura: queste informazioni hanno a che fare con la natura fisica del segnale;

• bit di partenza, numero di bit, tipo di segno e ordine dei bit; quest'ultimo indica se il bit più significativo è il primo, ed in tal caso è detto *Big Endian*, o l'ultimo, ovvero *Little Endian*.



Figura 9.4: Visualizzazione di un segnale in XNET.



(a) Un frame completo.

	7	6	5	4	3	2	1	0
0	7	6	5	4	3	2	1	0
1	15	14	13	12	11	10	9	8
2	7	6	5	4	3	2	1	0
3	15	14	13	12	11	10	9	8
4	7	6	5	4	3	2	1	0
5	15	14	13	12	11	10	9	8
▼ 1								

(b) I bit corrispondenti ad un singolo segnale.

Con le figg. 9.5a e 9.5b si mette in evidenza come un dato numerico viene rappresentato da un numero a 8 bit in cui un bit è riservato per il segno e gli altri 7 per il valore numerico.

In un *frame* CAN, le informazioni sopra riportate costituiscono soltanto il *cuore* del messaggio, parte della struttura più estesa che le incorpora. Il formato base è quello mostrato in fig. 9.6.



Figura 9.6: Messaggio CAN completo, da [9].

Nel messaggio completo si possono individuare diversi oggetti, la cui descrizione si può trovare in [9]. La parte contenente il dato numerico di interesse, ovvero la misura effettuata o il comando inviato, è la parte associata a *Data*.

Capitolo 10

I software National Instruments

10.1 NI Measurement and Automation eXplorer

Il software *NI MAX*, ovvero *Measurement and Automation eXplorer*, è lo strumento che consente di gestire i componenti National Instruments. È utile in fase di configurazione dei vari target e per controllare la presenza dei software necessari al corretto funzionamento dei programmi sui controllori.



Figura 10.1: Schermata principale di NI MAX.

Nell'albero è possibile visualizzare tutti i software presenti nel PC host utilizzabili con sistemi National Instruments, espandendo la voce *Software* sotto *My System*. Cliccando col destro si possono disinstallare o installare plug-in e driver e collegarsi al *NI Driver Manager* per il download di componenti aggiuntivi.

Sotto la voce *Remote Systems* è possibile configurare il PXI o i diversi controllori da utilizzare. È inoltre possibile verificare il corretto funzionamento delle schede collegate al PXI attraverso i menu dedicati. Al primo avvio, NI MAX scansiona tutti i sistemi di una rete per individuare la presenza o meno dei diversi target. Questi sono collegati al PC host

attraverso dei cavi *crossover* Ethernet. Il protocollo utilizzato è l'IPv4 e l'assegnazione dell'IP di host e target è automatica.

My System	🖬 Save 🚷	Refresh					y? Hide He
> mar Devices and interfaces ✓ Software						Back	
LabVIEW Runtime 2014 SP1	Settin	gs				How do Tw	
 LavVEW Runtime 2015 SP1 f10 LavVEW Runtime 2017 SP1 f3 LavVEW Runtime 2017 SP1 f3 LavVEW Runtime 2015 SP1 f3 LavVEW Runtime 2018 SP1 f3 N PXP Inform Services Configuration 19.1 N PXP Inform Services Runtime 19.5 Remote Systems P PA-10310C Chassis Inform Services Configuration PXP Information PXP	Settin Name Vendo Model Status Slot D Slot 1 2 3 4 2 5	etails Model NI PXI-8110 NI PXI-7842R NI PXI-7842R NI PXI-6123 how empty slots	Chass Natio PXI-1 Prese PXI15Iot1 PXI15Iot2 CAN1 DAQ1	is 1 nal Instruments 031DC nt Serial Number 01485834 01D9E46F 01D9D3FE	Vendor National Instruments National Instruments National Instruments National Instruments	How do I us PHOME ADDA STATES To anable the full fit your PKI system, as properly identified, is assigned a chass is devices it contained Matter and the full model, view module chassing and the full chassing contained the formation of the full of the full of the full system of the full of the full of the full of the full of the full of the full of the full of the full of the full of the full of the full of the full of the full of the full of the full of the full of the full of	se my unctionality of the hassis as such classis is number dowith sis and dowith ssis? ur PX chassis as with the he chassis. If applicable): isager linea-

Figura 10.2: Schermata principale di NI MAX.

Cliccando col tasto destro su *Software* è possibile scaricare i driver presenti sul PC anche sul target.

10.1.1 Aggiungere una scheda DAQ

Per aggiungere una scheda DAQ è sufficiente che questa sia inserita nel PXI al momento dell'avvio di NI MAX in modo che possa essere riconosciuta automaticamente. Per poter interfacciarsi con essa per scopi di debugging o programmazione è però necessario che sul PC host siano presenti i driver corrispondenti alla scheda utilizzata.

ne					
			mplitude vs. Samples Chart	Auto-scale chart	7
			,0068 -		
			,0066 -		
		w.	0004		
uration			,0004 -		
		w.	,0062 -		
imit	Min Input Limit		0,006 -		
10		-10	0050		
	Samples To Rei	ad	,0050		
1000		1000	,0056-		
			,0054 -		
			,0052 -		
			0,005-		gk.
				5,8m	ï
	init 10	Internation Intern	aration mit 10 Samples To Read 1000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	00060 - 0	

Figura 10.3: Schermata per la verifica di segnali analogici.

Una volta che la scheda è stata riconosciuta si possono eseguire diverse procedure, come la calibrazione automatica, la programmazione di task specifiche e il controllo del

funzionamento dei pin.

Quest'ultima è una operazione di debug molto efficace in quanto consente di testare le connessioni ai dispositivi terzi sia in input sia in output. Si possono verificare diversi tipi di porte, come quelle analogiche (fig. 10.3) o digitali (fig. 10.4), cliccando sul pulsante in alto *Test panels*.

1. Select Port	2. Select Direction	
port0	port0/ine0:7	portb/line0:7 Input (1) O O O O O O O O O O O O O O O O O O O
		port0 Direction 11111111 7 0
	3. Select State	
	Port/line State	High (1) Low (0) 7 0 All High
		port0 State 11111111 7 0
		Start Stop

Figura 10.4: Schermata per la verifica dei segnali digitali.

Se necessario si può prendere visione delle uscite della scheda tramite il pulsante $Device\ pinouts.$

10.1.2 Aggiungere una scheda CAN

Per aggiungere una scheda CAN è sufficiente che questa sia inserita nel PXI al momento dell'avvio di NI MAX in modo che possa essere riconosciuta automaticamente. Per poter interfacciarsi con essa per scopi di debugging o programmazione è però necessario che sul PC host siano presenti i driver corrispondenti alla scheda utilizzata.

Measurer	nent Setting	s Help							
•	1) 📑 /	M X (?					
Monitor	Statistics Trans	nit			Bus Lo	ad:			
ID	Time Stamp	Length	Data	Туре	Dir	Rate [Hz]	No of Fram	e Frame Name	e /
1									

Figura 10.5: Schermata per la verifica di ricezione e/o invio dei messaggi CAN.

Una volta che la scheda è stata riconosciuta si possono eseguire diverse procedure, come la calibrazione e il test di comunicazione. Questo è particolarmente importante per verificare che la connessione del dispositivo CAN sia avvenuta correttamente.

In questa schermata verranno visualizzati i *frame* grezzi, ovvero sotto forma di numeri esadecimali, poiché in NI MAX non è presente il database per la conversione dei segnali.

10.2 NI VeriStand

10.2.1 Introduzione

La programmazione del controllore PXI viene eseguita tramite il software già precedentemente citato *NI VeriStand*. Questo è uno dei tool più utilizzati per la programmazione di sistemi *real-time*. Un sistema operativo *real-time*, come quello del PXI, assicura che i processi avvengano in un tempo predeterminato e non casuale. Per la particolare applicazione da implementare, ovvero la guida autonoma, non è sicuro utilizzare un sistema operativo Windows in quanto questo non è un sistema *real-time* e i tempi dei processi dipendono dal livello di priorità delle operazioni richieste.

😻 Getting Started Window	- 🗆 X
File Tools Add-Ons Help	
₩NI VeriStand 2	017
🚵 New NI VeriStand Project	News
Most Recent Projects	Customize NI VeriStand
🔊 data_can_ekinox	Motion Control with NI VeriStand
📓 data_can_imu	Tengine Simulation
🙀 potenziometro_lin_SC	Hardware Fault Insertion
👪 Engine Demo	©⊃ Read All Blogs
🙀 prova_pc	Help
👪 Sinewave Delay	😵 What's New in NI VeriStand
Co Browse	 Getting Started with NI VeriStand Configuring and Running a Project
a biolise	NI VeriStand Video Tutorials
🍕 Configure Project 🛛 🚯 Run Project	Extending NI VeriStand with LabVIEW
	Licensed for Full Development System

Figura 10.6: Schermata introduttiva di VeriStand.

La macchina su cui è installato il software VeriStand e sulla quale avviene la programmazione viene indicata col termine di **host**. Viene indicato col termine **target** il controllore su cui viene caricato il programma. Questo gestisce i componenti fisici per diversi tipi di prove e consente anche la simulazione di componenti reali; può quindi essere utilizzato per prove di prototipazione, *Hardware-in-the-loop*, *Model-in-the-loop* e *Softwarein-the-loop*.

10.2.2 L'ambiente VeriStand e definizione di un progetto

Dalla schermata di fig. 10.6 si può cliccare su $File \implies New Project$ per iniziare un nuovo progetto oppure selezionare uno dei *Most recent projects*. Cliccando su *New Project* si aprirà una schermata, mostrata in fig. 10.7, in cui è possibile definire diversi parametri del progetto, tra cui il nome, la cartella dove salvare i progressi, il nome del System Definition File (o se utilizzarne uno già esistente) e della User Interface (o se usarne una già esistente).

Per progetti in cui si utilizza un unico controllore PXI^1 è consigliabile lasciare tutte le impostazioni di default.

 $^{^1\}dot{\rm E}$ possibile creare sistemi in cui si sfrutta una rete di controllori PXI.

roject System Definition UI Manager Properties		Project	System Definition	UI Manager	Properties	
Project Name		Create	e a new NI VeriStand	System Definitio	on File	
Untitled I		· ✓ Us	e project name			
Project Root Folder		Syste	m Definition File nam	ne		
C:\Users\Public\Documents\National Instruments\NI VeriSt 2017\Projects	and 🖉	O Use an	n existing NI VeriStan	d System Definit	tion File	
✓ Create folder for project Project Path		Syster	n Dermition File Path			
C:\Users\Public\Documents\National Instruments\NI VeriSt Projects\Untitled 1\Untitled 1.nivsproj	tand 2017\					
OK Cancel	Help			ОК	Cancel	Help
			(1) NT C		0	T 1 1 1
(a) Nome progetto.	×	😻 Create N	(b) Nome S	System D	efinition	i File.
(a) Nome progetto. Create New Project roject System Definition UI Manager Properties	×	Create N	(b) Nome S New Project System Definition	UI Manager	efinition Properties	i File.
(a) Nome progetto. Treate New Project oject System Definition UI Manager Properties O Create a new UI Manager project	×	Create N Project Creator	(b) Nome S New Project System Definition	UI Manager	efinition Properties	r File.
(a) Nome progetto.	×	Create N Project Creator \$123456	(b) Nome S New Project System Definition	UI Manager	efinition Properties	ı F'ıle.
(a) Nome progetto. Create New Project roject System Definition UI Manager Properties ○ Create a new UI Manager project ☑ Use project name UI Manager Project name UI Manager Project name	×	Create N Project Creator \$123456 Descript	(b) Nome S New Project System Definition	UI Manager	Properties	ı File.
(a) Nome progetto. Create New Project roject System Definition UI Manager Properties Create a new UI Manager project ✓ Use project name UI Manager Project name Untitled 1) Use an existing UI Manager project file UI Manager File Path	×	Vroject Project Creator \$123456 Descript Progette	(b) Nome 2 New Project System Definition	UI Manager	Properties	ı f'ile.
(a) Nome progetto. Create New Project moject System Definition UI Manager Properties Create a new UI Manager project UI Manager Project name UI Manager Project name UI Manager Project name UI Manager File Path Do not add a UI Manager project		Project Creator s123456 Descript Progette	(b) Nome S New Project System Definition	UI Manager	Properties	ı f'ile.

Figura 10.7: Schermata di nuovo progetto.

Una volta creato il nuovo progetto, si entrerà nella *Project Explorer Window*, mostrata in fig. 10.8. Da questa è possibile gestire tutti gli aspetti di un progetto VeriStand. Nella guida disponibile in laboratorio vengono illustrati tutti i campi selezionabili [22], mentre in questo lavoro verranno approfonditi solo quelli utili alla creazione del progetto.

Configurare il progetto

Per iniziare la configurazione del progetto occorre cliccare su System Definition File (fig. 10.9). Questo è il tool che consente di definire ogni aspetto del progetto stesso: il controller su cui verrà caricato il programma, le schede da utilizzare², i database per determinati protocolli di comunicazione, fattori di scala, modelli o programmi di terze parti, ecc. Nel seguito verrà mostrato un progetto di esempio per la lettura di un potenziometro lineare.

 $^{^{2}}$ È possibile che sul PXI siano montate più schede di quelle utilizzate. Per migliorare la performance è quindi opportuno indicare solo quali si intende sfruttare.

Figura 10.8: IDE VeriStand.

System Explorer - potenziometro_lin_SC.nivssdf	- 0	×
🏂 🗃 🖬 🗴 🖻 🖺 X A 🔳	Δ	
Poletacionetto (in Se Trays Chassis DAQ DAQ DAA Traing and Sync Gast Traing and Sync Gast Traing and Sync Gast Success Success Success Success Success Simulation Model System Channels Soles Sole Soles Sole Sole	System Explorer Window Use this window to create and/or modify a system definition file. You configure a system definition file by adding removing, and modifying options in the configuration tree, located on the left of the System Explorer window, using the System Explorer window, respective to the configuration tree, located on the left of the System Explorer window, using the System Explorer window, refer to the Configuration tree, substant channels, and some VeriStand Engine execution satirings. For more information about using the System Explorer window, refer to the Configuring and Running a Project book of the IV VeriStand Heip, variable by setting the System Explorer window, refer to the Configuring and Running a Project book of the IV VeriStand Heip, variable by setting the System Explorer window, refer to the Configuring and Running a Project book of the IV VeriStand Heip. Refer to the Components of a Project book for detailed descriptions of system definition files and the VeriStand File. Name Detensionation [in_SC Creation Creation Description	
	Version Major Minor Fix Build 1 0 0 0 23 Revision	
	Revision history	^
	Reset Add	

Figura 10.9: System Definition File.

Il primo oggetto da definire è il controller. Per poterlo definire, si clicca su *Controller*. I campi più importanti da modificare sono *Operating System* e *IP Address*, con i valori mostrati in fig. 10.10. *PharLap* è il sistema operativo *real-time* utilizzato dal PXI, mentre l'IP è quello che è possibile definire in NI MAX, come visto precedentemente. È consigliabile lasciare i parametri di *Processor Assignment, Target Decimation* e *Other Settings* ai valori predefiniti.

File Edit Tools Help		
ն 🗃 🖬 🗶 🖻 🖺 🗙 🗛 🚍 🥼	K, Hardware Discovery Wizard	
potenziometro_lin_SC SC STargets	Target Specification	
	Name PXI	Operating System IP Address PharLap V 169.254.18.49
E 💋 DAQ	Processor Assignments	
CANAIOG Input	Primary Control Loop mode Processor Automatic V -2 \$	Data Processing Loop mode Processor Automatic -2 +
2 Data Sharing ■ FPGA ■ ₩ NI-XINET ▲ 10 Timing and Sync	Target Decimations Data Processing Loop	
SLSC Custom Devices Simulation Models	Other Settings	
fx Calculated Channels fx Calculated Channels fx→ Stimulus	Maximum streamed channels Execution 512 Image: Construction of the stream of the stre	Filter DAQ Errors
Alarms Procedures XNET Databases	FPGA / Scan interface mode DAQ DIO Rate Use current 100 🖤	Warmup Time Hz 2 sec
A* Aliases	Timing Source Settings	
A Scales System Mappings WW Data Sharing Network	Primary Control Loop timing source Automatic Timing	
3 System Initialization	Target Rate Ti 100 🗣 Hz	ming Source Timeout

Figura 10.10: System Definition File: impostazione controller.

Definire le schede

Se si utilizza una scheda di acquisizione **DAQ**, come in questo caso, si può scegliere la frequenza di lavoro della porta digitale, al campo *DAQ DIO Rate*. Questa non deve essere necessariamente uguale al *Target Rate*, poco più in basso. Quest'ultimo indica la frequenza di lavoro del PXI. Questo valore va scelto con attenzione in quanto determina la velocità di esecuzione del programma.

Il secondo passaggio prevede la definizione delle schede da utilizzare. Per questo progetto verrà utilizzata la scheda *Multifunction* mostrata in precedenza. Si clicca col destro su $DAQ \implies ADD DAQ Device$ e si impostano i vari campi come mostrato in fig. 10.11. Come anche indicato nella guida ufficiale [22], è consigliabile definire solo le porte effettivamente necessarie. Il nome della scheda deve essere esattamente quello definito in NI MAX altrimenti la scheda non verrà riconosciuta. In figg. 10.12a e 10.12b viene mostrato come impostare opportunamente la lettura del potenziometro. Una volta che la scheda è correttamente definita, apparirà nell'albero del *System Definition File*. In *input configuration* (fig. 10.11) è possibile lasciare *default*. La scheda disponibile in laboratorio lavora con impostazione *Differential*, o differenziale, ovvero il sistema di misura risponde alla differenza di potenziale presente tra i pin (+) e (-). Altri metodi di misura sono mostrati in [9].

😻 Add DAQ Device	×
Device name	
DAQ1	
Description	
	^
	¥
Select DAQ device PXI-6123	\sim
Select input configuration Default	\sim
OK Cancel Help	b

Figura 10.11: Definizione scheda di acquisizione.

💟 Add DAQ Channels		X M Add DAQ Channels	- 🗆 ×
Select channel type to add Al Select measurement type Voltage	Sample mode Single-Point Initial value Units V	Select Channels to be added Select Channels to be added C Analog Input C A All	
Property	Value Unit		
Minimum Value	0 V		
Maximum Value	5,000000 V	AI5	
Input Configuration	Same as device	✓ ☐ ~ Al6	
		~~~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	
Help	Back Finish Next Cancel	Help Back Finish N	Vext Cancel

(a) Definizione tipo di porta.

(b) Scelta delle porte.





Figura 10.13: Scheda di acquisizione definita.

Se si deve utilizzare anche un dispositivo con protocollo CAN si può procedere come mostrato di seguito. Per questo progetto d'esempio non è necessario, ma sarà utile successivamente.

Per configurare una scheda CAN si deve espandere nell'albero la voce **NI-XNET** e cliccare col destro sulla voce CAN per selezionare ADD CAN port. Si apre un menu come quello di fig. 10.14 in cui si devono specificare correttamente il numero della porta CAN alla quale vogliamo collegare il dispositivo, il database da utilizzare e il *baud rate*. Il nome può essere qualsiasi.

ANET CAN port r	name	
prova		
XNET CAN port a	ddress	
CAN 2	$\sim$	
XNET database		
sbg2		~
CAN cluster		
Cluster		~
Baud rate [Bd]		
500000	<b>.</b>	

Figura 10.14: Definizione scheda CAN.

Il baud rate è una unità di misura della velocità di trasmissione. Si differenzia dai bit per secondo poiché ad un simbolo del baud rate *possono corrispondere più bit*. Nel caso della rete CAN, che è una trasmissione binaria, questi due numeri coincidono.

Un fatto a cui bisogna prestare molta attenzione è quello mostrato in fig. 10.15. Verificare che sotto l'opzione *Termination* sia impostato **On**, altrimenti la comunicazione CAN non potrà avvenire correttamente.

General	Port	Transmission Order	Automatic Frame Processing	986x Support	
Termina	tion				
On		$\sim$			
Echo	transmi	it?			
In put stu		ad time [e]			
input sti	ream rea	Input stre	eam queue size		
v, .		Automa	uc v		
Chec	k for co	mmunication state and	restart automatically		
Polling	rate (ms	]			
1	*				
Transce	iver Typ	e			
HS	$\sim$				

Figura 10.15: Ulteriori accorgimenti nella definizione della CAN.

Infine, per determinare quali informazioni si vogliono estrarre dalla rete CAN, occorre espandere *Incoming* dal menu della porta appena creata e cliccare col destro per selezionare *Import Frames...* Nella schermata che si apre, potrebbe essere necessario spuntare l'opzione *Reverse Incoming/Outgoing frames for ECU* per poter visualizzare i frame disponibili. In fig. 10.16 viene mostrato come selezionare i frame da importare. I frame selezionati compariranno sotto la voce *Incoming*.

NET Frames Import Options General Option	5
(NET database	Cluster
sbg2	Cluster
ECU	
Ekinox	Reverse Incoming/Outgoing frames
Search pattern	101 200
· · · · · · · · · · · · · · · · · · ·	Update while typing
KNET frames	
🖃 🗹 Cluster	
🖃 🗹 Ekinox	
SBG_ECAN_LOG_IMU_GYRO (290)	
GYRO_X	
GYRO_Y	
GYRO_Z	
SBG_ECAN_LOG_IMU_ACCEL (289)	
ACCEL_X	
ACCEL_Y	
ACCEL_Z	
<b>7</b>	
Show invalid frames	

Figura 10.16: Selezione dei frame del cluster.

**Nota:** Con *CAN Cluster* si indicano i pacchetti di dati inviabili sulla rete CAN. Con *ECU* si indica quali pacchetti sono dei *Transmitted frames* (l'equivalente dei *data frames* illustrati nella sezione sulla rete CAN) o dei *Received frames* (ovvero dei *remote frames*). Normalmente si ha la necessità di leggere i dati, quindi tra i *Received frames* non compariranno segnali.

#### Impostare canali personalizzati

I Calculated channels sono degli strumenti che consentono di compiere operazioni matematiche sui segnali acquisiti tramite le schede DAQ e/o CAN. In fig. 10.18 viene mostrato come il valore della tensione del potenziometro può essere normalizzato rispetto al valore massimo del canale definito in fig. 10.12a, selezionando il canale da modificare e l'operazione da eseguire. È possibile eseguire molte altre operazioni, come leggere il massimo o minimo valore, applicare un filtro passa basso, eseguire la media su n punti, ecc. come mostrato in fig. 10.17.

Maximum Minimum Lowpass Filter Peak & Valley Acceleration	Search
Average Conditional	ller

Figura 10.17: Operazioni disponibili.

lame					Units				
lormalized tension					V				
escription									
culated Channel Sett	inas								
ormula									
Var0]/5							^	11	0
							U		1
	Search 📔 🕰 🔳								
4 <b>m</b> 11-		1		(	)	7	8	9	/
4 🖼 (	dware Chassis								
4	🕫 DAQ	exp	sqrt	sin	asin	4	5	6	*
	A 🖊 DAQ1							_	
	Analog Inp	sign	rand	COS	acos	1	2	3	+
Þ 🏼 🖉 Simi	ulation Models	abr	int	tan	atan	0		~	_
<	>	aus	inc	Lan	atan	•	1		-
Variable Name	Channel								
Variable Name	/Hardware/Chas	sis/DAO/DA	O1/Ana	10	Variable	Name			
					Var0				
					Channe	1			
				×	~Q1/	/Analog	Input/Al	0	
<				>		-			
unnary									

Figura 10.18: Esempio di Calculated channel.

#### Importare un modello Simulink

In VeriStand è possibile importare dei modelli sviluppati in Matlab/Simulink opportunamente compilati. Questo consente di estendere enormemente le potenzialità di calcolo del PXI.

La procedura per configurare il proprio Matlab/Simulink per la compilazione è illustrata sul sito della National Instruments. In breve, è necessario assicurarsi che sul PC siano presenti i pacchetti Microsoft Visual C++ 2010 Redistributable e il Windows SDK 7.1, col quale è possibile installare il compilatore C++ che viene utilizzato da Matlab. Normalmente, in un computer sono già presenti i pacchetti Redistributable, quindi è necessario seguire i passaggi successivi per poter compilare modelli Simulink.

- 1. scaricare il Windows SDK 7.1 ed avviare l'installazione, ignorando il messaggio di errore ("Some components cannot be installed because Microsoft .NET Framework 4 is not installed" ecc.);
- 2. selezionare come in fig. 10.19 (spuntare anche Visual C++ 2010 Redistributable nel caso non siano installati già sul computer);
- 3. scaricare il file VC-Compiler-XXXX.exe³ ed avviare l'installazione;
- 4. avviare Matlab ed eseguire il comando mex -v -setup e automaticamente verrà impostato Windows SDK 7.1 come compilatore predefinito come in fig. 10.20. Nel caso ne siano presenti altri, la procedura più rapida per selezionare in Matlab il compilatore è disinstallare tutti quelli presenti prima di questo processo in modo che quello dell'SDK 7.1 sia l'unico presente.



Figura 10.19: Installazione SDK.

**Nota:** Questi passaggi possono essere eseguiti per diverse versioni di VeriStand e Matlab. Sul supporto della National Instruments vengono indicate le compatibilità tra i due programmi. Nel caso di questa tesi, l'ultima versione disponibile di Matlab compatibile con la VeriStand 2017 è la 2015a.

³https://www.microsoft.com/en-us/download/details.aspx?id=4422

```
... Looking for registry setting 'HKCU\SOFTWARE\Microsoft\VisualStudio\SxS\VS7' 11.0 ... No.
... Looking for registry setting 'HKLM\SOFTWARE\Wow6432Node\Microsoft\VisualStudio\SxS\VS7' 11.0 ...No.
... Looking for registry setting 'HKCU\SOFTWARE\Wow6432Node\Microsoft\VisualStudio\SxS\VS7' 11.0 ...No.
Did not find installed compiler 'Microsoft Visual C++ 2012 (C)'.
... Looking for compiler 'Microsoft Visual C++ 2013 Professional (C)' ...
... Looking for registry setting 'HKLM\SOFTWARE\Microsoft\VisualStudio\SxS\VS7' 12.0 ...No.
... Looking for registry setting 'HKCU\SOFTWARE\Microsoft\VisualStudio\SxS\VS7' 12.0 ...No.
... Looking for registry setting 'HKLM\SOFTWARE\Wow6432Node\Microsoft\VisualStudio\SxS\VS7' 12.0 ...No.
... Looking for registry setting 'HKCU\SOFTWARE\Wow6432Node\Microsoft\VisualStudio\SxS\VS7' 12.0 ...No.
Did not find installed compiler 'Microsoft Visual C++ 2013 Professional (C)'.
... Looking for compiler 'Microsoft Windows SDK 7.1 (C)' ...
... Looking for registry setting 'HKLM\SOFTWARE\Microsoft\Microsoft SDKs\Windows\v7.1' InstallationFold
... Looking for registry setting 'HKLM\SOFTWARE\Wow6432Node\Microsoft\VisualStudio\SxS\VC7' 10.0 ...Yes
... Looking for file 'c:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\bin\amd64\cl.exe' ...Yes.
... Looking for folder 'c:\Program Files (x86)\Microsoft Visual Studio 10.0' ...Yes.
Found installed compiler 'Microsoft Windows SDK 7.1 (C)'.
... Looking for compiler 'Microsoft Windows SDK 7.1 (C)' ..
... Looking for registry setting 'HKLM\SOFTWARE\Microsoft\Microsoft SDKs\Windows\v7.1' InstallationFold
... Looking for registry setting 'HKLM\SOFTWARE\Wow6432Node\Microsoft\VisualStudio\SxS\VC7' 10.0 ...Yes
... Looking for file 'c:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\bin\amd64\cl.exe' ...Yes.
... Looking for folder 'c:\Program Files (x86)\Microsoft Visual Studio 10.0' ... Yes.
Found installed compiler 'Microsoft Windows SDK 7.1 (C)'.
MEX configured to use 'Microsoft Windows SDK 7.1 (C)' for C language compilation.
Warning: The MATLAB C and Fortran API has changed to support MATLAB
     variables with more than 2^32-1 elements. In the near future
     you will be required to update your code to utilize the
     new API. You can find more information about this at:
     http://www.mathworks.com/help/matlab/matlab external/upgrading-mex-files-to-use-64-bit-api.html.
To choose a different language, select one from the following:
 mex -setup C++
 mex -setup FORTRAN
```

Figura 10.20: Impostazione del compilatore C++.

Per poter eseguire un modello Simulink in un sistema operativo real-time, che ha una determinata frequenza di lavoro, bisogna impostare correttamente i parametri del modello. In *Configuration parameters* (l'icona dell'ingranaggio), nella finestra *Solver* si deve impostare il tempo finale ad inf, il passo di esecuzione a Fixed-step e pari all'inverso della frequenza del target, e il solutore su discrete. Per compilare il modello, si cerchi NIVeriStand.tlc dopo aver cliccato su *Browse* nella finestra *Code Generation* e si clicchi su *Build*. Al termine del processo si otterrà il messaggio nel *Diagnostic Pane* mostrato in fig. 10.23.

Simulation time	
Start time: 0.0	Stop time: inf
Solver options	
Type: Fixed-step	Solver: discrete (no continuous states)
Fixed-step size (fundamental sample time):	1e-3
Tasking and sample time options	
Periodic sample time constraint:	Unconstrained •
Tasking mode for periodic sample times:	Auto 👻
$\hfill\square$ Automatically handle rate transition for data transfer	
$\hfill \square$ Higher priority value indicates higher task priority	

Figura 10.21: Impostazione del solver.

Target selection								
System target file:	NIVeriStand.tlc							Browse
Language:	С						•	
Description:	NI Phar Lap Real-	al-Time	ne & Windows Targ	jets				
Build process Makefile configura ✓ Generate make Make command:	ition file	make_	e_rtw					
Template makefile	e: N	NIVeri	riStand_vc.tmf					
Code Generation A Select objective: Check model befo	Advisor re generating code	U de: O	Unspecified Off			•	Check Mode	21
Generate code c Package code ar	only nd artifacts				Zi	o file name:	В	uild
					OK	Cancel	Help	Apply

Figura 10.22: Schermata di compilazione.



Figura 10.23: Messaggio di avvenuta compilazione.

Per inserire il modello nel progetto si deve cliccare col tasto destro su *Simulation Models* e aggiungere un modello. Nella schermata di fig. 10.24, cliccando sull'icona della cartella si inserisce il percorso al file .dll appena creato, che è il risultato della compilazione. Il modello appena inserito comparirà nell'albero.

neral	Cattings	Darameters and Signals	Innorts and Outports	
iciai	Settings	Parameters and Signals	inports and Outports	
Nam	e			
Math	Model_Buck	k		
Path				
C-\U	sers\salva\D	ocuments\MATLAB\TORIN	IO\TESI\arduino\	
Math	Model_Buck	k_niVeriStand_rtw\MathMo	del_Buck.dll	_
Simul	ation model	info		
Simul	ation model el rate: 300 H	info Iz		^
Simul	ation model el rate: 300 H	l info Iz		^
Simul Mode Path: Math	ation model el rate: 300 H C:\Users\sa Model Buck	info łz ka∖Documents∖MATLAB∖1 c niVeriStand stwiMatbMo	TORINO\TESI\arduino\	^
Simul Mode Path: Math Modi	ation model el rate: 300 H C:\Users\sa Model_Buck fication date	info łz Iva\Documents\MATLAB\ c_niVeriStand_rtw\MathMo /time: 14:49:45,000 17/07/7	'ORINO\TESI\arduino\ del_Buck.dll 2019	^
Simul Mode Path: Math Modi File s	ation model el rate: 300 F C:\Users\sa Model_Buck fication dat ize (Byte): 60	info Iz Iva\Documents\MATLAB\1 _niVeriStand_rtw\MathMo /time: 14:49:45,000 17/07/ 1416	'ORINO\TESI\arduino\ del_Buck.dll 2019	^
Simul Mode Path: Math Modi File s File v	ation model el rate: 300 H C:\Users\sa Model_Buck ffication dat ize (Byte): 60 ersion: 1.60	info iz lva\Documents\MATLAB\1 _niVeriStand_rtw\MathMo e/time: 14:49:45,000 17/07/2 416	'ORINO\TESI\arduino\ del_Buck.dll 2019	^
Simul Mode Path: Math Modi File s File v Produ	ation model el rate: 300 H C:\Users\sa Model_Buck fication dat ize (Byte): 60 ersion: 1.60 uct name: M aal name:	info iz Iva\Documents\MATLAB\1 c_niVeriStand_rtw\MathMo e/time: 14:49:45,000 17/07/2 416 lathModel_Buck	'ORINO\TESI\arduino\ del_Buck.dll 2019	^
Simul Mode Path: Math Modi File s File v Prode Intern Com	ation model el rate: 300 H C:\Users\sa Model_Buck fication dat ize (Byte): 60 ersion: 1.60 uct name: M nal name:	linfo łz Iva\Documents\MATLAB\\ c_niVeriStand_rtw\MathMo e/time: 14:49:45,000 17/07/ 1416 lathModel_Buck	"ORINO\TESI\arduino\ del_Buck.dll 0019	^
Simul Mode Path: Math Modi File s File v Prode Intern Com Legal	ation model el rate: 300 H C:\Users\sa Model_Buck fication dat ize (Byte): 60 ersion: 1.60 uct name: M nal name: pany name: copyright:	info Iz Iva\Documents\MATLAB\ c_niVeriStand_rtw\MathMo /time: 14:49:45,000 17/07/2 416 lathModel_Buck	'ORINO\TESI\arduino\ del_Buck.dll 019	^
Simul Mode Path: Math Modi File s File v Prode Intern Com Legal File d	ation model el rate: 300 H C:\Users\sa Model_Buck ffication dat ize (Byte): 60 ersion: 1.60 uct name: M nal name: pany name: copyright: copyright:	info iz Iva\Documents\MATLAB\1 _niVeriStand_rtw\MathMo _rtw:14:49:45,000 17/07/3 1416 IathModel_Buck	"ORINO\TESI\arduino\ del_Buck.dll 2019	^
Simul Mode Path: Math Modi File s File v Prode Intern Com Legal File d Mode	ation model el rate: 300 F C:\Users\sa Model_Buck fication dat ize (Byte): 60 ersion: 1.60 uct name: pany name: copyright: escription: el bitness: 32	info iz lva\Documents\MATLAB\7 c_niVeriStand_rtw\MathMo e/time: 14:49:45,000 17/07/3 416 lathModel_Buck ?-bit	TORINO\TESI\arduino\ del_Buck.dll 2019	^
Simul Mode Path: Math Modi File s File v Prode Intern Com Legal File d Mode	ation model el rate: 300 H C:\Users\sa Model_Buck fication dat ize (Byte): 6 ersion: 1.60 uct name: M al name: pany name: copyright: lescription: el bitness: 32	linfo Iz Iva\Documents\MATLAB\ c_niVeriStand_rtw\MathMo e/time: 14:49:45,000 17/07/ 1416 lathModel_Buck !-bit	°ORINO\TESI\arduino\ del_Buck.dll 0019	^
Simul Mode Path: Math Modi File s File v Produ Interr Com Legal File d Mode	ation model el rate: 300 H C:\Users\sa Model_Buck fication dat ize (Byte): 6 ersion: 1.60 uct name: M al name: pany name: copyright: lescription: el bitness: 32	linfo Iva\Documents\MATLAB\ c_niVeriStand_rtw\MathMo e/time: 14:49:45,000 17/07/ 1416 lathModel_Buck 2-bit	TORINO\TESI\arduino\ del_Buck.dll 0019	~

Figura 10.24: Inserimento di un modello nel progetto.

Dalla schermata principale del System Definition File (fig. 10.9 si possono collegare i canali fisici delle schede ai canali "virtuali" del modello tramite la configurazione dei System mappings cliccando sull'icona corrispondente, a destra dell'icona a forma di binocolo. Si effettuano le connessioni come in fig. 10.25.



Figura 10.25: Configurazione delle grandezze del modello.

#### 10.2.3 Il Workspace

Per poter visualizzare le informazioni di interesse, in VeriStand è possibile realizzare un'interfaccia grafica chiamata *Workspace*. In sostanza, il PXI invia all'host le informazioni che devono essere visualizzate o salvate su quest'ultimo tramite il cavo Ethernet, mentre tutto il programma viene eseguito sul PXI stesso.



Figura 10.26: Esempio di Workspace.

Nel Workspace si possono inserire molti tipi diversi di indicatori: grafici cartesiani, grafici di analisi in frequenza, indicatori numerici e a lancetta, maschere per il salvataggio dei dati, ecc. Nella fig. 10.26 viene mostrato un esempio di Workspace per la visualizzazione dei dati di un potenziometro. Per poter aggiungere elementi al Workspace si può premere la combinazione Ctrl+M o Screen  $\implies$  Edit Mode e selezionare il grafico desiderato dal menu. Nel caso di grafici semplici, in fig. 10.27 è indicato come visualizzare una misura sul grafico. Per effettuare un log dei dati, si deve selezionare il Logging control e impostare i campi mostrati in fig. 10.28 come richiesto dall'applicazione.

eneral Format & Precision						
Graph Title:	Plot Area Bao	kground Color:	Stream Rate:	UDP Buffer Size:	History Length:	Update Rat
Graph			100 Hz 🖨	8192 bytes 😫	10,0 sec 😂	5 Hz
Allow Resize?						
Search Te- A		Legend Text		Variable		
		pos		Targets/Contro	ller/Simulation Mode	ls/Models/F

Figura 10.27: Impostazione di un grafico semplice.

regory	Log file path		
nerai e Settings annels te	C:\Users\Public Logs\{Year}\{M	\Documents\National Instrumen onth}\{Day}\Log File_{Hour}_{Mit	tts\NI VeriStand 2017\Projects\potenziometro_lin_SC\  nute}_{Second}.csv
art Trigger	Log file type		
op Trigger amenting	CSV	~	
icros			
operties er Notes	Create Unique	ration	
rget Logs st-Processing itory			
	~		
			OK Cancel H
idit Settings - Data Loggin	g Control		OK Cancel H
idit Settings - Data Loggin regory	ig Control	annel for each channel group	OK Cancel H
idit Settings - Data Loggin egory eral Settings	g Control	annel for each channel group	OK Cancel H
idit Settings - Data Loggin egory eral Settings annels	Include time cha Absolute Time Selected channe	annel for each channel group	OK Cancel H Time channel name Time
idit Settings – Data Loggin egory ieral Settings annels e t Trigger	Include time cha Absolute Time Selected channe Group	annel for each channel group v Is Name	OK Cancel H
dit Settings - Data Loggin egory eral settings nunels e t Trigger perspin perspin	g Control Include time chan Absolute Time Selected channe Group Data	annel for each channel group sts Name Al0	OK Cancel H
dit Settings - Data Loggin egory ereal Settings nonels e t Trigger Drigger menting cros	Include time cha Absolute Time Selected channe Group Data	annel for each channel group	OK Cancel He Time channel name Time Channel path Targets/Controller/Hardware/Chassis/DAQ/DAQ1/Analo
dit Settings - Data Loggin egory ereal Settings mmels e t Trigger Drigger menting cros perties	Include time cha Absolute Time Selected channe Group Data	annel for each channel group v Is Al0	OK Cancel He Time Time Channel path Targets/Controller/Hardware/Chassis/DAQ/DAQ1/Anale
dit Settings - Data Loggin egory eral Settings innels t Trigger menting cros perties r Notes at Loos	Include time cha Absolute Time Selected channe Data	annel for each channel group v Is Name Al0	OK Cancel H
dit Settings - Data Loggin egory eral Settings mnets e t Trigger p Trigger rors perties rok perties rok set Logs tet Processing	g Control Include time cha Absolute Time Selected channe Group Data	annel for each channel group sis Name Al0	OK Cancel H Time channel name Time Channel path Targets/Controller/Hardware/Chassis/DAQ/DAQ1/Analog
dit Settings - Data Loggin egory ereal Settings innets t Trigger Trigger Trigger cros petties robates get Logs pettors pet Logs ory	Include time che Absolute Time Selected channe Group Data	annel for each channel group	OK Cancel H Time channel name Time Channel path Targets/Controller/Hardware/Chassis/DAQ/DAQ1/Analo
dit Settings - Data Loggin ergory reral Settings meta e T Trigger menting cros perties r Notes get Logs t-Processing tory	g Control Include time cha Absolute Time Selected channe Group Data	Is Name Al0	OK Cancel H
dit Settings - Data Loggin egory ereal Settings innels e t Trigger menting cros perties robust set Logs t-Frocessing tory	g Control Include time channe Selected channe Group Data	annel for each channel group st Name Al0	OK Cancel H
idit Settings - Data Loggin egory ereal Settings annels e th Trigger p Trigger cros perties r Notes get Logs t-Processing tory	Include time cha Absolute Time Selected channe Group Data	annel for each channel group	OK Cancel H
idit Settings - Data Loggin egory reral Settings annels e t Trigger p Trigger pringger perties rr Notes get Logs t-Processing tory	g Control	Is Name Al0	OK Cancel He Time channel name Time Channel path Targets/Controller/Hardware/Chassis/DAQ/DAQ1/Analo
dit Settings - Data Loggin egory teral Settings immels e t Trigger menting cros perties r Notes get Logs t-Frocessing tory	g Control Include time cha Absolute Time Selected channe Group Data	annel for each channel group v Is Al0	OK Cancel He Time Time Channel name Channel path Targets/Controller/Hardware/Chassis/DAQ/DAQ1/Anale
idit Settings - Data Loggin egory teral Settings annels e t Tringger mmenting cros perties r Notes r N	g Control Include time cha Absolute Time Selected channe Group Data	annel for each channel group s Name Al0	OK Cancel He Time channel name Time Channel path Targets/Controller/Hardware/Chassis/DAQ/DAQ1/Analo

Figura 10.28: Impostazione Logging Control.

## Capitolo 11

# Creazione del progetto di acquisizione dati

Per l'esecuzione delle prove preliminari e sperimentali e la visualizzazione in tempo reale delle grandezze di interesse è stato creato un progetto VeriStand che acquisisce i segnali provenienti dalla rete CAN a cui è collegata la Ekinox e da un ingresso analogico della DAQ a cui verrà mostrato come collegare Arduino nella sezione *Collegare Arduino al PXI*.

#### 11.1 Definizione delle schede da utilizzare

All'apertura del progetto si espande la voce *Controller* e sotto *NI XNET* aggiungere la scheda CAN come mostrato in fig. 10.14, includendo il database sbgCan.dbc. Il baud rate è stato impostato a 1000000 baud/s.

Aprendo il menu della scheda appena aggiunta si espande fino alla voce *Single port* e cliccando sul destro si sceglie *Import Frames* come in fig. 10.16.

Per la scheda DAQ si eseguono i passi come mostrato nelle figg. 10.11, 10.12a e 10.12b, includendo un'unica porta per le ragioni esposte nella sezione successiva.

#### 11.2 Interfaccia tra Arduino e PXI

In questo capitolo viene mostrato come interfacciare opportunamente Arduino con il controllore PXI in mancanza di schede seriali.

La necessità è quella di passare l'informazione più interessante per gli scopi di questa tesi, ovvero l'heading angle, all'interno del PXI. In questo modo si può subito interrompere la prova nel caso venga rilevata qualche anomalia o un comportamento inatteso da parte della scheda elettronica.

I pin di Arduino sono in grado di generare dei segnali di tensione in uscita, per il comando di altri componenti o la scrittura di informazioni; il PXI, attraverso la Multifunction I/O è in grado di leggere queste tensioni.

Facendo in modo di generare una tensione proporzionale al valore dell'angolo di interesse è quindi possibile confrontare l'angolo proveniente da Arduino con l'angolo, considerato di riferimento, della Ekinox.

#### 11.2.1 I pin analogici di Arduino

Arduino ha dei pin di analog input. Questi sono collegati ad un convertitore A/D interno che non funziona da convertitore D/A in uscita: questo vuol dire che anche i pin analogici, in output, hanno un funzionamento PWM al pari dei pin PWM dedicati della scheda per restituire una tensione *media*, al componente collegato, variabile tra 0 e idealmente 5V, ovvero la massima tensione sui pin di Arduino.

Il funzionamento in modulazione di larghezza di impulso¹ si basa sulla grandezza chiamata **duty cycle**. Detto T il periodo del treno di impulsi e  $t_{on}$  il periodo in cui il pin è alla massima tensione, si definisce come

$$dc = \frac{t_{on}}{T}$$

il duty cycle. Si comprende come il suo valore sia compreso tra 0 e 1. Come si può intuire, con un duty cycle pari a zero la tensione è nulla, mentre al 100% (ovvero 1) la tensione corrisponde al valore massimo.



Figura 11.1: Rappresentazione del PWM in Arduino.

Il tempo T è inversamente proporzionale alla frequenza f di output dei pin di Arduino, che di default è di 488 Hz. Una frequenza così elevata è il motivo per cui, per la maggior parte delle applicazioni, il valore di tensione misurato in uscita è proporzionale al valore del duty cycle ed apparentemente costante.

Collegando un'uscita analogica o PWM ad un ingresso analogico della DAQ NI si riesce però a seguire l'intero andamento temporale dell'input, ovvero a rilevare le variazioni del PWM istantaneamente. Questo costituisce un problema di interfaccia in quanto non esiste più la proporzionalità diretta tra valore dell'angolo e valore di tensione in uscita dal pin.

#### 11.2.2 Soluzioni possibili

Per risolvere il problema in maniera efficace si possono seguire tre strade.

¹La traduzione di Pulse Width Modulation, ovvero PWM.

La prima prevede la simulazione in ambiente Simulink di un circuito che modula la tensione in funzione di un duty cycle, chiamato *convertitore di step down* o *Buck*. Questa soluzione permetterebbe di utilizzare un ingresso digitale della DAQ al posto di un ingresso analogico, che ha il vantaggio di rilevare solo lo stato dell'ingresso, se ad alta o bassa tensione. Tuttavia, le problematiche legate a questo metodo sono molteplici:

- occorre conoscere con esattezza la frequenza di lavoro del PWM di Arduino, in modo da poter impostare una frequenza di acquisizione della DAQ² tale da evitare il fenomeno dell'aliasing. Un segnale ad onda quadra, oltre alla componente principale di frequenza, presenta componenti aggiuntive che sono multipli dispari della frequenza principale che andrebbero ulteriormente filtrate;
- il circuito di *step down* è caratterizzato da una dinamica del secondo ordine: considerando la tensione d'uscita come output e il duty cycle come input del sistema del secondo ordine, occorre calibrare in maniera molto precisa i valori delle impedenze, resistenze e capacità contenute nel circuito, per evitare di avere una risposta troppo lenta, attenuazione in caso di elevata dinamica e amplificazioni nel caso di vicinanza alle frequenze proprie del circuito virtuale.

Anche impostando la frequenza del PWM di Arduino dai 488 Hz di default a circa 30 Hz, per evitare di avere un segnale corrotto dalle *alias-frequenze* sarebbe stato necessario eseguire il loop sul target PXI ad almeno 300 Hz. Per l'applicazione finale, ovvero la guida autonoma, 300 Hz sono troppo elevati per l'esecuzione di una successiva logica di controllo.

La seconda prevede la costruzione di un circuito RC fisico il cui ingresso è il pin di Arduino e l'uscita un pin analogico della DAQ. Questa soluzione risolve tutti i problemi derivanti dalla scelta della frequenza di acquisizione e di esecuzione del modello, tuttavia anche in questo caso la presenza di una costante di tempo può rendere troppo lenta la variazione della grandezza di interesse in uscita.

L'utilizzo di un convertitore D/A o DAC è la soluzione più adatta per le esigenze di interfaccia di questo lavoro. Quello utilizzato in questo lavoro è il **Mouser MCP4921** (fig. 3.12).

A differenza dei pin PWM, questo dispositivo funziona tramite comunicazione *seriale* con Arduino e riceve le informazioni sotto forma di **bit**. Nel caso specifico di questo componente, l'informazione è il livello di tensione desiderato dal pin in uscita in riferimento ad una tensione impostata sul chip.

In questo modo, è possibile ottenere una tensione proporzionale all'angolo di heading calcolato da Arduino senza attenuazione e ritardi. L'unico problema è legato alle oscillazioni di tensione rilevabili dall'input della scheda Multifunction, ma per gli scopi di semplice visualizzazione e confronto delle unità inerziali si può inizialmente tralasciare questo aspetto.

### 11.3 Collegare Arduino al PXI

Per la connessione di Arduino è necessario costruire lo schema mostrato in fig. 11.3. Il chip comunica con Arduino attraverso la *Serial Peripheral Interface* o *SPI* e ne occuperà quindi i pin; dal pin 5 viene fornita la tensione di riferimento, mentre la tensione  $V_{out}$  viene inviata dal pin 8 ed è una frazione della tensione di alimentazione (fig.11.2).

²DAQ DIO Rate nel System Definition File.

Arduino	Mouser DAC
Pin $51$	Pin 4
$Pin \ 52$	Pin 3
$Pin \ 53$	Pin 2

Tabella 11.1: Connessioni Arduino-DAC.

DAC 4921	NI PXI 6123
Pin 8	Pin 30 (AI3+)
Pin 5-7	Pin 63-44 (AI3- e D GND)
Pin 1-6 ( $V_{cc} \in V_{rif}$ )	Pin 14 (5 V+, tensione di riferimento)

Tabella 11.2: Connessioni DAC-Multifunction. Dove compaiono due numeri si intendono dei collegamenti comuni.

V _{DD} 1	$\bullet \cup$	8 V _{OUT}
CS 2	9x1	7 ∨ _{ss}
SCK 3	5P4	6 V _{REF}
SDI 4	W	5 LDAC

Figura 11.2: Pin del DAC4921.



Figura 11.3: Connessione del chip MCP4921.

In particolare, nelle tab. 11.1 e 11.2 sono indicati i collegamenti da effettuare tra i vari componenti del sistema.

Tra i pin 5 e 6 del DAC è stato inserito un condensatore da  $10\mu F$  per stabilizzare la tensione di alimentazione e ridurre il rumore interno al chip.

Come visto in precedenza, per poter salvare tutte le grandezze di Arduino verrà utilizzato il terminale seriale PuTTY. Questo perché sarebbe altrimenti necessario un DAC per ognuno dei segnali da inviare al PXI.

Dal momento che è impossibile sincronizzare i tempi di avvio di questi due sistemi separati, per poter analizzare i dati in post-processing si è effettuato un ultimo collegamento tra scheda Multifunction e Arduino come indicato in tab. 11.3. Da un canale digitale della scheda I/O viene generato un *trigger* che Arduino è in grado di rilevare attraverso uno dei

Arduino	NI PXI 6123
Pin A3	Pin 17 (DO1)
GND	Pin 44 (GND)

Tabella 11.3: Connessioni Arduino-NI 6123.

suoi pin di input. Registrando il momento in cui questo segnale viene generato dal PXI e ricevuto da Arduino, è possibile allineare le storie temporali dei due segnali.

In questo modo sarà possibile confrontare in tempo reale gli andamenti dell'angolo di heading fornito dal filtro di Kalman sviluppato e quelli dati dalla Ekinox.



Figura 11.4: Circuito per la connessione di Arduino alla NI DAQ.

#### 11.3.1 Modifiche al programma Arduino

Per procedere è necessario modificare il programma Arduino affinché possa comunicare attraverso il DAC e leggere la tensione fornita dal PXI.

Per fare ciò si include la libreria del DAC4921, reperibile sul web, e si imposta la comunicazione attraverso il pin 53. Nel setup() si dichiara che il pin A3 sarà un pin di *input* attraverso la funzione pinMode. Prima di fare ciò è opportuno porre il pin in uno stato di alta impedenza tramite il comando digitalWrite(A3,0). In questo modo si evita il rischio che venga letta una tensione positiva.

Attraverso il metodo output della libreria DAC4921 si scrive sul pin 53 il valore compreso tra  $0 e 255^3$  proporzionale all'heading angle,

 $^{^3\}dot{\rm E}$ stato pensato in questo modo in modo da sfruttare la familiarità degli utenti Arduino con la funzione analog<br/>Write.

## Capitolo 12

## Il Workspace in VeriStand

Dopo aver definito tutti gli aspetti del *System Definition File*, si passa alla creazione del **Workspace**, ovvero la schermata dove il PXI permetterà all'utente di visualizzare tutte le variabili richieste. In fig. 12.1 è mostrata la schermata principale relativa ai dati forniti dalla IMU e al singolo dato proveniente dalla scheda Arduino. In particolare si distinguono le varie schermate secondo tab. 12.1.

La seconda schermata del Workspace è stata allestita per la lettura dei dati provenienti dalle due antenne GPS, in previsione di test in cui sia possibile stimare l'heading angle tramite il sistema Dual GPS. Le finestre presenti vengono illustrate in tab. 12.2.

N. elemento	Informazione	Unità di misura
1	Accelerazioni Ekinox	g
2	Angoli di orientamento Ekinox	deg
3	Componenti campo magnetico Ekinox	G
4	Giroscopio Ekinox	rad/s
5	Schermata confronto heading Arduino/Ekinox	deg
6	Intensità di campo magnetico Ekinox	mG
7	Maschera data logging	-
8	Pulsante trigger	V
9	Indicatore numerico heading Arduino	deg
10	Indicatore numerico heading Ekinox	deg
11	Indicatori numerici heading Dual GPS	deg

Tabella 12.1: Tabella per finestra Workspace 12.1.

N. elemento	Informazione	Unità di misura
1	Coordinate GPS 1	deg
2	Coordinate GPS EKF	deg
3	Coordinate GPS 2	deg
4	True heading GPS 1	deg
5	True heading GPS 2	deg
6	Componenti velocità GPS 1, s.r. NED	$\frac{m}{-}$
7	Componenți velocità GPS 2. s.r. NED	$\frac{m}{m}$
•		<u> </u>

Tabella 12.2: Tabella per finestra Workspace 12.2.



Figura 12.1: Workspace: finestra IMU.



Figura 12.2: Workspace: finestra Dual GPS.



Figura 12.3: Workspace: finestra per filtro passa basso.

N. elemento	Informazione	Unità di misura
1	Confronto tra accelerazioni grezze e filtrate	$\frac{m}{s^2}$
2	Confronto tra velocità angolari grezze e filtrate	$\frac{deg}{s}$

Tabella 12.3: Tabella per finestra Workspace 12.3.

#### 12.1 Ulteriori informazioni

In questa sezione vengono illustrate alcune precisazioni in merito alle finestre di tabb. 12.1 e12.2.

Gli indicatori (3) e (6) sono stati inseriti per eseguire un controllo in tempo reale sulla presenza di disturbi magnetici che si possono presentare durante le prove. In particolare, (6) è stato impostato come *Calculated Channel* la cui operazione è

$$H_{tot} = \sqrt{H_x^2 + H_y^2 + H_z^2}$$

dove con H si intende l'intensità del campo misurato dalla Ekinox.

Gli indicatori (9) e (10) sono stati inseriti per evidenziare eventuali scostamenti tra gli angoli di heading stimati dalle due IMU, che potrebbero essere difficilmente leggibili dal diagramma (5).

La Ekinox, date delle coordinate GPS iniziali, è in grado di stimare la nuova posizione GPS tramite un EKF; il risultato di questa stima è quindi mostrato nell'indicatore (2) di fig. 12.2. Ci si aspetta che, nel caso di utilizzo di due antenne GPS, (1), (2) e (3) coincidano.

Date le informazioni GPS, la Ekinox è in grado di calcolare le componenti della velocità di un veicolo nel sistema di riferimento locale NED. Questa informazione non è essenziale alle prove sperimentali.

#### 12.2 Un primo confronto

In vista delle prove sperimentali è stato testato l'algoritmo sviluppato su scheda Arduino rispetto all'algoritmo di EKF presente nella unità Ekinox. Il test è stato eseguito presso i laboratori del DIMEAS.

Dalla fig. 12.4 si nota come le prestazioni degli accelerometri e dei giroscopi delle due unità siano paragonabili. Inoltre, si evidenzia come la Ekinox necessiti di una calibrazione dell'accelerometro, dato un valore di offset della componente x. La stima degli angoli risulta sostanzialmente buona; la presenza di offset significativi negli angoli di roll e pitch indica un possibile miglioramento dei parametri nella scheda Arduino.

Dal momento che sul mezzo in prova andrà eseguita nuovamente la calibrazione di entrambi i sensori, si può affermare che qualitativamente l'algoritmo sviluppato ha un comportamento soddisfacente.



Figura 12.4: Prova di confronto.

## Capitolo 13

# Prove sperimentali per la valutazione dell'algoritmo

In questo capitolo vengono presentati i risultati di alcune prove sperimentali condotte per valutare le prestazioni del filtro di Kalman implementato su scheda Arduino.

Le prove si sono svolte su diversi tracciati, i quali verranno indicati volta per volta. Il veicolo utilizzato per le prove è un quadriciclo 100% elettrico in concessione al dipartimento DIMEAS del Politecnico di Torino.

#### 13.1 Setup sperimentale

Il setup sperimentale è mostrato in fig. 13.1. Per l'alimentazione del PXI è stato necessario utilizzare un cavo di alimentazione DC fornito da National Instruments, in quanto la tensione fornita dall'inverter AC non garantiva il corretto funzionamento del PXI. La tensione al controllore viene quindi fornita direttamente dalla batteria di servizio da 12 V del veicolo; una seconda batteria da 12 V viene utilizzata per l'alimentazione del sensore Ekinox e l'inverter viene comunque collegato alla batteria per permettere sia un rinvio dei cavi dalla batteria al PXI, sia l'eventuale connessione del PC di acquisizione per la ricarica.



Figura 13.1: Il setup sperimentale.

La scheda Arduino riceve l'alimentazione direttamente dal PC di acquisizione.
I sensori sono stati disposti sul tettuccio dell'auto attraverso delle apposite barre di fissaggio. In un test preliminare si è rilevato che la carrozzeria del veicolo non induce particolari disturbi nelle letture dei dispositivi.

Posizionando i sensori sulla mezzeria del veicolo, il particolare aggancio utilizzato per il sensore Ekinox ha portato a definire la conversione di assi e segni indicata in tab.13.1.

Direzione veicolo	Direzione Ekinox	Direzione Arduino
X (verso avanti)	-Y	X
Y (verso sinistra)	X	Υ
Z (verso l'alto)	Ζ	Ζ

Fabella	13.1:	Disposizione	assi.
Laboura	TO'T'	Disposizione	anni.



Figura 13.2: Disposizione dei sensori sul veicolo.

# 13.2 Luoghi delle prove

Le prove mostrate nel seguito sono state condotte nei viali interni del Politecnico e nel circondario. In particolare, i percorsi scelti sono mostrati nelle figg. 13.3 e 13.4a. Il percorso interno è lungo circa 120 metri, mentre il percorso esterno è di circa 1.2 chilometri.



Figura 13.3: Percorso interno, Google Maps.



(a) Percorso esterno prova 6, Google Maps.



(b) Percorso esterno prova 4, Google Maps.

#### 13.2.1 Variazione dei parametri del filtro

Prima di eseguire le prove sperimentali, è stata valutata la possibilità di variare i parametri delle matrici R di (5.9), (5.10) e (5.11), dato che le condizioni operative dei due sensori cambiano rispetto al caso in laboratorio e ci si aspetta più rumore presente nei sensori.

Per poter selezionare dei nuovi dati sono state eseguite diverse prove in rettilineo a velocità costante montando sul veicolo elettrico il solo sensore Arduino ed acquisiti i dati di accelerometro e giroscopio. In questo modo è stato possibile ricavare delle nuove varianze di misurazione, impostando le nuove matrici mostrate di seguito

$$R_x = \begin{bmatrix} 18.4751 & 0\\ 0 & 0.0062 \end{bmatrix}$$
(13.1)

$$R_y = \left[ \begin{array}{cc} 9.7981 & 0\\ 0 & 0.0125 \end{array} \right] \tag{13.2}$$

$$R_z = \begin{bmatrix} 13.6393 & 0\\ 0 & 0.0111 \end{bmatrix}$$
(13.3)

# 13.3 Prove interne

Con queste prove interne si è valutato il funzionamento del sistema sul veicolo e la corretta esecuzione degli algoritmi. Per evidenziare più aspetti, sono state eseguite diverse tipologie di prove, che verranno illustrate nei paragrafi appositi.

#### 13.3.1 Prova su percorso interno

In questa prova sono state eseguite due tipologie di manovre contemporaneamente, ovvero una serie di start and stop e il successivo arrivo al termine del percorso indicato in fig. 13.3.

Gli output dei sensori inerziali sono mostrati nelle figg.13.5, 13.6 e 13.7, mentre il confronto dell'heading e degli angoli di orientamento è mostrato in fig. 13.8.



Figura 13.5: Output accelerometri, prova interna.



Figura 13.6: Output giroscopi, prova interna.



Figura 13.7: Output magnetometri, prova interna. La media di Ekinox è calcolata sull'intera prova, mentre la media di Arduino è calcolata secondo l'algoritmo mostrato nel paragrafo 7.4.



Figura 13.8: Output Kalman filter, prova interna.

Dalle prime due figure è possibile notare come la marcia del veicolo produca delle vibrazioni che entrano come rumore nella lettura di entrambi i sensori. Questo fatto verrà investigato nelle successive prove, per evidenziare l'origine del disturbo.

Dalla fig.13.7 si nota invece come le variazioni rilevate dai due sensori nell'intensità di campo magnetico siano paragonabili fino a circa metà della prova, oltre la quale le due misure iniziano a divergere. È in ogni caso possibile dedurre che le forti variazioni dovute all'intensità magnetica rilevata siano date dalle variazioni subite dall'asse z per entrambi i magnetometri. Questo indica una maggiore sensibilità di questo asse rispetto agli altri due.

La stima del pitch angle, non corretta nella fase iniziale della prova, ha lievemente influenzato la stima dello yaw angle nel tratto iniziale, come si può notare da fig.13.10.

Il forte disaccordo tra le stime mostrate in fig.13.10 indica una necessità di ricalibrazione del magnetometro Arduino e, come detto poco sopra, alla cattiva stima del pitch angle.



Figura 13.9: Output assi magnetometri, prova interna.

Da fig.13.9 viene evidenziato come la differenza tra i valori di intensità letti in fig.13.7 sia dovuta alle componenti  $x \in y$ .

Ricordando quanto evidenziato nel capitolo dedicato alla calibrazione del magnetometro, è possibile additare questo fenomeno al fattore di scala impostato per la costruzione della matrice di compensazione degli assi. Tuttavia, anche se questo fattore di scala non è opportunamente impostato, ai fini di questo studio non è necessario che sia precisamente determinato, in quanto è il rapporto tra i valori rilevati sugli assi  $y \in x$  a determinare l'heading misurato.

Da fig. 13.10 viene evidenziato un offset circa costante tra le misure dei due sensori, trascurando la distorsione iniziale rilevata dal sensore Arduino. Questo fatto può essere dovuto principalmente alla calibrazione e settaggio dei parametri della stima eseguita su scheda Arduino.



Figura 13.10: Differenza tra heading stimati dai due sensori.

Il fatto che riveste più importanza è però la presenza del rumore nelle misure di accelerometro e giroscopio. Sebbene da fig. 13.6 sia ancora possibile distinguere quando il veicolo sta ruotando attorno ad un asse, diventa più difficoltoso identificare i punti di modesta frenata ed accelerazione da fig.13.5, mentre sono facilmente individuabili gli start and stop iniziali.

Come è evidenziato anche dalla figura successiva, questo fenomeno di introduzione di rumore nelle misure è presente solo quando il veicolo è in moto. Il segnale di velocità è stato ricavato dal GPS montato su Ekinox, ed è anche questo una stima che il sensore fa tramite l'EKF integrato.



Figura 13.11: Evidenza del fenomeno di amplificazione in moto.

#### 13.3.2 Valutazione del disturbo

Come possibili cause di disturbo sono state individuate le seguenti:

- trasmissione delle asperità del terreno verso i fissaggi delle barre, i quali non presentano alcun tipo di elemento smorzante;
- rotazione del motore elettrico, il quale inizia a girare solo quando viene richiesta coppia dal driver.

#### 13.3.3 Disturbo derivante dalla rotazione del motore

Per eliminare dalle possibili cause la seconda, sono state eseguite delle prove a veicolo spento. In queste prove è stata valutata la massima ampiezza raggiunta dalle accelerazioni rispetto a quella rilevata durante le prove con veicolo in moto.



Figura 13.12: Differenza nelle accelerazioni rilevate da Ekinox.

Come si può notare dalle figg. 13.12 e 13.13, la differenza è marcata soprattutto per quanto riguarda le accelerazioni  $y \in z$ . Tuttavia, nonostante il veicolo non fosse acceso, si è generato comunque del rumore nelle accelerazioni. Inoltre si rileva come siano indistinguibili i punti di frenata ed accelerazione sugli assi x a causa del rumore presente quando il veicolo è acceso.



Figura 13.13: Differenza nelle accelerazioni rilevate da Arduino.

Nonostante le basse velocità raggiunte col veicolo spinto da spento, le accelerazioni subiscono un rumore considerevole. Questo può portare a pensare che il problema sia legato al manto stradale su cui vengono effettuate le prove o al sistema di fissaggio, ma non alla rotazione del motore.

La differente velocità delle prove fa inoltre sì che a motore acceso il rumore sia presente con una frequenza più elevata. Questo fatto può essere verificato elaborando gli spettrogrammi, su finestre di 1 s e con 95% di overlap tra le finestre, delle prove di fig. 13.12. I diagrammi conseguenti di fig. 13.14 non permettono di ricavare molte informazioni in quanto è notevole la presenza di rumore bianco nel segnale.



Figura 13.14: A sinistra: FFT eseguita sui segnali a veicolo spinto (max scala -30 dB circa).



In questi diagrammi è tuttavia possibile notare che sono presenti delle componenti a bassa frequenza in entrambi i casi, mentre sono evidenti dei picchi a frequenza più elevata soltanto nelle prove in cui il veicolo è in movimento. Questi sono appena visibili nelle prove a veicolo spento, ma sono presenti, dunque non sono da additarsi al motore elettrico.

#### 13.3.4 Disturbo derivante dal sistema di fissaggio

Per poter valutare se i picchi presenti ad alta frequenza sono dovuti al sistema di fissaggio dei sensori, ovvero ai collegamenti delle due barre con il veicolo elettrico, sono state eseguite due prove per ogni sensore, nella prima delle quali il sensore è lasciato sulla barra mentre nella seconda è posto internamente all'abitacolo.

Lo scopo di queste prove non è la stima dell'angolo di heading ma la sola valutazione della fonte di rumore nei sensori. Il posizionamento interno dei sensori, in particolare sul pianale dell'auto davanti al sedile del passeggero, fa sì che il campo magnetico sia tanto distorto da non consentire una stima affidabile dell'angolo.

Ancora una volta ci si affida all'analisi in frequenza per rilevare le cause del rumore. In fig. 13.15 viene mostrato l'output del sensore Ekinox in due prove simili, con accelerazione, arresto e ritorno in retromarcia al punto di partenza con il sensore posizionato rispettivamente fuori e dentro.



Figura 13.15: Diagrammi della prova di confronto dentro/fuori.



Figura 13.16: A sinistra: FFT eseguita sui segnali a sensore esterno. A destra: FFT eseguita sui segnali a sensore interno.

Da ciò che si riscontra da fig. 13.16 si può notare che il posizionamento del sensore sul pianale ha portato ad una riduzione del rumore bianco presente nel segnale e anche dei picchi ad alta frequenza nelle componenti  $x \in y$ . Tuttavia, dei picchi a bassa frequenza sono ancora presenti.

Questo può far concludere che i picchi ad alta frequenza possono essere legati al sistema di fissaggio delle barre che amplifica delle componenti di vibrazione che partono dal veicolo.

## 13.4 Valutazione del filtraggio del segnale delle accelerazioni

Dalle prove precedenti è stato dedotto che, durante la marcia del veicolo ed il sensore posto sul tettuccio, si presentano dei rumori ad alta frequenza in entrambi i sensori¹. La presenza di rumore anche a bassa frequenza è una problematica di difficile risoluzione, ma in questa sezione si vuole valutare l'applicazione di un filtro passa basso da implementare sui segnali delle accelerazioni.

Questo è più uno studio di fattibilità, più che una soluzione vera e propria al problema. Il sensore Ekinox viene utilizzato *as-is.* Il software che lo gestisce, presentato nei capitoli precedenti, non consente di impostare dei filtri aggiuntivi. Per valutare l'efficacia di un filtro passa basso sui segnali di Ekinox si lavora in *post processing* implementando un filtro passa basso in Simulink.

Nella scheda Arduino è possibile invece implementare qualsiasi tipo di filtro, tramite codice, fintantoché l'onere computazionale non diventi troppo elevato.

Un fatto interessante per l'unità IMU MPU6050 utilizzata è la presenza *built-in* di filtri digitali passa basso. Per valutare le performance di questo filtro preesistente, in questa sede si è deciso di attivare tramite la seguente linea di codice

#### mpu.setDLPFMode(MPU6050_DLPF_BW_5)

il filtro passa basso su scheda Arduino.

#### 13.4.1 Implementazione in Simulink del filtro

Una criticità dell'implementazione di un filtro passa basso sui segnali di accelerazione è la perdita di informazioni utili: sebbene sia presente molto rumore, come visto in fig.13.14, filtrare ad una frequenza troppo bassa potrebbe portare a trascurare delle componenti compatibili con le frequenze tipiche della dinamica veicolo, ovvero meno di 10 Hz.

Per poter ottenere un filtro che lavori sia in post processing, sia in real time, con i segnali provenienti da Ekinox, si utilizza il seguente filtro del primo ordine discreto:

$$F(z) = \frac{t_s z + t_s}{\left(t_s + \frac{2}{\omega_b}\right) z + \left(t_s - \frac{2}{\omega_b}\right)}$$
(13.4)

dove  $t_s$  indica il tempo di campionamento,  $\omega_b$  la frequenza di taglio (in  $\frac{rad}{s}$ ) e con z la variabile discreta per il dominio delle frequenze.

L'eq. (13.4) è facilmente implementabile in un blocco *Discrete Transfer Fcn* come in fig. 13.17. La versione implementabile in VeriStand, secondo la procedura indicata nel paragrafo dedicato, è mostrata in fig.13.18. In tab. 13.2 sono mostrati i parametri impostati nel filtro per Ekinox.

¹Per il sensore di Arduino si assume che siano presenti gli stessi disturbi del sensore Ekinox.



Figura 13.17: Modello per le simulazioni del filtro.



Figura 13.18: Modello per l'implementazione del filtro.

Parametro	Valore
$t_s$	0.01 s
$\omega_b$	$2\pi[5, 2, 1, 0.5] \frac{rad}{s}$

Tabella 13.2: Valori utilizzati per lo studio del filtro.

#### 13.4.2 Considerazioni aggiuntive

Per valutare se il filtraggio possa comportare la perdita di informazioni rilevanti, l'analisi è stata svolta su una prova precedentemente mostrata, ovvero quella di fig. 13.5.

Per questa valutazione viene fatta l'ipotesi secondo la quale la velocità laterale del veicolo è trascurabile, sicché derivando l'informazione di velocità registrato durante la prova, questo dato possa esser fatto corrispondere alla sola accelerazione longitudinale. Questo porta ad aggiungere all'ipotesi appena introdotta che la dinamica di rollio di interesse è solo quella dovuta ai trasferimenti di carico in curva. Questo aspetto è in generale restrittivo e si deve valutare il filtraggio della sola componente longitudinale.

Dal momento che l'obiettivo del lavoro non è lo studio del comfort di guida, sull'accelerazione verticale ci si attende unicamente una riduzione del rumore come risultato accettabile.

#### 13.4.3 Risultati del filtraggio

Dall'applicazione del filtro sui dati della prova in fig.13.5 si ottengono i seguenti risultati.



Figura 13.19: Output del filtro per diverse frequenze.



Figura 13.20: Output del filtro per diverse frequenze - zoom su accelerazione longitudinale.



Figura 13.21: Confronto tra la derivata numerica della velocità di fig. 13.11 e le accelerazioni filtrate.

Dalle figg. mostrate si deduce che il filtraggio delle accelerazioni non comporta la perdita di informazioni per frequenze superiori ad 1 Hz. In particolare, come si può vedere da figg. 13.22, l'andamento della velocità nel tratto di start and stop è perfettamente preservato in tutte le prove in quanto a frequenza molto bassa, tipica della dinamica veicolo.

L'analisi in frequenza è stata ottenuta isolando il tratto mostrato a sinistra e applicando una finestra di Hanning per garantire la periodicità del segnale. L'andamento è ben visibile anche nel dominio della frequenza, con un picco pari all'ampiezza del ciclo a circa 0.3 Hz corrispondente alla frequenza delle oscillazioni.





(b) Zoom dell'analisi in frequenza del segnale originario, mostrato a fianco.

Figura 13.22: Segnale originario soggetto a filtraggio.

Si può concludere che, per ottenere un segnale meno caratterizzato da rumore, è possibile applicare un filtraggio anche inferiore ai 2 Hz, almeno sotto le ipotesi introdotte, ma non inferiore ad 1 Hz, in quanto come è riscontrabile da fig.13.20 si introduce un ritardo considerevole superando questa soglia.

#### 13.5 Prove esterne

Le prove condotte sul percorso di fig.13.4b hanno avuto una durata di circa 5 minuti. Ne sono state eseguite due, la prima delle quali per valutare il rumore dovuto alle accelerazioni e le performance generali del filtro su strada, la seconda per riscontrare eventuali fenomeni anomali rilevati.

#### 13.5.1 Prima prova

La prima prova è stata eseguita sul percorso di fig. 13.4b. Di seguito sono mostrati i risultati.



Figura 13.23: Andamento nel tempo della prova.



Figura 13.24: Andamento nel tempo delle velocità angolari.



Figura 13.25: Confronto delle stime degli angoli di orientamento.



Figura 13.26: Confronto degli angoli di heading stimati.



Figura 13.27: Scostamento degli angoli di heading stimati.

Le accelerazioni rilevate dalla MPU6050 di Arduino sono filtrate con un filtro passa basso a 5 Hz, mentre per ragioni di visualizzazione le accelerazioni di Ekinox filtrate a 2 Hz sono mostrate in fig. 13.31.

Dai risultati mostrati si nota come vi sia una maggiore risposta del filtro di Kalman implementato in Arduino alle variazioni istantanee delle accelerazioni. Sebbene evidenti, le variazioni non variano la misura di più di qualche grado per l'angolo di rollio, mentre per l'angolo di beccheggio o pitch lo scostamento è più evidente (fig.13.25). Questo è da additarsi ad un settaggio errato dei parametri del filtro di Kalman; non si può affermare che il problema sia dovuto alla calibrazione in quanto lo stesso problema sarebbe presente nell'angolo di roll.

L'angolo di heading viene stimato in maniera piuttosto fedele solo nella parte centrale della prova, dove lo scostamento dal valore fornito dal sensore di riferimento è inferiore ai 5 gradi; nelle restanti parti la stima risulta inefficace.



Figura 13.28: Valori di campo magnetico e variazioni rispetto al valore medio.

In fig.13.28 le variazioni sono state calcolate rispetto al valore medio della prova per Ekinox, mentre per il magnetometro di Arduino la media è stata calcolata con l'algoritmo presentato nel capitolo *Implementazione di un metodo threshold-based*.

Da questa prova si deduce che i parametri del filtro devono essere ottimizzati, soprattutto per quanto riguarda la stima degli angoli di heading e di pitch.

L'analisi in frequenza del sensore Ekinox ha mostrato ancora una volta la presenza di elevato rumore alle basse frequenze ed un picco nell'intorno dei 40 Hz, come già riscontrato nelle precedenti prove.



(a) Spettrogramma accelerazione x, prova 1.

(b) Spettrogramma accelerazione y, prova 1.

Figura 13.29: Spettrogrammi eseguiti nel tratto centrale della prova, tra 40 e 140 secondi, utilizzando finestre temporali di 1 secondo con 95% di overlap.

Nonostante ciò, un filtraggio a 2 Hz è sufficiente a ripulire parte del segnale e rendere distinguibili le varie fasi del moto, come si può vedere nella figura sottostante.



Figura 13.30: Spettrogramma accelerazione z, prova 1.



Figura 13.31: Confronto tra segnali grezzi e filtrati di Ekinox.

Si può notare che dalla FFT non è possibile ricavare molte informazioni a causa della presenza di rumore bianco piuttosto elevato. Il fatto che il rumore sia concentrato alle

basse frequenze costituisce inoltre un problema di non facile soluzione.

Da questa prima prova si può dedurre che sia necessario ottimizzare i parametri del filtro di Kalman programmato in Arduino, e che la rumorosità indotta nei segnali di accelerometro e giroscopio può essere controllata tramite l'applicazione di filtri passa basso.

#### 13.5.2 Seconda prova

In seguito ai risultati ottenuti nella prima prova esterna, si è deciso di aumentare il valore di R(1,1) per la misura del pitch angle in (13.2) circa raddoppiandolo rispetto al caso precedente:

$$R(1,1)_x = 18.7981$$

ricordando che, in generale, in un filtro di Kalman è possibile dare più o meno peso ad una delle misure andando a modificare i valori di R.

Per visualizzare l'efficacia dell'adattamento del valore di R per le misure dell'angolo di heading tramite magnetometro, viene visualizzato insieme agli altri risultati il valore del fattore  $f(m_i)$  presentato in (7.8) e chiamato nel filtro di Kalman con il nome di rCorr. Il fattore moltiplicativo è stato scelto in k = 50; in questo modo si farà in modo che la variazione del peso di questo parametro non sia troppo repentina verso lo zero.

Di seguito sono mostrati i risultati della prova, eseguita sul percorso di fig.13.4a.



Figura 13.32: Andamento nel tempo della prova.



Figura 13.33: Andamento nel tempo delle velocità angolari.



Figura 13.34: Confronto delle stime degli angoli di orientamento, con il nuovo valore di R(1,1) per il pitch angle misurato.

Da fig.13.34 si nota come il cambiamento del valore di R per il pitch ne ha portato al leggero miglioramento nella parte finale della prova, mentre ulteriori test sono necessari per verificare cosa rende inattendibile la stima alla partenza. Resta confermata l'ipotesi

che, sotto piccoli valori degli angoli di roll e pitch, la stima dell'angolo di heading non è affetta in modo significativo dal loro valore.



Figura 13.35: Confronto degli angoli di heading stimati.



Figura 13.36: Scostamento degli angoli di heading stimati.



Figura 13.37: Valori del fattore di adattamento di R.

In questa seconda prova si può vedere come le differenze dell'angolo di heading siano contenute in ogni momento della prova entro i 20 gradi. Questo conferma la ripetibilità delle misure del sensore Arduino, sebbene siano imprecise rispetto al riferimento dato da Ekinox. Dall'analisi di fig.13.37 si nota come il livello di confidenza della misura magnetica risulta sempre sufficientemente elevato. Intorno a 300 secondi è anche individuabile la risposta del fattore **rCorr** ad una variazione prossima al valore di soglia, indice che l'algoritmo di compensazione lavora in maniera attesa. Inoltre, la fiducia nella misura magnetica ha evitato un fenomeno di drift durante la fase di arresto della prova (fig.13.32) che si è invece presentato nel sensore Ekinox², come si vede in fig.13.39. Dunque in questo tratto il riferimento dato dal sensore Ekinox non permette di trarre conclusioni sulla qualità della stima effettuata da Arduino.



Figura 13.38: Andamento delle misure magnetiche e scostamenti. Le logiche per il calcolo del valore medio di riferimento sono quelle illustrate per la prova precedente.



Figura 13.39: Fenomeno di drift in Ekinox.

 $^{^{2}}$ La velocità indicata dal GPS è un output del filtro di Kalman, dunque è possibile che sia prossima, ma non precisamente pari, a zero quando il veicolo è stazionario.

Per quanto riguarda il contenuto in frequenza del segnale, si ripresentano i picchi nei dintorni dei 40 Hz. Come già era stato possibile ipotizzare nelle prove precedenti, questi picchi devono essere additati al sistema di fissaggio della Ekinox al veicolo, anche se normalmente questi fenomeni sono presenti a più elevate frequenze.

Anche in questo caso, con il filtro passa basso del primo ordine è possibile rilevare gli andamenti del veicolo durante lo svolgimento della prova, anche se non è stato possibile rimuovere tutto il rumore a bassa frequenza.



(a) Spettrogramma accelerazione x, prova 2.

(b) Spettrogramma accelerazione y, prova 2.



Figura 13.41: Spettrogramma accelerazione z, prova 2. Spettrogrammi eseguiti tra 90 e 160 secondi della prova, con finestre di 1 secondo e overlap del 95%.



Figura 13.42: Confronto tra accelerazioni filtrate e grezze di Ekinox, seconda prova esterna.

## 13.6 Conclusioni sulle prove sperimentali

Da quanto osservato nelle varie campagne di prove, si può dedurre che la gestione del filtro di Kalman in Arduino è un'operazione particolarmente complessa in quanto molti sono i parametri da tarare per favorire la buona stima delle grandezze di interesse.

La presenza di rumore nei segnali degli accelerometri di entrambi i sensori inoltre non permette di mantenere una performance adeguata dello stimatore a meno che non si riesca a filtrare opportunamente i segnali. Questa è la strada suggerita dai risultati di figg.13.31 e 13.42.

Il problema presenta anche un'altra criticità, in quanto la misura degli angoli di inclinazione del veicolo è data dal rapporto delle misure accelerometriche. Come si può vedere dalla fig. 13.43, in cui sono stati utilizzati nella (4.12) i segnali grezzi e filtrati ricavati dalla seconda prova, il filtraggio consente di ridurre il rumore anche in queste misure. La misura risulta comunque rumorosa, ma il suo effetto può essere diminuito in vista del *tuning* dei parametri del filtro di Kalman.



Figura 13.43: Riduzione del rumore nella misura dell'angolo $\theta.$ 

L'efficacia dell'algoritmo di compensazione magnetica non può essere giudicata in riferimento all'algoritmo di Kalman Filter presente in Ekinox, in quanto non se ne possono conoscere esattamente le modalità operative.

In condizioni senza particolari disturbi magnetici, tuttavia, la stima dell'angolo di heading effettuata dalla scheda Arduino presenta un certo accordo, a meno di un offset circa costante, con la stima ottenuta da Ekinox durante tratti stazionari o quasi stazionari.

Maggiori valutazioni possono essere fatte eseguendo ulteriori prove sperimentali in condizioni più disturbate, che influiscano con maggiore incisività sulle misure magnetiche effettuate dal sensore Arduino.

In definitiva, le prove sperimentali hanno mostrato come sia possibile ottenere delle performance in generale soddisfacenti montando i sensori magnetici esternamente al veicolo, ma non sufficienti ed adatti ai livelli di precisione richiesti dalla particolare applicazione, ovvero la navigazione autonoma, in cui l'algoritmo si trova ad operare.

I sensori inerziali restituiscono delle misure affette da grande rumore, tuttavia è possibile recuperare le informazioni necessarie attraverso opportuni filtraggi.

# Capitolo 14

# Valutazioni per un'applicazione speciale

In questo capitolo viene illustrato un veicolo speciale cingolato anfibio su cui verrà effettuata una campagna di prove sperimentali per valutare l'applicazione di questo lavoro a questo ambito particolare. Il veicolo oggetto è parte di un progetto volto alla trasformazione in mezzo a navigazione autonoma.

Quelle presentate in questa parte della tesi sono delle prove preliminari, volte a determinare la fattibilità dell'implementazione di uno stimatore per angolo di heading in un mezzo di trasporto così particolare come quello qui presentato.

#### 14.1 Il mezzo

Il veicolo in questione è il **BRT-AATV**, del quale la descrizione qui riportata in sintesi è reperibile in maniera più completa sul sito ufficiale [29]. È un veicolo anfibio cingolato bimodulare articolato a trazione integrale. È in grado di muoversi agilmente su qualsiasi tipo di terreno. Progettato per svolgere operazioni di soccorso e trasporto, ha una struttura modulare che consente l'installazione di attrezzature specifiche. La bassa pressione specifica al suolo consente il transito su strade asfaltate terreni innevati e terreni erbosi senza nessun danneggiamento. La cabina del modulo anteriore in vetroresina autoestinguente è realizzata per trasportare tre passeggeri più il conducente. La versione base del BRT è la versione cabinata, dotata di un ampio pianale sul modulo posteriore che è idoneo ad essere allestito con differenti soluzioni, dal cassone trasporto merci alla cabina per trasporto personale, a delle gru, ecc.



Figura 14.1: Il BRT-AATV.

# 14.2 Specifiche tecniche

Il BRT è spinto da un motore Iveco Diesel 3.0 con cambio automatico. La coppia viene distribuita ai pignoni dei quattro cingoli attraverso due differenziali open e un giunto cardanico tra i due moduli. Le sospensioni sono barre di torsione il cui fine corsa è costituito da un *bump stop*. Nella tabella seguente sono elencate alcune caratteristiche tecniche.

Lunghezza	$7800 \mathrm{mm}$
Larghezza	$2100 \mathrm{~mm}$
Altezza	$2500 \mathrm{~mm}$
Raggio di sterzatura	$\leq 6500 \text{ mm}$
Velocità massima su terra	$65 \frac{km}{h}$
Velocità massima in acqua	$3.5 \frac{km}{h}$
Pendenza superabile	$\geq 60\%$
Pendenza laterale	$\geq 30\%$
Autonomia	500  km

Tabella 14.1: Specifiche BRT.

# Capitolo 15

# Prove sperimentali per veicolo speciale

La campagna di prove è stata eseguita presso gli stabilimenti di Aris S.p.A., azienda produttrice del mezzo. In particolare, si è scelto uno spiazzo sufficientemente grande in modo da essere libero da distorsioni del campo magnetico e vibrazioni indotte da edifici e capannoni presenti nelle vicinanze.

In questa campagna di prove si è voluto determinare soprattutto il contenuto in vibrazione del mezzo durante le varie condizioni operative, in modo da poter valutare quale sia la migliore disposizione della sensoristica per la stima dell'angolo di heading e dei restanti angoli.

Nel dettaglio, in tab. 15.1 sono mostrate le prove effettuate, dove le colonne *Ekinox* ed *Arduino* indicano i sensori effettivamente utilizzati. Per l'acquisizione si è utilizzato un controllore PXI che acquisisce i dati provenienti dal sensore Ekinox e un PC per la registrazione dei dati di entrambi i sensori.

In tutte le prove dove i sensori sono stati utilizzati contemporaneamente, la IMU di Arduino è stata fissata meccanicamente sulla Ekinox ricordando la convenzione di segno¹, mostrata nuovamente in tab. 15.2. Per tutte le prove, i sensori sono stati posizionati sulla mezzeria del veicolo.

N° prova	Nome	Ekinox	Arduino	Note
1	Amb	Sì	Sì	Sensori poggiati su pallet all'esterno
2	I-S	Sì	Sì	Sensori fissati sul vano motore
3	I-A	Sì	Sì	Sensori fissati sul vano motore
4	I-SAS	Sì	Sì	Scaricamento batteria PXI a metà terzo step
5	I-Mov	Sì	Sì	Sensori fissati sul vano motore
6	E-S	Sì	Sì	Sensori fissati sul tetto della scocca
7	E-SAS	Sì	No	Danneggiamento sensore Arduino in fase di trasporto

Tabella 15.1: Prove eseguite. Legenda: Amb = valutazione ambiente, I = interno, E = esterno, A = veicolo acceso, S = veicolo spento, SAS = serie/spegnimentoaccensione/spegnimento, Mov = veicolo in movimento.

¹Dovuta principalmente alla definizione differente degli assi tra Arduino ed Ekinox.

Direzione veicolo	Direzione Ekinox	Direzione Arduino
X (verso avanti)	-Y	X
Y (verso sinistra)	X	Y
Z (verso l'alto)	Ζ	Ζ

Tabella 15.2: Disposizione assi.

### 15.1 Analisi in frequenza tramite Matlab

Per poter valutare il rumore presente nei vari ambienti di prova si esegue la Fast Fourier Transform delle prove statiche tramite il seguente codice Matlab:

```
interval = n:m;
1
\mathbf{2}
   h = hanning(length(Sensore(interval, dato)), 'symmetric');
3
   analisiFreq = fft (Sensore(interval, dato).*h,2<sup>^</sup>nextpow2(length(Sensore(
4
       interval, dato));
   p2 = abs(analisiFreq/length(Sensore(interval,dato)));
5
   p1 = p2(1:length(Sensore(interval, dato)))/2+1);
6
   p1(2:end-1) = 2*2*p1(2:end-1);
7
   baseFreq = freqAcq*(0:length(Sensore(interval,dato)))/2)/length(Sensore(
8
       interval,dato)));
9
10
   figure
11
   plot(baseFreq, p1), grid on
```

Questo codice effettua le seguenti operazioni:

- scelta dell'intervallo di applicazione della FFT;
- costruzione di una finestra di Hann di lunghezza pari al dato acquisito, in maniera tale da passare un segnale "periodico" all'algoritmo FFT (riga 2);
- calcolo della *double-sided-FFT* del dato contenuto in *Sensore* su cui viene applicato uno *zero-padding* come richiesto per migliorare la risoluzione dell'analisi FFT (riga 4);
- calcolo dell'ampiezza, normalizzata rispetto al numero di campioni, della *double-sided-FFT*. Questa operazione viene fatta in modo da riportare il contenuto energetico del segnale elaborato durante la FFT pari a quello originario² (riga 5);
- estrazione della *Single-sided-FFT*. La moltiplicazione per 2 serve allo stesso proposito della normalizzazione rispetto al numero di campioni (righe 6 e 7);
- calcolo dell'asse delle frequenze, i valori del quale saranno compresi tra zero e la frequenza di Nyquist ovvero la metà della frequenza di campionamento.

Per le prove eseguite è stata scelta una frequenza di campionamento di 100 Hz, in linea con le frequenze di lavoro idonei all'applicazione richiesta. Inoltre, non conoscendo la frequenza massima contenuta nei segnali rilevati nelle prove, si è preferito procedere acquisendo ad una frequenza prossima a quella necessaria in futuro.

Per le prove in movimento si utilizza invece una rappresentazione a *spettrogramma*, già vista nel capitolo precedente, che consente di visualizzare l'andamento delle frequenze nel tempo.

²Teorema di Parseval.

## 15.2 Valutazione dell'ambiente di prova

La prova 1, della durata di 5 minuti, è stata condotta per valutare l'eventuale influenza di fattori disturbanti per i sensori. In questo modo, nelle successive prove è possibile rimuovere dall'analisi dei dati l'eventuale presenza di questi disturbi.

Come è possibile vedere dalle figure seguenti, si può affermare che l'ambiente non presenti elementi di disturbo per i sensori, e che dunque tutti i successivi disturbi saranno da additare ad altre cause. Per una maggiore comprensione dei risultati, sono riportati gli istogrammi del giroscopio e le variazioni dei due magnetometri rilevate durante i 5 minuti di prova, mentre per gli accelerometri si riportano degli zoom delle storie temporali.

La breve durata delle prove non ha consentito di effettuare una vera e propria caratterizzazione dei sensori tramite analisi di Allan, dunque per evidenziare delle dispersioni sul breve periodo si sono scelte delle rappresentazioni a distribuzione.

Si rileva inoltre un comportamento anomalo del sensore Ekinox negli output delle accelerazioni di fig.15.1 e dei magnetometri di fig.15.3 che deve essere ulteriormente investigato.



Figura 15.1: Particolare delle letture degli accelerometri. Anomala la lettura dell'asse $\boldsymbol{y}$  di Ekinox.



Figura 15.2: Distribuzione delle letture dei giroscopi.



Figura 15.3: Variazioni delle misure magnetiche.

# 15.3 Valutazione della rumorosità del mezzo

Nella seconda prova i sensori sono stati installati sul pianale del vano motore in modo da poterli collocare sulla mezzeria del veicolo. Il vano e quasi tutti gli elementi dell'abitacolo

sono costruiti con materiali non ferromagnetici; tuttavia, la presenza di alcune piccole parti in acciaio e ferro hanno provocato una variazione nella lettura dell'intensità di campo magnetico per entrambi i sensori. Il magnetometro HMC5883L è stato dunque ricalibrato in modo da compensare queste variazioni con la procedura di calibrazione illustrata nel capitolo *La calibrazione*, mentre per il magnetometro di Ekinox non è stata effettuata alcuna nuova calibrazione.

La differenza tra i valori di intensità registrata dai due sensori è dovuta alla procedura di calibrazione del sensore Arduino illustrata in precedenza, in cui per estrarre la matrice di compensazione degli assi occorre fornire un valore di intensità di campo magnetico. Si ricorda che per la stima dell'angolo di heading e il funzionamento dell'algoritmo di adattamento di R nel filtro di Kalman implementato non è necessario conoscere con esattezza il valore delle componenti del campo magnetico, ma solo i loro rapporti e le loro variazioni.

Non avendo ricalibrato il sensore Ekinox per la nuova condizione operativa è stato possibile rilevare il disturbo introdotto dalle poche parti ferromagnetiche presenti nell'abitacolo del mezzo, pari a circa 150 mG.



(a) Andamento campo magnetico ambiente.

(b) Andamento campo magnetico interno al veicolo.

Figura 15.4: Confronto esterno-interno.

In questa prova si è rilevato che il veicolo, da spento, non introduce particolari disturbi sul funzionamento dei sensori inerziali. Per maggiore completezza si è eseguita una analisi in frequenza sui segnali registrati da Ekinox la quale, avendone registrato i segnali con un tempo di campionamento costante³, facilita il processo.

³Arduino ha un tempo di campionamento dipendente dal tempo di calcolo della iterazione corrente.



Figura 15.5: Particolare delle letture degli accelerometri.



Figura 15.6: Distribuzione delle letture dei giroscopi.



Figura 15.7: Variazioni delle misure magnetiche.



Figura 15.8: Analisi in frequenza, prova 2.

Dal diagramma di fig.15.8 si nota come siano presenti alcuni picchi, la cui ampiezza è però molto contenuta. La loro presenza è probabilmente da additarsi a dei fenomeni presenti all'interno del sensore o all'imprecisione compiuta dall'approssimazione tramite FFT.
### 15.4 Valutazione delle vibrazioni del mezzo

Nella prova 3, della durata di 5 minuti, si è lasciato il mezzo con il motore acceso. La prova è stata avviata alcuni minuti dopo l'accensione del mezzo per evitare il transitorio nelle misurazioni dei sensori. In questo caso la maggiore rumorosità del mezzo ha consentito di rappresentare con una distribuzione anche le accelerazioni, che nel tempo non mostrano più l'andamento "digitale" delle prove precedenti.



(a) Distribuzione delle letture degli accelerometri.

(b) Andamento degli accelerometri.

Figura 15.9: Output degli accelerometri.



Figura 15.10: Distribuzione delle letture dei giroscopi.



Figura 15.11: Variazioni delle misure magnetiche.



Figura 15.12: Analisi in frequenza, prova 3.

L'informazione più interessante si può ricavare andando a combinare i dati mostrati nella fila superiore di figg. 15.9a e 15.12. Il picco ben evidente intorno ai 27 Hz, presente in tutte e tre le accelerazioni, si può additare alla rotazione di idle del motore, insieme al picco più contenuto intorno ai 46 Hz, in quanto rispetto alla prova precedente questo è stato l'unico cambiamento nelle condizioni di rilevazione per i sensori. In generale, dalla analisi in frequenza si rileva che le vibrazioni indotte sul veicolo sono solo dovute agli ordini di rotazione del motore.

Nell'accelerazione x di Ekinox sono presenti diversi picchi; tuttavia, l'unico altro che riveste un certo interesse è quello a 46 Hz, presente in tutte le accelerazioni che, data la bassa frequenza di campionamento utilizzata, potrebbe essere una alias frequenza di uno degli ordini di armoniche introdotti dalla rotazione del motore.



(a) Confronto andamenti nel tempo dei giroscopi (b) Confronto andamenti nel tempo dei giroscopi a veicolo spento. a veicolo acceso.

Figura 15.13: Confronto spento-acceso.

Dal confronto tra le figure 15.13a e 15.13b si può inoltre notare l'importante fatto che il giroscopio di Ekinox non risenta delle vibrazioni in maniera significativa: la rumorosità è comunque elevata ma non subisce cambiamenti. Il giroscopio della IMU di Arduino subisce invece un notevole incremento della rumorosità che potrebbe alterare il funzionamento del filtro di Kalman.

#### 15.5 Valutazione del transitorio spento-acceso-spento

La prova 4, della durata di circa 8 minuti, è stata suddivisa nelle fasi illustrate in fig. 15.14.



Figura 15.14: Sequenza di prova 4.

In questa prova si è notato un comportamento anomalo dei magnetometri di entrambi i sensori. Nelle figg. seguenti si può rilevare come l'accensione del motore abbia portato alla distorsione del campo magnetico presente nel veicolo. Questo ha causato due diversi comportamenti:

- nel sensore di Arduino, la compensazione tramite il fattore *rCorr* nel filtro di Kalman non è stata sufficientemente rapida ad impedire il riconoscimento della distorsione magnetica; questo è dovuto al fatto che la variazione iniziale era prossima al valore di soglia impostato di 100 mG (fig.15.16);
- in Ekinox, il sensore ha lentamente driftato dopo aver correttamente rigettato la nuova misura del magnetometro dopo l'accensione del motore. Questo indica che anche il filtro presente nella Ekinox è adattativo, ma molto più insensibile alle variazioni di campo magnetico.



Figura 15.15: Evidente l'effetto della distorsione magnetica.



Figura 15.16: Prova interna, ciclo accensione-spegnimento.

Da fig.15.16 si vede come il superamento del valore di soglia non sia netto a sufficienza per escludere tutte le misure dovute all'accensione del motore, con conseguente aggiornamento della media; scegliendo un valore di th opportuno è invece possibile escludere il

magnetometro in maniera continua.

Anche in questo caso si ritrovano le frequenze precedentemente viste. La FFT di fig.15.17a è riferita alla sola parte a motore acceso, come si vede da fig.15.17b.



#### 15.6 Valutazione del posizionamento esterno del sensore

In questa prova non sono stati rilevati particolari cambiamenti rispetto alla prova 2, con veicolo spento e sensori posti esternamente.



Figura 15.18: Particolare delle letture degli accelerometri.



Figura 15.19: Distribuzione delle letture dei giroscopi.



Figura 15.20: Variazioni delle misure magnetiche.

### 15.7 Valutazione del rumore esterno a veicolo acceso

Nella prova 7 si è eseguito il ciclo indicato in tab. 15.1. Dalle figg. 15.22a e 15.22b si può notare come all'istante dell'accensione del motore il campo magnetico rilevato subisca una

variazione consistente, ma molto meno marcata rispetto a quella visibile in fig. 15.16.

In aggiunta alla minima influenza che questa variazione ha sulla stima dell'angolo di heading della Ekinox⁴, come si può notare in fig. 15.23, si può concludere che un posizionamento sulla parte superiore del veicolo può ridurre i problemi legati alla calibrazione del sensore.



(a) Evidente il fenomeno di accensione del (b) Particolare dell'andamento delle acceleraziomotore. ni.

Figura 15.21: Accelerazioni prova 7.



(a) Meno evidente il fenomeno di accensione del (b) Variazione del campo magnetico nella prova motore, stando alla scala della figura. interna.

Figura 15.22: Grandezze magnetiche prova 7.

⁴Come indicato in tab. 15.1, non è stato possibile utilizzare il sensore Arduino per la prova in oggetto.



Figura 15.23: Andamento degli angoli di orientamento prova 7.

Si conferma che la frequenza di rotazione del motore in idle è quella già rilevata nelle prove precedenti, come si può vedere nelle figg. 15.25. In particolare, si nota come l'ampiezza rilevata in corrispondenza della frequenza di rotazione del motore si sia lievemente ridotta per le componenti  $y \in z$ , mentre in x si ha un'ampiezza maggiore, dovuta probabilmente ai moti della carrozzeria del veicolo che, non essendo quella definitiva, è solo fissata sul telaio. Si nota inoltre come alla scocca del veicolo non venga trasmessa la vibrazione a 46 Hz su y.



Figura 15.24: Confronto tra le FFT a motore acceso.



Figura 15.25: Analisi in frequenza della prova a motore acceso: confronto.

#### 15.8 Comportamento del veicolo in moto

Come ultima prova, viene presentata la prova dinamica numero 5 (tab. 15.1). In questa prova si è voluta valutare la possibilità di tenere i sensori per la navigazione internamente al veicolo, sia la performance dello stimatore della scheda Arduino.

La prova, una manovra ad L con curva a sinistra, si è svolta nelle seguenti parti:

- veicolo stazionario e spento per un minuto;
- veicolo stazionario ed acceso per un minuto;
- avanzamento fino alla curva e arresto per un minuto;
- impiego curva ed arresto per un minuto;
- arrivo al termine del percorso, arresto per un minuto e spegnimento trenta secondi.

Il risultato della manovra è mostrato nella fig. 15.27.



Figura 15.26: Istanti di cambiamento condizioni prova 5.



Figura 15.27: Andamento orientamento prova 5.

Si nota che all'accensione del motore l'algoritmo ha correttamente escluso la variazione superiore al valore di soglia (impostato a 100 mG). Come si può vedere da fig.15.29, ciò ha comportato delle variazioni nelle misure sugli assi x ed y che non hanno alterato il rapporto tra le componenti che definiscono lo yaw angle da misure magnetiche in maniera significativa. Tuttavia, raggiunta la curva la stima ha subito una forte divergenza in quanto il magnetometro non è stato calibrato per la condizione operativa di motore acceso. Dopo l'esecuzione della curva, il magnetometro ha anche indotto la stima ad oscillare repentinamente tra 180 e -180 gradi in quanto, come si vede da fig.15.28, il fattore di compensazione per R non era sufficientemente basso da rifiutare la misura.

Ekinox, come già evidenziato precedentemente, non è praticamente influenzata da eventi magnetici istantanei.



Figura 15.28: Valutazione adattatività  ${\cal R}$  prova 5.



Figura 15.29: Andamento delle tre componenti magnetiche rilevate da Arduino.

Inoltre, dalla fig. 15.27, si nota che l'algoritmo di Kalman Filter implementato in Arduino è più influenzato dai fenomeni di dinamica rispetto ad Ekinox. Dalla figura seguente si nota come Arduino non restituisca una stima soddisfacente in quanto la differenza rispetto al valore fornito da Ekinox è sempre di almeno 10 gradi, per poi divergere una volta intrapresa la curva.



Figura 15.30: Scostamento tra heading di Ekinox ed heading Arduino.

In generale, la stima effettuata dalla IMU equipaggiata dalla scheda Arduino è fortemente influenzata da fattori magnetici, più che da effetti dinamici. Questa dipendenza può essere ridotta andando a variare i parametri del filtro di Kalman implementato e riducendo il peso dato alle misure magnetiche; un altro margine di miglioramento è rappresentato dalla calibrazione a motore acceso, in quanto la procedura di calibrazione può compensare i disturbi presenti.

Si può notare da fig.15.31 che la performance dell'algoritmo di compensazione dei disturbi magnetici può essere migliorata scegliendo un valore di th più basso, in accordo a quanto detto nel capitolo Implementazione di un metodo threshold-based.



Figura 15.31: Differenza di comportamento a seconda del valore di soglia th impostato.

Per quanto riguarda il contenuto in frequenza durante il moto, si assiste ad un aumento del rumore ad alta frequenza e alla presenza del picco in corrispondenza della velocità del motore da fermo, condizione che si è verificata svariate volte durante la prova.

Nell'accelerazione longitudinale compare anche evidente del rumore intorno ai 15 Hz. Questo può essere un fenomeno di amplificazione del picco presente in fig.15.12 alla stessa frequenza.

Sebbene sia presente del rumore anche alle basse frequenze, fig.15.32a suggerisce la possibilità di filtrare le accelerazioni misurate, mostrate in fig. successiva, con un filtro passa basso. È anzi necessario applicare un qualche tipo di filtraggio in modo da poter individuare le frenate e le accelerazioni, nonché eventuali trasferimenti di carico.



(a) Spettrogramma accelerazione x.

(b) Spettrogramma accelerazione y.



Figura 15.33: Spettrogramma accelerazione z.



Figura 15.34: Andamento delle accelerazioni, prova 5, con filtro e senza filtro.

Applicando in real-time un filtro passa basso a 2 Hz tramite modello Simulink si ottiene quanto mostrato in fig. 15.35, dove sono distinguibili le accelerazioni e le frenate.



Figura 15.35: Filtraggio di  $a_x$ , prova 5.

### 15.9 Conclusioni sulle prove sperimentali

In questa sessione di prove è stata valutata l'applicazione ad un veicolo speciale di due sensori IMU per la stima dell'angolo di heading e contestualmente degli altri due angoli di

orientamento. In base ai risultati ottenuti si può dedurre che la quantità di vibrazioni che il veicolo trasmette ai sensori non rende possibile la lettura di alcune quantità di interesse come le accelerazioni e le velocità angolari in maniera diretta, ma è necessario un filtraggio per rimuovere le componenti indesiderate.

Durante il moto del veicolo i tre angoli sono stati stimati con una buona precisione dal sensore Ekinox, anche se non è possibile dare giudizi assoluti sulla stima dell'angolo di heading in quanto, come visto nelle varie prove, entrambi i sensori patiscono l'accensione e messa in moto del motore che ne inficia la precisione. Non si può affermare che il fenomeno sia dovuto alla temperatura di funzionamento del motore in quanto le variazioni vengono percepite istantaneamente.

Il posizionamento esterno del sensore sembra essere la strada da percorrere per risolvere questo problema per quanto riguarda la stima dell'orientamento, mentre dà pochi benefici in termini di riduzione delle accelerazioni dovute alla rotazione del motore (fig.15.25).

Non si possono trarre conclusioni sul posizionamento esterno durante la marcia del veicolo, in quanto non è stato possibile garantire un fissaggio sicuro alla scocca del veicolo volto ad evitare danneggiamenti del sensore inerziale.

L'utilizzo di opportuni supporti per montare due antenne GPS da collegare alla Ekinox risolverebbe con ogni probabilità i problemi legati all'orientamento.

In generale, la stima effettuata dalla scheda Arduino sugli angoli di orientamento è sufficiente quando la dinamica è stazionaria o quasi stazionaria, mentre risente ancora in maniera significativa delle accelerazioni e delle variazioni di campo magnetico in maniera troppo marcata.

In particolare, la stima dell'angolo di heading è troppo influenzata dalla presenza di disturbi e sembra non essere adatta per questo tipo di applicazione. Tuttavia, una ottimizzazione offline dei parametri, unita ad una calibrazione opportuna del magnetometro eseguita nelle condizioni operative del mezzo ovvero a motore acceso potrebbe portare ad un miglioramento delle prestazioni dell'algoritmo. È inoltre necessario implementare un algoritmo di rifiuto delle misure magnetiche, similmente a quanto sembra essere implementato su Ekinox, per poter affidarsi unicamente alla stima del giroscopio in quei tratti in cui non si può considerare attendibile la misura magnetica.

### 15.10 Sviluppi futuri

In ottica degli sviluppi futuri del progetto, si può pensare di adottare un nuovo metodo per la stima degli angoli di orientamento e di conseguenza dell'angolo di heading del veicolo. È evidente che le particolari condizioni di cui il magnetometro necessita per il funzionamento ottimale non lo rendono adatto ad un utilizzo su un veicolo di questo tipo. L'implementazione di un algoritmo che sfrutti due antenne GPS risulta sì più complessa dal punto di vista di un'eventuale programmazione, ma più solida ed affidabile per quanto riguarda l'accuratezza del risultato.

In questo modo ci si può svincolare dalla necessità di posizionare il sensore esternamente al veicolo, fissandolo opportunamente in un luogo protetto dell'abitacolo, che a quel punto non necessiterebbe nemmeno più di essere amagnetico.

## Capitolo 16

# Conclusioni

In questo lavoro di tesi è stato sviluppato un algoritmo per filtro di Kalman per la stima dell'angolo di heading, anche adattato alla stima degli altri angoli, da utilizzare nell'ambito della navigazione autonoma dei veicoli. La peculiarità dell'applicazione rende necessario che la stima di questa grandezza sia sufficientemente precisa.

L'indagine sull'utilizzo di sensori low cost per lo sviluppo di tale algoritmo ha mostrato come la scelta di un modello matematico particolarmente semplice e di onere computazionale basso ha permesso di raggiungere dei risultati soddisfacenti se messi a confronto con l'unità inerziale di riferimento. Tuttavia, la performance dell'algoritmo è fortemente legata alle procedura di calibrazione dei vari sensori utilizzati. Accelerometro e giroscopio in generale non presentano delle criticità nella calibrazione, mentre non si può dire lo stesso del magnetometro. Inoltre, senza un'opportuna schermatura spesso la misura risulta troppo imprecisa e di conseguenza inaffidabile.

I vari metodi di compensazione introdotti hanno consentito di migliorare le prestazioni del semplice modello implementato, come si è potuto riscontrare nelle diverse prove sperimentali eseguite. Dal confronto con l'unità di riferimento, tuttavia, non è possibile affermare che le prestazioni siano paragonabili e in ogni situazione affidabili.

Il metodo di compensazione dei disturbi magnetici, introdotto in questa tesi, ha dato delle performance soddisfacenti nelle prove su strada, dove la particolare costruzione del veicolo di prova ha reso possibile una calibrazione soddisfacente ma non perfetta dati gli offset costanti rispetto al sensore di riferimento. Inoltre, l'adattamento continuo del valore della matrice R da associare alla misura dell'heading magnetico ha consentito di avere una stima meno nervosa delle effettive misure del sensore magnetico anche se non sempre solida.

Questo metodo è entrato in crisi nelle prove sul veicolo speciale; tuttavia, si ritiene che i problemi sorti, che ne hanno decretato in quella sede l'inefficacia, possano essere risolti con una calibrazione del sensore a veicolo con motore acceso e con una soglia per il riconoscimento dei disturbi più stringente, superata la quale viene utilizzato solo il segnale del giroscopio opportunamente corretto dal bias stimato dal modello.

Inoltre, è da valutare un metodo che utilizzi o una media mobile che viene aggiornata in maniera perpetua, senza reset ogni *tot* campioni, o una media costante ottenuta da valori di campo magnetico rilevati in fase di calibrazione. In questo modo, piuttosto che utilizzare un metodo *threshold-based* che lavora sull'intensità magnetica, se ne può utilizzare uno che aggiorni unicamente la varianza  $\sigma_{mag}^2$  delle misure magnetiche.

La necessità di interfacciare due sistemi molto diversi tra loro, ovvero un microcontrol-

lore elettronico seriale come Arduino e un controllore analogico/digitale come il PXI ha costretto l'introduzione di ulteriori elementi nel sistema. Per una maggiore compattezza, nella soluzione finale si suggerisce di utilizzare sensoristica preposta alla comunicazione con il controllore in oggetto.

**In conclusione**, lo studio preliminare sull'applicazione alla guida autonoma sviluppato in questa tesi merita un'attenzione ed approfondimento ulteriori. La strada intrapresa suggerisce l'utilizzo di unità di fascia superiore che non necessiti di complesse procedure di calibrazione e compensazione.

La maggiore precisione si può ottenere in un sistema che utilizzi anche la stima di almeno un'antenna GPS.

# Bibliografia

- [1] Daniel Roetenberg Henk J. Luinge Chris T. M. Baten and Peter H. Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, 2005.
- [2] Karl Berntorp. Joint wheel-slip and vehicle-motion estimation based on interial, gps and wheel speed sensors. *IEEE Transactions on Control Systems technology*, 2015.
- [3] Qingguo Li Bingfei Fan and Tao Liu. How magnetic disturbance influences the attitude and heading in magnetic and inertial sensor-based orientation estimation. Sensors, 2011.
- [4] Salvatore Campolo. Development of a motion and orientation sensor.
- [5] Natural Resources Canada.
- [6] Michael J. Caruso. Applications of magnetic sensors for low cost compass systems.
- [7] Michael J. Caruso. Applications of magnetoresistive sensors in navigation systems.
- [8] Terry Moore Christopher Hide and Martin Smith. Adaptive kalman filtering for low-cost ins-gps. *The Journal of Navigation*, 2003.
- [9] Andrea Civalleri. Allestimento di un banco prova per cambi a doppia frizione.
- [10] Deives Roberto Bareta Cristian Padilha Fontoura, Alan Alves dos Passos and Elton Joao Zanol. Comparison between vertical acceleration data from acquired signals and multibody model for an off-road vehicle. *Scientia cum industria*, 2018.
- [11] Lars Hammarstrand Erik Stenborg. Using a single band gnss receiver to improve relative positioning in autonomous cars. *IEEE Intelligent vehicle Symposium*, 2016.
- [12] Ramsey Faragher. Understanding the basis of the kalman filter via a simple and intuitive derivation. *IEEE Signal Processing Magazine*, 2012.
- [13] National Centers for Envornmental Information.
- [14] Gary Bishop Greg Welch. An introduction to the kalman filter. SIGGRAPH 2001, 2001.
- [15] I. Lee H. Kim. Localization of a car based on multi-sensor fusion. The International Archive of photogrammetry, 2018.
- [16] Songlai Han and Jinling Wang. A novel method to integrate imu and magnetometers in attitude and heading reference systems. -, -.
- [17] Djoko Purwanto Hany Ferdinando, Handry Khoswanto. Embedded kalman filter for imu on the atmega8535. *IEEE Symposium*, 2012.

- [18] Piotr Krauze et al. Jerzy Kasprzyk. Vibration control in semi-active suspension of the experimental off-road vehicle using information about suspension deflection. Archives of control sciences, 2017.
- [19] lifeAugmented. Parameters and calibration of a low-g 3-axis accelerometer.
- [20] Valerie Renaudin Muhammad Haris Afzal and Gerard Lachapelle. Magnetic field based heading estimation for pedestrian navigation environments. *Department of Geomatics Engineering University of Calgary*, 2011.
- [21] Pedro Neto, Nuno Mendes, and A. Paulo Moreira. Kalman filter-based yaw angle estimation by fusing inertial and magnetic sensing: a case study using low cost sensors. *Emerald Sensor Review*, 2015.
- [22] NI. VeriStand Coursebook. National Instruments, 2011.
- [23] Talat Ozyagcilar. Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors. Freescale Semiconductor Inc., 2011.
- [24] Talat Ozyagcilar. Calibrating an eCompass in the Presence of Hard- and Soft-Iron Interference. Freescale Semiconductor Inc., 2015.
- [25] Mark Pedley. *Tilt Sensing Using a Three-Axis Accelerometer*. Freescale Semiconductor Inc., 2011.
- [26] Subhas Mukhopadhyay Pengfei Gui, Liqiong Tang. Mems based imu for tilting measurement: Comparison of complementary filter and kalman filter based data fusion. *IEEE Proceedings*, 2015.
- [27] Fabio Scibona. Sistema di navigazione basato sul filtro di kalman per una piattaforma d'assetto a tre assi. Ingegneria aerospaziale, 2015.
- [28] Dan Simon. Kalman filtering. Embedded Systems Programming, 2001.
- [29] Aris SpA. Presentazione brt-aatv.
- [30] SBG Systems. Ekinox ins user manual.
- [31] Hassen Fourati Thibaud Michel, Pierre Geneves and Nabil Layaida. On attitude estimation with smartphones. International Conference on Pervasive Computing and Communications, 2017.
- [32] Jinling Wang Wei Li. Effective adaptive kalman filter for memsimu magnetometers integrated attitude and heading reference systems. *Journal of Navigation*, 2013.