

**UNIVERSITÀ DEGLI STUDI DI PADOVA**

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI  
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCATRONICA

---

*TESI DI LAUREA MAGISTRALE*

**SVILUPPO DI UN SIMULATORE PER SCI  
NAUTICO**

*Relatore:* Prof. Roberto Oboe

*Laureando:* Matteo Miraglia  
1019799-IMC

ANNO ACCADEMICO: 2012-13



## SOMMARIO

---

Lo sci nautico è uno sport acquatico nel quale lo sciatore è trainato da una barca tramite una corda. In questo sport, al contrario di altri come il surf, il vento, le onde e le condizioni atmosferiche avverse ostacolano notevolmente la sciata rendendo praticamente impossibile l'allenamento sul campo. Per queste ed altre motivazioni, come ad esempio il costo degli allenamenti, l'adozione di un simulatore per lo sci nautico, analogo a quelli già progettati in altri campi come quello aeronautico, sarebbe di notevole utilità, oltre ad essere una novità assoluta per questo sport. Ci si è posti perciò l'obbiettivo di realizzare un simulatore per sci nautico utilizzando due componenti fondamentali disposti l'uno sopra all'altro: una pedana dinamometrica dotata di opportuni sensori ottici dai quali è possibile risalire alle forze esercitate dallo sciatore su di essa e alla posizione del suo baricentro, ed una piattaforma di Stewart, cioè un manipolatore di tipo parallelo che ha lo scopo di simulare la dinamica della tavola durante la sciata. Si è riusciti inoltre a ricavare un modello di stato per la sciata rettilinea, ovvero nella direzione di moto del motoscafo, avente lo scopo di determinare la dinamica della tavola a partire dai dati rilevati dalla pedana dinamometrica. Oltre che sulla determinazione di tale modello, durante il lavoro di tesi, ci si è dedicati alla caratterizzazione dei sensori della pedana e al controllo della piattaforma con ottimi risultati. Sono infine state fatte delle prove per valutare le prestazioni dell'architettura realizzata per il simulatore constatando l'ottima capacità di individuazione del baricentro dello sciatore e delle forze da esso esercitate, la possibilità di imprimere i movimenti voluti alla piattaforma di Stewart in base ai dati rilevati e la possibilità di visualizzare in tempo reale la posizione del baricentro e traiettorie di riferimento. In base ai risultati raggiunti, una futura integrazione del modello di stato della sciata, nell'architettura realizzata pare un obiettivo di non difficile realizzazione.



## RINGRAZIAMENTI

---

Desidero fare un particolare ringraziamento al Prof. Roberto Oboe e al Dottor Riccardo Antonello per la disponibilità e l'aiuto datomi per svolgere questa tesi. Ringrazio inoltre tutti i Professori del Corso di Laurea in Ingegneria Meccatronica per avermi trasmesso, con professionalità e dedizione, le loro conoscenze. Infine desidero inoltre ringraziare i miei genitori, mio fratello, i miei parenti e i miei amici per essermi sempre stati vicino durante tutti i miei studi.



## INDICE

---

1	LO SCI NAUTICO	1
1.1	Cos'è lo sci nautico?	1
1.2	Le discipline	2
1.2.1	Lo slalom	2
1.2.2	Le altre discipline classiche	3
1.2.3	Le discipline "moderne"	4
1.3	Esigenza di un simulatore per lo sci nautico	5
1.4	Obiettivi del lavoro di tesi	6
2	MODELLO DELLA SCIATA RETTILINEA	7
2.1	Dalla realtà alla simulazione	7
2.2	Calcolo delle forze di drag e di lift	8
2.3	Determinazione modello per la sciata rettilinea	11
2.3.1	Il sistema sciatore-sci	11
2.3.2	Calcolo della forza e della coppia applicate dallo sciatore	14
2.3.3	Equazioni di moto dello sci e modello di stato	16
2.3.4	Schema a blocchi del simulatore	17
3	SETUP SPERIMENTALE	21
3.1	Architettura del simulatore	21
3.2	La pedana dinamometrica	22
3.2.1	Struttura della pedana	22
3.2.2	Schema Simulink di acquisizione ed elaborazione	24
3.2.3	Caratterizzazione dei sensori orizzontali	29
3.2.4	Calcolo delle forze orizzontali	34
3.2.5	Verifica caratterizzazione con piastre di cemento	34
3.2.6	Verifica caratterizzazione con persona sulla pedana	35
3.3	La piattaforma di Stewart	39
3.3.1	Struttura della piattaforma	39
3.3.2	Cinematica inversa di posizione	40
3.3.3	Schema di controllo della piattaforma	43
3.3.4	Compensazione della "zona morta" e feedforward	47
3.3.5	Verifica della bontà del controllo	52
3.4	Caratterizzazione dinamica verticale	54
3.4.1	Risposta in frequenza ideale	54
3.4.2	Verifica sperimentale della risposta in frequenza	56
3.5	Prova sperimentale preliminare del simulatore	58

3.5.1	Determinazione dei movimenti della piattaforma di Stewart	59
3.5.2	Risultati sperimentali	61
	Conclusioni	71

## APPENDIX 73

A	CODICI MATLAB	75
A.1	Cinematica inversa	75
A.2	Macchina a stati	76
A.3	Determinazione LUT velocità vs tensione	79
A.4	Concatenazione dati LUT	80
A.5	Adattamento LUT	81
A.6	Codice determinazione rigidzze orizzontali	81
A.7	Codice separazione sinusoidi	82
A.8	Codice per salvare ampiezza, fase e frequenza sinusoidi	83
A.9	Codice approssimazione ai minimi quadrati sinusoidi	84
A.10	Codice confronto risposte in frequenza	85
A.11	Codice di visualizzazione del baricentro	85
A.12	Codice di generazione del riferimento	92

## BIBLIOGRAFIA 95

## ELENCO DELLE FIGURE

---

Figura 1	Lo sci nautico.	1
Figura 2	Schema di un campo da slalom.	2
Figura 3	La disciplina dello slalom.	3
Figura 4	La disciplina del salto.	3
Figura 5	La disciplina delle figure.	4
Figura 6	Sci nautico a piedi nudi.	4
Figura 7	Forma dello sci e linea d'acqua.	9
Figura 8	Cambio direzione vettore velocità.	9
Figura 9	Forze esercitate dall'acqua.	10
Figura 10	Il sistema sciatore-sci.	12
Figura 11	Schema a blocchi del simulatore.	18
Figura 12	Architettura del simulatore.	21
Figura 13	La pedana dinamometrica.	23
Figura 14	Posizione dei sensori.	23
Figura 15	Dettaglio di un sensore ottico.	24
Figura 16	Schema di acquisizione/elaborazione.	25
Figura 17	Conversione da tensioni a forze: schema Simu-link.	26
Figura 18	Coordinate degli antivibranti.	28
Figura 19	Disposizione forze orizzontali.	29
Figura 20	Approssimazione ai minimi quadrati.	33
Figura 21	Curve caratteristiche forza vs displacement dei sensori orizzontali.	33
Figura 22	Determinazione forze orizzontali.	34
Figura 23	La piattaforma di Stewart.	40
Figura 24	Schema dei sistemi riferimento della piattaforma.	41
Figura 25	Schema di controllo della piattaforma.	43
Figura 26	Schema di acquisizione delle lunghezze dei cilindri.	44
Figura 27	Schema dell' <i>Homing controller</i> .	45
Figura 28	Schema del <i>Joint controller</i> .	46
Figura 29	Schema di un controllore PID del <i>Joint controller</i> .	46
Figura 30	Schema di condizionamento dei comandi alle servovalvole.	47
Figura 31	Compensazione della Zona Morta.	48
Figura 32	Schema determinazione LUT.	48
Figura 33	Macchina a stati.	49
Figura 34	Posizione cilindro nel tempo.	50
Figura 35	Esempio "LUT velocità vs tensione".	51
Figura 36	Esempio "LUT tensione vs tensione".	52

Figura 37	Bontà inseguimento lungo la direzione verticale a frequenza 0,5Hz. 53
Figura 38	Bontà inseguimento lungo la direzione verticale a frequenza 1,5Hz. 53
Figura 39	Bontà inseguimento lungo la direzione verticale a frequenza 2,5Hz. 54
Figura 40	Schema equivalente del simulatore. 55
Figura 41	Diagramma di Bode della risposta in frequenza. 58
Figura 42	Schema di calcolo dei movimenti della piattaforma di Stewart e di visualizzazione del baricentro per la prova sperimentale del simulatore. 59
Figura 43	Convertitore da $y_G$ a roll. 60
Figura 44	Andamento coordinate baricentro con movimenti sinusoidali. 62
Figura 45	Andamento orientazione piattaforma con movimenti sinusoidali. 62
Figura 46	Andamento posizione del baricentro con movimenti sinusoidali. 63
Figura 47	Andamento coordinate baricentro con movimenti liberi. 63
Figura 48	Andamento orientazione piattaforma con movimenti liberi. 64
Figura 49	Andamento posizione del baricentro con movimenti liberi. 64
Figura 50	Visualizzazione XY. 65
Figura 51	Confronto coordinate con target bidimensionale. 67
Figura 52	Andamento orientazione piattaforma con riferimento bidimensionale. 67
Figura 53	Andamento posizione del baricentro con riferimento bidimensionale. 68
Figura 54	Confronto coordinate con target monodimensionale. 69
Figura 55	Andamento orientazione piattaforma con target monodimensionale. 69
Figura 56	Andamento posizione del baricentro con target monodimensionale. 70

## ELENCO DELLE TABELLE

---

Tabella 1	Valori delle rigidzze equivalenti. 27
-----------	---------------------------------------

Tabella 2	Verifica con piastre: valori della $F_x$ misurati in N. 36
Tabella 3	Verifica con piastre: valori della $F_x$ teorici in N. 36
Tabella 4	Verifica con piastre: valori della $F_y$ misurati in N. 36
Tabella 5	Verifica con piastre: valori della $F_y$ teorici in N. 36
Tabella 6	Verifica con persona: valori delle componenti di forza misurate. 37
Tabella 7	Verifica con persona: valori delle componenti di forza attesi. 37
Tabella 8	Verifica con persona: valori coordinate del baricentro. 38
Tabella 9	Verifica con persona: valori del modulo della forza. 38
Tabella 10	Verifica con persona: valori del momento attorno ad X. 38
Tabella 11	Coordinate dei vertici dell'end-effector nel sistema mobile. 42
Tabella 12	Coordinate dei vertici della base fissa rispetto al sistema fisso. 42



## LO SCI NAUTICO

---

In questo capitolo introduttivo verrà presentato lo sci nautico. Visto che l'obiettivo di questa tesi è quello di realizzare un simulatore per questo sport affascinante ma poco conosciuto, si è ritenuto infatti indispensabile fornirne una breve presentazione focalizzandosi, in particolar modo, sull'attrezzatura richiesta, sulle diverse discipline e soprattutto sulle motivazioni alla base di questo progetto.

### 1.1 COS'È LO SCI NAUTICO?

Lo sci nautico è uno sport acquatico in cui lo sciatore è trainato da un motoscafo tramite una corda in nylon. Un'estremità di tale corda è attaccata direttamente al motoscafo mentre l'altra presenta un'impugnatura in ferro ricoperta di gomma a forma di triangolo, definita *bilancino*. Gli sci sono costituiti con materiali come Honeycomb, fibra di Carbonio, Graphite, Carbon-Kevlar e presentano un'anima interna in poliuretano espanso oppure in legno. Gli attacchi sono in gomma e ne esistono di svariati modelli, più o meno rigidi, poiché non tutti gli sciatori hanno sensazioni uguali con attacchi uguali. Lo sciatore sfrutta la velocità del motoscafo per incrementare la propria e, al contrario del surf, dove il vento e le onde sono sfruttati per muoversi, nello sci nautico sarebbero solamente un ostacolo allo svolgimento della sciata.



Figura 1: Lo sci nautico.

## 1.2 LE DISCIPLINE

Lo sci nautico si divide in tre discipline classiche ognuna con un diverso obiettivo: lo slalom, il salto e le figure. Ci si concentrerà ora principalmente sulla disciplina dello slalom perché è quella per cui è stato progettato il simulatore, essendo anche la più praticata.

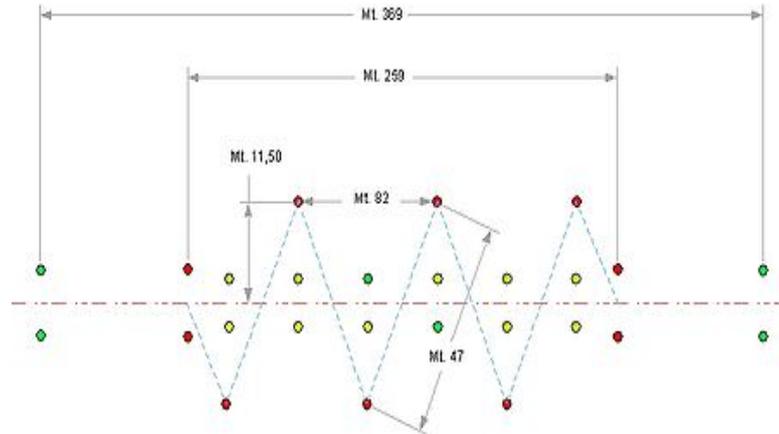
1.2.1 *Lo slalom*

Figura 2: Schema di un campo da slalom.

La struttura e le dimensioni del campo di gara dello slalom sono riportate in fig. 2. Il motoscafo attraversa un corridoio formato da dieci coppie di boe, all'esterno del quale sono posizionate altre sei boe di colore rosso alternate, cioè una da un lato e una dall'altro del corridoio centrale. Lo sciatore deve attraversare le onde del motoscafo da sinistra verso destra passando attraverso la seconda coppia di boe (detta *Gate*), aggirare le sei boe esterne, passando infine attraverso la penultima coppia di boe (detta *Exit*). La prima e l'ultima coppia di boe sono dette *Pre-Gate* e servono solo allo sciatore come riferimento. Lo scopo è quello di aggirare correttamente il maggior numero di boe possibile partendo da una certa velocità (22 km/h) e da una certa lunghezza della corda (18,25 m). Nel caso lo sciatore riesca ad aggirare tutte le boe, si procede, di volta in volta, ad aumentare la velocità del motoscafo di 3 km/h fino al raggiungimento di una velocità massima di 58 km/h per gli uomini dai 15 ai 36 anni e di 55 km/h per le altre categorie. Raggiunta la velocità massima si procede invece all'accorciamento della corda di traino a partire da 18,25 m secondo opportune lunghezze standardizzate. Chiaramente, all'aumentare della velocità e al diminuire della lunghezza della corda, la difficoltà aumenta e vincerà chi sarà arrivato al livello maggiore di difficoltà e, a pari livello, chi avrà aggirato più boe. Lo sci da slalom presenta una parte metallica (di solito di alluminio) di profondità variabile da cinque a otto centimetri, detta *Deriva*, il cui scopo è quello di dare

direzione allo sci ed evitare che la parte posteriore di esso sia libera da vincoli direzionali.



Figura 3: La disciplina dello slalom.

### 1.2.2 *Le altre discipline classiche*

Oltre allo slalom lo sci nautico comprende altre due discipline classiche: il salto e le figure.

Nel salto lo sciatore esegue il taglio della scia del motoscafo con lo scopo di arrivare alla massima velocità possibile ad affrontare un trampolino posto sulla sinistra della barca. Si aggiudica la vittoria lo sciatore che riesce ad atterrare alla distanza maggiore dal trampolino: al contrario del salto con lo sci da neve, non vi sono infatti giudizi tecnici sull'esecuzione del salto.



Figura 4: La disciplina del salto.

Le figure costituiscono invece la "danza" dello sci nautico. Lo sciatore deve eseguire una serie di acrobazie (dette appunto figure) in due passaggi da venti secondi l'uno con la velocità e la lunghezza della corda che preferisce. Non vi è, come accade in praticamente tutti gli altri sport di questo tipo, un giudizio sull'esecuzione della figura, ma la giuria giudica solamente se ciascuna figura è accreditabile o meno.



Figura 5: La disciplina delle figure.

### 1.2.3 Le discipline "moderne"

Oltre alle classiche, vi sono poi altre due discipline degne di nota e facenti parte dello sci nautico. Il *Wakeboard* è paragonabile allo Snowboard per lo sci nautico. Lo sciatore deve eseguire cinque figure in venti metri per due volte con una tavola più lunga di una normale tavola da figure e dotata di due derive ai capi per garantire maggiore stabilità e tenuta di direzione. Al termine del secondo passaggio deve eseguire inoltre un *Double-up* cioè una figura eseguita su un'onda molto più alta del normale grazie ad un'opportuna manovra eseguita dal pilota del motoscafo. In questo caso, al contrario che nelle normali figure, si tiene conto anche dell'eleganza ed altezza di ciascuna acrobazia e non solo della sua difficoltà.

Infine è possibile praticare lo sci nautico anche senza l'ausilio degli sci ma semplicemente usando i piedi come superficie di scivolamento sull'acqua. Tale specialità, che si divide sempre in slalom, salto e figure, è particolarmente spettacolare perché, essendo i piedi più piccoli degli sci, si riescono a raggiungere velocità più elevate.



Figura 6: Sci nautico a piedi nudi.

### 1.3 ESIGENZA DI UN SIMULATORE PER LO SCI NAUTICO

Il termine simulazione identifica una serie di metodi e applicazioni che consentono di imitare o rappresentare i sistemi reali. Essa cerca di costruire un dispositivo sperimentale che, a partire da un modello della realtà che consente di valutare e prevedere la dinamica di una serie di eventi o processi susseguenti all'imposizione di date condizioni, restituisca un comportamento del veicolo e una percezione sensoriale all'utente uguale alle condizioni reali. Nell'ambiente dello sci nautico un simulatore sarebbe di grande utilità per una serie di motivazioni:

- l'allenamento in acqua risulta molto più dispendioso rispetto ad uno con l'utilizzo di un simulatore a causa del costo del carburante, delle risorse umane impiegate, dell'usura delle attrezzature;
- temporali e giornate ventose sono d'ostacolo all'allenamento sul campo perché portano alla formazione di onde che, come visto in precedenza, ostacolano la sciata;
- d'inverno l'allenamento all'esterno è reso difficoltoso dalle basse temperature;
- il simulatore permette di ricreare le condizioni perfette per un buon allenamento;
- con il simulatore è possibile scomporre, analizzare ed allenare determinati movimenti in specifiche condizioni.

Chiaramente un simulatore non potrà mai sostituire completamente un adeguato allenamento sul campo in quanto non permetterà mai una rappresentazione perfetta della realtà, ma solo una modellazione di essa soggetta ad una serie di ipotesi semplificative. Esso dovrà perciò essere usato con lo scopo di migliorare le prestazioni in specifiche condizioni e come strumento per non dover sospendere del tutto gli allenamenti in condizioni meteo inadeguate.

Il simulatore che si è iniziato a realizzare in questa tesi, oltre che dalle motivazioni precedentemente riportate, trae importanza anche dal fatto di essere una novità per lo sci nautico. Infatti, mentre in altri campi come l'aeronautica, i simulatori sono già largamente impiegati, attualmente non si conoscono in letteratura tentativi di simulazione simili a quello proposto in questa tesi. Vi sono infatti solamente alcuni brevetti americani di tentativi di simulazione ma esclusivamente meccanica e molto datati.

Infine, ulteriore importanza al progetto di tesi è data anche dal fatto che lo stesso simulatore potrebbe essere riadattato per simulare anche altri sport, come ad esempio la discesa in Snowboard.

## 1.4 OBIETTIVI DEL LAVORO DI TESI

Lo scopo del progetto di tesi, che prosegue quanto già iniziato in [1] presso l'Università di Trento, con la collaborazione della quale è stato redatto questo progetto, è quello di realizzare un simulatore per sci nautico ed in particolare per la disciplina dello slalom. Per raggiungere tale finalità, i dati sulla forza esercitata dallo sciatore su di una pedana dinamometrica e quelli sulla posizione del suo baricentro, verranno utilizzati per movimentare opportunamente una piattaforma di Stewart (sopra la quale è montata la pedana sulla quale sale lo sciatore) con lo scopo di far avvertire allo sciatore sensazioni quanto più simili possibili a quelle che avrà durante l'allenamento in acqua. Dopo aver introdotto brevemente lo sci nautico e le sue diverse discipline all'interno del capitolo 1, nel capitolo 2 verrà ricavato un modello di stato per il caso semplificato della sciata rettilinea (cioè quella in cui lo sciatore rimane sempre orientato nella direzione di moto del motoscafo) il quale, tenendo conto delle diverse componenti di forza che agiscono sullo sciatore e sullo sci, ha lo scopo di legare i movimenti della piattaforma di Stewart agli spostamenti del peso del corpo dello sciatore sulla pedana riproducendo, in modo quanto più vicino alla realtà possibile la reale dinamica della sciata. Infine, nel corso del capitolo 3, verrà analizzato il setup sperimentale che si intende utilizzare per realizzare il simulatore, soffermandosi in particolare sui seguenti punti:

- descrizione dell'architettura generale del simulatore;
- descrizione della pedana dinamometrica e dello schema Simulink per il calcolo delle componenti di forza esercitate dallo sciatore sullo sci e della posizione del suo baricentro a partire dai dati inviati in tempo reale dai sensori posti sulla pedana;
- caratterizzazione statica dei sensori;
- descrizione della piattaforma di Stewart e del suo schema di controllo in Simulink;
- verifica della bontà del controllo effettuato;
- caratterizzazione dinamica del simulatore in direzione verticale;
- prove sperimentali dell'architettura realizzata.

## MODELLO DELLA SCIATA RETTILINEA

---

Nel capitolo 1 della tesi si sono presentate le diverse discipline dello sci nautico e si è vista l'esigenza di un simulatore per questo sport. E' chiaro però che, nel momento in cui si passa dalla realtà alla simulazione, si è costretti a fare delle semplificazioni: da un lato perché ci sono dei limiti legati a ciò che è disponibile in laboratorio e agli spazi ristretti in cui si è costretti a lavorare, dall'altro perché la realtà è troppo complessa per riprodurla tale e quale e sarà perciò necessario trovare un modello semplificato che cerchi di considerare le principali forze in gioco allo scopo di far percepire allo sciatore delle sensazioni quanto più simili possibile a quelle che realmente avverte durante la sciata. Si cercherà perciò ora di comprendere meglio quali gesti atletici ci si è posti l'obbiettivo di simulare, e, in tali situazioni, quali siano le forze in gioco, le semplificazioni necessarie, le equazioni che governano la dinamica che si cerca di riprodurre con il simulatore.

### 2.1 DALLA REALTÀ ALLA SIMULAZIONE

Nel corso del capitolo 1 si è visto come lo sci nautico si divida in tre discipline classiche: lo slalom, il salto e le figure. Si è anche detto come il simulatore oggetto di questa tesi cerchi di riprodurre esclusivamente la disciplina dello slalom. Il motivo è ovviamente quello che nelle altre due discipline si verifica il distacco dello sci dalla superficie dell'acqua, cosa che risulterebbe alquanto proibitiva da simulare; inoltre nelle figure la dinamica sarebbe alquanto complicata da riprodurre. La scelta ricade dunque sulla disciplina più semplice da simulare, ma che è anche quella maggiormente praticata. Tuttavia, anche ricreare un'intera gara di slalom non risulta poi così banale. In particolare l'aggiramento delle boe risulta alquanto difficile da riprodurre in laboratorio principalmente per i seguenti motivi:

- le curve sono molto strette e presentano quindi una dinamica eccessivamente complicata da modellare;
- durante le curve la posizione dello sciatore risulta molto laterale rispetto al motoscafo ed in laboratorio non c'è abbastanza spazio per simulare una situazione come questa.

Si è deciso dunque di concentrarsi sulla fase di taglio della scia del motoscafo durante la quale lo sciatore si porta da una parte all'altra della barca per poi aggirare la successiva boa. Un primo passo verso questo obbiettivo è quello di formulare un modello che consideri un

caso ulteriormente semplificato, ma comunque non banale da trattare: il caso di sciata rettilinea, cioè senza spostarsi a lato del motoscafo e senza ruotare lo sci attorno agli assi ortogonali alla direzione di moto (pitch e yaw nulli). Si cercherà di illustrare ora quali sono le forze in gioco in tale situazione allo scopo di individuare un modello opportuno per descrivere tale gesto. Vista poi l'assenza momentanea di un sistema per simulare il bilancino e la tensione della corda, si valuterà anche il caso di sciata libera (sempre rettilinea) come punto di partenza per lo studio di quella con traino.

## 2.2 CALCOLO DELLE FORZE DI DRAG E DI LIFT

Si vogliono innanzitutto calcolare le forze che l'acqua esercita sullo sci in direzione parallela (forza di drag o resistenza) e perpendicolare (forza di lift o portanza) a quella di moto del motoscafo. Per fare questo si sono fatte una serie di ipotesi preliminari, essenziali per semplificare la trattazione del problema anche se non sempre del tutto veritiere. Esse verranno ora elencate e commentate.

- La superficie dell'acqua è piatta. Anche se non lo potrà mai essere completamente, è vero anche che questa sarebbe la condizione ideale per questo sport dove si è visto che le onde sono d'intralcio alla sciata.
- Lo sci è di forma rettangolare. La fig. 7 mostra una geometria reale dello sci e una sua approssimazione rettangolare. Le dimensioni del rettangolo sono scelte in modo da mantenere la stessa area. Ovviamente quando lo sci è meno immerso (in genere ad alte velocità) l'errore commesso per l'area immersa calcolata è maggiore rispetto all'area immersa reale perché lo sci vero è più stretto al posteriore.
- Solo una parte dello sci è immersa in acqua ed è perciò necessario calcolare la cosiddetta "linea d'acqua" che separa aria ed acqua. Essa, come si vede in fig. 7, risulta in generale inclinata a causa del rollio (rotazione attorno all'asse  $x_b$ ) dello sci. In questo lavoro si considererà però un modello unidimensionale dove il rollio è trascurato (non vi è rotazione attorno ad  $x_b$ ) e la linea d'acqua è perciò orizzontale.
- Si trascurano le forze aerodinamiche dovute all'aria che investe la parte emersa dello sci.

Sotto le ipotesi elencate, è possibile calcolare le forze di drag ( $F_x$ ) e lift ( $F_z$ ) utilizzando il principio di conservazione dell'energia riportato nell'equazione 1.

$$m(t) * \frac{d}{dt} \left( \begin{matrix} v_x(t) \\ v_z(t) \end{matrix} \right) = \begin{matrix} F_x \\ F_z \end{matrix} \quad (1)$$

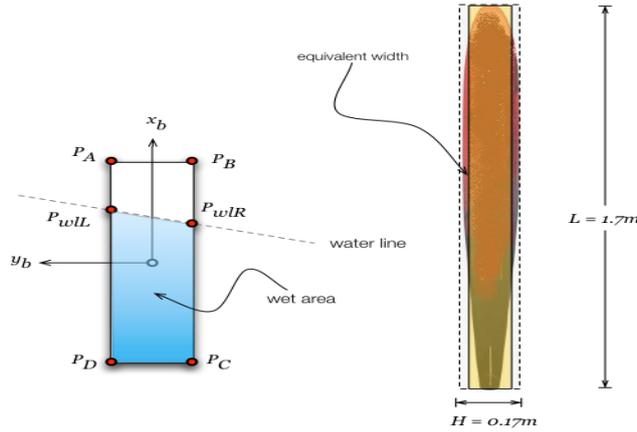


Figura 7: Forma dello sci e linea d'acqua.

In tale equazione  $m$  è la massa d'acqua che investe lo sci. Nell'applicarla si considera inoltre che la velocità dell'acqua sia costante in modulo e cambi solamente direzione nel momento in cui il flusso d'acqua entra in contatto con lo sci come si vede in fig. 8. Le due forze si considerano applicate nel baricentro dell'aria bagnata, anche se chiaramente sarebbero distribuite sulla superficie bagnata dello sci.

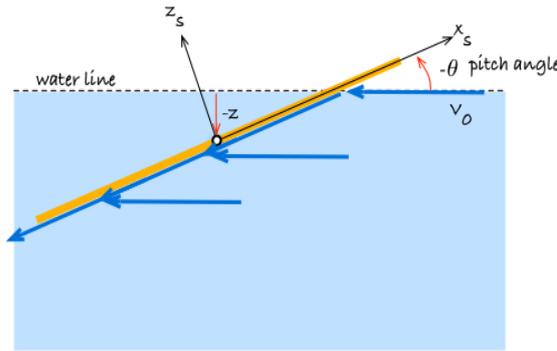


Figura 8: Cambio direzione vettore velocità.

Si considerano, da ora in poi, due sistemi di riferimento: uno fisso, indicato con il pedice "b" (cioè base) in fig. 7, e uno mobile in modo solidale con lo sci, indicato con il pedice "s" (cioè sci) in fig. 8. Di entrambi sono indicati solamente due assi su tre, ma il verso del terzo si ottiene considerando che sono entrambi destrorsi. Il passaggio dal sistema fisso a quello mobile si ottiene con la matrice di trasformazione  $T_s^b$  riportata in 2: il riferimento mobile risulta quindi traslato nella direzione  $z_b$  di  $-z$  e ruotato attorno a  $y_b$  di  $-\theta$  rispetto al fisso.

$$T_s^b = \begin{bmatrix} \cos(\theta(t)) & 0 & -\sin(\theta(t)) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta(t)) & 0 & \cos(\theta(t)) & -z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

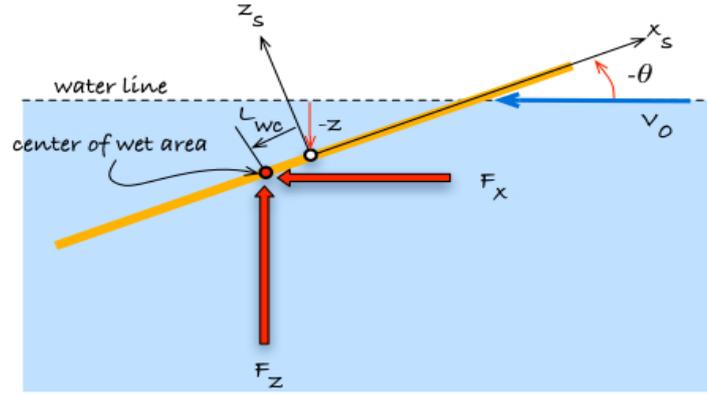


Figura 9: Forze esercitate dall'acqua.

Si faccia invece ora riferimento alla fig. 9 per il calcolo dei vari termini che compaiono nell'equazione 1. In essa  $z$  misura l'affondamento del baricentro geometrico dello sci, mentre  $\theta(t)$  è l'angolo di inclinazione dello sci rispetto alla superficie dell'acqua. L'area bagnata WA è data dall'equazione 3: essa è zero quando lo sci è completamente emerso ed è pari all'area del rettangolo quando lo sci è completamente immerso; negli altri casi è data dalla terza espressione. Con  $H_{ws}$  e  $L_{ws}$  sono indicate rispettivamente larghezza e lunghezza dello sci.

$$WA = \begin{cases} 0 & \text{se } z(t) \leq -\frac{1}{2}L_{ws}\sin(\theta(t)), \\ H_{ws}L_{ws} & \text{se } z(t) \geq \frac{1}{2}L_{ws}\sin(\theta(t)), \\ H_{ws} \left( \frac{z(t)}{\sin(\theta(t))} + \frac{1}{2}L_{ws} \right) & \text{altrimenti} \end{cases} \quad (3)$$

Tale area permette di calcolare la portata d'acqua  $Q$  in  $\text{m}^3/\text{s}$  che infatti è il prodotto dell'area bagnata per il vettore velocità  $V_0$  ortogonale all'area. Vale cioè l'equazione 4.

$$Q = WA * V_0 * \sin(\theta(t)) \quad (4)$$

Considerando, a questo punto, che il vettore velocità cambia direzione ma non modulo (come indicato in fig. 8), si possono calcolare le variazioni nel tempo di tale vettore e, applicando poi l'equazione 1, calcolare le forze di drag e lift che sono le componenti della forza  $F_W$  esercitata dall'acqua rispettivamente in direzione  $x_b$  e  $z_b$ . La componente in direzione  $y_b$  risulterà invece nulla come si vede dalle equazioni 5 dove  $\rho$  è la densità dell'acqua e il segno meno è dovuto al fatto che il vettore velocità ha componenti con verso negativo rispetto agli assi  $x_b$  e  $z_b$ .

$$F_W = \begin{Bmatrix} F_{Wx}(t) \\ F_{Wy}(t) \\ F_{Wz}(t) \end{Bmatrix} = -Q * \rho * \begin{Bmatrix} V_0(1 - \cos(\theta(t))) \\ 0 \\ -V_0\sin(\theta(t)) \end{Bmatrix} \quad (5)$$

Oltre ad esercitare una forza sullo sci, l'acqua produce anche un momento di beccheggio, cioè attorno all'asse  $y_b$ . Per calcolarlo bisogna individuare il centro dell'aria bagnata dove si immagina venga applicata l'intera forza dell'acqua, sebbene in realtà essa sia una forza distribuita. Il punto è il baricentro geometrico dell'area immersa ed è dato dalla coordinata  $L_{wc}$  nel sistema di riferimento dello sci (vedi fig. 9). Al limite potrebbe muoversi fino al bordo inferiore quando lo sci è completamente emerso oppure nel baricentro geometrico dello sci quando lo sci è completamente immerso. In pratica esso si ottiene con le equazioni 6.

$$L_{wc} = \begin{cases} -\frac{1}{2}L_{ws} & \text{se } z(t) \leq -\frac{1}{2}L_{ws}\sin(\theta(t)), \\ 0 & \text{se } z(t) \geq \frac{1}{2}L_{ws}\sin(\theta(t)), \\ \frac{1}{2}\frac{z(t)}{\sin(\theta(t))} - \frac{1}{4}L_{ws} & \text{altrimenti} \end{cases} \quad (6)$$

Il momento  $T_W$  è quindi dato semplicemente dal prodotto vettoriale fra il vettore che unisce il baricentro geometrico dello sci a quello dell'aria bagnata (che ha modulo  $L_{wc}$ ) e la forza dell'acqua  $F_W$ , come mostra l'equazione 7.

$$T_W = L_{wc} \times F_W \quad (7)$$

### 2.3 DETERMINAZIONE MODELLO PER LA SCIATA RETTILINEA

Si cercherà ora di ricavare le equazioni che governano la sciata rettilinea, allo scopo di riprodurne la dinamica e ricavarne un modello che possa essere utilizzato per la realizzazione del simulatore.

#### 2.3.1 Il sistema sciatore-sci

Innanzitutto è necessario capire quali sono le forze e le coppie che agiscono sullo sci e sullo sciatore e quali forze i due corpi considerati rigidi si scambiano. Tali forze e coppie serviranno per ricavare le equazioni di moto che descrivono il gesto della sciata rettilinea. Esse sono riportate in fig. 10.

Lo sci è controllato con le forze esercitate dallo sciatore su di esso: esse sono indicate con  $FS_x$  e  $FS_z$ . Queste forze sono forze interne del sistema sciatore-sci e di fatto lo sciatore controlla lo sci attraverso la posizione del suo baricentro. Infatti si vedrà che le forze interne scambiate tra sci e sciatore sono funzioni della posizione del baricentro dello sciatore, del tiro esterno  $T$  e delle forze d'inerzia. Poiché, come si vedrà in seguito, sarà possibile misurare solamente le forze scambiate, queste conterranno il contributo della forza peso, delle forze d'inerzia e del tiro  $T$  (se presente). Ipotizzando che il rollio sia nullo, non vi è alcuna forza scambiata in direzione  $y_s$ , mentre vi sarà un momento di beccheggio  $TS_y$  che sarà calcolato usando la posizione

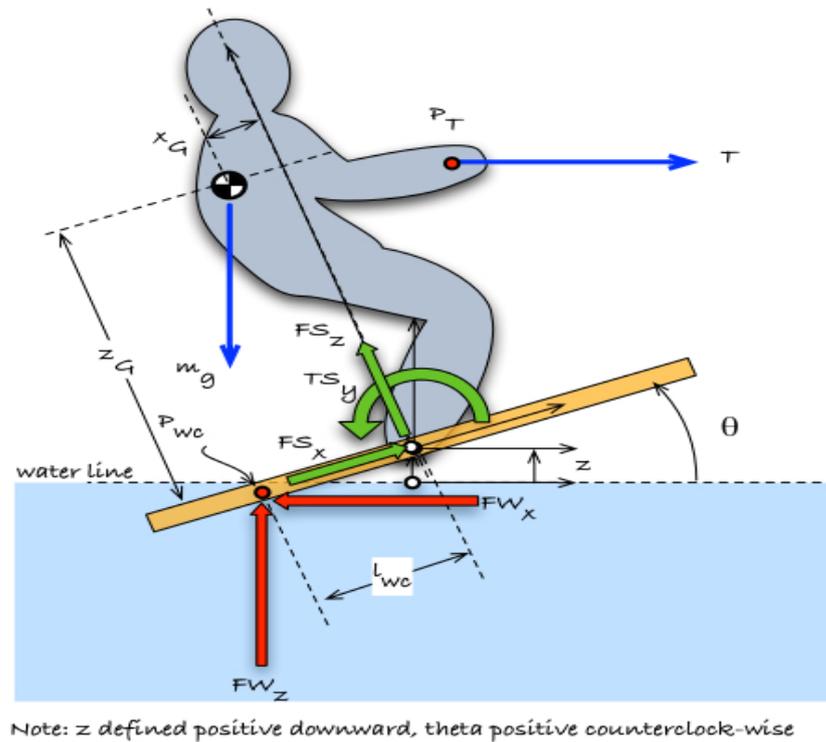


Figura 10: Il sistema sciatore-sci.

del baricentro. Le forze esterne che agiscono sul sistema sciatore-sci saranno invece le forze di drag  $FW_x$  e lift  $FW_z$  precedentemente calcolate e il tiro  $T$  che saranno equilibrate dalle forze di inerzia, dal peso dello sci ( $M_b * g$ ) e da quello dello sciatore ( $m_g = M * g$ ). I sistemi di riferimento considerati saranno sempre quello fisso dello sciatore (pedice  $b$ ) e quello mobile solidale allo sci (pedice  $s$ ) precedentemente illustrati: tuttavia il riferimento mobile sarà spostato rispetto a quello fisso anche in direzione  $x_b$  di una quantità  $x$  dipendente dal tempo. Inoltre l'asse  $z_b$  è positivo verso il basso, mentre  $z_s$  ha il verso che si ottiene ruotando il sistema fisso di  $\theta$ , dove  $\theta$  è di conseguenza considerato positivo in senso antiorario. Le coordinate del baricentro dello sciatore  $G$  nel sistema mobile vengono indicate con  $x_G$  e  $z_G$ , quelle del punto di applicazione della forza di tiro  $P_T$  con  $x_T$  e  $z_T$ . Infine con  $l_{wc}$  si indica la posizione del baricentro dell'aria bagnata  $P_{wc}$  sempre nel sistema mobile.

Si possono immediatamente calcolare le componenti di accelerazione del baricentro dello sciatore nel sistema mobile lungo gli assi  $x_s$  e  $z_s$  (rispettivamente indicate con  $AS_x$  e  $AS_z$ ) date dalle formule 8 e 9. Esse sono la somma di quattro componenti, ovvero: l'accelerazione centrifuga, quella di Coriolis e quelle dovute alla traslazione della

tavola lungo  $x_b$  e  $z_b$ .

$$\begin{aligned} AS_x(t) = & - \left( \frac{d}{dt} \theta(t) \right)^2 x_G - \left( \frac{d^2}{dt^2} \theta(t) \right) z_G \\ & - \sin(\theta(t)) \left( \frac{d^2}{dt^2} z(t) \right) + \left( \frac{d^2}{dt^2} x(t) \right) \cos(\theta(t)) \end{aligned} \quad (8)$$

$$\begin{aligned} AS_z(t) = & - \left( \frac{d}{dt} \theta(t) \right)^2 z_G + \left( \frac{d^2}{dt^2} \theta(t) \right) x_G \\ & - \sin(\theta(t)) \left( \frac{d^2}{dt^2} x(t) \right) - \left( \frac{d^2}{dt^2} z(t) \right) \cos(\theta(t)) \end{aligned} \quad (9)$$

Si scrivono ora le equazioni di equilibrio di Newton alle traslazioni lungo gli assi  $x_b$  (equazione 10) e  $z_b$  (equazione 11) e quella di equilibrio alle rotazioni di Eulero attorno ad  $y_b$  (equazione 12) per il sistema complessivo sciatore-sci: chiaramente le forze interne scambiate fra sci e sciatore non compaiono in tali equazioni.  $IY_b$  è il momento d'inerzia dello sci rispetto all'asse  $y_b$  che è dato dalla formula 13.

$$\begin{aligned} T(t) = & (-\sin(\theta(t))AS_z(t) + AS_x(t)\cos(\theta(t)))M \\ & + M_b \left( \frac{d^2}{dt^2} x(t) \right) - FW_x(t) \end{aligned} \quad (10)$$

$$\begin{aligned} FW_z(t) = & (\sin(\theta(t))AS_x(t) + AS_z(t)\cos(\theta(t)) + g)M \\ & - M_b \left( \frac{d^2}{dt^2} z(t) \right) + M_b * g \end{aligned} \quad (11)$$

$$\begin{aligned} & (-\sin(\theta(t))x_T - \cos(\theta(t))z_T)T(t) + (\sin(\theta(t))z_G \\ & - \cos(\theta(t))x_G)m_g + (AS_x(t)z_G - AS_z(t)x_G)M \\ & - \sin(\theta(t))l_{wc}FW_x(t) + \cos(\theta(t))l_{wc}FW_z(t) \\ & - IY_b \left( \frac{d^2}{dt^2} \theta(t) \right) = 0 \end{aligned} \quad (12)$$

$$IY_b = \frac{1}{12} * L_{ws} * H_{ws}^3 \quad (13)$$

Dall'equazione 10 si osserva che la forza longitudinale  $T$  serve a vincere la resistenza dell'acqua e le forze d'inerzia dello sciatore (quelle della tavola sono trascurabili). L'equazione 11 ci dice invece che la forza verticale esercitata dall'acqua equilibra il peso e le forze d'inerzia verticali (ad esempio quando lo sciatore si alza e abbassa o il contributo di quando si inclina avanti e indietro). L'equazione 12 governa

infine l'orientazione dello sci. Come si osserva essa dipende dalla posizione del baricentro dello sciatore, dall'effetto della tensione della fune e dalle forze d'inerzia. Il tutto si equilibra con le forze dell'acqua. Nel caso la posizione di sci e sciatore fosse fissa e il motoscafo avanzasse a velocità costante le tre equazioni di equilibrio assumerebbero la forma riportata in 14.

$$\left\{ \begin{array}{l} -T(t) - FW_x(t) = 0 \\ m_g + M_b * g - FW_z(t) = 0 \\ (-\cos(\theta)x_G + \sin(\theta)z_G)m_g \\ + (FW_z(t)\cos(\theta) - FW_x(t)\sin(\theta))l_{wc} \\ -T(t)z_T\cos(\theta) - T(t)x_T\sin(\theta) = 0 \end{array} \right. \quad (14)$$

Si osserva che la tensione  $T$  bilancia la forza di resistenza dell'acqua; la forza verticale (portanza) bilancia il peso di sciatore e sci.

### 2.3.2 Calcolo della forza e della coppia applicate dallo sciatore

Dopo aver capito quali sono gli equilibri di forze e coppie che si creano durante la sciata rettilinea, si cercherà ora di ricavare il momento e le forze che lo sciatore esercita sullo sci che sono le forze interne che si sarà in grado di rilevare e fungeranno perciò da ingresso al modello che dovrà riprodurre la dinamica della sciata. Le equazioni di moto dello sciatore ci permettono di ricavare il momento da esso applicato allo sci in base alla posizione nota del suo baricentro e le componenti della forza che egli scambia con lo sci lungo gli assi del sistema di riferimento mobile. Tali forze equilibrano la tensione della fune, la forza peso e le forze d'inerzia dovute al movimento della massa dello sciatore. Si scrivono innanzitutto le equazioni di equilibrio di Newton lungo gli assi  $x_s$  (equazione 15) e  $z_s$  (equazione 16) del sistema mobile e quella di equilibrio alle rotazioni di Eulero attorno all'asse  $y_s$  (equazione 17) per il solo sciatore.

$$M\sin(\theta(t))g + M * AS_x(t) - T(t)\cos(\theta(t)) + FS_x(t) = 0 \quad (15)$$

$$M\cos(\theta(t))g + M * AS_z(t) + \sin(\theta(t))T(t) + FS_z(t) = 0 \quad (16)$$

$$\begin{aligned} & ((\sin(\theta(t)))z_G - \cos(\theta(t))x_G)m_g + \\ & + (AS_x(t)z_G - AS_z(t)x_G)M \\ & - \sin(\theta(t))T(t)x_T - T(t)\cos(\theta(t))z_T + TS_y(t) = 0 \end{aligned} \quad (17)$$

Dalle equazioni 15 e 16 si ricavano a questo punto le componenti di accelerazione del baricentro dello sciatore per poi sostituirle nella

terza dalla quale si può ricavare il momento esercitato dallo sciatore che è quindi dato dall' equazione 18

$$\begin{aligned} TS_y(t) = & (-\sin(\theta(t))x_G + \sin(\theta(t))x_T - \cos(\theta(t))z_G \\ & + \cos(\theta(t))z_G)T(t) + FS_x(t)z_G - FS_z(t)x_G \end{aligned} \quad (18)$$

Si osserva che il momento trasferito dallo sciatore allo sci dipende dalla tensione  $T$ , dalle forze scambiate  $FS_x$  e  $FS_z$  e dalla posizione del baricentro e del punto di applicazione della fune. Nel caso di sciata senza traino ( $T(t) = 0$ ) il momento dipende solo dalla posizione del baricentro ed il contributo di  $FS_z$  è ben maggiore di quello di  $FS_x$  specie con angoli di inclinazione dello sci contenuti (vale l' equazione 19): di conseguenza il contributo  $FS_x$  si potrà anche trascurare.

$$TS_y(t) = FS_x(t)z_G - FS_z(t)x_G \quad (19)$$

Nel caso di sciata in condizioni stazionarie, ovvero con velocità del motoscafo costante e sciatore e tavola in posizione fissa ( $x(t) = V * t, z(t) = z, \theta(t) = \theta, AS_x(t) = 0, AS_z(t) = 0$ ), le tre equazioni di equilibrio per il solo sciatore con traino assumono la forma riportata in 20.

$$\begin{cases} M\sin(\theta)g - T(t)\cos(\theta) + FS_x(t) = 0 \\ M\cos(\theta)g + T(t)\sin(\theta) + FS_z(t) = 0 \\ (\sin(\theta)z_G - \cos(\theta)x_G)m_g - T(t)x_T\sin(\theta) - T(t)z_T\cos(\theta) + TS_y(t) = 0 \end{cases} \quad (20)$$

Nel caso invece di sciata senza traino, ma con accelerazione longitudinale ( $T(t) = 0, \theta(t) = \theta, z(t) = z$ ), le equazioni diventano invece quelle riportate in 21.

$$\begin{cases} M\sin(\theta)g + \left(\frac{d^2}{dt^2}x(t)\right)\cos(\theta)M + FS_x(t) = 0 \\ M\cos(\theta)g - \left(\frac{d^2}{dt^2}x(t)\right)\sin(\theta)M + FS_z(t) = 0 \\ (\sin(\theta)x_G + \cos(\theta)z_G)M \left(\frac{d^2}{dt^2}x(t)\right) \\ + (\sin(\theta)z_G - \cos(\theta)x_G)m_g = 0 \end{cases} \quad (21)$$

Come ultima semplificazione, a partire dalle equazioni 21, si consideri infine il caso in cui la posizione del baricentro dello sciatore sia al centro dello sci, cioè  $x_G = 0$ , esplicitando il momento e le due

componenti di forza esercitate dallo sciatore sullo sci. Le espressioni che si ottengono sono riportate in 22.

$$\begin{cases} FS_x(t) = -M\sin(\theta)g - \left(\frac{d^2}{dt^2}x(t)\right) \cos(\theta)M \\ FS_z(t) = \sin(\theta) \left(\frac{d^2}{dt^2}x(t)\right) M - M\cos(\theta)g \\ TS_y(t) = -\sin(\theta)z_G m_g - \cos(\theta)z_G M \left(\frac{d^2}{dt^2}x(t)\right) \end{cases} \quad (22)$$

### 2.3.3 Equazioni di moto dello sci e modello di stato

E' possibile a questo punto scrivere le equazioni di equilibrio di Newton alle traslazioni lungo gli assi  $x_b$  (equazione 23) e  $z_b$  (equazione 24) e quella di equilibrio alle rotazioni di Eulero attorno all'asse  $y_b$  (equazione 25) del sistema fisso per il solo sci. Esse servono a ricavare un modello di stato da utilizzare nel simulatore.

$$M_b \left(\frac{d^2}{dt^2}x(t)\right) - \cos(\theta(t))FS_x(t) + \sin(\theta(t))FS_z(t) - FW_x = 0 \quad (23)$$

$$-M_b \left(\frac{d^2}{dt^2}z(t)\right) + M_b * g - \sin(\theta(t))FS_x(t) - \cos(\theta(t))FS_z(t) - FW_z = 0 \quad (24)$$

$$\begin{aligned} -I_{y_b} M_b \left(\frac{d^2}{dt^2}\theta(t)\right) - \sin(\theta(t))l_{wc}FW_x(t) \\ + \cos(\theta(t))l_{wc}FW_z(t) - TS_y(t) = 0 \end{aligned} \quad (25)$$

Nel caso stazionario, cioè con  $x(t) = V * t$  (velocità costante della barca),  $z(t) = z$  e  $\theta(t) = \theta$ , le equazioni di equilibrio dello sci assumo la forma riportata in 26. In assenza della tensione della fune ( $T(t) = 0$ ), il cui contributo è contenuto nelle forze esercitate dallo sciatore, in direzione longitudinale non vi può essere equilibrio e ci sarà una decelerazione: le equazioni di equilibrio per lo sci saranno in questo caso quelle riportate in 27.

$$\begin{cases} -\cos(\theta)FS_x(t) + \sin(\theta)FS_z(t) - FW_x(t) = 0 \\ M_b * g - \sin(\theta)FS_x(t) - \cos(\theta)FS_z(t) - FW_z(t) = 0 \\ -\sin(\theta)l_{wc}FW_x(t) + \cos(\theta)l_{wc}FW_z(t) - TS_y(t) = 0 \end{cases} \quad (26)$$

$$\begin{cases} M_b * a_x - \cos(\theta)FS_x(t) + \sin(\theta)FS_z(t) - FW_x(x) = 0 \\ M_b * g - \sin(\theta)FS_x(t) - \cos(\theta)FS_z(t) - FW_z(t) = 0 \\ -\sin(\theta)l_{wc}FW_x(t) + \cos(\theta)l_{wc}FW_z(t) - TS_y(t) = 0 \end{cases} \quad (27)$$

Si considerano infine nuovamente le equazioni generali di equilibrio dello sci (23, 24 e 25) e da esse si ricavano le tre accelerazioni: quella longitudinale (lungo  $x_b$ ), quella trasversale (lungo  $z_b$ ) e quella angolare (attorno a  $y_b$ ). Si definisce come stato del sistema continuo il vettore di sei elementi riportato in 28. La dinamica del sistema sarà definita dalle equazioni riportate in 29 che stabiliscono come lo stato evolve nel tempo calcolandone la derivata a partire dalle forze esercitate dall'acqua ( $FW_x$  e  $FW_z$ ), dalla posizione del baricentro dell'area bagnata nel sistema mobile ( $l_{wc}$ ), dalle forze e dalla coppia impresse dallo sciatore ( $FS_x$ ,  $FS_z$  e  $TS_y$ ), dalla massa dello sci ( $M_b$ ), dal suo momento d'inerzia rispetto a  $y_b$  ( $IY_b$ ), dall'accelerazione di gravità ( $g$ ) e dalle prime tre componenti dello stato stesso ( $x(t)$ ,  $z(t)$  e  $\theta(t)$ ).

$$X(t) = \begin{pmatrix} x(t) \\ z(t) \\ \theta(t) \\ \dot{x}(t) \\ \dot{z}(t) \\ \dot{\theta}(t) \end{pmatrix} \quad (28)$$

$$\begin{cases} \dot{x}(t) = \frac{d}{dt}x(t) \\ \dot{z}(t) = \frac{d}{dt}z(t) \\ \dot{\theta}(t) = \frac{d}{dt}\theta(t) \\ \ddot{x}(t) = \frac{\cos(\theta(t))FS_x(t) - \sin(\theta(t))FS_z(t) + FW_x(t)}{M_b} \\ \ddot{z}(t) = -g + \frac{\sin(\theta(t))FS_x(t) + \cos(\theta(t))FS_z(t) + FW_z(t)}{M_b} \\ \ddot{\theta}(t) = \frac{\sin(\theta(t))l_{wc}FW_x(t) - \cos(\theta(t))l_{wc}FW_z(t) + TS_y(t)}{IY_b} \end{cases} \quad (29)$$

#### 2.3.4 Schema a blocchi del simulatore

Il modello di stato che si ottiene alla fine è un modello non lineare che potrà essere in futuro realizzato tramite un'opportuna S-Function in Matlab la quale realizza anche il calcolo delle forze esercitate dall'acqua sulla tavola. L'obiettivo è quello di realizzare uno schema a blocchi del simulatore come quello riportato in fig. 11. In un primo momento, si cercherà di simulare la sciata rettilinea senza traino essendo per ora sprovvisti di un sistema adeguato per simulare il bilancino e la corda. Come ingressi del modello (e quindi della S-Function) avremo le due componenti della forza esercitata dallo sciatore sullo

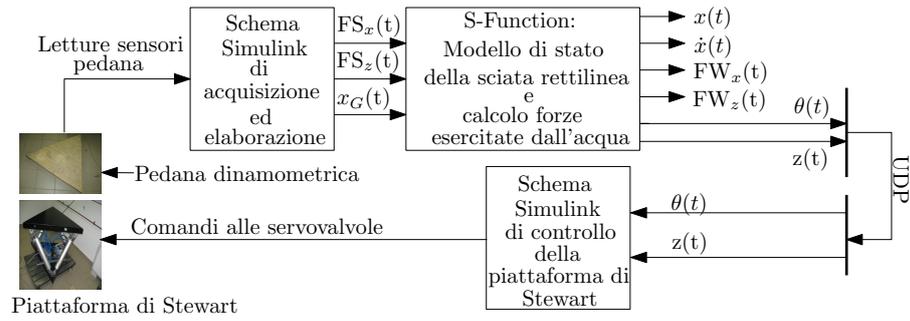


Figura 11: Schema a blocchi del simulatore.

sci  $FS_x$  ed  $FS_z$  (nel sistema mobile) e la posizione del suo baricentro  $x_G$  lungo l'asse  $x_s$  dello sci. Questi valori potranno essere ricavati tramite l'utilizzo di una pedana dinamometrica sulla quale sono posizionati dei sensori ottici di posizione che forniranno in uscita delle tensioni a partire dalle quali sarà possibile risalire ai tre ingressi della S-Function tramite uno schema Simulink di acquisizione ed elaborazione di tali tensioni. La funzione calcolerà poi al suo interno il momento esercitato dallo sciatore  $TS_y$  attorno all'asse  $y_s$  tramite l'equazione 19: in essa la coordinata  $z_G$  del baricentro lungo  $z_s$  non è nota, ma si potrà comunque trovare un sistema per stimarla, oppure trascurare il contributo di  $FS_x$  nell'equazione essendo comunque molto minore di quello di  $FS_z$ . Come uscite la S-Function renderà invece disponibili innanzitutto i valori delle componenti della forza esercitata dall'acqua sullo sci  $FW_x$  e  $FW_z$  (nel sistema fisso) calcolate a partire dall'inclinazione della tavola  $\theta(t)$ , dal suo affondamento  $z(t)$ , dalla sua velocità  $v(t)$  ( $v(t) = \dot{x}(t)$ ), dalle sue dimensioni  $H_{ws}$  e  $L_{ws}$  e dalla densità dell'acqua  $\rho$  come riportato nel paragrafo 2.2. Eventualmente da esse sarebbe poi possibile calcolare anche il momento esercitato dall'acqua  $T_W$  attorno a  $y_b$  come illustrato sempre nel paragrafo 2.2. All'inizio dovranno essere fissati chiaramente dei valori iniziali di inclinazione  $\theta_0$ , di affondamento  $z_0$  e di velocità  $V_0$ , che verranno invece poi aggiornati con l'aggiornamento dello stato del sistema che la funzione realizzerà poi come descritto in 29 avendo a disposizione, come visto, tutti i valori necessari. Essa renderà disponibili in uscita, in ogni istante, anche i valori dell'inclinazione della tavola  $\theta(t)$ , dell'affondamento  $z(t)$ , dell'avanzamento nella direzione di moto  $x(t)$  e della velocità nella stessa direzione  $v(t)$ . Tornando ad osservare lo schema di fig. 11, si vede come però gli ultimi due valori citati non verranno poi utilizzati in quanto lo spazio e i movimenti della piattaforma di Stewart utilizzata per simulare gli spostamenti della tavola sono limitati. Ci si accontenterà perciò di simulare la sciata rettilinea solamente in termini di variazione dell'affondamento e dell'inclinazione della tavola in tali condizioni a seconda di come lo sciatore varia la propria posizione sulla tavola a partire da una certa condizione iniziale. Per fare questo i valori di affondamento ed

inclinazione verranno trasmessi via UDP ad un secondo calcolatore (rispetto a quello che elaborava i dati raccolti dalla pedana dinamometrica) il quale, attraverso uno schema Simulink di controllo, riuscirà ad imprimere alla piattaforma di Stewart, al di sopra della quale verrà montata la pedana dinamometrica dove salirà lo sciatore, i movimenti desiderati tramite opportuni comandi alle sue servovalvole. Della pedana dinamometrica, della piattaforma di Stewart e di tutti gli altri elementi dell'architettura che andrà a realizzare lo schema a blocchi del simulatore riportato in fig. 11, oltre che degli schemi Simulink in esso presenti, si parlerà nel successivo capitolo 3; la S-Function è invece stata appena descritta. Allo stato attuale del progetto, però, sono state fatte solamente delle prove (illustrate sempre nel successivo capitolo 3) per testare l'architettura che è stata messa appunto per la realizzazione del simulatore nella quale però non è ancora stata inserita la S-Function, la cui realizzazione in Matlab è lasciata invece a chi volesse proseguire con la realizzazione di questo progetto.



## SETUP SPERIMENTALE

In questo capitolo verranno presentati i diversi componenti utilizzati al fine di realizzare il simulatore per lo sci nautico. Dopo una veloce presentazione dell'architettura generale dell'apparato di simulazione, ci si concentrerà sui due elementi fondamentali del simulatore: la pedana dinamometrica e la piattaforma di Stewart. Per quanto riguarda la pedana, se ne descriverà la struttura e verrà presentato lo schema Simulink di lettura dei sensori e come, all'interno di esso, si riesca a risalire alla posizione del baricentro dello sciatore e alle componenti di forza che egli esercita sulla pedana. Ci si soffermerà poi sulla caratterizzazione dei sensori e sul controllo della piattaforma di Stewart che sono state parte integrante del lavoro di tesi. Per concludere si valuteranno le prestazioni dell'architettura realizzata determinandone la risposta in frequenza in direzione verticale e eseguendo opportune prove sperimentali per verificarne il funzionamento.

### 3.1 ARCHITETTURA DEL SIMULATORE

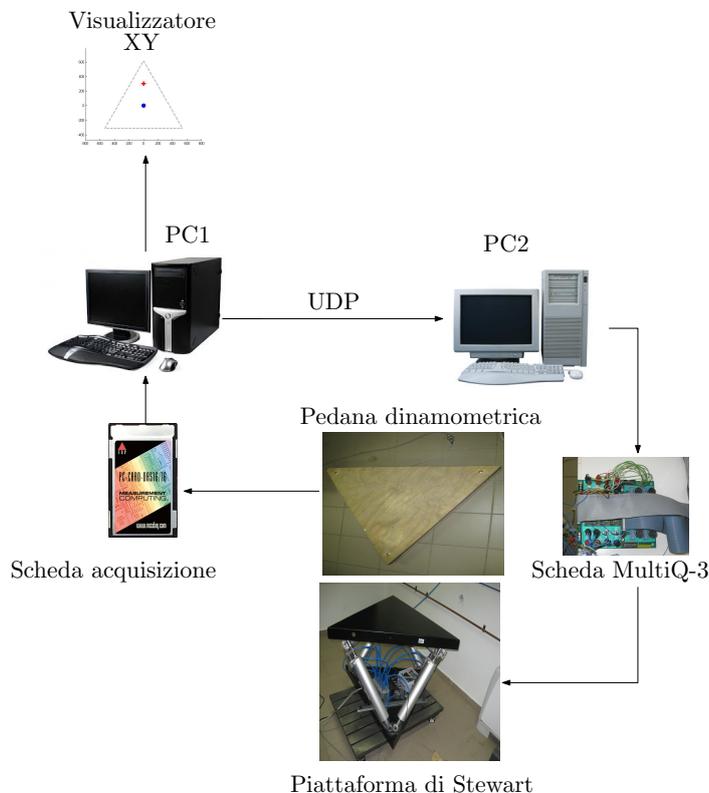


Figura 12: Architettura del simulatore.

I diversi componenti utilizzati per la realizzazione del simulatore sono rappresentati nello schema di fig. 12 che si andrà ora ad analizzare. Gli sci e la loro iterazione con l'acqua sono simulati attraverso l'utilizzo di una pedana dinamometrica in legno di forma triangolare montata sopra la piattaforma di Stewart che è un manipolatore di tipo parallelo. La pedana in legno ha lo scopo di rilevare le forze esercitate dallo sciatore su di essa attraverso l'utilizzo di sei sensori ottici di posizione, opportunamente posizionati: il dettaglio della struttura della pedana verrà presentato nel seguito del capitolo. I dati rilevati dai sensori vengono acquisiti, in tempo reale e sotto forma di tensioni analogiche, da un primo calcolatore, attraverso una scheda di acquisizione PC-CARD-DAS16/16A. L'acquisizione ed elaborazione in tempo reale è resa possibile dall'utilizzo di Matlab, Simulink e Real Time Windows Target ed in particolare dal blocchetto denominato "Analog Input". Mediante uno schema Simulink, il primo PC elabora le tensioni dei sensori per ottenere le forze esercitate dallo sciatore e la posizione del suo baricentro rispetto al centro della pedana. Tali dati sono poi trasmessi via UDP ad un secondo calcolatore il cui compito è quello di controllare il movimento della piattaforma di Stewart in base ai dati ricevuti allo scopo di simulare la sciata, sempre attraverso Simulink e Real Time Windows Target. Esso invia infine i segnali di comando alle servovalvole della piattaforma attraverso una scheda elettronica MULTIQ-3. Il primo calcolatore ha anche il compito di visualizzare in 2D la traiettoria del baricentro dello sciatore attraverso l'utilizzo di un Visualizzatore XY e di una opportuna S-Function.

### 3.2 LA PEDANA DINAMOMETRICA

Vista la struttura generale del simulatore, ci si concentrerà ora sui suoi due componenti fondamentali ovvero la pedana dinamometrica e la piattaforma di Stewart, partendo dalla prima con lo scopo di descriverne la struttura e spiegare come vengono misurate le forze di iterazione dello sciatore con la pedana e la posizione del baricentro. Infine ci si soffermerà sulla taratura dei sensori di misura delle forze "orizzontali".

#### 3.2.1 *Struttura della pedana*

La pedana dinamometrica, che si può osservare in fig. 13, è costituita da due piattaforme triangolari in legno collegate da elementi cedevoli di elevata rigidità in gomma, detti antivibranti. Essi sono posizionati negli angoli della struttura con l'asse passante lungo le bisettrici dei tre angoli interni del triangolo e, deformandosi lungo i tre assi cartesiani, permettono alla piattaforma superiore di muoversi rispetto a quella inferiore che rimane fissa. La pedana viene posizionata sopra la piattaforma di Stewart allo scopo di rilevare le forze impresse dallo

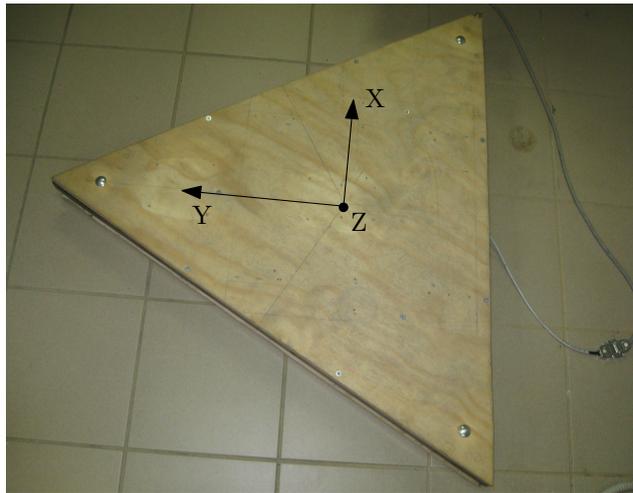


Figura 13: La pedana dinamometrica.

sciatore al simulatore ed individuare la posizione del suo baricentro. Tali dati, attraverso il modello di stato illustrato nel capitolo 2, verranno tradotti in opportuni movimenti da imporre alla piattaforma di Stewart al fine di simulare la sciata. Oltre ai tre elementi cedevoli, la pedana dinamometrica è dotata anche di sei sensori ottici di posizione: tre servono a rilevare le variazioni di distanza lungo l'asse verticale Z, mentre gli altri tre permettono di misurare i movimenti sul piano XY. E' inoltre presente un telaio in ferro saldato sulla piattaforma superiore su cui sono posizionati i sensori: la scelta dei materiali è dettata dalla necessità di avere una ridotta flessibilità e un peso contenuto. Il dettaglio della disposizione dei sensori è illustrato in fig. 14: i sensori verticali sono collocati in prossimità degli antivibranti, mentre quelli orizzontali lungo i lati del triangolo.

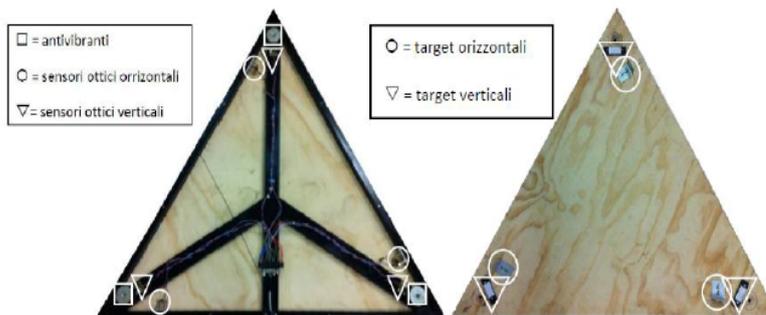


Figura 14: Posizione dei sensori.

La pedana è chiaramente equilatera e delle stesse dimensioni di ciascuna delle due basi della piattaforma di Stewart ed è dotata di un cavo multipolare che, attraverso la scheda PC-CARD-DAS16/16A, permette di inviare i dati rilevati dai sensori al computer che gli ela-

bora per calcolare gli spostamenti di riferimento da trasmettere al computer che controlla la piattaforma di Stewart. Tale cavo consente inoltre di alimentare i sensori stessi. Il sistema di riferimento della pedana coincide esattamente con quello mobile della piattaforma di Stewart che verrà illustrato in seguito e presenta l'origine nel baricentro geometrico del triangolo, l'asse Z verso l'alto, l'asse Y nella direzione dell'altezza del triangolo e rivolto verso il vertice e l'asse X disposto in modo da completare un sistema di riferimento destrorso (vedi fig. 13).

### 3.2.2 Schema Simulink di acquisizione ed elaborazione

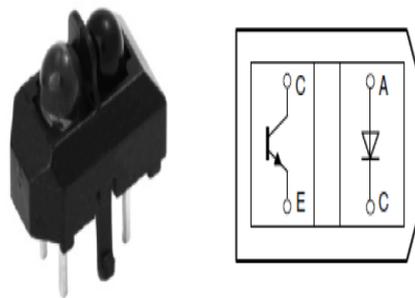


Figura 15: Dettaglio di un sensore ottico.

In fig. 15 è mostrato un dettaglio dei sensori ottici utilizzati dalla pedana. I sensori ottici sono dotati di un fotoemittitore ad infrarossi e non sono sensibili alle lunghezze d'onda del visibile grazie ad una schermatura al piombo. Il fascio di infrarossi proveniente dal fotoemittitore viene riflesso dall'oggetto di cui si vuole misurare la distanza e, a seconda di quest'ultima, varierà la quantità di energia ricevuta da un fototransistor, la quale si traduce a sua volta in una variazione della corrente di base che va a modulare la corrente di collettore. Quest'ultima va ad incidere sulla differenza di potenziale ai capi della resistenza fra emettitore e riferimento di terra: tale tensione aumenta al diminuire della distanza tra emettitore e oggetto ed è quella misurata dal sensore.

Si vuole ora analizzare lo schema di fig. 16 che mostra uno schema a blocchi equivalente allo schema Simulink che, a partire dalle sei tensioni misurate dai sensori, elabora i dati da trasmettere via UDP al computer che comanda la piattaforma di Stewart e visualizza la posizione del baricentro attraverso un'opportuna S-Function. Si descriverà ora sola la parte dello schema che calcola le diverse componenti di forza e la posizione del baricentro: del resto dello schema si parlerà nel paragrafo 3.5. Attraverso la scheda PC-CARD-DAS16/16A ed il blocco "Analog Input" di Real Time Windows Target è possibile leggere le tensioni prodotte dai sei sensori. Le sei tensioni acquisite ven-

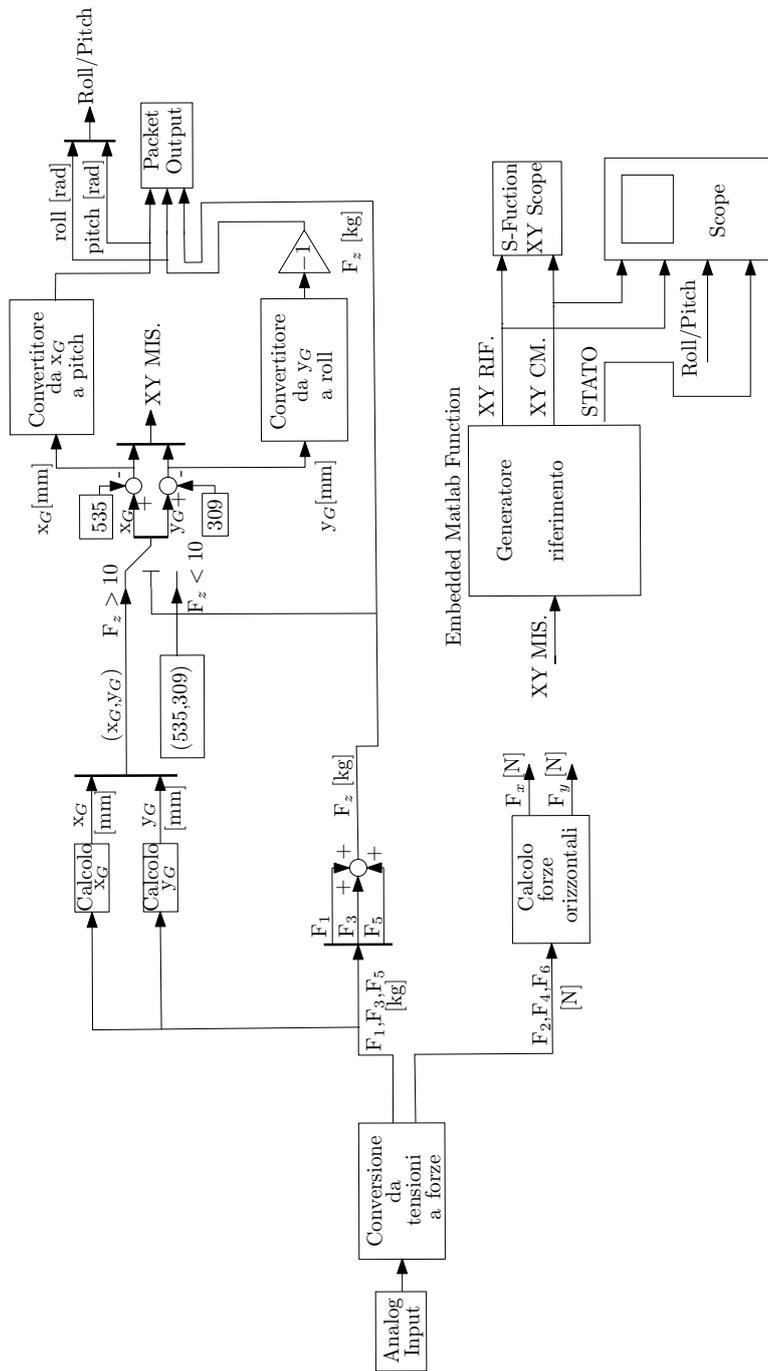


Figura 16: Schema di acquisizione/elaborazione.

gono mandate in ingresso al sottosistema denominato "Conversione da tensioni a forze" il cui contenuto è schematizzato in fig. 17.

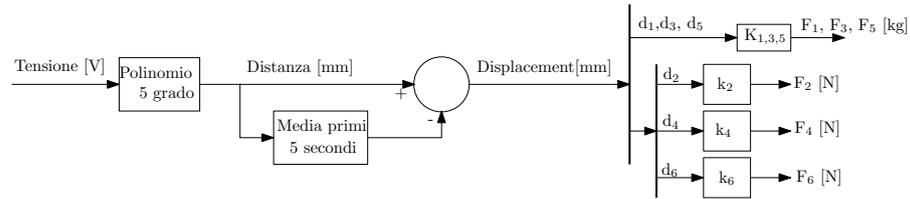


Figura 17: Conversione da tensioni a forze: schema Simulink.

Si risale innanzitutto dalle tensioni alle corrispondenti misure di distanza attraverso dei polinomi di quinto grado che altro non sono che le funzioni inverse delle caratteristiche da spostamento a tensione dei sensori ottici. All'accensione del simulatore si deve eseguire una "tara" in modo tale che la distanza fornita dai sensori sia sempre nulla a pedana scarica e ogni spostamento successivo sia rilevato rispetto a questa situazione di partenza. Questo viene fatto affinché non si vadano a misurare le deformazioni dovute alla forza di gravità che si tradurrebbero poi in delle forze non nulle anche nel momento in cui non vi è alcun peso sopra la pedana. Senza tale accorgimento infatti la componente in Z risulterebbe pari alla forza di gravità a pedana scarica, mentre le forze orizzontali potrebbero comunque non risultare perfettamente nulle. La "tara" viene eseguita facendo la media delle misure di distanza dei sensori per cinque secondi, e sottraendo poi tale media alle misure successive. Si ottengono quindi dei "displacement" dei sensori rispetto alla condizione in cui la pedana è scarica. A questo punto, volendo eseguire un calcolo preciso, a partire dalle coordinate dei sensori e dai "displacement" ottenuti, si dovrebbe risalire, attraverso opportune equazioni trigonometriche linearizzate, alla posizione ( $x$ ,  $y$ ,  $z$ ) e all'orientazione (roll, pitch e yaw) della piattaforma superiore rispetto a quella inferiore della pedana. Da tali informazioni e dalle coordinate degli antivibranti, con altre equazioni analoghe alle precedenti, si dovrebbe poi ricavare la deformazione dei tre antivibranti lungo i tre assi cartesiani. Gli antivibranti a questo punto vengono assimilati a delle molle con valori di rigidità noti lungo le tre direzioni cartesiane: moltiplicando la deformazione degli antivibranti per la rigidità in una certa direzione si ottiene dunque la forza esercitata dallo sciatore sull'antivibrante in quella direzione. L'uso del condizionale è tuttavia motivato dal fatto che tali calcoli risultavano complicati da realizzare con l'utilizzo di uno schema Simulink e si è quindi preferita una strada più semplice che consiste semplicemente nel moltiplicare i "displacement" dei sensori (in mm) per le rigidità degli antivibranti basandosi sul fatto che, essendo le coordinate dei sensori molto simili a quelle degli elementi in gomma, tali "displacement" dovranno essere molto simili alle deformazioni degli antivibranti nelle medesima direzione.

Per quanto riguarda i sensori che misurano gli spostamenti verticali (individuati dai numeri dispari 1, 3, 5) il valore di rigidità (indicato con  $K_{1,3,5}$  in fig. 17) è lo stesso: la moltiplicazione dei "displacement" per tale valore permette di ottenere le tre forze verticali (indicate con  $F_1, F_3, F_5$ ) che lo sciatore esercita sui tre antivibranti in kg. Esse risulteranno sempre negative essendo ovviamente rivolte verso il basso, cioè contro la direzione dell'asse Z. Allo stesso modo i tre "displacement" relativi ai sensori orizzontali (individuati dai numeri pari 2, 4, 6) vengono moltiplicati per un valore di rigidità equivalente (che in questo caso risulta diverso per i tre antivibranti) allo scopo di ottenere le tre forze orizzontali (indicate con  $F_2, F_4, F_6$ ) trasmesse dallo sciatore ai tre antivibranti espresse però in N: i tre valori di rigidità equivalente sono indicati con  $k_2, k_4$  e  $k_6$  in fig. 17.

Sommando semplicemente le tre forze verticali è possibile calcolare la forza peso complessiva esercitata dallo sciatore in direzione verticale e individuare la posizione del suo baricentro, come verrà illustrato nel prossimo paragrafo; Le tre forze orizzontali consentono invece di calcolare le forze  $F_x$  e  $F_y$  che lo sciatore esercita complessivamente sulla pedana in direzione X e Y rispettivamente: anche questo calcolo verrà esposto in seguito.

Per maggiori dettagli sulla caratterizzazione dei sensori ottici, sul calcolo teorico delle deformazioni degli antivibranti e sulla determinazione della loro rigidità verticale si può consultare la prima tesi svolta sul progetto del simulatore di sci nautico, di cui questo scritto è la diretta continuazione, ovvero [1]. I valori di rigidità equivalente orizzontale sono invece stati oggetto di un'opportuna caratterizzazione che verrà descritta in seguito. I sei valori di rigidità equivalente sono riportati nella tab. 1

Sensori verticali	
$k_1$ [kg/mm]	51.1
$k_3$ [kg/mm]	51.1
$k_5$ [kg/mm]	51.1
Sensori orizzontali	
$k_2$ [N/mm]	222.8827
$k_4$ [N/mm]	207.8633
$k_6$ [N/mm]	277.9213

Tabella 1: Valori delle rigidità equivalenti.

In realtà gli assi a cui si fa riferimento sono quelli di un sistema mobile e perciò la forza misurata non sarà l'intera forza peso (orientata con direzione uguale e verso opposto all'asse Z del riferimento fisso della piattaforma di Stewart) bensì la sua componente lungo

la direzione Z del sistema mobile della piattaforma di Stewart che coincide con quello della pedana. Essendo però le inclinazioni in termini di pitch e roll contenute essa non sarà molto differente dalla forza complessiva. Nel seguito, quando si parlerà di sensori "verticali" si intenderà sempre che misurano forze (o meglio spostamenti poi convertiti in forze) lungo l'asse Z della pedana dinamometrica (cioè quello mobile della piattaforma) dette a loro volta "verticali", mentre quando si parlerà di sensori "orizzontali" si intenderanno quelli che misurano forze (o meglio spostamenti poi convertiti in forze) lungo il piano XY sempre della pedana, dette a loro volta "orizzontali".

### 3.2.2.1 Determinazione del baricentro

La posizione del baricentro viene determinata attraverso una media ponderata delle tre forze verticali misurate sui tre antivibranti. Le coordinate in direzione X e Y sono infatti ottenute come media di quelle corrispondenti dei tre antivibranti pesate attraverso le forze verticali che ogni antivibrante misura: verosimilmente infatti più il baricentro dello sciatore è spostato verso un antivibrante e più quello misurerà una componente di forza elevata. Le coordinate del baricentro ( $x_G$ ,  $y_G$ ) si ottengono quindi attraverso le formule 30 e 31. Si osservi il disegno di fig. 18 che contiene i valori delle coordinate degli antivibranti.

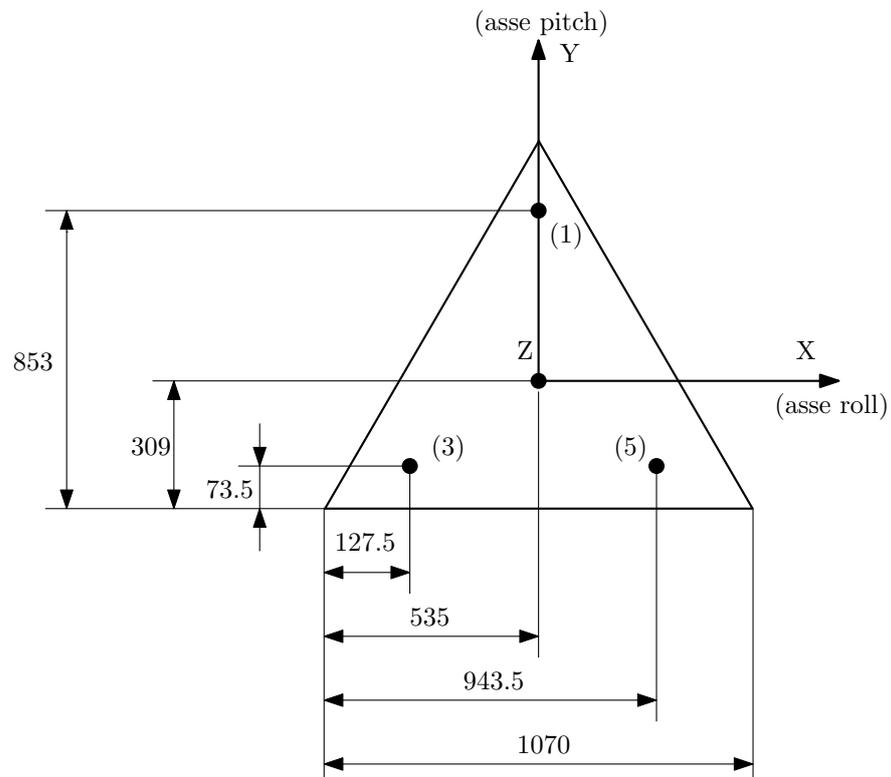


Figura 18: Coordinate degli antivibranti.



rigidezze equivalenti ( $k_2, k_4, k_6$ ), si ottengono le tre forze indicate in fig. 19 (indicate con  $F_2, F_4, F_6$  anche negli schemi di fig. 16 e 17). Esse, al contrario di quelle verticali (che erano in kg in modo da avere una verifica immediata rispetto al peso dello sciatore), sono espresse in N. La caratterizzazione dei valori di rigidezza equivalente in direzione orizzontale è stata parte integrante del lavoro di tesi. Verranno ora esposte le considerazioni teoriche su cui si basa la caratterizzazione.

### 3.2.3.1 Considerazioni teoriche per la determinazione delle rigidezze equivalenti orizzontali

Si consideri la disposizione di forze orizzontali riportata in fig. 19: le forze sui tre antivibranti sono quelle misurate nel momento in cui viene applicata una forza  $F$  sul lato inferiore del triangolo ad una certa distanza  $b$  dall'asse  $Y$ , nella sua stessa direzione, ma con verso opposto. Le forze misurate si considerano orientate quindi secondo quello che è assunto come il loro verso positivo. E' possibile scrivere tre equazioni di equilibrio: una riguardante le componenti di forza lungo l'asse  $X$ , una quelle lungo l'asse  $Y$  e una riguardante l'equilibrio dei momenti attorno all'asse  $Z$ . Esse danno origine al sistema 32 che è un sistema di tre equazioni in tre incognite, ovvero le tre forze orizzontali sui tre antivibranti; è nota invece la forza applicata  $F$ .

$$\begin{cases} +\frac{1}{2}F_2 & +\frac{1}{2}F_4 & -F_6 & -F & = & 0 \\ +\frac{\sqrt{3}}{2}F_2 & -\frac{\sqrt{3}}{2}F_4 & & & = & 0 \\ -\frac{\sqrt{3}}{2}l*F_2 & -\frac{\sqrt{3}}{2}l*F_4 & -\frac{\sqrt{3}}{2}l*F_6 & -b*F & = & 0 \end{cases} \quad (32)$$

Si ipotizza, a questo punto, che la forza  $F$  sia applicata al centro del lato del triangolo, ovvero che  $b$  sia nullo, e, risolvendo il sistema, si ottengono i valori delle tre forze sugli antivibranti (in funzione di  $F$ ) che sono riportati nel sistema 33.

$$\begin{cases} F_2 = F_4 = \frac{F}{3} \\ F_6 = -\frac{2}{3}F \end{cases} \quad (33)$$

A questo punto però si deve tener conto che ognuna delle tre forze calcolate è la forza con cui la molla si oppone alla deformazione provocata dalla forza applicata ed è perciò pari ad un valore di rigidezza equivalente moltiplicato per il "displacement" misurato dal sensore. Il segno meno sulla  $F_6$  sta ad indicare che la forza di reazione su tale antivibrante sarà in realtà diretta con verso opposto rispetto a quello con cui è stata disegnata inizialmente e sono state impostate le equazioni, mentre le altre due forze sono state disegnate correttamente. In effetti, se si applica una forza verso il basso, si avrà una

deformazione verso il basso dell'antivibrante corrispondente al sensore 6 e quindi una forza che si oppone a tale deformazione che ha verso opposto, cioè verso l'alto nel disegno. Le altre due forze invece sono dirette verso l'esterno perché la deformazione della molla sarà verso l'interno del triangolo. Tuttavia non si è interessati alle forze di reazione delle molle bensì alle forze che, applicando la forza  $F$  alla pedana, sono trasmesse a ciascuna molla. Esse sono quelle a cui ogni molla si oppone con le forze di reazione e, per il principio di azione e reazione, hanno lo stesso modulo ma verso opposto ad esse. Si ottengono dunque le tre equazioni del sistema 34 in cui i segni meno all'inizio di ciascuna equazione sono proprio dovuti al fatto che ora si considerano le forze trasmesse e non più quelle di reazione. Da tali equazioni si possono ricavare i valori delle rigidezze equivalenti di ciascun antivibrante riportati in 35.

$$\begin{cases} -k_2 d_2 = \frac{F}{3} \\ -k_4 d_4 = \frac{F}{3} \\ -k_6 d_6 = -\frac{2}{3}F \end{cases} \quad (34)$$

$$\begin{cases} k_2 = -\frac{1}{3} \frac{F}{d_2} = -\frac{1}{3} k'_2 \\ k_4 = -\frac{1}{3} \frac{F}{d_4} = -\frac{1}{3} k'_4 \\ k_6 = \frac{2}{3} \frac{F}{d_6} = \frac{2}{3} k'_6 \end{cases} \quad (35)$$

I ragionamenti precedentemente esposti suggeriscono un metodo per determinare la rigidezza equivalente di ciascun antivibrante in direzione orizzontale. Se si applica una forza di spinta  $F$  al centro del lato inferiore della pedana dinamometrica (quello dove si trova il sensore 6), la rigidezza dell'antivibrante in basso a destra (e cioè quello in corrispondenza del sensore 6) sarà pari ai due terzi del rapporto tra la forza applicata ed il "displacement" misurato dal sensore 6. Le altre due forze saranno invece pari  $-1/3$  del rapporto tra la forza applicata e il "displacement" misurato dal sensore corrispondente. Dal momento che si considerano "displacement" positivi verso l'esterno del triangolo,  $d_6$  risulterà positivo, mentre  $d_2$  e  $d_4$  saranno negativi e quindi tutte le rigidezze equivalenti avranno un valore positivo come è giusto che sia. Per simmetria della struttura, se invece di applicare la forza al lato corrispondente al sensore 6, la si applica ad uno degli altri lati, si avrà comunque che la rigidezza equivalente sul lato di applicazione della forza sarà pari ai due terzi del rapporto tra forza applicata e "displacement" misurato dal sensore su quel lato, mentre le altre due rigidezze saranno pari a  $-1/3$  del rapporto tra forza applicata e "displacement" corrispondente il quale risulterà negativo.

### 3.2.3.2 Procedura pratica per la determinazione delle rigidzze equivalenti orizzontali

Per mettere in pratica la procedura esposta alla fine del paragrafo precedente si è posizionata la pedana dinamometrica sopra alla piattaforma di Stewart (mantenuta con tutti i cilindri al minimo della lunghezza in modo che la piattaforma non si sollevasse) e si è aggranciando un lato alla volta al muro nel suo punto intermedio con dei moschettoni. Fra il lato della pedana ed il muro, ed in direzione ortogonale ad essi, si è posto un tensionatore, che permetteva di imprimere valori di forza crescente semplicemente avvitando sempre più, ed una cella di carico con sensibilità di 10 N per misurare il valore della forza di volta in volta impressa. Per ciascun lato della pedana sono state imposte forze di valore crescente da 0 a circa 900 N con intervallo di circa 100 N, rilevando il "displacement" corrispondente al lato su cui veniva impressa la forza attraverso lo schema di fig. 17 a cui era stato aggiunto un opportuno visualizzatore. Per ciascun lato sono state fatte almeno quattro prove di questo tipo. A questo punto i dati raccolti sono stati elaborati utilizzando il codice Matlab riportato nell'appendice A.6. Inserendo il numero del sensore ed il numero di prove fatte, esso concatena tutti i valori di forza impressi sul lato corrispondente a quel sensore in un unico vettore  $F$  e tutti i "displacement" misurati dal sensore in un altro chiamato  $x$ . La cella di carico assume per convenzione come forze positive le forze di compressione: essa quindi avrà rilevato valori di forza negativi essendo stata applicata una forza di trazione. Questo motiva il fatto che i valori di forza vengano cambiati di segno visto che si era assunto che la forza di trazione applicata ad un lato fosse considerata positiva. I due vettori individuano una serie di punti su un grafico avente come ascissa il "displacement" in mm e come ordinata la forza in N. Tali punti individuano un andamento che può essere approssimato con una retta ai minimi quadrati. In particolare la soluzione ai minimi quadrati dell'equazione 36 è uno scalare  $K$  pari alla pendenza della retta il cui valore è dato dall'equazione 37 nella quale  $x^{-1}$  è la pseudoinversa di  $x$ .

$$F = K * x \quad (36)$$

$$K = (x^T x)^{-1} x^T F = x^{-1} F \quad (37)$$

In fig. 20 si può osservare come i vari dati raccolti per ciascuno dei tre sensori sia approssimato tramite la retta ai minimi quadrati. In base ai ragionamenti riportati nel paragrafo precedente, il valore di rigidzza equivalente si determina moltiplicando la pendenza  $K$  per  $2/3$  come indica la terza equazione del sistema 35 nella quale  $K$  corrisponde a  $k'_6$ . In fig. 21 si possono osservare i tre andamenti lineari

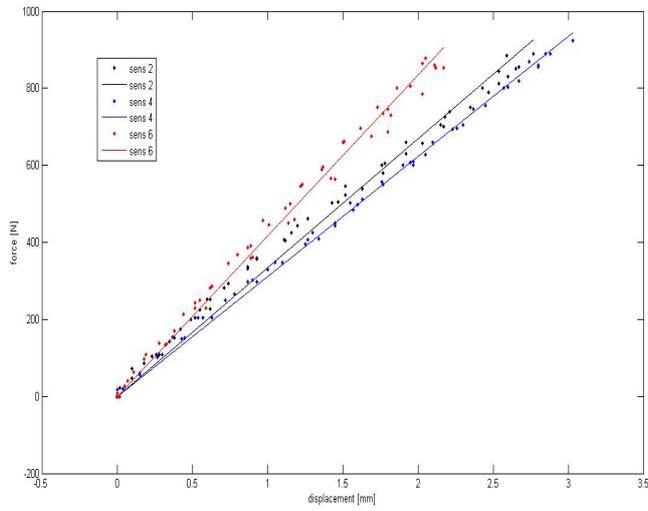


Figura 20: Approssimazione ai minimi quadrati.

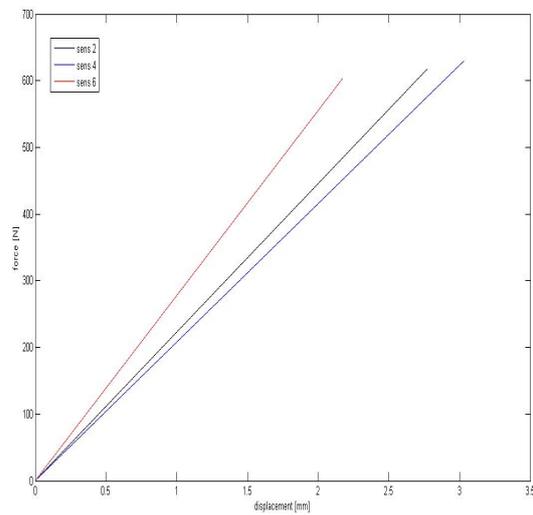


Figura 21: Curve caratteristiche forza vs displacement dei sensori orizzontali.

che approssimano il comportamento delle molle equivalenti orizzontali, rappresentando cioè il legame fra "displacement" misurato dal sensore e forza trasmessa all' antivibrante.

### 3.2.4 Calcolo delle forze orizzontali

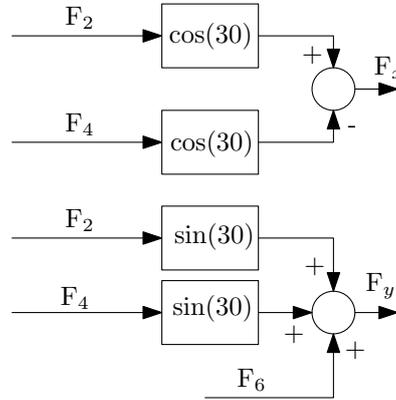


Figura 22: Determinazione forze orizzontali.

Come detto in precedenza, moltiplicando i tre "displacement" per le rigidità orizzontali equivalenti, si ottengono le forze orizzontali trasmesse ai tre antivibranti nel momento in cui la pedana dinamometrica è caricata con un certo peso. La loro direzione è quella in cui viene misurato il "displacement" corrispondente ed il verso positivo è assunto verso l' esterno della piattaforma. Esse sono le  $F_2$ ,  $F_4$ ,  $F_6$  nelle fig. 19 e 16 e sono espresse in N. All' interno del blocco "Calcolo forze orizzontali" di tale schema (il cui contenuto è rappresentato in fig. 22), a partire da tali forze, vengono calcolate le forze complessive esercitate nelle direzioni X e Y, indicate con  $F_x$  ed  $F_y$ . Esso compone semplicemente le tre forze come indicato dalle equazioni 38 e 39.

$$F_x = F_2 * \cos(30) - F_4 * \cos(30) \quad (38)$$

$$F_y = F_2 * \sin(30) + F_4 * \sin(30) - F_6 \quad (39)$$

### 3.2.5 Verifica caratterizzazione con piastre di cemento

Al fine di verificare la bontà delle caratterizzazioni effettuate, la pedana dinamometrica è stata posizionata sopra la piattaforma di Stewart e caricata con tre piastre di cemento di forma quadrata per un peso complessivo di circa 69 kg. Chiaramente le piastre sono state caricate dopo l'esecuzione della tara in modo che si potessero misurare le forze esercitate da esse sulla pedana. La piattaforma di Stewart è stata

poi orientata con diverse inclinazioni in termini di roll ( $\rho$ ) e pitch ( $\theta$ ), quindi rispetto agli assi X e Y del sistema fisso della piattaforma. In base alla convenzione adottata, quindi, l'orientazione finale si ottiene sempre ruotando prima attorno ad X e poi attorno a Y. Per ogni orientazione sono state rilevate le componenti di forza  $F_x$ ,  $F_y$  ed è stato verificato che i loro valori fossero sufficientemente vicini a quelli attesi per ciascuna orientazione. Tali valori attesi vengono calcolati a partire dalla matrice di rotazione che consente di passare dal sistema fisso al sistema mobile che è il prodotto della matrice dei coseni direttori di una rotazione attorno ad X e di una rotazione attorno a Y: essa è riportata in 40. La matrice dei coseni direttori è quella che ha sulle colonne le coordinate dei versori del nuovo sistema di riferimento rispetto a quello di partenza.

$$R_p^b = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\rho & -\sin\rho \\ 0 & \sin\rho & \cos\rho \end{bmatrix} \quad (40)$$

Nel sistema fisso (pedice b, cioè base) della base della piattaforma di Stewart le tre piastre eserciteranno solo una forza  $F$  in direzione Z (cioè la direzione verticale) pari al loro peso complessivo. Nel sistema mobile (pedice p, cioè pedana), invece, la forza che esercitano si può scomporre nelle tre componenti in direzione X, Y e Z di tale sistema di riferimento. Il valore atteso delle tre componenti nel sistema mobile, che sono poi quelle che possono essere misurate con lo schema di fig. 16, viene ottenuto con l'equazione 41.

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}_p = [R_p^b]^T * \begin{Bmatrix} 0 \\ 0 \\ F \end{Bmatrix}_b = \begin{Bmatrix} -\sin(\theta) \\ \sin(\rho)\cos(\theta) \\ \cos(\rho)\cos(\theta) \end{Bmatrix} F \quad (41)$$

La forza complessiva  $F$  è pari a  $-68,97$  kg ed è assunta negativa perché rivolta verso il basso mentre l'asse Z del sistema fisso è rivolto verso l'alto.

Confrontando le tab. 2 e 4 rispettivamente con le tab. 3 e 5 si può notare una corrispondenza abbastanza buona fra i valori misurati ed attesi, ovviamente accettando una tolleranza di un paio di kg e qualche errore di misura.

Bisogna inoltre tenere presente che, se si volesse una maggiore precisione, sarebbe necessario considerare pure la presenza di isteresi nella caratterizzazione degli antivibranti che invece nel presente lavoro è stata trascurata per semplicità.

### 3.2.6 Verifica caratterizzazione con persona sulla pedana

La verifica con le piastre consente di provare rapidamente diverse orientazioni della piattaforma e fornisce una valutazione in linea di

		Roll [°]				
		-10	-5	0	5	10
Pitch [°]	-10	-1.557	-1.505	-1.384	-0.692	-1.236
	-5	-1.004	-0.677	-0.737	-0.488	-0.464
	0	-0.326	0.068	0.002	0.066	0.31
	5	0.458	0.744	0.71	0.713	1.026
	10	1.163	1.344	1.403	1.505	1.655

Tabella 2: Verifica con piastre: valori della  $F_x$  misurati in N.

		Roll [°]				
		-10	-5	0	5	10
Pitch [°]	-10	-1.22	-1.22	-1.22	-1.22	-1.22
	-5	-0.612	-0.612	-0.612	-0.612	-0.612
	0	0	0	0	0	0
	5	0.612	0.612	0.612	0.612	0.612
	10	1.22	1.22	1.22	1.22	1.22

Tabella 3: Verifica con piastre: valori della  $F_x$  teorici in N.

		Roll [°]				
		-10	-5	0	5	10
Pitch [°]	-10	1.287	0.609	-0.004	-1.119	-1.334
	-5	1.383	0.821	-0.137	-0.897	-1.371
	0	1.406	0.818	-0.057	-0.582	-1.394
	5	1.389	0.79	-0.074	-0.671	-1.392
	10	1.096	0.761	-0.093	-0.675	-1.361

Tabella 4: Verifica con piastre: valori della  $F_y$  misurati in N.

		Roll [°]				
		-10	-5	0	5	10
Pitch [°]	-10	1.202	0.603	0	-0.603	-1.202
	-5	1.215	0.611	0	-0.611	-1.215
	0	1.22	0.613	0	-0.613	-1.22
	5	1.215	0.611	0	-0.611	-1.215
	10	1.202	0.603	0	-0.603	-1.202

Tabella 5: Verifica con piastre: valori della  $F_y$  teorici in N.

massima di una buona caratterizzazione dei sensori. Si vuole ora verificare che i valori delle forze misurate siano compatibili con quelle attese nel momento in cui sopra la pedana vi è una persona in posizione di sciata la cui distribuzione di peso è ben diversa da quella delle piastre. Ad esempio il baricentro della persona è molto più alto e la sua proiezione sulla pedana si sposta molto di più nel momento in cui si inclina la piattaforma rispetto a quello delle piastre. Si ricorda inoltre che, durante la sciata rettilinea che ci si è posti l'obiettivo di simulare, si passerà per diverse inclinazioni dell'angolo di roll, mentre non vi sarà alcuna inclinazione in termini di pitch: di conseguenza, per quanto riguarda le prove con persona sulla pedana, si valuteranno solamente diverse inclinazioni in termini di roll, lasciando sempre un angolo di pitch nullo. Si è quindi scelto di verificare i valori delle tre componenti di forza  $F_x$ ,  $F_y$ ,  $F_z$  misurate (tab. 6) con quelli attesi (tab. 7) per diversi valori dell'angolo di roll, andando a rilevare pure i valori della posizione del baricentro (tab. 8). Si osservi come, essendo le inclinazioni di piccola entità, le componenti di forza in Z siano praticamente pari all'intera forza o comunque di poco inferiori. La corrispondenza fra valori teorici e misurati delle componenti della forza in X e Y risulta buona considerando anche il fatto che la persona non riuscirà mai a rimanere perfettamente eretta in posizione verticale senza sbilanciarsi leggermente a destra o a sinistra.

		Componenti forza misurate		
		$F_x$ [N]	$F_y$ [N]	$F_z$ [kgf]
Roll [°]	-10	0	1.24	-69.94
	-5	-0.126	0.818	-71.92
	0	-0.032	-0.02	-68.97
	5	0.272	-0.768	-70.53
	10	0.208	-1.317	-69.54

Tabella 6: Verifica con persona: valori delle componenti di forza misurate.

		Componenti forza attese		
		$F_x$ [N]	$F_y$ [N]	$F_z$ [kgf]
Roll [°]	-10	0	1.22	-67.922
	-5	0	0.613	-68.708
	0	0	0	-68.97
	5	0	-0.613	-68.708
	10	0	-1.22	-67.922

Tabella 7: Verifica con persona: valori delle componenti di forza attesi.

		Coordinate baricentro	
		$x_G$ [mm]	$y_G$ [mm]
Roll [°]	-10	3.82	150.67
	-5	-18.62	89.2
	0	-26.54	-5.5
	5	-18.72	-95.61
	10	-24.87	-142.66

Tabella 8: Verifica con persona: valori coordinate del baricentro.

Roll [°]	F[kgf]
-10	70.99
-5	72.38
0	68.97
5	70.98
10	70.76

Tabella 9: Verifica con persona: valori del modulo della forza.

Roll [°]	$M_x$ [kgf*mm]
-10	-10538.81
-5	-6414.3
0	-376.51
5	6747.70
10	9922.41

Tabella 10: Verifica con persona: valori del momento attorno ad X.

Il baricentro si sposta all'indietro con inclinazioni positive e in avanti con inclinazioni negative come era lecito attendersi. E' possibile infine ricavarsi i valori del modulo della forza  $F$  e del momento attorno all'asse  $X M_x$  utilizzando le equazioni 42 e 43.

$$F = \sqrt{F_x^2 + F_y^2 + F_z^2} \quad (42)$$

$$M_x = F_1 * |y_1| - F_3 * |y_3| - F_5 * |y_5| \quad (43)$$

In quest'ultima  $y_1, y_3, y_5$  sono le coordinate degli antivibranti in  $Y$  rispetto al baricentro geometrico della pedana. Nella prima tutte le forze dovranno ovviamente avere la stessa unità di misura. I valori ottenuti sono riportati nelle tab. 9 e 10. Il fatto che i valori del modulo siano tutti simili e vicini al reale peso della persona è un'ulteriore conferma che non si stanno commettendo errori così come l'andamento dei valori del momento.

### 3.3 LA PIATTAFORMA DI STEWART

Si passerà ora ad analizzare l'altro componente fondamentale del simulatore ovvero la piattaforma di Stewart con lo scopo di descriverne la struttura e spiegare come ne è stato realizzato il controllo.

#### 3.3.1 *Struttura della piattaforma*

La piattaforma di Stewart è un manipolatore di tipo parallelo a sei gradi di libertà in cui la base fissa (ancorata al pavimento con un opportuno supporto) e l'end-effector, nel nostro caso entrambi di forma triangolare, sono collegati da sei pistoni pneumatici attuati attraverso delle servovalvole. Un manipolatore parallelo presenta infatti più catene cinematiche in parallelo fra la base fissa e l'organo terminale, al contrario di uno seriale dove la catena cinematica è unica. I punti di contatto tra cilindri e basi triangolari superiore e inferiore sono in corrispondenza dei vertici di quest'ultime. Inoltre i cilindri sono disposti in modo tale da avere il punto di contatto con la pedana superiore in comune con uno dei due cilindri adiacenti e il punto di contatto con la pedana inferiore con l'altro. I cilindri sono dei giunti di tipo prismatico cioè restringono il moto ad un'unica traslazione lungo il loro asse, mentre nei punti di contatto con le pedana sono presenti un totale di dodici giunti sferici per impedire ai cilindri di ruotare su se stessi. L'allungamento e accorciamento dei cilindri permette alla piattaforma mobile di traslare e ruotare attorno ai tre assi cartesiani per un totale, come si diceva, di sei gradi di libertà. In fig. 23 si può osservare la piattaforma di Stewart utilizzata per il simulatore.



Figura 23: La piattaforma di Stewart.

I vantaggi di utilizzare un manipolatore parallelo rispetto ad uno seriale stanno nella maggiore capacità di carico (visto che viene ripartito fra i sei attuatori), nell'accuratezza del posizionamento, nelle elevate accelerazioni (per le basse masse in gioco) e nella costruzione semplice. La cinematica diretta di posizione si può però risolvere solo in casi particolari e con un elevato sforzo computazionale; tuttavia, la cinematica inversa, che è quella necessaria per il controllo, risulta molto semplificata e risolvibile per via analitica.

### 3.3.2 *Cinematica inversa di posizione*

Per cinematica inversa di posizione di un manipolatore si intende la configurazione che deve assumere ciascun giunto a partire dalla posizione finale desiderata dell'end-effector rispetto ad un sistema di riferimento fisso, passando cioè, come si dice, dalle coordinate nello spazio cartesiano a quelle nello spazio dei giunti. Nel caso della piattaforma di Stewart le coordinate dei giunti sono in sostanza le lunghezze dei sei cilindri; le coordinate cartesiane sono date invece dalla posizione dell'origine del sistema di riferimento solidale con la piattaforma superiore rispetto a quella dell'origine del sistema di riferimento fisso della piattaforma inferiore e dagli angoli di rotazione del sistema mobile rispetto ai tre assi di quello fisso. Come si diceva, il calcolo della cinematica inversa di posizione risulta essere piuttosto semplice per la piattaforma di Stewart ed è inoltre il primo passo fondamentale per il controllo che è stato realizzato.

Nello schema di fig. 24 si evidenziano due sistemi di riferimento: uno fisso e solidale con la base della piattaforma ed uno mobile in modo solidale con la sua parte superiore. Entrambi hanno l'origine coincidente con il baricentro geometrico del triangolo equilatero su cui sono posizionati. Il sistema mobile, indicato con il pedice  $p$ , è posizionato in modo tale che l'asse  $y_p$  sia lungo l'altezza del triangolo,

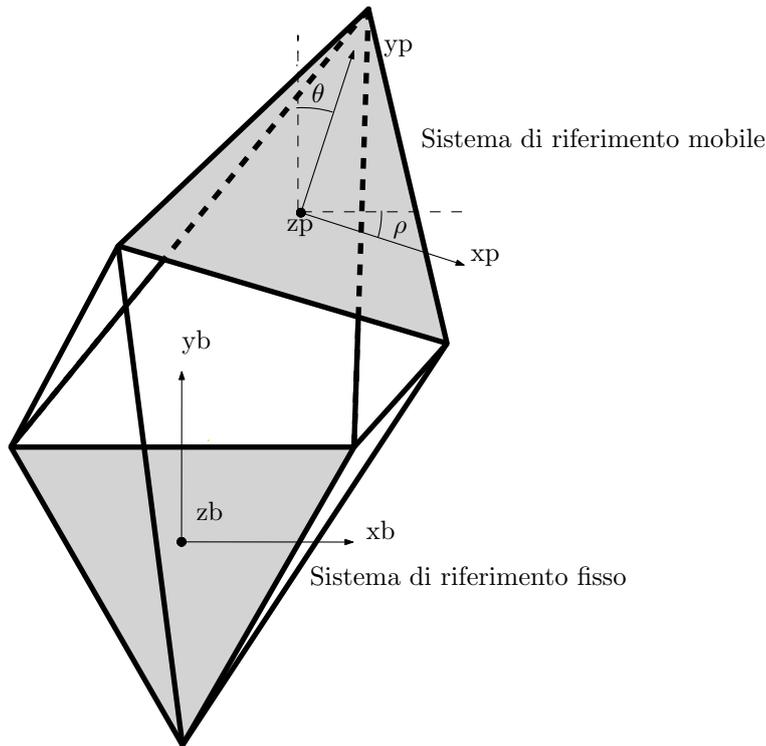


Figura 24: Schema dei sistemi riferimento della piattaforma.

ovvero lungo "l'asse dello sci", e abbia verso positivo in direzione del vertice; l'asse  $x_p$  è orientato verso la destra dell'origine e l'asse  $z_p$  verso l'alto. A "riposo" i due triangoli sono ruotati di  $180^\circ$  uno rispetto all'altro e traslati in altezza: il sistema fisso è orientato come il mobile in tale condizione ma ovviamente traslato lungo zeta. Nella generica configurazione del manipolatore rappresentata, il sistema di riferimento mobile risulta traslato e ruotato rispetto a quello fisso: precisamente il sistema mobile è ruotato di  $\rho$  attorno all'asse  $x_b$  (asse di roll), di  $\theta$  attorno a  $y_b$  (asse di pitch) e di  $\psi$  attorno a  $z_b$  (asse di yaw) e ha l'origine traslata lungo i tre assi. In fig. 24 sono evidenziate solo le rotazioni attorno a  $x_b$  e  $y_b$ , ma si può avere anche una rotazione attorno a  $z_b$ : i versi positivi per le rotazioni si ottengono con la "regola del cavatappi", cioè chiudendo le rimanenti dita della mano destra attorno al pollice orientato nella verso positivo dell'asse considerato. Nel disegno gli angoli illustrati sono perciò negativi. Si vuole far notare come, rispetto ai due sistemi di riferimento (fisso e mobile solidale allo sci) utilizzati nei modelli illustrati nel corso del secondo capitolo, per entrambi l'asse X e l'asse Y risultino invertiti. Si ha cioè che, mentre nei modelli si adotta la convenzione usuale di considerare l'asse X come quello lungo l'asse dello sci per il sistema mobile e lungo la direzione di moto per il sistema fisso, nel nostro simulatore a tali direzioni corrisponderanno gli assi Y, mentre gli assi X saranno quindi quelli ortogonali ad essi nel piano orizzontale. Considerando quindi sempre  $x_b$  come asse di roll e  $y_b$  come asse di pitch,

nel simulatore il roll è l'inclinazione rispetto all'ipotetica superficie dell'acqua, mentre il pitch è quella attorno all'asse corrispondente alla direzione di moto: nei modelli del capitolo 2 era il contrario. Questa scelta è necessaria per mantenersi conformi a quelli che sono i sistemi di riferimento comunemente adottati per la piattaforma di Stewart utilizzata per il simulatore.

L'obiettivo è ora quello di illustrare come, a partire dalla posizione del sistema di riferimento mobile rispetto al fisso e dalla sua orientazione in termini di roll, pitch e yaw (detti angoli aeronautici), sia possibile risalire alla lunghezza che dovranno avere i sei cilindri per ottenere quella specifica configurazione. Secondo la convezione utilizzata, la posizione raggiunta è sempre quella che si ha ruotando prima attorno a  $x_b$ , poi attorno a  $y_b$ , ed infine attorno a  $z_b$ : se cambiasimo l'ordine delle rotazioni otterremmo infatti una configurazione finale diversa. Le coordinate dei vertici della piattaforma triangolare superiore rispetto al sistema mobile da cui si parte per il calcolo sono riportati nella tab. 11. Esse sono state ottenute mediante semplici misure di lunghezza così come le coordinate dei vertici del triangolo "fisso" nel sistema fisso riportate invece in tab. 12.

	cil. 1	cil.2	cil.3	cil.4	cil.5	cil.6
x[cm]	-42.5	-2.5	-40	40	2.5	42.5
y[cm]	-21.667	47.67	-26	-26	47.67	-21.667
z[cm]	0	0	0	0	0	0

Tabella 11: Coordinate dei vertici dell'end-effector nel sistema mobile.

	cil. 1	cil.2	cil.3	cil.4	cil.5	cil.6
x[cm]	-42.5	-40	-2.5	2.5	40	42.5
y[cm]	21.667	26	-47.67	-47.67	26	21.667
z[cm]	0	0	0	0	0	0

Tabella 12: Coordinate dei vertici della base fissa rispetto al sistema fisso.

E' necessario, innanzitutto, ottenere le coordinate dei vertici del triangolo "mobile" rispetto al sistema fisso partendo dalle sei coordinate cartesiane indicate con  $x_b^0, y_b^0, z_b^0, \rho, \theta, \psi$ : conoscendo le coordinate rispetto al sistema mobile, ottengo quelle rispetto al sistema

fisso utilizzando le equazioni 44 e 45.

$$R_p^b = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\rho & -\sin\rho \\ 0 & \sin\rho & \cos\rho \end{bmatrix} \quad (44)$$

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_b = [R_p^b] * \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_p + \begin{Bmatrix} x_p^o \\ y_p^o \\ z_p^o \end{Bmatrix}_b \quad (45)$$

La prima equazione consente infatti di calcolare la matrice di rotazione per passare dal sistema fisso della base a quello mobile della pedana superiore. Le coordinate nel sistema fisso si ottengono poi moltiplicando le coordinate nel sistema mobile per tale matrice di rotazione e sommando le coordinate dell'origine del sistema mobile nel sistema fisso, secondo l'equazione 45. E' quindi possibile ottenere la lunghezza di ciascun cilindro i-esimo attraverso l'uso del teorema di Pitagora sapendo le coordinate nel sistema fisso dei suoi due punti estremi (indicati con i pedici p e b nella formula 46).

$$\text{lungh.cilindro}_i = \sqrt{(x_{pi} - x_{bi})^2 + (y_{pi} - y_{bi})^2 + (z_{pi} - z_{bi})^2} \quad (46)$$

### 3.3.3 Schema di controllo della piattaforma

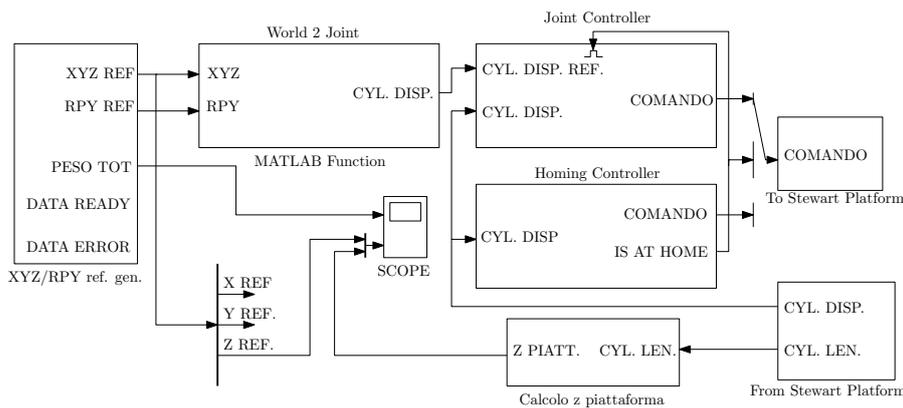


Figura 25: Schema di controllo della piattaforma.

Lo schema di controllo in Simulink della piattaforma di Stewart è riportato in fig. 25: si cercherà ora di spiegarne il funzionamento. All'interno di un blocco chiamato "XYZ/RPY ref. gen." è presente il blocchetto "Packet Input" il quale riceve i dati trasmessi via UDP dal

calcolatore che legge ed elabora i dati provenienti dalla pedana dinamometrica: ogni dato è trasmesso come double, cioè un pacchetto di 8 byte. I dati ricevuti sono le sei coordinate cartesiane, cioè la posizione dell'end-effector rispetto al riferimento fisso e la sua orientazione in termini di roll, pitch e yaw. Viene inoltre ricevuto il peso totale della persona sulla pedana che servirà però solamente per la caratterizzazione dinamica descritta in 3.4.2. Le uscite "Data Ready" e "Data Error" permettono di verificare se ci sono errori nella trasmissione UDP, tramite l'aggiunta di un opportuno visualizzatore allo schema: la prima assumerà valore 1 se si sta ricevendo un dato, la seconda se vi è un errore nella trasmissione; in caso contrario le due uscite assumeranno il valore 0. Quindi, la trasmissione sta avvenendo correttamente solo nel caso in cui "Data Ready" sia a 1 e "Data Error" a 0. La MATLAB Function denominata "World 2 Joint" contiene invece un opportuno codice Matlab (riportano nell'appendice A.1) che esegue esattamente i calcoli di cinematica inversa di posizione precedentemente illustrati.

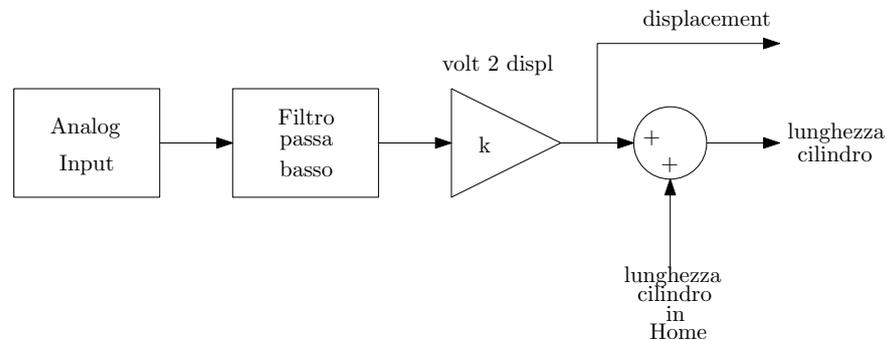


Figura 26: Schema di acquisizione delle lunghezze dei cilindri.

Il blocchetto "From Stewart Platform" serve ad acquisire le lunghezze dei sei cilindri in tempo reale: il suo contenuto viene mostrato in fig. 26; chiaramente tale schema è in realtà replicato per tutte e sei le tensioni lette. In particolare il blocco "Analog Input" acquisisce, attraverso gli ingressi analogici della scheda MULTIQ-3, le sei tensioni in uscita dai potenziometri montati su ciascun cilindro. I potenziometri forniscono una tensione proporzionale al loro allungamento: tale tensione è compresa nel range  $\pm 5$  V e, in particolare, risulta minima in corrispondenza della minima lunghezza dei cilindri e massima alla massima lunghezza. Quando il valore letto è nullo per tutti i cilindri (cioè sono tutti a metà del loro massimo allungamento), ci troviamo nella posizione di *Home* in cui si porterà la piattaforma all'accensione. Le tensioni lette vengono filtrate e moltiplicate successivamente per un opportuno guadagno: in questo modo otteniamo i "displacement" rispetto alla posizione intermedia per ciascun cilindro, mentre, aggiungendo la lunghezza del cilindro in posizione di *Home* ai "displacement", si ottengono le lunghezze effettive dei cilindri.

Nello schema di fig. 25 sono presenti due "controllori": un *Homing controller* e un *Joint controller*. All'accensione del programma viene at-

tivato solamente l'*Homing controller* la cui struttura interna è riportata in fig. 27: la sua uscita "IS AT HOME" è infatti a zero (il flip-flop S-R è subito resettato) e quindi l'altro controllore non è abilitato. Esso è un controllo ad isteresi: l'attuale posizione dei cilindri viene confrontata con la posizione di *Home*, dove i "displacement" sono a zero (corrispondente ad una traslazione in verticale di 68 cm della piattaforma mobile rispetto alla fissa). Se l'errore che ne risulta è positivo, viene dato alle servovalvole un comando positivo perché significa che il cilindro è più corto del necessario; viceversa, se l'errore è negativo, verrà dato alle servovalvole un comando negativo perché il cilindro è più lungo del necessario. Quando il modulo dell'errore è sufficientemente piccolo per tutti i cilindri, il flip-flop S-R viene settato e così il segnale "IS AT HOME" assume il valore 1 dando in uscita dal blocco il valore zero per ciascuna servovalvola (che corrisponde alla posizione di *Home*), ma soprattutto abilitando il *Joint controller* a cui da quel momento passa il controllo delle servovalvole.

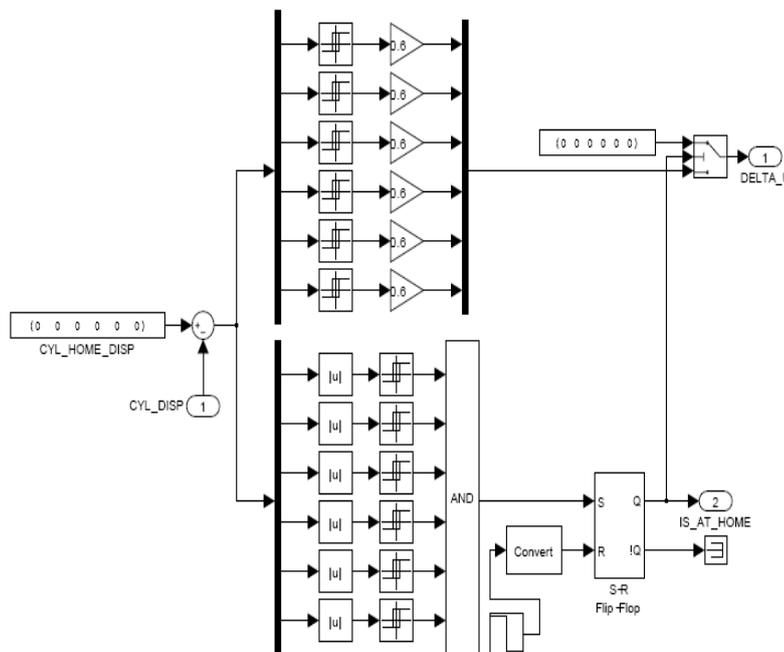


Figura 27: Schema dell'*Homing controller*.

Il *Joint controller* esegue un controllo di posizione ai giunti, cioè su ogni singolo cilindro attraverso sei controllori PID. Lo schema a blocchi di fig. 28 mostra la struttura di tale controllore il quale è formato a sua volta da sei controlli ai giunti di tipo PID + feedforward. L'ingresso di ciascun controllore è dato dal riferimento di posizione (in termini di "displacement" e calcolato attraverso la cinematica inversa) che dovrebbe avere il cilindro nella configurazione della piattaforma voluta, cioè dal suo accorciamento o allungamento rispetto alla posizione intermedia. Tale ingresso viene confrontato con il "displa-

cement" effettivo del cilindro al fine di calcolare l'errore e realizzare un semplice controllore PID a catena chiusa.

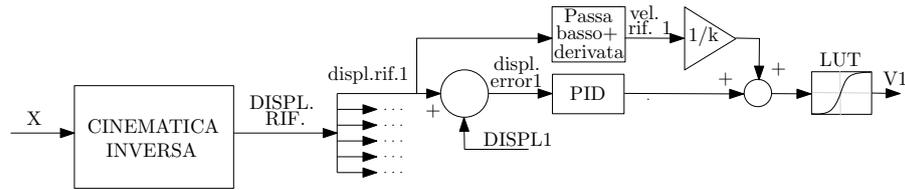


Figura 28: Schema del *Joint controller*.

La taratura del controllore è stata eseguita in modo manuale, cioè aggiustando i valori dei guadagni proporzionale ( $K_p$ ), integrale ( $K_i$ ) e derivativo ( $K_d$ ) a seconda del comportamento della piattaforma di Stewart. In particolare l'effetto dell'aumento dei tre guadagni è il seguente:

- $K_p$  aumenta la banda del sistema e quindi la sua prontezza, diminuisce l'errore a regime, ma aumenta le sovraelongazioni
- $K_i$  annulla l'errore a regime, ma crea problemi di stabilità
- $K_d$  diminuisce sovraelongazioni e transitori e aumenta il margine di fase

I valori sono stati scelti quindi in modo da realizzare un compromesso tra la precisione del posizionamento e la reattività del controllo. Il derivatore è in realtà realizzato assieme ad un filtro passa basso perché altrimenti introdurrebbe impulsi in presenza di variazioni a gradino del riferimento. Il comando alle servovalvole è dato dalla somma delle tre componenti a cui si aggiunge però anche un'azione di feedforward che ha lo scopo di velocizzare l'inseguimento dando in anticipo il grosso del comando, sgravando di tale compito il controllore PID che ha invece il solo compito di "correggere gli errori". Del feedforward e della Lookup Table di fig. 28 si parlerà nel prossimo paragrafo. Lo schema di ciascuno dei sei controllori è riportato in fig. 29.

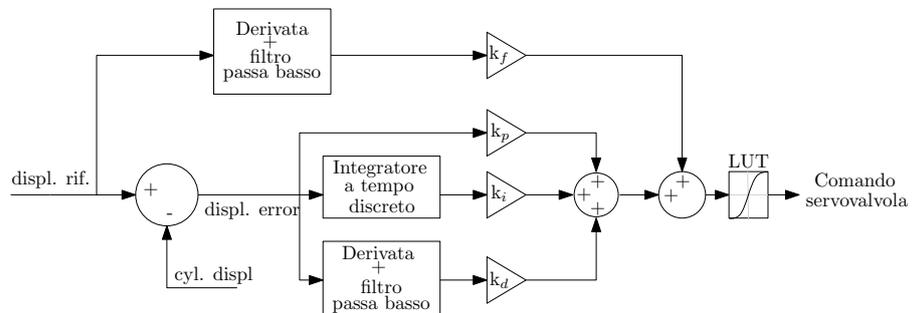


Figura 29: Schema di un controllore PID del *Joint controller*.

Il segnale di comando in tensione viene adattato a quello richiesto in ingresso dalle servovalvole moltiplicandolo per un opportuno guadagno e sommando un opportuno offset all'interno del blocco "To Stewart Platform" di fig. 25.

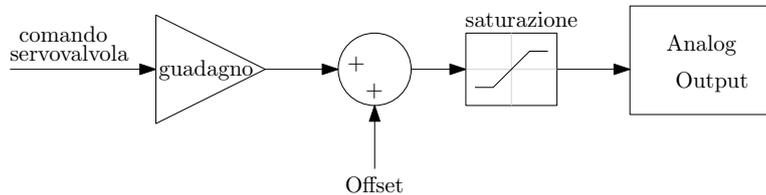


Figura 30: Schema di condizionamento dei comandi alle servovalvole.

Esso contiene anche delle saturazioni in modo da non far andare in battuta i cilindri e il suo contenuto è rappresentato in fig. 30. Il blocchetto "Analog Output", reso disponibile da Real Time Windows Target, fornisce in tempo reale i comandi alle servovalvole attraverso le uscite analogiche della scheda MULTIQ-3. Anche in esso in realtà lo schema è replicato per tutti e sei i comandi alle servovalvole.

#### 3.3.4 Compensazione della "zona morta" e feedforward

Le servovalvole vengono controllate da un segnale elettrico in tensione da 0 a 10 V. A seconda della tensione imposta si riesce a modulare il flusso d'aria compressa che fluisce attraverso la servovalvola: a 5 V la valvola è chiusa nei due sensi, mentre più mi avvicino ai valori massimo e minimo di tensione e più aria fluisce in uno o nell'altro senso determinando la velocità di allungamento (per valori maggiori di 5 V) o accorciamento (per valori minori) del cilindro. Le servovalvole presentano una cosiddetta "zona morta" nell'intorno dei 5 V, cioè in presenza di una tensione in ingresso di poco superiore o inferiore a questo valore la valvola rimane quasi del tutto chiusa ed il cilindro non si muove: questo rende problematico il controllo. Nelle fig. 25 e 31 la zona morta è a cavallo degli zero volt: questo perché l'uscita dei due controllori è positiva se si desidera un allungamento del cilindro e negativa se si desidera un accorciamento, mentre il condizionamento affinché essa sia compresa fra 0 e 10 V è fatta successivamente (come visto in precedenza) all'interno del blocco "To Stewart Platform".

Come si evidenzia nello schema a blocchi di fig. 31, la servovalvola e il pistone si possono quindi schematizzare come un sistema che riceve in ingresso una tensione in Volt la quale corrisponde ad una certa velocità in cm/s del pistone secondo una relazione che risulta pressoché lineare nei due tratti esterni alla zona morta, in corrispondenza della quale invece, pur con un valore non nullo di tensione in

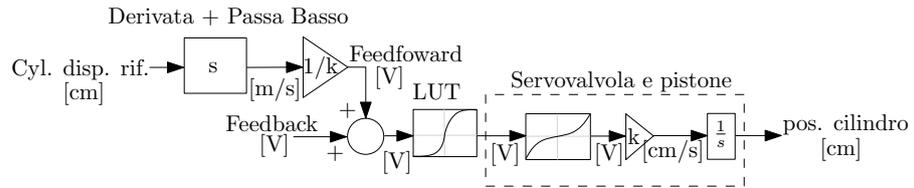


Figura 31: Compensazione della Zona Morta.

ingresso, non ho invece alcuna velocità in uscita. I tre tratti saranno poi raccordati e per praticità immagineremo che la curva abbia i due tratti esterni alla zona morta a pendenza unitaria e che abbia una tensione sia in ingresso che in uscita e poi vi sia in serie un guadagno  $k$  che attua la conversione tra tensione e velocità e che corrisponde alla reale pendenza dei due tratti. I due tratti esterni non sono in realtà lineari ma si assumeranno tali con pendenza pari alla massima. Integrando la velocità del pistone nel tempo si ottiene poi la sua posizione finale in corrispondenza di una data tensione in ingresso per un certo tempo. Al fine di rendere il sistema lineare per il controllo, si è provveduto a compensare la zona morta pre-moltiplicando la funzione non lineare per la sua inversa la quale avrà a sua volta una tensione sia in ingresso sia in uscita e pendenza unitaria nei tratti "lineari". Per determinare quest'ultima curva, corrispondente alla LUT (Look Up Table) di fig. 28, si sono dovute eseguire delle prove specifiche utilizzando lo schema di fig. 32 che, all'interno dell' Embedded MATLAB Function, riproduce il comportamento di una macchina a stati di cui ora si spiegherà brevemente il funzionamento.

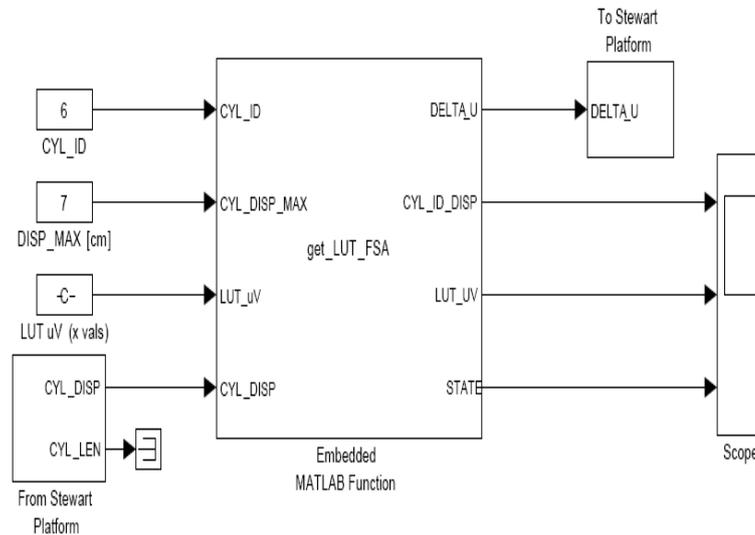


Figura 32: Schema determinazione LUT.

Il codice di tale funzione è riportato nell'appendice A.2. Tali prove sono state eseguite per ciascun cilindro dato che ogni servovalvola ha una "caratteristica velocità vs tensione" differente, pur presentando tutte lo stesso tipo di andamento. La funzione riceve in ingresso il numero del cilindro su cui si vuole eseguire la prova, l'allungamento massimo che potrà raggiungere tale cilindro durante la prova (fissato poi a 7 cm), e il "displacement" attuale del cilindro, oltre ai valori di tensione per i quali si vogliono ottenere i corrispondenti valori di velocità in uscita: la LUT è infatti ottenuta per punti. Le uscite saranno i valori di velocità, i comandi per i cilindri e lo stato in cui ci si trova in ogni istante. La funzione, come si diceva, riproduce infatti il comportamento di una macchina a stati. Lo stato 0 è uno stato iniziale che serve solo ad attendere alcuni istanti all'accensione affinché arrivino in ingresso dati validi per poi portarmi nello stato 1 che corrisponde alla posizione di *Home*, descritta in precedenza. Quando la piattaforma è arrivata in *Home*, nel caso ci siano ancora valori di tensione da imporre, viene selezionato il prossimo valore, viene inizializzato un timer e il sistema si porta nello stato 2. Nello stato 2 si impone il valore di tensione selezionato al cilindro sotto analisi, mentre gli altri restano fermi. Quando il cilindro ha raggiunto la lunghezza prefissata, oppure è scaduto il tempo massimo consentito per raggiungerla, la piattaforma torna in *Home* (cioè nello stato 1) e si passa ad imporre il successivo valore di tensione. Infine, quando tutti i valori di tensione sono stati imposti al cilindro, il sistema passa nello stato 3 e la prova ha termine. In fig. 33 è riportato lo schema della macchina a stati appena descritta.

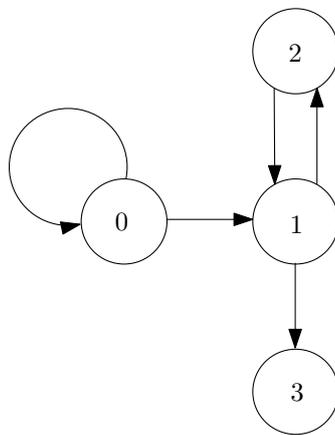


Figura 33: Macchina a stati.

Nello scope dello schema di fig. 32 vengono visualizzati l'andamento della posizione del cilindro rispetto a quella di *Home*, il valore di tensione imposto e lo stato in cui mi trovo. Il primo andamento è un grafico con diversi tratti a rampa, ognuno corrispondente ad un valore diverso di tensione noto, come quello in fig. 34: esso riguarda la prova fatta per valori negativi; un'analogia prova viene fatta per i

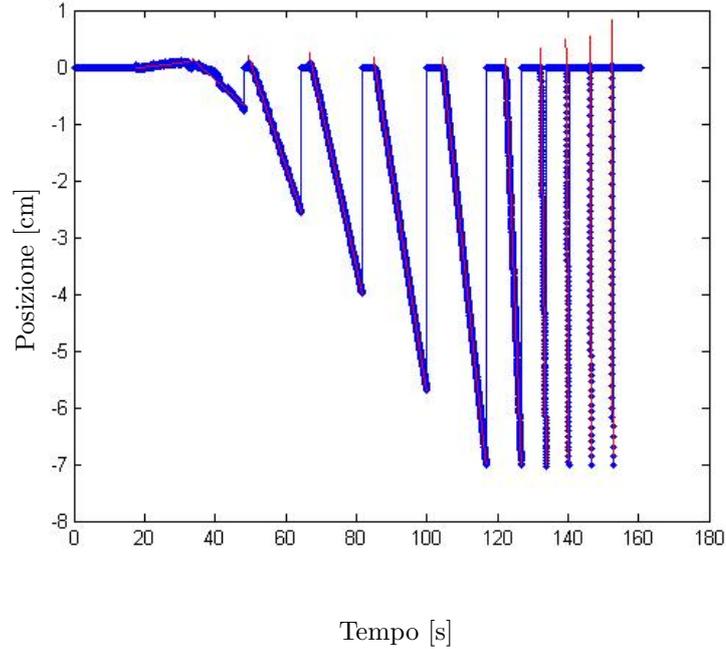


Figura 34: Posizione cilindro nel tempo.

valori positivi in quanto non è stato possibile salvare tutti i dati in un'unica prova. A partire dai dati raccolti sulla posizione dei cilindri si utilizza il codice nell'appendice A.3 per costruire la LUT con tensione in ingresso e corrispondente velocità del cilindro in uscita. Il codice seleziona ogni intervallo di dati e individua la retta che meglio approssima ciascun andamento a rampa. In particolare si cerca di determinare il vettore  $\theta$  che soddisfi l'equazione 47 il cui primo elemento costituisce la pendenza della retta e quindi la velocità di spostamento in cm/s del cilindro comandato con la tensione corrispondente al tratto lineare sotto esame.

$$Y = \begin{bmatrix} X & 1 \end{bmatrix} * \theta = \begin{bmatrix} X & 1 \end{bmatrix} * \begin{bmatrix} m \\ q \end{bmatrix} \quad (47)$$

Chiaramente i termini  $m$ ,  $q$ ,  $1$ ,  $X$  e  $Y$  hanno tanti elementi quanti sono i tratti lineari cioè i punti con cui si costruisce la LUT. Il vettore  $\theta$  è determinato come la soluzione ai minimi quadrati dell'equazione 47, ovvero sarà dato dall'equazione 48, dove  $[X \ 1]^{-1}$  è la pseudoinversa di  $[X \ 1]$ .

$$\theta = \begin{bmatrix} m \\ q \end{bmatrix} = \left( \begin{bmatrix} X & 1 \end{bmatrix}^T \begin{bmatrix} X & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} X & 1 \end{bmatrix}^T Y = \begin{bmatrix} X & 1 \end{bmatrix}^{-1} Y \quad (48)$$

A questo punto è necessario concatenare i dati ottenuti per tensioni positive e negative per ottenere un'unica curva non lineare che

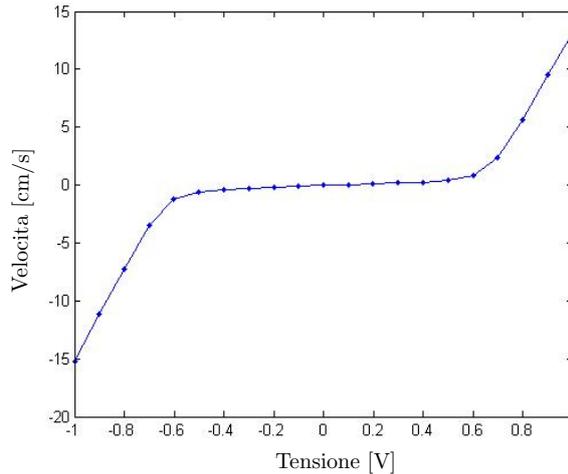


Figura 35: Esempio "LUT velocità vs tensione".

lega la velocità del cilindro alla tensione di comando ("LUT velocità vs tensione"). Tale operazione è resa possibile attraverso il codice Matlab riportato in appendice A.4. Un esempio di "LUT velocità vs tensione" è riportato in fig. 35. Infine, l'ultimo passo consiste nello scalare opportunamente la "LUT velocità vs tensione" per ottenere quella "tensione vs tensione" semplicemente dividendo i valori della prima per la pendenza dei due tratti esterni alla zona morta ottenuta semplicemente facendo il rapporto dei valori di velocità e tensione e facendo la media dei cinque risultati più elevati attraverso il codice riportato nell'appendice A.5. Questa operazione, come detto in precedenza, separa la LUT complessiva da tensione a velocità in una LUT da tensione a tensione, che ha lo stesso andamento ma è opportunamente scalata per avere tensione anche in uscita, e in un guadagno  $k$  (pari alla pendenza originaria dei due tratti) che fa la conversione da tensione a velocità. Questa separazione concettuale è resa necessaria dal fatto che nello schema di fig. 28 viene poi utilizzata l'inversa della "LUT tensione vs tensione" per compensare la non linearità in quanto il segnale di feedback è un segnale di tensione, così come il comando che poi viene dato alle elettrovalvole. Il codice permette anche di plottare con colori diversi le sei "LUT tensione vs tensione" e i valori della derivata di tali funzioni in ogni punto al fine di verificare che siano sempre positivi e quindi che la funzione sia effettivamente invertibile. Le "LUT tensione vs tensione" e le loro derivate sono riportate in fig. 36. Si noti che ogni cilindro ha una LUT diversa.

A questo punto è possibile pure comprendere come viene prodotto il segnale di feedforward del controllore attraverso l'ausilio dello schema di fig. 31. Esso corrisponde al comando teorico di tensione che dovrei avere per ottenere la lunghezza del cilindro desiderata in modo tale che il grosso del comando che viene dato al cilindro sia proprio quello teorico e che il PID compensi solo eventuali errori sul

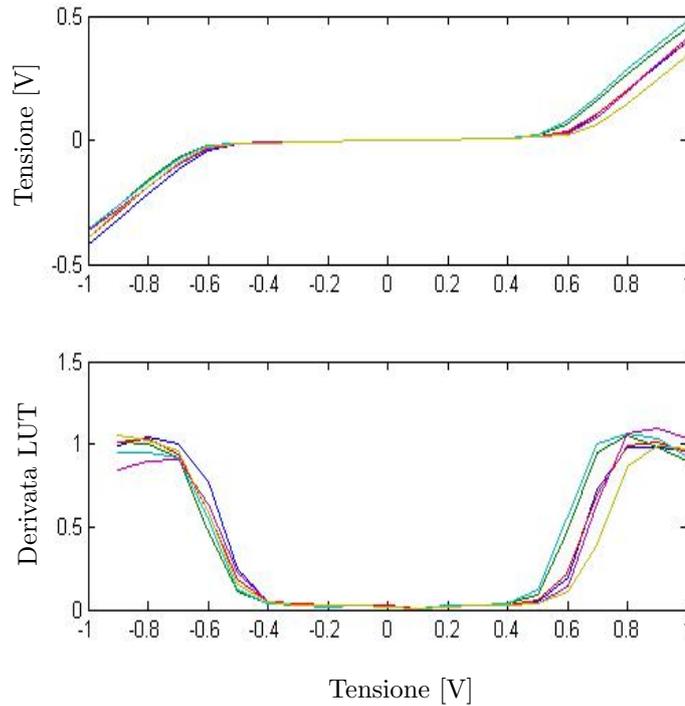


Figura 36: Esempio "LUT tensione vs tensione".

posizionamento finale. Il riferimento di "displacement" del cilindro viene innanzitutto derivato (con filtro passa basso) in modo da ottenere il corrispondente riferimento di velocità. Esso viene poi convertito nella corrispondente tensione di riferimento ottenuta dividendo il riferimento di velocità per la pendenza  $k$  della "LUT velocità vs tensione". Chiaramente la presenza del feedforward rende possibile una velocizzazione del controllo, ma crea anche uno spiacevole rumore dovuto alla derivata filtrata presente nello schema specie perché siamo in presenza di segnali rumorosi provenienti dalla pedana e non di riferimenti. Diminuendo la frequenza di taglio del filtro il rumore diminuisce ma diminuisce anche l'utilità del feedforward: è stato perciò necessario trovare un giusto compromesso.

### 3.3.5 Verifica della bontà del controllo

Come esempio per verificare la bontà del controllo effettuato, si è voluto verificare quanto pronto e preciso fosse l'inseguimento di posizione ad una sinusoida di riferimento lungo l'asse  $z_b$ . Per calcolare la posizione effettiva della pedana mobile della Piattaforma di Stewart lungo l'asse  $z_b$ , all'interno del blocco "Calcolo  $z$  piattaforma" dello schema di fig. 25, si è eseguito il calcolo che verrà ora descritto. Si effettua innanzitutto una media delle lunghezze dei sei cilindri calcolate, come visto in precedenza, dal blocco "From Stewart Platform": la media consente di utilizzare lo stesso valore per tutti i cilindri sem-

plificando il conto e basandosi sul fatto che, muovendosi lungo  $z_b$  esse saranno comunque tutte simili. Si calcola poi la proiezione dei cilindri nel piano  $x_b y_b$  (indicata con  $P$ ) a partire dalle coordinate dei punti di contatto di questi con la pedana superiore (pedice  $p$ ) ed inferiore (pedice  $b$ ) attraverso l'equazione 49: il valore sarà lo stesso per tutti i cilindri e cioè quello riportato nell'equazione stessa. Le coordinate dei punti di contatto sono le stesse nei sistemi fisso e mobile visto che essi saranno semplicemente traslati lungo l'asse  $z_b$  e sono quelle riportate nelle tabelle 11 e 12.

$$P = \sqrt{(x_p - x_b)^2 + (y_p - y_b)^2} = 43,31\text{cm} \tag{49}$$

Lo spostamento effettivo lungo  $z_b$  si calcola a questo punto con l'equazione 50: in essa  $L$  è la lunghezza media dei cilindri.

$$Z = \sqrt{L^2 - p^2} \tag{50}$$

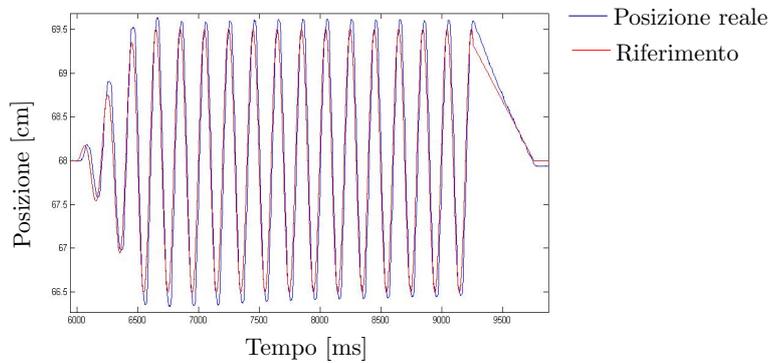


Figura 37: Bontà inseguimento lungo la direzione verticale a frequenza 0,5Hz.

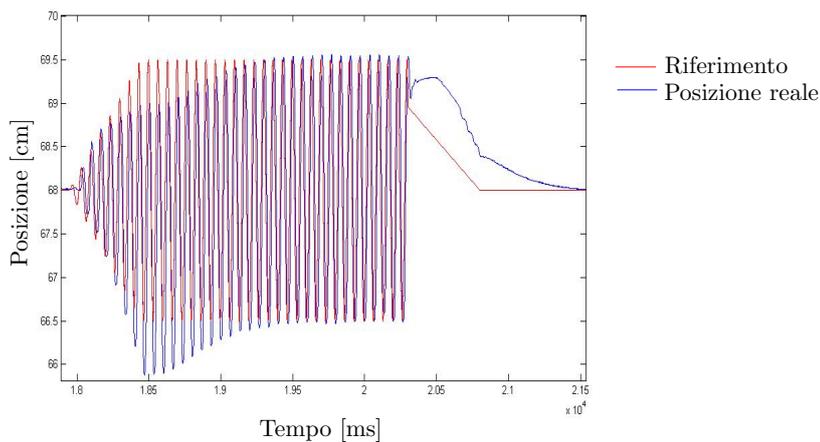


Figura 38: Bontà inseguimento lungo la direzione verticale a frequenza 1,5Hz.

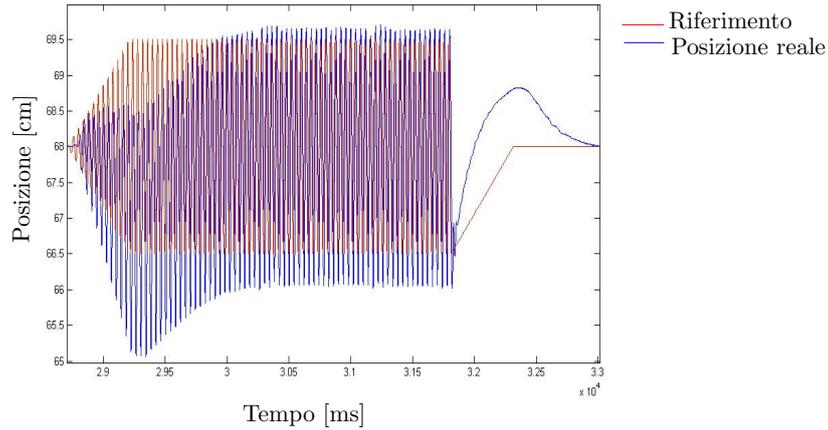


Figura 39: Bontà inseguimento lungo la direzione verticale a frequenza 2,5Hz.

La bontà dell'inseguimento si può osservare nelle fig. 37, 38 e 39: essa decade ovviamente con l'aumento della frequenza. Per le frequenze a cui dovrà lavorare il simulatore l'inseguimento è comunque adeguato. L'esempio è stato fatto lungo  $z_b$ , ma è presumibile attendersi analoghi risultati per altri tipi di posizionamento.

### 3.4 CARATTERIZZAZIONE DINAMICA VERTICALE

Al fine di valutare le prestazioni complessive del sistema realizzato, si è voluto effettuare una caratterizzazione dinamica individuando la risposta in frequenza fra la posizione verticale (cioè lungo l'asse  $z_b$  del sistema fisso) e la forza complessiva misurata in tale direzione. In questo modo si è voluto verificare fino a quali frequenze fosse in grado di lavorare il sistema realizzato e quindi se fosse in grado di operare anche in presenza di una dinamica molto veloce.

#### 3.4.1 Risposta in frequenza ideale

Il sistema sotto esame può essere schematizzato come in fig. 40: la massa  $m_u$  (*unsuspended mass*, cioè massa non sospesa), la cui posizione è indicata con  $z_u$ , è costituita dalla pedana mobile della piattaforma di Stewart e da quella inferiore della pedana dinamometrica che come visto è posta sopra alla prima tramite dei morsetti; la massa  $m_s$  (*suspended mass*, cioè massa sospesa), la cui posizione è indicata con  $z_s$ , rappresenta invece la parte superiore della pedana dinamometrica. In questa analisi si assume che gli antivibranti che collegano le due masse siano dotati non solo di una rigidezza  $k$  (pari a quella con cui sono stati caratterizzati staticamente), ma anche di un coefficiente di attrito viscoso  $b$  e verranno quindi schematizzati come il parallelo di una molla e di uno smorzatore.

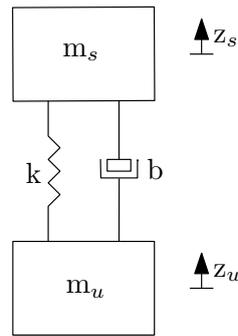


Figura 40: Schema equivalente del simulatore.

Si vuole ora ottenere l'espressione della risposta in frequenza che ha come in ingresso la posizione  $z_u$  della massa non sospesa, imposta muovendo nella posizione opportuna la pedana superiore della piattaforma di Stewart, e come uscita la forza complessiva misurata dagli antivibranti in direzione verticale che, per ciò che abbiamo visto in precedenza nella tesi, sarà data dall'equazione 51.

$$F_z = k(z_s - z_u) \quad (51)$$

Scrivendo l'equazione di moto per la massa sospesa, si avrà che la forza d'inerzia sarà equilibrata dalle forze di reazione della molla (proporzionale alla deformazione) e dalla forza di attrito viscoso (proporzionale alla differenza di velocità fra le due masse): vale cioè l'equazione 52.

$$m_s \ddot{y}_s = k(y_u - y_s) + b(\dot{y}_u - \dot{y}_s) \quad (52)$$

Si può a questo punto passare nel dominio delle trasformate di Laplace e ricavare l'espressione della trasformata della posizione della massa sospesa ( $Z_s$ ) in funzione di quella della massa non sospesa ( $Z_u$ ) riportata in 53.

$$Z_s = \frac{bs + k}{m_s s^2 + bs + k} Z_u \quad (53)$$

Infine si sostituisce la relazione 53 nell'equazione 51 riscritta chiaramente sempre nel dominio di Laplace, ricavando la funzione di trasferimento tra spostamento della massa non sospesa e forza misurata, riportata in 54.

$$P(s) = \frac{F_z(s)}{Z_u(s)} = -\frac{ks^2}{s^2 + \frac{b}{m_s}s + \frac{k}{m_s}} \quad (54)$$

La risposta in frequenza cercata si ottiene semplicemente ponendo  $s = j\omega$  nella 54 ed è riportata in 55. Per bassi valori di frequenza essa assume la prima espressione riportata in 56, comportandosi come una

massa libera, per valori elevati si comporta invece come una molla (seconda espressione riportata in 56).

$$P(j\omega) = \frac{F_z(j\omega)}{Z_u(j\omega)} = -\frac{-\omega^2 k}{-\omega^2 + j\omega \frac{b}{m_s} + \frac{k}{m_s}} \quad (55)$$

$$P(j\omega) = \frac{F_z(j\omega)}{Z_u(j\omega)} = \begin{cases} -\omega^2 m_s & \text{se } \omega \ll \sqrt{\frac{k}{m_s}} \\ -k & \text{se } \omega \gg \sqrt{\frac{k}{m_s}} \end{cases} \quad (56)$$

### 3.4.2 Verifica sperimentale della risposta in frequenza

Al fine di verificare che la risposta in frequenza ricavata sia effettivamente quella corretta per descrivere il sistema di simulazione, si è eseguita una verifica sperimentale imponendo alla piattaforma di Stewart dei movimenti sinusoidali in direzione verticale attorno alla posizione di *Home* a diverse frequenze e ricostruendo la risposta in frequenza per punti. Le frequenze considerate sono state le seguenti: 0.5 Hz, 1 Hz, 1.5 Hz, 2 Hz, 2.5 Hz, 3 Hz e 4 Hz. Si è quindi applicato ciascun riferimento sinusoidale per un certo tempo con ampiezza nota alla piattaforma di Stewart passando poi al successivo e salvando due valori: la posizione in direzione verticale della piattaforma mobile (corrispondente a  $z_u$  e calcolata come descritto nel paragrafo 3.3.5) e la forza misurata lungo la stessa direzione ( $F_z$ ). A questo punto è stato necessario separare ciascuna sinusoide a diversa frequenza all'interno dei due segnali acquisiti: lo si è fatto utilizzando il programma in appendice A.7 che si basa sull'individuare i punti dove la derivata del segnale si annulla per individuare l'inizio e la fine di ciascun tratto. Bisogna poi considerare che i segnali raccolti non saranno perfettamente sinusoidali perché ad essi sarà sovrapposto del rumore di misura: il codice in appendice A.8 seleziona ciascuna sinusoide in ingresso (spostamento) e in uscita (forza) e per ognuna di esse richiama il codice in appendice A.9 il quale svolge le seguenti funzioni:

- togliere eventuali trend dai dati;
- convertire la forza da kg a N;
- convertire lo spostamento da cm a m;
- trovare la sinusoide che rappresenta l'approssimazione ai minimi quadrati dei due segnali.

In particolare, per quanto riguarda l'approssimazione ai minimi quadrati, una generica sinusoide approssimante i dati raccolti si può rappresentare come riportato nelle equazioni 57.

$$\begin{aligned}
\hat{y}(t) &= A_s \sin(\omega t + \phi_s) \\
&= A_s \sin(\omega t) \cos(\phi_s) + A_s \cos(\omega t) \sin(\phi_s) \\
&= (A_s \cos \phi_s) \sin(\omega t) + (A_s \sin \phi_s) \cos(\omega t) \\
&= \begin{bmatrix} \sin(\omega t) & \cos(\omega t) \end{bmatrix} \begin{bmatrix} A_s \cos(\phi_s) \\ A_s \sin(\phi_s) \end{bmatrix} \\
&= \phi^T(t) * \theta
\end{aligned} \tag{57}$$

Lo scopo è quindi quello di stimare  $\theta$  tale che  $\hat{y}(t)$  sia lo stimatore ai minimi quadrati del segnale generico  $y(t)$ , ovvero trovare  $\hat{\theta}_{LS}$  secondo l'equazione 58.

$$\hat{\theta}_{LS} = \operatorname{argmin}_{\theta} [(y(t) - \hat{y}(t))^2] = \operatorname{argmin}_{\theta} [(y(t) - \phi^T(t) \hat{\theta}_{LS})^2] \tag{58}$$

Tale  $\hat{\theta}_{LS}$  è quindi dato dalla 59 dove  $y(t)^{-1}$  è la pseudoinversa del vettore  $y(t)$  contenente tutti i campioni del segnale sotto esame.

$$\hat{\theta}_{LS} = [y(t)^T y(t)]^{-1} y(t)^T \phi(t) = y(t)^{-1} \phi(t) \tag{59}$$

Dal vettore  $\hat{\theta}_{LS}$ , che è formato da due elementi, è a questo punto immediato ricavare l'ampiezza  $A_s$  e la fase  $\phi_s$  attraverso le equazioni 60 e 61.

$$A_s = \sqrt{\theta(1)^2 + \theta(2)^2} \tag{60}$$

$$\phi_s = \operatorname{atan2}[\theta(2), \theta(1)] \tag{61}$$

Tale procedura viene eseguita dal programma in appendice A.9 sia per lo spostamento in ingresso, sia per la forza in uscita. Esso ricava infine modulo e fase della risposta in frequenza ( $A$  e  $\phi$  rispettivamente) punto per punto tramite le equazioni 62 e 63, partendo dalle ampiezze dello spostamento in ingresso ( $A_z$ ), della forza in uscita ( $A_F$ ) e dalle loro fasi (rispettivamente  $\phi_z$  e  $\phi_F$ ).

$$A = \frac{A_F}{A_z} \tag{62}$$

$$\phi = \phi_F - \phi_z \tag{63}$$

A questo punto il controllo ripassa al programma in appendice A.8 che selezionerà un altro tratto (corrispondente ad un'altra frequenza) per ingresso e uscita richiamando nuovamente il programma in

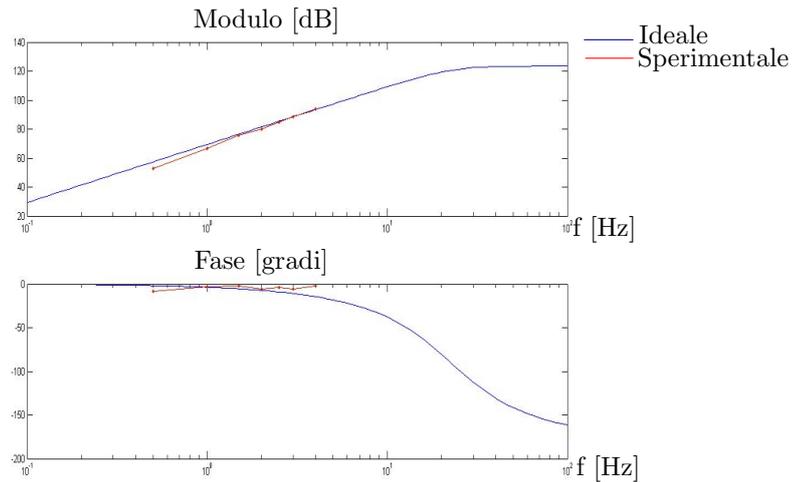


Figura 41: Diagramma di Bode della risposta in frequenza.

appendice A.9 finché non sono state analizzate tutte le frequenze, ognuna corrispondente ad un punto della risposta in frequenza che si vuole determinare. Il programma in appendice A.8 salva inoltre tutte le frequenze, le fasi e i moduli della risposta in frequenza calcolati in un'unica variabile.

Infine il programma in appendice A.10 ha la funzione di rappresentare il confronto in termini di modulo e fase fra la risposta in frequenza ideale riportata in 54 e quella reale calcolata per punti: tale confronto è riportato in fig. 41. Purtroppo con gli attuatori pneumatici di cui è dotata la piattaforma di Stewart non si riesce a sollecitare il sistema oltre i 4 Hz e questo risulta perciò essere un limite dei componenti adottati che non consente al simulatore di riprodurre dinamiche più veloci. Tuttavia, almeno a basse frequenze, c'è comunque una buona corrispondenza fra la risposta in frequenza attesa e quella effettiva del sistema fra spostamento e forza in direzione verticale sia in termini di modulo che di fase, come si può osservare dalla fig. 41.

### 3.5 PROVA SPERIMENTALE PRELIMINARE DEL SIMULATORE

La piattaforma di Stewart, la pedana dinamometrica e gli altri componenti dell'architettura descritta nel paragrafo 3.1, avranno, come detto, lo scopo finale di realizzare un simulatore di sci nautico e in particolare di simulare la sciata rettilinea. Il modello di stato ricavato nel capitolo 2 verrà quindi utilizzato in futuro per ricavare i movimenti da far eseguire alla piattaforma di Stewart a partire dai dati rilevati dalla pedana dinamometrica in termini di forze esercitate dallo sciatore e posizione del baricentro. Esso potrà essere realizzato attraverso una S-Function in modo da realizzare lo schema a blocchi descritto in 2.3.3. Allo stato attuale del progetto, anche per motivi di tempo, non è però ancora stato possibile realizzare ed integrare nello sche-

ma Simulink che elabora i dati letti dalla pedana dinamometrica tale S-Function: tale operazione rimane perciò un possibile sviluppo futuro per chi volesse proseguire con la realizzazione del simulatore. Nonostante ciò, si è comunque voluta testare l'affidabilità dell'architettura realizzata ed in particolare come la pedana dinamometrica sia in grado di seguire gli spostamenti del baricentro dello sciatore ed, in base ad essi, comandare opportuni movimenti della piattaforma di Stewart trasmettendo tali informazioni via UDP al computer che ne realizza il controllo. Si è inoltre testata la possibilità di visualizzare correttamente la posizione del baricentro dello sciatore. Di seguito verranno descritte le prove sperimentali effettuate al fine di effettuare tali verifiche e ne verranno riportati e commentati i risultati ottenuti.

### 3.5.1 Determinazione dei movimenti della piattaforma di Stewart

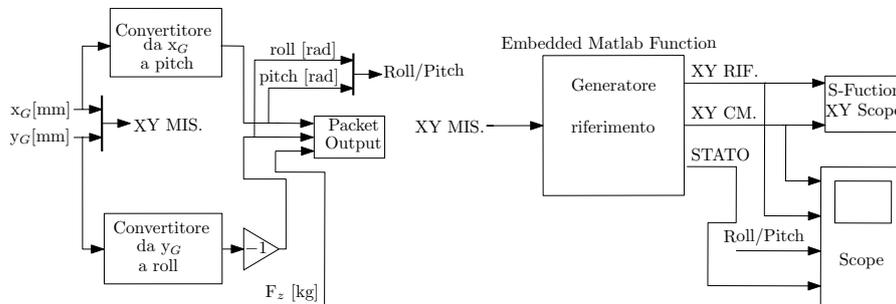


Figura 42: Schema di calcolo dei movimenti della piattaforma di Stewart e di visualizzazione del baricentro per la prova sperimentale del simulatore.

Si consideri quindi nuovamente lo schema di fig. 16 di cui in fig. 42 si riporta solo la parte che ora ci interessa analizzare cioè quella che, a partire dalla posizione del baricentro, ne visualizza la posizione e calcola gli angoli di roll e pitch che dovrà assumere la piattaforma di Stewart. Al fine di testare l'iterazione fra pedana dinamometrica e movimenti della piattaforma di Stewart si è fatto in modo che la rotazione attorno all'asse di roll (cioè l'asse  $x_b$  del sistema fisso) fosse proporzionale alla posizione  $y_G$  del baricentro lungo l'asse  $y_s$  del sistema di riferimento della pedana: quindi più lo sciatore si sposta in avanti col peso nel verso positivo dell'asse  $y_s$  e tanto più la piattaforma si inclinerà verso il basso (angoli di roll negativi); più lo sciatore si sposta indietro col peso nel verso negativo dell'asse  $y_s$  e tanto più la piattaforma si inclinerà verso l'alto (angoli di roll positivi). Allo stesso modo la rotazione attorno all'asse di pitch (cioè l'asse  $y_b$  del sistema fisso) sarà proporzionale alla posizione  $x_G$  del baricentro lungo l'asse  $x_s$ : quindi più lo sciatore si sposta verso destra col peso nel verso positivo dell'asse  $x_s$  e tanto più la piattaforma si abbasserà dalla parte destra (angoli di pitch positivi); più lo sciatore si sposta verso sinistra

col peso nel verso negativo dell'asse  $x_s$  e tanto più la piattaforma si abbasserà dalla parte sinistra (angoli di pitch negativi). Il guadagno pari a  $-1$  nello schema di fig. 42 è necessario proprio perché se il peso si sposta nella direzione positiva di  $y_b$  si dovranno comandare alla piattaforma angoli di roll negativi e viceversa. La conversione dalla coordinata  $y_G$  del baricentro alla corrispondente rotazione attorno a  $x_b$  della piattaforma di Stewart si basa sulla proporzione 64.

$$L_y : \tan(45) = y_G : \tan(\rho) \quad (64)$$

Essa stabilisce che debba essere costante il rapporto fra la coordinata  $y_G$  del baricentro e la pendenza dell'asse  $y_s$  del sistema di riferimento della pedana rispetto all'asse  $y_b$  del sistema fisso. Chiamata  $L_y$  la coordinata che si avrebbe se la piattaforma di Stewart si inclinasse in termini di roll di  $45^\circ$ ,  $L_y$  si determina tenendo conto che la massima inclinazione che realmente può assumere la piattaforma è di poco superiore ai  $10^\circ$  e quindi che quest'ultima debba essere la massima inclinazione che si possa realmente avere. Nell'equazione 64 si dovrà quindi sostituire la massima inclinazione desiderata ( $\rho_{max} = 10^\circ$ ) e il massimo valore della coordinata  $y_G$  ( $y_{G_{max}} = 544 \text{ mm}$  se si suppone di rimanere sempre interni al triangolo formato dai tre antivibranti) per determinare  $L_y$ : si applica quindi l'equazione 65.

$$L_y = \frac{y_{G_{max}}}{\tan(\rho_{max})} = 3085 \text{ mm} \quad (65)$$

A questo punto, nota  $L_y$ , è immediato calcolare il valore dell'angolo di roll corrispondente ad una certa coordinata  $y_G$  del baricentro dello sciatore tramite l'equazione 66.

$$\rho = \arctan\left(\frac{y_G}{L_y}\right) \quad (66)$$

Il contenuto del blocco denominato "Convertitore da  $y_G$  a roll" di fig. 42 è riportato in fig. 43 ed esegue proprio il calcolo riportato in 66 per stabilire che inclinazione imporre in ogni momento alla piattaforma a partire dalla coordinata  $y_G$  del baricentro. E' presente anche un filtro passa basso a  $1\text{Hz}$  per togliere il rumore di misura dai riferimenti da passare alla piattaforma ed un guadagno  $k$  per aggiustare l'inclinazione desiderata.

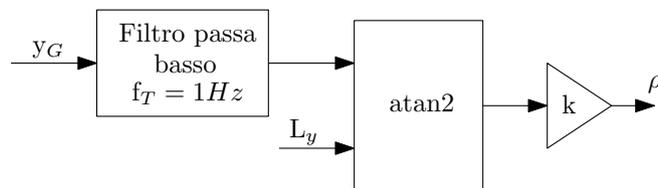


Figura 43: Convertitore da  $y_G$  a roll.

Uno schema analogo, basato su calcoli e ragionamenti analoghi, è contenuto anche all'interno del blocco "Convertitore da  $x_G$  a pitch"

di fig. 42 al fine di imporre una rotazione intorno all'asse  $y_b$  (asse di pitch) proporzionale alla coordinata  $x_G$  del baricentro. In questo caso avremo:  $\theta_{max} = 10^\circ$ ,  $y_{Gmax} = 408,5$  mm ed  $L_x = 2311$  mm.

### 3.5.2 Risultati sperimentali

Sono stati effettuati quattro diversi esercizi al fine di testare l'impianto di simulazione, tutti basati sull'iterazione fra coordinate del baricentro e orientazione della piattaforma di Stewart descritta in 3.5.1. Di seguito verranno descritte le quattro prove e verranno commentati i risultati ottenuti: in ciascuna di esse la pedana dinamometrica è posizionata sopra la piattaforma di Stewart e lo sciatore sale sopra tale struttura dotato di imbracatura di sicurezza agganciata al soffitto. Gli schemi di fig. 16 e 42 si riferiscono alla terza e quarta prova dove la persona dovrà seguire un target di riferimento generato dall'Embedded Matlab Function denominata "Generatore riferimento": nelle altre due prove lo schema appare semplificato non essendovi bisogno di tale generatore e la S-Function di visualizzazione riceverà in ingresso solamente la posizione del baricentro dello sciatore che egli potrà comunque seguire su uno schermo messo di fronte alla pedana. Il codice della S-Function di visualizzazione per gli esercizi con riferimento è riportato in appendice A.11: negli altri due casi è analogo ma con meno ingressi.

#### 3.5.2.1 Movimenti "sinusoidali"

In questo primo esercizio allo sciatore è richiesto di portarsi nel centro geometrico della pedana, di fare dei movimenti portando il peso a destra e a sinistra per cinque volte, di riportarsi al centro a piedi uniti e infine di fare un'altra serie di movimenti in modo da portare il suo peso in avanti e indietro per altre cinque volte. Come si vede dai grafici di fig. 44, il baricentro dello sciatore si sposterà in modo sinusoidale a cavallo dell'asse  $y_s$  durante i primi cinque spostamenti e poi con lo stesso andamento, ma da una parte all'altra dell'asse  $x_s$ , durante i secondi cinque spostamenti. Si può quindi concludere che la pedana è riuscita a seguire i movimenti del baricentro dello sciatore durante l'intera prova visualizzandolo correttamente tramite la S-Function preposta a tale scopo.

In fig. 45 si può invece osservare come alla piattaforma di Stewart vengano imposti degli angoli di pitch e roll tali da far muovere la piattaforma assecondando i movimenti dello sciatore. La fig. 46 mostra infine come si è mosso il baricentro dello sciatore all'interno della pedana durante la prova.

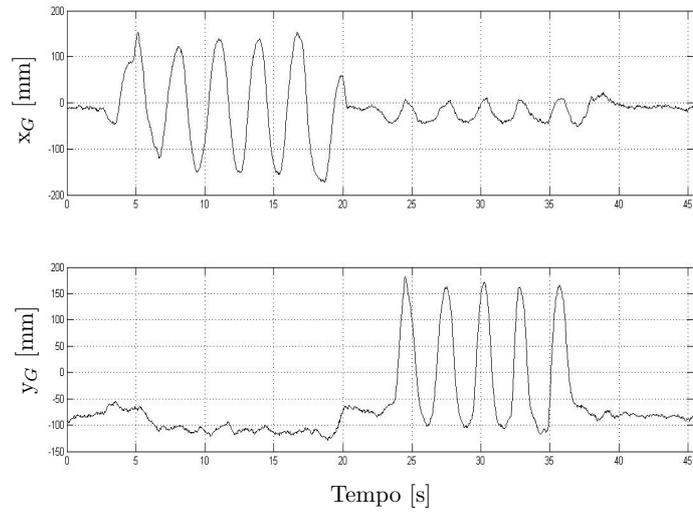


Figura 44: Andamento coordinate baricentro con movimenti sinusoidali.

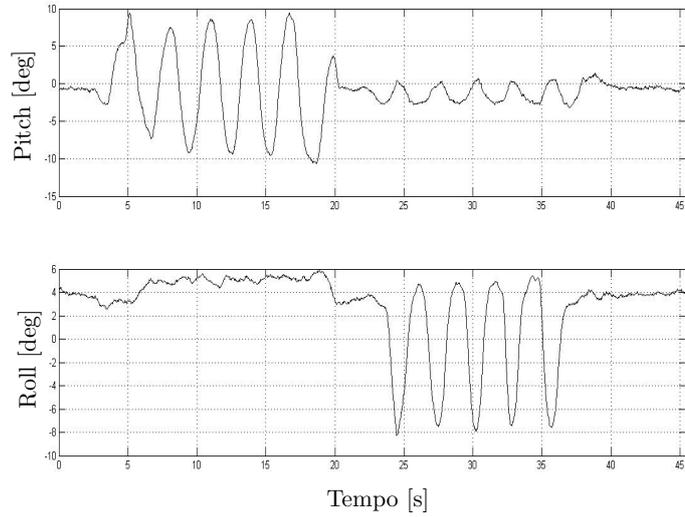


Figura 45: Andamento orientazione piattaforma con movimenti sinusoidali.

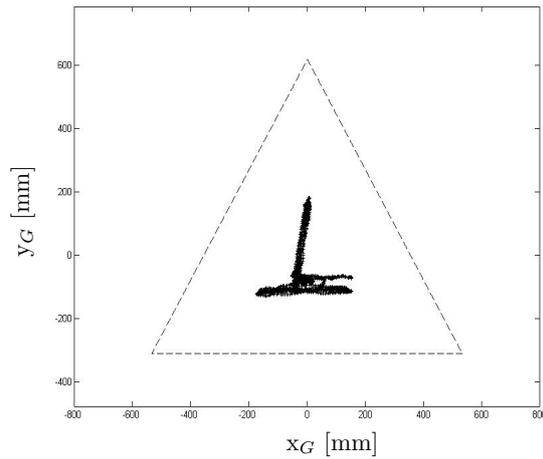


Figura 46: Andamento posizione del baricentro con movimenti sinusoidali.

### 3.5.2.2 Movimenti liberi

Nel secondo esercizio lo sciatore è libero di muoversi nella pedana e, durante la prova, cercherà di occupare tutto lo spazio a disposizione: in questo modo si vuole provare che la pedana è in grado di seguire gli spostamenti del suo baricentro ovunque, e non solamente all'interno dello spazio limitato in cui si svolgeva la prima prova, e comandare di conseguenza la piattaforma di Stewart. Le fig. 47, 48 e 49 danno conferma di quanto appena affermato.

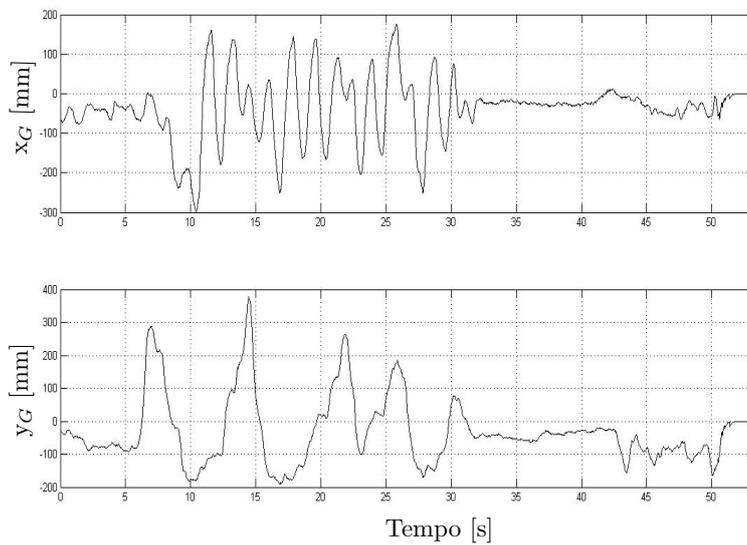


Figura 47: Andamento coordinate baricentro con movimenti liberi.

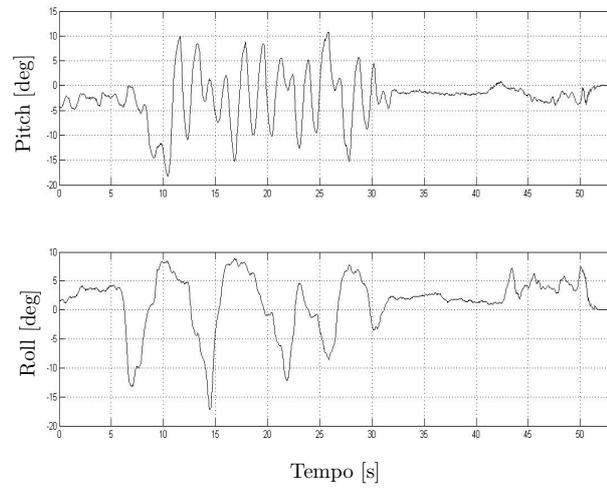


Figura 48: Andamento orientazione piattaforma con movimenti liberi.

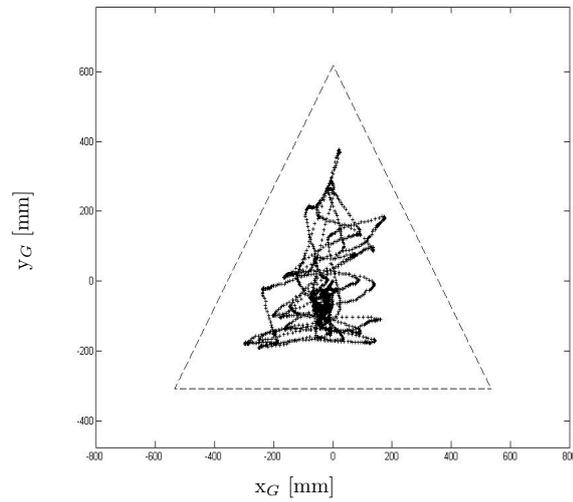


Figura 49: Andamento posizione del baricentro con movimenti liberi.

### 3.5.2.3 *Inseguimento del riferimento bidimensionale senza attacchi*

Nella terza prova lo sciatore dovrà inseguire un riferimento generato in maniera casuale all'interno della pedana spostando il suo peso in modo tale da sovrapporre il suo baricentro al target di riferimento visualizzato nello schermo. Nel momento in cui il baricentro dello sciatore si sovrappone al target con una certa tolleranza, il pallino che lo rappresenta sullo schermo cambia colore e, se egli riesce a mantenere la posizione per qualche secondo, si genererà poi un nuovo riferimento da raggiungere. In fig. 50 viene mostrato ciò che vede lo sciatore durante la prova: il suo baricentro è rappresentato dal pallino blu (che diventerà verde quando si sarà sovrapposto al target), mentre il riferimento è rappresentato dalla croce in rosso.

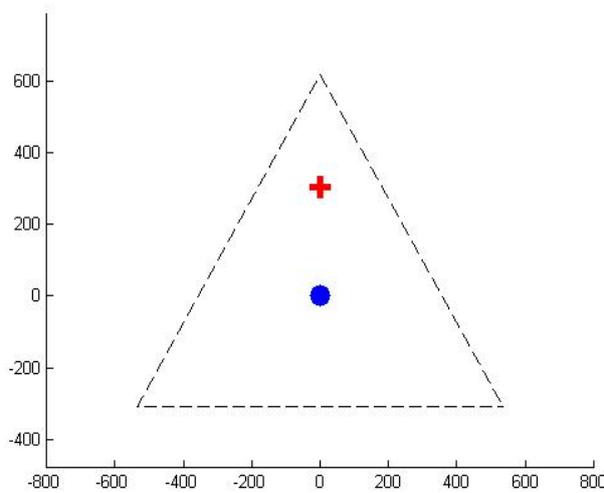


Figura 50: Visualizzazione XY.

Questa prova è molto utile per testare la precisione con cui la pedana riesce ad individuare la posizione del baricentro, ma soprattutto per verificare come lo sciatore sia in grado di spostare il suo peso dove desidera e di conseguenza imporre dei movimenti desiderati alla piattaforma. Tale configurazione, infatti, nel momento in cui il modello di stato della sciata verrà integrato nello schema Simulink del simulatore, potrà servire per individuare una traiettoria ideale che lo sciatore dovrà seguire per equilibrare correttamente le forze che tendono a ribaltarli, come la forza di tiro o quelle esercitate dall'acqua sulla tavola.

Come detto il target di riferimento viene generato dall'Embedded Matlab Function denominata "Generatore riferimento" di fig. 42: il codice di tale funzione è riportato nell'appendice A.12 e riproduce una macchina a due stati di cui ora verrà descritto il funzionamento. La funzione riceve in ingresso le coordinate del baricentro lette dalla pedana e si porta inizialmente nello stato 1, cioè quello in cui il

baricentro dello sciatore non è sovrapposto al target, cosa che invece identifica lo stato 2. Viene quindi chiamata una funzione preposta alla generazione casuale di un nuovo riferimento: esso sarà generato all'interno di un triangolo equilatero interno alla pedana e avente baricentro coincidente con quello di quest'ultima. Come parametri in ingresso tale funzione richiede la lunghezza della base del triangolo equilatero dove si muoverà il riferimento e le coordinate del precedente riferimento che alla prima chiamata vengono identificate come quelle del baricentro della pedana. A partire dalla lunghezza della base del triangolo dove si trovano le possibili locazioni del target, la funzione riesce a generare casualmente un nuovo punto all'interno di esso di cui restituisce le coordinate con la condizione che esso sia sufficientemente distante dal precedente. A questo punto un controllo ad isteresi verifica se la posizione del baricentro dello sciatore si trova o meno all'interno di una regione di tolleranza attorno al riferimento. Nel momento in cui il baricentro entra in tale zona di tolleranza la macchina a stati si porta nello stato 2 in cui dovrà rimanere per un certo tempo, durante il quale viene continuamente verificata la posizione ed il sistema può essere riportato nello stato 1 qualora il baricentro uscisse dalla zona di tolleranza. Se lo sciatore riesce a mantenere la posizione per il tempo stabilito, viene generato un nuovo riferimento sufficientemente lontano dal precedente, riportando quindi la macchina nello stato 1. Le uscite del "Generatore riferimento" saranno la posizione del riferimento, quella del baricentro dello sciatore e lo stato che verranno passate alla S-Function di visualizzazione che, come detto, consentirà di mostrare le due posizioni in uno schermo in modo che lo sciatore possa vedere la posizione del riferimento e quella del suo baricentro all'interno della pedana ed eseguire l'esercizio. Lo stato serve invece alla S-Function per far cambiare il colore del pallino rappresentante il baricentro dello sciatore a seconda che egli si sia portato sopra al riferimento o meno.

Per quanto riguarda i risultati ottenuti, la fig. 51 mostra come lo sciatore sia perfettamente in grado di sovrapporre il suo baricentro al riferimento senza particolari difficoltà, cosa che avviene in corrispondenza dei tratti rosa; la piattaforma lo seguirà nei suoi spostamenti come dimostrano i riferimenti che gli vengono imposti in termini di pitch e roll riportati in fig. 52. Infine, in fig. 53, vengono mostrate le posizioni di baricentro (in nero) e target (in rosso) durante la prova: si nota come i riferimenti vengano generati in modo da coprire più possibile lo spazio a disposizione. Si può quindi concludere che, dopo l'integrazione del modello della sciata, sarà molto probabilmente possibile dare modo allo sciatore di allenarsi seguendo un riferimento che gli indichi i movimenti da effettuare per seguire la traiettoria ideale e mantenere l'equilibrio; la possibilità di avere sensazioni simili a quelle che avrà sul campo di gara sarà garantito dai movimenti della piattaforma di Stewart in base alla sua posizione su di essa.

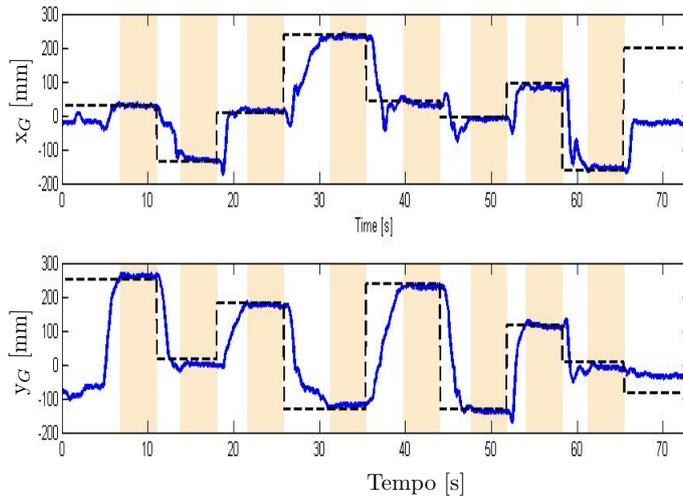


Figura 51: Confronto coordinate con target bidimensionale.

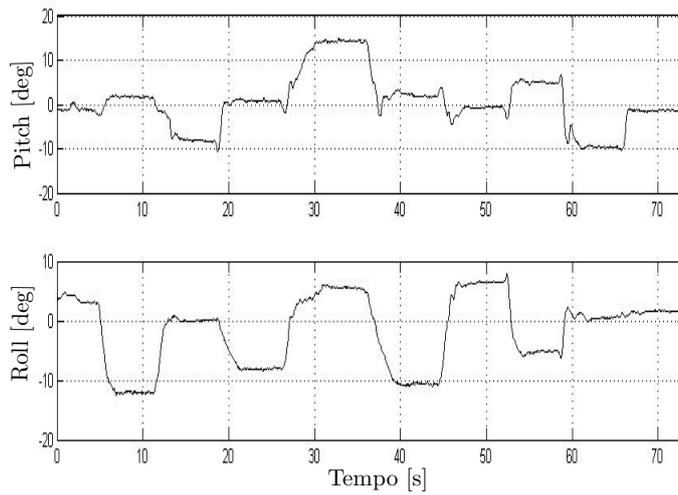


Figura 52: Andamento orientazione piattaforma con riferimento bidimensionale.

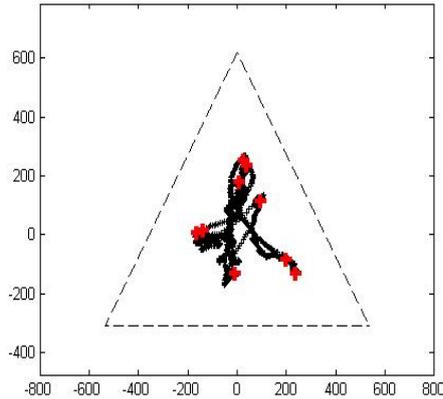


Figura 53: Andamento posizione del baricentro con riferimento bidimensionale.

#### 3.5.2.4 *Inseguimento del riferimento monodimensionale con attacchi*

Nelle prime tre prove che sono state illustrate lo sciatore poteva muovere liberamente i piedi sulla pedana per svolgere gli esercizi richiesti. In realtà però nello sci nautico i piedi sono posizionati all'interno di due attacchi posti uno dietro l'altro sulla tavola da sci. Si è quindi voluto testare l'apparato di simulazione anche in presenza di tali attacchi dove lo sciatore deve infilare i piedi. In questo caso però il baricentro dello sciatore non si sposta significativamente lungo l'asse  $x_s$  della pedana ovvero lateralmente rispetto alla ipotetica direzione di moto. Affinché ciò avvenisse sarebbe infatti necessario che anche la pedana si spostasse nella stessa direzione in cui lo sciatore sbilancia il suo peso e non che si inclinasse solamente perché altrimenti è chiaro che comunque il suo baricentro rimarrà in posizione centrale dove si trovano i piedi. E' chiaro però anche che una cosa del genere non potrebbe in ogni caso essere simulata all'interno dello spazio stretto del laboratorio ed è comunque anche vero che fin dall'inizio ci si era posti l'obiettivo di simulare solamente la sciata rettilinea ed è su essa che è stato costruito il modello di stato illustrato nel capitolo 2 per il quale la coordinata  $x_G$  non è necessaria in quanto assunta sempre nulla. Tenendo conto di tutta questa serie di considerazioni, per eseguire quest'ultima prova si è modificato leggermente il "Generatore riferimento" per avere un target che si spostasse solamente avanti e indietro lungo l'asse  $y_s$ . Nel quarto esercizio perciò lo sciatore dovrà portare avanti e indietro il proprio peso con i piedi all'interno degli attacchi per cercare di sovrapporre il proprio baricentro al target di riferimento visualizzato. Anche in questo caso, come si può osservare in fig. 54, lo sciatore è perfettamente in grado di eseguire l'esercizio portandosi rapidamente col proprio baricentro sopra al target. La coordinata  $x_G$  tende a rimanere sempre al di sotto del riferimento, ma in realtà essa rimane comunque sempre all'interno della tolleranza.

za che si era stabilita che è pari a 40 mm. Si può quindi concludere che anche in questa situazione, che rispecchia maggiormente le reali condizioni di sciata rettilinea che si vogliono simulare, il sistema è in grado di seguire gli spostamenti del baricentro dello sciatore ed imporre movimenti alla piattaforma di Stewart in base alle sue coordinate come si vede in fig. 55. La fig. 56 mostra infine la posizione di riferimento (in rosso) e baricentro (in nero) durante la prova: si vede ancora una volta come il riferimento venga generato in modo distribuito lungo la direzione di moto e come, con semplici movimenti del peso avanti e indietro, lo sciatore riesca facilmente a sovrapporsi ad esso con il suo baricentro.

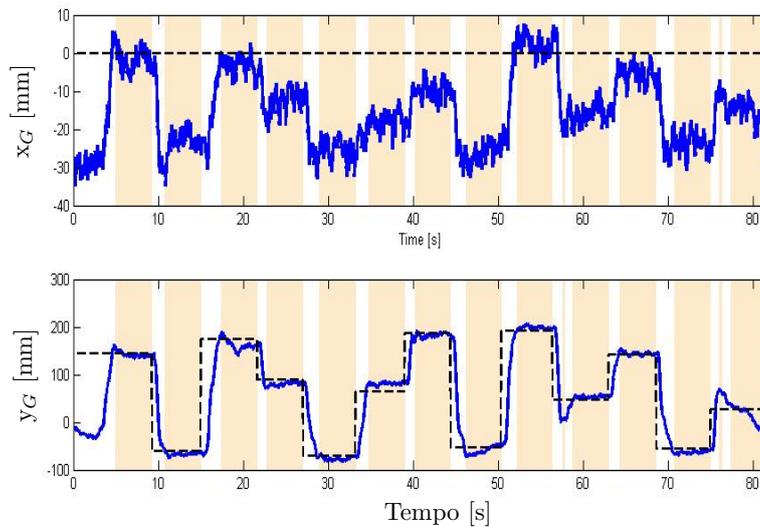


Figura 54: Confronto coordinate con target monodimensionale.

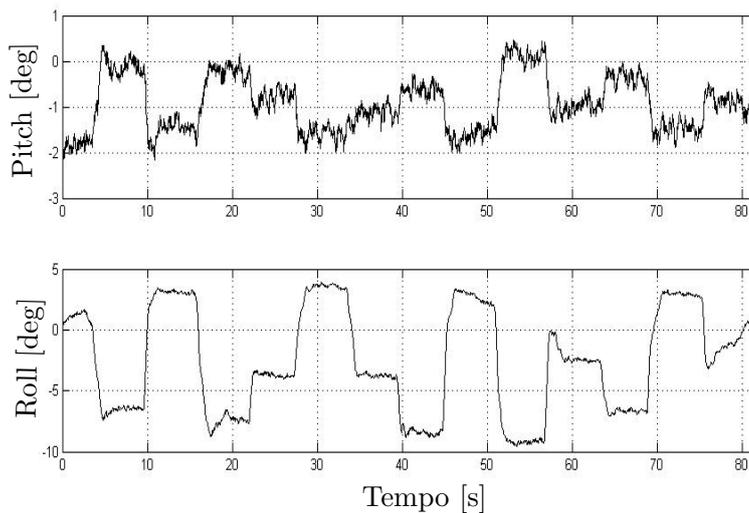


Figura 55: Andamento orientazione piattaforma con target monodimensionale.

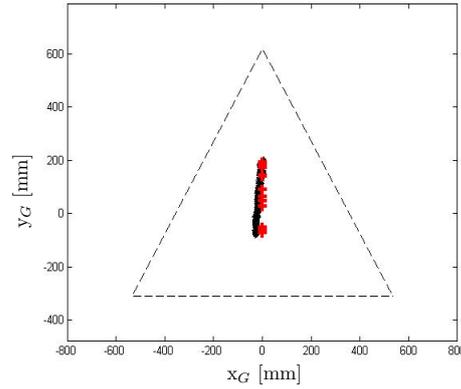


Figura 56: Andamento posizione del baricentro con target monodimensionale.

### 3.5.2.5 Considerazioni finali sulle prove effettuate

Dall'insieme delle quattro prove ed in particolare da quest'ultima, si può affermare che il sistema di simulazione è perfettamente in grado di seguire la posizione del baricentro dello sciatore ed imporre movimenti conseguenti alla piattaforma di Stewart che poi questa realmente esegue in quanto si è già discusso in precedenza della bontà del controllo realizzato (paragrafo 3.3.5). Si può quindi concludere che il simulatore, da quanto si è potuto verificare sperimentalmente, sembrerebbe pronto ad essere completato con l'inserimento di una S-Function che riproduca il modello di stato della sciata rettilinea, in quanto esso non fa altro che legare in modo diverso (e più complesso di un semplice calcolo proporzionale) la posizione del baricentro ai movimenti della piattaforma per dare sensazioni molto più simili ad una reale sciata.

## CONCLUSIONI

---

In questo progetto di tesi si è cercato di sviluppare un simulatore per sci nautico e in particolare per la disciplina dello slalom utilizzando una pedana dinamometrica posizionata sopra ad una piattaforma di Stewart: la pedana ha il compito di rilevare in tempo reale la posizione e gli spostamenti del baricentro dello sciatore e le forze che egli esercita sulla pedana allo scopo di imprimere alla piattaforma di Stewart i movimenti più adeguati per riprodurre la dinamica e le sensazioni della sciata. A tale scopo è stato sviluppato un opportuno modello di stato restringendosi però, a causa della complessità del problema e dell'attrezzatura e dello spazio limitato disponibili in laboratorio, al caso semplificato della sciata rettilinea con e senza traino. E' stato elaborato uno schema in Simulink per rilevare correttamente e visualizzare la posizione del baricentro dello sciatore in tempo reale: a tale scopo si è dovuta effettuare anche una taratura dei sensori preposti a misurare le forze esercitate dallo sciatore sulla pedana in direzione "orizzontale" la cui verifica sperimentale eseguita sia con piastre di cemento, sia con una persona sulla pedana, ha portato a dei buoni risultati, pur trascurando l'isteresi che in realtà li caratterizza. Si è inoltre progettato un controllo per la piattaforma di Stewart che è risultato particolarmente difficoltoso a causa di una zona morta presente nelle servovalvole che si è dovuta compensare adeguatamente ottenendo però anche in questo caso dei risultati più che soddisfacenti. Si è infine testato l'intero apparato di simulazione con delle semplici prove in cui al variare della posizione del baricentro si andava a ruotare in modo proporzionale la piattaforma di Stewart verificando che il sistema progettato è perfettamente in grado di seguire e visualizzare in tempo reale gli spostamenti dello sciatore sulla pedana e produrre dei movimenti conseguenti della piattaforma. Purtroppo, soprattutto per motivi di tempo, non è ancora stato possibile realizzare una S-Function che riproduca il modello di stato della sciata rettilinea allo scopo di muovere la piattaforma in modo da simulare le effettive condizioni di sciata. Tuttavia, da quanto verificato sperimentalmente, con l'apparato di simulazione che è stato predisposto, non sembrerebbero esserci particolari difficoltà nel completare questo progetto. Fra i possibili sviluppi futuri vi è inoltre quello di progettare un sistema per simulare anche il bilancino e la tensione della corda di traino, ad esempio con l'utilizzo di un motore elettrico, e quello di trovare un sistema per stimare l'altezza del baricentro della persona oltre alla sua posizione sul piano. Fra i limiti dell'apparato di simulazione realizzato vale infine la pena di citare l'impossibilità di riprodurre dinamiche troppo veloci (ovvero sopra ai 3 o 4 Hz) a

causa degli attuatori pneumatici della piattaforma di Stewart e il fatto che gli angoli di inclinazione raggiunti nello slalom sono di norma superiori alle massime inclinazioni (attorno ai  $10^\circ$ ) che si riescono ad ottenere con l'utilizzo della piattaforma di Stewart e che quindi gli angoli dovranno essere scalati opportunamente rispetto alla realtà.

## APPENDIX





## CODICI MATLAB

---

### A.1 CINEMATICA INVERSA

```
1 function [CYL_DISP, CYL_LEN, ERR] = World2Joint(XYZ, RPY,
2     ...
3     CYL_LEN_home, CYL_LEN_min, CYL_LEN_max, ...
4     CYL_XYZb, CYL_XYZp)
5     %#codegen
6
7     %% Temporary state to be used in case of kinematics
8     inversion errors
9     persistent CYL_LEN_prev;
10
11    if isempty(CYL_LEN_prev),
12        CYL_LEN_prev = zeros(6,1);
13        CYL_LEN_prev = CYL_LEN_home;
14    end;
15
16    %% Compute cylinder elongations
17    a = zeros(3,6);
18    CYL_LEN = zeros(6,1);
19    ERR = zeros(6,1);
20
21    Rbp = get_DCM(RPY);
22
23    for k=1:6,
24        a(:,k) = Rbp*CYL_XYZp(:,k) + XYZ - CYL_XYZb(:,k);
25        CYL_LEN(k) = norm(a(:,k));
26        if CYL_LEN(k) < CYL_LEN_min(k),
27            ERR(k) = -k;
28        elseif CYL_LEN(k) > CYL_LEN_max(k),
29            ERR(k) = k;
30        end;
31    end;
32
33
34    %% In case of errors, keep the same pose
35    if any(ERR),
36        CYL_LEN = CYL_LEN_prev;
37    end;
38
39
40    %% Output cylinders displacements
41    CYL_DISP = zeros(6,1);
42    CYL_DISP = CYL_LEN - CYL_LEN_home;
43
44
```

```

45 %% Update previous value of L
46 CYL_LEN_prev = CYL_LEN;
47
48 end % World2Joint
49
50
51
52 % .....%
53
54 function DCM = get_DCM(RPY)
55
56 cr = cos(RPY(1)); sr = sin(RPY(1));
57 cp = cos(RPY(2)); sp = sin(RPY(2));
58 cy = cos(RPY(3)); sy = sin(RPY(3));
59
60
61 DCM = zeros(3,3);
62
63 DCM(1,1) = cy*cp;
64 DCM(2,1) = sy*cp;
65 DCM(3,1) = -sp;
66
67 DCM(1,2) = cy*sp*sr - sy*cr;
68 DCM(2,2) = sy*sp*sr + cy*cr;
69 DCM(3,2) = cp*sr;
70
71 DCM(1,3) = cy*sp*cr + sy*sr;
72 DCM(2,3) = sy*sp*cr - cy*sr;
73 DCM(3,3) = cp*cr;
74
75 end % get_DCM

```

## A.2 MACCHINA A STATI

```

1 function [DELTA_U, CYL_ID_DISP, LUT_UV, STATE] = get_LUT_FSA
   (CYL_ID, CYL_DISP_MAX, LUT_uV, CYL_DISP)
2
3 persistent FSA_state;
4 persistent kLUT;
5 persistent kTIMEOUT;
6
7
8 %% Init persistent vars
9
10 if isempty(FSA_state),
11     FSA_state = 0;
12 end;
13
14 if isempty(kLUT),
15     kLUT = 0;
16 end;
17
18 if isempty(kTIMEOUT),
19     kTIMEOUT = 0;

```

```

20 end;
21
22
23 %% Determine FSA output
24
25 CYL_DISP_ERR = zeros(6,1) - CYL_DISP;
26
27 if FSA_state == 0,           % at startup, do nothing for
    one simulation step (wait for valid measurement input)
28     DELTA_U = zeros(6,1);
29     CYL_ID_DISP = 0;
30     LUT_UV=0;
31 elseif FSA_state == 1,      % bring platform to home
    position
32     DELTA_U = home_ctrl(CYL_DISP_ERR);
33     CYL_ID_DISP = 0;
34     LUT_UV=0;
35 elseif FSA_state == 2      % open servovalve and wait
    for proper cylinder elongation
36     DELTA_U = zeros(6,1);
37     DELTA_U(CYL_ID) = LUT_uV(kLUT);
38     CYL_ID_DISP = CYL_DISP(CYL_ID);
39     LUT_UV=LUT_uV(kLUT);
40 else                        % end of test; do nothing
41     DELTA_U = zeros(6,1);
42     CYL_ID_DISP = 0;
43     LUT_UV=0;
44 end;
45 STATE = FSA_state;
46
47
48 %% Determine next FSA state
49
50 if FSA_state == 0,
51     FSA_state = 1;
52 elseif FSA_state == 1,
53     if is_at_home(CYL_DISP_ERR),
54         % is at home
55         if kLUT < length(LUT_uV),
56             FSA_state = 2;
57             kLUT = kLUT + 1;
58             kTIMEOUT = 100*15; % timeout [s] (15s with
                Ts = 0.01s)
59         else
60             FSA_state = 3;
61         end;
62     else
63         % is NOT at home
64         FSA_state = 1;
65     end
66 elseif FSA_state == 2,
67     if ( abs(CYL_DISP(CYL_ID)) > CYL_DISP_MAX ) || (
        kTIMEOUT <= 0),
68         FSA_state = 1;
69     else
70         kTIMEOUT = kTIMEOUT - 1;
71         FSA_state = 2;

```

```

72     end;
73 else
74     FSA_state = 3;
75 end;
76
77
78 end
79
80 % .....%
81
82 function DELTA_U = home_ctrl(CYL_DISP_ERR)
83
84     persistent yh_prev;
85
86     %  init persistent vars
87     if isempty(yh_prev),
88         yh_prev = ones(6,1);
89     end;
90
91     %  get hysteresis outputs
92     yh = hyst_cmp(CYL_DISP_ERR, -0.005, 0.005, -1, 1,
93         yh_prev);
94
95     %  update previous hysteresis outputs
96     yh_prev = yh;
97
98     %  get homing controller outputs
99     DELTA_U = 0.6 * yh;
100 end
101
102 % .....%
103
104 function y = is_at_home(CYL_DISP_ERR)
105
106     persistent yh_prev;
107
108     %  init persistent vars
109     if isempty(yh_prev),
110         yh_prev = zeros(6,1);
111     end;
112
113     %  check cylinder displacements
114     yh = hyst_cmp(abs(CYL_DISP_ERR), 0.05, 0.08, 1, 0,
115         yh_prev);
116
117     %  update previous hysteresis outputs
118     yh_prev = yh;
119
120     %  return 1 if all the displacements are below the spec
121     %  threshold
122     y = all(yh);
123 end
124
125 % .....%
126
127 function y = hyst_cmp(u, uL, uH, yL, yH, y_prev)

```

```

126
127     N = length(y_prev);
128     y = zeros(N,1);
129
130     for k = 1:N,
131         if (u(k) > uH) && (y_prev(k) == yL),
132             y(k) = yH;
133         elseif (u(k) < uL) && (y_prev(k) == yH),
134             y(k) = yL;
135         else
136             y(k) = y_prev(k);
137         end;
138     end;
139
140 end

```

### A.3 DETERMINAZIONE LUT VELOCITÀ VS TENSIONE

```

1 %% Load experimental data
2
3 expres_path = './expres/';
4 expres_filename = 'expres_cyll_uV_neg';
5
6 load(strcat(expres_path, expres_filename));
7
8 %% Get sampling time
9
10 t = expres.time;
11 Ts = mean(diff(t)); % sampling time;
12
13
14 %% Select useful data
15
16 y = expres.signals(1).values; % cylinder
    displacement
17 u = expres.signals(2).values; % servovalve
    input voltage
18 state = expres.signals(3).values; % FSA state
19
20 % find first/last index of each valid test
21 kb = find(diff(state) == 1); % first index(es)
22 ke = find(diff(state) == -1); % last index(es)
23
24
25 %% Compute cylinder velocity (LUT y-axis entry) for each
    test
26
27 LUTN = length(kb);
28 LUTx = zeros(1,LUTN);
29 LUTy = zeros(1,LUTN);
30 THv = zeros(2,LUTN);
31 kdel = 0;
32 for k = 1:LUTN,

```

```

33     Y = y(kb(k)+kdel:ke(k)); %
           measurement vector
34     N = ke(k)-(kb(k)+kdel)+1;
35     PHI = [(kb(k)+kdel:ke(k))*Ts; ones(1,N)].'; %
           regressor matrix
36     THv(:,k) = PHI\Y; % LS estimate
37     LUTx(k) = mean(u(kb(k)+kdel+1:ke(k)));
38     LUTy(k) = THv(1,k);
39 end;
40
41
42 %% Plot results
43 figure;
44 plot(t, y, '.-');
45 hold on;
46 for k = 1:LUTN,
47     tls = t(kb(k):ke(k));
48     yls = [tls, ones(length(tls),1)] * THv(:,k);
49     plot(tls, yls, 'r');
50 end
51
52 figure;
53 plot(LUTx, LUTy, '.-');
54
55 %
56 save LUT_cyl1_uV_neg LUTx LUTy

```

#### A.4 CONCATENAZIONE DATI LUT

```

1 %% Load experimental data
2
3 load LUT_cyl6_uV_pos
4 LUTx_pos = LUTx;
5 LUTy_pos = LUTy;
6 load LUT_cyl6_uV_neg
7 LUTx_neg = LUTx;
8 LUTy_neg = LUTy;
9
10 %% Concatenated data
11
12 LUTx = [fliplr(LUTx_neg), LUTx_pos];
13 LUTy = [fliplr(LUTy_neg), LUTy_pos];
14
15 %LUTy(11) = 0;
16
17 %% Plot LUT
18
19 plot(LUTx, LUTy, '.-')
20
21 %% Save LUT
22
23 save LUT_CYL6 LUTx LUTy;

```

## A.5 ADATTAMENTO LUT

```

1 %% Load speed vs volt LUTs
2
3 load data_LUT_speed_vs_volt;
4
5 LUT_uV_v = LUT;      % in [V], out [cm/s]
6 LUT = {};           % in [V], out [V]
7 Kff = zeros(1,6);   % new FF gains
8
9 %% Get volt vs volt LUTs
10
11 for k = 1:6,
12     dvdV = sort(diff(LUT_uV_v(k).v)./diff(LUT_uV_v(k).uV),
13                 'descend');
14     dvdV_max = mean(dvdV(1:5));
15     LUT(k).uV = LUT_uV_v(k).uV;
16     LUT(k).yV = LUT_uV_v(k).v/dvdV_max;
17     LUT(k).dvdV_max = dvdV_max;
18 end;
19
20 %% Save data
21
22 save data_LUT_volt_vs_volt LUT;
23
24
25 %% Plot results
26
27 figure;
28 cmap = colormap(lines(6));
29 for k = 1:6,
30     subplot(2,1,1);
31     plot(LUT(k).uV, LUT(k).yV, 'Color', cmap(k,:))
32     hold on
33
34     subplot(2,1,2);
35     plot(LUT(k).uV(2:end), diff(LUT(k).yV)./diff(LUT(k).uV)
36           , 'Color', cmap(k,:))
37     hold on
38 end;

```

## A.6 CODICE DETERMINAZIONE RIGIDENZE ORIZZONTALI

```

1 %% Load data
2
3 path = 'C:\Documents and Settings\lab\Desktop\Matteo\
4        STEWART_PLATFORM_Control\caratt_sens_taglio\';
5 filename = 'sens_taglio_%d_prova_%d';
6 sensor_id = 6;
7 N = 4;

```

```

7
8
9 %% Get spring constant LS estimate
10
11 x = [];
12 F = [];
13 for k = 1:N,
14     str = strcat(path, sprintf(filename,sensor_id,k));
15     data = xlsread(str);
16     F = [F; -data(:,1)];
17     x = [x; data(:,2)];
18 end;
19 K = x\F;
20
21
22 %% Plot results
23
24 xM = max(x);
25 Fh = K*[0;xM];
26
27 figure;
28 plot(x,F, '.');
29 hold on
30 plot([0;xM], Fh, 'k—');
31
32 fprintf('Spring constant (sensor %d): %5.3f N/mm\n',
        sensor_id, K)

```

## A.7 CODICE SEPARAZIONE SINUSOIDI

```

1 load DATI_CARATT_DINAMICA\
    sin_exc_freq_0p5_3_step_0p5_amp1p5_real;
2 % load DATI_CARATT_DINAMICA\sin_exc_freq_4_amp_0p25;
3
4 t = ZRIF_Z_PESO.time;
5 Ts = mean(diff(t));
6 r = ZRIF_Z_PESO.signals(1,1).values(:,2);
7 r = r-r(1);
8 u = ZRIF_Z_PESO.signals(1,1).values(:,1);
9 y = ZRIF_Z_PESO.signals(1,2).values(:,1);
10
11 kv = find(diff(r)~=0)+1;
12 kv1 = [1; find(diff(kv)>10)+1];
13 kv2 = [find(diff(kv)>10); length(kv)];
14 ks = round(5/Ts);
15
16 N = length(kv1);
17
18 filename = 'sin_exc_';
19 for k=1:N,
20     yp = y(kv(kv1(k)+ks:kv2(k)-ks));
21     up = u(kv(kv1(k)+ks:kv2(k)-ks));
22     save(strcat(filename,num2str(k)), 'yp', 'up');
23 end;

```

## A.8 CODICE PER SALVARE AMPIEZZA, FASE E FREQUENZA SINUSOIDI

```
1 %% Get frequency response function (FRF) from experimental
  data
2
3 path_in    = './SIN_EXC/';
4 path_out   = './DATI_RISP_FREQ/';
5 filename   = 'sin_exc_';
6
7 N = 7;
8 fv = [0.5:0.5:3,4];
9
10 FRF = zeros(1,N);
11 FRF_mag_dB = zeros(1,N);
12 FRF_phase_deg = zeros(1,N);
13 for k=1:N,
14     load(strcat(path_in,'sin_exc_',num2str(k)));
15
16     wu = 2*pi*fv(k);
17     get_amp_and_phase;
18
19     FRF_mag_dB(k) = db(A);
20     FRF_phase_deg(k) = phi_deg;
21     FRF(k) = A * exp(j*phi_deg/2/pi);
22 end;
23
24 % unwrap phase
25 FRF_phase_deg = unwrap(FRF_phase_deg*pi/180)*180/pi-180;
26
27
28 %% Show results
29
30 figure;
31 subplot(2,1,1);
32 semilogx(fv, FRF_mag_dB);
33 ylabel('Magnitude [dB]')
34 subplot(2,1,2);
35 semilogx(fv, FRF_phase_deg);
36 ylabel('Phase [deg]')
37 xlabel('Frequency [Hz]')
38
39 %% Save data
40
41 FRF_f = fv;
42 save(strcat(path_out,'FRF_data'), 'FRF', 'FRF_mag_dB', '
    FRF_phase_deg', 'FRF_f');
```

## A.9 CODICE APPROSSIMAZIONE AI MINIMI QUADRATI SINUSOIDI

```

1 %% Load data
2
3 %load 'SIN_EXC/sin_exc_7';
4
5 % remove bias from data
6 up = detrend(up);
7 yp = detrend(yp);
8
9 % define time vector
10 Ts = 0.02;
11 N = length(yp);
12 t = ((0:N-1)*Ts).';
13
14 % define sin input freq
15 %wu = 2*pi*4;
16
17 %% Find LS fitting of input u
18 up = 0.01*up; % [m]
19 Y = up;
20 PHI = [sin(wu*t), cos(wu*t)];
21 th = PHI\Y;
22 Au = norm(th);
23 phiu = atan2(th(2),th(1));
24
25 uh = Au*sin(wu*t+phiu);
26
27
28 %% Find LS fitting of output y
29 yp = 9.81*yp; % [N]
30 Y = yp;
31 PHI = [sin(wu*t), cos(wu*t)];
32 th = PHI\Y;
33 Ay = norm(th);
34 phiy = atan2(th(2),th(1));
35
36 yh = Ay*sin(wu*t+phiy);
37
38 %% Compute amplitude and phase of freq resp (at freq fu=wu
    /2/pi)
39 A = Ay/Au;
40 phi = phiy-phiu;
41 phi_deg = phi*180/pi;
42 f=wu/2/pi;
43
44 return

```

## A.10 CODICE CONFRONTO RISPOSTE IN FREQUENZA

```

1 %% Get experimental FRF
2
3 load './DATI_RISP_FREQ/FRF_data';
4
5
6 %% Get plant nominal tf and FRF
7 mn = 70 + 5;
8 kn = 9.81*51*1e3*3;
9 bn = 2*0.707*sqrt(kn/mn)*mn;
10
11 % Plant nominal tf
12 sysPn = tf([-kn*mn 0 0], [mn bn kn]);
13
14 % Plant FRF
15 f = logspace(log10(0.1), log10(100), 100);
16 [magPn, phaPn] = bode(sysPn, 2*pi*f);
17 magPn = squeeze(magPn);
18 phaPn = squeeze(phaPn);
19
20 %% Plot results
21
22 figure;
23
24 subplot(2,1,1);
25 semilogx(f, db(magPn), 'b');
26 hold on
27 semilogx(f, db(magPi), 'r');
28 semilogx(FRF_f, FRF_mag_dB, 'k.-');
29
30 subplot(2,1,2);
31 semilogx(f, phaPn, 'b');
32 hold on
33 semilogx(f, phaPi, 'r');
34 semilogx(FRF_f, FRF_phase_deg, 'k.-');

```

## A.11 CODICE DI VISUALIZZAZIONE DEL BARICENTRO

```

1 function [sys,x0,str,ts] = sfun_scopexy_2(t,x,u,flag)
2 %SFUNTMPL General M-file S-function template
3 % With M-file S-functions, you can define you own
   ordinary differential
4 % equations (ODEs), discrete system equations, and/or
   just about
5 % any type of algorithm to be used within a Simulink
   block diagram.
6 %
7 % The general form of an M-File S-function syntax is:
8 % [SYS,X0,STR,TS] = SFUNC(T,X,U,FLAG,P1,...,Pn)
9 %

```

```

10 % What is returned by SFUNC at a given point in time, T,
11 % depends on the
12 % value of the FLAG, the current state vector, X, and the
13 % current
14 % input vector, U.
15 %
16 % FLAG      RESULT          DESCRIPTION
17 % -----  -----
18 %
19 % 0          [SIZES,X0,STR,TS] Initialization, return system
20 % sizes in SYS,
21 %                               initial state in X0, state
22 % ordering strings
23 %                               in STR, and sample times in
24 % TS.
25 % 1          DX              Return continuous state
26 % derivatives in SYS.
27 % 2          DS              Update discrete states SYS =
28 % X(n+1)
29 % 3          Y              Return outputs in SYS.
30 % 4          TNEXT          Return next time hit for
31 % variable step sample
32 %                               time in SYS.
33 % 5                               Reserved for future (root
34 % finding).
35 % 9          []            Termination, perform any
36 % cleanup SYS=[].
37 %
38 %
39 % The state vectors, X and X0 consists of continuous
40 % states followed
41 % by discrete states.
42 %
43 % Optional parameters, P1,...,Pn can be provided to the S
44 % -function and
45 % used during any FLAG operation.
46 %
47 % When SFUNC is called with FLAG = 0, the following
48 % information
49 % should be returned:
50 %
51 % SYS(1) = Number of continuous states.
52 % SYS(2) = Number of discrete states.
53 % SYS(3) = Number of outputs.
54 % SYS(4) = Number of inputs.
55 % Any of the first four elements in SYS can
56 % be specified
57 % as -1 indicating that they are dynamically
58 % sized. The
59 % actual length for all other flags will be
60 % equal to the
61 % length of the input, U.
62 % SYS(5) = Reserved for root finding. Must be zero.
63 % SYS(6) = Direct feedthrough flag (1=yes, 0=no). The
64 % s-function
65 % has direct feedthrough if U is used during
66 % the FLAG=3

```

```

48 %           call. Setting this to 0 is akin to making a
           promise that
49 %           U will not be used during FLAG=3. If you
break the promise
50 %           then unpredictable results will occur.
51 %           SYS(7) = Number of sample times. This is the number
of rows in TS.
52 %
53 %
54 %           X0       = Initial state conditions or [] if no
states.
55 %
56 %           STR       = State ordering strings which is generally
specified as [].
57 %
58 %           TS        = An m-by-2 matrix containing the sample time
59 %                       (period, offset) information. Where m =
number of sample
60 %                       times. The ordering of the sample times
must be:
61 %
62 %                       TS = [0         0,         : Continuous sample
time.
63 %                           0         1,         : Continuous, but
fixed in minor step
64 %                               sample time.
65 %                       PERIOD OFFSET, : Discrete sample time
where
66 %                               PERIOD > 0 & OFFSET
< PERIOD.
67 %                       -2         0];         : Variable step
discrete sample time
68 %                               where FLAG=4 is used
to get time of
69 %                               next hit.
70 %
71 %           There can be more than one sample time
providing
72 %           they are ordered such that they are
monotonically
73 %           increasing. Only the needed sample times
should be
74 %           specified in TS. When specifying more than
one
75 %           sample time, you must check for sample hits
explicitly by
76 %           seeing if
77 %           abs(round((T-OFFSET)/PERIOD) - (T-OFFSET
)/PERIOD)
78 %           is within a specified tolerance, generally
1e-8. This
79 %           tolerance is dependent upon your model's
sampling times
80 %           and simulation time.
81 %
82 %           You can also specify that the sample time
of the S-function

```

```

83 %           is inherited from the driving block. For
           functions which
84 %           change during minor steps, this is done by
85 %           specifying SYS(7) = 1 and TS = [-1 0]. For
           functions which
86 %           are held during minor steps, this is done
           by specifying
87 %           SYS(7) = 1 and TS = [-1 1].
88
89 % Copyright 1990–2002 The MathWorks, Inc.
90 % $Revision: 1.18.2.1 $
91
92
93 %
94 % The following outlines the general structure of an S-
           function.
95 %
96 switch flag,
97
98 %%%%%%%%%%%
99 % Initialization %
100 %%%%%%%%%%%
101 case 0,
102     [sys,x0,str,ts]=mdlInitializeSizes;
103
104 %%%%%%%%%%%
105 % Derivatives %
106 %%%%%%%%%%%
107 case 1,
108     sys=mdlDerivatives(t,x,u);
109
110 %%%%%%%%%%%
111 % Update %
112 %%%%%%%%%%%
113 case 2,
114     sys=mdlUpdate(t,x,u);
115
116 %%%%%%%%%%%
117 % Outputs %
118 %%%%%%%%%%%
119 case 3,
120     sys=mdlOutputs(t,x,u);
121
122 %%%%%%%%%%%
123 % GetTimeOfNextVarHit %
124 %%%%%%%%%%%
125 case 4,
126     sys=mdlGetTimeOfNextVarHit(t,x,u);
127
128 %%%%%%%%%%%
129 % Terminate %
130 %%%%%%%%%%%
131 case 9,
132     sys=mdlTerminate(t,x,u);
133
134 %%%%%%%%%%%
135 % Unexpected flags %

```

```

136  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
137  otherwise
138      error(['Unhandled flag = ', num2str(flag)]);
139
140  end
141
142  % end sfuntmpl
143
144  %
145  %=====
146  % mdlInitializeSizes
147  % Return the sizes, initial conditions, and sample times
148  % for the S-function.
149  %=====
150  %
151  function [sys,x0,str,ts]=mdlInitializeSizes
152
153  %
154  % call simsizes for a sizes structure, fill it in and
155  %convert it to a sizes array.
156  %
157  % Note that in this example, the values are hard coded.
158  %This is not a recommended practice as the characteristics
159  %of the block are typically defined by the S-function
160  %parameters.
161  %
162  sizes = simsizes;
163
164  sizes.NumContStates = 0;
165  sizes.NumDiscStates = 4;
166  sizes.NumOutputs = 0;
167  sizes.NumInputs = 7;
168  sizes.DirFeedthrough = 1;
169  sizes.NumSampleTimes = 1; % at least one sample time is
    needed
170
171  sys = simsizes(sizes);
172
173  %
174  % initialize the initial conditions
175  %
176  x0 = [0 0 0 0];
177
178  %
179  % str is always an empty matrix
180  %
181  str = [];
182
183  %
184  % initialize the array of sample times
185  %
186  ts = [0.2 0];
187
188  % Open figure and set axes
189  close all;
190  figure('Renderer','painters')

```

```

191 axes('XLim',[-800 800],'YLim',[-400 800], 'Drawmode','fast'
      );
192
193 a = [-pi/6, pi/2, 7/6*pi];
194 r = 1070/sqrt(3);
195 xv = r.*cos(a);
196 yv = r.*sin(a);
197
198 line([xv, xv(1)], [yv, yv(1)], ...
      'LineStyle', '-', 'LineWidth', 2, 'Color', [0 0 0]);
199
200
201 hold on;
202
203 plot(x0(1), x0(2), ...
      'LineWidth', 4, 'Marker', 's', 'MarkerSize', 20, ...
      'MarkerFaceColor', [1 1 1], ...
      'MarkerEdgeColor', [1 1 1]);
204
205
206
207
208 plot(x0(3), x0(4), ...
      'LineWidth', 4, 'Marker', 's', 'MarkerSize', 20, ...
      'MarkerFaceColor', [1 1 1], ...
      'MarkerEdgeColor', [1 1 1]);
209
210
211
212
213 plot(x0(1), x0(2), ...
      'LineWidth', 4, 'Marker', 'o', 'MarkerSize', 14, ...
      'MarkerFaceColor', [0 0 1], ...
      'MarkerEdgeColor', [0 0 1]);
214
215
216
217
218 plot(x0(3), x0(4), ...
      'LineWidth', 4, 'Marker', '+', 'MarkerSize', 14, ...
      'Color', 'r');
219
220
221
222 axis equal;
223
224 % Set most axes mode properties to manual
225 pn = {'ALimMode',...
      'CLimMode',...
      'TickDirMode','XLimMode',...
      'YLimMode','ZLimMode',...
      'XTickMode','YTickMode',...
      'ZTickMode','XTickLabelMode',...
      'YTickLabelMode','ZTickLabelMode'};
226
227
228
229
230
231
232 pv = cell(1,12);
233 for k = 1:12
234     pv(k) = {'manual'};
235 end
236 set(gca,pn,pv)
237
238 % Remove axis ticks
239 set(gca, 'XTick', []);
240 set(gca, 'YTick', []);
241
242
243 % end mdlInitializeSizes
244
245 %
246 %=====

```

```
247 % mdlDerivatives
248 % Return the derivatives for the continuous states.
249 %=====
250 %
251 function sys=mdlDerivatives(t,x,u)
252
253 sys = [];
254
255 % end mdlDerivatives
256
257 %
258 %=====
259 % mdlUpdate
260 % Handle discrete state updates, sample time hits,
261 % and major time step requirements.
262 %=====
263 %
264 function sys=mdlUpdate(t,x,u)
265
266 sys = [u(1), u(2), ...
267        u(4), u(5) ];
268
269 % end mdlUpdate
270
271 %
272 %=====
273 % mdlOutputs
274 % Return the block outputs.
275 %=====
276 %
277 function sys=mdlOutputs(t,x,u)
278
279 hc = get(gca, 'Children');
280 hr = hc(1);
281 hm = hc(2);
282 hrg = hc(3);
283 hmg = hc(4);
284
285 set(hmg, 'XData', x(1), 'YData', x(2));
286 set(hrg, 'XData', x(3), 'YData', x(4));
287
288 if u(7)==2,
289     set(hm, 'XData', u(1), 'YData', u(2), ...
290           'MarkerFaceColor', [0 0.6 0], ...
291           'MarkerEdgeColor', [0 0.6 0]);
292 else
293     set(hm, 'XData', u(1), 'YData', u(2), ...
294           'MarkerFaceColor', [0 0 1], ...
295           'MarkerEdgeColor', [0 0 1]);
296 end;
297 set(hr, 'XData', u(4), 'YData', u(5));
298
299 drawnow;
300
301 sys = [];
302
303 % end mdlOutputs
```

```

304
305 %
306 %=====
307 % mdlGetTimeOfNextVarHit
308 % Return the time of the next hit for this block.
309 %Note that the result is absolute time.
310 %Note that this function is only used when you specify a
311 %variable discrete-time sample time [-2 0] i
312 %n the sample time array in mdlInitializeSizes.
313 %=====
314 %
315 function sys=mdlGetTimeOfNextVarHit(t,x,u)
316
317 sampleTime = 1;
318 % Example, set the next hit to be one second later.
319 sys = t + sampleTime;
320
321 % end mdlGetTimeOfNextVarHit
322
323
324 %=====
325 % mdlTerminate
326 % Perform any end of simulation tasks.
327 %=====
328 %
329 function sys=mdlTerminate(t,x,u)
330
331 sys = [];
332
333 % end mdlTerminate

```

#### A.12 CODICE DI GENERAZIONE DEL RIFERIMENTO

```

1 function [XYZ_ref, XYZ_cm, state] = FSA(XYZ_meas)
2 % This block supports the Embedded MATLAB subset.
3 % See the help menu for details.
4
5 persistent FSA_state;
6 persistent XYZ_curr_ref collisionX_prev collisionY_prev;
7 persistent timeout_count;
8
9 if isempty(FSA_state)
10     FSA_state = 1;
11     [XYZ_curr_ref] = get_XYZ_ref(1070/2,0);
12     collisionX_prev = 0;
13     collisionY_prev = 0;
14     timeout_count = 0;
15 end;
16
17 XYZ_ref = zeros(3,1);
18 XYZ_cm = zeros(3,1);
19
20 switch FSA_state
21     case 1

```

```
22     XYZ_ref = XYZ_curr_ref;
23     XYZ_cm  = XYZ_meas;
24
25     case 2
26         XYZ_ref = XYZ_curr_ref;
27         XYZ_cm  = XYZ_meas;
28
29     end;
30     state = FSA_state;
31
32     %% Evaluate position errors between measured position and
33     ref target
34     errX = abs(XYZ_meas(1)-XYZ_curr_ref(1));
35     errY = abs(XYZ_meas(2)-XYZ_curr_ref(2));
36
37     %% Define error thresholds for reference target collision
38     detection
39     errX_thrL = 20;   errX_thrH = 40;
40     errY_thrL = 20;   errY_thrH = 40;
41
42     %% Collision detection
43     collisionX = hyst_cmp(errX, errX_thrL, errX_thrH, 1, 0,
44     collisionX_prev);
45     collisionY = hyst_cmp(errY, errY_thrL, errY_thrH, 1, 0,
46     collisionY_prev);
47
48     %% State update
49     switch FSA_state
50     case 1
51         if collisionX && collisionY,
52             timeout_count = 20;
53             FSA_state = 2;
54         else
55             FSA_state = 1;
56         end;
57     case 2
58         if collisionX && collisionY,
59             if timeout_count == 0,
60                 [XYZ_curr_ref] = get_XYZ_ref(1070/2,
61                 XYZ_curr_ref(1:2));
62                 FSA_state = 1;
63             else
64                 timeout_count = timeout_count-1;
65                 FSA_state = 2;
66             end;
67         else
68             FSA_state = 1;
69         end;
70     end;
71
72     %% Update previous vars
73     collisionX_prev = collisionX;
74     collisionY_prev = collisionY;
75
76     end
```

```

73 % .....
74 %
75 function [XYZ] = get_XYZ_ref(lmax, XYcurr)
76     x = Inf; y = Inf;
77     while( norm([x;y]-XYcurr)<200 )
78         l = lmax*rand;
79         h = 0.5*sqrt(3)*l;
80         x = 0.5*l*2*(rand-0.5);
81         y = lmax/sqrt(3)-h;
82     end;
83     XYZ = zeros(3,1);
84     XYZ(1) = x;
85     XYZ(2) = y;
86     XYZ(3) = 0;
87
88 end
89
90
91 % .....
92 %
93 function y = hyst_cmp(u, uL, uH, yL, yH, y_prev)
94     N = length(y_prev);
95     y = zeros(N,1);
96
97     for k = 1:N,
98         if (u(k) > uH) && (y_prev(k) == yL),
99             y(k) = yH;
100        elseif (u(k) < uL) && (y_prev(k) == yH),
101            y(k) = yL;
102        else
103            y(k) = y_prev(k);
104        end;
105    end;
106
107 end

```

## BIBLIOGRAFIA

---

- [1] M. Bisinella. *Studio preliminare di un simulatore per lo sci nautico*.  
Tesi di laurea, Università degli Studi di Trento, 2009/2010.

