



UNIVERSITA' DEGLI STUDI DI PADOVA

FACOLTA' DI INGEGNERIA

Corso di Laurea in Ingegneria dell'Automazione

**ALGORITMI DI STABILIZZAZIONE DELLA
CAMMINATA DI ROBOT UMANOIDE**

Laureando

Carlo Tavian

Relatore

Prof. Emanuele Menegatti

ANNO ACCADEMICO 2010/2011

A ...

«citazione» *autore*

Indice

1	Introduzione	1
1.1	Camminata passiva	4
1.2	Camminata attiva	5
1.2.1	Camminata statica	6
1.2.2	Camminata dinamica	7
1.3	Virtual Model Control	9
2	Apparato Sperimentale	13
2.1	Robovie-X	13
2.2	Software	15
3	Scelta del tipo di camminata	19
3.1	Limitazioni dell'apparato sperimentale	19
3.2	Suddivisione dell'algoritmo della camminata	21
4	Protocollo USB	23
4.1	Struttura del pacchetto USB	24
4.2	Lettura e scrittura delle variabili interne	24
4.2.1	Scrittura variabile interna	25
4.2.2	Lettura di variabile interna	26
4.2.3	Lettura dei sensori inerziali	26
5	Cinematica di posizione	29
5.1	La cinematica inversa	29
5.2	Piano Frontale	32

5.3	Piano Sagittale	34
6	Sensore inerziale e filtro	37
6.1	Filtro per sensori inerziali e inerziali/magnetici	39
6.1.1	Definizione di quaternione	40
6.1.2	Alcune proprietà dei quaternioni	41
6.1.3	Calibrazione del sensore inerziale	45
6.2	Ricavare l'orientazione dal sensore inerziale	49
6.2.1	Ricavare l'orientazione dall'output dei giroscopi	49
6.2.2	Ricavare l'orientazione dall'output degli accelerometri	50
6.2.3	Adattamenti per la scheda VS-IX001	52
6.2.4	Fusione delle stime di giroscopi ed accelerometri	53
6.2.5	Risultati	54
7	Equilibrio su piano ad inclinazione variabile	59
7.1	Struttura del programma	59
7.1.1	Posizione eretta	59
7.1.2	Calibrazione ed inizializzazione	61
7.1.3	Processi di lettura IMU e di comando dei motori	61
7.1.4	Processo di comando dei motori	61
7.1.5	Tuning dei parametri	67
8	Camminata	71
8.1	Correzioni tempo-varianti e tempo-invarianti	71
8.2	Problemi dei giroscopi durante la camminata	72
8.2.1	Bias variabile dei giroscopi	72
8.2.2	Eliminazione del bias dei giroscopi	73
8.3	Rilevazione dell'orientazione del piano durante la camminata	76
8.4	Correzione del beccheggio	77
8.5	Correzione del rollio	79
8.6	Risultati	82
9	Conclusioni e sviluppi futuri	87

INDICE

vii

Bibliography

90

Capitolo 1

Introduzione

La tecnologia dei robot è utilizzata da molto con profitto in molteplici campi: per lavori pesanti o pericolosi, lavori in ambienti difficilmente raggiungibili o malsani per l'uomo, per velocizzare operazioni ripetitive, eccetera. La differenza tra un robot e una semplice macchina utensile è l'adattabilità, i bracci robotici ad esempio infatti possono svolgere differenti compiti in base all'utensile che viene montato sull'ultimo link: lo stesso braccio robotico può effettuare una saldatura oppure operazioni di *pick-and-place* a seconda che il suo hand-effector sia collegato ad un saldatore od a una pinza. Ciononostante, questi robot sono studiati e hanno caratteristiche morfologiche adattate specificamente al loro compito e, per la maggior parte sono vincolati all'ambiente o alla cella di lavoro in cui operano. Negli ultimi anni vanno aumentando gli studi di robot adatti all'interazione con gli umani sotto molteplici aspetti: come ausilio in compiti particolari, come interfaccia per altre macchine, o più ambiziosamente come rimpiazzo per la maggior parte di compiti che può essere svolta da esseri umani. Particolarmente avanzata in questo campo è la società giapponese, dove i robot sono già presenti in diversi ambienti per svolgere compiti semplici, come guidare gli ospiti all'interno degli edifici di una società. Ci sono diversi aspetti cruciali da sviluppare nella transizione tra robot industriali e robot inseriti nell'ambiente umano: sicurezza (nelle applicazioni industriali non è mai permesso avere un robot attivo in presenza di un'operatore), percezione dell'ambiente, percezione delle

persone, interfacce uomo-macchina e, non ultimo aspetto, la mobilità. Nella maggior parte delle applicazioni industriali i robot sono vincolati alla cella o al limite si muovono su percorsi prestabiliti su carrelli e rotaie; i robot destinati all'interazione con umani devono poter godere di una certa mobilità ed in particolare i loro sistemi di movimento devono potersi ben adattare ad un ambiente che è studiato per umani. Attualmente, la locomozione più efficiente dal punto di vista energetico è senza dubbio quella su ruote, ma sia gli ambienti umani che quelli naturali presentano scalini e ostacoli che rendono questo mezzo di locomozione poco efficace. Si è perciò passati alla locomozione per mezzo di gambe: osservando la natura, l'uomo ha ideato vari tipi di locomozione attraverso l'uso di differenti numeri e tipi di gambe e creando soprattutto robot bipedi, quadrupedi ed esapodi, prendendo spunto da umani, quadrupedi ed insetti.

Idealmente, un numero maggiore di gambe significa implica semplicità maggiore nel mantenere l'equilibrio ed una maggior adattabilità a terreni sconnessi; questa scelta ad esempio è stata utilizzata per prototipo di un veicolo adatto alla locomozione in ambienti naturali. Considerando però i robot orientati all'interazione umana, c'è da considerare come ulteriore aspetto: l'accettazione da parte delle persone. Questi robot dovranno essere presenti 24 ore su 24 nelle case e negli ambienti frequentati da umani e devono svolgere efficientemente il loro compito ma anche risultare graditi agli utenti, e difficilmente un robot che ricordi un insetto susciterà empatia nelle persone, al contrario un robot con le sembianze umane costituirà una presenza molto più familiare. Bisogna tener presente però che l'empatia delle persone verso i robot dipende in maniera peculiare dalla somiglianza di questi ultimi con gli umani: esiste un fenomeno psicologico (*Uncanny Valley*) secondo cui ad un certo livello di somiglianza con gli umani, un robot genera una certa repulsione psicologica. Anche il movimento del robot gioca un ruolo importante, amplificando ulteriormente il fenomeno (fig 1.2); un movimento naturale e somigliante all'andatura umana contribuisce significativamente all'accettazione dello stesso.

La somiglianza morfologica dei robot è importante sia dal punto di vista dell'accettazione da parte degli umani, che dal punto di vista funzionale: se

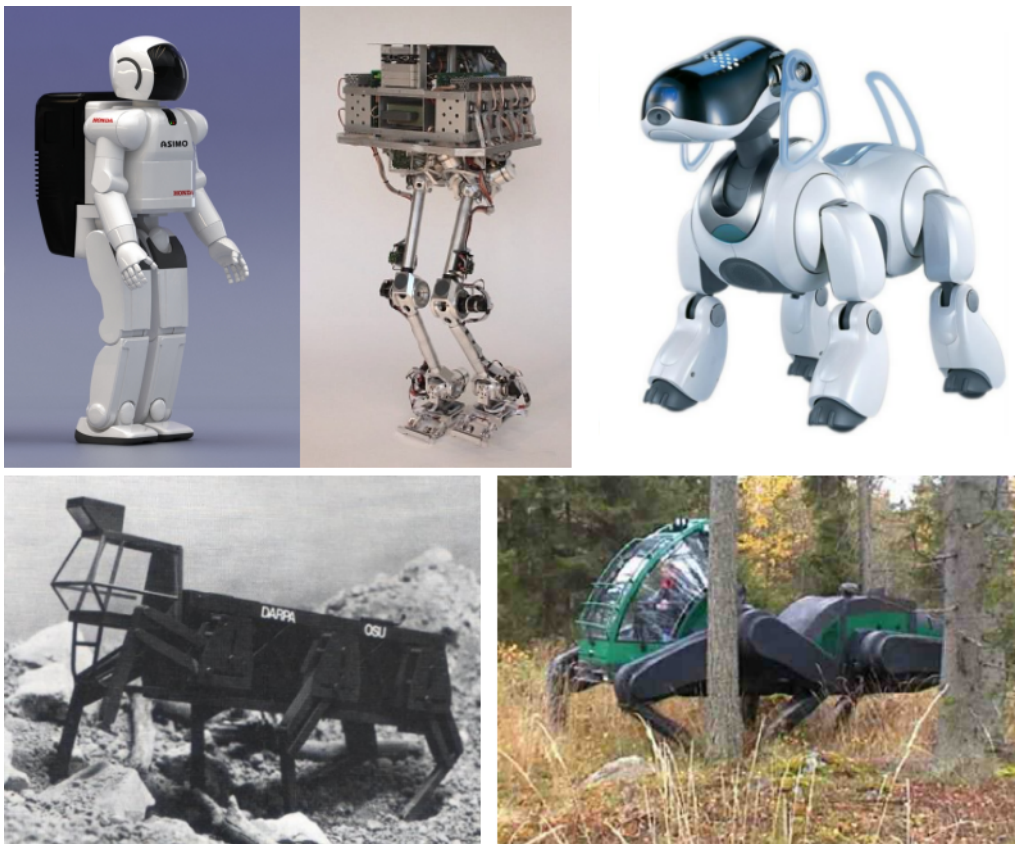


Figura 1.1: Locomozione attraverso gambe: esempi di robot bipedi, quadrupedi ed esapodi

si riuscisse a replicare in tutti i suoi aspetti la struttura umana attraverso un robot, questi potrebbe interagire con l'ambiente attraverso le stesse interfacce utilizzate dalle persone, senza richiedere così alcun adattamento dell'ambiente. In questa direzione si sta muovendo anche la NASA che attualmente sta lavorando ad un robot umanoide chiamato "Robonaut 2" destinato ad assistere gli astronauti a bordo della ISS ed è progettato per poter utilizzare gli stessi strumenti utilizzati dagli astronauti (come ad esempio un trapano) con le sue mani, senza necessità di una particolare interfaccia con lo strumento. Per tutti questi motivi è utile studiare la camminata bipede per robot umanoide, al fine di riuscire ad ottenere una andatura il più possibile naturale e simile a quella umana sia dal punto di vista funzionale che dal punto di vista

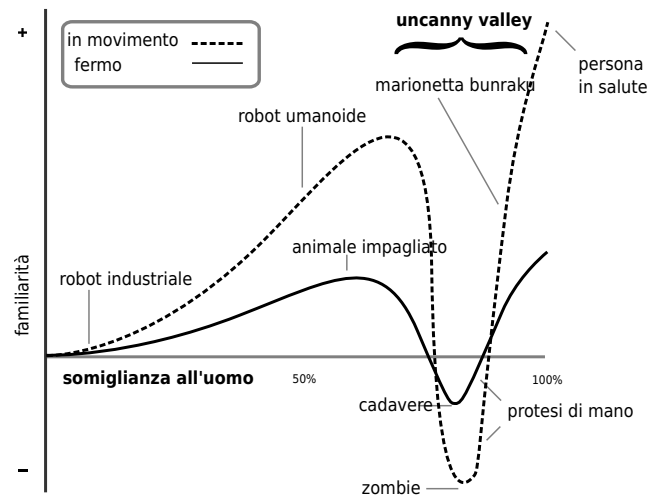


Figura 1.2: *Uncanny Valley*: Relazione tra emozioni suscitate dalla vista di un robot e somiglianza del robot agli umani.

estetico. Nel corso degli anni sono stati sviluppati diversi tipi di approcci per lo studio e la riproduzione della camminata bipede, che verranno menzionati brevemente nelle prossime sezioni.

1.1 Camminata passiva

Si possono dividere gli approcci in varie categorie; la prima distinzione è tra *camminata passiva* ed *attiva*. La camminata passiva consiste in una camminata senza attuatori che sfrutta la pendenza del piano o che richiede energia solo per l'inizio del movimento. Tra i primi modelli ideati c'è il *walker* di T. McGeer [1], composto da una coppia di gambe rigide provviste di un unico grado di libertà, cioè libere di ruotare sull'"anca", dove è posta una massa. Tale robot è modellizzabile come una coppia di pendoli ed è in grado di avanzare in maniera stabile¹ lungo un piano leggermente inclinato senza bisogno di alcun attuatore e sistema di controllo. Un ulteriore sviluppo viene dall'aggiunta di un grado di libertà a metà gamba (ginocchio) [2] sempre senza alcun attuatore (fig: 1.4)

¹cioè il passo ha un ciclo limite stabile

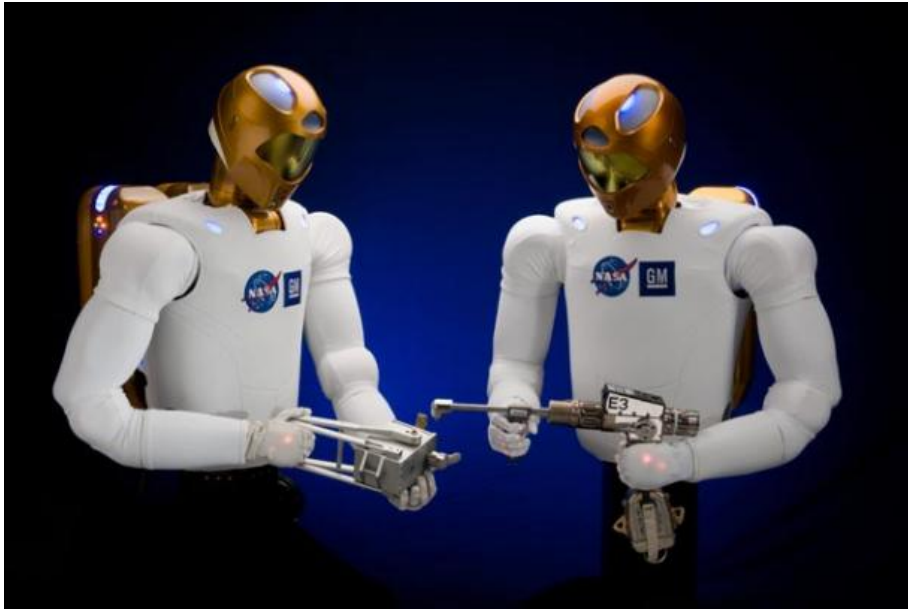


Figura 1.3: *Robonaut 2*, umanoide progettato dalla NASA in collaborazione con General Motors per assistere gli astronauti a bordo della ISS

Dal punto di vista applicativo i *passive walkers* non rivestono un ruolo importante in quanto estremamente limitati: non possono portare pesi, il movimento non può essere controllato o modificato e dipende esclusivamente dal piano su cui il robot cammina.

1.2 Camminata attiva

Al contrario della camminata passiva, la camminata attiva prevede l'uso di attuatori sui gradi di libertà del robot, anche se a volte alcuni gradi di libertà non sono attuati. Il moto delle gambe non avviene più grazie all'energia potenziale, ma dipende dagli attuatori e dal sistema di controllo che ne comanda la traiettoria, che può essere predefinita o calcolata in tempo reale a seconda delle condizioni ambientali rilevate dal robot attraverso i suoi sensori. Le camminate attive si dividono a loro volta in due tipi di camminate: statiche e dinamiche.

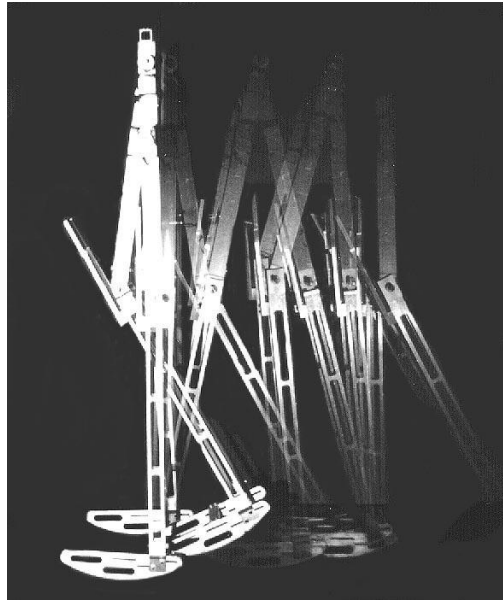


Figura 1.4: *Passive walker* ideato da Tad McGeer

1.2.1 Camminata statica

La caratteristica che contraddistingue la camminata statica è il fatto che il robot, in ogni istante è in equilibrio statico, cioè la proiezione del suo centro di massa (CDM) è interna al poligono di appoggio. In questo modo, fermando i motori in qualsiasi istante, il robot si troverà in una posa di equilibrio.

In questo tipo di approccio si considerano esclusivamente le forze gravitazionali, trascurando le forze di inerzia. Il passo di una camminata statica si compone dall'alternarsi di 2 fasi, doppio supporto (entrambi i piedi appoggiati) e singolo supporto (solo un piede appoggiato), generalmente secondo il seguente schema:

- Doppio supporto: in questa fase il robot poggia su entrambi i piedi.
- Spostamento del baricentro: il peso viene spostato sul piede che farà da appoggio, per permettere il sollevamento dell'altro piede.
- Singolo supporto: il piede libero viene alzato e spostato in avanti.

- Doppio supporto: il piede viene riappoggiato ritornando alla situazione iniziale

In ciascuna di queste fasi il baricentro deve trovarsi all'interno del poligono di appoggio, che in ogni fase varia a seconda del piede appoggiato a terra. Si consideri un robot composto da N link assimilabili a corpi rigidi di massa m_i . Il baricentro del robot coordinate $CDM = [c_x \ c_y \ c_z]$:

$$c_x = \frac{\sum_1^N m_i l_{ix}}{\sum_1^N m_i} \quad (1.1)$$

$$c_y = \frac{\sum_1^N m_i l_{iy}}{\sum_1^N m_i} \quad (1.2)$$

$$c_z = \frac{\sum_1^N m_i l_{iz}}{\sum_1^N m_i} \quad (1.3)$$

con $l_i = [l_{ix} \ l_{iy} \ l_{iz}]$ centro di massa del link i -esimo. La proiezione sul piano di appoggio del centro di massa sarà dunque (c_x, c_y) e dovrà essere posizionata all'interno del poligono convesso rappresentante l'area di appoggio. Per rendere più semplice il posizionamento del baricentro all'interno del poligono i robot progettati per effettuare camminate statiche hanno tipicamente piedi molto ampi. Dal momento che per la camminata statica si considerano solamente le forze gravitazionali trascurando quelle d'inerzia questo tipo di camminata è considerevolmente più lenta di altri tipi in quanto gli effetti dinamici non devono essere di entità tale da minare l'equilibrio del robot.

1.2.2 Camminata dinamica

La camminata statica rappresenta la naturale evoluzione della camminata dinamica: anziché considerare solamente la posizione del baricentro, nella camminata dinamica si considerano gli effetti dinamici dei movimenti del robot. Nella camminata dinamica giocano un ruolo di primaria importanza due punti, la *centro di pressione* (CoP) e lo *zero moment point* (ZMP).

Lo ZMP è definito come il punto all'interno del poligono di appoggio (frontiera esclusa) tale che la somma dei momenti di tutte le forze agenti sul robot (forza gravitazionale, d'inerzia e forze esterne al robot) sia nulla.

La reazione vincolare del piano genera sul piede una pressione, che divisa per l'area di appoggio si traduce in una forza F_p . Il CoP è definito come il punto in cui, applicando la forza F_p , si genera un momento pari al momento generato dalla pressione del piano di appoggio sul piede. Il CoP rappresenta dunque il punto del piede in cui si può considerare applicata tutta la reazione vincolare ed esiste quando il piede tocca terra; quando lo ZMP esiste, esso coincide con il CoP [3].

Per mantenere il robot in equilibrio dinamico è dunque necessario calcolare dove sarebbe lo ZMP se la superficie di appoggio fosse sufficientemente ampia; questo punto è chiamato *Fictitious ZMP*. Si chiamino \mathbf{M}_c il momento del robot attorno al centro di massa \mathbf{r} , \mathbf{f} la reazione vincolare del piano e \mathbf{T} il momento angolare generato dalla reazione vincolare attorno al punto \mathbf{p} . Per ottenere l'equilibrio dinamico si impone che le risultanti delle forze e dei momenti siano nulle:

$$m(\mathbf{r} - \mathbf{p}) \times (\ddot{\mathbf{r}} + \mathbf{g} + \mathbf{M}_c - \mathbf{T}) = 0 \quad (1.4)$$

$$\mathbf{f} = m(\ddot{\mathbf{r}} + \mathbf{g}) \quad (1.5)$$

Lo FZMP è la proiezione sul piano di quel punto \mathbf{p} che soddisfa le precedenti equazioni ponendo $\mathbf{T} = [0 \ 0 \ T_z]$; cioè il punto in cui se fosse possibile applicare la forza corrispondente alla reazione vincolare renderebbe la somma dei momenti nulla. La terza componente è diversa da zero in quanto si presuppone che l'attrito del piede sul piano sia sempre sufficiente per impedire rotazioni sull'asse Z. Per mantenere il robot in equilibrio nel caso di non esistenza dello ZMP è dunque necessario posizionare il piede in modo tale che il poligono di appoggio contenga al suo interno lo FZMP.

Per effettuare una camminata dinamica è dunque necessario avere una strumentazione che consenta di calcolare lo ZMP e lo FZMP. Questo comprende:

- Sensori di pressione sul piede (almeno tre per ogni piede, considerando il piede rigido) che consentono di misurare il CoP (e di conseguenza lo ZMP se esiste) e l'intensità della reazione vincolare
- Trasduttori per la posizione di ogni giunto, in modo da poter calcolare la posizione del centro di massa per ogni link
- Sensore inerziale, per poter conoscere l'orientazione del busto nello spazio
- Sufficiente capacità di calcolo, per poter risolvere le equazioni ad ogni ciclo del sistema di controllo

La camminata dinamica rappresenta al momento attuale il tipo di camminata più efficiente e veloce ed è utilizzata su molti robot e sulla maggioranza dei robot più avanzati, quali ad esempio Asimo della Honda.

1.3 Virtual Model Control

La camminata dinamica attraverso lo ZMP è una tecnica consolidata e con un forte background teorico, ma può essere difficile da mettere in pratica, in quanto si basa su diversi assunti non sempre verificati (piano con attrito infinito) e necessita di una profonda conoscenza del robot e della sua dinamica. Perciò spesso vengono utilizzati metodi basati sull'intuizione per creare degli algoritmi efficienti. Il *Virtual Model Control* [4] rappresenta una tecnica per realizzare degli algoritmi di camminata attraverso l'utilizzo di intuitive metafore di vincoli fisici. Si tratta cioè di supporre di applicare al *walker* degli immaginari oggetti fisici (collegandoli tra due link del robot o tra un link del robot e un riferimento esterno) per muovere il robot nel modo desiderato. Ad esempio, con riferimento alla figura 1.5, si può collegare il robot con una molla ad una ipotetica guida orizzontale per mantenere il busto ad una determinata distanza da terra opponendosi a scostamenti della posizione di riposo; mentre un'altra molla mantiene il busto in posizione verticale opponendosi a rotazioni. Il sistema di controllo del robot analizza posizione e velocità e calcola le forze che agirebbero sui vari link del robot stesso se esistessero

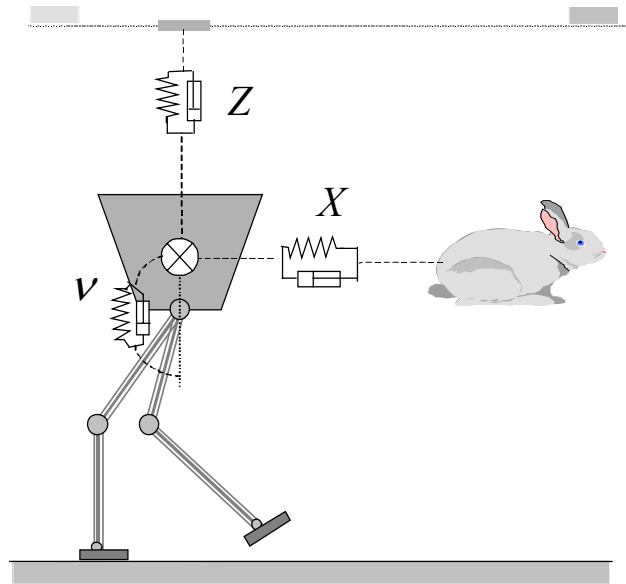


Figura 1.5: Esempio di *virtual model control* applicato ad un robot bipede con 7 membri, tratto dall'articolo [4]

realmente gli oggetti fisici immaginati. Una volta determinate le forze agenti sui vari link del robot, attraverso i calcoli di dinamica inversa vengono determinati momenti che i giunti devono sviluppare per esercitare le medesime forze sui link, ottenendo idealmente lo stesso effetto degli oggetti fisici. Gli oggetti fisici possono anche essere applicati solo in determinate situazioni, oppure a seconda dello stato del robot: in [4] viene utilizzata una macchina a stati finiti (fig 1.6) per determinare quando applicare la determinati oggetti, nel caso specifico la molla che fa avanzare il robot viene collegata nelle fasi di doppio supporto.

Caratteristiche necessarie per questo tipo di controllo sono:

- possibilità di comandare la coppia erogata dai motori (molti robot sono attuati da servomotori che permettono di comandare i giunti specificando unicamente la posizione da raggiungere)
- trasduttori che indichino la posizione dei giunti

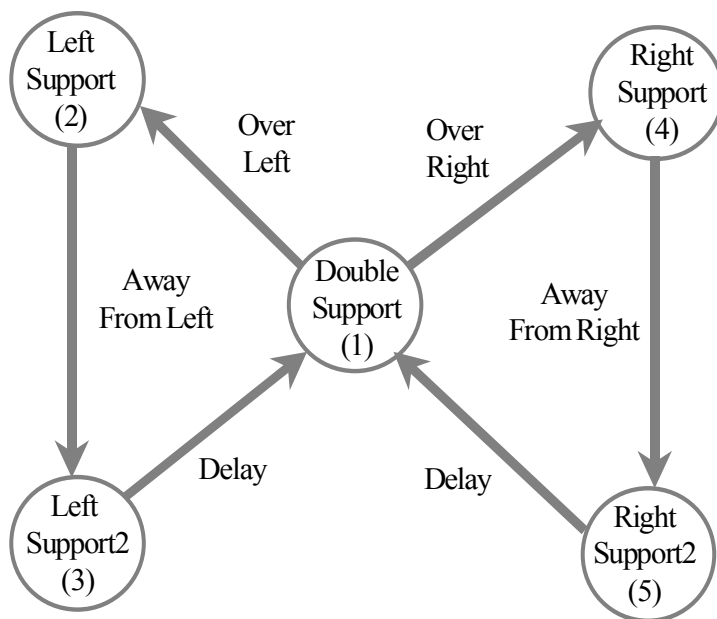


Figura 1.6: Macchina a stati per il virtual model control del robot *Spring Turkey*

Capitolo 2

Apparato Sperimentale

L'apparato sperimentale utilizzato consiste in quattro elementi: il robot, il controller, il software in dotazione e un elaboratore.

2.1 Robovie-X

Il modello di robot utilizzato in questa tesi è lo *Standard Robovie-X (KT-X)* (fig 2.3), prodotto dalla V-Stone. Il robot è di dimensioni $343 \times 180 \times 71$ mm per un peso di 1.3kg. Possiede 17 gradi di libertà, 3 per ogni braccio, 5 per ogni gamba e 1 per la testa. Ogni grado di libertà è attuato da un servomotore analogico modello VS-S092J capace di sviluppare una coppia di 9.03N/m e velocità massima di 9.52rad/s. La scheda di elaborazione on-board è fornita di un cpu a 60MHz (modello H8) con 64kB di ram e 512kB di ROM dove vengono caricati i movimenti e la mappatura del controller. I servomotori vengono controllati dalla scheda attraverso 17 uscite in PWM. Il robot è fornito anche di un altoparlante per riprodurre dei suoni wav anch'essi caricati nella ROM della scheda. La scheda può comunicare con alcuni tipi di controller wireless, tra cui i controller PlayStation2 compatibili; e può venire programmata (mediante il software *RobovieMaker2*) attraverso un'interfaccia USB2.0. Possiede inoltre un selettore a nove posizioni che permette di scegliere tra nove diversi set di movimenti e configurazioni di controller senza dover riprogrammare il robot. Completa l'equipaggiamento del robot



Figura 2.1: Robovie-X, l'umanoide usato in questo lavoro.



Figura 2.2: Scheda di elaborazione del Robovie-X

la scheda opzionale VS-IX001 contenente il sensore inerziale, formato da un accelerometro a tre assi e un giroscopio a due assi (manca l'asse del giroscopio corrispondente all'asse longitudinale del robot). Il sensore inerziale verrà descritto dettagliatamente nella sezione “*Sensore inerziale e filtro*”

Il robot è alimentato da un pacco batterie ricaricabile NiMh 6V composto da 5 celle con capacità di 1600mAh

2.2 Software

Il software fornito a corredo del robot è chiamato *RobovieMaker 2*. È disponibile unicamente per Windows, ma è possibile utilizzarlo all'interno di una *virtual machine*; per questa tesi è stato utilizzato la versione open source di *VirtualBox*. *RobovieMaker 2* Consiste principalmente in due pannelli: nel pannello di sinistra (*Pose Area*) sono presenti delle *slidebar* che consentono di controllare il valore di ogni variabile di giunto, oltre che la luminosità dei led ed altri parametri di minore importanza. Nel pannello di destra (*Motion Area*) c'è uno spazio in cui è possibile programmare dei movimenti attraverso un semplice linguaggio a blocchi, che permette di memorizzare le pose definite nella *Pose Area* e di impostare le transizioni tra le varie pose. È possibile utilizzare anche strutture condizionali (if-then-else) ed effettuare

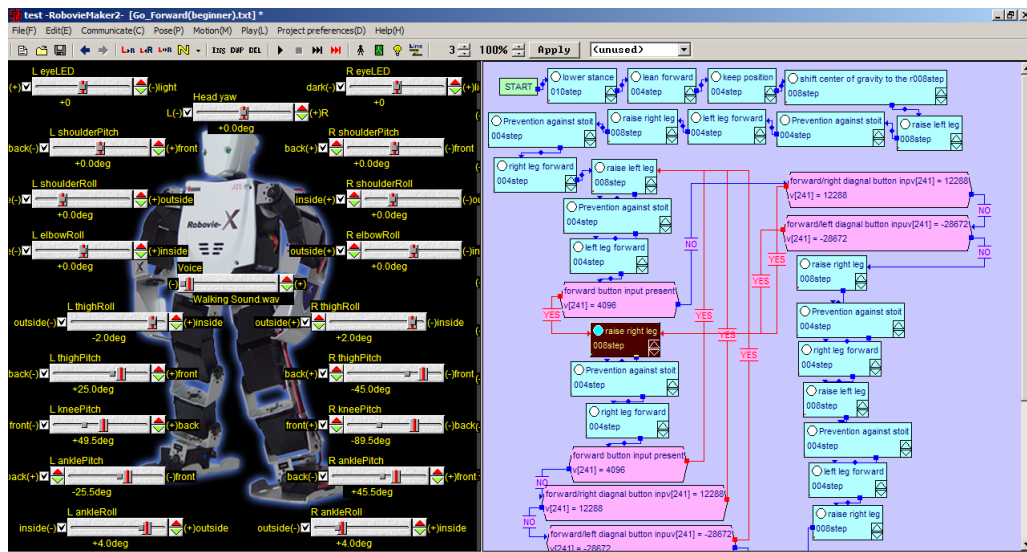


Figura 2.3: Schermata principale del programma *Roboviemaker2*

semplici operazioni tra un numero limitato di variabili. È importante sottolineare come le pose all'interno della *posa area* non possano essere definite in funzione di alcun parametro: l'unico modo di variare la posizione di un giunto è agire attraverso la relativa slidebar. Inoltre non è possibile in alcun modo interrompere, accelerare, variare l'esecuzione di una posa in corso, poiché tutte le operazioni sulle variabili vengono effettuate tra il termine di una posa e la successiva, e dunque non è neppure possibile iniziare un'altra posa o elaborare variabili nel contempo. L'unico modo per variare la posizione del motore al di fuori del programma creato nella *motion area* è rappresentato dalla possibilità di sommare, al comando ciascun motore, una variabile. Questa variabile però sarà la stessa per ciascuna posa, e non è possibile modificarla durante l'esecuzione di una posa ma solo attraverso i blocchi di calcolo nell'intervallo tra il termine di una posa e l'inizio della posa successiva.

Una volta creato il programma di movimento del robot nella *Motion Area* è possibile caricarlo nella memoria del robot ed associare un evento per l'esecuzione di tale programma. L'evento può essere la pressione di un tasto o di una combinazione di tasti sul controller, o la variazione di una variabile interna del robot (ad esempio la variabile che rappresenta il valore letto

dall'accelerometro.)

Durante l'esecuzione di un programma, se il robot è collegato via USB al computer è possibile accedere alla lettura delle variabili interne attraverso il programma.

Capitolo 3

Scelta del tipo di camminata

Il robot ed il software a disposizione impongono alcune limitazioni. Innanzi tutto, l'impossibilità di definire delle pose in funzione di alcuni parametri porta all'impossibilità di progettare una camminata intelligente¹ attraverso il solo linguaggio a blocchi del programma *RobovieMaker 2*; rendendo necessaria la scrittura di un qualche software per la comunicazione con il robot.

3.1 Limitazioni dell'apparato sperimentale

L'apparato sperimentale presenta delle limitazioni di impediscono l'utilizzo di alcuni degli approcci presentati per la camminata di un robot umanoide. Innanzitutto i motori possono essere controllati solamente in posizione, e non in coppia, rendendo impossibile un approccio del tipo “*virtual model control*”. Inoltre, i motori non dispongono di un modo per restituire la posizione istantanea, non permettendo così alla scheda di sapere se la posizione inviata al motore è stata raggiunta o meno² Infine, il robot non dispone di sensori di

¹Cioè una camminata il cui comportamento varia a seconda delle condizioni esterne rilevate dai sensori

²I servomotori analogici presenti effettuano il posizionamento preciso grazie ad un controllo in feedback (PID) interno al servomotore stesso. Non dispongono però di un metodo per comunicare la posizione raggiunta alla scheda: dei tre fili che compongono il cavo del servomotore due sono utilizzati per l'alimentazione mentre l'ultimo è utilizzato per il comando in PWM inviato dalla scheda al motore.

pressione sul piede; questa mancanza rende impossibile misurare il CoP e lo ZMP e renderà necessario fare alcuni assunti, ad esempio presupporre di conoscere quale piede è appoggiato a terra oppure sapere che il piede appoggia con tutta la sua superficie a terra.

Per quanto riguarda la scheda, la V-Stone non fornisce alcun modo di programmarla all'infuori del precedentemente citato *RobovieMaker2* con le limitazioni precedentemente esposte quali la semplicità delle strutture condizionali presenti, l'impossibilità di interrompere o variare una posa, l'impossibilità di definire una posa in modo parametrico. Questo implica la necessità di effettuare le elaborazioni su un computer esterno alla scheda del robot stesso. Purtroppo la V-Stone non ha reso pubblico il protocollo di comunicazione dei suoi robot, rendendo necessario del lavoro di reverse engineering sull'apparato sperimentale. Tale lavoro, che verrà esposto nel prossimo capitolo, si basa significativamente sul lavoro di R. Bonetto [5], che ha effettuato il reverse engineering del software relativo al robot RB2000, e sul codice di A.Casasola e N.Frezzato [6] che, partendo dal lavoro di R. Bonetto hanno scritto un software Windows in C che consente di comandare i motori attraverso il protocollo USB. Il lavoro è consistito principalmente nella realizzazione di un software in C++ per il sistema operativo Linux che implementasse le funzioni già create da Frezzato e Casasola aggiungendo la possibilità di leggere i dati della piattaforma inerziale e di scrivere e leggere le variabili interne della scheda del robot.

Il protocollo USB2.0 non consente la comunicazione Full-Duplex, non permette cioè di trasmettere e ricevere contemporaneamente. In più questo protocollo non garantisce la costanza della velocità di trasmissione dei pacchetti. Questo rende difficoltoso garantire il funzionamento dell'algoritmo della camminata qualora si decidesse di inviare costantemente i valori di ogni singolo motore attraverso la porta USB.

3.2 Suddivisione dell'algoritmo della camminata

Si è posta la necessità di scegliere dove effettuare l'elaborazione dell'algoritmo per effettuare la camminata. Date le limitazioni del protocollo USB2.0 espresse nel paragrafo precedente effettuare tutta l'elaborazione sul PC esterno ed inviare solamente i comandi ai motori del robot avrebbe potuto rivelarsi problematico, specialmente nel caso di una camminata dinamica. D'altra parte, l'impossibilità di effettuare operazioni complesse tramite il linguaggio a blocchi del *RobovieMaker2* e l'impossibilità di programmare la scheda in altro modo rende improponibile l'implementazione dell'algoritmo di una camminata "intelligente" (cioè che tenga conto dei dati rilevati dai sensori inerziali, azioni esterne, ecc) interamente sull'unità di elaborazione del robot. Si ricordi che, siccome è prevista per il futuro l'integrazione di questo lavoro con un "visual gyroscope" attualmente in studio, l'interfacciamento con un elaboratore esterno è indispensabile in quanto la scheda a bordo del robot non è in grado di interfacciarsi con la telecamera, né dispone di una sufficiente potenza computazionale per elaborare delle immagini. L'utilizzo di un elaboratore esterno per l'algoritmo della camminata non rappresenta perciò in questa fase una limitazione, a patto che l'algoritmo sia di complessità contenuta in modo tale da permettere, in futuro, la sua esecuzione in un sistema embedded³ di limitate prestazioni. Dal momento che esistono già delle camminate performanti fornite già dalla V-Stone attraverso il suo software si è deciso di procedere progettando degli algoritmi di "correzione" della camminata. Si è deciso cioè di programmare il robot per effettuare una camminata deterministica, cioè programmata staticamente sulla scheda del Robovie-X ed elaborarne le correzioni dinamiche in risposta a stimoli ambientali su un PC esterno, comunicante via USB con la scheda del robot. L'elaboratore esterno avrà accesso esclusivamente alle variabili interne della scheda, e non potrà comandare direttamente i motori, ma solo impostarne degli offset attraverso la scrittura di alcuni variabili interne al robot. Gli

³Un sistema di dimensioni molto ridotte, che possa essere montato a bordo del robot

algoritmi di correzione idealmente dovranno essere quanto più possibile indipendenti dal robot o dalla camminata implementata, in modo da richiedere poche operazioni per essere utilizzati su altri robot o in seguito a modifiche della camminata stessa.

I vantaggi di tale approccio sono i seguenti:

- possibilità di utilizzare camminate progettate specificamente per il robot utilizzato
- maggiore robustezza a ritardi ed errori di comunicazione
- facilità di integrazione con sistemi di visione computazione in futuri sviluppi

Le limitazioni di tale approccio sono invece:

- Limitatezza delle azioni intraprendibili (non potendo ad esempio interrompere un movimento in corso)
- Necessità di collegamento fisico tra PC e scheda di elaborazione (aggiungibile in futuro con sistema embedded a bordo)

Capitolo 4

Protocollo USB

La prima fase del lavoro è stata progettare una infrastruttura di comunicazione tra la scheda del robot e l'elaboratore esterno collegato via USB. Dal momento che l'azienda produttrice non fornisce alcuna specifica del protocollo di comunicazione tra il software e la scheda on-board è necessario prima di tutto comprendere il protocollo di comunicazione attraverso tecniche di *reverse engineering*. Gran parte del lavoro è stato effettuato da R. Bonetto [5], che ha effettuato il *reverse engineering* del software *RobovieMaker2* e lo *sniffing* dei pacchetti USB, comprendendo la struttura base del pacchetto e i metodi per il comando dei motori del robot RB2000, anch'esso prodotto dalla V-Stone. Partendo da questo lavoro, N. Frezzato e A.Casasola [6] hanno adattato implementato in linguaggio C i metodi necessari al comando dei motori del robot Robovie-X. Date le librerie utilizzate, il software creato era disponibile unicamente per Windows.

Agli scopi di questo lavoro, rimanevano da implementare alcune funzioni:

- Lettura giroscopi ed accelerometri
- Lettura variabili interne robot
- Scrittura su variabili interne robot

Oltre a continuare il reverse engineering del protocollo USB per scoprire come effettuare ed implementare tali operazioni, si è voluto re-implementare le funzioni presenti in C++, utilizzando librerie che permettessero di far girare

il software in ambiente Linux. Per la cattura dei pacchetti USB è stato usato il software USBTrace; le operazioni di cattura ed analisi sono state effettuate in collaborazione con Nicola Carlon. L'implementazione in C++ del software per Linux è stata sviluppata a partire da un'implementazione di Fabio Della Libera e ne eredita alcune funzioni.

4.1 Struttura del pacchetto USB

Viene riportata ora la descrizione della struttura comune ad ogni pacchetto, determinata da R. Bonetto. Ogni pacchetto USB, sia in ingresso che in uscita, è formato da 64 byte¹, organizzati in 32 Word. Ogni coppia rappresenta un valore, con la convenzione *Little Endian*, cioè con il byte più significativo trasmesso dopo il byte meno significativo. In risposta ad un singolo pacchetto inviato dal PC si possono ricevere uno o tre pacchetti. Tutti i pacchetti hanno la seguente struttura, indipendentemente dalla sua funzione:

- byte 00: **0x55**
- byte 01: rappresenta la quantità di pacchetti inviati/ricevuti consecutivamente. La prima cifra è l'indice del pacchetto corrente, la seconda è la quantità totale di pacchetti. Ad esempio, se il byte 01 vale "13", vuol dire che il pacchetto corrente è il secondo di tre pacchetti (il precedente avrebbe il byte 01 posto a "03" ed il successivo a "23")
- byte 02, 03, 05: rappresentano un codice che identifica il tipo di trasmissione.
- byte 04: solitamente usato quando il pacchetto tratta di un indice.

4.2 Lettura e scrittura delle variabili interne

La struttura di memoria della scheda di elaborazione del robot è costituita da 256 variabili, ciascuna indicata da un indice da 1 a 256. Molte di queste

¹numerati da 0 a 63

sono riservate per l'utilizzo esclusivo del robot, altre sono disponibili esclusivamente in lettura o in scrittura. Sono stati determinati dei range di alcune variabili:

- Le variabili da 66 a 128 sono disponibili in lettura e scrittura per l'utente
- Le variabili 129, 130 e 131 rappresentano il valore degli accelerometri (nell'ordine X, Y, Z) e sono di sola lettura
- Le variabili 132 e 133 rappresentano il valore dei giroscopi X e Y e sono di sola lettura
- Le variabili immediatamente successive sono di sola lettura, probabilmente per schede diverse dalla VS-IX001 che ospitano più sensori

Non sono stati indagati i significati di altri range di variabili.

4.2.1 Scrittura variabile interna

Per la scrittura di una generica variabile R/W si invia un pacchetto USB al robot così strutturato:

- byte 00: **0x55**
- byte 01: **0x01** (singolo pacchetto)
- byte 02, 03, 05: **0xE0,0x08,0x02** (codice della richiesta)
- byte 04: Indice della variabile moltiplicato per 2
- byte 06: Parte bassa (8 bit meno significative) del valore da assegnare alla variabile
- byte 07: Parte alta (8 bit più significativi) del valore da assegnare alla variabile

Tutti i byte non specificati possono essere posti a zero.

4.2.2 Lettura di variabile interna

Per effettuare la lettura di una variabile interna R/W è necessario inviare un apposito pacchetto e poi monitorare in pacchetti in entrata per leggere la risposta inviata dalla scheda del robot. Il pacchetto di richiesta di lettura è così composto:

- byte 00: **0x55**
- byte 01: **0x01** (singolo pacchetto)
- byte 02, 03, 05: **0xE0,0x08,0x02** (codice della richiesta)
- byte 06: Indice della variabile moltiplicato per 2

Tutti i byte non specificati possono essere posti a zero. Una volta inviato tale pacchetto, il robot risponderà inviando un pacchetto contenente il valore richiesto; è quindi necessario leggere i pacchetti in entrata fintantoché non si riceverà il pacchetto di risposta, che verrà riconosciuto analizzando i primi 4 byte:

- byte 01: **0x01** (singolo pacchetto)
- byte 02,03: **0xE0,0x08**
- byte 04: Indice della variabile (moltiplicato per 2) di cui era stata richiesta la lettura
- byte 06: Valore della variabile, 8 bit meno significativi
- byte 07: Valore della variabile, 8 bit più significativi

4.2.3 Lettura dei sensori inerziali

La lettura dei sensori inerziali (indici da 129 a 133) viene fatta in maniera diversa dalle variabili R/W con indirizzo da 65 a 128. Infatti utilizzando il precedente tipo di pacchetti che utilizza il solo byte 06 per specificare l'indice della variabile da leggere moltiplicato per due non è possibile accedere ad indirizzi superiori a 127 (in quanto $127 \cdot 2 = 00xFE$, mentre $128 \cdot 2 = 0x100$ e

necessiterebbe di due byte). Si è ipotizzato che questa differenza del metodo di lettura sia dovuta al fatto che le variabili successive alla 127 siano da considerarsi, nelle intenzioni dei programmatori, variabili riservate al sistema e quindi non normalmente accessibili agli utenti. Sempre mediante il medesimo procedimento di cattura dei pacchetti USB, si è scoperto che il software *RobovieMaker 2* accede a queste cinque variabili contemporaneamente inviando al robot un pacchetto così strutturato:

- byte 00: **0x55**
- byte 01: **0x01** (singolo pacchetto)
- byte da 02 a 12: **0xE0,0x00,0x10,0x04,0x00,0x0B,0x20,0x00,0x8A,0xEC,0x06** (codice della richiesta)
- byte 13: **0x09** indice del pacchetto, la risposta conterrà questo indice per identificare il pacchetto
- byte 14,15,16: **0xE9,0X02,0X0A**

il resto dei byte possono essere posti a zero. Una volta spedito tale pacchetto viene immediatamente spedito il pacchetto id risposta, contenente:

- byte 04,05: Accelerometro, asse X
- byte 06,07: Accelerometro, asse Y
- byte 08,09: Accelerometro, asse Z
- byte 10,11: Giroscopio, asse X
- byte 12,13: Giroscopio, asse Y

Ciascun valore è da interpretare con la convenzione Little Endian come per i valori precedentemente mostrati

Capitolo 5

Cinematica di posizione

Per far effettuare qualsiasi azione al robot è necessario sapere dove sono posizionati i vari link che lo compongono e quali siano le configurazioni dei giunti necessarie per posizionare ed orientare ogni link nel modo voluto. È cioè necessario conoscere il legame tra le variabili di giunto e la posizione di ogni link del robot. Nelle applicazioni di robotica industriale, tipicamente interessa conoscere il legame tra le variabili di giunto e la posizione e orientamento dell'ultimo link di un braccio (chiamato hand-effector) rispetto ad un sistema di riferimento solidale con l'ambiente (riferimento "world") solitamente posta vicino al punto di fissaggio del braccio robotico stesso. Questo concetto è spesso valido anche nelle applicazioni di robotica umanoide, considerando come hand-effector la mano del robot rispetto al braccio oppure il piede rispetto alla gamba, e ponendo il sistema di riferimento (ad esempio) sul busto. Per progettare una camminata è necessario soprattutto sapere dove si trovano i piedi rispetto al baricentro ed essere in grado di posizionarli ed orientarli nel punto voluto. Nelle prossime sezioni verrà perciò analizzata la cinematica di posizione inversa della gamba del Robovie-X.

5.1 La cinematica inversa

La cinematica inversa consiste nel determinare i valori delle variabili di giunto che posizionano e orientano in un certo modo l'hand-effector. La cinematica-

ca inversa di posizione non è un problema sempre analiticamente risolvibile; cioè non di tutti i bracci robotici sono conosciute le relazioni tra la posizione dell'ultimo link e le variabili di giunto, rendendo necessari approcci alternativi, quali la ricerca della soluzione per via numerica (ad esempio approssimando la relazione tra giunti e posizione dell'hand-effector con il suo sviluppo di Taylor [7] e cercando il punto stabilito mediante il metodo di Newton-Raphson) o l'approccio mediante "teaching box", dove un operatore muove il giunto nelle posizioni prestabilite registrando i valori delle variabili di giunto. Entrambi i metodi presentano notevoli svantaggi, allungando sensibilmente i tempi di calcolo oppure richiedendo un addestramento specifico ad ogni cambio di operazioni da svolgere; tutto ciò rende molto desiderabile poter calcolare le variabili di giunto attraverso un procedimento analitico

Piper ha enunciato un teorema che dà una condizione sufficiente affinché si possa effettuare la cinematica inversa analiticamente [?]: quando un robot presenta 6 assi, di cui 3 concorrenti in un punto. In questo tipo di bracci robotici, si ha una "separazione" delle variabili di giunto: l'orientazione dell'hand-effector è data dai valori dei giunti dei 3 assi concorrenti (che formano così un polso sferico), mentre il posizionamento è ottenuto mediante gli altri 3 giunti.

Il braccio robotico che si vuole analizzare è la gamba del Robovie-X. Essa è articolata da cinque assi di rotazione, due giacenti sul piano sagittale e tre sul piano frontale. Non avendo assi di rotazione giacenti sul piano trasversale, l'orientazione del piede perde un grado di libertà: non è cioè possibile ruotare il piede sull'asse Z del robot. Date le caratteristiche della gamba (cinque assi, di cui gli ultimi due non sono concorrenti) non vale la condizione sufficiente precedentemente enunciata, ma in realtà è possibile applicare un procedimento molto simile, separando le ultime due variabili, che determineranno l'orientamento del piede, dalle tre precedenti, che ne determineranno la posizione nello spazio¹.

La posizione e l'orientamento del piede sono dunque descrivibili da 5 variabili:

¹Ferma restando la limitazione sulle possibili orientazioni del piede a causa dell'impossibilità di rotazione su un asse trasversale al robot.

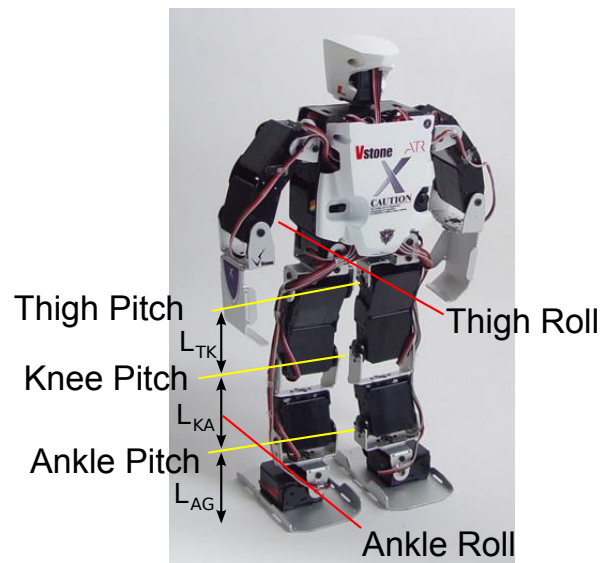


Figura 5.1: Assi di rotazione della gamba del Robovie-X

1. dx : scostamento del piede rispetto al giunto “thigh roll” lungo l’asse X;
2. dy : scostamento del piede rispetto al giunto “thigh roll” lungo l’asse Y;
3. dz : scostamento del piede rispetto al giunto “thigh roll” lungo l’asse Z;
4. α : rotazione del piede lungo l’asse X del robot;
5. β : rotazione del piede lungo l’asse Y del robot.

Per effettuare i calcoli relativi alla cinematica inversa è necessario conoscere la distanza tra i vari assi. Con riferimento alla figura 5.1:

1. $\overline{TrollT} = 34mm$
2. $\overline{TK} = 62mm$
3. $\overline{KA} = 60mm$
4. $\overline{AG} = 48mm$

Verrà trascurata la distanza tra l’asse “ankle roll” ed il piano, supponendo così che il robot poggi con un segmento dell’asse a terra.

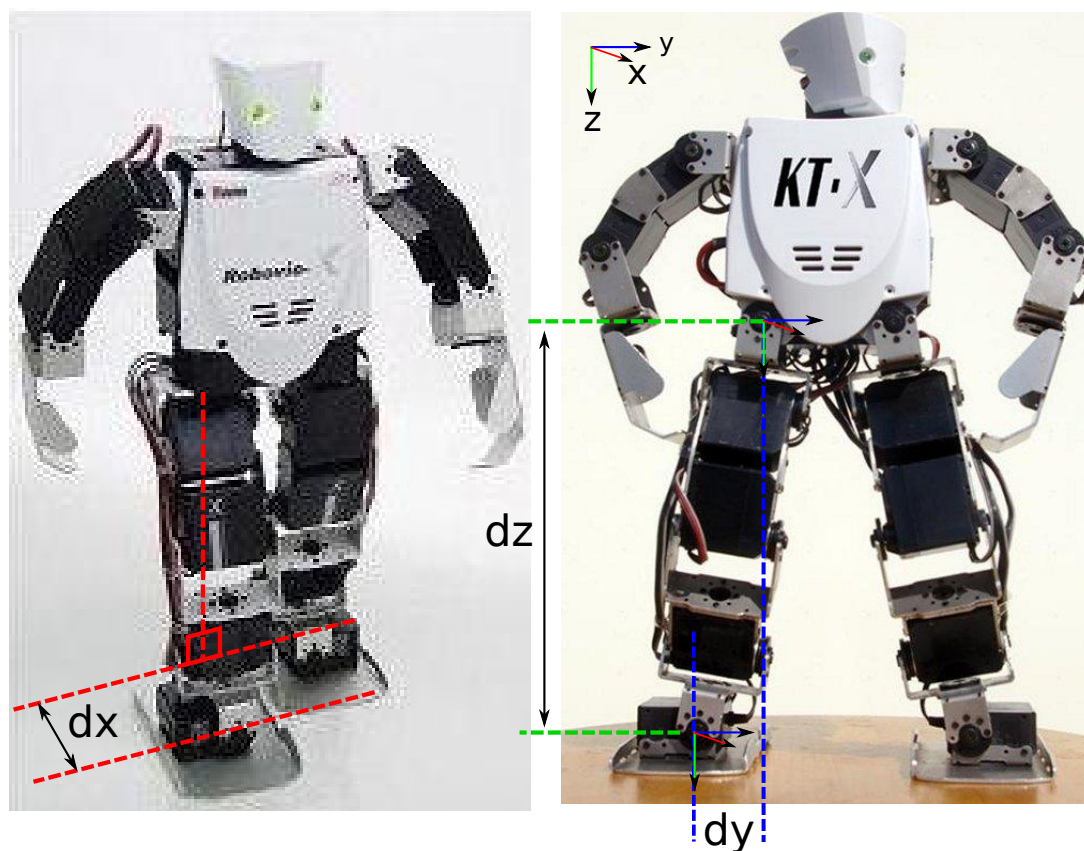


Figura 5.2: Traslazione del piede

Sia L la distanza tra il piede e il punto centrale del segmento d'appoggio, che è data da:

$$L = \sqrt{dx^2 + dy^2 + dz^2}$$

5.2 Piano Frontale

Si consideri ora il piano frontale; si vogliono determinare i valori dei giunti “thigh roll” e “ankle roll”. L'unico grado di libertà che permette di traslare il piede lungo l'asse Y è il giunto “thigh roll”. Per prima cosa è necessario dunque determinare il valore di tale giunto che trasla il piede lungo Y della quantità desiderata dy . La lunghezza della proiezione nel piano frontale della

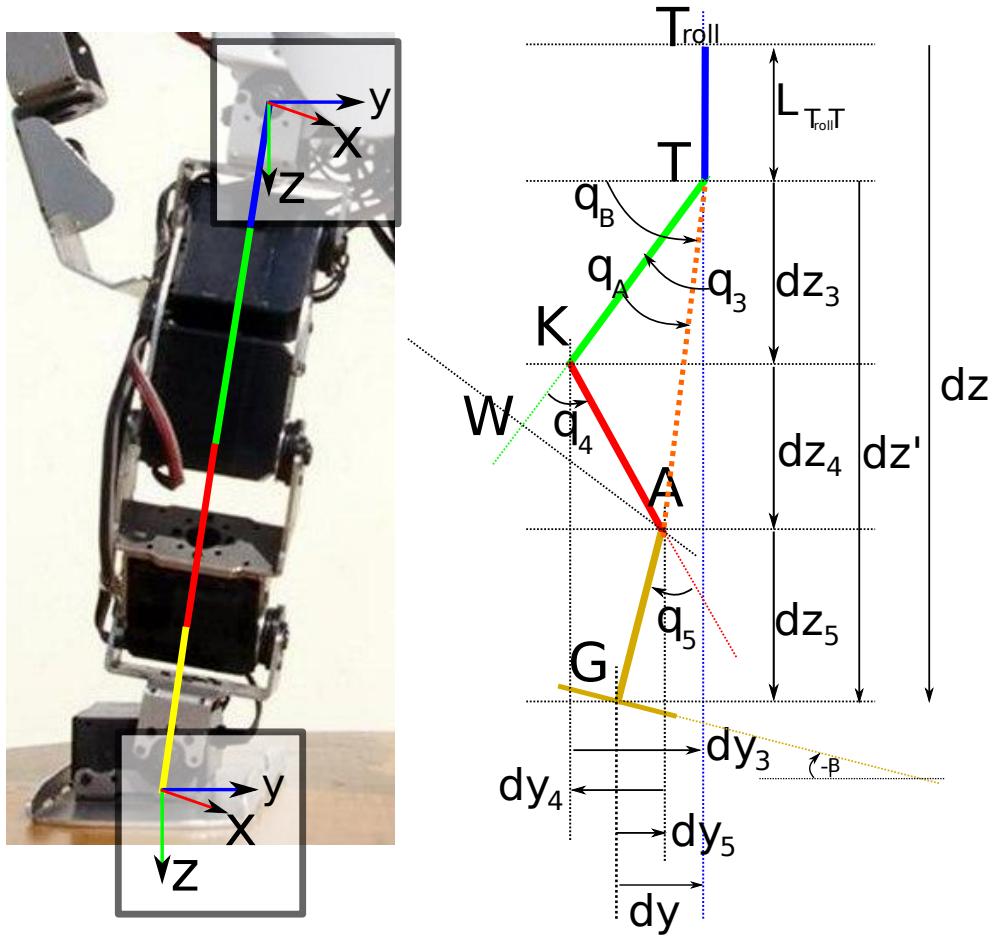


Figura 5.3: Cinematica inversa della gamba: a sinistra, schema dei link sul piano frontale; a destra sul piano sagittale

gamba è:

$$L_{front} = \sqrt{dy^2 + dz^2}$$

Per ottenere uno spostamento dy è dunque necessario ruotare il giunto “thigh roll” di un angolo q_1 tale che: $dy = L_{front} \cdot \sin(q_1)$, dunque:

$$q_1 = \arcsin(dy/L_{front})$$

Nel software *RobovieMaker 2* si può vedere che, relativamente alla gamba sinistra, angoli positivi ruotano la gamba verso l'interno. Dunque a dy positivi (spostamento della gamba sinistra verso l'interno) corrispondono angoli di q_1 positivi. Per la gamba destra, angoli positivi ruotano la gamba verso l'esterno, quindi a dy positivi corrispondono si nuovo angoli di q_1 positivi.

Una volta calcolato q_1 , sappiamo che la terna di riferimento del piede è ruotata lungo l'asse X di un angolo q_1 . Se vogliamo portare la terna ad un angolo α rispetto alla terna di riferimento del busto è necessario ruotare il piede lungo l'asse X di un angolo:

$$q_2 = \alpha - q_1$$

In questo caso il verso di rotazione è opposto: per questioni di ingombro il motore della caviglia è montato ribaltato rispetto al motore dell'anca e quindi (relativamente alla gamba sinistra) per angoli positivi sposta il piede verso l'esterno, contrariamente all'anca.

5.3 Piano Sagittale

Analizzando ora il piano sagittale si determineranno i rimanenti gradi di libertà, relativi agli assi “pitch”. Per prima cosa, si impone la posizione e la rotazione del piede lungo l'asse Y e si calcola la posizione del punto A.s

Ruotando il piede di un angolo β il punto A viene a trovarsi nelle coordi-

nate:

$$\begin{cases} A_x = & dx \\ A_y = & -dy + dy_5 = -dy + L_{AG} \cdot \sin(\beta) \\ A_z = & -dz + dz_5 = -dz + L_{AG} \cdot \cos(\beta) \end{cases}$$

Una volta determinato il punto A, dobbiamo trovare q_3 e q_4 tali che l'estremo del secondo link vada a coincidere con A. Passiamo in coordinate polari sul piano su cui giace la gamba: A si trova ad una distanza \overline{TA} , e ad un angolo ϕ .

Determiniamo la distanza \overline{TA} :

$$\overline{TA} = \sqrt{(-dy - dy_5)^2 + (dz' - dz_5)^2 + dx^2}$$

con $dz' = -dz + \overline{T_{roll}T}$

Avendo precedentemente fissato q_1 , questa distanza dipende esclusivamente dal valore dell'angolo q_4 ; che può dunque essere determinato attraverso il teorema del coseno:

$$\begin{aligned} \overline{TA} &= \overline{TK}^2 + \overline{KA}^2 - 2 \overline{TA} \overline{TK} \cos(180 - q_4) \Rightarrow \\ \Rightarrow q_4 &= \pm \arccos \left(\frac{\overline{TA}^2 - \overline{TK}^2 - \overline{KA}^2}{2 \overline{TK} \overline{KA}} \right) \quad [\cos(q_4) = -\cos(180 - q_4)] \end{aligned}$$

Da questa formula si ottengono due soluzioni poiché ci sono due modi per ottenere una gamba di una certa lunghezza: piegando il ginocchio in avanti, come in una gamba umana, o con il ginocchio indietro. In questo caso scegliamo la configurazione "ginocchio avanti" che è quella (secondo la convenzione RobovieMaker2 per la gamba sinistra) con $q_4 > 0$

Fissata la lunghezza \overline{TA} attraverso q_4 , l'estremo del secondo link è libero di muoversi su una circonferenza di raggio \overline{TA} passante per A. Determiniamo il valore dell'angolo q_B per il quale il secondo estremo si sovrappone ad A:

$$q_B = \arctan2(-TA_z, -TA_y)$$

Dove atan2 è un funzione a due argomenti (presente sia in matlab che nella libreria math di C) che restituisce l'angolo il cui seno è il primo argomento e il coseno il secondo. Allo stesso modo determiniamo q_A

$$q_A = \text{atan2}(\overline{KA} \sin(q_4), \overline{TK} + \overline{KA} \cos(q_4))$$

Ponendo attenzione ai segni, determiniamo il valore dell'angolo q_3 :

$$90 - q_3 = q_B - q_a \Rightarrow q_3 = 90 - (q_B - q_A)$$

Infine, determiniamo l'angolo q_5 :

$$-\beta = q_3 - q_4 + q_5 \Rightarrow q_5 = -\beta - q_3 + q_4$$

Per la gamba destra, essendo i motori di "pitch" montati specularmente, i valori di q_3, q_4, q_5 , vanno presi di segno opposto.

Capitolo 6

Sensore inerziale e filtro

Per poter effettuare una stabilizzazione della camminata di un robot umanoide è necessario uno strumento che possa misurarne gli spostamenti e l'orientazione nello spazio. Il Robovie-X è equipaggiato con una unità di misura inerziale (IMU) fornita dalla V-Stone completa di connettore da collegare alla scheda di elaborazione del robot. Questa IMU è composta da un ac-

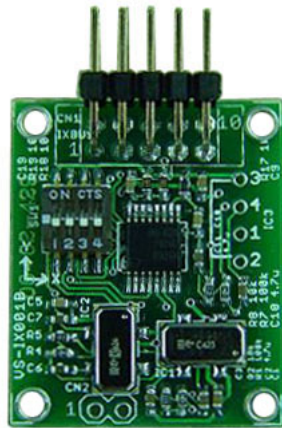


Figura 6.1: Scheda VS-IX001 in dotazione al Robovie-X

celerometro a 3 assi e un giroscopio a due assi; manca dunque un metodo per misurare la rotazione attorno all'asse Z dell'umanoide. La scheda di elaborazione dell'umanoide copia i valori in uscita dagli accelerometri e giroscopi in cinque variabili di sistema; nell'organizzazione della memoria del

robot queste corrispondono alle variabili accessibili a partire dall'indirizzo 129. Attraverso il cavo USB è possibile far leggere queste cinque variabili al programma che implementa l'algoritmo di stabilizzazione. È importante notare come il protocollo USB adottato per la comunicazione scheda-PC non permetta di avere timing affidabili, poiché il protocollo USB non è gestito direttamente dal programma ma dal sistema operativo su cui gira. Inoltre, il protocollo USB fino alla versione 2.0 è solamente half-duplex: questo implica la necessità di sospendere le operazioni di lettura dei sensori quando è necessario trasmettere dei comandi ai motori, e viceversa. Si rivela dunque necessario realizzare un filtro per ottenere l'orientazione del robot a partire dai dati grezzi che sia il più possibile robusto alle imprecisioni degli istanti di acquisizioni dei dati. La connessione USB e la IMU hanno dimostrato prestazioni sufficienti per acquisire campioni fino alla velocità di 240Hz. Bisogna però tener conto dell'obiettivo secondario della tesi, che è quello di realizzare una stabilizzazione della camminata che possa coadiuvare o sostituire la IMU con un "visual gyroscope". Questo tipo di giroscopi realizzati con telecamere e tecniche di visione computazionale hanno tipicamente framerate non superiori a 15Hz, e diventa quindi utile realizzare un filtro che sia efficace anche per basse velocità di campionamento. Altra caratteristica desiderabile è che il filtraggio sia computazionalmente leggero. Nello scenario di questa tesi l'elaborazione dei dati grezzi dell'IMU è effettuata da un elaboratore esterno al robot che ha risorse di calcolo in sovrabbondanza per l'elaborazione di tali dati; in futuro però si desidera equipaggiarlo con una scheda di calcolo per gestire in completa autonomia e senza l'ausilio di un elaboratore esterno tutte le operazioni necessarie e questa caratteristica diventa di primaria importanza come in tutti i dispositivi di piccole dimensioni dove risorse di calcolo e energia elettrica sono risorse scarse.

Riassumendo, le caratteristiche desiderabili nell'algoritmo di filtraggio dei dati in uscita dall'IMU sono:

- Robustezza alla variazione degli intervalli di campionamento dovute alle limitazioni del protocollo USB
- Leggerezza computazionale

- Facilmente integrabile con i dati di un eventuale giroscopio visivo

6.1 Filtro per sensori inerziali e inerziali/magnetici

Dopo aver analizzato vari filtri disponibili in letterature, la scelta è ricaduta sul filtro e sull'implementazione descritta nell'articolo [8]. Questo filtro fa uso della rappresentazione mediante quaternioni per fondere i valori forniti dall'accelerometro con i valori forniti dal giroscopio ed eventualmente da una bussola. Dal momento che i giroscopi forniscono la posizione angolare tramite l'integrazione di un segnale rappresentante la velocità angolare, un piccolo bias di quest'ultimo si traduce in un costante *drift* del segnale della posizione ed è quindi importante poter stimare questo fenomeno od eliminarne gli effetti. Anche utilizzando una IMU con accelerometro e giroscopio a 3 assi non è possibile calcolare univocamente l'orientazione del robot: in primo luogo manca un riferimento assoluto per l'orientazione iniziale sull'asse Z , in quanto i giroscopi per fornire l'orientazione richiedono una posizione iniziale nota e la direzione del campo gravitazionale è parallela all'asse Z e dunque non è possibile utilizzare la misura degli accelerometri per inizializzare i giroscopi. In secondo luogo, anche ammettendo di imporre a priori un'orientazione iniziale del robot nota, i giroscopi per la loro natura sono soggetti a drift e dunque è necessario "correggere" la loro misura con i dati degli accelerometri. Come prima però, gli accelerometri non possono rilevare in alcun modo rotazioni su un asse parallelo al campo gravitazionale, rendendo così impossibile correggere la stima dell'orientazione lungo l'asse Z . Per questo, Madgwick propone anche un'implementazione che utilizza una IMU in combinazione con un sensore magnetico (bussola digitale): questo tipo di sensore fornisce una stima della rotazione sull'asse mancante il cui errore non dipende dal tempo, permettendo così di correggere tutti i bias dei giroscopi. Il robot impiegato in questa tesi non dispone però del giroscopio sull'asse Z , e non dispone del sensore magnetico; risultando così incapace di determinare rotazioni attorno a quest'asse. Questo non è una significativa limitazione nel

caso della stabilizzazione della camminata rettilinea, in quanto raramente si verificano rotazioni significative su quest'asse tali da pregiudicare la camminata del robot; al contempo però non è possibile correggere variazioni di rotta che naturalmente occorrono a causa di irregolarità del terreno. Ricordando che l'obiettivo secondario è sviluppare un algoritmo che possa essere coadiuvato da un visual gyroscope, una possibile soluzione è utilizzare l'output di per sostituire la bussola e l'asse di giroscopio mancante. Caratteristiche dei dati in uscita da un giroscopio di questo tipo sono:

- Basso framerate ($<20\text{fps}$)
- Nessun drift (in quanto i dati non sono risultato di integrazioni come nei giroscopi tradizionali)
- Precisione dipendente da tipo di ambiente

Al momento attuale è disponibile un'implementazione di giroscopio visivo implementata da M.Finotto [9], che ha buone prestazioni in un ambiente appositamente creato (le pareti sono formate da pannelli con righe verticali ed orizzontali); e si stanno svolgendo ricerche possibili miglioramenti di tale strumento orientati soprattutto al funzionamento in ambienti non appositamente predisposti. Le misure ottenute da un giroscopio visivo possono essere integrate nella stima dell'angolazione in vari modi a seconda della precisione prevista. Se il giroscopio visivo ha una precisione bassa può essere utilizzato per il calcolo dle drift dei normali giroscopi, oppure può venire integrato nel filtro sostituendo il ruolo della bussola descritto in [8]. Se invece si riescono ad ottenere misure molto precise, sacrificando magari la velocità di campionamento, la misura dei giroscopi può essere utilizzata per re-inizializzare ad un valore certo e preciso la stima dell'orientazione, utilizzando accelerometri e giroscopi per la stima dell'orientazione nell'intervallo di tempo compreso tra due aggiornamenti del giroscopio visivo.

6.1.1 Definizione di quaternione

Il filtro implementato fa uso estensivo della rappresentazione delle rotazioni mediante quaternioni, di cui vengono esposte di seguito le principali proprietà

di interesse in questo lavoro. I quaternioni sono un'estensione dei numeri complessi ideata da Hamilton nel 1843. I numeri complessi possono essere visti come una rappresentazione di un punto su un piano e si basano sulla definizione dell'unità immaginaria $i^2 = -1$. Hamilton, volendo estendere questo concetto allo spazio, pose le definizioni:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (6.1)$$

$$\begin{aligned} ij &= k = -ij \\ jk &= i = -jk \\ ki &= j = -ki \end{aligned} \quad (6.2)$$

Un quaternione è definito come un numero complesso di quattro componenti:

$$q = q_0 + iq_1 + jq_2 + kq_3$$

ed è dunque rappresentabile con una 4-tupla:

$$q = [q_0, q_1, q_2, q_3]$$

Spesso è anche rappresentato in forma di uno scalare e un vettore a tre componenti:

$$q = q_0 + \mathbf{q}$$

6.1.2 Alcune proprietà dei quaternioni

Coniugio

Il complesso coniugato di un quaternione $q = q_0 + \mathbf{q}$, denotando con q^* è definito come:

$$q^* = q_0 - \mathbf{q} = q_0 - iq_1 - jq_2 - kq_3$$

. Conseguentemente vale la relazione: $(pq)^* = q^*p^*$

Norma

La norma di un quaternione q è definita come il prodotto scalare del coniugato per il quaternione q :

$$|q| = \sqrt{q^*q}$$

Un quaternione si dice unitario se la sua norma è 1.

Inverso di un quaternione

L'inverso di un elemento q è definito come l'elemento q^{-1} tale che: $q^{-1}q = 1$ e $qq^{-1} = 1$. Premoltiplicando ambo i membri della precedente equazione per q^* si ottiene:

$$q^*qq^{-1}1 = q^* \Rightarrow |q|^2q^{-1} = q^* \Rightarrow q^{-1} = \frac{q^*}{|q|^2}$$

Si deduce quindi che se q è un quaternione unitario, allora $q^{-1} = q^*$.

Prodotto tra quaternioni

Il prodotto tra due quaternioni p e q è definito come il prodotto tra due polinomi nelle incognite q_i, p_i con $i \in [0, 3]$. Tenendo conto della definizione dei numeri complessi i, j, k segue che:

$$p \times q = p_0q_0 - \mathbf{p} \cdot \mathbf{q} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}$$

. Si noti che il prodotto tra quaternioni non è commutativo, poiché (con riferimento alla formula precedente) non lo è il prodotto vettoriale di due vettori in \mathbb{R}^3 . Esprimendo i quaternioni come 4-tuple, il prodotto fra quaternioni si

può esprimere come:

$$p \times q = \begin{bmatrix} a_1 b_1 - a_2 b_2 - a_3 b_3 - a_4 b_4 \\ a_1 b_2 + a_2 b_1 + a_3 b_4 - a_4 b_3 \\ a_1 b_3 - a_2 b_4 + a_3 b_1 + a_4 b_2 \\ a_1 b_4 + a_2 b_3 - a_3 b_2 + a_4 b_1 \end{bmatrix}^T$$

Relazione tra quaternioni e vettori in \mathbb{R}^3

I numeri complessi nella forma $1 \cdot a + i \cdot b$ sono rappresentabili come vettori nel piano considerando la parte reale a come moltiplicata per il versore $\mathbf{1} = [1, 0]$ e la parte immaginaria b moltiplicata per il versore $\mathbf{i} = [0, 1]$. Allo stesso modo esiste una corrispondenza biunivoca tra i vettori in \mathbb{R}^3 e quaternioni *puri*, ovvero i quaternioni a parte scalare nulla: $q = iq_1 + jq_2 + kq_3$, considerando i, j, k come i tre versori di \mathbb{R}^3 . Riprendendo le definizioni 6.3, e riconsiderandole come prodotto vettoriale otteniamo:

$$\begin{aligned} i \times j &= k &= -j \times i \\ j \times k &= i &= -k \times j \\ k \times i &= j &= -i \times k \end{aligned} \tag{6.3}$$

esattamente come nei prodotti vettoriali tra vettori $\in \mathbb{R}^3$.

Relazione tra quaternioni e rotazioni di vettori in \mathbb{R}^3

Una rotazione in $\in \mathbb{R}^3$ è un'operazione che associa ad un vettore $v_1 \in \mathbb{R}^3$ un vettore $v_2 \in \mathbb{R}^3$. Una rotazione di θ lungo un vettore \mathbf{r} può anche essere descritta come un cambio di riferimento dalla terna A originale ad una terna B con origine coincidente ma ruotata lungo il vettore \mathbf{r} di un angolo $-\theta/2$

Secondo la relazione esposta nel paragrafo precedente, una rotazione in $\in \mathbb{R}^3$ può essere dunque considerata come un'operazione che associa ad un

quaternione puro q un altro quaternione puro p . Si consideri il quaternione di modulo unitario:

$${}^A_B \hat{q} = \left[\cos \frac{\theta}{2} \quad -r_x \sin \frac{\theta}{2} \quad -r_y \sin \frac{\theta}{2} \quad -r_z \sin \frac{\theta}{2} \right]$$

e un vettore rappresentato dal quaternione puro ${}^A v$. Il vettore ${}^B \mathbf{v}$ definito come ${}^B \mathbf{v} = {}^A_B \hat{q} \times {}^A \mathbf{v} \times {}^A_B \hat{q}^*$ Rappresenta una rotazione del vettore ${}^A \mathbf{v}$ di un angolo $-\theta/2$ attorno al vettore rappresentato dalla parte non scalare di ${}^A_B \hat{q}$: $[r_x, r_y, r_z]$; allo stesso modo il quaternione ${}^A_B \hat{q}$ rappresenta un cambio di coordinate da un riferimento A ad un riferimento B ruotato di un angolo $-\theta/2$ attorno al vettore \mathbf{r} , la cui corrispondente matrice di cambio base è:

$${}^A_B R = \begin{bmatrix} 2q_1^2 - 1 + 2q_2^2 & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 + q_3q_4) \\ 2(q_2q_3 + q_1q_4) & 2q_1^2 - 1 + 2q_3^2 & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 + q_1q_2) & 2q_1^2 - 1 + 2q_4^2 \end{bmatrix} = {}^B_A R^T$$

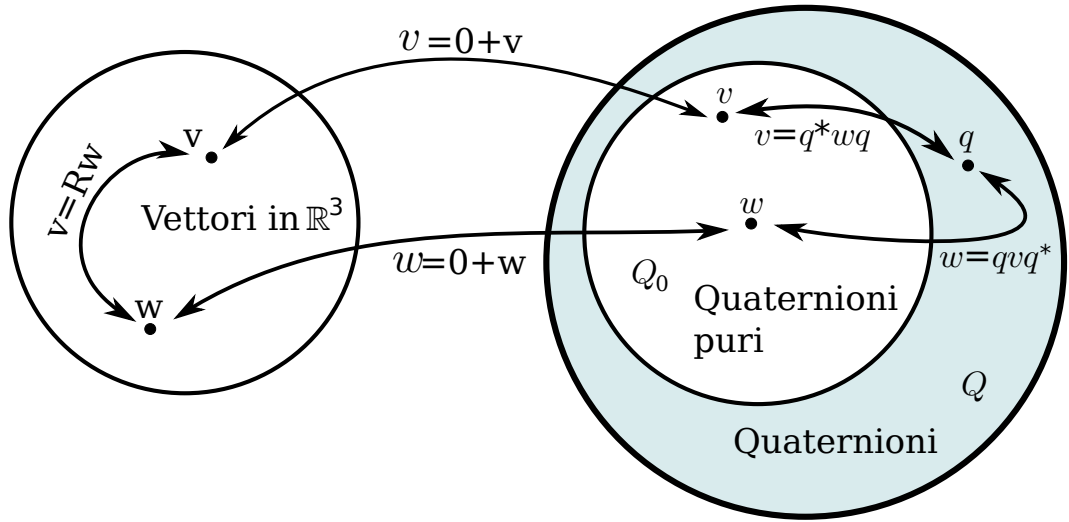


Figura 6.2: Relazione tra vettori in \mathbb{R}^3 , quaternioni e rotazioni.

Le rotazioni sono componibili attraverso una semplice moltiplicazione:

date le rotazioni ${}^A_B\hat{q}$ e ${}^B_C\hat{q}$, la rotazione composta ${}^A_C\hat{q}$ è data da:

$${}^A_C\hat{q} = {}^B_C\hat{q} \times {}^A_B\hat{q}$$

6.1.3 Calibrazione del sensore inerziale

Il sensore inerziale è il VS-IX001, fornito dalla V-Stone insieme al Robovie-X. Il sensore inerziale è composto da un accelerometro a 3 assi, un giroscopio a due assi e un convertitore ADC a 12 bit (ADS7828). Non vengono forniti né i valori di fondo scala dei sensori, né il coefficiente di trasformazione tra output del sensore e dimensione fisica misurata. È quindi necessario effettuare una calibrazione dei due sensori.

Accelerometri

L'accelerometro con cui è equipaggiato il robot è un accelerometro a 3 assi, posto in modo tale che l'asse Z è normale al piano trasversale e diretto verso i piedi e l'asse X è normale al piano sagittale e diretto verso la parte sinistra del robot. L'output degli accelerometri presenta alcuni elementi di disturbo. Innanzi tutto, alcune letture degli accelerometri non hanno successo, restituendo valori nulli (0) o al limite dell'intervallo del tipo di dato *integer* (± 32767). Un altro errore presente è dovuto al non preciso montaggio del sensore all'interno del torso del robot, cosicché gli assi dell'accelerometro non corrispondono agli assi del sistema di riferimento posto sul torso del robot. Ulteriore elemento di disturbo è la differenza della scala lungo i diversi assi dell'accelerometro: ad un'accelerazione di 1G lungo l'asse X L'IMU risponde con un valore differente rispetto ad un'accelerazione della stessa intensità lungo gli assi Z o Y. Infine, è presente un rumore di misura modellabile come un rumore bianco additivo. Per effettuare la calibrazione dell'accelerometro sono stati acquisiti tre set di dati, ciascuno con il robot posto in modo tale che la direzione della gravità coincidesse con un diverso asse. In questo modo è stato possibile rilevare il valore corrispondente all'accelerazione di 1G lungo ogni asse. In modo simile è stato rilevato il valore corrispondente allo zero di ogni asse, posizionando il robot in modo tale che l'asse da calibrare

appartenesse al piano perpendicolare al vettore gravità. I valori degli offset sono stati sottratti all'output degli accelerometri, e gli assi X,Y sono stati riscaldati affinché i valori rappresentanti lungo questi assi un'accelerazione di 1G corrispondessero al valore segnato dall'asse Z.

L'unità di misura degli accelerometri è irrilevante, dal momento che il vettore composto dalle misure dei tre assi verrà normalizzato; ad ogni modo un'accelerazione di 1G corrisponde ad un output (una volta sottratto l'offset e riscaldati gli assi come precedentemente spiegato) pari a circa 1000.

Per eliminare le fonti di disturbo durante l'acquisizione vengono eliminati tutti i campioni con valore in modulo superiore a 6000, cioè campioni che rappresenterebbero accelerazioni pari o superiori a 6G, accelerazioni decisamente superiori a quelle ottenibili dal robot e probabilmente maggiori del massimo rilevabile dagli accelerometri. Per quanto riguarda i campioni a zero, per essere eliminati efficacemente si necessiterebbe di un filtro rendendo necessaria la memorizzazione di un certo numero di campioni ed un aumento dei calcoli richiesti. Dal momento che il filtro che verrà illustrato nelle prossime sezioni ha già un effetto passa-basso sull'angolazione rilevata dagli accelerometri (a causa dell'impiego di un algoritmo di discesa del gradiente con velocità di convergenza limitata) si è preferito rinunciare all'eliminazione di tali disturbi.

Riassumendo, il segnale proveniente dagli accelerometri viene condizionato:

- eliminando i campioni rappresentanti un'accelerazione maggiore di 6G
- sottraendo l'offset iniziale
- riscaldando gli assi X, Y per avere lo stesso fondo scala dell'asse Z

Giroscopi

Il giroscopio presente sulla scheda è un giroscopio a due assi che, a meno di imprecisioni di montaggio, dovrebbero coincidere con gli assi X e Y dell'accelerometro e del robot. La maggior parte dei giroscopi forniscono in output un valore proporzionale alla velocità angolare del sensore. I giroscopi presenti

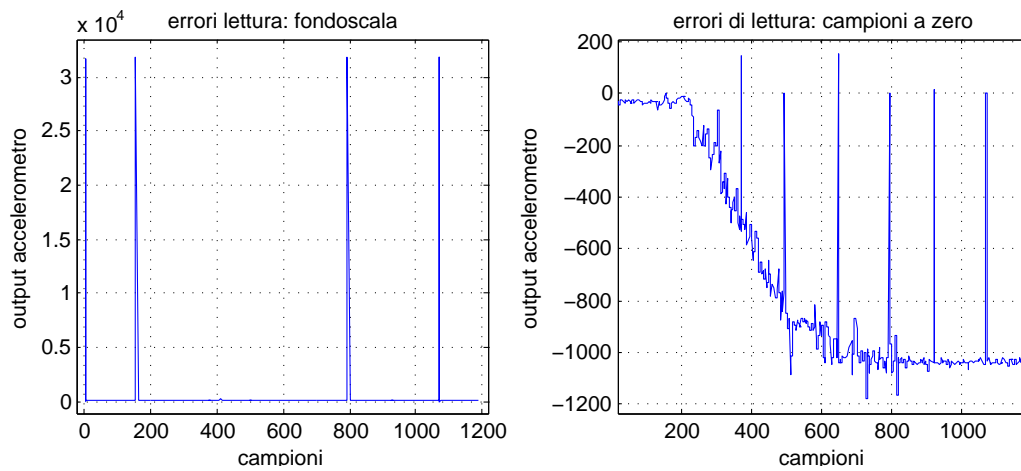


Figura 6.3: Errori di lettura dei sensori inerziali: campioni a fondo scala (figura di sinistra) e campioni a zero (destra).

sull'unità VS-IX001 invece forniscono un valore proporzionale all'accelerazione angolare del sensore ed è quindi necessaria un'integrazione dei dati ottenuti per ottenere la velocità angolare. Questo fatto accentua una problematica tipica dei giroscopi: il drift. Un giroscopio tipicamente ha un piccolo offset che provoca un output non nullo anche se la velocità angolare è nulla. Siccome l'orientazione si ottiene per integrazione della velocità angolare, l'offset si trasforma in uno scostamento dallo zero che cresce linearmente con il tempo. Nei giroscopi dell'unità VS-IX001 l'output rappresenta però l'accelerazione angolare del sistema: è dunque necessaria una doppia integrazione per ottenere l'orientazione e un offset non nullo diventa uno scostamento dallo zero che cresce quadraticamente rispetto al tempo. Un ulteriore problema è che l'offset non è costante, ma varia nel tempo quando il giroscopio è sottoposto ad accelerazioni. Anche in questo caso non è disponibile il coefficiente moltiplicativo necessario per esprimere la grandezza misurata in deg/s^2 . Per la calibrazione sono state acquisite due serie di dati (una per l'asse X e una per l'asse Y) seguendo questa procedura: il robot veniva lasciato immobile per un primo intervallo $[t_0, t_1]$, poi ruotato di 90° nell'intervallo $[t_1, t_2]$ ed infine lasciato immobile per un altro intervallo $[t_2, t_3]$. Integrando i dati raccolti si ottiene una stima della velocità angolare affetta dal problema del

drift causato dall'offset (grafico fig 6.4). Dalla figura si può anche vedere come l'offset sia costante sia nell'intervallo antecedente la rotazione che in quello successivo (in quanto la velocità angolare stimata cresce in maniera lineare) ma anche come l'offset non sia lo stesso in quanto le due rette hanno coefficiente angolare diverso, confermando così il fenomeno della variazione dell'offset precedentemente menzionato. Per stimare l'offset iniziale è sufficiente calcolare il coefficiente angolare della retta relativo al primo intervallo $[t_0, t_1]$:

$$\Delta_{offset} = \frac{v(t_1) - v(t_0)}{t_1 - t_0}$$

Una volta ottenuta la stima dell'offset iniziale sottraendo ai dati raccolti l'offset e integrando i valori ottenuti si ottiene una stima non dipendente dall'offset iniziale, ma solamente dalla variazione di quest'ultimo¹ (figura). Per ottenere l'orientamento è necessario calcolare il coefficiente moltiplicativo che lega l'output del giroscopio espresso in "livelli²" al secondo (*lev/s*) alla velocità angolare espressa in gradi (o radianti) al secondo. Per far questo si sfrutta la conoscenza dell'angolo di cui è stato ruotato il robot: l'integrale del segnale della velocità calcolato nell'intervallo $[t_1, t_2]$ è proporzionale all'orientazione e il suo valore calcolato in t_2 dev'essere pari a 90° , dunque:

$$c_p = \frac{90}{x(t_2)} \left[\frac{deg}{lev} \right]$$

. Come gli accelerometri, anche i giroscopi soffrono di simili errori di lettura; che vengono trattati in maniera simile eliminando le letture con valori maggiori di 6000. Anche in questo caso gli errori di lettura che restituiscono campioni uguali a zero vengono tralasciate, in quanto non influiscono significativamente sulla misura dell'angolazione³.

¹il compito di correggere questo disturbo è demandato al filtro

²L'output del giroscopio non è espresso in una unità di misura definita. Chiamiamo livelli l'unità di misura fittizia dei dati in uscita.

³Soprattutto poiché l'output grezzo del giroscopio va integrato per ottenere la velocità angolare, nell'approssimazione a tempo discreto dell'integrale un singolo campione ha un peso inversamente proporzionale alla frequenza di campionamento e dunque nel caso in esame molto piccolo. Infatti, nel grafico relativo al giroscopio in fig. 6.3, non è possibile notare gli effetti tali errori di lettura.

In conclusione, i dati ottenuti dai giroscopi prima di essere usati come ingressi per il filtro per l'orientazione devono essere condizionati:

- eliminando i campioni superiori a 6000
- sottraendo l'offset iniziale
- integrando nel tempo i valori ottenuti, per passare alla velocità angolare

6.2 Ricavare l'orientazione dal sensore inerziale

È possibile ricavare una stima dell'orientazione sia partendo dai dati forniti dai giroscopi (velocità angolare), sia a partire dai dati degli accelerometri. Le due stime (che chiameremo \hat{q}_w e \hat{q}_∇) hanno delle limitazioni intrinseche, il compito del filtro è effettuare una fusione di queste due stime per ottenere una migliore approssimazione dell'orientazione (\hat{q}_{est}), quanto meno possibile sensibile ai problemi delle singole stime.

6.2.1 Ricavare l'orientazione dall'output dei giroscopi

Poniamo un riferimento iniziale chiamato *world*. Supponiamo che il riferimento *sensor* solidale ai giroscopi (posto sul busto del robot) sia coincidente con la terna *world* all'istante iniziale t_0 ; sia infine ${}^S\omega = [0 \ \omega_x \ \omega_y \ \omega_z]$ un vettore rappresentante le velocità angolari istantanee del sistema *sensor* espresse nel medesimo riferimento. Il quaternion ${}^S_W\hat{q}$ rappresenta la rotazione che sovrappone il frame *world* al frame *sensor*, quindi la sua derivata ${}^S_W\dot{\hat{q}}$ rappresenta la velocità di rotazione del frame *world* espresso rispetto al frame *sensor*:

$${}^S_W\dot{\hat{q}} = \frac{1}{2} {}^S_W\hat{q} \times {}^S\omega \quad (6.4)$$

Per ottenere l'orientazione della terna *world* rispetto alla terna di riferimento è sufficiente integrare la velocità di rotazione. Essendo il segnale in uscita dagli accelerometri digitale, denotando con il pedice t l'istante di

campionamento e con Δt il periodo di campionamento si ottiene:

$${}^S_W \dot{q}_{\omega,t} = \frac{1}{2} {}^S_W \hat{q}_{est,t-1} \times {}^S \omega_t \quad (6.5)$$

$${}^S_W q_{\omega,t} = {}^S_W \hat{q}_{est,t-1} + {}^S_W \dot{q}_{\omega,t} \Delta t \quad (6.6)$$

La stima dell'orientazione attraverso i giroscopi soffre del problema della divergenza dell'errore della stima a causa della necessità di effettuare integrazioni del segnale in ingresso.

6.2.2 Ricavare l'orientazione dall'output degli accelerometri

Supponiamo dapprima che il robot non sia sottoposto ad accelerazioni lineari. L'unica accelerazione che gli accelerometri misurano è quindi l'accelerazione di gravità, la cui direzione è ben nota nel riferimento *world*. Misurando la direzione dell'accelerazione di gravità nel riferimento *sensor* è dunque possibile calcolare la trasformazione che porta il riferimento *sensor* a coincidere con *world*, a meno di una rotazione su un asse parallelo alla direzione dell'accelerazione di gravità (tipicamente l'asse Z della terna *world*). Questo porta ad una quantità infinita di rotazioni ${}^S_W \hat{q}$ plausibili tra cui scegliere. Per fissare un'unica soluzione scegliamo la rotazione ${}^S_W \hat{q}$ che minimizzi una funzione di costo:

$$\min_{{}^S_W \hat{q}} f({}^S_W \hat{q}, {}^W \hat{d}, {}^S \hat{s}) \quad (6.7)$$

$$f({}^S_W \hat{q}, {}^W \hat{d}, {}^S \hat{s}) = {}^S_W \hat{q}^* \times {}^W \hat{d} \times {}^S_W \hat{q} - {}^S \hat{s} \quad (6.8)$$

dove ${}^S \hat{s}$ rappresenta la stima della direzione dell'accelerazione di gravità nel riferimento *sensor* e ${}^W \hat{d}$ nel riferimento *world*. Essendo ${}^S \hat{s}$ e ${}^W \hat{d}$ dei quaternioni puri, la funzione $f({}^S_W \hat{q}, {}^W \hat{d}, {}^S \hat{s})$ risulta anch'essa essere un quaternioni puro. Sviluppando i calcoli e tralasciando il primo elemento sempre nullo la

funzione di costo diventa:

$$f({}^S_W\hat{q}, {}^W\hat{d}, {}^S\hat{s}) = \begin{bmatrix} 2d_x(\frac{1}{2} - q_3^2 - q_4^2) + 2d_y(q_1q_4 + q_2q_3) + 2d_z(q_2q_4 + q_1q_3) - s_x \\ 2d_x(q_2q_3 + q_1q_4) + 2d_y(\frac{1}{2} - q_2^2 - q_4^2) + 2d_z(q_1q_2 + q_3q_4) - s_y \\ 2d_x(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 + q_1q_2) + 2d_z(\frac{1}{2} - q_2^2 - q_3^2) - s_z \end{bmatrix} \quad (6.9)$$

Se la terna di riferimento world ha l'asse Z parallelo e concorde alla direzione dell'accelerazione di gravità, si ha che ${}^W\hat{d} = {}^W\hat{g} = [0 \ 0 \ 0 \ 1]$ e la funzione f si semplifica notevolmente perdendo i primi due termini di ogni riga:

$$f({}^S_W\hat{q}, {}^W\hat{g}, {}^S\hat{s}) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - s_x \\ 2(q_1q_2 + q_3q_4) - s_y \\ 2(\frac{1}{2} - q_2^2 - q_3^2) - s_z \end{bmatrix} \quad (6.10)$$

Per minimizzare la funzione di costo viene utilizzato il procedimento di discesa del gradiente:

$${}^S_W\hat{q}_{k+1} = {}^S_W\hat{q}_k - \mu \frac{\nabla f({}^S_W\hat{q}, {}^W\hat{d}, {}^S\hat{s})}{\|\nabla f({}^S_W\hat{q}, {}^W\hat{d}, {}^S\hat{s})\|}, \quad k = 0, 1, \dots, n \quad (6.11)$$

$$(6.12)$$

La stima iniziale ${}^S_W\hat{q}_0$ per il calcolo dell'orientazione nell'istante t è posta uguale alla stima di ${}^S_W\hat{q}$ nell'istante $t - 1$; $J_f^T({}^S_W\hat{q}, {}^W\hat{d})$ è lo Jacobiano della funzione di costo che diventa, nel caso in cui ${}^W\hat{d} = {}^W\hat{g}$:

$$J_f^T({}^S_W\hat{q}, {}^W\hat{g}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \quad (6.13)$$

La lunghezza del passo μ ed il numero n di passi sono variabili di progetto, che dipendono dalle caratteristiche del movimento del corpo di cui si vuole

calcolare l'orientamento: è necessario che la velocità di convergenza dell'algoritmo di discesa del gradiente sia maggiore della velocità di rotazione del sistema *sensor*. La lunghezza del passo di discesa del gradiente dovrà essere proporzionale alla differenza di orientazione rispetto al passo precedente e dunque sarà proporzionale alla lunghezza dell'intervallo di campionamento e alla velocità indicata dai giroscopi e per essere ottimale dovrebbe dipendere anche dalle derivate della funzione di costo. Per diminuire la complessità computazionale dell'algoritmo, in [8] viene proposto di utilizzare un solo passo ($n = 1$) e una lunghezza del passo di campionamento

$$\mu_t = \alpha \|\mathcal{S}_W \dot{q}_{w,t}\| \Delta t, \quad \alpha > 1 \quad (6.14)$$

dove α è fattore moltiplicativo che serve ad includere l'effetto del rumore nelle misure di giroscopi ed accelerometri. Questa scelta è supportata dai risultati delle prove sperimentali effettuate dall'autore dell'articolo.

6.2.3 Adattamenti per la scheda VS-IX001

Durante una normale camminata di un robot umanoide le principali rotazioni del busto avvengono sugli assi X e Y (rollio e beccheggio), inoltre non disponendo sulla scheda VS-IX001 di un giroscopio lungo l'asse Z del Robot si è impossibilitati a calcolare le rotazioni su tale asse. La soluzione adottata consiste nel considerare sempre nulla la rotazione su questo asse; questa supposizione è realistica, in quanto Robovie-X, durante l'esecuzione di una camminata rettilinea devia solo leggermente ruotando sull'asse Z. Per ottenere valori di $\mathcal{S}_W \hat{q}$ privi di componente di rotazione sull'asse Z, non è sufficiente fornire in ingresso un valore fittizio di ω in cui è posto $\omega_z = 0$ in quanto l'algoritmo di discesa del gradiente (nel calcolare il minimo della funzione di costo) può fornire soluzioni $\mathcal{S}_W \hat{q}$ comprendenti rotazioni sull'asse Z. Questo comporta significative rotazioni soprattutto in alcune configurazioni "sensibili" quali ad esempio ponendo l'asse Z del sistema "sensor" sul piano perpendicolare alla direzione della gravità. In queste configurazioni, viene rilevata una continua rotazione sull'asse Z. Per evitare questo, è necessario

vincolare il problema di ottimizzazione:

$$\tilde{Q} = \{q \mid q = q_0 + iq_1 + jq_2 + k \cdot 0, q_i \in \mathbb{R}\} \quad (6.15)$$

$$\min_{\hat{q}_W^S} f(\hat{q}_W^S, \hat{d}^W, \hat{s}^S), \hat{q}_W^S \in \tilde{Q} \quad (6.16)$$

Per tali \hat{q}_W^S , la funzione di costo ed il suo gradiente diventano:

$$f(\hat{q}_W^S, \hat{g}^W, \hat{s}^S) = \begin{bmatrix} -2q_1q_3 - s_x \\ 2q_1q_2 - s_y \\ 2(\frac{1}{2} - q_2^2 - q_3^2) - s_z \end{bmatrix} \quad (6.17)$$

$$J_f^T(\hat{q}_W^S, \hat{g}^W) = \begin{bmatrix} -2q_3 & 0 & -2q_1 \\ 2q_2 & 2q_1 & 0 \\ 0 & -4q_2 & -4q_3 \end{bmatrix} \quad (6.18)$$

6.2.4 Fusione delle stime di giroscopi ed accelerometri

Una volta calcolate le orientazioni secondo i giroscopi e gli accelerometri queste devono essere combinate per ottenere la stima dell'orientazione \hat{q}_W^S . La combinazione avviene attraverso una media pesata dal coefficiente γ_t :

$$\hat{q}_W^S = \gamma_t \hat{q}_{\nabla,t}^S + (1 - \gamma_t) \hat{q}_{\omega,t}^S, \quad 0 \leq \gamma_t \leq 1 \quad (6.19)$$

La stima della posizione angolare calcolata dai giroscopi $\hat{q}_{\omega,t}^S$ soffre del fenomeno del drift, che la fa divergere progressivamente, mentre la stima calcolata dagli accelerometri $\hat{q}_{\nabla,t}^S$ ha una certa velocità di convergenza dovuta al processo di discesa del gradiente; un buon valore di γ_t può dunque essere quello che rende uguale la velocità di convergenza di $\hat{q}_{\nabla,t}^S$ (che corrisponde a $\frac{\mu_t}{\Delta t}$)

e la velocità con cui diverge $\overset{S}{W}\hat{q}_{\omega,t}$ (che chiamiamo β):

$$(1 - \gamma_t)\beta = \gamma_t \frac{\mu_t}{\Delta t} \quad (6.20)$$

$$\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t} + \beta} \quad (6.21)$$

Per valori di α abbastanza grandi (6.14) il termine β a denominatore diventa trascurabile, quindi γ_t è approssimabile a:

$$\gamma_t \approx \frac{\beta \Delta t}{\mu_t} \quad (6.22)$$

Similmente, nell'equazione 6.11 il termine $\overset{S}{W}\hat{q}_t$ diventa anch'esso trascurabile:

$$\overset{S}{W}\hat{q}_{t+1} \approx -\mu \frac{\nabla f}{\|\nabla f\|}, \quad k = 0, 1, \dots, n \quad (6.23)$$

Sostituendo nell'equazione 6.19:

$$\begin{aligned} \overset{S}{W}\hat{q}_{est} &\approx \frac{\beta \Delta t}{\mu_t} \left(-\mu_t \frac{\nabla f}{\|\nabla f\|} \right) + \left(1 - \frac{\beta \Delta t}{\mu_t} \right) (\overset{S}{W}\hat{q}_{est,t-1} + \overset{S}{W}\dot{q}_{\omega,t} \Delta t) \\ &\approx \beta \Delta t \left(-\frac{\nabla f}{\|\nabla f\|} \right) + (\overset{S}{W}\hat{q}_{est,t-1} + \overset{S}{W}\dot{q}_{\omega,t} \Delta t) \end{aligned} \quad (6.24)$$

dove al secondo termine è stato trascurato γ_t in quanto $\gamma_t \ll 1$. Chiamando $\overset{S}{W}\dot{q}_{\epsilon,t} = \frac{\nabla f}{\|\nabla f\|}$, possiamo riscrivere la precedente equazione come:

$$\overset{S}{W}\hat{q}_{est,t} = \overset{S}{W}\hat{q}_{est,t-1} + \overset{S}{W}\dot{q}_{est,t} \Delta t \quad (6.25)$$

$$\overset{S}{W}\dot{q}_{est,t} = \overset{S}{W}\dot{q}_{\omega,t} - \beta \overset{S}{W}\dot{q}_{\epsilon,t} \quad (6.26)$$

L'intero procedimento è riassunto schematicamente in figura 6.5

6.2.5 Risultati

Il risultato è mostrato in figura 6.6. Come si può notare, ad una rotazione a velocità abbastanza regolare⁴ (lo si nota dalla regolarità di variazione

⁴La rotazione è stata effettuata a mano

nel grafico relativo all'output degli accelerazioni) non corrisponde una velocità altrettanto regolare rilevata dal giroscopio, che presenta un andamento irregolare. Questo richiede di diminuire il peso del giroscopio nella stima dell'orientamento aumentando il valore del coefficiente β (eq. 6.19). In una rotazione lenta come quella presentata la stima dei giroscopi non sarebbe essenziale, in quanto le accelerazioni lineari sono trascurabili ed la stima degli accelerometri risulterebbe abbastanza precisa. Quando però le rotazioni diventano veloci e sovrapposte ad accelerazioni lineari la stima degli accelerometri diventa inaffidabile in quanto questi non rilevano più solamente la direzione dell'accelerazione di gravità ma una somma di quest'ultima con le accelerazioni impresse al busto del robot. Come verrà esposto nella sezione 8.2 in queste situazioni però degradano significativamente anche le prestazioni dei giroscopi e sarà necessario adottare altri condizionamenti del segnale proveniente dai giroscopi.

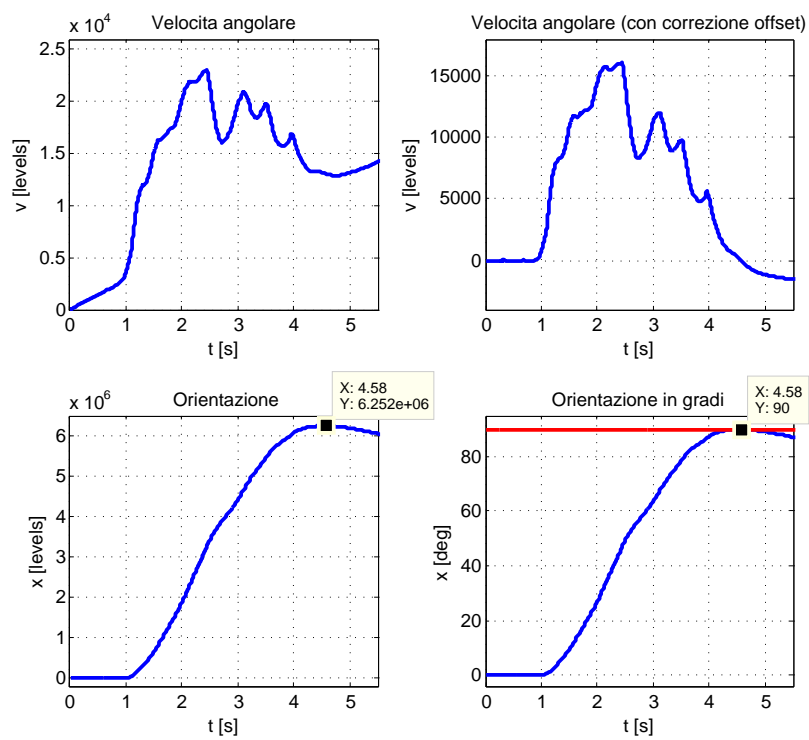


Figura 6.4: Grafici relativi alla calibrazione dei giroscopi

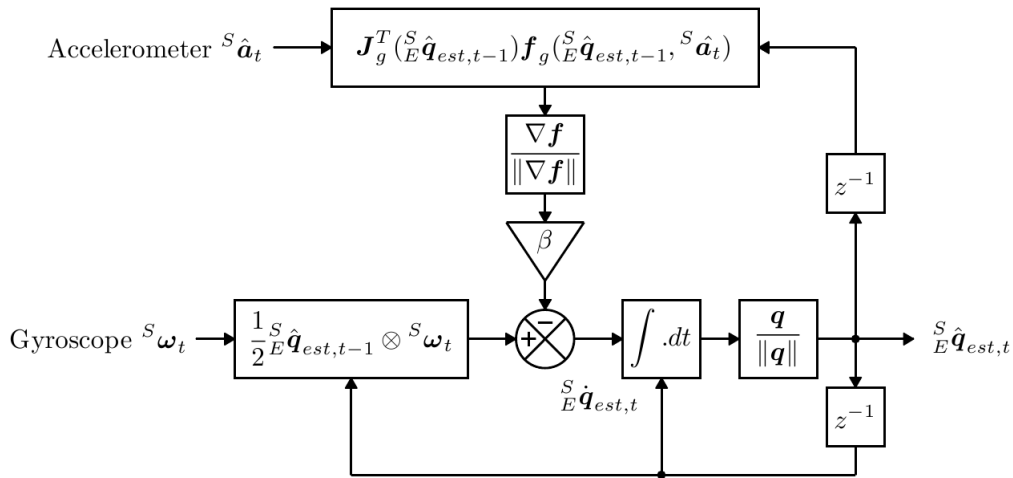


Figura 6.5: Schema del funzionamento del filtro IMU

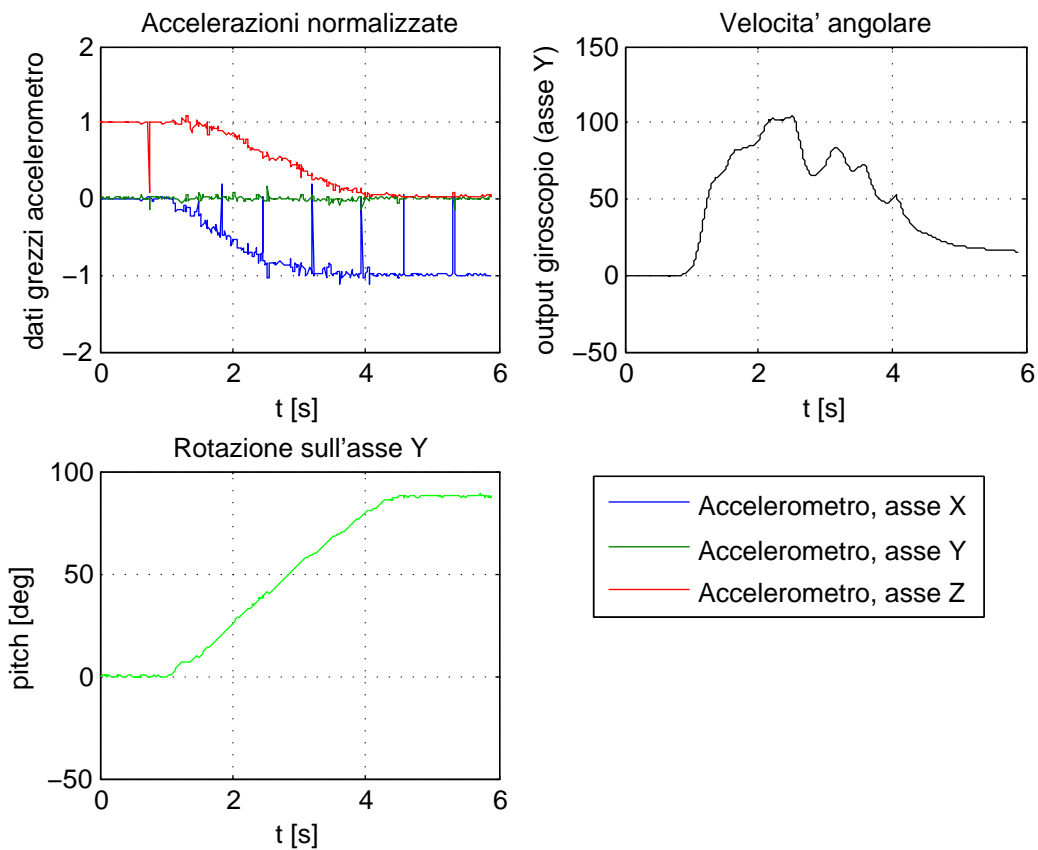


Figura 6.6: Angolo di rollio calcolato dal filtro IMU

Capitolo 7

Equilibrio su piano ad inclinazione variabile

Prima della stabilizzazione della cammina sono stati effettuati dei test per la stabilizzazione della postura eretta del robot su un piano con inclinazione variabile sugli assi X e Y. Lo scopo è testare il filtro per il calcolo dell'orientazione e il metodo di comando dei servomotori, analizzare eventuali instabilità la presenza di oscillazioni.

7.1 Struttura del programma

7.1.1 Posizione eretta

Come prima operazione, è necessario definire l'orientazione che il robot deve mantenere. La posizione (che verrà chiamata "eretta") deve garantire un margine di stabilità statica e il completo controllo dei gradi di libertà: se la gamba fosse già completamente estesa, non essendo possibile allungarla ulteriormente verrebbe meno la possibilità di inclinare il robot secondo l'asse X. Si è deciso quindi di stabilire la seguente posizione eretta:

- Busto "verticale", cioè con l'asse Z diretto secondo l'asse verticale
- Asse di pitch di anche e caviglie giacenti sul piano verticale del robot

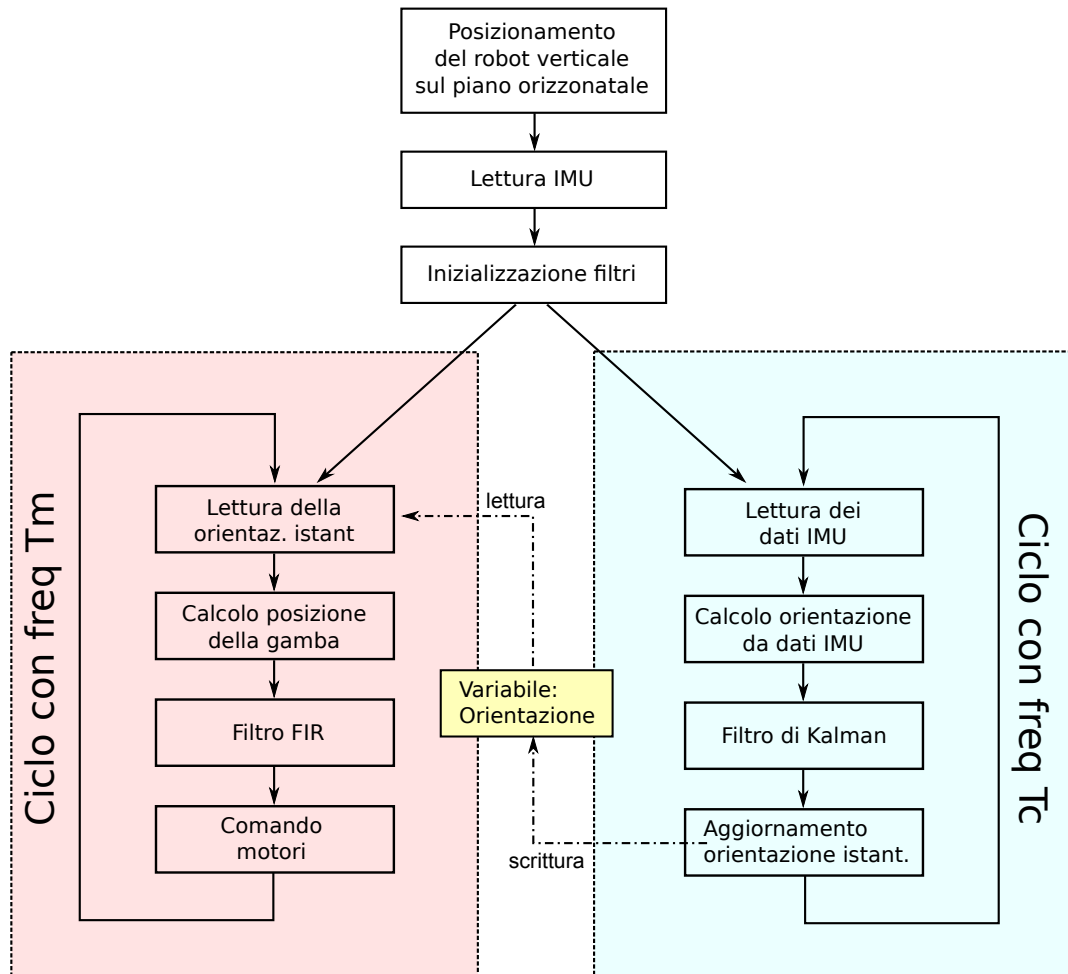


Figura 7.1: Struttura del programma

- Gamba piegata in modo tale da accorciarne la lunghezza di 2.2cm.

La lunghezza della gamba è stata decisa in base ad una limitazione relativa alla massima inclinazione della caviglia. L'asse di roll della caviglia permette una massima inclinazione di 20° verso l'esterno. Questo implica che il robot può reagire al massimo ad una inclinazione di 20° del piano sull'asse X. Per reagire ad una tale inclinazione, deve alzare la gamba interna di un tratto di lunghezza

$$\Delta H = \Delta_A \cdot \tan(20) \approx 2.18\text{cm}$$

Dove Δ_A è la distanza tra i centri dei poligoni di appoggio rappresentati dai piedi e corrisponde, nella posa eretta, alla distanza tra gli assi di roll delle anche.

7.1.2 Calibrazione ed inizializzazione

Una volta posto il robot nella configurazione eretta su un piano inclinato, è necessario imporre la sua orientazione come orientazione di riferimento, cioè memorizzare l'orientazione rilevata dal sensore inerziale in modo da poter calcolare la differenza delle future orientazioni rispetto a quella di base. Nel contempo, vengono inizializzati i filtri del sensore IMU ed il filtro di Kalman.

7.1.3 Processi di lettura IMU e di comando dei motori

La lettura del sensore inerziale può avvenire con una frequenza massima di 200Hz. Il comando dei motori non può avvenire alla stessa frequenza (nel software RobovieMaker il comando avviene tipicamente con una frequenza di 50Hz) ed abbassare la frequenza di campionamento per portarla alla frequenza di comando dei motori significherebbe peggiorarne significativamente le prestazioni. Per questo motivo si è deciso di utilizzare due processi distinti, uno operante alla frequenza di 200Hz che effettua la lettura e l'elaborazione dei dati provenienti dal sensore inerziale, ed uno che operi ad una frequenza inferiore che effettui i calcoli relativi alla cinematica inversa di posizione degli arti del robot e ne comandi gli attuatori. I processi procedono in maniera completamente indipendente l'uno dall'altro comunicando unidirezionalmente ed esclusivamente attraverso la una variabile che rappresenta l'orientazione (pitch e roll) del piano; che viene aggiornata dal processo di lettura e calcolo dell'orientazione e letta dal processo di comando dei motori.

7.1.4 Processo di comando dei motori

A questo processo è demandato il compito di calcolare le azioni da intraprendere in base all'orientazione istantanea rilevata dal processo di lettura IMU

ed il comando dei motori, informazione contenuta in due parametri: rollio e beccheggio.

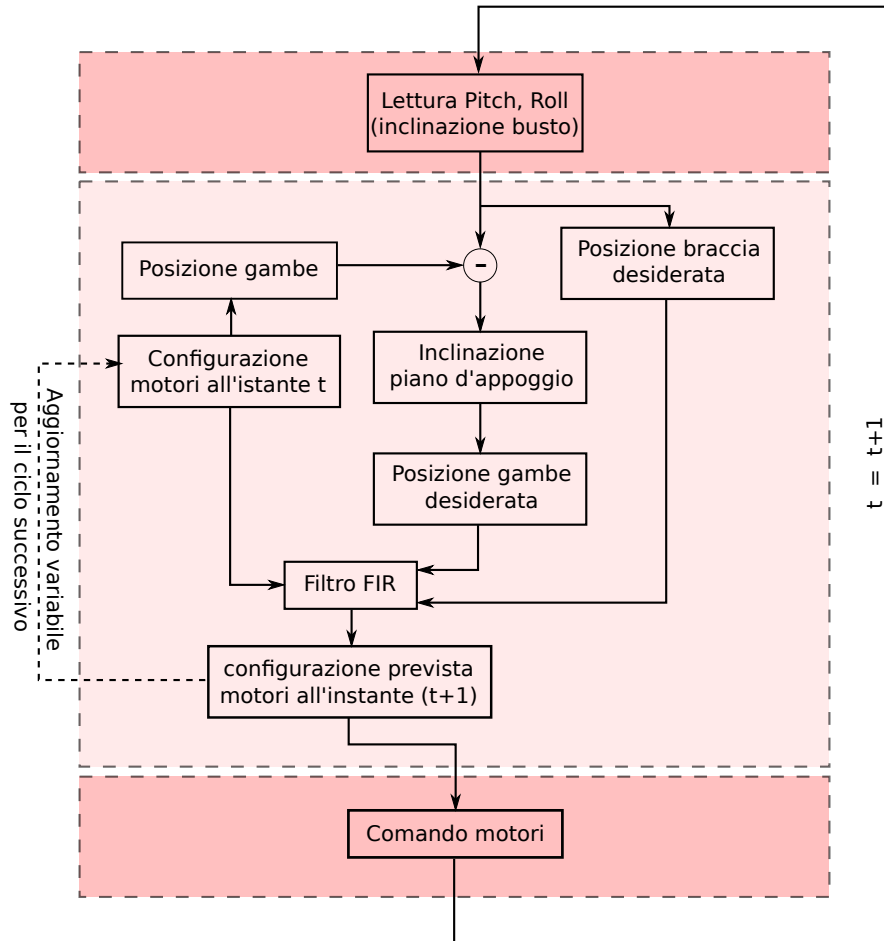


Figura 7.2: Struttura del ciclo di comando dei motori

Dal sensore inerziale presente sul torso del robot è possibile ricavare solamente l'orientazione del torso, per ottenere l'inclinazione del piano è necessario fare l'ipotesi che i piedi siano sempre a contatto con il piano di appoggio. In tal modo se si conosce la posizione dei giunti delle gambe, attraverso la cinematica diretta di posizione è possibile risalire alla rotazione busto-piedi e calcolare l'inclinazione del terreno. Purtroppo, non disponendo il Robovie-X di un modo per determinare la posizione dei singoli giunti, sarà necessario ipotizzare che la posizione dei corrisponda all'ultimo comando im-

partito. Perché entrambe queste ipotesi si possano considerare realistiche è necessario che gli spostamenti richiesti ai giunti tra due comandi consecutivi sufficientemente brevi¹ da poter considerare il movimento completato prima del successivo comando, e che tali comandi non generino momenti tali da alzare il robot sugli spigoli del piede. Per ovviare parzialmente a questa limitazione sono stati presi due provvedimenti: un filtro FIR, per rendere meno brusco il segnale di comando e l'utilizzo delle braccia per spostare il baricentro e diminuire così il momento dovuto alla forza di gravità. Supponendo di mantenere il busto ad una data orientazione ed altezza rispetto al piano di appoggio, le variabili da calcolare per mantenere il piede appoggiato a terra sono l'orientazione del piede (rollio e beccheggio) e la distanza del piede dal busto, cioè la lunghezza della gamba. Si procede dividendo la determinazione di queste variabili in due fasi.

Reazione al beccheggio

Nella prima fase, si calcolano le variabili influenzate dalla rotazione del piano sull'asse Y del riferimento del robot, cioè l'asse perpendicolare al suo piano sagittale. L'angolo (che verrà chiamato θ) si considera positivo quando il robot tende a cadere in avanti. Per opporsi a questa rotazione si agisce sui giunti di pitch delle caviglie, ruotandole di un angolo pari alla rotazione θ mentre per contrastare lo spostamento del baricentro le braccia vengono anch'esse ruotate concordemente alla rotazione del piano di un angolo $c_b\theta$ (con α variabile di progetto) in modo tale da spostare il peso "a monte" riavvicinando la proiezione del baricentro al centro del poligono di appoggio. le variazioni avvengono simmetricamente sia per le braccia che per le gambe.

Reazione al Rollio

Per rollio del piano di appoggio si intende la rotazione del piano rispetto all'asse trasversale del robot, cioè la rotazione che alza il lato destro rispetto al sinistro o viceversa. Rispetto alla terna di riferimento definita dal sensore inerziale del robot, il rollio è la rotazione rispetto all'asse X, rotazione con-

¹oppure il tempo tra due comandi sufficientemente lungo

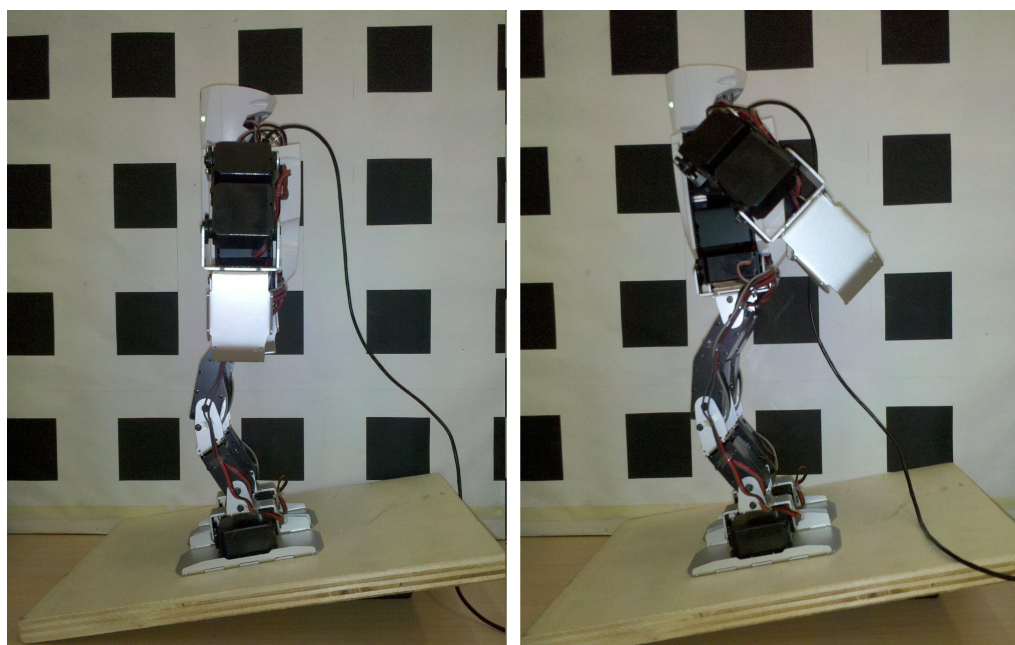


Figura 7.3: Reazione a rotazioni del piano sull'asse Y; nell'immagine di sinistra a correzione avvenuta, nell'immagine a destra durante lo spostamento

siderata positiva quando si alza il lato destro del robot. L'angolo di rollio verrà chiamato ϕ . Per mantenere il busto orizzontale è necessario adeguare la lunghezza delle gambe alla differenza di altezza del punto di appoggio (punto appartenente alla proiezione dell'asse di rollio dell'anca sul piano di appoggio). Per fare ciò è possibile seguire tre approcci: abbassare il piede "a valle", alzare il piede "a monte", effettuare contemporaneamente entrambe le azioni. Per semplicità è stata scartata l'ultima opzione, ed è stata scelta la seconda perché permette di mantenere il baricentro più basso rispetto alla prima opzione. Dovendo essere il piede appoggiato interamente a terra, è necessario ruotare anche la caviglia sull'asse di rollio di un angolo ϕ . La rotazione di quest'ultima è molto limitata verso l'esterno (massimo 20°), ciò limita l'inclinazione massima correggibile. L'inclinazione massima potrebbe essere leggermente aumentata allargando le gambe, in quanto la rotazione permessa alla caviglia verso l'interno è maggiore.

Per contrastare lo spostamento del baricentro anche in questo caso è stato

fatto uso delle braccia, effettuando una rotazione di un angolo $c_r\phi$ in direzione concorde al piano del braccio rivolto “a monte” rispetto all’inclinazione del piano; mentre il braccio a valle viene mantenuto adiacente al fianco del robot. Il coefficiente c_r dovrà essere maggiore del coefficiente c_b , in quanto nel caso del rollio si dispone del peso di un solo braccio per spostare il baricentro.

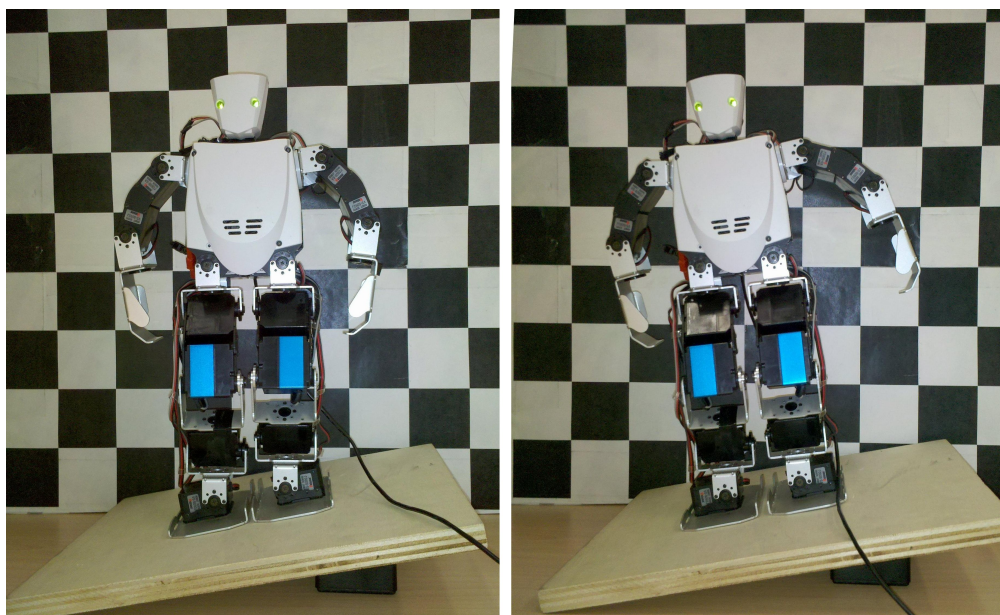


Figura 7.4: Reazione a rotazioni del piano sull’asse X; nell’immagine di sinistra a correzione avvenuta, nell’immagine a destra durante lo spostamento

In entrambi i casi, alzare il braccio portandolo ad un angolo superiore a 90° rispetto al sistema di riferimento “world” è controproducente, in quanto lo spostamento del baricentro dovuto alla posizione del braccio è proporzionale al seno dell’angolo. Per questo motivo è utile “saturare” il valore degli angoli θ e ϕ in modo da impedire che il braccio si alzi al di sopra di tale angolo. Gli angoli θ_w e ϕ_w del braccio rispetto alla terna “world” sono dati dalla somma degli angoli dell’orientazione del busto e dell’angolo del braccio rispetto al busto:

$$\begin{aligned}\theta_w &= \theta + c_b\theta \\ \phi_w &= \phi + c_b\phi\end{aligned}\tag{7.1}$$

Imponendo dunque che il braccio non ruoti di più di 90° sia sull'asse X che sull'asse Y nel riferimento "world" si ottengono le seguenti equazioni:

$$\left\{ \begin{array}{ll} \theta_{arm} = c_p \theta & \text{se } |\theta(1 + c_b)| \leq 90^\circ \\ \theta_{arm} = 90^\circ - \theta & \text{se } \theta(1 + c_b) > 90^\circ \\ \theta_{arm} = \theta - 90^\circ & \text{se } \theta(1 + c_b) < -90^\circ \end{array} \right. \quad (7.2)$$

$$\left\{ \begin{array}{ll} \phi_{dx} = c_r \phi & \text{se } 0 \leq \phi(1 + c_r) \leq 90^\circ \\ \phi_{dx} = 90^\circ - \phi & \text{se } \phi(1 + c_r) > 90^\circ \\ \phi_{dx} = 0 & \text{se } \phi < 0^\circ \end{array} \right. \quad (7.3)$$

$$\left\{ \begin{array}{ll} \phi_{sx} = c_r \phi & \text{se } -90 \leq \phi(1 + c_r) \leq 0^\circ \\ \phi_{sx} = -90^\circ - \phi & \text{se } \phi(1 + c_r) < -90^\circ \\ \phi_{sx} = 0^\circ & \text{se } \phi > 0^\circ \end{array} \right. \quad (7.4)$$

Ciclo di comando

Il ciclo di comando comincia con la lettura dell'orientazione, recuperando il valore degli angoli di rollio e beccheggio calcolati e memorizzati dal filtro IMU (vedi fig. 7.2) Conoscendo l'ultimo comando inviato ai motori, e presupponendo che i motori abbiano completato il loro movimento, viene calcolata la posizione e l'orientamento dei piedi rispetto al busto. Se è verificata l'ipotesi secondo cui l'intera superficie dei piedi è sempre a contatto col piano, sottraendo all'inclinazione reale del busto l'inclinazione prevista a partire dalla posizione dei piedi si ottiene l'inclinazione del piano di appoggio. Una volta nota l'inclinazione di quest'ultimo viene calcolata la reazione richiesta al robot, ricavando secondo i procedimenti precedentemente descritti la posizione dei piedi e dunque la "configurazione obiettivo" dei giunti delle gambe. La "posizione obiettivo" delle braccia, invece, viene invece calcolata a partire dall'inclinazione assoluta del busto, allo scopo di riavvicinare il baricentro al

centro del poligono di appoggio. Una volta ottenuta la “configurazione obiettivo” di tutti i giunti del robot, questa viene filtrata affinché la dinamica di ogni singolo giunto non presenti variazioni troppo brusche che pregiudicherebbero la stabilità del robot e le ipotesi² fatte. Per mantenere quanto più possibile computazionalmente leggero il software di gestione dei movimenti del robot si è scelto di utilizzare un semplice filtro FIR di ordine uno con funzione di trasferimento:

$$\frac{Y(s)}{X(s)} = \frac{cz}{z - (1 - c)} \quad (7.5)$$

cioè, nel dominio del tempo,

$$y(t) = (1 - c) \cdot y(t - 1) + c \cdot x(t) \quad (7.6)$$

rendendo così necessaria la memorizzazione di una sola variabile, la configurazione dei motori all’istante precedente. In figura 7.6 è visibile il grafico della risposta al gradino unitario e ad un generico segnale rumoroso del filtro.

7.1.5 Tuning dei parametri

²piedi sempre a contatto col terreno e movimenti completati entro il successivo comando

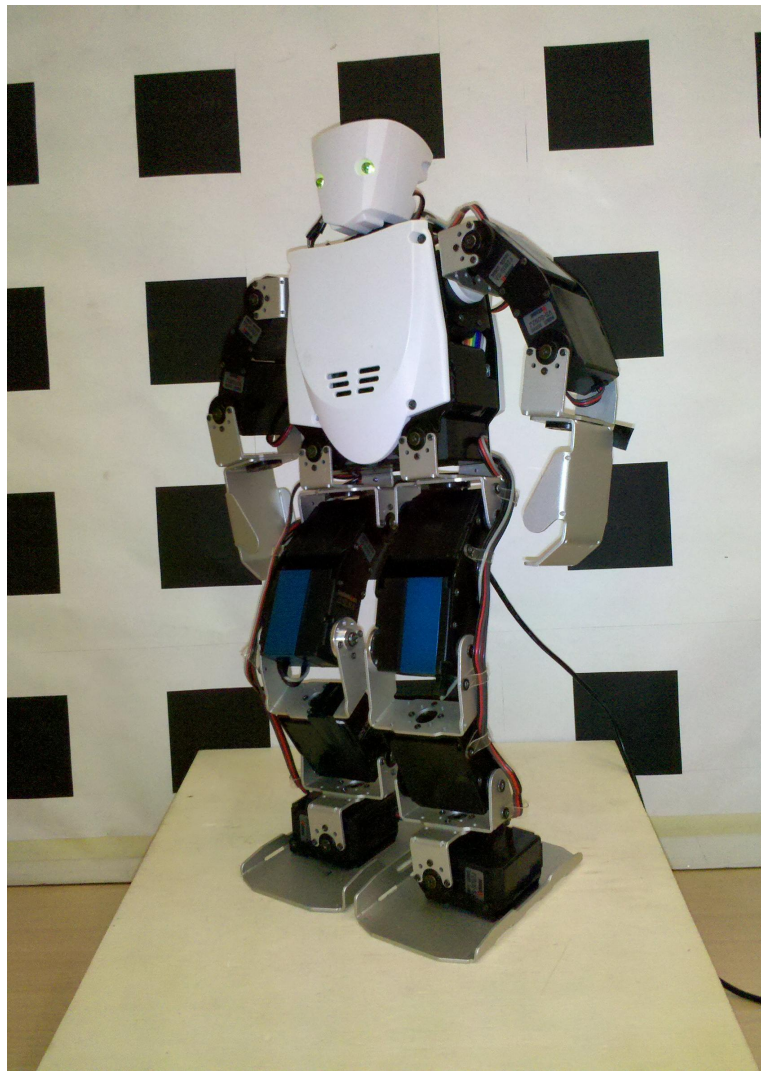


Figura 7.5: Robovie-X su un piano inclinato sia rispetto all'asse X che rispetto all'asse Y.

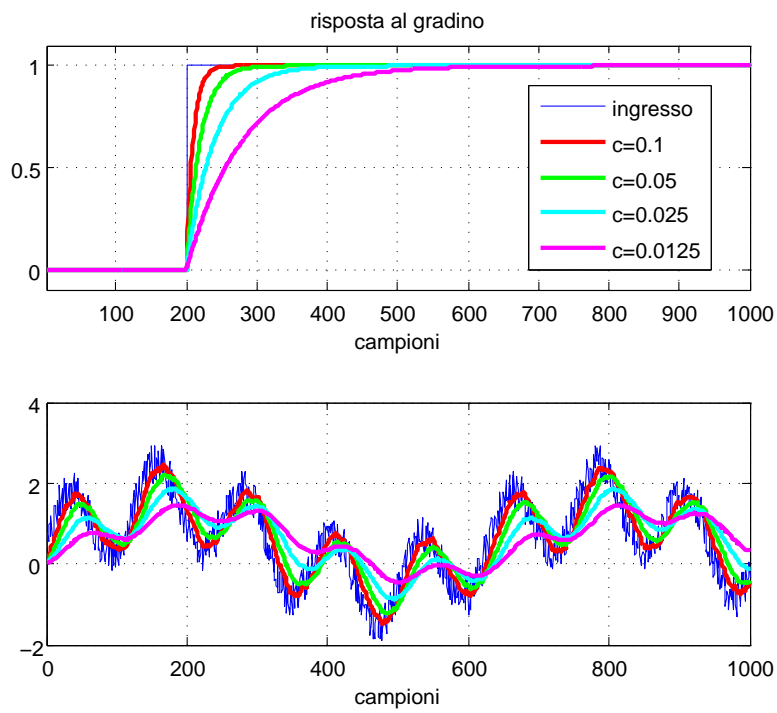


Figura 7.6: Filtro, risposta al gradino ed ad un generico segnale rumoroso

Capitolo 8

Camminata

Il concetto di stabilizzazione della camminata include in particolare due aspetti: la correzione degli scostamenti dalle posizioni ideali dovuti a stimoli esterni (ostacoli, inclinazioni del piano, spinte, ecc...) e gli approcci per la limitazione di oscillazioni tipiche della camminata stessa, per aumentare la stabilità del robot o favorire le interazioni con l'ambiente (ad esempio acquisizione di immagini, trasporto di oggetti). In questo lavoro si è preferito concentrarsi sul primo problema, in particolare la stabilizzazione della posizione del busto al variare dell'inclinazione del piano di camminata.

8.1 Correzioni tempo-varianti e tempo-invarianti

Nell'ambito delle correzioni applicabili alla camminata di un umanoide si possono distinguere due categorie di correzioni: quelle che, a parità di disturbi esterni risultano costanti nelle varie fasi della camminata e quelle che dipendono dalla configurazione istantanea del robot. Questi tipi di correzioni sono in relazione alla struttura del robot ed al tipo di giunti. Ad esempio, consideriamo una gamba come quella del Robovie-X e supponiamo di voler aumentare l'inclinazione del busto verso. Questa correzione si può ottenere ruotando la caviglia sull'asse Y e tale correzione è indipendente dal fatto che il robot sia nella fase del doppio supporto piuttosto che nella fase di avanzamento di uno dei due piedi. Se invece vogliamo una correzione come

abbassare il baricentro diminuendo la distanza tra anca e caviglia (correzione ottenibile agendo sui giunti di ginocchio, anca e caviglia) è necessario calcolare una diversa correzione per ogni istante (vedi fig.8.5, il concetto sarà ripreso nelle prossime sezioni).

8.2 Problemi dei giroscopi durante la camminata

8.2.1 Bias variabile dei giroscopi

Prima di iniziare a progettare gli algoritmi di correzione della camminata sono state analizzate le caratteristiche della camminata utilizzata. Durante questa fase si è rilevato un problema, dovuto alla variazione del bias dei giroscopi. Come menzionato nella sezione relativa al filtro IMU, gli output giroscopi sono soggetti a variazioni di offset quando sottoposti ad accelerazioni lineari che a causa dell'integrazione si trasformano in fenomeni di drift¹. Nei test precedentemente effettuati (cioè rotazioni del robot lungo diversi assi), questi fenomeni vengono corretti dal filtro in quanto le accelerazioni lineari risultavano contenute e regolari. Durante una camminata, invece le accelerazioni lineari sono di entità maggiore, e rendono inaffidabile la stima ottenuta. Una stima effettuata con l'ausilio dei soli accelerometri presentava valori sovrastimati; in quanto le accelerazioni lineari del robot durante le oscillazioni della camminata si sommano alle componenti delle accelerazioni di gravità; allo stesso tempo nel caso di grandi e veloci variazioni di angolo la stima senza l'ausilio dei giroscopi diventa più lenta, a causa della velocità massima della convergenza della stima dell'orientazione tramite accelerometri nel filtro IMU. La soluzione ideale consiste quindi nel trattare i dati dei giroscopi allo scopo di eliminare la variazione del bias.

¹I quali, nel caso particolare della scheda VS-IX001 sono parabolici anziché lineari, dato che il valore grezzo in uscita dai giroscopi è una accelerazione.

8.2.2 Eliminazione del bias dei giroscopi

Il metodo scelto per l'eliminazione del bias si basa su un'ipotesi che è realistica nel caso della camminata. Durante la camminata il busto del robot oscilla, per effetto dello spostamento del baricentro è dovuto al cambio di appoggio e allo spostamento delle gambe attorno ad una posizione di equilibrio. La velocità angolare del busto oscillerà anch'essa attorno al valore nullo. Se ne deduce che, in condizioni del tutto ideali, la media campionaria su un intervallo di lunghezza multipla rispetto alla lunghezza di un passo deve avere valore nullo. Supponendo invece che sia presente un offset (costante in quell'intervallo), la media campionaria assumerà il valore dell'offset. La strategia di eliminazione del bias adottata consiste quindi nel sottrarre ad ogni campione del segnale di velocità angolare la media di N campioni precedenti. Questa strategia elimina efficacemente l'offset, ma presenta dei costi:

- Memoria: è necessario memorizzare N campioni, quindi aumenta l'occupazione di memoria², linearmente al crescere di N .
- Semplicità: per avere una buona stima è necessario conoscere la durata del passo. È quindi un ulteriore parametro che va aggiustato non solo in relazione alle caratteristiche del sensore inerziale, ma anche in base alla camminata.
- Attenuazione componente continua: nel caso di movimenti diversi dall'oscillazione della camminata (ad esempio nel caso della rotazione utilizzata nel capitolo relativo al filtro IMU), la stima del modulo della velocità angolare risulta sensibilmente minore di quello reale, in quanto cade l'ipotesi del segnale a media nulla ed al segnale, oltre all'offset, viene sottratta parte della sua componente continua.

²Non aumenta la complessità di calcolo, poiché, memorizzando la somma degli N termini è sufficiente ad ogni passo k sottrarre il campione corrispondente all'istante $k - (N - 1)$ e sommare il campione all'istante k , richiedendo così solo 2 operazioni in più indipendentemente da N

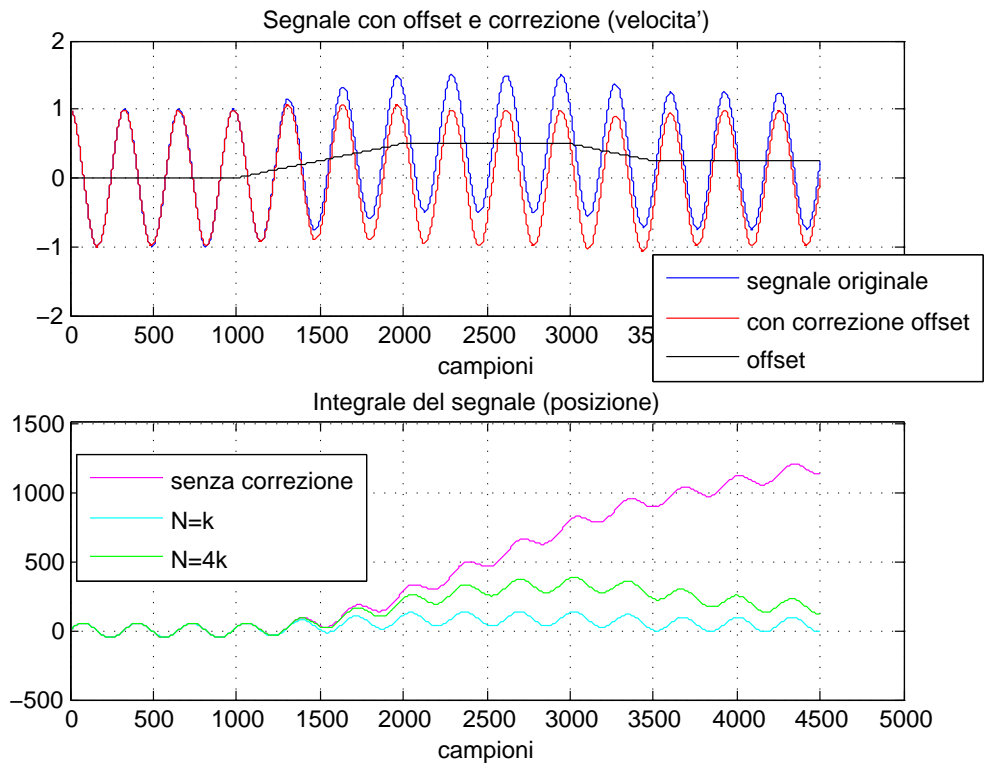


Figura 8.1: Segnale con offset ed effetto della correzione al variare di N .

Se venissero utilizzati solamente i giroscopi per calcolare la rotazione del robot, l'effetto di quest'ultimo problema sarebbe, sul lungo periodo, riportare la stima della rotazione a zero in quanto la componente continua viene interpretata come un offset. Nel filtro IMU presentato, però, l'integrazione dei valori dei giroscopi non avviene sulla stima precedente, ma sulla stima risultante dalla fusione dei dati di giroscopi ed accelerometri (vedi fig. 6.5). L'effetto dei giroscopi sul filtro è dunque quello di migliorare la risposta del filtro alle oscillazioni ad alta frequenza e rallentare la risposta del filtro a segnali con componente continua o frequenze significativamente inferiori a quella del passo, rallentamento inversamente proporzionale al parametro β . A livello pratico, questo vuol dire ad esempio che in risposta ad cambiamento di inclinazione del piano il filtro convergerà al valore corretto in maniera più lenta che attraverso una stima effettuata mediante i soli accelerometri, ma al contempo cambiamenti di inclinazione a frequenze vicine o superiori a quelle

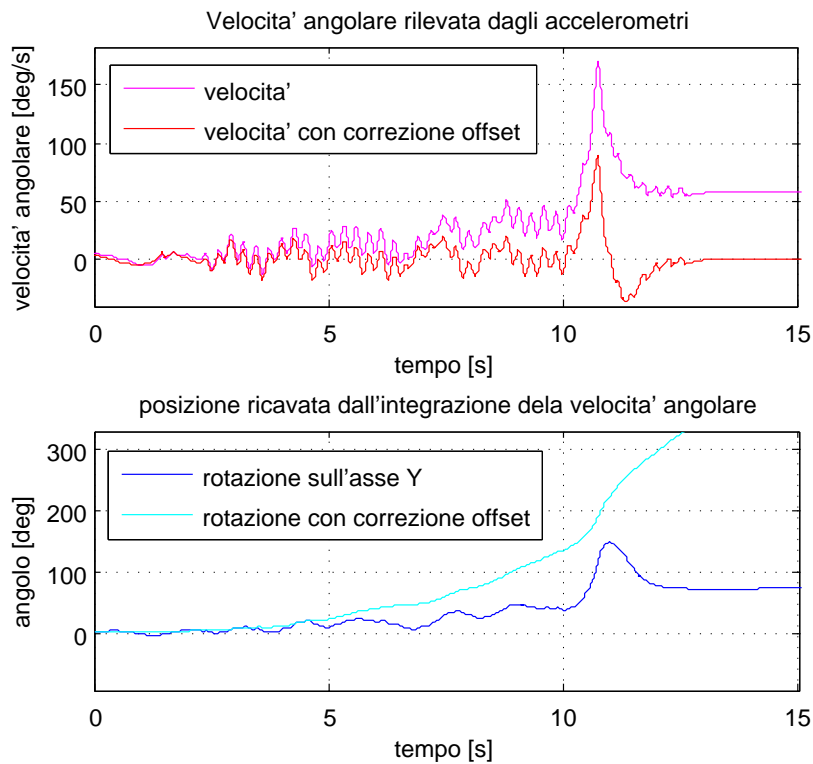


Figura 8.2: Effetto della correzione dell'offset sul segnale reale, rappresentante il beccheggio durante una camminata.

del passo verranno seguiti in maniera più precisa e più pronta. Il valore di N scelto influisce direttamente sia sulla velocità e precisione nell'eliminazione dell'offset, sia sull'attenuazione della componente continua: valori di N piccoli³ rendono la stima dell'offset più veloce, ma aumentano il fenomeno di eliminazione della componente continua. Viceversa, valori di N maggiori diminuiscono il fenomeno di attenuazione della componente continua ma eliminano in maniera meno efficace l'offset. In figura 8.1 si possono vedere dei risultati simulati al variare di N per un ipotetico segnale a media nulla su un periodo di K campioni N a cui è stato sommato un offset variabile.

³Si tenga presente che esiste un valore minimo, la lunghezza di un periodo

8.3 Rilevazione dell'orientazione del piano durante la camminata

Durante qualsiasi tipo di camminata il busto di un robot è soggetto a continui cambi di orientamento e ad accelerazioni. I cambi di orientamento dovuti alle diverse posizioni del robot, gli spostamenti non previsti a causa di giochi ed elasticità della struttura meccanica, gli errori di rilevazione del sensore inerziale dovuti alle accelerazioni a cui il robot è sottoposto vanno a formare un errore che si somma al segnale rappresentante l'inclinazione del piano, rendendone difficoltoso il rilevamento. Procedendo al rilevamento dell'inclinazione del piano con lo stesso metodo esposto nel precedente capitolo ci si troverebbe di fronte alla necessità di effettuare ad ogni aggiornamento il calcolo della cinematica diretta della gamba. Inoltre, dal momento che il robot non possiede sensori di pressione sui piedi, bisognerebbe presupporre di sapere quale piede in un determinato istante è appoggiato a terra e presupporre che tutta la sua superficie sia a contatto con il piano di appoggio; infine il sapere quale piede è appoggiato a terra richiederebbe la conoscenza delle caratteristiche della camminata implementata. Per evitare tutto questo, si è pensato di stimare l'inclinazione del piano solamente a partire dall'inclinazione del busto. Durante una camminata il busto oscillerà sia sull'asse X che sull'asse Y attorno ad una posizione di equilibrio determinata dall'inclinazione del piano. Se consideriamo come segnale l'inclinazione del piano e l'oscillazione come rumore additivo, è possibile utilizzare un filtro di Kalman per determinare l'inclinazione del piano come il valore più probabile date le misure passate. L'approssimazione maggiore è data dal fatto che il rumore additivo non è Gaussiano, come invece sarebbe richiesto dalle ipotesi del filtro di Kalman [10]. La struttura del filtro utilizzato è:

$$\begin{cases} x(k+1) &= Ax(k) + v_x(k) \\ y(k) &= Cx(k) + v_y(k) \end{cases} \quad (8.1)$$

si è considerato il processo $x(k)$ come un processo a due componenti, rappresentanti il valore dell'inclinazione del piano lungo gli assi X e Y, e si è posto $C = I_2$. Per l'inizializzazione del filtro vengono raccolti 200 campioni iniziali a robot immobile, lo stato viene inizializzato ponendolo uguale alla media campionaria; e imponendo nella matrice delle varianze della stima iniziali P_0

$$P = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix} \quad (8.2)$$

p_{z_1} e p_2 uguali alla varianza campionaria. Il risultato è visibile in figura 8.3

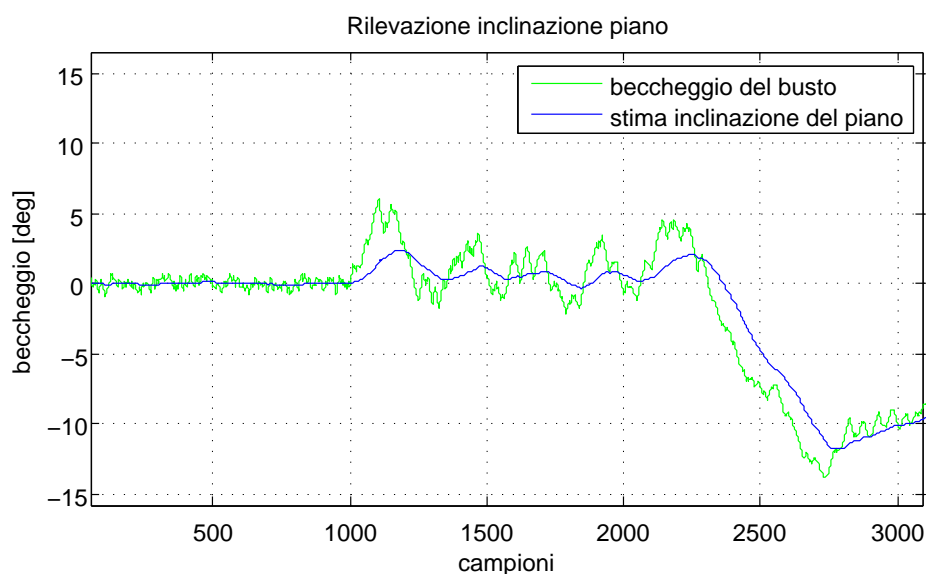


Figura 8.3: Stima dell'inclinazione del piano dai dati dell'inclinazione del busto

8.4 Correzione del beccheggio

Nel caso esaminato la correzione del beccheggio avviene in maniera molto simile a quanto esposto nel capitolo 7, utilizzando lo stesso principio: l'inclinazione delle caviglie viene utilizzata per correggere l'assetto del busto, mentre le braccia vengono utilizzate per lo spostamento del baricentro “a

monte” rispetto all’inclinazione assunta dal piano. Per spostamento della braccia in questo caso si intende lo spostamento del centro di oscillazione: durante la camminata predefinita le braccia oscillano ruotando lungo l’asse Y simmetricamente rispetto alla verticale per contrastare lo spostamento del baricentro dovuto allo spostamento delle gambe. Se il busto del robot si inclina, il centro dell’oscillazione viene spostato in modo da riportare il baricentro verso il centro de poligono di appoggio.

Riguardo ai piedi, in ogni fase della camminata predefinita le caviglie vengono mantenute orizzontali, quindi per adattarle all’inclinazione del piano è sufficiente ruotarle concordemente alla rotazione del piano.

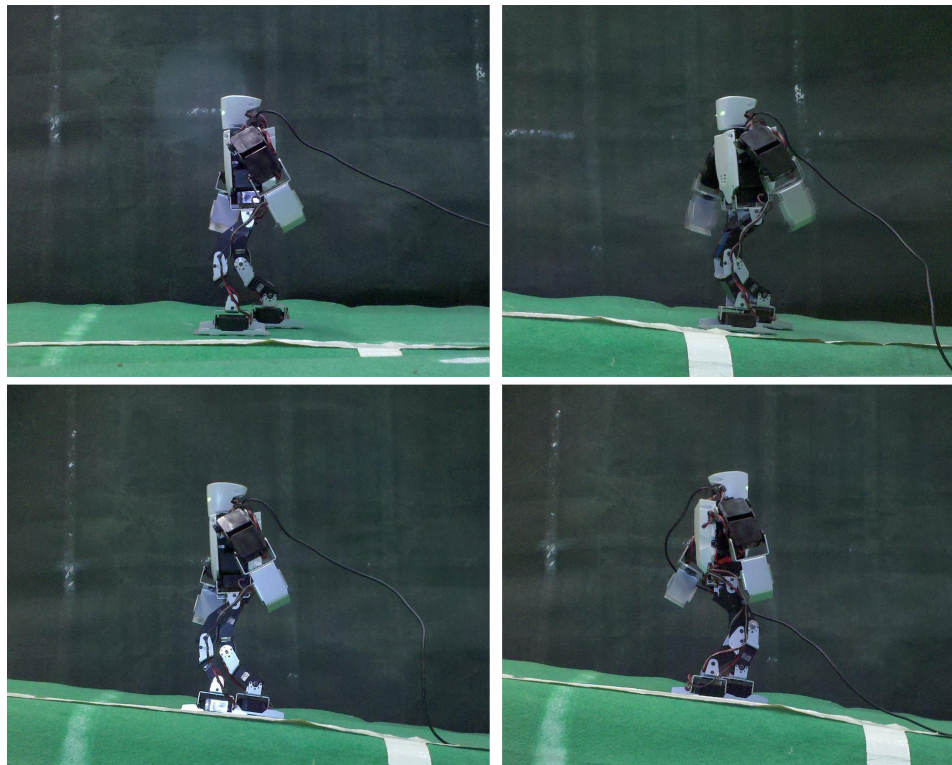


Figura 8.4: Correzione del beccheggio durante la camminata su piano inclinato

8.5 Correzione del rollio

La correzione del rollio richiede un procedura più complicata, in quanto ricade nel tipo di correzioni che abbiamo chiamato “tempo-varianti”. Se il piano viene inclinato ruotando lungo l’asse X, si dovrà alzare il piede a monte per poter mantenere il busto orizzontale, agendo sui giunti di caviglie, ginocchio ed anca. La correzione da applicare in questi giunti, però, varia a seconda della posizione della gamba:

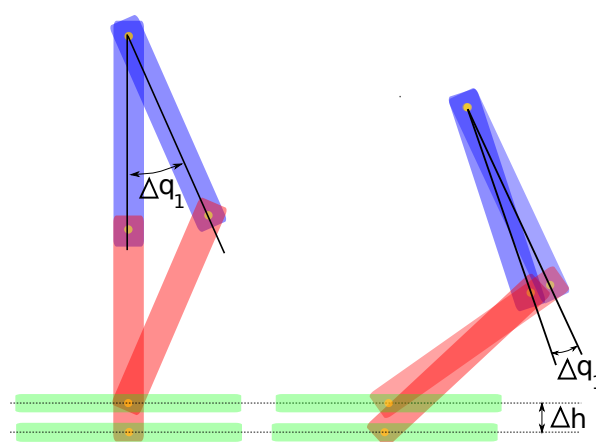


Figura 8.5: Differenti configurazioni richiedono differenti offset per ottenere il medesimo effetto

Determinazione della posizione della gamba

Per poter applicare l’offset è dunque necessario conoscere in che posizione è la gamba, tuttavia l’hardware del Robovie-X non permette di leggere la posizione dei giunti che quindi è necessario stimare. La camminata è costituita da una serie di pose e da dei tempi di esecuzione. Conoscendo il tipo di interpolazione utilizzato è possibile risalire, per ogni istante t , al comando inviato al motore. Per facilitare la sincronizzazione del timer dell’algoritmo con quello della scheda onboard è stata utilizzata una variabile accessoria che chiameremo *status*. Nel programma a blocchi della camminata, creato attraverso la *motion area* di *RobovieMaker* è stato inserito un “blocco operazioni” prima di ogni posa. Questo blocco assegna alla variabile *status* un indice, che

andrà a rappresentare la mossa che è in esecuzione. Questa variabile viene letta dall'algoritmo di camminata implementato sul PC contemporaneamente alla lettura dei giroscopi (la richiesta avviene nello stesso pacchetto) e dunque viene letta con una frequenza molto maggiore rispetto alla frequenza di comando dei motori (200Hz vs 50Hz). La conoscenza della variabile status permette dunque di conoscere il tempo trascorso dall'inizio dell'ultima mossa con una approssimazione⁴ massima di 5ms, indipendente dal tempo trascorso dall'inizio della camminata rendendo robusta la sincronizzazione.

Interpolazione

Scopo dell'interpolazione è raccordare dei punti (che nel caso in esame sono rappresentati dai valori dei giunti nelle diverse pose) con una funzione con certe caratteristiche di continuità. Per stimare correttamente il comando inviato ai motori è necessario conoscere il tipo di interpolazione utilizzato dal software; in *RobovieMaker 2* l'interpolazione predefinita è la *cubic spline* [11]. Una spline di ordine n è una funzione⁵ \mathbb{C}^{n-1} composta da $k - 1$ polinomi di ordine al più n che interpola k punti. Ogni tratto compreso tra due punti è interpolato da un diverso polinomio il cui valore e derivate calcolati negli estremi corrispondono a quelli dei polinomi precedenti e successivi. Interpolare i valori dei giunti nelle diverse pose con una *cubic spline* significa dunque trovare una funzione $S(x)$:

$$S(x) = \begin{cases} s_1(x) & \text{per } x_1 < x < x_2 \\ s_2(x) & \text{per } x_2 < x < x_3 \\ \vdots & \\ s_{n-1}(x) & \text{per } x_{n-2} < x < x_{n-1} \end{cases} \quad (8.3)$$

⁴L'approssimazione è solo per difetto in quanto ovviamente la lettura di un nuovo valore può avvenire solo dopo il cambiamento di posa e cioè in ritardo, mai in anticipo.

⁵cioè continua con derivate continue fino all' $(n - 1)$ -esima

Dove ogni funzione $s_i(x)$ è un polinomio di terzo grado:

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad (8.4)$$

Si tratta dunque di determinare (per ogni giunto) i $4(n - 1)$ coefficienti dei polinomi che formano $S(X)$. Per fare ciò si impongono tre condizioni:

- La funzione $S(x)$ deve passare per i punti da interpolare (quindi $s_i(0) = d_i = x_i$)
- La funzione deve essere continua e con derivate continue (quindi si ha che $s_i(x_i) = s_{i+1}(x_i) \forall i$ e lo stesso per le derivate)
- Condizioni al contorno per i polinomi iniziali e finali

Le condizioni al contorno definiscono il comportamento della funzione ai bordi; essendo il programma della camminata ciclico come condizioni al contorno si imporrà $s_{n-1}(x_i) = s_1(x_1)$ e lo stesso per le derivate. Il calcolo dei coefficienti si riduce alla risoluzione di un sistema lineare di ordine n , per i dettagli sul procedimento del calcolo si rimanda al paper [12].

Correzioni

Una volta ottenuti i valori dei giunti comandati dalla scheda ad un certo istante t , attraverso la cinematica diretta viene calcolata la posizione e l'orientazione del piede. La posizione del piede viene modificata attraverso le correzioni stabilite e i nuovi valori dei giunti ricavati attraverso con il procedimento della cinematica inversa. Una volta ottenute i valori dei giunti che posizionerebbero e orienterebbero il piede nel modo desiderato, sottraendo a questi i valori dei giunti comandati dalla scheda si ottengono gli offset. Anche nel caso del rollio le correzioni in risposta al rollio del piano sono analoghe a quanto presentato nel capitolo 7: le caviglie vengono ruotate di un angolo pari alla rotazione del piano, e viene corretta la differenza di altezza tra le gambe alzando la gamba interna. Per contribuire a mantenere il baricentro all'interno del poligono di appoggio, il braccio "a monte" del robot viene al-

zato ruotandolo di un angolo proporzionale alla rotazione del busto rispetto alla direzione del vettore gravità.

La complessità computazionale di questo algoritmo dipende:

- dal numero i giunti da comandare, $O(n)$
- dal numero di pose che rappresentano la camminata⁶, $O(n^2)$.

Una volta decisa la camminata da adottare non è possibile diminuire la complessità dell'algoritmo diminuendo il numero di pose. È invece immediato diminuire la complessità dell'algoritmo calcolando l'interpolazione e la cinematica inversa per i soli giunti la cui correzione dipenda dalla posizione assunta dal robot: ad esempio l'angolo del giunto dell'asse di rollio delle caviglie non dipende dalla posa assunta dal robot e dunque può essere tralasciato nel calcolo dell'interpolazione, come pure l'angolo del giunto di rollio delle braccia.

8.6 Risultati

Per quanto riguarda la correzione del beccheggio del piano, i risultati sono soddisfacenti. Per valutare l'effettiva efficacia della stabilizzazione della camminata sono stati eseguiti tre tipi di test:

- Camminata su un piano inclinato
- Camminata su un piano la cui inclinazione viene cambiata sollevandolo manualmente
- Camminata passando da un piano orizzontale ad un piano inclinato.

La difficoltà della seconda prova è data soprattutto dall'irregolarità del cambiamento di inclinazione e dalle oscillazioni del piano, in quanto il sollevamento avviene in maniera poco regolare per il fatto di essere effettuato manualmente; la difficoltà del terzo test è dato invece dal cambiamento istantaneo di angolazione del piano.

⁶Interpolare n punti mediante una *cubic spline* significa risolvere un sistema lineare di dimensione n , operazione che ha complessità $O(n^2)$

La camminata predefinita ha reso il robot capace solamente di affrontare, oltre alla normale camminata sul piano, solamente una camminata su di un piano inclinato con inclinazioni inferiori o uguali a 3.5° ; inclinazioni superiori ne provocavano a caduta. Al contrario, la camminata con correzione del beccheggio ha permesso al robot di superare tutti e tre i test, i primi due per angoli anche notevoli (10° o più, vedi tabella in fig. 8.6)

Camminata	predefinita	con correzioni
su piano inclinato	3.5°	10°
su piano inclinabile	3°	12°
prima orizzontale, poi inclinato	-	5°

I principali fattori limitanti per quanto riguarda l'angolo limite raggiungibile relativamente ai primi due test sono la tendenza del robot a scivolare a causa dell'insufficiente attrito col piano e (soprattutto) la tendenza a ruotare in modo imprevedibile sull'asse Z. Questo avviene perché durante la camminata in salita e con le vibrazioni del piano e le oscillazioni naturale del robot un lato del piede si solleva parzialmente da terra, facendo così da perno. Tale effetto è stato mitigato notevolmente aumentando la rigidità del piano inclinato di appoggio. La pendenza massima è minore nel caso del piano inclinato perché il punto dolente è la partenza, che imprime una significativa accelerazione al robot tendendo a sbilanciarlo; al contrario sul piano ad inclinazione variabile quando il piano giunge a inclinazioni limite la camminata è già in situazione di regime e non sono richieste al robot accelerazioni.

In figura 8.6 si possono vedere i risultati delle correzioni del beccheggio applicate; l'istante di partenza della camminata è segnato in rosso. Nel primo grafico è rappresentato il beccheggio della normale camminata sul piano. Nel secondo grafico il robot effettua la camminata predefinita su un piano inclinato di 3° valore vicino alla massima inclinazione affrontabile senza correzioni. In questo caso si vede come il robot abbia la stessa inclinazione del piano finché resta immobile, ma una volta iniziata la camminata tende a "cedere"

alla forza di gravità inclinandosi lentamente sempre di più in modo concorde al piano. Nella terzo grafico la stessa pendenza viene affrontata con l'ausilio delle correzioni, si può notare come il robot parta con il busto verticale e mantenga la verticalità per tutta la camminata. Nel quarto e quinto grafico viene aumentata la pendenza; questo si traduce in un maggior scostamento dalla posizione verticale dovuto all'accelerazione della partenza, che viene comunque corretto e compensato nel tempo. L'ultimo grafico è relativo alla prova del piano ad inclinazione variabile. In questa prova il piano viene inclinato manualmente a partire da un secondo dall'inizio della camminata, fino ad un valore di 12° che rappresenta la massima pendenza affrontabile dal robot. Dal momento che il piano viene sollevato a mano, all'inclinazione si somma il disturbo delle oscillazioni generate dalla mano che solleva il piano. La lentezza della risposta dei motori introdotta appositamente dal filtro FIR limita la velocità di risposta del robot all'inclinazione del piano: infatti in un primo momento il robot si inclina assieme al piano e solo dopo diversi secondi riesce a riportare il busto in posizione verticale. Non è possibile utilizzare parametri più veloci per il filtro FIR in quanto in seguito a disturbi generano instabilità sotto forma di oscillazioni che si amplificano sempre più, soprattutto a causa del fatto che tali oscillazioni sollevano a volte i piedi da terra impedendo al robot di determinare la reale inclinazione del piano.

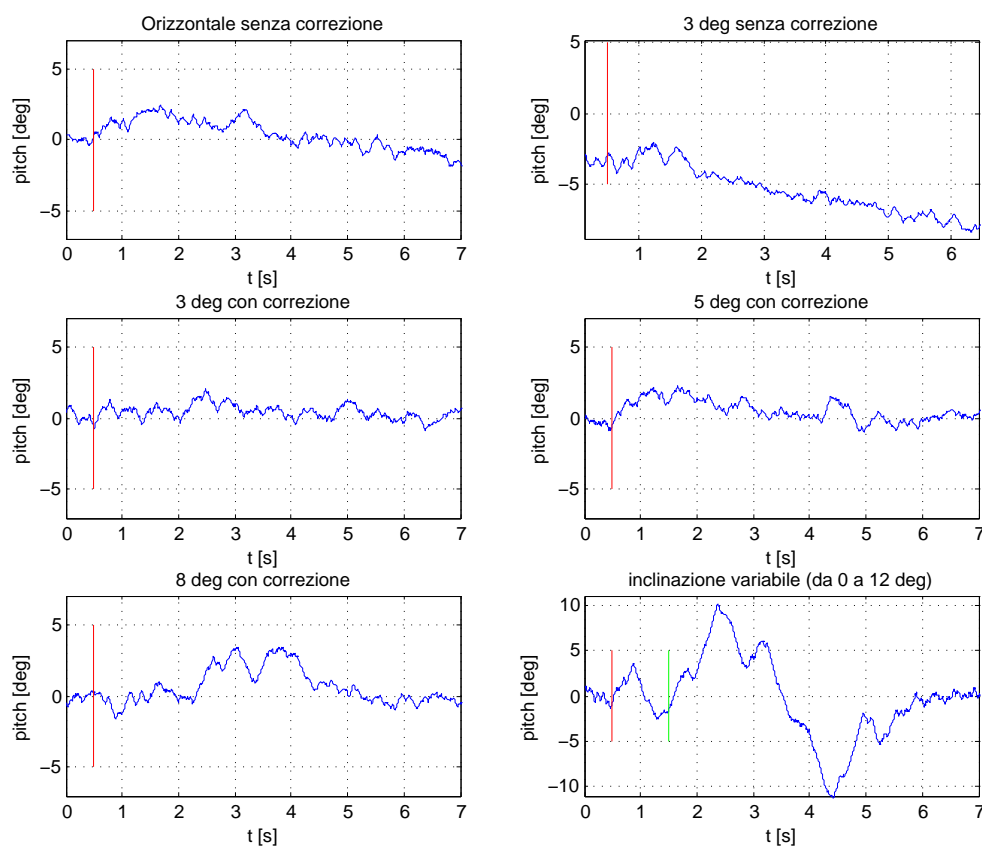


Figura 8.6: Effetti delle correzioni sul beccheggio conseguente all'inclinazione del piano

Capitolo 9

Conclusioni e sviluppi futuri

Partendo dalla conoscenza di alcune funzioni base, durante lo svolgimento di questa tesi sono state determinati vari aspetti del protocollo di comunicazione USB del *Robovie-X* (attraverso l'analisi dei pacchetti inviati da *RobovieMaker 2*), è stato sviluppato un software di comunicazione tra il robot ed il PC ed è stata implementata una camminata "intelligente".

Nonostante la semplicità della dotazione sensoristica del robot è stato possibile implementare delle correzioni di camminata efficienti; la strategia adottata di separare la parte deterministica della camminata (implementata sul robot) dal trattamento dei dati dei sensori inerziali e conseguenti correzioni alla camminata (calcolate sul PC) ha permesso di ottenere una efficace camminata i cui principi possono essere applicati anche a camminate e robot diversi¹ previo tuning dei pochi parametri presenti; senza la necessità di una conoscenza approfondita delle caratteristiche del robot² o della camminata stessa.

Il trattamento dei dati ricavati dal sensore inerziale si è rivelato efficace fornendo una stima dell'orientazione sufficientemente precisa nonostante le limitazioni dei giroscopi (misura delle accelerazioni angolari piuttosto che delle velocità, offset che variano significativamente se sottoposti ad accelerazioni, errori di lettura, mancanza di un giroscopio sull'asse Z). Tali limitazioni si prevede verranno in futuro mitigate dall'adozione di un giroscopio

¹Ma con struttura delle gambe simile

²Peso dei link, velocità e coppia erogabili dai motori, ecc..

visivo implementato mediante una videocamera omnidirezionale³. Grazie alla struttura del filtro IMU presentato le stime del giroscopio visivo potranno essere semplicemente fuse con le stime inerziali usandole in luogo delle stime del sensore magnetico oppure utilizzandolo per l'eliminazione degli offset dei giroscopi.

Le correzioni implementate rappresentano solo uno degli aspetti della stabilizzazione della camminata. La diminuzione delle oscillazioni provocate dalla camminata stessa, ad esempio, richiederebbero delle correzioni che vanno al di là di alcuni offset richiedendo di poter agire direttamente sui parametri della camminata. Il fatto di implementare la camminata sul robot attraverso il software *RobovieMaker 2* rappresenta una limitazione perché il linguaggio utilizzato non permette di interrompere, o variare la velocità di una *motion* e non permette neppure di variare in maniera semplice il flusso logico del programma. Per questo potrebbe essere interessante analizzare le possibilità di programmazione della scheda a bordo del robot per verificare la possibilità di programmarla utilizzando un linguaggio diverso da quello del *RobovieMaker 2*.

Nella progettazione di tutti gli algoritmi si è sempre tenuto conto della necessità di ottenere la più bassa complessità computazionale possibile, nell'ottica di una futura implementazione su un sistema embedded a bordo del robot, sistema che sarebbe necessario anche per l'utilizzo della telecamera come giroscopio visivo.

³Una prototipo è già stato realizzato presso lo stesso laboratorio ed attualmente è in studio un'altra versione dello stesso.

Bibliografia

- [1] T. McGerr, *Passive dynamic walking*.
International Journal of Robotics Research, Vol.9 No.2, 1990.
- [2] T. McGerr, *Passive walking with knees*.
Proc. 1990 IEEE Robotics and Automation Conference, Cincinnati, 1990.
- [3] P.Sardain and G. Bessonnet, *Forces Acting on a Biped Robot. Center of Pressure—Zero Moment Point*.
Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions, 2004.
- [4] J.Hu J.Pratt C.Chew H.Herr and G.Pratt *Adaptive Virtual Model Control of a Bipedal Walking Robot*.
IEEE International Joint Symposia on Intelligence and Systems, 1998.
- [5] R. Bonetto, *Applicativo open source per il controllo del robot RB2000*.
Università degli studi di Padova, 2008.
- [6] A. Casasola, *Realizzazione di un traduttore da un linguaggio pseudo-naturale ad un file di comandi per il robot umanoide Robovie-X*.
Università degli studi di Padova, 2010.
- [7] G. Legnani, *Robotica Industriale*.
Casa editrice Ambrosiana, 2002.
- [8] S. O. Madgwick, *An efficient orientation filter for inertial and inertial/magnetic sensor arrays*.
CONTROLLA Review per IROS 2010, 2010.

- [9] M. Finotto, *Visione omnidirezionale su robot umanoide*.
Università degli Studi di Padova, 1998.
- [10] G. Picci, *Filtraggio Statistico (Wiener, Levinson, Kalman) e applicazioni*.
Edizioni Libreria Progetto Padova.
- [11] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*.
Morgan Kaufmann, San Francisco, 1998.
- [12] S. McKinley and L. Megan, *Cubic Spline Interpolation*.
<http://online.redwoods.cc.ca.us/instruct/darnold/laproj/Fall98/SkyMeg/proj.pdf>,
1998.

Elenco delle figure

1.1	Locomozione attraverso gambe: esempi di robot bipedi, quadrupedi ed esapodi	3
1.2	<i>Uncanny Valley</i> : Relazione tra emozioni suscitate dalla vista di un robot e somiglianza del robot agli umani.	4
1.3	<i>Robonaut 2</i> , umanoide progettato dalla NASA in collaborazione con General Motors per assistere gli astronauti a bordo della ISS	5
1.4	<i>Passive walker</i> ideato da Tad McGeer	6
1.5	Esempio di <i>virtual model control</i> applicato ad un robot bipede con 7 membri, tratto dall'articolo [4]	10
1.6	Macchina a stati per il virtual model control del robot <i>Spring Turkey</i>	11
2.1	Robovie-X, l'umanoide usato in questo lavoro.	14
2.2	Scheda di elaborazione del Robovie-X	15
2.3	Schermata principale del programma <i>Roboviemaker2</i>	16
5.1	Assi di rotazione della gamba del Robovie-X	31
5.2	Traslazione del piede	32
5.3	Cinematica inversa della gamba: a sinistra, schema dei link sul piano frontale; a destra sul piano sagittale	33
6.1	Scheda VS-IX001 in dotazione al Robovie-X	37
6.2	Relazione tra vettori in \mathbb{R}^3 , quaternioni e rotazioni.	44

6.3	Errori di lettura dei sensori inerziali: campioni a fondo scala (figura di sinistra) e campioni a zero (destra).	47
6.4	Grafici relativi alla calibrazione dei giroscopi	56
6.5	Schema del funzionamento del filtro IMU	57
6.6	Angolo di rollio calcolato dal filtro IMU	57
7.1	Struttura del programma	60
7.2	Struttura del ciclo di comando dei motori	62
7.3	Reazione a rotazioni del piano sull'asse Y; nell'immagine di sinistra a correzione avvenuta, nell'immagine a destra durante lo spostamento	64
7.4	Reazione a rotazioni del piano sull'asse X; nell'immagine di sinistra a correzione avvenuta, nell'immagine a destra durante lo spostamento	65
7.5	Robovie-X su un piano inclinato sia rispetto all'asse X che rispetto all'asse Y.	68
7.6	Filtro, risposta al gradino ed ad un generico segnale rumoroso	69
8.1	Segnale con offset ed effetto della correzione al variare di N . . .	74
8.2	Effetto della correzione dell'offset sul segnale reale, rappresentante il beccheggio durante una camminata.	75
8.3	Stima dell'inclinazione del piano dai dati dell'inclinazione del busto	77
8.4	Correzione del beccheggio durante la camminata su piano inclinato	78
8.5	Differenti configurazioni richiedono differenti offset per ottenere il medesimo effetto	79
8.6	Effetti delle correzioni sul beccheggio conseguente all'inclinazione del piano	85