

UNIVERSITÀ DEGLI STUDI DI PADOVA

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

TESI DI LAUREA MAGISTRALE

Asservimento visivo in unità di posizionamento pan-tilt-zoom stabilizzate inerzialmente mediante utilizzo di sensori MEMS

Relatore: Ch.mo Prof. Angelo Cenedese

Correlatori: Dr. Riccardo Antonello, Prof. Roberto Oboe

Laureando: Ramello Alessandro
Matricola: 1014513

ANNO ACCADEMICO: 2012-2013

SOMMARIO

Le telecamere utilizzate in applicazioni di videosorveglianza sono sovente collocate su supporti mobili o su veicoli e, in molte circostanze, il movimento del supporto su cui essa poggia provoca una variazione indesiderata del campo di vista. Tale variazione crea numerosi problemi se tale camera viene utilizzata per operazioni di monitoraggio di aree: movimenti troppo bruschi della base possono causare la perdita dell'obiettivo oggetto del controllo per l'uscita di esso dal piano immagine; in tale situazione non si riesce pertanto ad avere informazioni sulla posizione del target.

Il primo obiettivo di questo lavoro di tesi è quello di realizzare un sistema di controllo, basato sulle misure di un sensore di misura inerziale (*Inertial Measurement Unit - IMU*), che permetta di ottenere un'immagine stabile e totalmente disaccoppiata dai movimenti della base. Una volta raggiunto questo obiettivo viene sviluppato un sistema di controllo con asservimento visivo che permetta di effettuare il tracking di un particolare oggetto mediante le informazioni estratte dall'immagine mediante un opportuno algoritmo di elaborazione delle immagini.

Obiettivo non secondario del lavoro svolto è stato quello di dimostrare la fattibilità di questo nuovo sistema di controllo senza dover apportare modifiche strutturali nel brandeggio.

RINGRAZIAMENTI

Ringrazio il Professor Angelo Cenedese e il Professor Roberto Oboe per la loro disponibilità dimostrata durante tutto il periodo di lavoro e per avermi fornito tutto il materiale necessario.

Ringrazio il Dr. Riccardo Antonello e il Dr. Nicholas Felicini che hanno messo a disposizione le loro grandi conoscenze e mi hanno fornito un grande aiuto nella fase sperimentale della tesi.

Ringrazio Roberto Losco e tutto il personale addetto ai laboratori per la grande prontezza dimostrata nel superare gli imprevisti verificatisi nel periodo di tesi.

Ringrazio la mia famiglia che mi ha sempre sostenuto, moralmente ed economicamente, fornendomi la possibilità di studiare ed imparare e mettendomi sempre nelle migliori condizioni per poterlo fare.

Ringrazio Monica per essermi sempre gioiosamente vicino.

Voglio ringraziare calorosamente tutti i miei amici e compagni di studio, che mi hanno dato la forza di non mollare nei momenti di sconforto ed hanno gioito con me dei bei risultati ottenuti.

INDICE

1	INTRODUZIONE	1
1.1	Stabilizzazione per piattaforme inerziali	1
1.2	Videosorveglianza	2
1.2.1	Brandeggio	3
1.2.2	Sorveglianza Marittima	5
1.3	Computer Vision	6
1.3.1	Visual Servoing	7
1.4	Descrizione del problema	9
1.4.1	Struttura della tesi	11
2	SETUP SPERIMENTALE	13
2.1	Introduzione	13
2.2	Piattaforma di Gough-Stewart	14
2.2.1	Cenni Storici	14
2.2.2	Struttura della piattaforma di Stewart	15
2.2.3	Elettrovalvole e circuito pneumatico	17
2.2.4	Scheda Multi Q-3	18
2.2.5	Schede per il condizionamento e gestione dei segnali	19
2.3	Brandeggio Ulisse-Compact	20
2.3.1	Hybryd Step Motor	22
2.3.2	Trasmissione del moto	25
2.4	9DOF-RAZOR IMU, SEN-10736	26
2.5	Scheda di acquisizione sensoray 626	27
2.6	Driver	30
2.7	Connessione PC	31
2.7.1	Introduzione	31
2.7.2	TCP - Transmission Control Protocol	32
2.7.3	UDP - User Datagram Protocol	32
2.7.4	Confronto tra UDP e TCP	33
2.8	Setup Sperimentale completo	34
3	ELABORAZIONE VIDEO	37
3.1	Introduzione	37
3.2	La libreria <i>Intel OpenCV</i>	38
3.3	Immagini digitali e rappresentazione del colore	39
3.4	Features, edge e corner delle immagini	41
3.5	Algoritmo di Canny	43
3.6	Trasformata di Hough	47
3.6.1	Generalità	47
3.6.2	Coordinate cartesiane e coordinate polari	47
3.6.3	La trasformata di hough per il riconoscimento delle linee	49
3.7	Individuazione del target fisso nello scenario	52

3.8	Il filmato digitale	53
3.9	Acquisizione video	54
3.10	Socket per l'invio e la ricezione dei dati	56
3.11	Principali funzioni <i>OpenCv</i> utilizzate	57
4	SISTEMA DI CONTROLLO	61
4.1	Introduzione	61
4.2	Funzioni di trasferimento dei motori	61
4.3	Struttura del controllo	63
4.3.1	Anello di stabilizzazione inerziale	64
4.3.2	Anello di controllo video	67
4.4	Realizzazione del sistema di controllo	69
4.4.1	Sincronizzazione nella comunicazione seriale	69
4.4.2	Impostazione sistema di invio e ricezione dei dati	71
4.5	Schema simulink completo del Sistema di Controllo	72
5	RISULTATI SPERIMENTALI	77
5.1	Introduzione	77
5.2	Test di stabilizzazione dell'immagine	78
5.2.1	Movimentazione della piattaforma con segnale sinusoidale e target posto a 70cm	79
5.2.2	Movimentazione della piattaforma con dati del moto ondosio e target posto a 70cm	84
5.2.3	Movimentazione della piattaforma con dati del moto ondosio, target a 3.6m e fattore di zoom 3x	88
5.3	Test di inseguimento del target	90
	Conclusioni	97
	APPENDIX	99
A	FILE MATLAB,C,C++	101
A.1	Codice C++ necessario per l'identificazione del target fisso.	101
A.2	S-function per leggere dati dalla IMU	106
A.3	File matlab per il design del controllore inerziale	110
	BIBLIOGRAFIA	113

ELENCO DELLE FIGURE

Figura 1	Tipica applicazione di ISPs	1
Figura 2	Esempio di sistema PTZ	4
Figura 3	Notazione angoli aeronautici	4
Figura 4	Tipica boa utilizzata per installare apparecchiature scientifiche	5
Figura 5	Applicazioni di Computer Vision	7
Figura 6	Applicazioni di <i>Image Based Visual Servoing</i>	9
Figura 7	Setup sperimentale completo	13
Figura 8	Piattaforma di Gough per il test di pneumatici	14
Figura 9	Simulatore di volo di Stewart	15
Figura 10	Simulatore Cyber Air Space	16
Figura 11	Particolare circuito pneumatico	17
Figura 12	Schema elettrovalvola	18
Figura 13	Scheda Multi-Q3	19
Figura 14	Schede PCB (printed circuite board) utilizzate.	20
Figura 15	Insieme brandeggio Ulisse Compact e relativa tastiera.	21
Figura 16	Rotore e statore di un motore a passo ibrido.	23
Figura 17	Hybrid step motor	24
Figura 18	Hybrid step motor	25
Figura 19	Cinghia di trasmissione del moto di tilt	26
Figura 20	9DOF-RAZOR IMU, SEN-10736	26
Figura 21	Scheda Sensoray-626	28
Figura 22	Finestra proprietà <i>Analog Output</i>	29
Figura 23	Finestra proprietà <i>Configuration Parameters</i>	29
Figura 24	Finestra proprietà <i>External Mode Control Panel</i>	29
Figura 25	Driver MicroStar-B	30
Figura 26	I 4 driver utilizzati collegati tra loro ed al brandeggio	31
Figura 27	Cavo ethernet utilizzato per connettere i due PC	34
Figura 28	Scenario di ripresa con target fisso costituito dalla croce nera su sfondo bianco	38
Figura 29	Rappresentazione RGB	39
Figura 30	Sistema di riferimento	41
Figura 31	Immagine in cui sono state selezionate tre zone con caratteristiche diverse	42
Figura 32	Fase di soppressione dei non-massimi	46
Figura 33	Sogliatura immagine	47

Figura 34	Rappresentazione in coordinate polari	48
Figura 35	Punti nello spazio immagine e rispettive rette nello spazio dei parametri	49
Figura 36	Trasformazione di un punto nel piano immagine a piano dei parametri in coordinate polari	51
Figura 37	Rappresentazione di una serie di punti allineati con la trasformata di Hough	52
Figura 38	Videocamera interna al brandeggio Ulisse Compact: Sony FCB-EX480CP	54
Figura 39	Framegrabber utilizzato e relativi cavi di interfacciamento con videocamera	55
Figura 40	Fasi di elaborazione dell'immagine.	59
Figura 41	Funzioni di trasferimento dei motori di PAN e di TILT individuate	62
Figura 42	Schema a blocchi del sistema di controllo in generale.	63
Figura 43	Funzioni di trasferimento dei motori (linea blu) e con relativa aggiunta dell'integratore dovuto alla presenza dello slew-rate limiter (linea rossa).	65
Figura 44	Densità spettrale del movimento ondoso analizzato	66
Figura 45	Confronto tra il diagramma di Bode del sistema da controllare (in blu) e il diagramma di Bode della funzione di anello compresa di controllore (in rosso)	67
Figura 46	Andamento della risposta a gradino nel sistema nel caso di controllore tarato con i valori corretti (linea nera) e con i valori sbagliati (linea grigia)	68
Figura 47	Maccchina a stati equivalente per la lettura dei dati inviati dalla IMU	70
Figura 48	Ricezione dei dati trasmessi via UDP	71
Figura 49	Elaborazione ed invio del dato relativo alla misura di roll	72
Figura 50	Schema di controllo complessivo	74
Figura 51	Blocchetti simulink ausiliari al sistema di controllo	75
Figura 52	Frame tipici acquisiti durante i test di stabilizzazione delle immagini. Nella parte sinistra è riportata l'immagine non compensata e in quella destra l'immagine compensata	78

- Figura 53 Angoli di roll,pitch e yaw rilevati tramite la IMU nel caso di stabilizzazione inerziale disattivata e attivata con segnale di riferimento sinusoidale; in linea nera sottile sono riportati i segnali di riferimento forniti alla piattaforma 79
- Figura 54 Andamenti delle coordinate del centro della croce e inclinazione della retta orizzontale nel caso in cui la compensazione inerziale e del movimento di roll siano entrambe attive (linea blu) e nel caso siano entrambe assenti (linea rossa), movimentazione con segnale sinusoidale e target a 70cm 81
- Figura 55 Coordinate del target fisso nel piano X,Y nel caso di compensazione inerziale disattivata e attivata, movimentazione con segnale sinusoidale e target a 70cm 82
- Figura 56 Andamenti delle coordinate del target fisso nel caso sia attivo il solo controllo video (linea rossa) e nel caso sia presente anche l'anello di stabilizzazione inerziale (linea blu), movimentazione con segnale sinusoidale e target a 70cm 83
- Figura 57 Coordinate del centro della croce nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, movimentazione con segnale sinusoidale e target a 70cm. 83
- Figura 58 Angoli di roll,pitch e yaw rilevati tramite la IMU nel caso di stabilizzazione inerziale disattivata e attivata con movimentazione tramite segnale di moto ondoso; in linea nera sottile sono riportati i segnali di riferimento forniti alla piattaforma 84
- Figura 59 Andamenti delle coordinate del centro della croce e inclinazione della retta orizzontale nel caso in cui la compensazione inerziale e del movimento di roll siano entrambe attive (linea blu) e nel caso siano entrambe assenti (linea rossa), movimentazione con segnale di moto ondoso e target a 70cm 85
- Figura 60 Coordinate del centro della croce nel piano X,Y nel caso di stabilizzazione inerziale disattivata e attivata con movimentazione tramite segnale di moto ondoso e target a 70cm 86

Figura 61	Andamenti delle coordinate del centro della croce nel caso di solo controllo video (linea grigia) e nel caso sia presente anche l'anello di stabilizzazione inerziale con movimentazione tramite segnale di moto ondoso, target a 70cm 87
Figura 62	Andamenti delle coordinate del centro della croce nel piano X,Y nel caso sia presente il solo controllo video (parte sinistra) e nel caso sia presente anche l'anello di stabilizzazione inerziale (parte destra) con segnale di moto ondoso, target a 70cm 87
Figura 63	Andamenti delle coordinate del centro della croce e inclinazione della retta orizzontale nel caso in cui la compensazione inerziale e del movimento di roll siano entrambe attive (linea blu) e nel caso siano entrambe assenti (linea rossa), movimentazione con segnale di moto ondoso e presenza di un fattore di zoom 3x 88
Figura 64	Coordinate del centro della croce nel piano X,Y con movimentazione tramite segnale di moto ondoso nel caso di stabilizzazione inerziale disattivata e attivata, zoom 3x e target a 3.6m 89
Figura 65	Andamenti delle coordinate del centro della croce nel caso sia attivo il solo controllo video (linea rossa) e nel caso sia presente anche l'anello di stabilizzazione inerziale (linea blu), movimentazione tramite segnale di moto ondoso, zoom 3x e target a 3.6m 89
Figura 66	Coordinate del centro della croce nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, movimentazione tramite segnale di moto ondoso, zoom 3x e target a 3.6m 90
Figura 67	Frame tipici acquisiti dalla telecamera durante i test di inseguimento del target 91
Figura 68	Andamenti delle coordinate del target mobile nel caso di solo controllo video (linea rossa) e nel caso sia presente anche l'anello di stabilizzazione inerziale (linea blu), movimentazione tramite segnale sinusoidale e target a 70cm. 92

Figura 69	Coordinate del target mobile nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con movimentazione tramite segnale sinusoidale, target a 70cm 92
Figura 70	Andamenti delle coordinate del target mobile nel caso di solo controllo video (linea rossa) e nel caso sia presente anche la stabilizzazione inerziale (linea blu), movimentazione tramite segnale di moto ondoso, target a 70cm 93
Figura 71	Coordinate del target mobile nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche la stabilizzazione inerziale, movimentazione tramite segnale di moto ondoso, target a 70cm 93
Figura 72	Andamenti delle coordinate del target mobile nel caso di solo controllo video (linea rossa) e nel caso sia presente anche l'anello di stabilizzazione inerziale (linea blu) con segnale di riferimento di moto ondoso e zoom 3x 95
Figura 73	Andamenti delle coordinate del target mobile nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con segnale di riferimento di moto ondoso e zoom 3x 95

ELENCO DELLE TABELLE

Tabella 1	Caratteristiche della piattaforma di Stewart utilizzata 17
Tabella 2	Deviazioni standard degli angoli di roll-pitch-yaw nel caso di stabilizzazione inerziale disattivata e attivata con movimentazione con segnale di riferimento sinusoidale 80
Tabella 3	Deviazioni standard dell'inclinazione della retta individuata nel caso in cui la stabilizzazione del movimento di roll sia assente e nel caso essa sia presente, movimentazione con segnale sinusoidale e target posto a 70cm. 81

Tabella 4	Deviazioni standard delle coordinate del centro della croce individuata nel caso di compensazione inerziale disattivata e attivata, movimentazione con segnale sinusoidale e target a 70cm 82
Tabella 5	Deviazioni standard delle coordinate del target fisso nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, movimentazione con segnale sinusoidale e target a 70cm 83
Tabella 6	Deviazioni standard degli angoli di roll-pitch-yaw nel caso di stabilizzazione inerziale disattivata e attivata con movimentazione tramite segnale di moto ondoso 84
Tabella 7	Deviazioni standard dell'inclinazione della retta individuata, movimentazione tramite segnale di moto ondoso e target posto a 70cm 85
Tabella 8	Deviazioni standard delle coordinate X,Y del centro della croce con movimentazione tramite segnale di moto ondoso e target posto a 70cm 86
Tabella 9	Deviazioni standard delle coordinate del centro della croce individuata nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con segnale di riferimento del moto ondoso 87
Tabella 10	Deviazioni standard delle coordinate X,Y del centro della croce con movimentazione tramite segnale di moto ondoso, zoom 3x e target a 3.6m 89
Tabella 11	Deviazioni standard delle coordinate del centro della croce nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con segnale di moto ondoso, zoom 3x e target a 3.6m 90
Tabella 12	Deviazioni standard delle coordinate del target mobile nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, movimentazione tramite segnale sinusoidale, target a 70cm 92
Tabella 13	Deviazioni standard delle coordinate del target mobile nel caso sia presente il solo controllo video e nel caso sia presente anche la stabilizzazione inerziale, movimentazione tramite segnale di moto ondoso, target a 70cm 94

Tabella 14	Deviazioni standard delle coordinate del target mobile nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con segnale di riferimento di moto ondoso e zoom 3x	95
------------	--	----

INTRODUZIONE

1.1 STABILIZZAZIONE PER PIATTAFORME INERZIALI

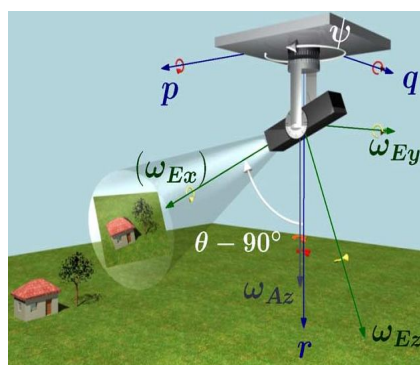
La stabilizzazione per piattaforme inerziali ISPs (*Inertially Stabilized Platforms*) viene utilizzata per stabilizzare e/o mantenere indirizzati una vasta gamma di sensori come fotocamere, videocamere, telescopi, equipaggiamenti di armamento, ecc... Tali sistemi hanno in comune la caratteristica di poter esser montati su qualsiasi tipo di veicolo mobile come satelliti, sottomarini, aerei, autoveicoli e navi. Le applicazioni in cui questi dispositivi vengono utilizzati includono la videosorveglianza, l'inseguimento di target, la stabilizzazione di telecamere e il controllo di sistemi di puntamento.

I sistemi di cui stiamo parlando sono di solito denominati *gimbal*. Essi sono costituiti da un insieme di strutture, ingranaggi e attuatori in cui spesso sono installati molti tipi di sensori, quali giroscopi, accelerometri, magnetometri ed unità GPS. Grazie alle loro caratteristiche costruttive questi sistemi permettono la stabilizzazione del campo visivo (LOS- *Line of Sight*) del carico posto su di essi, ruotandolo e controllandolo opportunamente. Attraverso la lettura dei sensori installati nel gimbal si può misurare l'orientamento e la posizione della struttura. Basandosi successivamente su queste misure, è possibile realizzare qualsiasi controllo di stabilizzazione o di puntamento.

Il gimbal è progettato, tipicamente, per permettere il puntamento e la stabilizzazione rispetto due o più assi e molte applicazioni richiedono almeno due gimbal ortogonali tra loro, per ottenere alcuni gradi di libertà aggiuntivi o per ottenere un'immagine finale totalmente disaccoppiata dai movimenti della base del sistema.



(a) Elicottero sulla quale è collocata una telecamera con gimbal



(b) Scenario tipico per le stabilizzazione inerziale del LOS[2].

Figura 1: Tipica applicazione di ISPs

L'obiettivo comune dei vari sistemi ISPs presenti letteratura è quello di effettuare la stabilizzazione o il controllo del LOS di un oggetto rispetto a un altro oggetto oppure rispetto ad uno spazio inerziale.

Un'applicazione tipica è riportata in Figura 1; essa ci permette di capire l'effettiva importanza di queste applicazioni. In particolare in Figura 1b, in blu è rappresentato un sistema di riferimento (SdR) solidale alla base per descriverne i movimenti, in questo caso corrispondenti ai movimenti dell'elicottero, mentre in verde è rappresentato un SdR solidale ai 3 assi del gimbal. La possibilità di ruotare in modo opportuno il SdR del gimbal rispetto a quello della base permette di ottenere un'immagine caratterizzata dall'avere l'oggetto da inseguire (in questo caso la casa) posizionato sempre al centro dell'immagine stessa.

Più in generale si può dire che il movimento che deve essere compensato è sempre il movimento tra due corpi, che spesso sono in movimento uno rispetto all'altro. Si cerca quindi di ridurre il più possibile l'errore di inseguimento del target ossia l'errore che si commette cercando di mantenere puntato il LOS su un target. Si vuole inoltre minimizzare il jitter, ossia l'oscillazione indesiderata del sensore che può compromettere la qualità dei dati raccolti da quest'ultimo. L'oscillazione può essere causata da vibrazioni generate dal mezzo in cui sono montati i sensori o da disturbi ambientali quali terreni sconnessi, vento, onde e quant'altro.

1.2 VIDEOSORVEGLIANZA

Uno sistema di sorveglianza può esser definito come uno strumento tecnologico che amplia le potenzialità dell'essere umano fornendo ad esso sia una percezione estesa dell'ambiente che lo circonda sia la capacità di analizzare gli eventi che si manifestano nell'ambiente stesso. La percezione umana è infatti limitata, così come la capacità di analizzare gli eventi che si verificano, poichè l'essere umano, per sua stessa natura, possiede la capacità di memorizzare ed analizzare contemporaneamente una serie limitata di dati. Si pensi ad esempio come l'uomo, data la limitatezza dello spazio visivo di cui dispone, sia in grado di analizzare dati provenienti solamente da un'area limitata dello spazio che lo circonda[1].

Con il termine videosorveglianza si intendono tutte quelle applicazioni in cui l'attività di vigilanza viene svolta attraverso l'utilizzo di telecamere o di altri strumenti in grado di assicurare la trasmissione di immagini strategicamente posizionate[13].

I campi di applicazione della videosorveglianza sono numerosi, tra cui:

- Applicazioni per garantire la sicurezza nel campo dei trasporti, come il monitoraggio nelle stazioni ferroviarie o metropolitane, negli aeroporti, nelle autostrade e nelle strade urbane;

- Applicazioni nel campo marittimo per sorveglianza di zone costiere e aree portuali;
- Applicazioni riguardanti la sicurezza o il controllo di qualità nel campo industriale, il monitoraggio di impianti nucleari o di cicli di trasformazione industriale;
- Applicazioni nel campo militare che consentono la sorveglianza aerea e di edifici strategici.

Gli ambiti e le tecniche di videosorveglianza possono essere organizzate in 3 categorie:

1. Selezione di un particolare target ed il suo inseguimento nei vari ambienti possibili;
2. Analisi del movimento di oggetti e di persone ed identificazione del moto, cioè tecniche che consentono di descrivere accuratamente l'andatura e la postura di una persona;
3. Analisi attiva. Questa é l'area più importante per la ricerca futura. Si vuole fare in modo che il sistema di videosorveglianza sia in grado di rivelare e classificare autonomamente gli eventi e le interazioni tra questi.

L'utilizzo di algoritmi di tracking automatici basati su sistemi multiscala consente un'analisi approfondita e accurata dell'ambiente da monitorare senza la necessaria presenza dell'operatore umano. Tali sistemi multiscala sono generalmente costituiti da un insieme di telecamere, di cui alcune fisse e altre mobili, che forniscono informazioni destinate ad essere analizzate da particolari algoritmi di elaborazione delle immagini al fine di effettuare il tracking corretto di un particolare oggetto presente sulla scena[3].

Gli sviluppi futuri nel campo della videosorveglianza sono principalmente volte ad effettuare la fusione di tecniche di tracking 2-D e 3-D, elaborazione di modelli tridimensionali di persone e veicoli, rilevazione di anomalie e predizione di comportamenti, riconoscimento semantico e sorveglianza da remoto.

1.2.1 Brandeggio

Per brandeggio, in senso generico, s'intende un supporto che può ruotare contemporaneamente sia lungo un asse verticale, sia lungo un asse orizzontale. Esso si lega al concetto di videosorveglianza quando si ha l'esigenza di muovere a distanza una telecamera in tutte le orientazioni possibili. Con il termine brandeggio per videosorveglianza, in campo industriale, s'intende l'insieme di motori, custodie e il sistema di controllo integrato. Quest'ultimo è necessario per i movimenti di pan e tilt, e per la gestione delle ottiche motorizzate (zoom,

focus, iris), della telemetria e del segnale video.

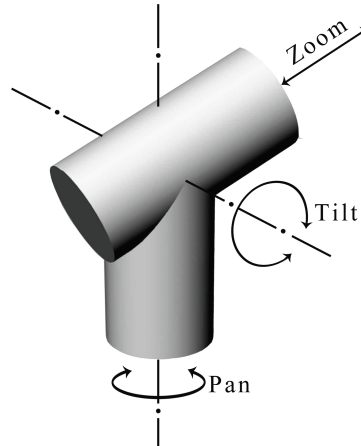


Figura 2: Esempio di sistema PTZ

Questi sistemi prendono anche il nome di telecamere PTZ (vedi Figura 2), ossia telecamere in grado di eseguire movimentazioni di pan (orizzontale), tilt (verticale) e zoom; la corretta movimentazione degli assi delle camera consente di compensare eventuali movimenti della base e di inseguire opportuni obiettivi.

I movimenti della base si esprimono spesso nei seguenti angoli aeronautici:

- *Roll*: rotazione attorno all'asse longitudinale, detto anche *asse di fusoliera*;
- *Pitch*: rotazione attorno all'asse laterale, detto anche *asse dell'ala*;
- *Yaw*: rotazione attorno all'asse verticale.

Tali angoli aeronautici vengono espressi chiaramente in Figura 3

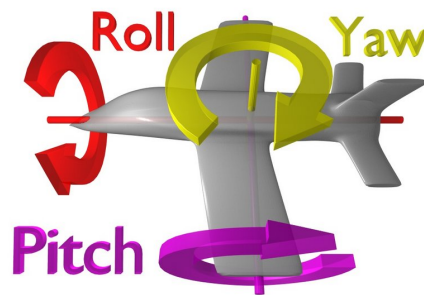


Figura 3: Notazione angoli aeronautici

1.2.2 Sorveglianza Marittima

Mentre la sorveglianza terrestre e aerea da sempre ha attirato un notevole interesse e un conseguente notevole sviluppo, in ambito marittimo questo tema non ha richiamato un'attenzione significativa. Tuttavia, in questo ambito vi sono molti interessanti aspetti e problematiche riguardanti la stabilizzazione del moto. Le principali applicazioni interessate sono boe per videosorveglianza in zone portuali e aree marine di interesse, boe con apparecchiature scientifiche a bordo, videosorveglianza per navi etc...

Questi sensori galleggianti collocati tipicamente su boe o navi, sono influenzati da un continuo movimento dovuto alla superficie dell'acqua. Le onde possono avere forme e dimensioni che variano anche molto rapidamente nel tempo. È quindi necessario in alcuni casi montare tali sensori sopra a delle strutture ISP al fine di compensare il moto ondoso rendendo così possibile la sorveglianza di opportune zone costiere e marine. Nel caso gli strumenti utilizzati per effettuare la sorveglianza siano costituiti da sensori ottici o da telecamere la presenza di un movimento alla base non compensato potrebbe portare all'errata individuazione dei target richiesti.

Si capisce subito dunque quanto sia necessario sviluppare un opportuno sistema di controllo che permetta di raggiungere gli obiettivi preposti.



Figura 4: Tipica boa utilizzata per installare apparecchiature scientifiche

1.3 COMPUTER VISION

La Visione Artificiale (Visione Computazionale o Computer Vision-CV) si definisce come la disciplina che studia i modelli e metodi per abilitare le macchine alla comprensione e interpretazione delle informazioni visuali presenti in immagini fisse bidimensionali o in sequenze video. Lo scopo principale della visione artificiale è quello di emulare la visione biologica, di natura tridimensionale, non solo come l'acquisizione di una fotografia bidimensionale di un'area ma soprattutto come l'interpretazione del contenuto di questa riproducendo sui calcolatori elettronici il percorso cognitivo compiuto dall'essere umano nell'interpretazione della realtà che lo circonda attraverso le immagini che esso percepisce per mezzo degli occhi.

Un sistema di visione artificiale è costituito dall'integrazione di componenti ottiche, elettroniche e meccaniche che permettono di acquisire, registrare ed elaborare immagini sia nello spettro della luce visibile che al di fuori di essa (infrarosso, ultravioletto, raggi X, ecc.). Il risultato dell'elaborazione è il riconoscimento di determinate caratteristiche dell'immagine per varie finalità di controllo, classificazione, selezione, ecc...

Un sistema di visione artificiale è costituito da diverse componenti:

- telecamere e ottiche;
- sistema di illuminazione;
- l'oggetto da esaminare;
- il sistema di acquisizione e di elaborazione dell'immagine;
- l'interfaccia uomo macchina;
- le interfacce con l'ambiente esterno;

L'immagine catturata dalla telecamera viene elaborata per determinarne le caratteristiche di interesse. Una volta che tali caratteristiche vengono determinate, vengono confrontate con caratteristiche di riferimento e vengono prese delle decisioni opportune per la funzionalità del sistema.

Un problema classico nella visione artificiale è quello di determinare se l'immagine contenga o no determinati oggetti (*Object recognition*). Il problema può essere risolto efficacemente e senza difficoltà per oggetti specifici in situazioni specifiche per esempio il riconoscimento di oggetti geometrici come poliedri, il riconoscimento di volti o caratteri scritti a mano. Le cose si complicano nel caso di oggetti arbitrari in situazioni arbitrarie.

Grazie alle funzionalità e alla elevata potenza di calcolo i sistemi di visione possono trovare campi di applicazione virtualmente illimitati. In particolare fra le applicazioni in campo industriale figurano:

- monitoraggio di nastri in continuo;
- Lettura di caratteri e codici;
- Misure non a contatto;
- Riconoscimento difetti e rispetto delle tolleranze.

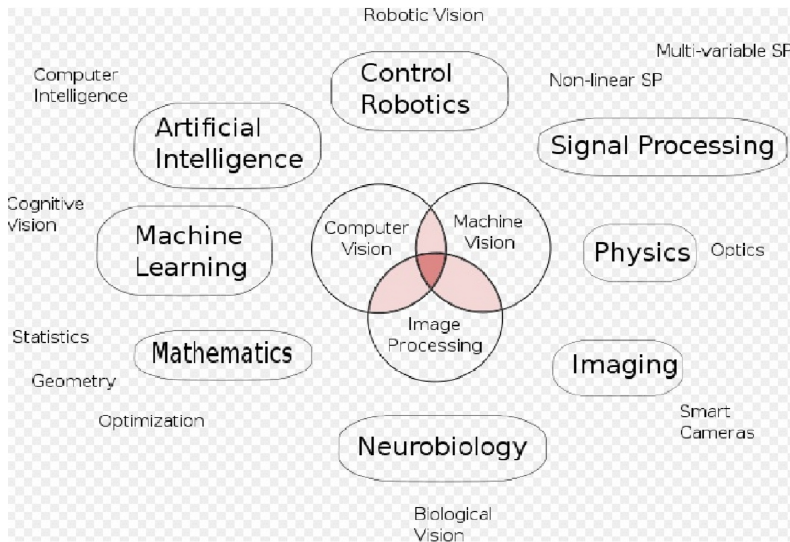


Figura 5: Applicazioni di Computer Vision

Come si può notare dalla Figura 5, la *Computer Vision* in unione con la *Machine Vision* e l'*Image Processing* viene utilizzata in altre notevoli applicazioni al di fuori del campo industriale. Una di queste applicazioni è la videosorveglianza in cui il riconoscimento di particolari oggetti o soggetti all'interno del campo visivo può causare situazioni di allarme più o meno pericolose.

1.3.1 Visual Servoing

Il Visual Servoing, abbreviato con VS, anche conosciuto come asserimento visivo basato sull'immagine, è una tecnica di controllo in retroazione che usa come feedback l'informazione estratta da sensori visivi per controllare il movimento di un robot. Per effettuare tale controllo vengono estratte delle informazione visive riguardanti alcune caratteristiche dell'oggetto o della sua dinamica. Le informazioni visive possono essere acquisite tramite una videocamera che può essere collocata direttamente sull'oggetto oppure in una zona circostante ad esso.

Lo scopo del visual servoing è quello di minimizzare l'errore tra la posizione di riferimento e quella attuale di un oggetto presente nello scenario; esprimendo tale concetto attraverso una formula matematica si ottiene:

$$e(t) = s(m(t), a) - s^* \quad (1)$$

dove i parametri indicati hanno il presente significato:

- $\mathbf{m}(t)$ è un vettore comprendente un insieme di dati provenienti da un algoritmo che elabora l'immagine in tempo reale per estrarne le caratteristiche principali, es. le coordinate di un punto d'interesse o le coordinate del centro di un oggetto;
- \mathbf{a} è un vettore contenente una serie di parametri che approfondiscono la conoscenza del sistema;
- $\mathbf{s}(\mathbf{m}(t), \mathbf{a})$ vettore costruito a partire dalle misure rilevate che fornisce la descrizione matematica dei punti oggetto di interesse al momento attuale;
- \mathbf{s}^* valore di riferimento del vettore $\mathbf{s}(t)$;

L'obiettivo dunque dell'asservimento visivo basato sull'immagine è quello di cercare di minimizzare tale errore $e(t)$ attraverso un'opportuna legge di controllo che si basi su informazioni estratte dall'immagine in tempo reale tramite sensori ottici e videocamere [9].

In generale le tecniche di Visual Servoing sono classificate secondo le seguenti tipologie:

- Image Based (IBVS), basato sull'immagine,
- Position Based (PBVS), basato sulla posizione
- Hybrid Approach, un approccio ibrido.

Nella tecnica IBVS, la legge di controllo è basata sull'errore calcolato sulla base della differenza tra una determinata caratteristica sul piano immagine attuale e una posizione desiderata sempre sul piano immagine della medesima caratteristica. Questa tecnica non coinvolge in nessun modo la stima della posizione dell'oggetto nello spazio o in un qualche sistema di riferimento che non sia l'immagine stessa. Le caratteristiche in esame possono essere linee, punti, momenti geometrici, ecc...

Il PBVS è una tecnica (ad una telecamera) basata su un modello. Infatti, la posizione dell'oggetto viene stimata rispetto alla telecamera. E' importante notare la differenza con la tecnica precedente. Mentre prima si utilizzava solo l'errore sul piano immagine e non veniva effettuato nessun tipo di calcolo di stima della posizione, in questo caso si ha una ricostruzione della posizione dell'oggetto nello spazio 3D (in un piano cartesiano desiderato) e con questa posizione viene impartito il comando al controllore che fa muovere il robot. Per questo motivo questa tecnica viene anche chiamata visual servoing 3D.

Gli approcci ibridi mettono insieme aspetti del IBVS e PBVS.

In Figura 6 è rappresentato un tipico schema di applicazione di IBVS nel campo industriale.

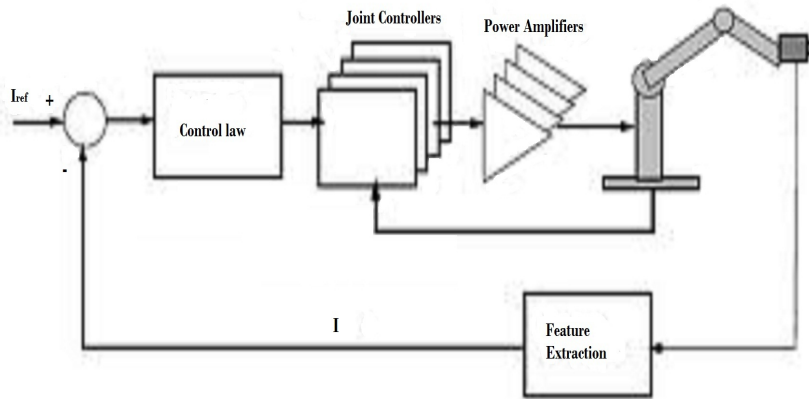


Figura 6: Applicazioni di *Image Based Visual Servoing*

1.4 DESCRIZIONE DEL PROBLEMA

In applicazioni di videosorveglianza e di monitoraggio di aree, è spesso richiesto di poter aggiustare la linea di vista (*line of sight -LOS*) per eseguire alcune attività tipiche come il pattugliamento e il monitoraggio di alcune aree ritenute di particolare interesse. Questa regolazione viene spesso eseguita tramite delle telecamere del tipo pan-tilt-zoom (PTZ), le quali consentono una rotazione della telecamera attorno a due assi perpendicolari al proprio asse ottico [10].

In molti casi questi dispositivi sono collocati sopra pali molto alti (che dunque si piegano per effetto del vento), boe galleggianti, o sui veicoli in movimento come camion, auto, elicotteri ed aerei. A causa del moto ondoso, la boa o la nave su cui il brandeggio è fissato può oscillare e tali movimenti influiscono sulla stabilità del campo visivo della camera, rendendo così difficoltoso il monitoraggio di obiettivi specifici collocati sul piano immagine. Per sopperire a questi movimenti indesiderati dei supporti vengono in soccorso le tecniche di stabilizzazione delle immagini (*optical image stabilization -OIS*).

Al giorno d'oggi molte telecamere commerciali sono già dotate di un proprio sistema di stabilizzazione delle immagini e tale sistema può essere realizzato sia via hardware che via software: nel primo caso le lenti della fotocamera vengono spostate secondo le misure effettuate da un sensore di misura inerziale collocato sul telaio della macchina, mentre nel secondo caso la compensazione è eseguita in tempo reale da un algoritmo che elabora le differenze tra due frame successivi ed elimina i movimenti residui rilevati, essendo questi dovuti ai movimenti indesiderati dei supporti, almeno in uno scenario in cui tutti gli oggetti presenti siano fissi. Purtroppo le telecamere dotate di questo tipo di sistemi sono molto costose per le tradizionali applicazioni

di videosorveglianza, dove in genere sono richieste unità a basso costo ed in grado di operare in ambienti difficili.

In tali circostanze un modo conveniente di operare consiste nello stabilizzare l'immagine rispetto ad un sistema di riferimento inerziale fisso, movimentando opportunamente i motori dell'unità di posizionamento per contrastare i movimenti indesiderati. Operando secondo questo concetto, il brandeggio in questione può essere considerato come un sistema di stabilizzazione inerziale (*ISP-Inertially Stabilized Platform*), in cui cioè esso è in grado di mantenere costante la posizione e l'orientamento di una sua parte caratteristica (ad es. la videocamera) rispetto ad un sistema di riferimento fisso. Ovviamente nei dispositivi di tipo PTZ è possibile effettuare solamente una parziale compensazione dei movimenti attraverso un'opportuna movimentazione dei motori di PAN e di TILT che consente di mantenere stabilizzato il campo visivo della camera. Tuttavia si può pensare ad una stabilizzazione totale dell'immagine, anche lungo i rimanenti gradi di libertà del meccanismo, cioè si possono compensare le rotazioni attorno all'asse ottico e i movimenti lungo le 3 direzioni ortogonali nel SdR inerziale. Le informazioni necessarie per effettuare la stabilizzazione possono essere fornite da una serie di giroscopi, i quali forniscono in uscita le velocità angolari relative ai 3 assi e da cui è possibile ricavare le varie posizioni angolari, oppure da un sensore di misura inerziale (Inertial Measurement Unit - IMU) che fornisce direttamente una stima della posizione rispetto al SdR inerziale.

Recenti sviluppi nel campo dei sensori MEMS (Micro Electro Mechanical Systems) hanno portato alla realizzazione di piccoli ed economici sensori inerziali dotati di accelerometri, giroscopi e magnetometri. Essendo questo tipo di sensori più piccoli di quelli tradizionali, essi possono essere facilmente collocabili sui sistemi come quelli oggetto di questo studio e permettono la realizzazione di sistemi di stabilizzazione inerziale in modo conveniente ed economico.

Lo scopo del presente lavoro di tesi è quello di realizzare la stabilizzazione totale dell'immagine secondo i concetti appena espressi in un'unità di posizionamento PTZ a basso costo, dotata di due motori passo-passo ibridi (Hybrid Step Motors-HSMs), e cercare di individuare i benefici che si possono ottenere attraverso l'utilizzo di tale attrezzatura nel campo della stabilizzazione dell'immagine e dell'inseguimento di un opportuno target. Il sensore di misura inerziale, IMU, è direttamente attaccato alla telecamera per registrarne i movimenti ed è montato in modo tale che i suoi assi, denominati assi di yaw, pitch e roll, coincidano con gli assi di pan, tilt e zoom della telecamera. Tramite questa disposizione le misure di imbardata (yaw) e beccheggio (pitch) vengono utilizzate per effettuare la stabilizzazione agendo sui motori del brandeggio mentre la misura della rotazione intorno all'asse ottico (roll) è utilizzata per compensare quest'ultimo movimento producendo una contro rotazione del piano immagine.

Il lavoro presentato è il naturale proseguimento del lavoro svolto dai due tesisti precedenti, Davide Pilastro[11] e Simone Viola [17], i quali hanno impostato il setup sperimentale necessario per realizzare un primo sistema di controllo che permetta di stabilizzare l'immagine lungo le direzioni di yaw e pitch. Il proseguimento è consistito nel raggiungere i seguenti obiettivi:

1. stabilizzazione totale delle immagini compensando, in real-time, anche il movimento di roll che produceva rotazioni indesiderate dell'immagine attorno all'asse ottico della camera;
2. progetto di un sistema di controllo basato sull'asservimento visivo che consenta di inseguire un target mobile che si muove ad una velocità e distanza arbitraria dalla camera.

1.4.1 *Struttura della tesi*

Dopo questa introduzione di carattere generale l'elaborato si sviluppa in 4 capitoli. Nel primo di questi viene presentato il setup sperimentale utilizzato illustrando accuratamente tutti i componenti del sistema. Nel secondo vengono presentate le nozioni di base nell'ambito dell'elaborazione delle immagini e vengono forniti gli strumenti necessari per estrapolare dalle immagini le nozioni utili ai fini della nostra applicazione. Il terzo capitolo illustra il sistema di controllo sviluppato spiegandone il funzionamento. Nell'ultimo capitolo vengono infine presentate le prove sperimentali effettuate e vengono illustrati approfonditamente i risultati ottenuti al fine di valutare i vantaggi e gli svantaggi del sistema di controllo progettato.

SETUP SPERIMENTALE

2.1 INTRODUZIONE

Per lo svolgimento di questo elaborato si è reso necessario allestire e successivamente utilizzare un'apparato sperimentale abbastanza complesso che si sviluppa principalmente in 3 sottosistemi. Il primo è la piattaforma di Gough-Stewart che costituisce il vero e proprio banco di prova mediante il quale viene simulato il movimento ondoso a cui può essere sottoposta una telecamera posizionata su una boa collocata in un punto qualsiasi in mare. Il secondo sottosistema, invece, è costituito dal brandeggio Ulisse Compact, il cui comportamento è posto sotto attenta osservazione. A contorno di questi due dispositivi principali sono stati utilizzati una IMU board (*Inertial Measurement Unit*), schede di potenza e schede di acquisizione dati essenziali per l'implementazione dei vari loop di controllo e per il monitoraggio del sistema nel suo insieme. L'insieme piattaforma di Stewart più brandeggio è illustrato in Figura 7.

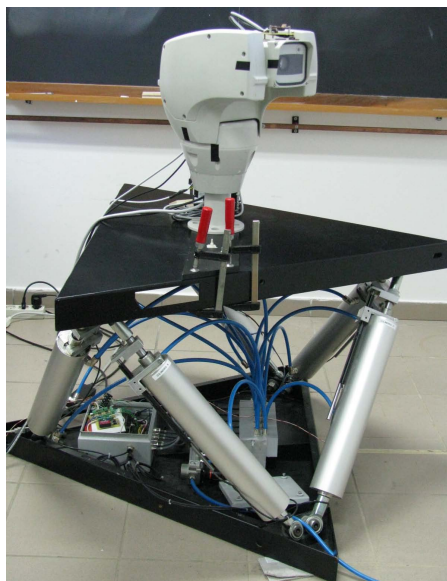


Figura 7: Setup sperimentale completo

Il terzo sottosistema è costituito dal PC utilizzato per effettuare l'elaborazione in real time dell'immagine che viene catturata tramite un opportuno frame-grabber che cattura i vari frame provenienti dalla telecamera presente all'interno del brandeggio.

Il setup sperimentale completo risulta dunque così composto:

- Piattaforma di Gough-Stewart;
- Brandeggio Ulisse Compact;
- Driver di potenza per il comando dei motore;
- Scheda di acquisizione PCI Sensoray 626;
- Due PC con software Matlab/Simulink completo di toolbox per simulazioni in Real-Time;
- Un PC per l'elaborazione in real-time delle immagini, la loro visualizzazione e la registrazione del video con software VLC.
- Framegrabber per catturare i frame inviati dalla camera presente all'interno del brandeggio;
- 9DOF-RAZOR IMU.

2.2 PIATTAFORMA DI GOUGH-STEWART

2.2.1 *Cenni Storici*

I primi studi sulla piattaforma di Gough-Stewart (o semplicemente piattaforma di Stewart) risalgono al 1813 quando il matematico francese Cauchy era interessato al possibile moto di un ottaedro articolato; tale macchina fu poi costruita nel 1947 da Gough, un ingegnere aeronautico, che la utilizzò per la movimentazione di pneumatici durante l'esecuzione di test dinamici finalizzati allo studio dell'usura, vedi Figura 8.

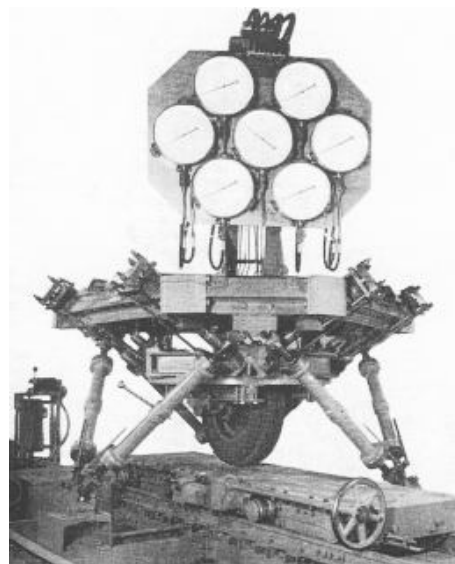


Figura 8: Piattaforma di Gough per il test di pneumatici

Nel frattempo, però, Pollard, un ingegnere automobilistico, brevettò nel 1942 un dispositivo di verniciatura caratterizzato da una struttura cinematica chiusa: questo meccanismo, chiamato tripode perché costituito da 3 arti, in realtà non fu mai costruito ma viene normalmente considerato la prima proposta di un robot parallelo ed il suo concetto è stato sviluppato in molte realizzazioni moderne.

Nel 1965 fu pubblicato un importante articolo da Stewart, che propose l'adattamento dell'esapode di Gough come simulatore di volo (vedi Figura 9): già in questo momento apparvero chiare al mondo scientifico le grandi potenzialità di questa nuova tipologia di meccanismi, tanto che vennero subito chiamati "piattaforme di Stewart" dal nome del suo ultimo proponente, anche se la loro struttura cinematica è spesso più simile allo schema originale di Gough.

L'effettivo sviluppo della tecnologia fu però frenato dalla grande complessità delle relazioni cinematiche, che rendevano difficoltoso lo studio dei meccanismi e quasi impossibile la realizzazione di un efficiente sistema di controllo per la loro movimentazione.

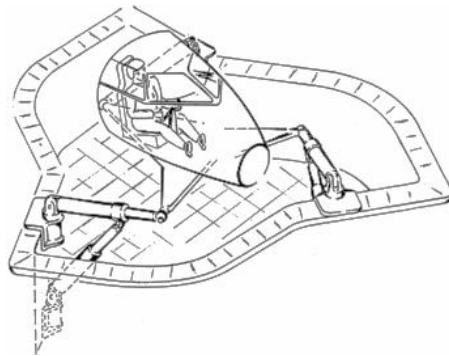


Figura 9: Simulatore di volo di Stewart

2.2.2 Struttura della piattaforma di Stewart

Allo stato attuale, nella sua configurazione classica la piattaforma di Stewart è costituita da un dispositivo terminale collegato al telaio fisso tramite un sistema di attuazione parallelo, in cui gli attuatori lineari che costituiscono gli arti o gambe della piattaforma sono disposti in maniera tale da avere a due a due un punto in comune alternativamente con la base fissa e l'endeffector: il collegamento avviene mediante 6 giunti sferici identici sulla base e 6 giunti sferici identici sulla piattaforma mobile. Lo scopo dell'utilizzo della piattaforma di Stewart all'interno di questo lavoro è di realizzare un simulatore di moto ondoso ed è costituito dal modello Cyber Air Space della Virtogo, il quale viene venduto come simulatore nel campo dei videogiochi.

Tale manipolatore parallelo, come si può notare in Figura 10 è costituito da:



Figura 10: Simulatore Cyber Air Space

- 2 basi, una fissa appoggiata al pavimento e una mobile;
- 6 attuatori lineari che collegano la base fissa a quella mobile mediante dei giunti rotoïdali. Ogni attuatore è costituito da un pistone pneumatico a doppio effetto e da un potenziometro a resistenza variabile che permette di ottenere misure sull'allungamento dei singoli cilindri;
- 6 elettrovalvole proporzionali, una per ogni attuatore, che regolano il flusso dell'aria compressa nelle due camere, inferiore e superiore, dei pistoni pneumatici;
- scheda di acquisizione dati che gestisce i segnali input-output tra la piattaforma di Gough-Stewart e il PC;
- scheda per il condizionamento dei segnali che gestisce l'apertura delle 6 valvole;
- scheda di adattamento della tensione per alimentare correttamente i potenziometri e la scheda di condizionamento del segnale

I vantaggi principali nell'utilizzare un manipolatore a 6 DOF (Degree Of Freedom) come quello realizzato in questa sede sono:

1. Elevato rapporto tra carico utile e peso della struttura;
2. Buone prestazioni dinamiche ed elevata rigidità della struttura poichè il carico viene distribuito su tutte e 6 le sue gambe.

Lo svantaggio di questo tipo di manipolatore parallelo rispetto ad un normale manipolatore seriale è la limitatezza dello spazio di lavoro e la presenza di condizioni singolari. In ultimo non bisogna dimenticare la complessità dell'analisi cinematica sia diretta che inversa tipica di questo tipo di manipolatori ed inoltre il non perfetto posizionamento della parte superiore della struttura a causa della presenza di

attuatori pneumatici. Nella Tabella 1 sono riportate le caratteristiche principali del manipolatore utilizzato.

Model number- Cyber Air Space	2.1
DOF	6
Pitch[deg]	25
Yaw[deg]	9
Roll[deg]	25
Alzata[m]	0.88
Max Speed[m/s]	114
Corsa [m]	0.2
Massimo carico applicabile [Kg]	300
Peso a vuoto, compresa poltrona [Kg]	90
Altezza [m]	0.68
Consumo aria (m ³ a 100 PSI)	12.4

Tabella 1: Caratteristiche della piattaforma di Stewart utilizzata

2.2.3 Elettrovalvole e circuito pneumatico

I cilindri pneumatici presenti all'interno del manipolatore sono alimentati da un circuito che preleva l'aria compressa dalla linea di distribuzione presente in laboratorio. A monte di tale circuito è installata una valvola di regolazione che permette il controllo della pressione in ingresso mentre subito a valle del circuito è presente un collettore che distribuisce l'aria a tutte e 6 le elettrovalvole come illustrato in Figura 11

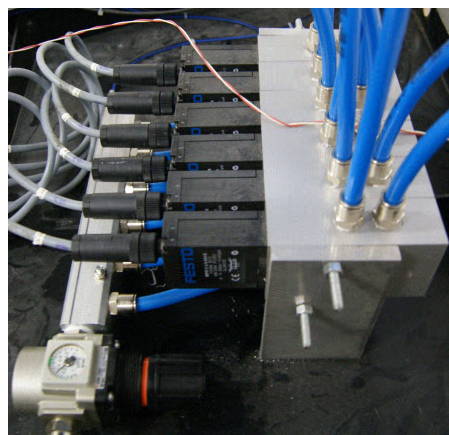


Figura 11: Particolare circuito pneumatico

Le valvole vengono controllate da un segnale elettrico di tensione che

varia da 0 a 10 V. Tale segnale viene utilizzato come riferimento dall'elettronica di controllo di un solenoide, il quale genera un campo magnetico che condiziona la posizione di una slitta presente all'interno delle elettrovalvole. A seconda della posizione della slitta varierà in modo corrispondente il flusso d'aria che passa attraverso le due camere della valvola. In particolare il flusso d'aria varia in modo continuo e proporzionale al segnale di controllo applicato. La condizione di riposo della valvola, cioè quella in cui entrambe le uscite sono chiuse si verifica quando la tensione di ingresso è pari a 5 V. Quando il riferimento varia da tale condizione di equilibrio o verso gli 0 V o verso i 10 V inizia a fluire aria in una delle due uscite. In questa maniera viene controllato il flusso d'aria convogliato in una delle due camere del pistone e di conseguenza la sua velocità di estensione o di accorciamento.

Le elettrovalvole, il cui schema è riportato in Figura 12, devono essere alimentate con una tensione compresa tra 17 e 30 V in DC. Tale alimentazione viene fornita attraverso un alimentatore ECOLINE CP SNT 250 W che fornisce in uscita una tensione di 24 V in DC e un massimo di 10 A. Tale alimentatore è parecchio sovradimensionato per questa applicazione, ed è per questo che, in seguito, viene utilizzato anche come sorgente per le schede di regolazione della tensione e per i driver dei motori.

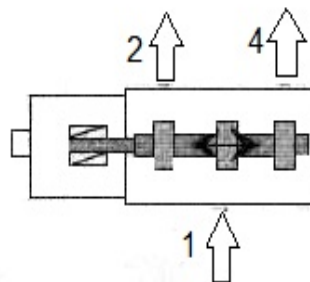


Figura 12: Schema elettrovalvola

2.2.4 Scheda Multi Q-3

Il dispositivo MultiQ-3 è una scheda di acquisizione e controllo dei dati multiuso ed in questa applicazione viene utilizzata per gestire l'interfacciamento tra calcolatore e piattaforma di Stewart (Figura 13). La scheda è dotata di:

- 8 ingressi analogici single-ended
- 8 uscite analogiche
- 16 bit di ingresso digitali

- 16 bit di uscita digitali
- 4 timer programmabili
- 8 ingressi encoder decodificati in quadratura

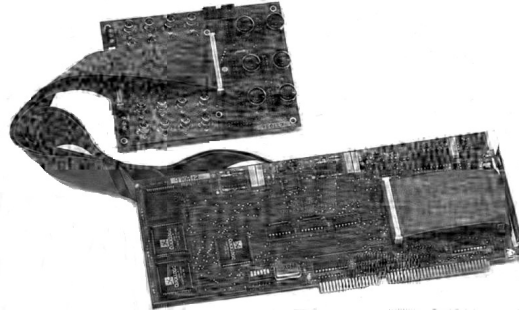


Figura 13: Scheda Multi-Q3

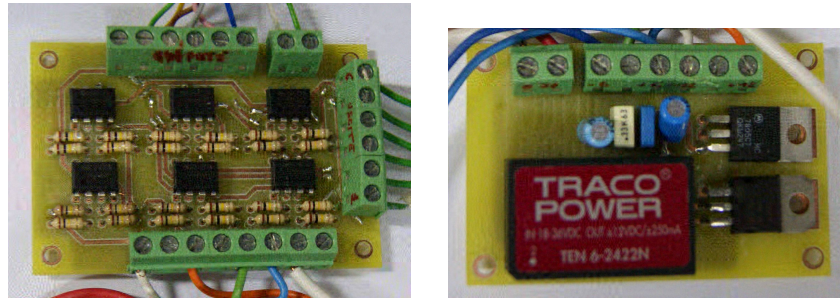
Vengono utilizzati sei degli ingressi analogici per leggere le uscite dei potenziometri, e quindi avere una misura della lunghezza attuale degli attuatori, e sei uscite analogiche per imporre i segnali di controllo alle servovalvole. Viene inoltre utilizzata una ulteriore uscita analogica per realizzare la sincronizzazione tra il calcolatore che si occupa del controllo della piattaforma e quello che si occupa del controllo della telecamera in fase di raccolta dei dati.

Per installare la scheda è necessario innanzitutto spegnere il computer, inserire i cavi a nastro piatto attraverso il backplane e poi inserire la scheda MultiQ nello slot ISA della motherboard. Infine vanno inseriti i due cavi a nastro piatto nella morsettiera che permette un più facile accesso agli ingressi e alle uscite della scheda stessa. A causa dell'assenza di driver disponibili si è reso necessario compilare due particolari S-Function per interfacciare la scheda al software Matlab. Le due S-function implementate servono una per la conversione digitale-analogica dei comandi alle servovalvole e una per la conversione analogica-digitale dei segnali provenienti dai potenziometri.

2.2.5 Schede per il condizionamento e gestione dei segnali

Analizzando con cura la parte di gestione dei segnali elettrici in ingresso al sistema si è notata una diversità tra il range dei segnali in ingresso alle servovalvole (0-10V) e quelli della scheda di acquisizione (-5V,5V). A tal motivo è dovuta la presenza di una scheda elettronica per il condizionamento e la gestione dei segnali tra i due componenti del sistema sopra citati (vedi Figura 14a).

La soluzione adottata, scelta per semplicità e economicità costruttiva, è costituita da sei semplici amplificatori operazionali LM741 in configurazione invertente e sommatore di tensione. Infatti per passare da ± 5 V a 0-10 V basta andare a sommare 5 V al segnale in ingresso.



(a) Scheda di condizionamento del segnale (b) Scheda per la gestione della tensione

Figura 14: Schede PCB (printed circuit board) utilizzate.

Per garantire il corretto funzionamento di tale scheda è necessario alimentare opportunamente gli operazionali. A ciò è dovuta la presenza di un'ulteriore scheda (Figura 14b) che converte la tensione in uscita dall'alimentatore da 24 V ai $\pm 12V$ necessari. Tale compito viene affidato ad un normale convertitore DC/DC.

Nella stessa scheda di gestione della tensione sono presenti altri due componenti: LM7905 e LM7805. Questi componenti sono degli stabilizzatori di tensione rispettivamente a $-5V$ e a $+5V$. Queste ulteriori uscite di tensione servono per alimentare i potenziometri lineari utilizzati nelle misure delle lunghezze dei link del manipolatore. I potenziometri lavorano come dei partitori di tensione variabili e danno in uscita una tensione proporzionale all'allungamento, cioè se il potenziometro è completamente contratto l'uscita si porta al potenziale di alimentazione più basso mentre, se è completamente esteso tale uscita si porta al potenziale più alto. Si è scelto di alimentarli a $\pm 5V$ in modo tale che le loro uscite si ritrovino sempre all'interno di questo range di tensione e quindi non è stato necessario inserire un'ulteriore scheda di adattamento dei livelli di tensione per l'acquisizione dei segnali dai potenziometri.

2.3 BRANDEGGIO ULISSE-COMPACT

I sistemi integrati ULISSE prodotti dalla VideoTec S.p.a. uniscono un brandeggio ad alta velocità con rotazione continua, una custodia per telecamera e un ricevitore di telemetria multiprotocollo e sono estremamente facili da installare ed utilizzare. Per rispondere ad ogni esigenza d'installazione, sono stati sviluppati differenti modelli in taglia compatta, media e maxi, con scelta tra controllo analogico o IP, con telecamera integrata, opzione di illuminatore ad IR e sensori di posizionamento assoluto. La gamma ULISSE è la soluzione ideale per

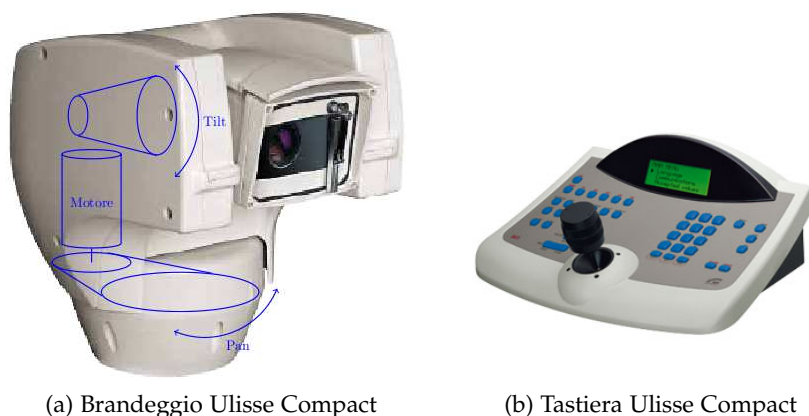
ogni tipo di applicazione per video sorveglianza in esterno ¹.

Per l'applicazione in oggetto è stato utilizzato il brandeggio della serie ULISSE denominato ULISSE COMPACT e riportato in Figura 15. Le sue caratteristiche principali sono:

- rotazione orizzontale continua
- Rotazione verticale da -90° a $+90^\circ$
- Velocità orizzontale e verticale variabile: da 0.1° fino a $200^\circ/\text{s}$
- Temperatura di esercizio: $-40^\circ\text{C}/+60^\circ\text{C}$
- Peso, 12.5 kg;
- Dimensioni, 660x330x570 mm.

La struttura portante è costituita da un telaio in lega leggera al quale vengono fissati tutti i componenti. Le movimentazioni di pan e di tilt vengono eseguite da due hybrid step motor collegati con l'asse da controllare mediante cinghie dentate e i due assi sono posizionati in modo tale da essere ortogonali e indipendenti tra loro e rispetto al LOS (Line of Sight).

All'interno della struttura trovano posto due schede elettroniche chiamate di Base e di Body; nella scheda base si trovano i circuiti per realizzare l'alimentazione del brandeggio a partire dall'alimentazione di rete. Essa rende disponibile il segnale proveniente dalla telecamera in uscita ed inoltre prevede alcuni ingressi che consentono l'azionamento dei motori tramite una tastiera in dotazione assieme al brandeggio.



(a) Brandeggio Ulisse Compact

(b) Tastiera Ulisse Compact

Figura 15: Insieme brandeggio Ulisse Compact e relativa tastiera.

La scheda body racchiude le funzioni di controllo e di comunicazione del brandeggio. Questa permette quindi il pilotaggio dei motori degli assi di rotazione (pan e tilt), e delle ottiche (zoom, focus e iris).

¹ Per maggiori informazioni sui vari prodotti sviluppati dalla ditta Videotec S.p.a consultare <http://www.videotec.com>

In questa scheda, tra le altre cose, sono presenti un microcontrollore e un FPGA. Il compito dell'FPGA è di generare i segnali di controllo cioè opportune rampe di accelerazione e comandi di velocità. La motivazione dell'utilizzo di questo dispositivo risiede nel fatto che esso consente la gestione dei vari processi con tempi certi mentre ciò non avverrebbe con un normale microcontrollore. L'FPGA genera dei riferimenti in corrente che vengono elaborati da un DAC il quale li converte in segnali analogici. Questi riferimenti sono progettati per ottenere sette distinti valori di velocità di rotazione del brandeggio. Tali valori sono una suddivisione su scala esponenziale della velocità massima del brandeggio impostabile dall'utente. Sull'asse di tilt è presente anche un piccolo encoder che ha la sola funzione di permettere una registrazione iniziale dell'asse stesso e l'individuazione dei limiti massimi di rotazione imposti dalla struttura (calibrazione dell'asse).

Dall'esterno il brandeggio si presenta come in Figura 15a, con un contenitore in plastica che racchiude tutti i componenti sopra descritti ed è completamente scomponibile per permettere agilmente all'operatore di agire dall'esterno nel caso ce ne sia bisogno. Lungo tutte le giunture sono presenti delle guarnizioni per renderlo impermeabile e quindi adatto ad operare all'esterno sotto l'azione degli agenti atmosferici. Ai fini del progetto si sono scollegate tutte le schede elettroniche interne ed i riferimenti ai motori sono dati attraverso dei driver aggiuntivi comandati direttamente dal PC utilizzato per effettuare il controllo degli assi.

2.3.1 *Hybryd Step Motor*

I motori adottati nell'Ulisse Compact sono hybrid step motor a due fasi (o HY). Questa particolare categoria di motori passo passo, sfrutta il principio dei motori a riluttanza variabile (o VR) e contemporaneamente quello dei motori a magneti permanenti (o PM). La struttura dello statore è praticamente quella di un motore VR, con la differenza che mentre nel motore VR attorno ad ogni dente trova posto l'avvolgimento di una singola fase, nei motori HY attorno ad ogni dente trovano posto due fasi diverse. Pertanto un dente non è più associabile ad una singola fase, bensì ad una coppia di fasi avvolte generalmente con versi opposti. Il rotore ha una struttura particolare. Il suo nucleo è costituito da un magnete permanente cilindrico, che produce un flusso assiale unipolare come si può notare dalla Figura 16a. La parte esterna, visibile in Figura 16b è composta da una struttura dentata, tipica del motore VR, e i denti delle due sezioni sono disallineati tra loro di mezzo passo di dentatura. Per maggiori approfondimenti relativi a tali tipi di motori si consiglia di consultare [18] I motori a passo, in generale, sono quindi caratterizzati da un movimento a scatti regolari, corrispondenti ad un preciso angolo di

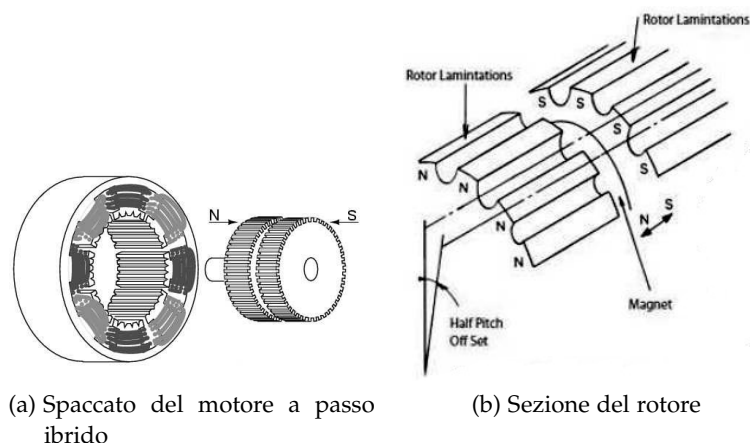


Figura 16: Rotore e statore di un motore a passo ibrido.

rotazione detto passo.

Nel caso dei motori a passo ibridi, proprio per l'effetto del disallineamento tra le due strutture dentate, si riesce ad ottenere un numero molto maggiore di passi rispetto ad un PM o ad un VR a scapito però di un maggior costo di costruzione. I vantaggi principali dei motori HY sono:

- grande risoluzione;
- alto rendimento;
- errore non cumulativo;
- elevata coppia di tenuta (holding torque);
- bassi costi se paragonati con altri tipi di motore con analoghe prestazioni.

Per contro, il principale difetto è la generazione di coppia non costante (ripple di coppia e coppia di cogging) che causa l'instaurarsi di vibrazioni nella movimentazione. Un ulteriore problema di questo tipo di motori è il limite imposto dalla curva di pull-in. Questa curva racchiude la regione di avviamento in cui il motore sotto carico può avviarsi o fermarsi istantaneamente, senza perdita di sincronismo.

Per una maggiore comprensione di questo aspetto è necessario analizzare il comportamento del motore a passo che, a causa del suo principio di funzionamento, assomiglia molto a quello del motore sincrono. Tutti i dispositivi meccanici, a causa della loro inerzia, non possono subire accelerazioni troppo brusche. In un classico motore elettrico la richiesta di un improvviso aumento di velocità provoca un momentaneo aumento dell'assorbimento di corrente ed un graduale e relativamente lento aumento di velocità. Nel caso di un motore a passo, se cambia improvvisamente la frequenza in ingresso al dispositivo di pilotaggio, il motore tenta di adeguarsi (quasi) istantaneamente alla

nuova velocità: se la coppia disponibile è sufficiente vi è una brusca accelerazione, praticamente istantanea; se la coppia non basta a vincere l'inerzia e gli attriti, il motore semplicemente si ferma e, senza un intervento esterno, non è più in grado di ripartire: si dice che il motore ha perso il passo. Questo comportamento deriva dal fatto che la velocità del motore è rigidamente controllata dall'elettronica e che la corrente assorbita è fissa e praticamente indipendente dalla coppia meccanica resistente.

Il comportamento è molto simile anche durante la fase di arresto del motore quando interviene una coppia esterna di valore elevato. Mentre il motore a corrente continua risponderebbe a tale situazione adeguandosi alle nuove condizioni ed aumentando l'assorbimento di corrente, lo stepper motor non può rallentare e se la coppia è sufficientemente piccola egli è in grado di fermarsi altrimenti il motore perde il passo e si ferma. Un esempio di curva di pull-in e pull-out è riportato in Figura 17.

Il range di avviamento dipende dall'inerzia del carico applicato. In altre parole, le curve di pull-in sono misurate in funzione di diversi momenti di inerzia e dunque forniscono una possibile coppia di avviamento già depurata dalla componente inerziale. La curva di pull-in è quindi un grosso limite perché limita di fatto la velocità di partenza e di arresto del motore.

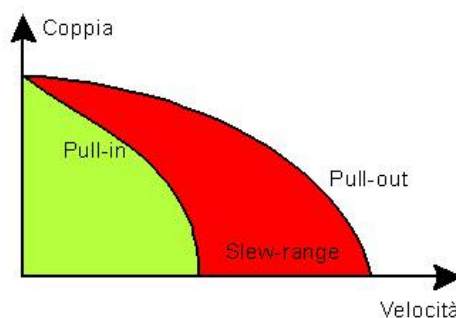


Figura 17: Hybrid step motor

La curva denominata di pull-in, comprendente l'area disegnata in verde, rappresenta l'area in cui è possibile effettuare variazioni di velocità senza eccessive precauzioni, cambiando semplicemente la frequenza delle correnti di alimentazione delle fasi. La zona indicata in rosso può essere attraversata solo a patto di variare lentamente la velocità; il cambiamento di frequenza deve quindi essere tanto più lento quanto più ci allontaniamo dalla curva di pull-in. La curva di pull-out indica dunque la massima coppia resistente che può essere applicata al motore ad una determinata velocità al fine di evitarne il blocco. Tali curve sono comprensive anche dei dispositivi meccanici collegati ai motori e dal tipo di pilotaggio ad essi applicato.

I motori presenti all'interno del brandeggio per movimentare gli assi

di pan e di tilt sono identici tra loro e sono riportati in Figura 18. Da essa in particolare si può notare il passo dei suddetti motori 1.8° , e la massima corrente applicabile per ciascuna fase di essi, 2.8A.

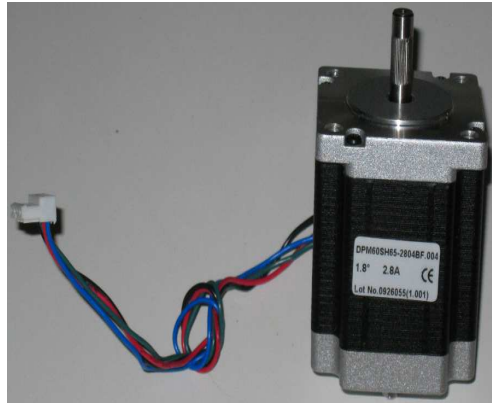


Figura 18: Hybrid step motor

2.3.1.1 *Tecnica di Pilotaggio*

I motori presenti all'interno della struttura sono gestiti sfruttando un pilotaggio in microstepping. Questa tecnica consiste nell'inviare alle due fasi del motore due correnti sinusoidali in quadratura tra loro. Tanto maggiore sarà la corrente in una fase rispetto a quella nell'altra, tanto più il rotore si posizionerà vicino al dente di statore corrispondente alla fase con corrente più elevata. Questa tecnica consente di ottenere una maggiore fluidità del movimento, un aumento della precisione nel posizionamento, per via dell'aumento della risoluzione, ed una diminuzione delle vibrazioni durante il moto. Per regolare la corrente servono naturalmente circuiti molto complessi. In pratica, nelle applicazioni industriali si usano processori dedicati oppure appositi circuiti integrati. Il vantaggio, va ribadito, consiste nell'eliminazione del funzionamento a scatti, uno dei più marcati difetti dei motori a passo nelle applicazioni di precisione.

2.3.2 *Trasmissione del moto*

La trasmissione del moto dai motori ai rispettivi assi di pan e tilt è realizzata tramite un accoppiamento a cinghia dentata con rapporto di trasmissione $\tau = 1/7$. Le cinghie dentate permettono di trasmettere grandi potenze senza slittamenti e con rendimenti elevati ed inoltre di ottenere un movimento poco rumoroso.

Tuttavia questo sistema di trasmissione presenta anche un notevole svantaggio in quanto porta ad avere inevitabili elasticità dovute alla struttura della cinghia. Questo introduce delle risonanze che provocano l'instaurarsi di fenomeni vibratorii indesiderati. Nel caso dell'asse di tilt le vibrazioni sono molto più accentuate rispetto all'asse di pan.

Ciò si pensa sia dovuto al fatto che l'asse di tilt presenta un'inertza notevolmente minore rispetto all'asse di pan e ciò comporta una minor attitudine allo smorzamento delle vibrazioni indotte dal sistema di trasmissione. La cinghia dentata ed il meccanismo di trasmissione del moto presenti all'interno del brandeggio Ulisse Compact sono rappresentati in Figura 19.

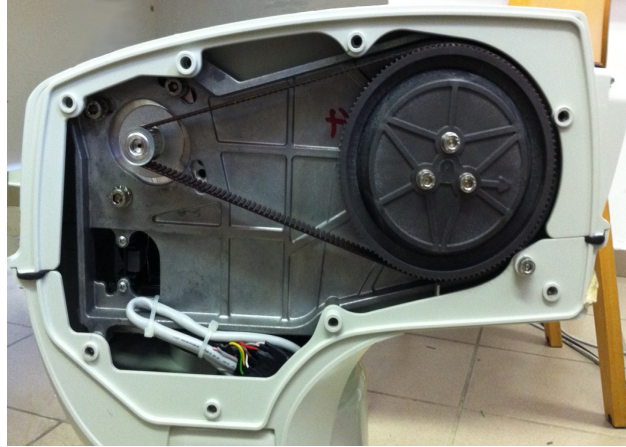
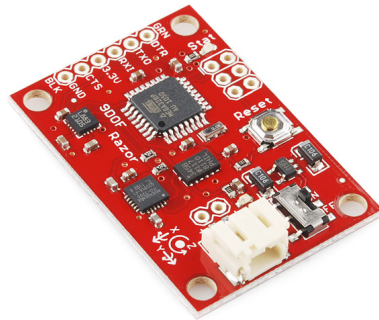
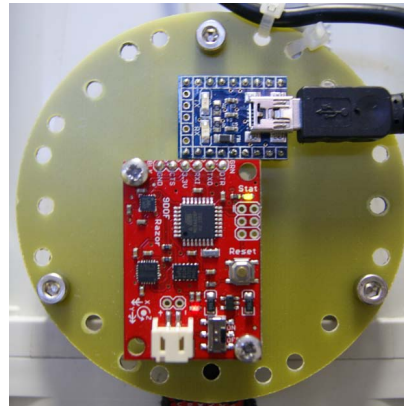


Figura 19: Cinghia di trasmissione del moto di tilt

2.4 9DOF-RAZOR IMU, SEN-10736



(a) Sensore IMU



(b) sensore IMU con adattatore seriale/USB montati sulla scheda PCB di interfaccia. Il tutto fissato sulla videocamera

Figura 20: 9DOF-RAZOR IMU, SEN-10736

All'interno del setup sperimentale realizzato è presente la scheda 9DOF-RAZOR IMU (Figura 20a) la quale è un sensore inerziale (*Inertial Measurement Unit*) che incorpora al suo interno tre sensori:

- ITG-3200, un giroscopio a tre assi;

- ADXL345, un accelerometro a tre assi;
- HMC5883L un magnetometro , anch'esso a tre assi;

Il tutto determina 9 gradi di libertà per quanto riguarda le misure inerziali.

I segnali in uscita vengono processati dalla scheda ATmega328 montata a bordo della scheda IMU e programmata attraverso il bootloader Arduino8 MHz (stk500v1). Per connettere la scheda al computer è sufficiente utilizzare un'adattatore simulando attraverso una porta USB un ingresso seriale e fornendo così l'alimentazione alla scheda attraverso la porta USB stessa. Per interfacciare la scheda IMU all'adattatore seriale si è reso necessario inserire un'opportuna scheda PCB, come si può facilmente notare dalla Figura 20b.

A bordo della scheda IMU, attraverso un opportuno algoritmo di *sensor fusion*, avviene una prima elaborazione dei dati derivanti dai 3 tipi di sensori presenti in essa; ciò permette di ottenere in uscita valori di yaw-pitch-roll che sono più facili da analizzare. Con il termine *sensor fusion* indichiamo l'insieme delle operazioni attraverso le quali misure derivanti da sensori di diversa natura vengono combinate insieme al fine di fornire in uscita dati che possono esser trattati con maggior semplicità ².

Questo meccanismo di gestione dei segnali permette alla scheda 9DOF-RAZOR di essere utilizzata come un dispositivo di controllo in tutti quei sistemi dove è richiesta la misura di orientazioni di corpi rigidi quali ad esempio i sistemi di stabilizzazione delle immagini (caso in oggetto), controlli per veicoli autonomi, UAVs (Unmanned aerial vehicle).

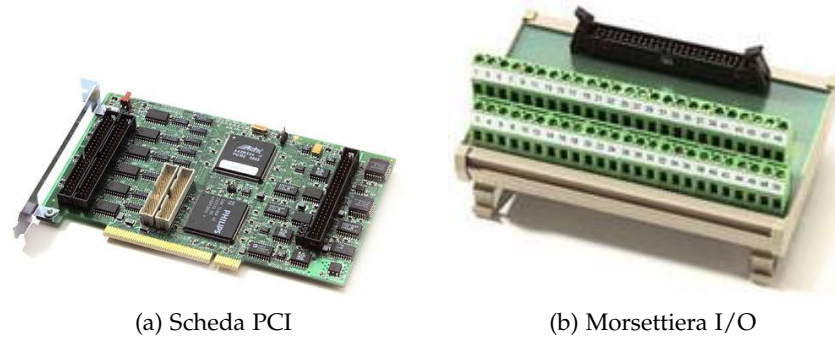
2.5 SCHEDA DI ACQUISIZIONE SENSORAY 626

Per progettare il controllo dell'Ulisse Compact si è fatto uso del software Matlab/Simulink associato al toolbox Real Time Windows Target. Questo tool permette di interfacciare in tempo reale il sistema attraverso due blocchi principali di I/O denominati Analog Input e Analog Output. Per acquisire i dati derivanti dal modello fisico si è fatto uso della scheda di acquisizione sensoray 626, visibile in Figura 21, che presenta le seguenti caratteristiche:

- 16 ingressi analogici single-ended con risoluzione di 14 bit ed input range selezionabile tra 5 V e 10 V;
- 4 uscite analogiche con risoluzione di 14 bit ed output range di 10 V ;
- 6 Ingressi encoder con risoluzione 24 bit;

² Per maggiori informazioni riguardanti l'algoritmo di *sensor fusion* implementato e gli schemi elettrici relative alla scheda PCB utilizzata per interfacciare la scheda IMU con l'adattatore seriale si consiglia di consultare [17]

- 48 I/O digitali.



(a) Scheda PCI

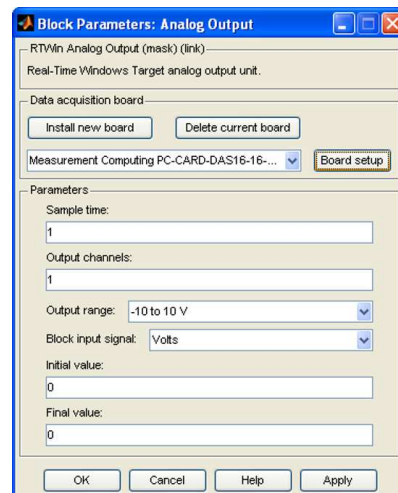
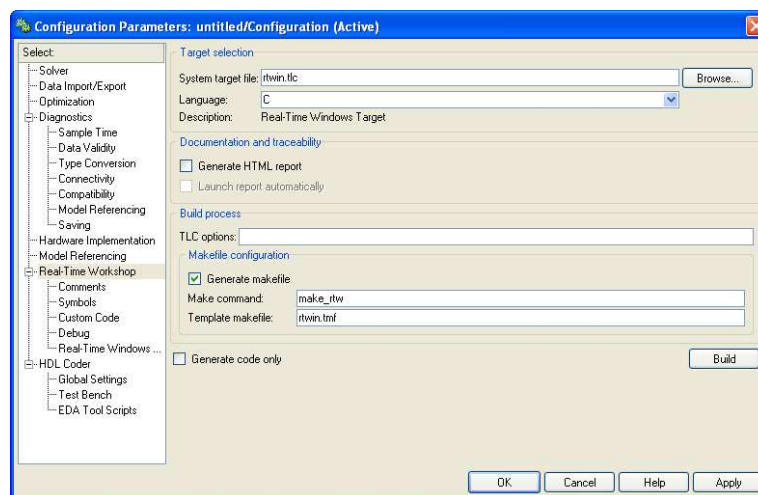
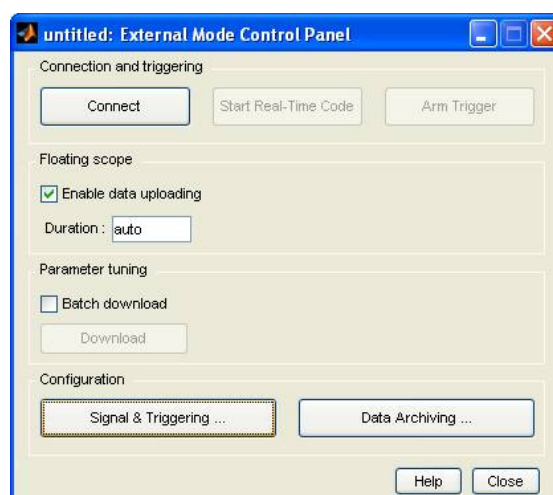
(b) Morsettiera I/O

Figura 21: Scheda Sensoray-626

Per consentire il funzionamento in real-time il file simulink contenente lo schema del controllo viene convertito in un file eseguibile che verrà utilizzato durante il normale funzionamento. Per una corretta simulazione risulta essenziale settare opportunamente i parametri dei blocchi I/O oltre ovviamente a tutti i parametri generali di simulazione.

Vengono qui di seguito descritti i passi principali per compiere queste due operazioni:

1. digitare nella *CommandWindow* di *Matlab* l'istruzione: *rtwintgt-install*;
2. entrare nell'ambiente *Simulink* di *Matlab*;
3. come prima applicazione è sufficiente selezionare dalla libreria di *Simulink* il blocchetto *Analog Output* all'interno del toolbox *Real-time Windows Target* e visualizzare la finestra delle proprietà (Figura 22). Da questa finestra è necessario selezionare *Install New Board* e scorrere l'elenco delle schede di acquisizione fino a trovare la scheda sopra citata. Fatto ciò, per verificarne il corretto funzionamento è sufficiente selezionare *Board Setup* ed effettuare il *Test*.
4. selezionare dal menù a tendina di *Simulink*: *Simulation* \Rightarrow *Configuration Parameters*: selezionare *Browser* e quindi *Rtwin.tlc*, (Figura 23).
5. dal menù a tendina di *Simulink* selezionare *Tools* \Rightarrow *External Mode Control Panel* \Rightarrow *Signal and triggering* : da qui è possibile selezionare la dimensione dei buffer su cui andare a salvare i dati, Figura 24.
6. per la configurazione del blocchetto *Analog Input* di *Simulink* è sufficiente ripetere solamente il punto 3.

Figura 22: Finestra proprietà *Analog Output*Figura 23: Finestra proprietà *Configuration Parameters*Figura 24: Finestra proprietà *External Mode Control Panel*

2.6 DRIVER

Nell'applicazione in oggetto i motori presenti all'interno del brandeggio vengono scollegati dalla scheda body poichè vengono controllati in modo indipendente dal PC attraverso opportuni driver. I driver utilizzati sono prodotti dall' "Agenzia Ing.Pini s.r.l." e fanno parte della linea *Speeder Motion*, Figura 25.

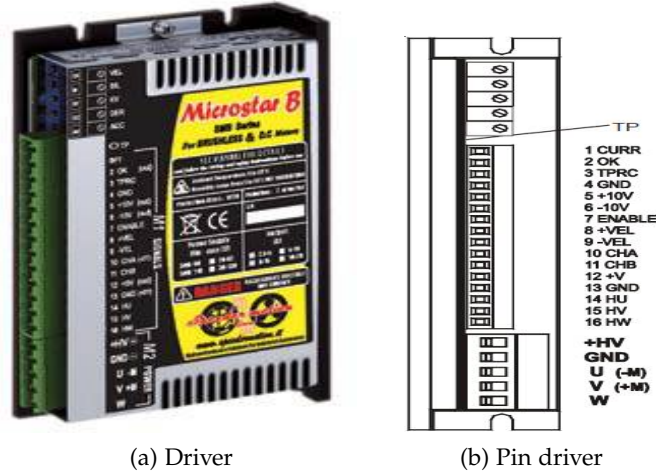


Figura 25: Driver MicroStar-B

Tali convertitori sono azionamenti adatti al pilotaggio sia di motori brushless che di motori a corrente continua ma le loro molteplici modalità operative li rendono adattabili, con qualche considerazione aggiuntiva, anche a questo tipo di applicazione.

Le modalità operative supportate da tali tipi di motori sono:

- controllo di velocità da ingresso analogico in cui il motore è controllato da un riferimento analogico (differenziale o di modo comune) di velocità in ingresso tra i morsetti \pm VEL
- controllo di coppia in cui il motore è controllato con un riferimento analogico di coppia. Questa regolazione consente di pilotare il convertitore in torque mode attraverso l'ingresso analogico in modo comune TPRC.

Sfruttando la seconda modalità operativa e applicando all'ingresso TPRC un segnale di tensione V_{ing} di max ± 10 V si ottiene in uscita un segnale in corrente con andamento analogo all'ingresso ma scalato di una costante che dipende dalla taglia dell'azionamento.

In altre parole l'azionamento può essere visto come un amplificatore di transconduttanza. Essendo l'azionamento utilizzato di taglia 10/20A la formula per determinare il massimo valore di V_{ing} da applicare in TPRC in modo da ottenere la massima corrente imponibile al motore I_{pk} è la seguente:

$$V_{ing} = \frac{10}{20} \cdot I_{pk} \quad (2)$$

Essendo I_{pk} pari a 2.8A si ottiene

$$V_{ing} = \frac{10}{20} \cdot 2.8 = 1.4V \quad (3)$$

Si osserva quindi che la costante di transconduttanza, K_{pk} , che lega la variazione della corrente in uscita con la variazione della tensione in ingresso risulta pari a 2 A/V.

Nel caso in esame sono stati usati due driver per motore, uno per fase, operanti in modalità controllo di coppia. Così facendo si è riusciti a trasformare i segnali in tensione imposti dal controllo sviluppato in Simulink in segnali di corrente adatti al pilotaggio delle singole fasi dei motori a passo illustrati precedentemente. Per consentire tale funzionamento si è quindi necessario operare su ognuno dei quattro driver alcuni collegamenti, sia per la parte di segnale che per la parte di potenza.

I 4 driver utilizzati e i rispettivi collegamenti effettuati sono riportati in Figura 26 dove si può notare anche il collegamento con il brandeggio Ulisse.

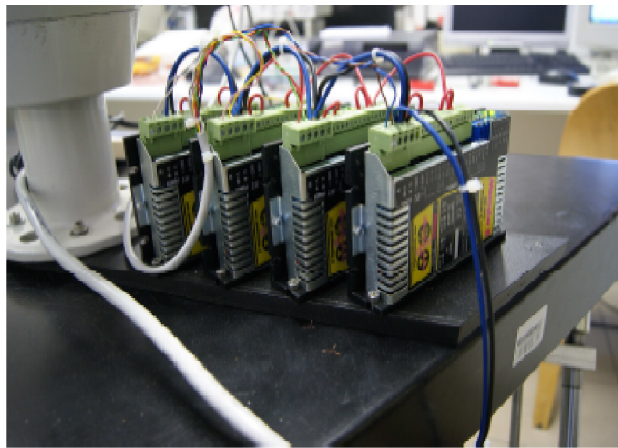


Figura 26: I 4 driver utilizzati collegati tra loro ed al brandeggio

2.7 CONNESSIONE PC

2.7.1 Introduzione

Uno degli obiettivi che ci si è preposti fin dall'inizio di questo lavoro è di effettuare la compensazione del movimento di roll in real-time. Per far ciò si devono utilizzare le misure della IMU riguardanti il roll stesso ed inviare i dati in tempo reale dal PC che rileva i dati della IMU ad un altro PC sul quale è in esecuzione un'applicazione scritta utilizzando il linguaggio di programmazione C++ al fine di elaborare le immagini in tempo reale e di cui ne parleremo approfonditamente nel prossimo capitolo.

Nell'allestire la connessione tra i due host remoti ci si è posti la domanda di quale livello di trasporto utilizzare per la nostra applicazione. Il livello di trasporto può essere considerato il cuore di un'architettura di rete poichè esso ha il compito di instaurare un collegamento logico tra le applicazioni residenti su host remoti [16].

Analizziamo ora brevemente i due tipi di protocolli che abbiamo preso in considerazione motivando successivamente la nostra scelta.

2.7.2 TCP - Transmission Control Protocol

Il protocollo TCP è un protocollo del livello di trasporto orientato alla connessione. Il TCP nasce dalla necessità di rendere affidabile un canale di comunicazione basato sui servizi del protocollo IP. Le caratteristiche principali del protocollo TCP sono:

- garantisce la consegna dello stream di dati da un host a un altro, senza duplicati o perdita d'informazioni;
- il flusso di dati proveniente dall'applicazione può essere frazionato da TCP;
- il servizio offerto da TCP è dato dal trasporto di un flusso di byte bidirezionale tra due applicazioni in esecuzione su host differenti

Poiché il trasferimento di pacchetti di per se non è affidabile, si utilizza una tecnica conosciuta come *“positive acknowledgment”* con ritrasmissione per garantirne l'affidabilità. Il trasmettitore mantiene in memoria i pacchetti che invia e aspetta l'acknowledgment per poter trasmettere i successivi. Inoltre il trasmettitore possiede anche un timer per i pacchetti inviati che ritrasmette nel caso il timer scada.

2.7.3 UDP - User Datagram Protocol

Il protocollo UDP è un protocollo del livello di trasporto non orientato alla connessione, significa che non è necessario che si instauri una vera e propria connessione tra il trasmettitore e il ricevitore. Offre quindi un servizio *“best effort”* che permette la possibilità di perdere e di far arrivare in modo non ordinato i pacchetti. La non connessione tra trasmettitore e ricevitore permette di non avere ulteriori ritardi, di semplificare lo stato di connessione del ricevitore e del trasmettitore, di avere header più piccoli e di non avere controllo della congestione, prevede quindi che le applicazioni possano comunicare tra di loro con il minimo delle informazioni possibili. Questo protocollo, non essendo sensibile ai ritardi, è molto adatto per applicazioni con streaming multimediali. Anche UDP si appoggia al protocollo IP.

2.7.4 Confronto tra UDP e TCP

Analizziamo ora le principali differenze tra i due modelli di trasporto illustrati per effettuare la scelta più opportuna per la nostra applicazione.

Per prima cosa UDP non offre nessuna garanzia dell'arrivo dei datagrammi né del loro ordine di arrivo ma li lascia gestire al livello applicazione, al contrario il TCP, considerato affidabile grazie ai meccanismi di *acknowledgment* e di ritrasmissione su timeout, riesce a garantire la consegna dei dati così come vengono inviati. Tali meccanismi sono necessari per questo tipo di protocollo poiché l'elaborazione da parte dell'applicazione non parte se tutti i pacchetti non arrivano in ordine. Altra differenza caratteristica sta nella connessione. Infatti il protocollo TCP è orientato alla connessione, pertanto per stabilire, mantenere e chiudere una connessione, è necessario inviare pacchetti di servizio i quali costituiscono un ulteriore aumento dell'overhead di comunicazione. Al contrario UDP è un protocollo non orientato alla connessione e invia solo i datagrammi richiesti a livello applicativo con un notevole risparmio. Le informazioni inviate da TCP sono lette come stream di byte, mentre per UDP sono considerate come sequenze di datagrammi. Inoltre, essendo il protocollo UDP non orientato alla connessione, ogni pacchetto viene trattato in maniera indipendente e non si fa alcun tentativo di controllo della congestione, al contrario il TCP assume che la perdita di pacchetti significhi congestione e ciò fa sì che il protocollo TCP utilizzi un algoritmo per l'ottimizzazione del throughput andando a ridurre brutalmente la quantità di pacchetti inviati (cosa che per una comunicazione real-time non va per niente bene).

Si vede quindi, con questo breve confronto, che l'utilizzo del protocollo TCP rispetto a UDP è, in generale, preferito quando è necessario avere garanzie sulla consegna dei dati, sull'ordine di arrivo dei singoli segmenti o sui tempi di consegna. È infatti utilizzato da molte delle applicazioni più popolari di internet. Al contrario il protocollo UDP è usato principalmente quando l'interazione tra i due host è idempotente o nel caso si abbiano forti vincoli sulla velocità. Molto utile risulta questo protocollo per le applicazioni in tempo reale come nel nostro caso. A noi interessa infatti che questo protocollo riesca a trasportare i dati senza eccessivi ritardi, poiché ci deve essere un'interazione in tempo reale tra i due host. Per tutti i motivi sopra citati si è scelto quindi di stabilire una connessione UDP tramite il cavo ethernet incrociato riportato in Figura 27.

Il cavo ethernet incrociato o *“crossover”* è un tipo di cavo di rete usato per connettere assieme dei dispositivi di computer direttamente dove invece vengono normalmente connessi mediante uno switch di rete, hub o router.



(a) Cavo

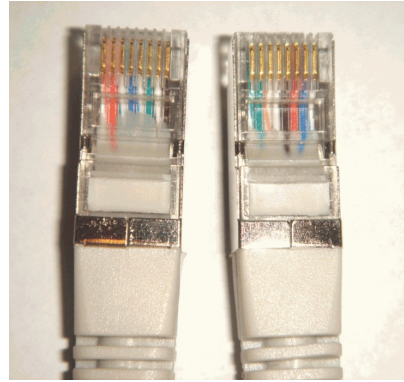
(b) Particolare che illustra la *crossatura* del cavo utilizzato

Figura 27: Cavo ethernet utilizzato per connettere i due PC

Il vantaggio di usare tale tipo di cavo rispetto ad usarne uno dritto sta appunto nel fatto che in questo modo è possibile connettere direttamente due PC mentre nel cavo dritto ciò non è possibile e bisogna per forza utilizzare un hub o altri mezzi di comunicazione. Nel modello *Simulink* per gestire l'invio e la ricezione dei dati via UDP si sono utilizzati i blocchi *PacketInput* e *PacketOutput* del tool *Real Time Windows Target* configurandoli in modo opportuno impostando gli indirizzi di rete locale che si intendono utilizzare per la trasmissione.

2.8 SETUP SPERIMENTALE COMPLETO

Il setup sperimentale completo risulta così composto :

- Simulatore di moto ondoso costituito dalla piattaforma di Stewart e dal PC per il relativo controllo;
- Brandeggio Ulisse Compact e PC per relativo controllo;
- PC dotato di framegrabber che effettua l'elaborazione video in real-time.

Nel primo componente il riferimento di movimento ondoso espresso in roll, pitch e yaw, viene elaborato dal PC e fornito alla piattaforma come spostamento lineare di ogni singolo pistone. Allo stesso tempo, al fine di creare un feedback per i controllori, vengono acquisiti i reali allungamenti dei pistoni e trasformati in segnali di roll, pitch e yaw al fine di poter controllare i reali movimenti della piattaforma. Questo loop di controllo permette la movimentazione della piattaforma con qualsiasi segnale di riferimento purché sia fornito come riferimento di rotazione sui tre assi cartesiani e che rientri nei limiti massimi di movimento della struttura.

La seconda parte del setup sperimentale è costituita dal sistema di

stabilizzazione dell'immagine. Esso è composto dal brandeggio Ulisse Compact con i rispettivi azionamenti, dal sensore IMU, e dal PC dotato di scheda per l'acquisizione dei dati Sensoray 626. Il brandeggio è installato direttamente sopra la piattaforma così come i quattro driver dei motori. La stabilizzazione dell'immagine viene effettuata attraverso un opportuno sistema di controllo presente all'interno del PC, il quale fornisce adeguati segnali di riferimento ai motori. All'interno di tale controllo è presente un segnale di retroazione di posizione fornito dalla IMU.

Per quanto riguarda il posizionamento della IMU vi sono due possibili scelte:

- collocarla sulla piattaforma mobile;
- collocarla sulla testa del brandeggio.

Si comprende subito come sia molto più utile la collocazione del sensore di misura inerziale sulla testa del brandeggio in quanto una sua collocazione sulla piattaforma mobile comporterebbe o l'inserimento di sensori di misura aggiuntivi oppure l'utilizzo di sistema di controllo in catena aperta, il quale non risulterebbe affidabile per la possibilità di una non corretta compensazione nel caso di perdita di passo del motore a passo.

La terza parte del setup sperimentale è formata dal PC utilizzato per l'elaborazione video in real-time. Attraverso opportune routine scritte utilizzando il linguaggio di programmazione C++, di cui parleremo ampiamente nel prossimo capitolo, vengono estratte alcune caratteristiche dei frame attuali che possono essere utili ai fini del controllo che ci proponiamo di realizzare. Il sistema di controllo del brandeggio e il PC dotato di framegrabber che effettua l'elaborazione video, comunicano tra di loro attraverso un cavo ethernet che consente la trasmissione bidirezionale tra i due al fine di effettuare tutte le elaborazioni possibili in real-time.

3.1 INTRODUZIONE

Come precedentemente annunciato, il primo obiettivo del nostro lavoro è quello di effettuare la compensazione in real-time del movimento di roll. Tale compensazione può esser fatta solamente via video poichè i gradi di libertà del brandeggio sono due e movimentando correttamente i motori del brandeggio stesso è possibile effettuare la compensazione solamente dei movimenti di yaw e di pitch . Per effettuare la compensazione video del roll si sono dunque utilizzate delle funzioni in parte create ad hoc e in parte messe a disposizione dalla libreria *Intel OpenCv*, necessarie per effettuare l'elaborazione in real-time dell'immagine.

L'idea sviluppata per raggiungere lo scopo precedentemente illustrato, consiste nell'utilizzare l'informazione del movimento di roll derivante dalla IMU per applicare una rotazione di ugual valore assoluto ma di segno opposto all'immagine in modo che essa non ruoti al ruotare della base ottenendo così un'immagine totalmente disaccoppiata dai movimenti della base. Inoltre poichè il centro di rotazione della piattaforma non coincide con il centro di rotazione della camera si è notato che il movimento di roll imposto alla base produce uno spostamento laterale del centro immagine anche con i movimenti di yaw e di pitch totalmente compensati.

L'utilizzo del metodo di riconoscimento dei bordi di Canny e della trasformata di Hough per il riconoscimento delle linee, ha permesso di effettuare l'elaborazione in real-time dell'immagine e di calcolare il punto di incontro delle due rette costituenti la croce presente all'interno dello scenario nel caso di target fisso. Lo spostamento di tale punto d'incontro rispetto all'origine costituisce lo spostamento residuo precedentemente introdotto in presenza dei movimenti di yaw e pitch compensati. Le coordinate x ed y (in pixel) di tale punto costituiscono gli ingressi di due anelli di controllo implementati in un altro PC tramite software Matlab/Simulink e di cui parleremo diffusamente nel prossimo capitolo.

Si capisce subito come si sia reso necessario creare delle apposite strutture per gestire il flusso di invio (delle coordinate (x,y) del punto d'incontro delle rette) e di ricezione (del movimento di roll misurato tramite la IMU e gestito dal PC che esegue il controllo) tra i due host remoti. Tale struttura sono i Sockets. e verranno illustrati all'interno di questo capitolo.

All'interno di questo capitolo verranno approfonditi alcuni aspetti ri-

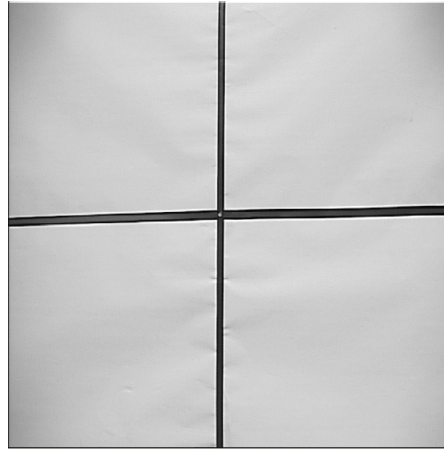


Figura 28: Scenario di ripresa con target fisso costituito dalla croce nera su sfondo bianco

guardanti la trattazione delle immagini tramite la libreria Intel OpenCv come il metodo di Canny utilizzato per effettuare l'*edge detection* e la trasformata di Hough utilizzata per il riconoscimento delle linee presenti all'interno dell'immagine. Verranno illustrati i concetti che stanno alla base del funzionamento dei Sockets e in conclusione al capitolo saranno elencate le principali funzioni utilizzate per effettuare l'elaborazione in real-time dell'immagine.

3.2 LA LIBRERIA *intel opencv*

OpenCv è una libreria *open source* utilizzata nel campo della *computer vision*. La libreria è nata dalla sezione di ricerca della *Intel* dunque le applicazioni che fanno uso dell'interfaccia di programmazione da esse fornita sono ottimizzate per processori *Pentium*, in quanto possono sfruttare le Integrated Performance Primitives (IPP) che sono delle librerie prodotte da Intel che hanno lo scopo di ottimizzare le applicazioni multimediali.

Tale libreria è scritta sia in C che in C++ ed è inoltre multiplatforma essendo utilizzabile in Linux, Windows e Mac, senza dover apportare considerevoli modifiche. Inoltre, il fatto che la libreria sia *open source*, assicura un costante miglioramento delle funzioni implementate poichè chiunque voglia migliorare il codice presente eliminando bug, aggiungendo funzioni, scrivendo codice o modificando quello esistente può farlo rispettando alcuni canoni imposti dagli sviluppatori. Al momento attuale la libreria contiene più di 500 funzioni che coprono molti campi della computer vision come la sicurezza e la videosorveglianza, la robotica, l'elaborazione immagine nel campo della medicina e la visione stereoscopica. OpenCV si presta perfettamente per effettuare operazioni in real-time poichè presenta tempi di esecuzione e compilazioni notevolmente inferiori a degli script di Matlab,

che pur danno risultati soddisfacenti [5]. Tutti questi motivi giustificano la scelta di utilizzare tale libreria per effettuare le operazioni necessarie al fine di raggiungere i nostri scopi, come la rilevazione dei contorni di un'immagine, l'individuazione delle linee nello scenario ripreso, la rotazione e molte altre operazioni.

3.3 IMMAGINI DIGITALI E RAPPRESENTAZIONE DEL COLORE

L'immagine viene di norma rappresentata all'interno dei calcolatori come una matrice, $m \times n$, di elementi discreti, detti pixel, ciascuno corrispondente ad un colore, ed è definita come una funzione $F(x,y)$ dell'intensità luminosa, il cui valore o ampiezza corrispondente ad una generica coppia di coordinate spaziali (x,y) , indica il valore dell'intensità luminosa del punto identificato dalle coordinate stesse [6]. Il numero di pixel presenti dipende dalla dimensione del formato dell'immagine che si desidera ottenere. L'acquisizione delle immagini stesse avviene tramite dispositivi che per loro stessa natura introducono degli errori e del rumore che devono essere filtrati e corretti nel modo opportuno. I colori presenti nelle varie immagini sono rappresentati tramite opportune combinazioni di tre colori fondamentali (RGB, Red Green Blue).

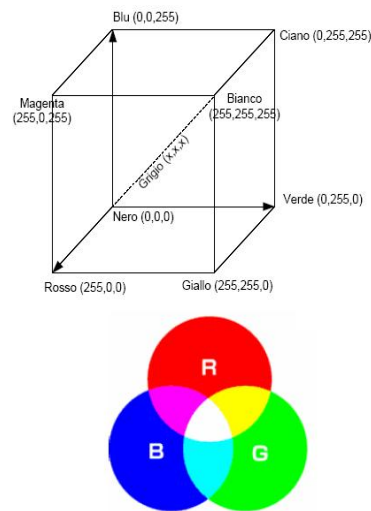


Figura 29: Rappresentazione RGB

Come si può notare dalla Figura 29, lo spazio RGB viene rappresentato come un cubo, in cui lo spigolo in basso a sinistra rappresenta il nero, quello opposto il bianco, i grigi si trovano sul segmento che unisce il nero ed il bianco ed un generico colore sarà rappresentato da un punto all'interno del cubo. L'occhio umano percepisce la somma delle 3 componenti come un unico colore.

Le immagini digitali con cui dobbiamo lavorare possono essere di due tipi:

1. vettoriale
2. raster, anche detta *bitmap*

Nel primo caso sono descritti degli elementi primitivi, quali linee o poligoni, che vanno a comporre l'immagine mentre nel secondo l'immagine è composta da una matrice di punti, che sono i pixel precedentemente menzionati, la cui colorazione è definita tramite uno o più valori numerici. Mentre nelle immagini a colori viene memorizzato solitamente il livello di intensità dei colori fondamentali, nelle immagini monocromatiche in scala di grigio, dette impropriamente in bianco e nero, il valore indica l'intensità del grigio, che varia dal nero al bianco. Il numero (detto anche profondità) di colori o di livelli di grigio possibili dipende dal massimo numero di combinazioni permesse dalla quantità di bit utilizzata per ognuno di questi dati: un'immagine con 1 bit per pixel avrà al massimo due combinazioni possibili (0 e 1) e quindi potrà rappresentare solo due colori o solo bianco e nero; nelle immagini a 4 bit per pixel, si possono rappresentare al massimo 16 colori o 16 livelli di grigio; un'immagine a 8 bit per pixel, 256 e così via. Da quanto detto si comprende facilmente che qualsiasi immagine a colori possa essere rappresentata tramite la seguente funzione vettoriale:

$$F(x, y) = f_1(x, y) + f_2(x, y) + f_3(x, y) \quad (4)$$

in cui ciascuna componente è detta canale ed indica la funzione di luminosità in funzione delle coordinate spaziali di uno dei colori fondamentali. Se si considera anche il fatto che le immagini vengono percepite per mezzo della luce riflessa dagli oggetti, è possibile considerare la $F(x, y)$ come costituita da due componenti principali:

1. Componente di illuminazione, indicata con $i(x, y)$, che rappresenta la quantità di luce diretta incidente sulla scena
2. Componente di riflessione, indicata con $r(x, y)$, che rappresenta la quantità di luce riflessa dagli oggetti presenti sulla scena.

Allora un'immagine può essere rappresentata in maniera alternativa come segue:

$$F(x, y) = i(x, y) \cdot r(x, y) \quad (5)$$

dove le componenti di illuminazione e riflessione risultano limitate secondo le relazioni:

$$0 < i(x, y) < \infty \quad 0 < r(x, y) < 1 \quad (6)$$

in cui per 0 si intende l'assorbimento totale mentre per 1 la totale riflessione. Nel caso si considerino immagini in movimento bisogna inserire all'interno di tali equazioni anche la dipendenza dal tempo. I principali tipi di immagini con cui ci troviamo normalmente a lavorare sono:

1. Immagine a colori
2. Immagine a toni di grigio
3. Immagine in bianco e nero (B/W)

Un'immagine a toni di grigio è rappresentata come una matrice di interi indicanti i valori di intensità luminosa di ogni pixel su una scala di grigi. È convenzione assumere i livelli discreti di grigio egualmente spazati in un intervallo ben definito di valori, che normalmente va da 0 a 255 nel caso si usino 8 bit per indicare il grado di grigio. In genere il valore zero indica il nero, mentre il valore massimo indica il bianco. I valori compresi tra zero ed il valore massimo individuano tutti i toni di grigio intermedi. Questo tipo di immagine viene ampiamente utilizzato nella *Computer Vision* in quanto necessita di minore onerosità computazionale vista la trattazione di semplici matrici unidimensionali.

Le immagini a colori risultano invece più difficili da trattare vista la rappresentazione che se ne fa in memoria. Un colore infatti può essere decomposto nella somma di tre colori fondamentali rosso, verde e blu (RGB), ciascuno ovviamente preso con un valore opportuno. Dato che lo spazio RGB è cartesiano, il colore di un pixel è rappresentato da un vettore e le sue componenti rappresentano la quantità di rosso, verde e blu rispettivamente necessarie a ottenere quel colore.

Un'immagine viene invece detta binaria se i pixel che la descrivono possono assumere solamente due valori: 0 e 1, cioè se fisicamente può assumere solamente due colori, bianco e nero. Questo tipo di immagine riveste un ruolo di grande importanza nel campo della *Computer Vision* poichè sono la base dei processi di riconoscimento ed individuazione dei target.

L'origine del sistema di riferimento viene normalmente posta in alto a sinistra con gli assi orientati come in Figura 30.

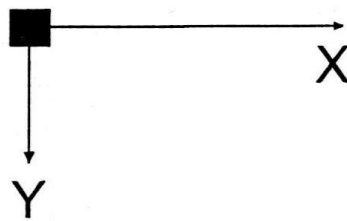


Figura 30: Sistema di riferimento

3.4 FEATURES, EDGE E CORNER DELLE IMMAGINI

Partendo dall'idea di rappresentazione digitale delle immagini si nota che la presenza di oggetti o di particolari punti di interesse è associata a una ben determinata caratteristica nell'immagine stessa. In

Figura 31 è riportata un'immagine in cui sono state selezionate tre zone con caratteristiche ben diverse tra loro.

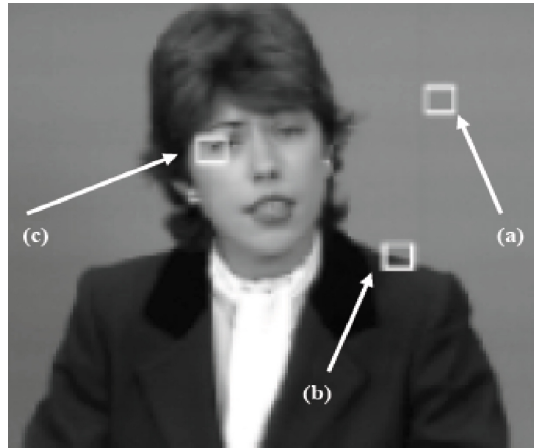


Figura 31: Immagine in cui sono state selezionate tre zone con caratteristiche diverse

Le features (caratteristiche) di un'immagine sono definite come parti di immagine che sono:

- locali: caratteristica locale di un'immagine
- significativi: sono interessanti per il problema specificata
- rilevabili: esiste un modo per determinarle

La regione che l'occhio umano percepisce come facente parte dello sfondo viene definita *featureless* ed è caratterizzata nella rappresentazione digitale da valori di intensità uniformi. Nel caso invece la zona in esame presenti una discontinuità della funzione di distribuzione, allora tale zona verrà definita *edge* se tale discontinuità è lungo una sola direzione, *corner* se la funzione appare discontinua in due direzioni. Edge e corner hanno insita nella loro struttura una certa invarianza alle condizioni di illuminazione e, a meno di casi estremi, è altamente probabile che una discontinuità sui valori di intensità di un'immagine digitale permanga al variare di tali condizioni. Facendo dunque riferimento alla Figura 31 si può evidenziare come il punto (a) corrisponda a un *featureless*, il punto (b) ad *edge* mentre il punto (c) ad un *corner* [7].

In letteratura sono presenti numerosi algoritmi per la determinazione di *edge* e *corner* all'interno dell'immagine tra cui citiamo quello di Harris, quello di Marr-Hildreth, di Canny e quello di Kitchen. Tra tutti questi citati, abbiamo scelto quello di Canny perchè si tratta di quello che meglio si adatta per la nostra applicazione e sarà illustrato approfonditamente nel prossimo paragrafo.

3.5 ALGORITMO DI CANNY

L'algoritmo di Canny (Canny, 1986) è un operatore per il riconoscimento dei contorni. Il dato di input dell'algoritmo è un'immagine in toni di grigio, sulla quale vengono effettuate una serie di operazioni. Prima di passare alla descrizione accurata del funzionamento di tale algoritmo per l'edge detection, è utile inserire il concetto di operatore di Sobel necessario per comprendere il funzionamento del filtro di Canny.

La funzione di magnitude (ampiezza) del gradiente di un'immagine da analizzare per la localizzazione dei bordi è la seguente:

$$d(x, y) = \sqrt{\Delta_x^2 + \Delta_y^2} \quad (7)$$

con

$$\Delta_x = I(x+1, y) - I(x, y) \quad \Delta_y = I(x, y+1) - I(x, y) \quad (8)$$

dove $I(x, y)$ è la funzione di intensità dei vari punti di coordinate (x, y) . Un'alternativa alla ricerca dei massimi della derivata del primo ordine è quella di trovare l'attraversamento dello zero nella derivata seconda che può essere approssimata come la differenza tra le derivate prime di due pixel adiacenti:

$$f''(x) = \Delta_x - \Delta_{x+1} \quad (9)$$

ed andando a sostituire la formulazione del gradiente espressa nella (7) all'interno di (9) si ottiene:

$$f''(x) = -\Delta_x + 2\Delta_{x+1} - \Delta_{x+2} \quad (10)$$

Le equazioni appena indtrodotte sono utili in quanto stanno alla base del funzionamento del *Filtro di Sobel*, il quale a sua volta è utilizzato dall'operatore di Canny per individuare gli edge.

La funzione (7) che esprime il gradiente della funzione di intensità può essere notevolmente influenzata dal rumore presente all'interno dell'immagine dovuto alla modalità con cui viene acquisita la stessa. Per eliminare tale problema la funzione gradiente viene convoluta con due matrici di cui una relativa all'asse x (F_x) che si occupa di estrarre la componente orizzontale e una relativa all'asse y (F_y) che si occupa di estrarre la componente verticale. Le due matrici suddette sono riportate in (11).

$$F_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad F_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad (11)$$

Per capire il funzionamento del filtro di Sobel appena illustrato prendiamo in considerazione una matrice quadrata della stessa dimensione dei filtri; x è l'elemento da analizzare.

$$F = \begin{pmatrix} a & b & c \\ d & x & e \\ f & g & h \end{pmatrix} \quad (12)$$

L'azione di ogni filtro è quella di mascherare ogni elemento della matrice da analizzare con il peso relativo. Come si può notare applicando le matrici di filtraggio lungo gli assi x e y espresse dalle (11) viene dato un peso maggiore agli elementi che si trovano sopra ed accanto all'elemento da analizzare. Da ciò si deduce subito che i contorni non possono essere individuati in nessuno dei punti che appartengono ai punti più esterni dell'immagine (in pratica la prima e l'ultima riga/colonna della matrice che descrive l'immagine). Per ottenere il gradiente lungo le due direzioni x e y è sufficiente sommare i valori contenuti all'interno delle matrici filtrate. In particolare nel nostro esempio la matrice (12) filtrata dalle due matrici di (11) risulta:

$$M_x = \begin{pmatrix} -1 \cdot a & 0 \cdot b & 1 \cdot c \\ -2 \cdot d & 0 \cdot x & 2 \cdot e \\ -1 \cdot f & 0 \cdot g & 1 \cdot h \end{pmatrix} \quad (13)$$

$$M_y = \begin{pmatrix} -1 \cdot a & -2 \cdot b & -1 \cdot c \\ 0 \cdot d & 0 \cdot x & 0 \cdot e \\ 1 \cdot f & 2 \cdot g & 1 \cdot h \end{pmatrix}$$

mentre i rispettivi gradienti risultano:

$$\begin{aligned} G_x &= -1 \cdot a - 2 \cdot d - 1 \cdot f + 1 \cdot c + 2 \cdot e + 1 \cdot h \\ G_y &= -1 \cdot a - 2 \cdot b - 1 \cdot c + 1 \cdot f + 2 \cdot g + 1 \cdot h \end{aligned} \quad (14)$$

A questo punto basta calcolare il gradiente come segue:

$$G = |G_x| + |G_y| \quad (15)$$

e confrontare questo valore ottenuto con la soglia. Se il gradiente risulta essere maggiore della soglia allora il punto rispetto al quale è stato calcolato il gradiente è un punto di contorno, altrimenti no. Siamo ora pronti per descrivere l'algoritmo di Canny che è stato utilizzato ai fini del nostro progetto. Le origini di tale algoritmo vanno cercate nel 1986 quando lo statunitense John F. Canny sviluppò un algoritmo per il riconoscimento dei contorni che viene ormai definito come lo standard in questo campo. L'algoritmo si basa su 3 obiettivi:

- tasso d'errore basso: tutti i bordi devono essere trovati con alta probabilità;

- buona localizzazione: la distanza tra il bordo reale e quello individuato deve essere minima;
- unicità della risposta per bordo: ogni bordo reale deve generare un'unica risposta.

Analizzando l'ampiezza del gradiente illustrato precedentemente Canny evidenziò delle ampie rampe intorno ai massimi locali in corrispondenza di bordi e quindi una semplice sogliatura non è sufficiente perchè a causa del rumore ci ritroveremo con dei massimi che non hanno riscontro nell'immagine reale priva di rumore.

L'algoritmo presentato si divide quindi in 4 fasi :

1. Riduzione del rumore;
2. Ricerca del gradiente di luminosità dell'immagine;
3. Soppressione dei non-massimi;
4. Individuazione dei contorni mediante sogliatura con isteresi.

La riduzione del rumore avviene effettuando l'operazione di convoluzione tra l'immagine e un filtro gaussiano con deviazione standard unitaria il cui scopo è quello di eliminare il rumore presente alle alte e alle basse frequenze. Il valore della deviazione standard del filtro da applicare può essere variato in modo da prediligere il riconoscimento di bordi piccoli e netti rispetto a quelli grandi e gradualmente, mentre la dimensione del filtro viene predisposta di conseguenza dal software che gestisce l'algoritmo. La seconda fase del procedimento consiste nella ricerca del gradiente di luminosità all'interno dell'immagine sotto studio. Le definizioni di gradiente lungo l'asse x , lungo l'asse y e la loro somma risultano:

$$\begin{aligned} G_x(x, y) &= \frac{dF(x, y)}{dx} \\ G_y(x, y) &= \frac{dF(x, y)}{dy} \\ G &= \sqrt{G_x^2 + G_y^2} \end{aligned} \quad (16)$$

Nella maggior parte dei casi i valori dei gradienti nelle due direzioni vengono calcolate utilizzando l'operatore di Sobel visto in precedenza e rappresentato dalle equazioni (11). Questo operatore di filtraggio garantisce una discreta approssimazione pur mantenendo bassa la difficoltà computazionale. Si definisce θ l'angolo di direzione del contorno calcolato nel seguente modo:

$$\theta = \arctan \frac{G_y}{G_x} \quad (17)$$

In questa fase appena descritta dunque viene restituita per ogni pixel la direzione e il valore di massimo gradiente.

La fase successiva è chiamata fase di *soppressione dei non massimi*. La strategia adottata per identificare i massimi locali, consiste nell'eliminare i non-massimi basandosi sull'idea che ogni pixel centrato in un bordo avrà un'intensità superiore rispetto ai 2 pixel vicini nella direzione del gradiente. Dividiamo quindi l'intorno di un ogni pixel in 4 regioni (giallo, verde, rosso e blu) corrispondenti a 4 direzioni, come illustrato in Figura 34.

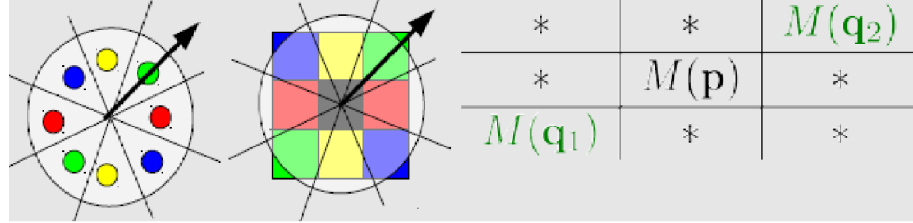


Figura 32: Fase di soppressione dei non-massimi

Verifichiamo in che regione cade il vettore gradiente del pixel in centro. Se l'intensità del pixel centrale $I(p)$ è inferiore all'intensità di almeno uno dei 2 pixel vicini lungo la direzione del gradiente, il pixel viene soppresso. Facendolo per ogni pixel otteniamo una nuova immagine I_N rappresentata dall'equazione (18).

$$I_N = \begin{cases} M(p) & \text{se } M(p) \geq M(q_{1,2}) \\ 0 & \text{altrimenti.} \end{cases} \quad (18)$$

L'ultima fase dell'algoritmo consiste nell'effettuare la sogliatura di I_N per eliminare i falsi bordi. Usare una singola soglia potrebbe avere come conseguenza il riconoscimento di falsi bordi se questa è troppo bassa oppure l'eliminazione di bordi reali se troppo alta. L'algoritmo di Canny utilizza quindi due soglie τ_L e τ_H in modo da avere:

$$2 \leq \frac{\tau_H}{\tau_L} \leq 3 \quad (19)$$

Ogni punto il cui gradiente sia superiore al valore di soglia alta è automaticamente definito parte del contorno (ne viene assegnato il valore 1), inoltre ogni punto contiguo ad un punto del contorno che abbia un valore del gradiente superiore al valore di soglia bassa entra a far parte del contorno, assegnando così valore 0 a tutti i punti rimanenti e quelli al di sotto della soglia bassa. Esprimendo questo concetto in formule matematiche ricaviamo, a partire dall'immagine di partenza, due immagini di bordi "forti" (I_{NH}) e bordi "deboli" (I_{NL}):

$$\begin{aligned} I_{N,H}(x, y) &= I_N(x, y) \geq \tau_H \\ I_{N,L}(x, y) &= \tau_L \leq I_N(x, y) < \tau_H \end{aligned} \quad (20)$$

In Figura 33 i pixel marcati in I_{NH} sono considerati validi. A seconda del valore della soglia T_H i bordi presentano dei buchi. Per ottenere

bordi più lunghi si considerano bordi validi anche tutti quei pixels marcati in I_{NL} che sono connessi direttamente o tramite una catena a un pixel marcato in I_{NH} .

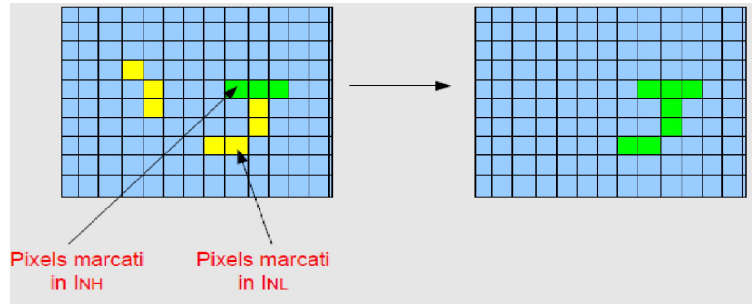


Figura 33: Sogliatura immagine

3.6 TRASFORMATA DI HOUGH

3.6.1 Generalità

Uno dei problemi maggiormente presenti nel campo dell'elaborazione delle immagini digitali è l'identificazione di eventuali rette presenti in esse. Nei casi più semplici il problema si riduce a figure che contengono un certo numero di punti discreti che stanno su uno sfondo bianco. È chiaro che il problema può essere risolto per ogni grado di accuratezza desiderato, semplicemente testando le rette passanti per ogni coppia di punti. Secondo questo procedimento se si hanno a disposizione n punti, il numero di operazioni richieste è proporzionale a n^2 , e per n grande la complessità computazionale potrebbe diventare ingestibile per applicazioni pratiche.

Un metodo ingegnoso, dovuto ad Hough, consiste nel sostituire il problema originale di individuazione di punti allineati con un problema matematicamente equivalente che consiste nell'individuare rette concorrenti. Questo metodo consiste nel mappare ogni punto presente nell'immagine in una retta nello spazio dei parametri. Lo spazio dei parametri è definito da una rappresentazione parametrica utilizzata per descrivere le rette presenti nel piano immagine. Questa tecnica, introdotta da Hough nel 1962, ha il vantaggio di essere molto robusta al rumore ed è ampiamente utilizzata nel campo dell'elaborazione delle immagini digitali e per il riconoscimento dei volti tramite la sua forma generalizzata per circonferenza ed ellissi.

3.6.2 Coordinate cartesiane e coordinate polari

Una generica retta può essere espressa tramite la seguente espressione:

$$y = mx + q \quad (21)$$

dove (x,y) sono le coordinate cartesiane di un generico punto per cui passa la retta indicata, m è il coefficiente angolare di tale retta e q l'offset lungo l'asse delle ordinate. Tale rappresentazione ha lo svantaggio che non tutte le rette possono essere rappresentate nel piano dei parametri $(m-q)$, poichè nel caso di retta completamente verticale il coefficiente angolare tende ad essere infinito e la retta non è così più quantitativamente rappresentabile. Per ovviare a tale problema bisogna optare per una differente rappresentazione: una valida alternativa è costituita dalla rappresentazione tramite coordinate polari. Le coordinate polari rappresentano un sistema di rappresentazione bidimensionale nel quale ogni punto è identificato da un angolo e da una distanza rispetto ad un punto fisso detto polo. Tale sistema di coordinate polari è in corrispondenza biunivoca con un sistema di coordinate cartesiane, cioè ad ogni vettore di coordinate cartesiane corrisponde uno ed uno solo vettore in coordinate polari. Siccome il sistema di coordinate è bidimensionale, ogni punto è determinato da due coordinate polari: la coordinata angolare e la coordinata radiale. La coordinata angolare di solito è denotata con la lettera θ ed identifica l'angolo che la retta a 0 gradi deve spazzare in senso antiorario per andare a sovrapporsi a quella che congiunge il punto al polo. Mentre la coordinata radiale denotata dalla lettera ρ rappresenta la distanza dal punto ad un punto fisso detto polo. Nel nostro caso il polo viene fatto coincidere con l'origine degli assi.

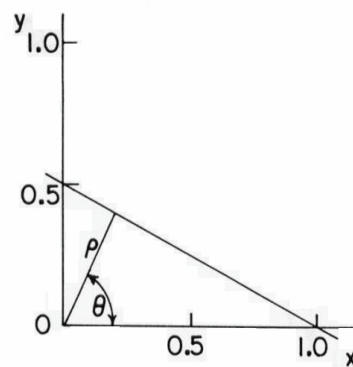


Figura 34: Rappresentazione in coordinate polari

Attraverso alcune semplici ed opportune relazioni matematiche si ottiene la rappresentazione della stessa retta indicata in (21) in coordinate polari ¹:

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (22)$$

La rappresentazione di tale retta nello spazio dei parametri è univoca se si scelgono i parametri $\theta \in [0, \pi]$ mentre ρ può assumere qualsiasi valore nei reali.

¹ Per maggiori approfondimenti sul procedimento matematico necessario per effettuare il passaggio da coordinate cartesiane a coordinate polari consultare il documento *Studio e applicazione della trasformata di Hough per il riconoscimento di target umani*[12]

3.6.3 La trasformata di hough per il riconoscimento delle linee

La retta espressa in (21) può essere riscritta come:

$$y - mx - q = 0 \quad (23)$$

Si consideri lo spazio dell'immagine (x,y) e lo spazio dei parametri (m,q) ; se vengono fissati i valori di \hat{m} e \hat{q} l'equazione può essere vista come una mappatura di tipo biunivoco dallo spazio dei parametri allo spazio dell'immagine e restituisce i valori dei punti presenti nell'immagine appartenente alla retta di parametri stessi. Se invece i parametri fissati sono (\hat{x},\hat{y}) abbiamo una mappatura inversa e cioè dallo spazio immagine allo spazio dei parametri che restituisce dei parametri (m,q) che nello spazio immagine rappresentano delle rette passanti per i punti (x,y) .

Si consideri ora uno spazio immagine in cui sono presenti due punti $P_1 (x_1,y_1)$ e $P_2 (x_2,y_2)$, rappresentati in Figura 35, e scriviamo entrambe le equazioni relative alle rette passanti per i suddetti punti in coordinate cartesiane.

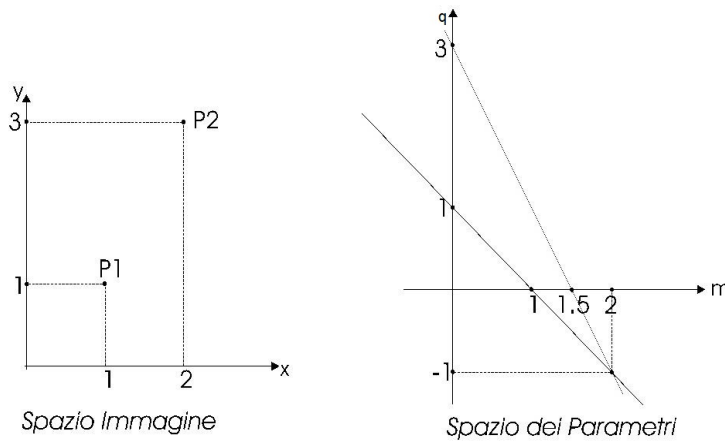


Figura 35: Punti nello spazio immagine e rispettive rette nello spazio dei parametri

$$\begin{cases} \hat{y}_1 - m\hat{x}_1 - q = 0; \\ \hat{y}_2 - m\hat{x}_2 - q = 0; \end{cases} \quad (24)$$

La retta passante per i due punti la si ricava risolvendo il sistema (24) da cui si ricavano facilmente i parametri m e q della retta stessa.

Se considerassimo al posto di due, un numero elevato di punti tutti allineati tra di loro, troveremo che nello spazio dei parametri tutte le rispettive curve si incontrerebbero in un unico punto che corrisponde al punto di coordinate (m,q) che sono il coefficiente angolare e l'offset della retta passante per tutti i punti indicati.

La trasformata di Hough si basa proprio su questo concetto. Data

una forma da individuare rappresentata mediante un insieme di parametri, ogni punto dell'immagine determina una curva nello spazio dei parametri compatibili con quel punto. Il fatto che un elevato numero di curve presenti nello spazio dei parametri si incontrino in un determinato punto, sta ad indicare che un elevato numero di punti dello spazio immagine sono compatibili con una particolare forma da individuare, e che quindi tale forma sia presente all'interno dell'immagine.

Per poter effettivamente utilizzare la trasformata di Hough è necessario effettuare una quantizzazione dello spazio dei parametri e successivamente rappresentarli in un array di memoria, anche detto array accumulatore (*"accumulator array"*), avente sulle righe i valori di ρ e sulle colonne i valori di θ . A questo punto si può facilmente capire il motivo che consiglia di utilizzare le coordinate polari invece che quelle cartesiane poichè nel primo caso è possibile rappresentare qualsiasi tipo di retta presente nello scenario ripreso pur imponendo un limite massimo ai due parametri ρ e θ , cosa che invece non è possibile con le coordinate cartesiane poichè in presenza di rette verticali il coefficiente angolare m tende a infinito. Gli intervalli di variazione di ρ e θ sono fissati sulla base delle caratteristiche dell'immagine originale. Tipicamente si ha:

$$-\rho_{\max} \leq \rho \leq \rho_{\max} \quad -\pi \leq \theta \leq \pi \quad (25)$$

dove $\rho_{\max} = 0.5 * (NR^2 + NC^2)^{1/2}$ e (NR, NC) sono le dimensioni dell'immagine. Il numero dei livelli di quantizzazione va poi scelto in base all'accuratezza desiderata. Una scelta quasi sempre soddisfacente è $\max(NR, NC)$.

Per ogni punto dello spazio immagine è necessario tracciare la curva ad esso associato all'interno dell'accumulatore. Ciò viene effettuato valutando l'equazione che definisce la mappatura dallo spazio immagine allo spazio dei parametri ed incrementando ogni cella dell'array accumulatore per il quale tale equazione è soddisfatta. Questa operazione viene chiamata *"voting"*. Ogni cella dell'array accumulatore conterrà un valore che rappresenta il numero di voti, che è pari al numero di curve che attraversano la cella. Perciò le celle che avranno il valore più elevato corrisponderanno a dei punti nello spazio dei parametri in cui si intersecano numerose curve.

Grazie alla struttura ad accumulatore si capisce subito quanto questo procedimento sia robusto al rumore. È improbabile che punti *casuali* dovuti al rumore vadano tutti ad accumularsi in modo coerente tale da produrre dei falsi picchi all'interno dell'array accumulatore e portando ad una errata individuazione della forma cercata.

Supponiamo ora di avere un insieme di punti di coordinate (x_i, y_i) e di considerare la rappresentazione tramite coordinate polari. Si capisce facilmente che a ciascun punto nel piano immagine corrisponde

una sinusoide nel piano dei parametri (Figura 36) governata dall'equazione:

$$\rho = x_i \cos \theta + y_i \sin \theta \quad (26)$$

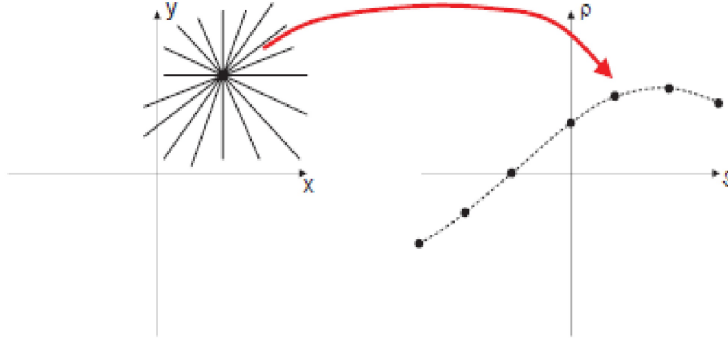


Figura 36: Trasformazione di un punto nel piano immagine a piano dei parametri in coordinate polari

L'insieme di tutte le sinusoidi relative ai vari punti allineati tra loro avranno un unico punto in comune. Questo punto nel piano (ρ, θ) denominato (ρ_0, θ_0) , definisce la retta passante per i punti collineari situati nel piano immagine. Il problema di individuazione di punti allineati può essere convertito in uno equivalente di individuazione di curve che si intersecano nello stesso punto e, seguendo lo stesso ragionamento, è possibile esprimere anche una proprietà duale a questa.

Dato un insieme di k punti nello spazio dei parametri (ρ_i, θ_i) che appartengono tutti alla retta di equazione:

$$\rho = x_0 \cos \theta + y_0 \sin \theta \quad (27)$$

è facile mostrare che ad ogni punto (ρ_i, θ_i) corrisponde una retta passante per (x_0, y_0) . Le principali proprietà della trasformata di Hough si riassumono in quattro punti:

- *Un punto nel spazio immagine corrisponde alla presenza di una curva sinusoidale nello spazio dei parametri;*
- *Un punto nel piano dei parametri corrisponde alla presenza di una retta nel spazio immagine;*
- *Punti che giacciono sulla stessa retta nello spazio immagine, corrispondono a curve che si intersecano in uno stesso punto nello spazio dei parametri;*
- *Punti che giacciono sulla stessa curva nel piano dell'immagine corrispondono a rette che si intersecano in uno stesso punto nel spazio immagine.*

In Figura 37a, sono rappresentati una serie di punti allineati tra loro mentre in Figura 37b la loro rappresentazione nella matrice di Hough; il punto d'incontro tra tutte le sinusoidi ha coordinate ρ e θ che sono i parametri della retta che passa tra i punti allineati rappresentati nello spazio immagine.

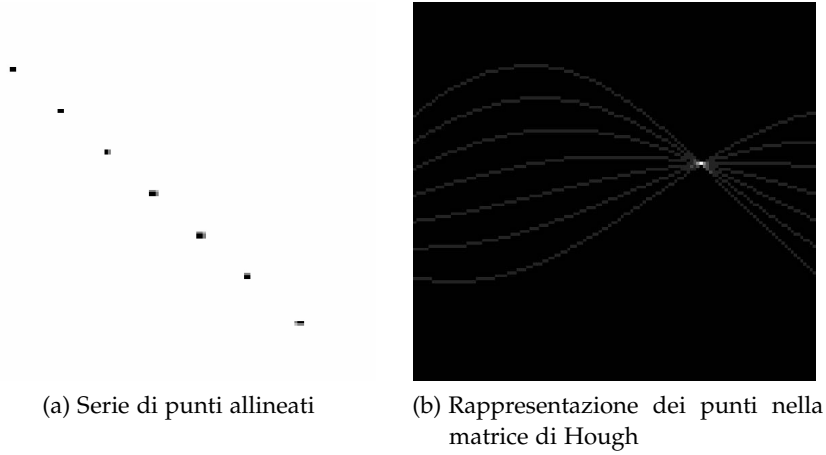


Figura 37: Rappresentazione di una serie di punti allineati con la trasformata di Hough

3.7 INDIVIDUAZIONE DEL TARGET FISSO NELLO SCENARIO

L'applicazione della trasformata di Hough per il riconoscimento delle linee ha permesso di effettuare il "tracking" del target fisso costituito dalla croce presente all'interno dello scenario. Anche se in prima impressione il tracking di un target fisso può sembrare un'applicazione di scarsa utilità in questo caso non lo è affatto perchè permette di dimostrare l'effettiva possibilità di compensare in tempo reale il movimento di roll sia per quanto riguarda la rotazione indotta sull'immagine dal movimento della base sia per quanto riguarda il movimento lungo l'asse x dell'immagine dovuto al fatto che il centro della camera non corrisponde al centro di rotazione del sistema.

Una volta individuate le linee, tramite la trasformata di Hough, e averle tracciate sopra l'immagine reale si è calcolato il punto di incontro di tali rette individuate e tale punto costituisce di fatto il target fisso da cui si vuole effettuare il monitoraggio. Il procedimento per calcolare tale punto è descritto qui di seguito.

Per prima cosa bisogna esprimere le due rette oggetto dello studio in coordinate polari e creare il relativo sistema.

$$\begin{cases} \rho_1 = x \cos(\theta_1) + y \sin(\theta_1); \\ \rho_2 = x \cos(\theta_2) + y \sin(\theta_2); \end{cases} \quad (28)$$

dove $(\rho_1, \theta_1, \rho_2, \theta_2)$ sono le coordinate polari delle due rette individuate con ovvio significato dei simboli. Passando a una notazione matriciale otteniamo:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (29)$$

Prima di procedere ricordiamo il processo di inversione di una matrice quadrata d'ordine 2. Consideriamo la seguente matrice A:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (30)$$

Applicando la formula di inversione della matrice applicato al caso della matrice d'ordine 2 si ottiene:

$$A^{-1} = \frac{1}{\det(A)} * \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad (31)$$

Applichiamo ora tale formula al nostro sistema scritto in notazione matriciale ed espresso in (29) per ricavare le coordinate (x,y) del punto d'incontro.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \end{bmatrix}^{-1} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \quad (32)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{\cos(\theta_1) \sin(\theta_2) - \sin(\theta_1) \cos(\theta_2)} \begin{bmatrix} \sin(\theta_2) & -\cos(\theta_2) \\ -\sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \end{bmatrix} \quad (33)$$

da cui si possono ricavare le coordinate, x e y cercate:

$$\begin{aligned} x &= \frac{1}{\cos(\theta_1) \sin(\theta_2) - \sin(\theta_1) \cos(\theta_2)} (\rho_1 \sin(\theta_2) - \rho_2 \cos(\theta_2)) \\ y &= \frac{1}{\cos(\theta_1) \sin(\theta_2) - \sin(\theta_1) \cos(\theta_2)} (\rho_2 \cos(\theta_1) - \rho_1 \sin(\theta_1)) \end{aligned} \quad (34)$$

Tali coordinate, espresse in pixel, sono riferite al SdR rappresentato in precedenza. Esse vengono inviate tramite UDP al sistema di controllo implementato tramite software Matlab/Simulink in un altro PC.

L'immagine sotto studio ha dimensioni 720x576 e quindi i valori delle coordinate inviate saranno compresi entro questi due valori.

3.8 IL FILMATO DIGITALE

Un filmato, o un video, è una sequenza di immagini digitali riprodotte ad una velocità sufficientemente alta da fornire, all'occhio umano, l'illusione del movimento. La frequenza di riproduzione dei fotogrammi (*frame rate*) viene misurata in hertz (Hz), nei monitor a scansione progressiva, oppure viene semplicemente espressa in termini di

fotogrammi per secondo (fps).

Il frame rate minimo di un filmato, affinché l'apparato visivo umano non percepisca sfarfallii o alterazioni del movimento, si attesta sui 30 fps (o 30 Hz), ma può anche essere superiore per gli oggetti che si muovono molto veloci sullo schermo. Detto ciò, è facile intuire, che l'operazione di video-processing equivale di fatto ad elaborazioni su immagini digitali (o di image-processing) dove ogni immagine è costituita da un preciso frame del video in esame. Pertanto quando si acquisisce un video con lo scopo di elaborarlo in real time è importante evitare qualunque algoritmo di compressione dell'immagine che comporterebbe una perdita di informazioni video.

3.9 ACQUISIZIONE VIDEO

Per l'acquisizione dei video si è utilizzata la videocamera di serie interna al brandeggio riportata in Figura 38.



Figura 38: Videocamera interna al brandeggio Ulisse Compact: Sony FCB-EX480CP

La videocamera a colori ha uno zoom ottico fino a 18x e un sensore di visione CCD a 380.000 pixel in formato PAL. Il controllo dello zoom avviene manualmente tramite la tastiera associata al brandeggio, mentre per quanto riguarda l'iris e il focus esiste la possibilità di una gestione automatica svolta dalla scheda body dell'Ulisse Compact.

L'alimentazione alla videocamera viene fornita attraverso le schede elettroniche interne al brandeggio, le quali gestiscono le immagini acquisite e le rendono disponibili in uscita. Il segnale video ottenuto è di tipo composito in formato PAL con frequenza video di 25 frame al secondo.

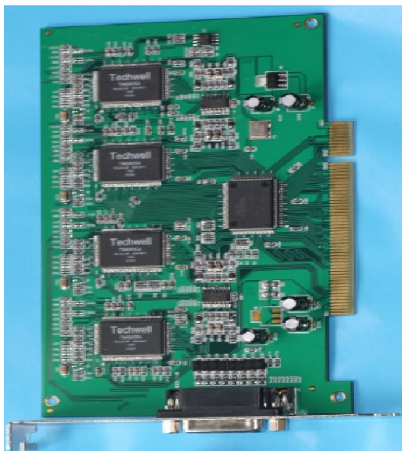
Il segnale video, soprattutto se a componenti (es.RGB), ha una larghezza di banda molto ampia, comportando tutta una serie di problemi per essere registrato o trasmesso. Di conseguenza, sono spesso usate tecniche di compressione per migliorare la gestione del segnale, aumentare la durata delle registrazione oppure aumentare il numero di canali di trasmissione. Dal momento che la visione umana è

molto più sensibile alle variazioni di livello luminoso piuttosto che ai cambiamenti di colore, nel caso di formato standard YUV, si può sfruttare questo principio per ottimizzare la compressione, dedicando più banda alla luminanza (Y) e meno alla differenza delle componenti cromatiche (U,V).

Il video viene acquisito tramite framegrabber collegato alla telecamera e collocato sul PC in cui viene effettuata l'elaborazione in real-time dell'immagine. Il framegrabber in questione è costituito dalla scheda LMLTW44 prodotta da Linux Media Labs (<http://www.linuxmedialabs.com/>). Tale scheda PCI basa il suo funzionamento sull'utilizzo del chip TW6805² il quale si occupa di catturare l'immagine in ingresso e di renderla disponibile per l'elaborazione.

Le caratteristiche principali del framegrabber utilizzato e riportato in Figura 39a sono:

- campionatura video tramite standard YUV 4:2:2 e secondo i seguenti formati:
 - NTSC 640x480 60 fps
 - PAL 758x576 50 fps
- 4 full rate or 8 multiplexed composite video input



(a) Scheda PCI dotata di framegrabber



(b) Cavi per interfacciamento tra scheda PCI e Videocamera

Figura 39: Framegrabber utilizzato e relativi cavi di interfacciamento con videocamera

Il vantaggio di utilizzare tale scheda sta nel fatto che le sue funzionalità sono tutte completamente supportate dal sistema operativo Linux e i drivers relativi alla scheda sono di facile installazione.

L'uscita video della telecamera inserita all'interno del brandeggio è

² Per maggiori informazioni sul chip utilizzato dalla scheda PCI installata per catturare le immagini consultare [14]

stata collegata al primo canale in ingresso alla scheda PCI ed è stato così possibile catturare le immagini in tempo reale e registrare gli opportuni video tramite il software VLC prodotto da VIDEO-LAN organization e liberamente scaricabile dal sito internet <http://www.videolan.org/>.

3.10 SOCKET PER L'INVIO E LA RICEZIONE DEI DATI

Come anticipato nell'introduzione di questo capitolo, all'interno della nostra applicazione si è reso necessario l'utilizzo di un'apposita struttura che sia in grado di gestire un flusso di dati bidirezionale tra i due PC che effettuano rispettivamente il controllo del brandeggio Ulisse e l'elaborazione in real time dell'immagine.

I sockets sono un *modo* per comunicare con altri programmi, usando dei files *descrittori* standard di UNIX. Quando un programma sotto UNIX fa un qualsiasi lavoro di I/O, lo fa leggendo o scrivendo su un file detto *descrittore* (file descriptor). Questo file può essere una connessione via internet, un terminale, un file su disco o una pipe. Quindi, quando vogliamo comunicare con altri programmi, anche attraverso la rete, lo facciamo attraverso un *file descriptor*. Per comunicare attraverso questo tipo di file bisogna fare una chiamata a sistema e tale chiamata è proprio la `socket()`. Tale chiamata restituisce il *socket descriptor* e noi comunicheremo attraverso esso tramite le funzioni di invio e ricezione.

Ci sono due tipi di sockets:

- Datagram sockets : utilizzate per comunicazioni del tipo *connection less* come UDP...(come la nostra)
- Stream sockets: utilizzano protocollo TCP (Trasmission Control Protocol)

Per maggiori informazioni sulla struttura di questi due tipi di sockets consultare [8]. In questa sede ci limitiamo ad elencarle indicando che noi utilizzeremo le Datagram Sockets in quanto abbiamo instaurato una connessione UDP tra i due host remoti.

La struttura che contiene tutti i dati necessari per impostare la comunicazione è una struttura del tipo `sockaddr_in` ed all'inizio della procedura è necessario riempire tutti i campi della struttura suddetta con i valori opportuni come l'indirizzo IP della macchina di destinazione, la porta di destinazione ed altri valori necessari per la corretta comunicazione. Una volta inizializzata correttamente questa struttura si è pronti per instaurare la comunicazione voluta tramite la chiamata alla funzione `socket()` e alla successiva assegnazione delle socket create alle porte opportune (tramite la chiamata alla funzione `bind()`) già indicate all'interno delle strutture precedentemente inizializzate.

Per l'invio dei dati è stata utilizzata la funzione:

```
int sendto() ( int sockfd, const void *msg, int len, unsigned int flags,
               const struct sockaddr *to, int tolen)
```

dove:

- **sockfd** è il file descrittore restituito dalla chiamata alla `socket()` ed utilizzato per la connessione;
- **msg** è un puntatore ai dati che vogliamo mandare;
- **len** è la lunghezza dei dati, in byte;
- **flags** viene posto a 0;
- **to** è un puntatore a una struttura caratteristica delle sockets e contiene l'indirizzo IP e la porta di destinazione;
- **tolen** può essere posto a 0;

Per la ricezione si è utilizzata la funzione:

```
int recvfrom()( int sockfd, void *buf, int len, unsigned int flags, struct
                 sockaddr *from, int *fromlen)
```

dove:

- **sockfd**, **len** e **flags** hanno lo stesso significato assunto nella funzione precedente;
- **buf** è il buffer da cui leggere le informazioni ricevute;
- **from** è un puntatore ad una struttura tipica delle sockets e conterrà l'indirizzo IP e la porta della macchina d'origine
- **fromlen** è un puntatore a una struttura tipica delle sockets e conterrà la lunghezza dell'indirizzo messo in `from`.

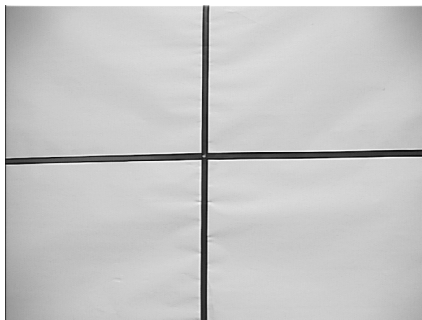
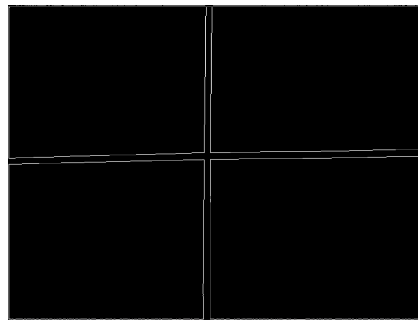
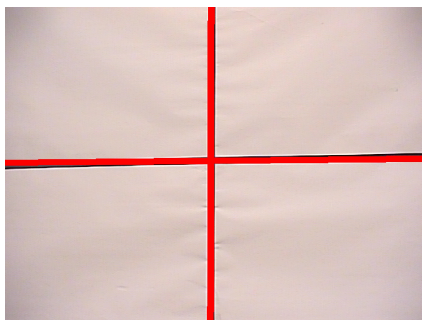
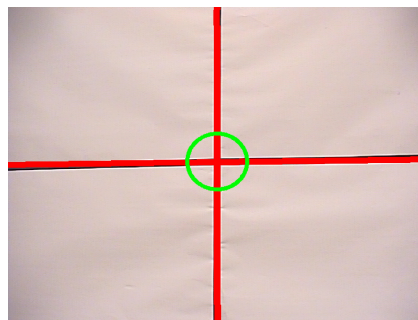
La parte di codice relativa alle sockets è inserita all'interno del codice scritto in C++ sia per l'applicazione relativa al target fisso sia per quella relativa al target mobile e sono entrambe riportate in Appendice.

3.11 PRINCIPALI FUNZIONI *opencv* UTILIZZATE

Utilizzando le nozioni espresse nei paragrafi precedenti si è sviluppata una routine in C++ che ci permette di raggiungere gli scopi preposti. Vogliamo ora illustrare le principali funzioni di OpenCv utilizzate analizzando il compito da loro svolto, i parametri richiesti in ingresso e ciò che danno in uscita. Le funzioni sono analizzate in ordine di esecuzione nel programma. Il codice completo sviluppato è riportato in appendice [A.1](#).

- ***void cvtColor*** (*const CvArr* src, CvArr* dst, int code*)
 Converte l'immagine iniziale in un'immagine di diverso formato (es. conversione da immagine a colori a immagine in scala di grigi). I parametri hanno il seguente significato:
 - *src* è un puntatore a una struttura contenente l'immagine originale
 - *dst* è un puntatore a una struttura contenente l'immagine di destinazione
 - *code* contiene il codice per convertire l'immagine nel formato desiderato
- ***void cvCanny*** (*const CvArr* img, CvArr* edges, double lowThresh, double highThresh, int apertureSize = 3*)
 Applica l'operatore di Canny per la ricerca dei contorni all'immagine in ingresso. Questa funzione si aspetta in ingresso un'immagine in scala di grigi. I parametri sono:
 - *img* è un puntatore a una struttura contenente l'immagine originale
 - *edges* è un puntatore a una struttura contenente l'immagine elaborata
 - *lowThresh, highThresh* soglie utilizzate dall'algoritmo di Canny
 - *apertureSize* valore utilizzato dall'operatore di Sobel interno all'algoritmo di Canny
- ***void HoughLines*** (*InputArray image, OutputArray lines, double rho, double theta, int threshold*)
 Esegue trasformata di Hough sull'immagine in ingresso e restituisce un array bidimensionale contenente le coordinate polari delle rette individuate. Il significato dei parametri è:
 - *image* contiene l'immagine a cui viene applicata la trasformata
 - *lines* vettore bidimensionale contenente coordinate polari delle rette individuate
 - *rho* risoluzione in pixel della coordinata polare *rho*
 - *theta* risoluzione in pixel della coordinata polare *theta*
 - *threshold* soglia sotto la quale i valori della matrice di accumulazione non vengono considerati
- ***void cvLine*** (*CvArr* img, CvPoint pt1, CvPoint pt2, CvScalar color*)
 Permette di tracciare linee di un certo colore nell'immagine di partenza dati due punti. I parametri hanno il seguente significato:
 - *image* immagine a cui viene disegnata la linea
 - *pt1, pt2* punti d'inizio e di fine della linea

- *color* colore in cui viene tracciata la linea
- *Mat getRotationMatrix2D (Point2f center, double angle, double scale)*
Restituisce una matrice di rotazione dato un determinato angolo di cui si vuole ruotare l'immagine. Il significato dei parametri è:
 - *center* centro dell'immagine da ruotare
 - *angle* angolo di cui si vuole ruotare l'immagine
 - *scale* fattore di scala
- *void warpAffine (InputArray src, OutputArray dst, InputArray M, Size dsize)*
Applica una trasformazione affine all'immagine iniziale con il seguente valore dei parametri
 - *src* immagine iniziale a cui applicare la trasformazione
 - *dst* immagine ruotata
 - *M* matrice di rotazione
 - *dsize* dimensioni dell'immagine iniziale

(a) 1° passo: *cvtColor()*(b) 2° passo: *cvCanny()*(c) 3° passo: *HoughLines()*

(d) 4° passo: Tracciamento cerchio con centro nel punto di incontro delle rette individuate

Figura 40: Fasi di elaborazione dell'immagine.

In Figura 40 sono rappresentate le varie fasi per l'elaborazione delle immagini qui riassunte:

1. Trasformiamo l'immagine originale in un'immagine in scala di grigi, Figura 40a;
2. Applichiamo l'operatore di Canny per la rilevazione dei contorni, Figura 40b;
3. Applichiamo algoritmo di riconoscimento delle linee, Figura 40c;
4. Calcoliamo il punto di incontro delle linee identificate, Figura 40d.

SISTEMA DI CONTROLLO

4.1 INTRODUZIONE

L'obiettivo di questo capitolo è quello di illustrare il sistema di controllo (SdC) realizzato al fine di compensare i movimenti di yaw, pitch e roll della base sulla quale è collocato il brandeggio Ulisse. Il controllo sviluppato consente di poter inseguire con la telecamera un target, fisso o in movimento, e di mantenerlo sempre posizionato al centro del piano immagine della telecamera stessa al fine di poterne monitorare il comportamento.

La compensazione dell'immagine si basa sulle misure degli spostamenti ricavati tramite un sensore di misura inerziale (IMU) collocato sulla parte superiore del brandeggio.

Tale compensazione viene fatta sia a livello hardware, ruotando opportunamente i motori di pan e di tilt comandati da un segnale prodotto da un SdC che riceve in ingresso le misure di yaw e di pitch, sia a livello software. Quest'ultimo componente si occupa di effettuare la rotazione dell'immagine per consentire la compensazione del movimento di roll ed inoltre permette l'individuazione e la selezione del target di cui si vuole monitorare il comportamento e fornisce in uscita i dati necessari per effettuarne il tracking.

Il capitolo inizia con la descrizione e l'analisi delle funzioni di trasferimento dei motori di pan e di tilt determinate, per poi passare nella parte centrale alla descrizione accurata del SdC sviluppato. Nel capitolo successivo vengono illustrati ed analizzati i dati sperimentali ottenuti dalle prove effettuate in laboratorio utilizzando il simulatore di moto ondoso presentato nei capitoli precedenti.

4.2 FUNZIONI DI TRASFERIMENTO DEI MOTORI

Prima di impostare la legge di controllo da applicare è indispensabile individuare le funzioni di trasferimento dei motori di pan e di tilt che legano i movimenti dei motori con i rispettivi segnali di controllo.

Nel nostro caso, il sistema da modellizzare è difficilmente rappresentabile tramite delle leggi fisiche poichè non siamo a conoscenza di molti parametri che influenzano la dinamica dei motori come le forze d'attrito e le inerzie dei motori. Per determinare le funzioni di trasferimento cercate è necessario disporre una procedura di identificazione non-parametrica: tale metodo consiste nell'applicare un *opportuno* se-

gnale di riferimento in ingresso al sistema e misurarne le uscite ¹. Con il termine *opportuno* si intende un segnale in grado di eccitare tutte le dinamiche del sistema e di non risentire di alcune non linearità presenti, per offrire ad esso una gran quantità di informazioni a qualunque frequenza.

Le funzioni di trasferimento, rappresentate tramite il loro diagramma di Bode di modulo e fase, relative ai due motori di PAN e di TILT sono riportate in Figura 41; è di fondamentale importanza osservare che la procedura di identificazione è stata effettuata mediante l'utilizzo di un accelerometro collocato sulla testa del brandeggio per rilevare le accelerazioni nelle varie direzioni e quindi le funzioni di trasferimento riportate rappresentano il legame tra l'accelerazione in uscita e il segnale di posizione applicato in ingresso.

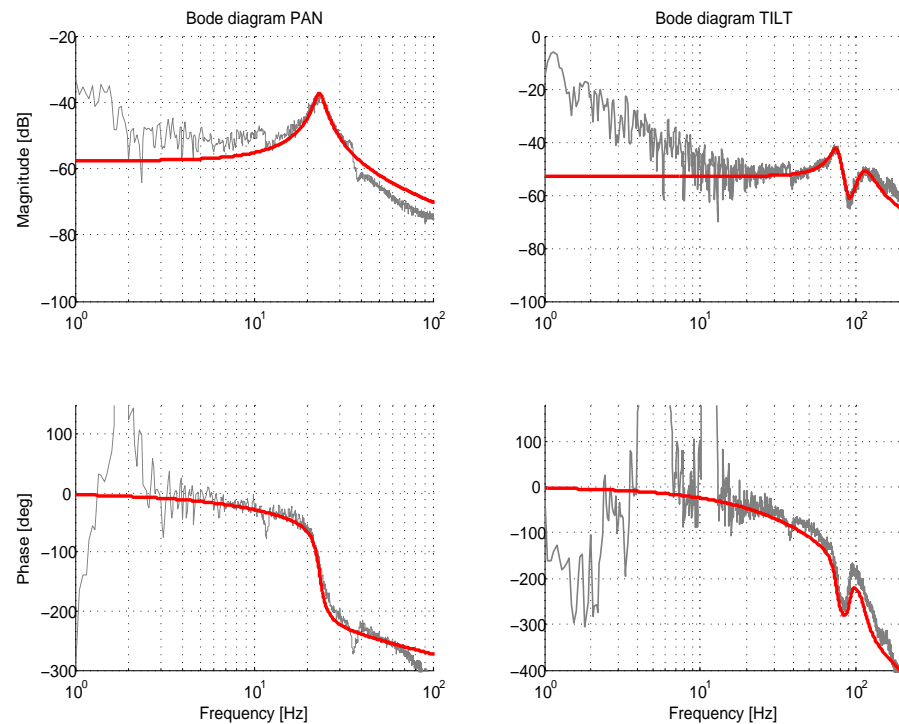


Figura 41: Funzioni di trasferimento dei motori di PAN e di TILT individuate

Nella parte sinistra viene rappresentata la fdt del motore di PAN; in essa si può facilmente osservare la presenza di un doppio polo riso-
nante ad una frequenza di circa 23 Hz e di due coppie di poli-zeri
risonanti prima e dopo il doppio polo stesso, rispettivamente a cir-
ca 11 Hz e circa 36 Hz. Il sistema viene quindi modellizzato tramite
un'operazione di fitting², ipotizzando un controllo in bassa frequen-

¹ Per maggiori informazioni sulla procedura di identificazione parametrica utilizzata consultare [11]

² Il fitting viene realizzato tramite un'opportuna funzione di matlab, `invfreq`, che parametrizzata correttamente permette di ottenere una funzione di trasferimento con le caratteristiche volute

za³, come un sistema avente solamente un doppio polo risonante a 23 Hz e tralasciando i contributi in alta e bassa frequenza. Il guadagno statico dell'anello di PAN è circa di -57dB .

Nella parte destra viene riportata la funzione di trasferimento relativa all'asse di TILT. In questo caso si nota la presenza di un doppio polo-zero risonante attorno agli 80 Hz e di un doppio polo a circa 115 Hz. Assumendo, come nel caso dell'anello di PAN, di realizzare un controllo in bassa frequenza si è tenuto conto solamente di questi due contributi tralasciando gli altri collocati a frequenze più elevate. In questo caso il guadagno statico risulta pari a -52dB .

4.3 STRUTTURA DEL CONTROLLO

Lo schema generale del sistema di controllo realizzato è riportato in Figura 42. Per raggiungere gli scopi preposti ad ogni motore passo-passo è applicata una struttura di controllo composta da due anelli di regolazione: uno più esterno che si occupa della stabilizzazione video ed uno interno che effettua la stabilizzazione inerziale.

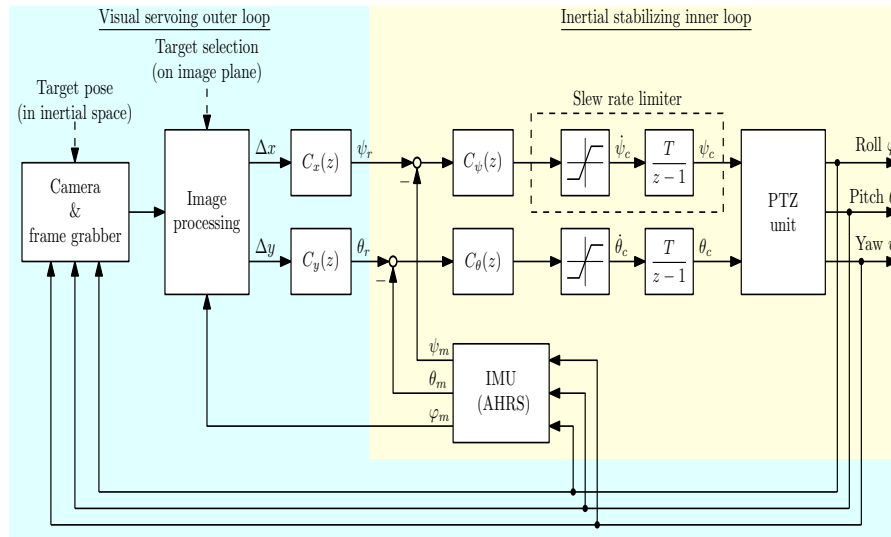


Figura 42: Schema a blocchi del sistema di controllo in generale.

Quando un eventuale oggetto collocato nello spazio non è presente all'interno del piano immagine oppure, se presente, l'operatore non lo ritiene di suo interesse, la stabilizzazione inerziale avviene rispetto alla posizione iniziale della piattaforma mantenendo il LOS della camera costante mentre l'anello di controllo video fornisce al sistema di controllo inerziale dei riferimenti nulli pur effettuando la compensazione del roll grazie alla lettura di tale movimento tramite la IMU. Una volta che il target è stato individuato, l'anello esterno elabora

³ Poichè la banda passante del sensore IMU è pari a 50 Hz, il controllo potrà avere banda passante di qualche Herz

l'immagine e fornisce all'anello interno dei segnali di controllo diversi da zero che consentono di mantenere il target al centro del piano immagine nonostante i suoi movimenti e i movimenti della base su cui poggia il brandeggio.

Nei successivi paragrafi saranno illustrati approfonditamente i due regolatori posti in cascata, quello esterno (anello di controllo video) e quello più interno (anello di stabilizzazione inerziale), mettendone in risalto le principali caratteristiche.

4.3.1 Anello di stabilizzazione inerziale

Come abbiamo preannunciato nell'introduzione, lo scopo dell'anello di stabilizzazione inerziale è di mantenere il LOS della camera costante, rispetto ad un certo riferimento, nonostante i movimenti della base su cui è collocato il brandeggio.

Come si può notare dallo schema generale di Figura 42 i segnali in ingresso all'anello di stabilizzazione interno sono:

- angoli di yaw e pitch rilevati tramite la IMU;
- angoli di riferimento di yaw e di pitch prodotto dall'anello di controllo video esterno, il quale elabora dei segnali in modo da poter inseguire un determinato obiettivo.

I segnali di errore di yaw e di pitch, ricavati dalla differenza tra il segnale di riferimento e le misure della IMU, costituiscono gli ingressi di due anelli di controllo, rispettivamente di PAN e di TILT, totalmente indipendenti tra loro in quanto non sono presenti fenomeni di mutuo accoppiamento tra gli assi dei due motori presenti. Come anticipato nell'introduzione al capitolo, attraverso questi due anelli di controllo è possibile compensare solamente i movimenti di yaw e di pitch poiché i gradi di libertà del brandeggio sono due e la compensazione del roll deve dunque avvenire a livello software.

Nella progettazione dei due anelli di controllo inerziale è necessario tenere presente che i motori passo-passo presenti all'interno del brandeggio sono attuatori comandati in posizione e quindi i segnali di comando devono avere velocità di variazione limitate per evitare fenomeni di perdita di passo (*step-losses*) da parte dell'azionamento, in grado cioè di mantenere il punto di funzionamento del motore all'interno della propria curva di pull-out.

Questa specifica può essere facilmente soddisfatta inserendo nell'anello di controllo un integratore con ingresso saturato (*slew-rate limiter*) per generare i segnali di riferimento per i motori. In Figura 42 si può notare la presenza di questo accorgimento nei due anelli di controllo, evidenziato dal riquadro tratteggiato.

Dopo tale accorgimento la progettazione dei due sistemi di controllo può avvenire utilizzando metodi convenzionali come quello di Bode,

tenendo conto, però, che il sistema da controllare ora presenta un integratore aggiuntivo all'interno della funzione di trasferimento e le funzioni di trasferimento dei due motori sono quelle riportate in Figura 41. Aggiungendo a tali funzioni l'integratore dovuto allo slew rate limiter si ottengono le funzioni riportate in Figura 43 che rappresentano le funzioni di trasferimento da cui partire per effettuare la sintesi del controllore.

La scelta della banda passante da imporre ai due anelli di controllo deve essere tale da permettere la compensazione della maggior parte delle componenti in frequenza che costituiscono un moto onduoso.

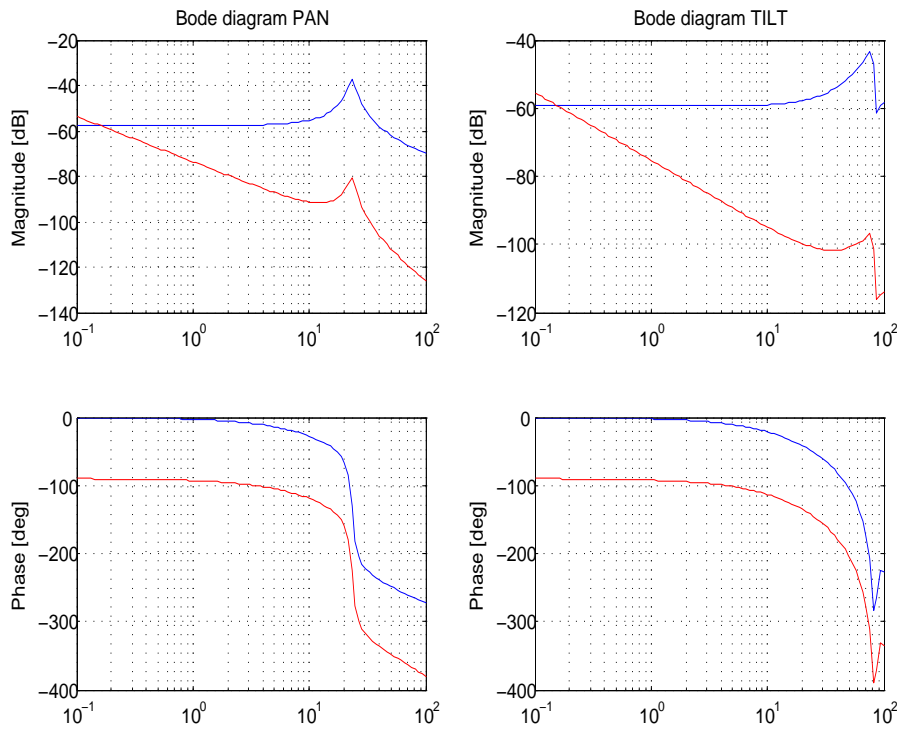


Figura 43: Funzioni di trasferimento dei motori (linea blu) e con relativa aggiunta dell'integratore dovuto alla presenza dello slew-rate limiter (linea rossa).

Analizzando i dati scientifici utilizzati al fine di simulare il moto onduoso mediante l'applicazione di un opportuno segnale di movimentazione alla piattaforma di Stewart [15], si è osservato che la densità spettrale di tale segnale presenta un unico picco nell'intorno di 0.2 Hz, come evidenziato in Figura 44.

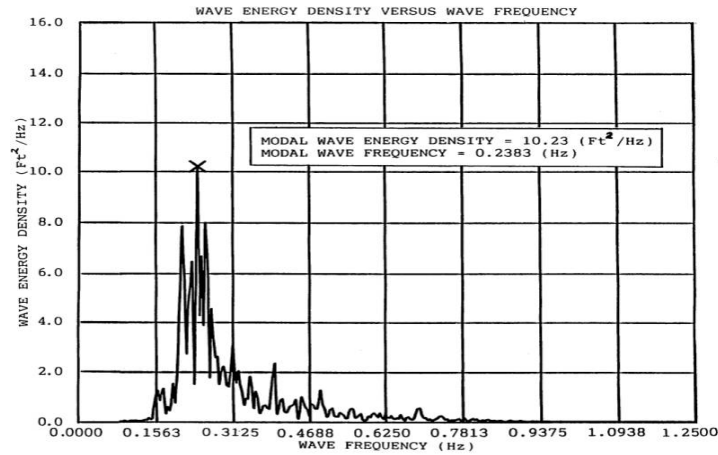


Figura 44: Densità spettrale del movimento ondoso analizzato

Risulta dunque opportuno, andare a realizzare degli anelli di controllo che presentino una banda passante, se possibile, superiore alla frequenza in cui si ha il picco di densità spettrale in modo tale da poter compensare le componenti con maggiore energia.

La scelta delle specifiche che il sistema deve essere in grado di soddisfare è stata fatta non solo sulla base della densità spettrale di energia delle onde, ma anche sulla base delle prestazioni che si potevano ottenere dai controlli. A tal fine si sono scelte le seguenti specifiche:

- Anello di PAN
 - Margine di fase 80°
 - Frequenza di attraversamento $\omega_{gc} = 1\text{Hz}$
- Anello di TILT
 - Margine di fase 80°
 - Frequenza di attraversamento $\omega_{gc} = 0.2\text{Hz}$

Si capisce subito dunque come un semplice regolatore PI di posizione sia sufficiente per soddisfare le specifiche richieste. I guadagni dei regolatori sono stati ottenuti impostando l'attraversamento a 0dB, alla frequenza prestabilita, della funzione di anello $L(s)$ ed imponendo il margine di fase richiesto in entrambi i casi. Le funzioni di trasferimento comprensive del controllore progettato e del sistema da controllare sono riportate in Figura 45.

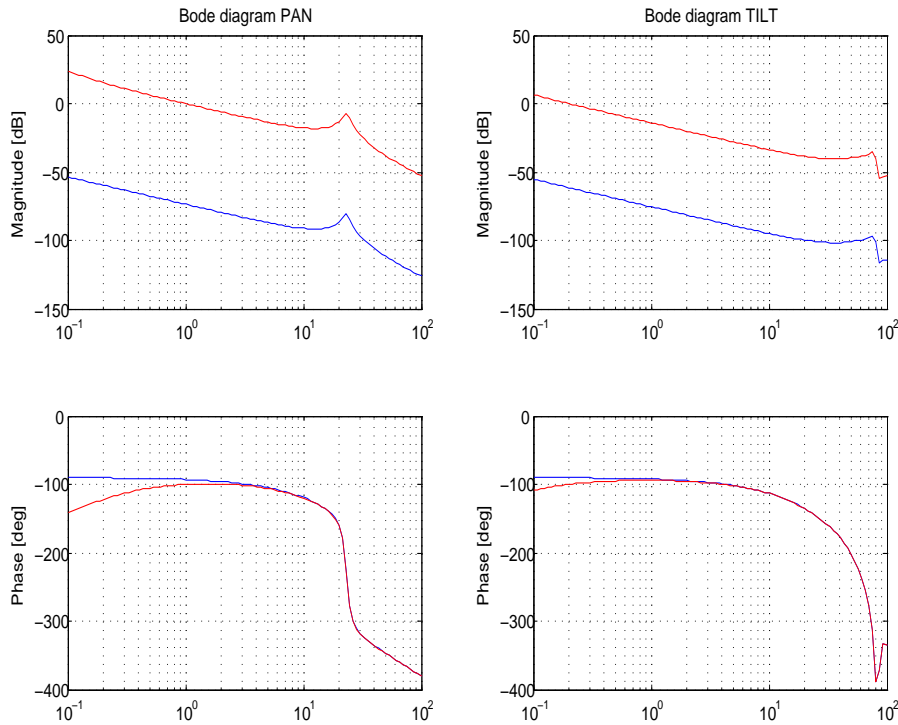


Figura 45: Confronto tra il diagramma di Bode del sistema da controllare (in blu) e il diagramma di Bode della funzione di anello compresa di controllore (in rosso)

4.3.2 Anello di controllo video

L'anello di stabilizzazione inerziale descritto in precedenza ha la funzione di mantenere gli angoli di pan e di tilt della telecamera costanti rispetto a dei determinati riferimenti nello spazio inerziale. Modificando opportunamente tali riferimenti è possibile variare l'orientamento della telecamera al fine di monitorare il comportamento di un obiettivo specifico presente sul piano immagine ed allo stesso tempo disaccoppiare totalmente il piano immagine della telecamera stessa dai movimenti della base su cui poggia il brandeggio. Sebbene il compito di monitoraggio dell'obiettivo potrebbe idealmente essere svolto anche su un dispositivo non stabilizzato inerzialmente, è evidente come il fatto di avere un anello interno di stabilizzazione inerziale agevola notevolmente il compito. Difatti anche se l'obiettivo da monitorare fosse fermo, senza l'anello di stabilizzazione inerziale avremo un moto apparente sul piano immagine dovuto ai movimenti del supporto. Il compito dell'anello interno è quindi quello di eliminare tale movimento apparente e permettere all'anello esterno di concentrarsi solamente sui movimenti effettivi del target.

I segnali di riferimento per l'anello interno di stabilizzazione inerziale possono essere forniti da un operatore attraverso un'opportuna interfaccia oppure da un anello di controllo video esterno (*visual servoing*

outer loop) che si occupa di elaborare l'immagine in real-time. L'anello di controllo è stato progettato predisponendo un algoritmo di elaborazione delle immagini basato sui concetti espressi nel capitolo 3; tale algoritmo viene utilizzato per rilevare le coordinate di un determinato target rispetto al centro del piano immagine della camera. Poiché l'anello di stabilizzazione inerziale non è in grado di contrastare le rotazioni dell'immagine attorno all'asse ottico della camera dovute al movimento del supporto, tale stabilizzazione viene effettuata tramite tale algoritmo utilizzando la misura dell'angolo di roll fornito dalla IMU ed applicando una rotazione all'immagine di ugual modulo ma verso opposto.

Le coordinate prodotte in uscita dall'algoritmo, vengono fornite a due regolatori PI, $C_x(z)$ e $C_y(z)$ visibili in Figura 42, che generano rispettivamente i riferimenti di PAN e di TILT, per l'anello di stabilizzazione inerziale al fine di mantenere il centro del campo visivo della camera allineato con il bersaglio.

La sintonizzazione dei due controllori è stata fatta manualmente valutando la risposta ad una variazione a gradino della posizione del bersaglio sul piano immagine e tarando i parametri dei controllori in modo da avere una risposta al gradino del sistema complessivo equiparabile con quella che si ha con il solo controllo video attivato. In Figura 46 è riportato un esempio dell'andamento della coordinata x di un oggetto al seguito di una variazione a gradino della sua posizione in corrispondenza di due valori dei parametri dei controllori, di cui uno corretto e l'altro no.

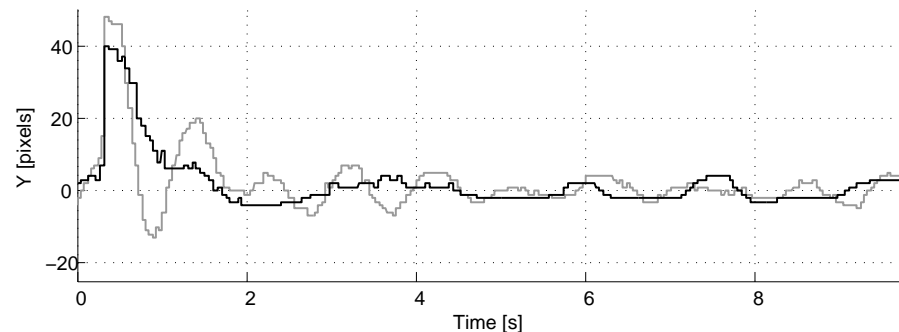


Figura 46: Andamento della risposta a gradino nel sistema nel caso di controllore tarato con i valori corretti (linea nera) e con i valori sbagliati (linea grigia)

È importante notare come l'operazione di sintonizzazione effettuata dipenda molto pesantemente dalla distanza dell'obiettivo dalla fotocamera e dall'eventuale zoom inserito. I parametri del controllo devono essere chiaramente variati in seguito alla variazione di una di queste due componenti, per ottenere una risposta al gradino significativa in qualsiasi condizione operativa.

4.4 REALIZZAZIONE DEL SISTEMA DI CONTROLLO

Per permettere la realizzazione del sistema di controllo appena descritto si è reso necessario risolvere due problematiche:

- sincronizzazione della comunicazione seriale per il trasferimento delle informazioni della scheda IMU al calcolatore;
- invio e ricezione dei dati tra i due calcolatori che gestiscono rispettivamente l'elaborazione dell'immagine e il controllo dei motori;

4.4.1 *Sincronizzazione nella comunicazione seriale*

Le informazioni riguardanti l'orientazione della scheda IMU e, di conseguenza della videocamera, vengono trasferite al calcolatore attraverso un protocollo di trasmissione seriale asincrono. Il firmware presente all'interno della scheda è stato configurato in modo tale che, non appena i dati sono pronti, una successione di misure (yaw, pitch e roll) venga inviata al computer.

La trasmissione seriale è una modalità di comunicazione tra dispositivi digitali nella quale le informazioni sono comunicate una di seguito all'altra e giungono sequenzialmente al ricevente nello stesso ordine in cui le ha trasmesse il mittente. I vantaggi della comunicazione seriale rispetto alla comunicazione parallela sono:

- necessità di un minor numero di fili con conseguente riduzione dei costi;
- tolleranza maggiore degli errori di trasmissione.

Lo svantaggio di avere una minor complessità circuitale si ripercuote in una maggiore difficoltà di gestione del flusso dei dati. In particolare nella trasmissione seriale asincrona (dove non è presente un filo per il segnale di clock che permette il sincronismo delle operazioni di invio e ricezione) la sincronizzazione avviene usando i dati trasmessi. In questo caso il trasmettitore, cioè la scheda IMU, invia una sequenza particolare (preambolo) e predefinita di bit di *partenza* seguita dalla sequenza di dati vera e propria utile ai fini del controllo. Tale costituzione della sequenza di invio permette di gestire le sequenze di dati errate come tali, come ad esempio nel caso in cui avvengano alcune interruzioni della linea di comunicazione che non consentono una corretta sincronizzazione.

Nella nostra applicazione il preambolo posto prima di ogni sequenza di dati utili di yaw, pitch e roll è costituito dalla sequenza di bit `0xFF00FF00`. La sequenza di dati significativi è invece impostata nel seguente modo:

\$ <yaw> <pitch> <roll> #

dove le varie misure angolari sono costituite da una word.

Nell'ambiente software Simulink è presente un blocchetto funzionale adibito alla lettura dei dati proveniente da una porta seriale. Tuttavia per implementare tale tipo di comunicazione si è reso necessario utilizzare una S-function che consenta di identificare dapprima il preambolo e successivamente la struttura dei dati da analizzare.

Tale blocchetto implementa una macchina a stati finiti il cui funzionamento è descritto da una serie di stati i a cui si accede sequenzialmente se la lettura allo stato i -esimo è avvenuta nel modo corretto. Il funzionamento della macchina implementata è illustrato Figura 47 in cui i 5 stati presenti sono:

- **Stato 0:** Lettura del bit 0xFF;
- **Stato 1:** Lettura del bit 0x00;
- **Stato 2:** Lettura del bit 0xFF;
- **Stato 3:** Lettura del bit 0x00;
- **Stato 4:** Lettura dei dati angolari rilevati dalla IMU.

Una volta terminata la lettura si ritorna allo stato 0 e il processo si ripete ciclicamente. Il listato dell'S-Function utilizzata è riportata in appendice A.2.

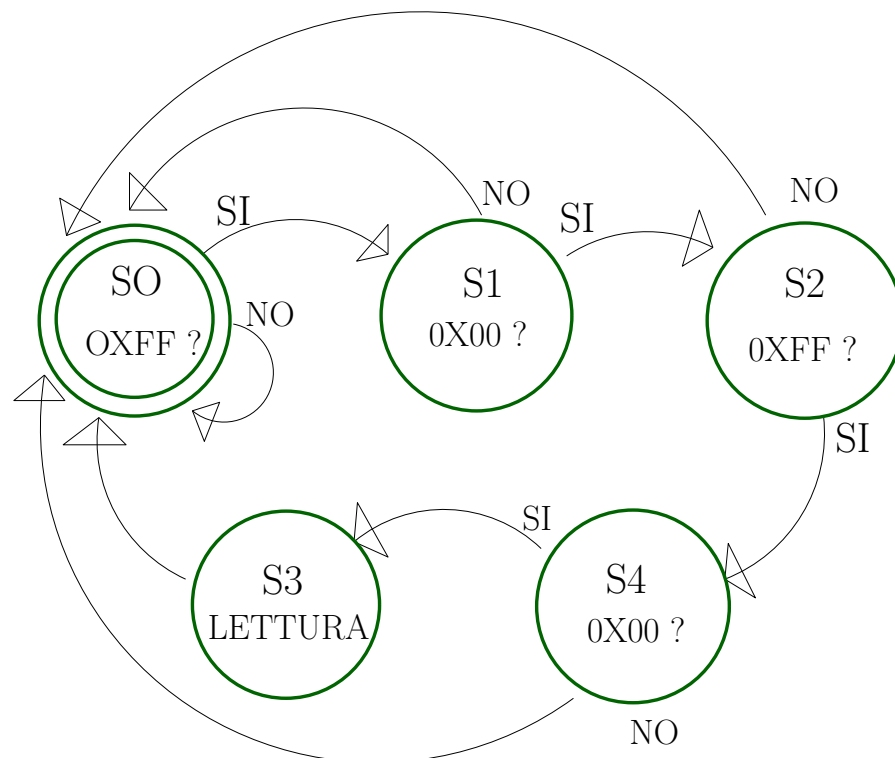


Figura 47: Macchina a stati equivalente per la lettura dei dati inviati dalla IMU

Prima di concludere il paragrafo è necessario osservare come l'introduzione di bit aggiuntivi nella fase di trasmissione riduca la velocità di quest'operazione; tuttavia tale riduzione non influisce sulle prestazioni del sistema in quanto si riesce ad ottenere una frequenza di trasmissione di 40 Hz che è molto maggiore della frequenza con cui vengono acquisiti i frame video, 25 Hz.

4.4.2 Impostazione sistema di invio e ricezione dei dati

Come anticipato nei capitoli precedenti la trasmissione dei dati da un PC all'altro avviene utilizzando il protocollo di comunicazione UDP. La connessione tra i due PC viene realizzata tramite un cavo ethernet crossato come illustrato nel capitolo 2.

Mentre nel PC che gestisce l'elaborazione dell'immagine la comunicazione è gestita tramite la creazione di opportune socket all'interno della routine C++ creata, nel PC che effettua il controllo dei motori si è reso necessario utilizzare i blocchetti messi a disposizione dal software Simulink: *Packet Input* e *Packet Output*. Tali blocchetti sono stati configurati inserendo il tipo di comunicazione voluta (UDP) e gli indirizzi IP del PC stesso e di quello con cui comunicare.

Il sottosistema utilizzato per gestire il flusso di dati in entrata è illustrato in Figura 48; si nota subito che, prima di esser resi disponibili per il sistema di controllo, i dati ricevuti in ingresso subiscono una fase di elaborazione. Tale fase è dovuta al fatto che i dati vengono trasmessi come *char* ed è necessario trasformarli in interi prima di poterli utilizzare. Inoltre, a monte della trasmissione, le coordinate del punto di interesse individuate vengono moltiplicate per 10 in modo da poter trattare tali dati con maggiore semplicità e quindi a valle della trasmissione è necessario dividere i valori ricevuti dello stesso fattore. Nel caso di individuazione di un target fisso costituito da una croce, tale trattamento è riservato anche al terzo valore trasmesso che corrisponde all'inclinazione della retta individuata, utile ai fini della nostra applicazione per dimostrare la compensazione del movimento di roll.

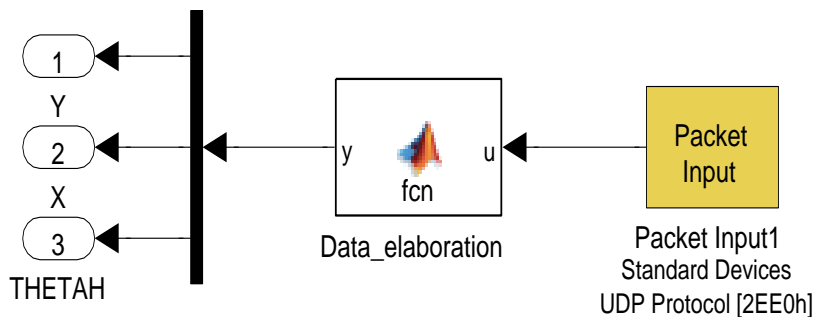


Figura 48: Ricezione dei dati trasmessi via UDP

L'invio del dato relativo al movimento di roll rilevato tramite la IMU avviene tramite il sottosistema illustrato in Figura 49. Prima di essere inviato il dato viene sottoposto ad una preventiva fase di controllo attraverso un opportuno circuito, introdotto dopo aver osservato che il sensore, per quanto riguarda il movimento di roll, produce un segnale molto rumoroso e contenente dei picchi che devono essere necessariamente eliminati o quantomeno ridotti per garantire la corretta compensazione di questo tipo di movimento. Il circuito di controllo inserito permette di eseguire ciò confrontando il valore attuale di roll con il valore precedente e basandosi sul fatto che quando questi due valori differiscono di molto significa che il valore attuale non è corretto e deve essere eliminato mantenendo in memoria il valore precedente ed attendendo un ulteriore valore di misura corretto.

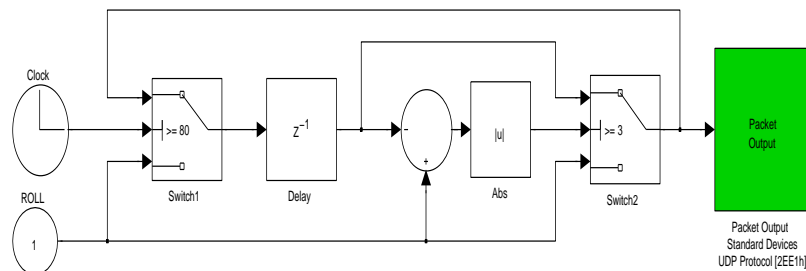


Figura 49: Elaborazione ed invio del dato relativo alla misura di roll

4.5 SCHEMA SIMULINK COMPLETO DEL SISTEMA DI CONTROLLO

Lo schema complessivo del controllo realizzato è riportato in Figura 50; come si può notare i controlli relativi agli assi di PAN e di TILT hanno la medesima struttura e variano solamente i guadagni e alcune costanti moltiplicative.

Ognuno dei due sistemi di controllo rappresentati all'interno della figura stessa è così composto:

- Anello di controllo video (*blocchetto verde*);
- Blocchetto di interfaccia tra anello esterno e quello interno, che di fatto costituisce il regolatore PI esterno;
- Anello di controllo inerziale (*blocchetto arancione*);
- Switch che permette di poter attivare solo l'anello di controllo interno, solo quello esterno oppure tutto il sistema completo;
- Slew rate limiter (*blocchetto giallo*);
- Blocchetti di interfaccia con i driver di comando dei motori (*blocchetti blu*).

La retroazione dell'anello di controllo interno è costituita da un segnale di errore ricavato dalla differenza tra il valore letto dalla IMU ed un segnale di riferimento che corrisponde al valore dell'angolo rilevato all'inizio della simulazione. Ciò consente di effettuare la compensazione inerziale rispetto a qualsiasi posizione di partenza in quanto tale anello di controllo interno ha la funzione di mantenere costante l'orientazione della camera nonostante i movimenti della base. Inoltre nell'anello di retroazione interno sono presenti delle costanti che permettono di convertire il segnale d'errore da gradi in radianti e realizzare una semplice compensazione dovuta a una mancata calibrazione della IMU.

I segnali in uscita dai controllori sono dei riferimenti di posizione espressi in angoli elettrici per i due motori. Tali segnali di riferimento vengono utilizzati per comandare gli attuatori in micro-stepping.

Attivando o disattivando, attraverso lo switch presente, una specifica parte del controllo progettato è stato possibile, in fase di simulazione, notare le differenze di prestazioni nel caso fosse attivato solo il controllo video, solo quello inerziale, oppure tutto il sistema nella sua interezza e valutare quindi i vantaggi che si hanno nell'utilizzo di un sistema di controllo piuttosto che un altro.

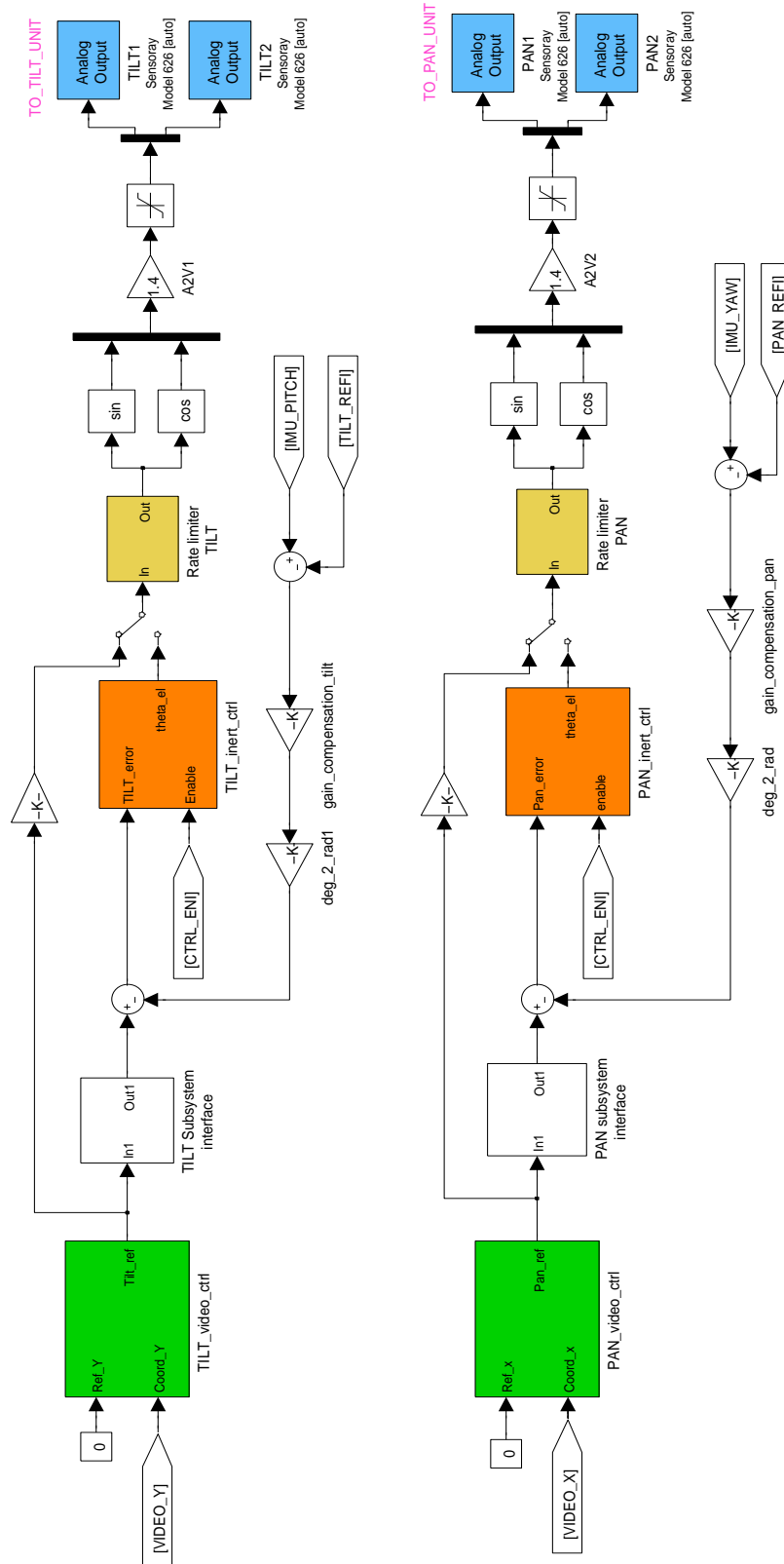


Figura 50: Schema di controllo complessivo

In Figura 51 sono riportati i blocchetti simulink inseriti a fianco dello schema di controllo totale illustrato nella figura precedente. Tali blocchetti hanno le seguenti funzioni:

- gestire l'invio e la ricezione dei dati attraverso connessioni UDP (*blocchetti gialli*);
- gestire i dati provenienti dalla IMU (*blocchetto verde*);
- effettuare la registrazione di tutti i dati ottenuti sincronizzati con il movimento della base attraverso la lettura di un dato valore che viene posto ad 1 non appena inizia la prova da valutare (*blocchetto arancione*)
- registrare i valori letti dalla IMU all'inizio della simulazione per poter effettuare la stabilizzazione inerziale rispetto a tali valori iniziali (*blocchetto azzurro*)

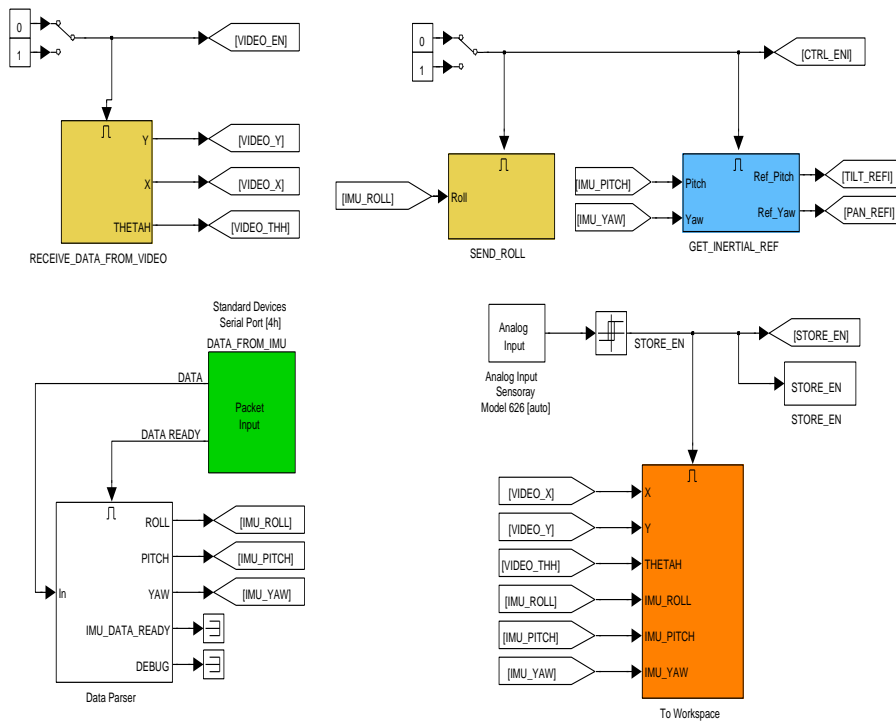


Figura 51: Blocchetti simulink ausiliari al sistema di controllo

RISULTATI SPERIMENTALI

5.1 INTRODUZIONE

In questo capitolo verranno presentate le prove eseguite al fine di valutare la bontà del controllo progettato per raggiungere gli obiettivi preposti all'inizio; tali obiettivi sono:

- effettuare la stabilizzazione totale delle immagini disaccoppiandole quindi da qualsiasi movimento della base;
- realizzare il tracking di un bersaglio mobile che si muove ad una velocità arbitraria ad una distanza arbitraria.

Per effettuare le prove, tramite software *Matlab*, è stato creato un target fisso costituito da una croce nera collocata su uno sfondo bianco ed un target mobile che si muove ad una velocità costante orizzontalmente in senso positivo e negativo. I listati matlab relativi ai due tipi di target sono riportati in appendice.

La piattaforma di Stewart su cui poggia il brandeggio viene movimentata secondo due profili di movimento:

1. movimento sinusoidale nelle tre direzioni, yaw-pitch-roll così composto:
 - movimento di solo moto di roll con ampiezza massima pari a 10° (30s)
 - posizione di home (5s)
 - movimento di solo moto di pitch con ampiezza massima pari a 10° (30s)
 - posizione di home (5s)
 - movimento di solo moto di yaw con ampiezza massima pari a 10° (30s)
2. movimento generato tramite dei dati reali di moto ondoso rilevati a bordo di una nave della guardia costiera [15].

Le prove sono state eseguite in due diverse condizioni operative:

- bersaglio a 70 cm dalla telecamera e assenza di zoom;
- bersaglio a 3.60 m dalla telecamera e zoom di 3x.

Per ognuna di queste due condizioni si è reso necessario effettuare l'operazione di sintonizzazione dei guadagni nelle modalità descritte nel capitolo precedente.

5.2 TEST DI STABILIZZAZIONE DELL'IMMAGINE

La prova effettuata per verificare la bontà del controllo proposto al fine di stabilizzare l'immagine consiste nell'individuazione di un target fisso e valutarne gli spostamenti nel piano immagine causati solamente dal moto della base su cui poggia il brandeggio; tale target fisso, come già anticipato, è costituito da due linee disposte rispettivamente in modo orizzontale e verticale in modo da formare una croce. Le linee vengono individuate tramite l'algoritmo di riconoscimento delle linee illustrato nel capitolo 3 eseguendo quindi i seguenti passi in successione:

1. Individuazione delle linee;
2. Calcolo delle coordinate del punto di incontro espresse in pixel;
3. Tracciamento delle linee individuate a video;
4. Tracciamento cerchio nel punto di incontro determinato;
5. Disegno croce nel centro dell'immagine.

Come evidenziato in Figura 52, l'effetto della stabilizzazione introdotta è quello di visualizzare un'immagine in cui, nonostante i movimenti della base del brandeggio, la linea orizzontale è mantenuta sempre in tale posizione mentre viene invece ruotata l'immagine e ciò lo si nota dalla rotazione presente ai bordi.

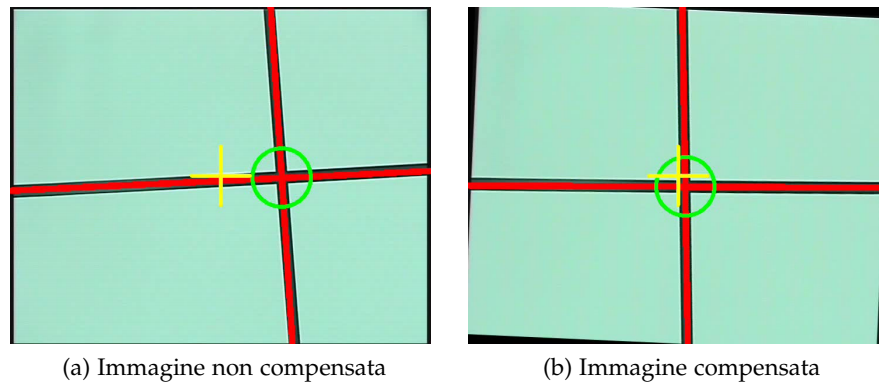


Figura 52: Frame tipici acquisiti durante i test di stabilizzazione delle immagini. Nella parte sinistra è riportata l'immagine non compensata e in quella destra l'immagine compensata

In figura si vede come, nel caso di immagine totalmente compensata, il centro della croce vada a collocarsi in prossimità del centro immagine.

È importante notare come ciò non avverrebbe se fosse presente solamente l'anello di stabilizzazione inerziale poichè esso tenderebbe solamente a mantenere costante l'orientazione della camera nonostante i movimenti di yaw e di pitch della base mentre non rilevarebbe

uno spostamento residuo del centro della croce sull'asse x del piano immagine causato dal fatto che il centro di rotazione della piattaforma non coincide con il centro della camera. Tale diversità produce dunque un'oscillazione residua lungo l'asse x (jitter residuo) che non verrebbe rilevata se non ci fosse l'algoritmo di elaborazione video, il quale rileva tale spostamento e lo invia in ingresso al SdC dei motori in modo da ridurlo.

Nei prossimi tre paragrafi verranno illustrati i dati raccolti nelle seguenti prove effettuate:

- piattaforma di Stewart movimentata con segnale sinusoidale e target a 70cm dalla camera;
- piattaforma di Stewart movimentata con segnale simulante il moto ondoso e target a 70cm dalla camera;
- piattaforma di Stewart movimentata con segnale simulante il moto ondoso e target posto a 3.6m dalla camera con inserimento di un fattore di zoom 3x.

5.2.1 *Movimentazione della piattaforma con segnale sinusoidale e target posto a 70cm*

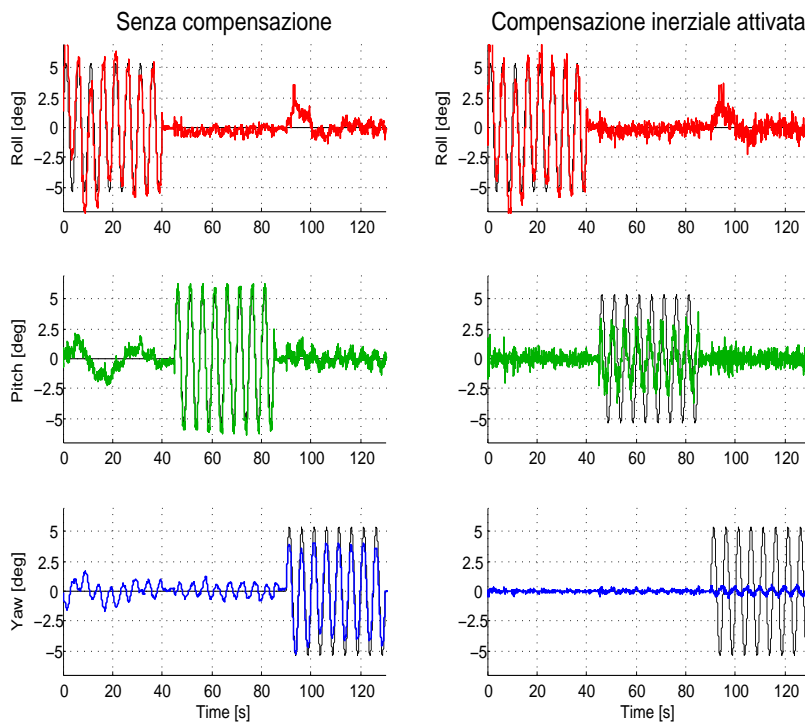


Figura 53: Angoli di roll, pitch e yaw rilevati tramite la IMU nel caso di stabilizzazione inerziale disattivata e attivata con segnale di riferimento sinusoidale; in linea nera sottile sono riportati i segnali di riferimento forniti alla piattaforma

Angolo	Simbolo	No comp. [deg.]	Inertial comp. [deg]
Roll	σ_R	2.23	2.30
Pitch	σ_P	2.34	0.99
Yaw	σ_Y	1.66	0.15

Tabella 2: Deviazioni standard degli angoli di roll-pitch-yaw nel caso di stabilizzazione inerziale disattivata e attivata con movimentazione con segnale di riferimento sinusoidale

In Figura 53 sono rappresentate le misure rilevate tramite la IMU riguardanti gli angoli di roll, pitch e yaw, confrontando il caso in cui la stabilizzazione inerziale sia disattivata e quello in cui è attivata. Come era doveroso aspettarsi nelle prove dove la stabilizzazione inerziale è disattivata, le misure inerziali della IMU sono praticamente sovrapponibili al segnale di riferimento poichè non è presente alcuna azione di controllo. Nel caso in cui il controllo inerziale è invece attivato, si nota una diminuzione dei movimenti di yaw e pitch. Ovviamente per il movimento di roll, invece, la situazione rimane analoga al caso in cui non sia presente la stabilizzazione poichè tramite il brangdeggi è possibile effettuare la stabilizzazione solamente secondo due direzioni. In Tabella 2 sono riportati i valori delle deviazioni standard degli spostamenti nel caso di compensazione inerziale attivata e disattivata; è possibile constatare come la stabilizzazione inerziale della camera è dimostrata dalla diminuzione delle deviazioni standard lungo gli assi di yaw e pitch mentre non si verifica nessun cambiamento sul valore del roll. Si nota inoltre come la compensazione di yaw sia notevolmente migliore di quella di pitch; ciò si spiega ricordando come l'anello di yaw presenta una minore reattività al rumore rispetto all'anello di pitch possedendo un'inerzia maggiore ed inoltre si è riscontrata una maggiore rumorosità nel segnale di misura della IMU relativo al movimento di pitch rispetto a quello di roll.

Un'ulteriore notazione che è necessario fare riguarda la presenza di una cross-relazione tra le misure di orientazione dei vari assi della IMU; si può infatti notare dalla Figura 53 come le oscillazioni di roll vadano ad influenzare anche le misure di yaw, come pure esiste una correlazione tra i segnali di yaw e pitch.

Come descritto nei capitoli precedenti, la stabilizzazione del movimento di roll avviene a livello software applicando una rotazione all'immagine di ugual valore ma segno opposto al valore misurato dalla IMU. La bontà di tale componente software può essere verificata rilevando l'inclinazione della linea identificata sul piano immagine, la quale, con immagine totalmente disaccoppiata dai movimenti della base, dovrebbe idealmente mantenersi orizzontale e quindi avere un'inclinazione prossima allo zero.

In Figura 54 sono rappresentati gli andamenti dell'inclinazione della retta orizzontale e delle coordinate x-y del centro della croce riferite al centro del piano immagine, nel caso siano presenti sia la stabilizzazione inerziale che del movimento di roll e nel caso in cui esse siano entrambe assenti.

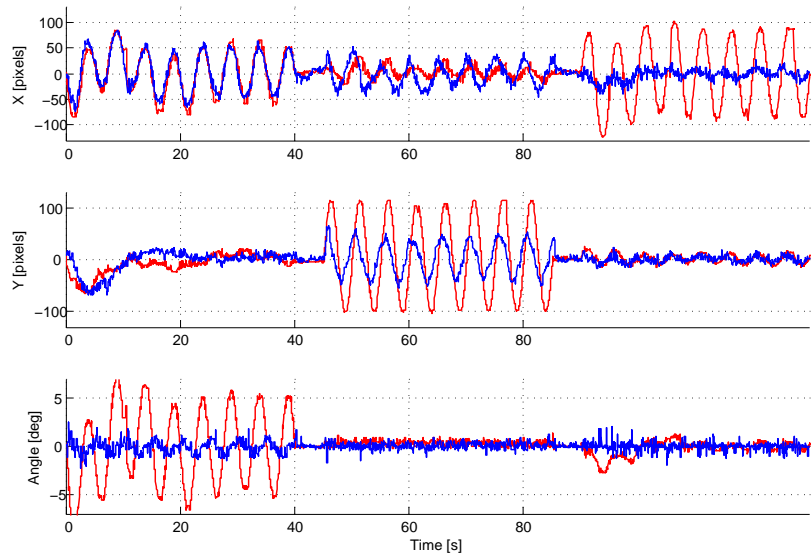


Figura 54: Andamenti delle coordinate del centro della croce e inclinazione della retta orizzontale nel caso in cui la compensazione inerziale e del movimento di roll siano entrambe attive (linea blu) e nel caso siano entrambe assenti (linea rossa), movimentazione con segnale sinusoidale e target a 70cm

Parametro	Simbolo	No comp. [deg.]	In. & Roll comp. [deg]
Inclinazione retta orizzontale	σ_{ϕ}	0.04	0.008

Tabella 3: Deviazioni standard dell'inclinazione della retta individuata nel caso in cui la stabilizzazione del movimento di roll sia assente e nel caso essa sia presente, movimentazione con segnale sinusoidale e target posto a 70cm.

Si nota subito come il movimento di roll risulti opportunamente compensato e ciò è confermato dai valori riportati in Tabella 3, dove si ha un riduzione di un fattore 5 della varianza del valore dell'inclinazione della retta rispetto al caso senza compensazione del roll. I benefici ottenuti dall'introduzione dell'anello di stabilizzazione inerziale si possono notare anche in Figura 55 dove sono riportati gli andamenti nel piano X,Y delle coordinate stesse nei due casi; si nota chiaramente come nel caso sia presente la stabilizzazione inerziale i valori delle coordinate sono molto più concentrati attorno al punto

(0,0) rispetto al caso in cui la stabilizzazione non sia presente. Il cerchio tracciato nelle due figure ha raggio pari a $3\sigma_r$, dove σ_r è il valore della deviazione standard radiale delle coordinate nei casi analizzati. Tale cerchio sarà presente anche in tutte le figure successive che rappresentano l'andamento nel piano x-y del coordinate del target, sia esso fisso o mobile, nelle varie prove effettuate.

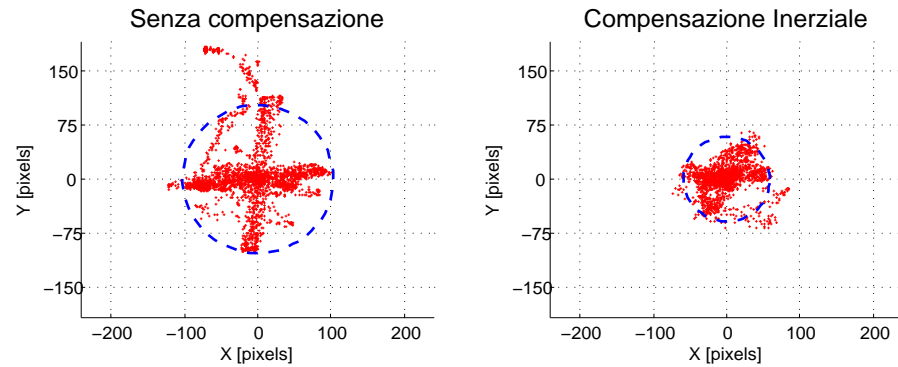


Figura 55: Coordinate del target fisso nel piano X,Y nel caso di compensazione inerziale disattivata e attivata, movimentazione con segnale sinusoidale e target a 70cm

Direzione	Simbolo	No comp. [pixels]	Inertial comp. [pixels]
asse x	σ_x	42.59	25.69
asse y	σ_y	43.38	21.45
radiale	σ_r	34.28	19.6

Tabella 4: Deviazioni standard delle coordinate del centro della croce individuata nel caso di compensazione inerziale disattivata e attivata, movimentazione con segnale sinusoidale e target a 70cm

Dopo aver analizzato i vantaggi che si ottengono dall'inserimento di un sistema di controllo avente il compito di effettuare la stabilizzazione inerziale e del movimento di roll, prendiamo ora in considerazione il caso in cui venga aggiunto a tale sistema di controllo, un anello di controllo video che riceve in ingresso le coordinate di un determinato target situato sul piano immagine e fornisce dei segnali di riferimento per il sistema di stabilizzazione inerziale al fine di consentire il monitoraggio del target stesso. L'anello di controllo sarà pertanto collocato esternamente a quello precedentemente illustrato.

Nelle figure 56 e 57 sono riportati i risultati di tali prove.

Si nota chiaramente come l'utilizzo di un sistema di controllo composto da un anello esterno di controllo video e uno interno di stabilizzazione inerziale sia la soluzione che permette di ottenere i risultati migliori e ciò è confermato anche dai valori riportati in Tabella 5.

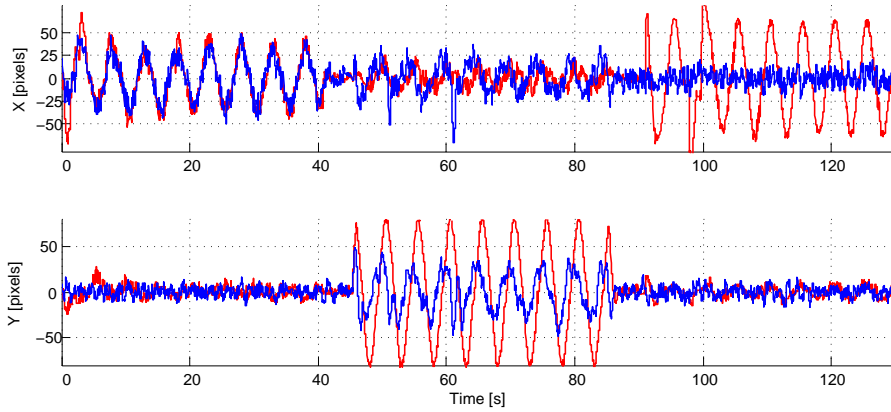


Figura 56: Andamenti delle coordinate del target fisso nel caso sia attivo il solo controllo video (linea rossa) e nel caso sia presente anche l'anello di stabilizzazione inerziale (linea blu), movimentazione con segnale sinusoidale e target a 70cm

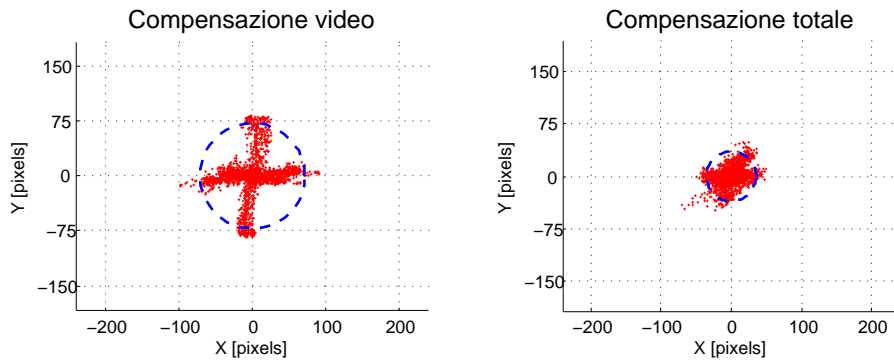


Figura 57: Coordinate del centro della croce nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, movimentazione con segnale sinusoidale e target a 70cm.

Direzione	Simbolo	Video comp. [pixels]	Full comp. [pixels]
asse x	σ_x	28.92	16.42
asse y	σ_y	30.11	13.20
radiale	σ_r	23.85	11.78

Tabella 5: Deviazioni standard delle coordinate del target fisso nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, movimentazione con segnale sinusoidale e target a 70cm

5.2.2 Movimentazione della piattaforma con dati del moto ondoso e target posto a 70cm

Nella seguente sezione vengono riportati i dati raccolti effettuando le stesse prove illustrate nel paragrafo precedente ma movimentando la piattaforma con un particolare segnale che simula il moto ondoso a cui è sottoposta una boa situata in mare sulla quale viene idealmente collocato il brandeggio.

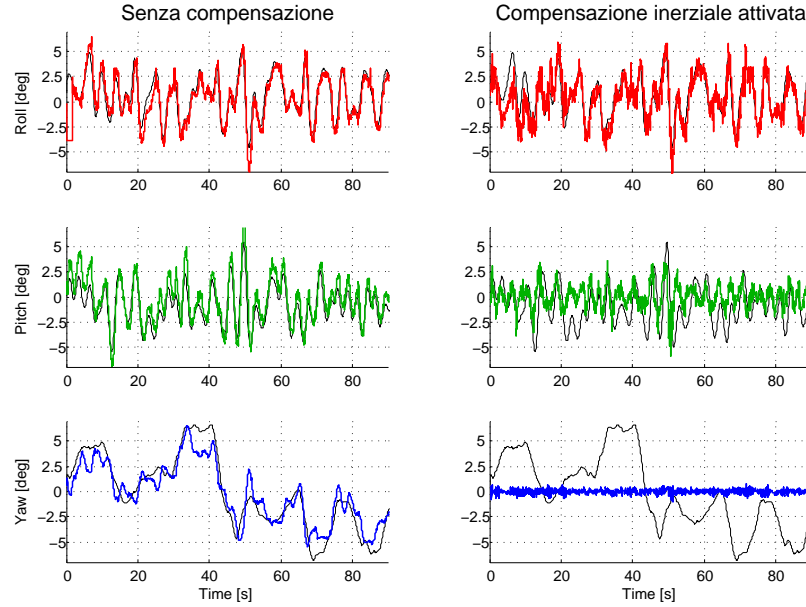


Figura 58: Angoli di roll,pitch e yaw rilevati tramite la IMU nel caso di stabilizzazione inerziale disattivata e attivata con movimentazione tramite segnale di moto ondoso; in linea nera sottile sono riportati i segnali di riferimento forniti alla piattaforma

Angolo	Simbolo	No comp. [deg.]	Inertial comp. [deg]
Roll	σ_R	2.55	2.54
Pitch	σ_P	2.30	1.05
Yaw	σ_Y	2.77	0.18

Tabella 6: Deviazioni standard degli angoli di roll-pitch-yaw nel caso di stabilizzazione inerziale disattivata e attivata con movimentazione tramite segnale di moto ondoso

In Figura 58 sono rappresentate le misure rilevate tramite la IMU riguardanti gli angoli di roll, pitch e yaw, confrontando il caso in cui la stabilizzazione inerziale sia disattivata (parte sinistra) e quello in cui è attivata (parte destra); le considerazioni da farsi relativamente a tale figura sono analoghe a quelle della sezione precedente e, con l'ausilio della Tabella 6 si nota come in questo caso la compensazione

di yaw sia ancora migliore rispetto al caso precedente. In Figura 59 sono rappresentati gli andamenti delle coordinate del target fisso, rispetto al centro del piano immagine, nel caso di stabilizzazione inerziale disattivata e attivata ed è inoltre riportato l'andamento dell'inclinazione della retta orizzontale nel caso di compensazione del roll attivata e disattivata.

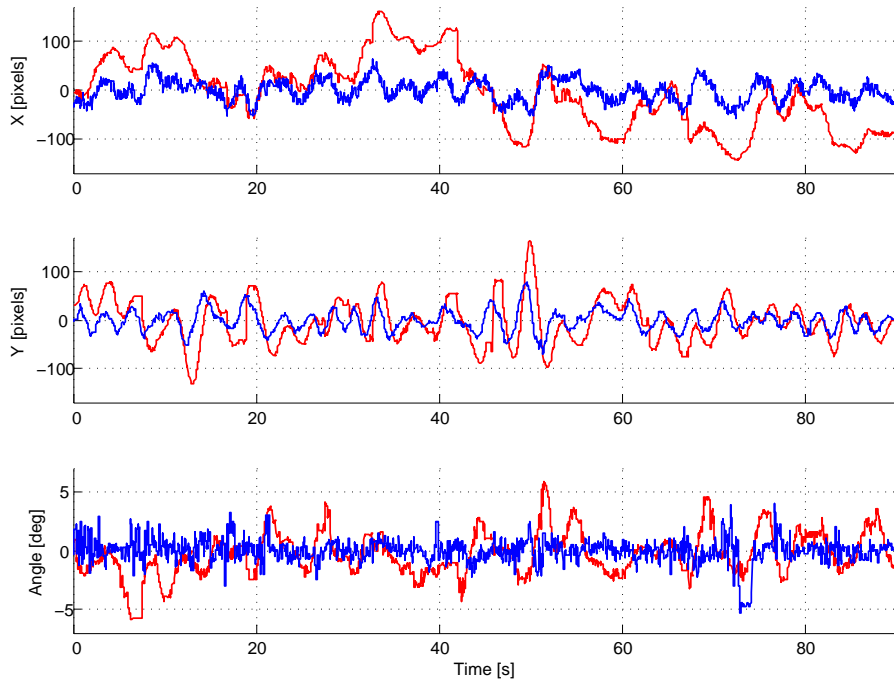


Figura 59: Andamenti delle coordinate del centro della croce e inclinazione della retta orizzontale nel caso in cui la compensazione inerziale e del movimento di roll siano entrambe attive (linea blu) e nel caso siano entrambe assenti (linea rossa), movimentazione con segnale di moto ondoso e target a 70cm

Parametro	Simbolo	No comp. [deg.]	In. comp. [deg]
Inclinazione retta orizzontale	σ_ϕ	0.039	0.017

Tabella 7: Deviazioni standard dell'inclinazione della retta individuata, movimentazione tramite segnale di moto ondoso e target posto a 70cm

L'andamento nel piano X,Y delle coordinate del centro della croce con segnale di riferimento di moto ondoso viene riportato in Figura 60, mentre nella successiva Tabella 8 sono riportati i valori delle varianze degli spostamenti del centro della croce nel caso di controllo inerziale disattivato e attivato.

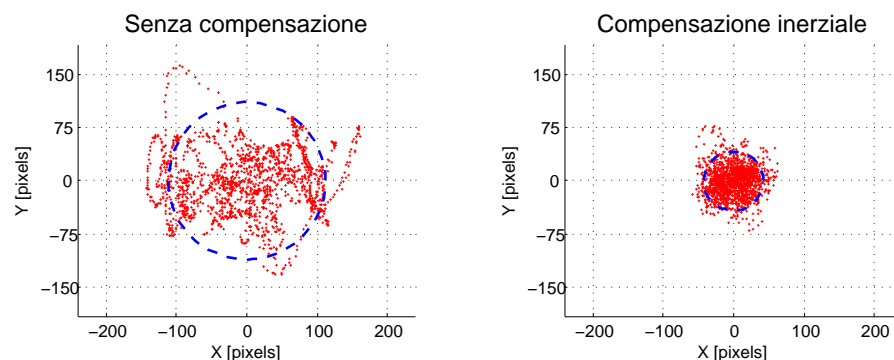


Figura 60: Coordinate del centro della croce nel piano X,Y nel caso di stabilizzazione inerziale disattivata e attivata con movimentazione tramite segnale di moto ondoso e target a 70cm

Direzione	Simbolo	No comp. [deg.]	In. comp. [deg.]
asse x	σ_x	72.05	22.31
asse y	σ_y	44.61	20.76
radiale	σ_r	37.2	14.02

Tabella 8: Deviazioni standard delle coordinate X,Y del centro della croce con movimentazione tramite segnale di moto ondoso e target posto a 70cm

Nelle figure 61, 62 e Tabella 9 riportate nella pagina seguente sono riportati gli andamenti delle coordinate del centro della croce riferite al centro del piano immagine, nel tempo e nello spazio X-Y, nel caso sia attivo il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale.

Anche in questo caso si verifica un miglioramento delle prestazioni del sistema e tale miglioramento risulta maggiore nell'asse di pitch rispetto all'asse di yaw per i problemi descritti nella precedente sezione illustrando le prove con segnale sinusoidale.

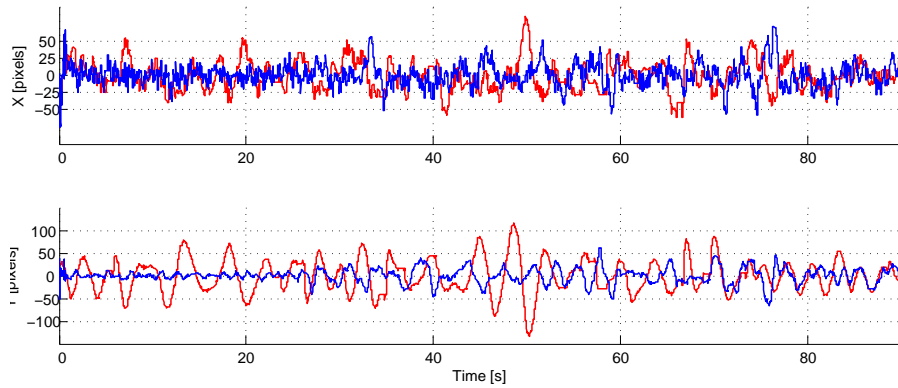


Figura 61: Andamenti delle coordinate del centro della croce nel caso di solo controllo video (linea grigia) e nel caso sia presente anche l'anello di stabilizzazione inerziale con movimentazione tramite segnale di moto ondoso, target a 70cm

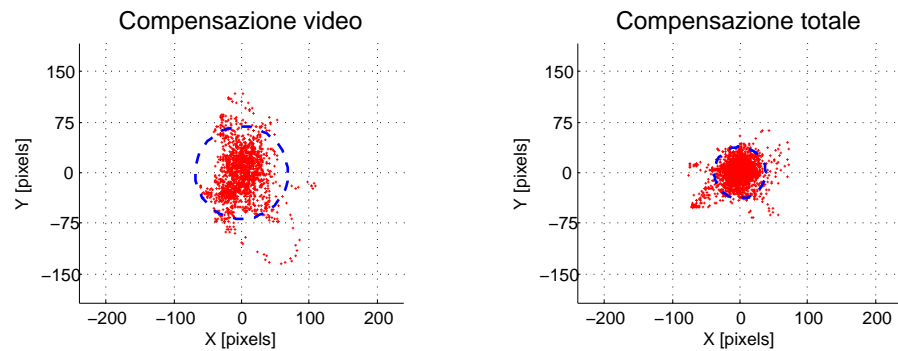


Figura 62: Andamenti delle coordinate del centro della croce nel piano X,Y nel caso sia presente il solo controllo video (parte sinistra) e nel caso sia presente anche l'anello di stabilizzazione inerziale (parte destra) con segnale di moto ondoso, target a 70cm

Direzione	Simbolo	Video comp. [pixels]	Full comp. [pixels]
asse x	σ_x	22.16	15.90
asse y	σ_y	35.20	16.30
radiale	σ_r	22.80	12.71

Tabella 9: Deviazioni standard delle coordinate del centro della croce individuata nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con segnale di riferimento del moto ondoso

5.2.3 *Movimentazione della piattaforma con dati del moto ondoso, target a 3.6m e fattore di zoom 3x*

I risultati illustrati in questa sezione sono stati ottenuti da prove effettuate ponendo il target a una distanza di 3.60m dalla camera ed inserendo un fattore di zoom di 3x. Come si vede dalle figure riportate in seguito, l'aumento della distanza dalla camera e l'inserimento dello zoom hanno portato ad un generale peggioramento delle prestazioni rispetto alle prove eseguite con il target posto a 70cm dalla camera e in assenza di zoom; ciò è dovuto al fatto che le vibrazioni indotte nel sistema di controllo dal segnale rumoroso proveniente dal sensore di misura vengono amplificate notevolmente all'aumentare della distanza e tale aumento è ancora maggiore nel caso sia presente lo zoom. Nelle figure seguenti sono riportati gli andamenti delle coordinate, nel tempo e nello spazio X-Y, nel caso sia attivo il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, sempre col fattore di zoom 3x inserito.

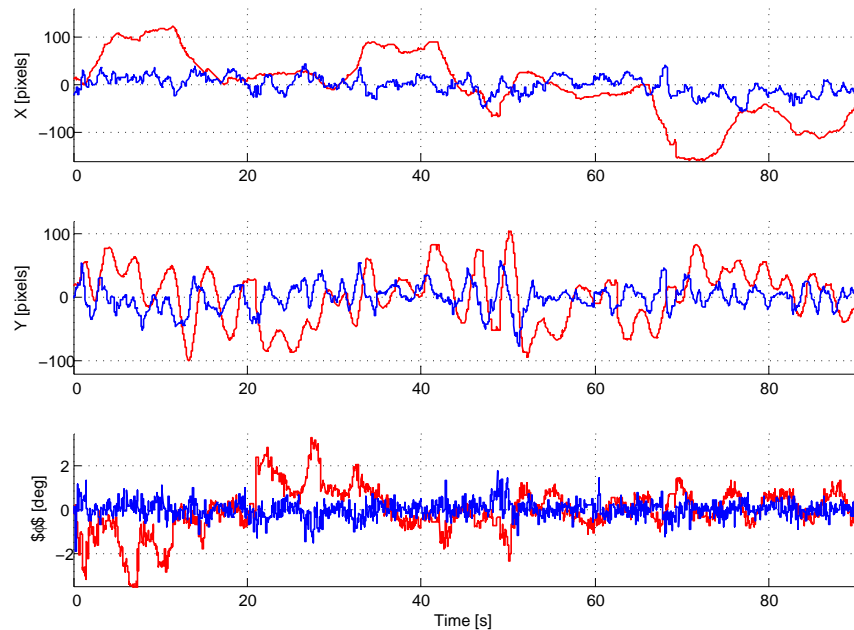


Figura 63: Andamenti delle coordinate del centro della croce e inclinazione della retta orizzontale nel caso in cui la compensazione inerziale e del movimento di roll siano entrambe attive (linea blu) e nel caso siano entrambe assenti (linea rossa), movimentazione con segnale di moto ondoso e presenza di un fattore di zoom 3x

Dalla Figura 63 si nota subito come il miglioramento cercato, seppure presente, sia minore rispetto alle prove analizzate in precedenza. In questo caso la varianza dell'inclinazione della retta orizzontale individuata passa da 0.017 nel caso in cui non sia attiva alcuna forma di stabilizzazione, a 0.0066 nel caso in cui sia attiva la compensazione inerziale e del movimento di roll.

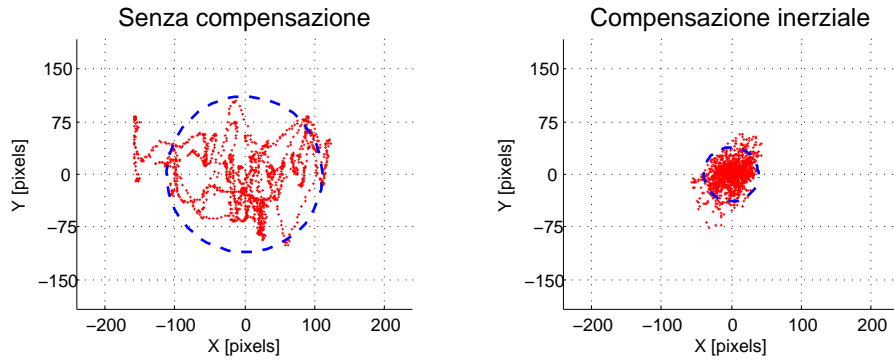


Figura 64: Coordinate del centro della croce nel piano X,Y con movimentazione tramite segnale di moto ondoso nel caso di stabilizzazione inerziale disattivata e attivata, zoom 3x e target a 3.6m

Direzione	Simbolo	No comp. [deg.]	In. comp. [deg.]
asse x	σ_x	67.77	18.28
asse y	σ_y	41.91	18.64
radiale	σ_r	36.99	12.97

Tabella 10: Deviazioni standard delle coordinate X,Y del centro della croce con movimentazione tramite segnale di moto ondoso, zoom 3x e target a 3.6m

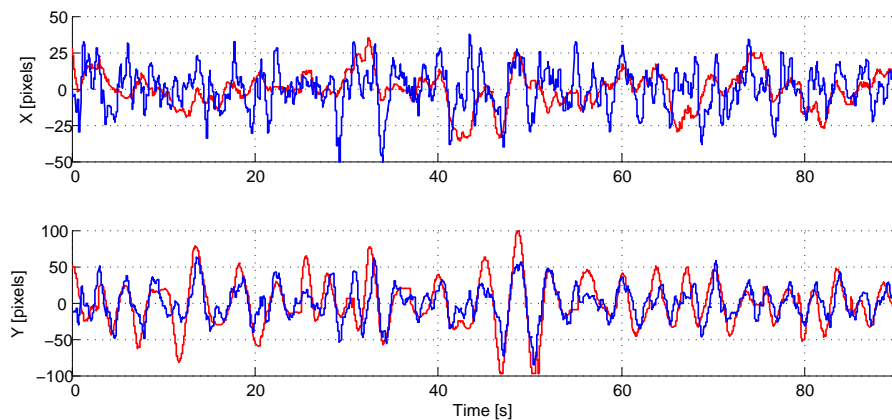


Figura 65: Andamenti delle coordinate del centro della croce nel caso sia attivo il solo controllo video (linea rossa) e nel caso sia presente anche l'anello di stabilizzazione inerziale (linea blu), movimentazione tramite segnale di moto ondoso, zoom 3x e target a 3.6m

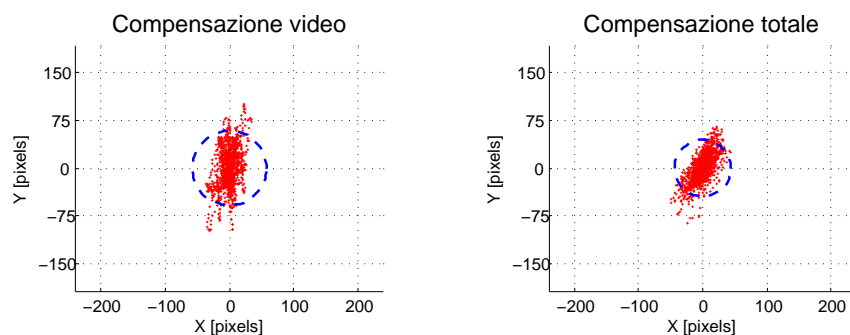


Figura 66: Coordinate del centro della croce nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, movimentazione tramite segnale di moto ondoso, zoom 3x e target a 3.6m

Direzione	Simbolo	Video comp. [pixels]	Full comp. [pixels]
asse x	σ_x	11.91	14.40
asse y	σ_y	31.12	22.46
radiale	σ_r	19.30	14.66

Tabella 11: Deviazioni standard delle coordinate del centro della croce nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con segnale di moto ondoso, zoom 3x e target a 3.6m

Come preannunciato all'inizio del capitolo, in queste condizioni operative si hanno dei miglioramenti minori rispetto ai due casi precedenti dovuti al fatto che la presenza dello zoom in combinazione con l'allontanamento del target amplifica i disturbi introdotti nel sistema dal sensore di misura inerziale.

5.3 TEST DI INSEGUIMENTO DEL TARGET

Il target mobile necessario per effettuare le prove di inseguimento del target è costituito da un punto che si muove di moto rettilineo uniforme sull'asse delle ascisse e creato tramite software Matlab con lo scopo di simulare il comportamento di una nave che viene rilevata a video tramite un particolare algoritmo di tracking [4].

Tale algoritmo si basa sulla differenza di colore, e quindi di contrasto, tra il bersaglio da inseguire e il resto del piano immagine e permette all'operatore di selezionare il bersaglio tramite il mouse direttamente sullo schermo. Per garantire dunque una corretta individuazione del bersaglio sul piano immagine, il punto simulante il comportamento della nave è stato realizzato di colore nero collocato su sfondo bianco per cercare di ridurre al minimo i casi di falsa identificazione.

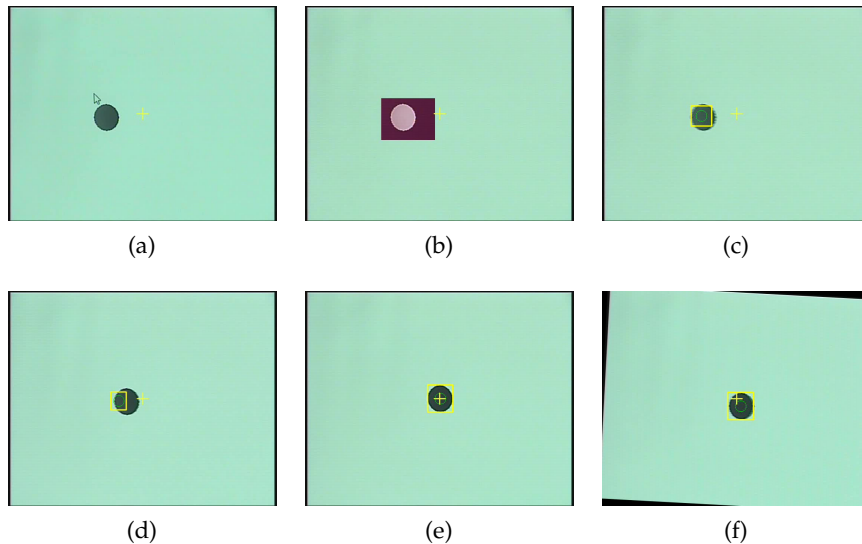


Figura 67: Frame tipici acquisiti dalla telecamera durante i test di inseguimento del target

In Figura 67 sono ripotate le fasi che si susseguono nel monitoraggio di un determinato obiettivo:

- Individuazione del target da parte dell'operatore e successiva selezione, Figura 67a e Figura 67b;
- L'algoritmo di riconoscimento individua il target all'interno del rettangolo giallo di selezione, Figura 67c;
- L'intero sistema di controllo riceve i riferimenti opportuni e mantiene l'oggetto al centro dell'immagine effettuando la compensazione totale della stessa, Figura 67d ,Figura 67e e Figura 67f.

L'obiettivo del controllo sviluppato è di mantenere tale bersaglio selezionato sempre al centro del piano immagine e fare in modo che le sue coordinate, riferite al centro immagine, siano sempre prossime allo zero in modo da assicurare il corretto inseguimento del corpo. Come per i test di stabilizzazione dell'immagine, anche per valutare la bontà del controllo al fine di eseguire il tracking di un particolare oggetto, si sono svolte le prove nelle 3 diverse condizioni operative precedentemente illustrate.

Per quanto riguarda le prove effettuate con segnale di riferimento sinusoidale, in Figura 68 è rappresentato l'andamento delle coordinate dell'oggetto in movimento effettuando il confronto tra il caso in cui sia presente il solo controllo video (linea grigia) e quello in cui sia presente anche il controllo di stabilizzazione inerziale (linea nera) mentre in Figura 69 e Tabella 12 sono rappresentate rispettivamente le coordinate dell'oggetto nel piano X-Y e le loro varianze.

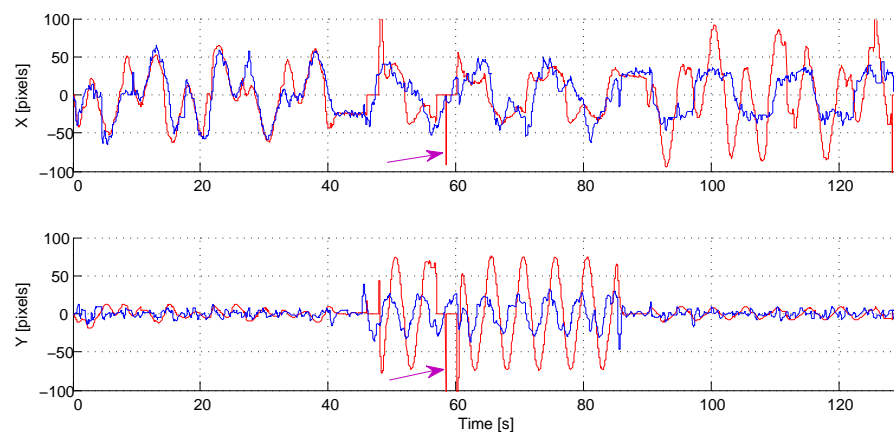


Figura 68: Andamenti delle coordinate del target mobile nel caso di solo controllo video (linea rossa) e nel caso sia presente anche l'anello di stabilizzazione inerziale (linea blu), movimentazione tramite segnale sinusoidale e target a 70cm.

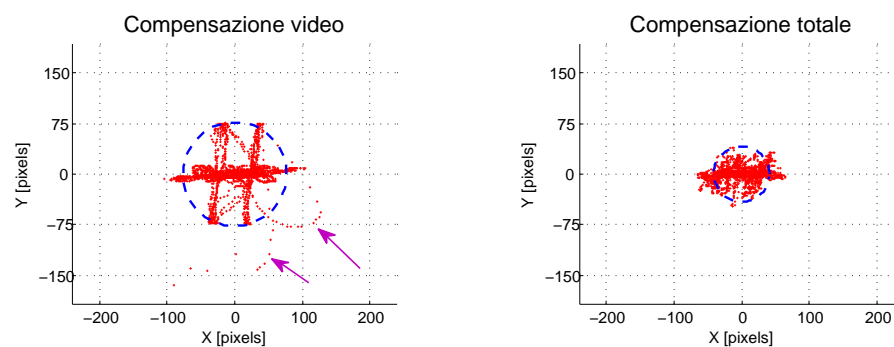


Figura 69: Coordinate del target mobile nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con movimentazione tramite segnale sinusoidale, target a 70cm

Direzione	Simbolo	Video comp. [pixels]	Full comp. [pixels]
asse x	σ_x	36.32	28.44
asse y	σ_y	27.81	11.14
radiale	σ_r	25.65	13.78

Tabella 12: Deviazioni standard delle coordinate del target mobile nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale, movimentazione tramite segnale sinusoidale, target a 70cm

I risultati delle prove svolte movimentando la piattaforma con movimentazione tramite segnale di riferimento di moto ondoso sono invece riportati in Figura 70, Figura 71 e Tabella 13.

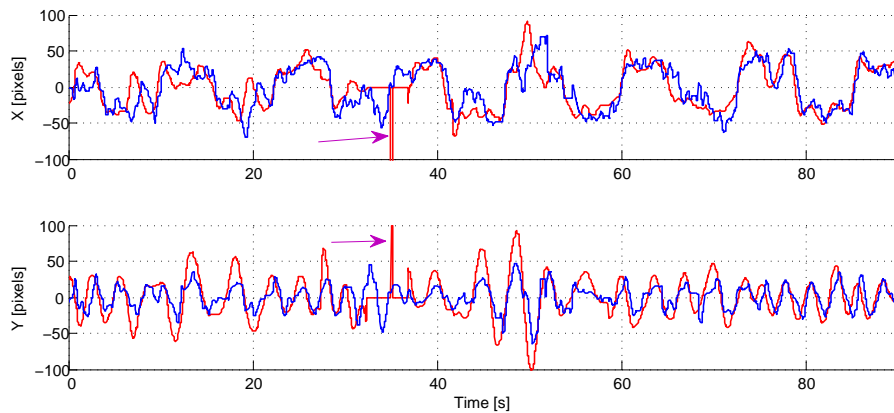


Figura 70: Andamenti delle coordinate del target mobile nel caso di solo controllo video (linea rossa) e nel caso sia presente anche la stabilizzazione inerziale (linea blu), movimentazione tramite segnale di moto ondoso, target a 70cm

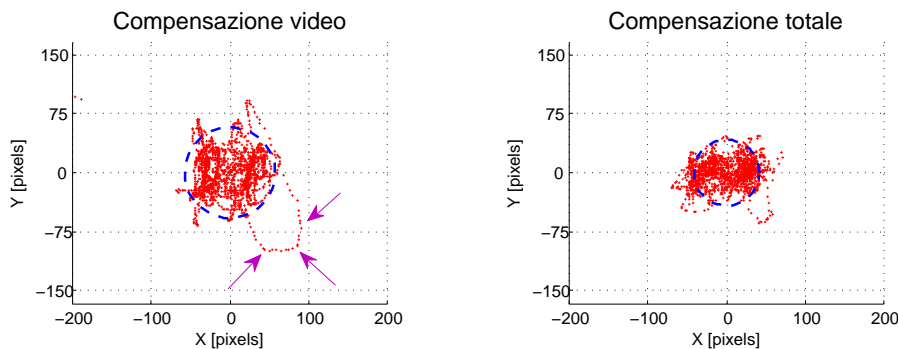


Figura 71: Coordinate del target mobile nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche la stabilizzazione inerziale, movimentazione tramite segnale di moto ondoso, target a 70cm

Da entrambe le prove effettuate emerge chiaramente il miglioramento ottenuto nell'inserimento di un anello di controllo inerziale in parallelo ad un anello di controllo video poichè abbiamo un significativo aumento delle prestazioni del sistema che consentono un miglior monitoraggio dell'oggetto in movimento. È importante osservare come durante le prove con il solo controllo video attivato, la camera perda il target da monitorare poichè non è in grado di inseguirlo a seguito di un suo movimento e del movimento della base; in questo caso è necessario selezionare nuovamente il target per continuare la fase di monitoraggio. Tale perdita del bersaglio da parte del sistema di controllo si può notare, attraverso le figure presentate, in due modi:

Direzione	Simbolo	Video comp. [pixels]	Full comp. [pixels]
asse x	σ_x	29.72	29.90
asse y	σ_y	27.20	16.31
radiale	σ_r	19.19	13.96

Tabella 13: Deviazioni standard delle coordinate del target mobile nel caso sia presente il solo controllo video e nel caso sia presente anche la stabilizzazione inerziale, movimentazione tramite segnale di moto ondoso, target a 70cm

- Dalle figure 68 e 70, che rappresentano rispettivamente l'andamento delle coordinate nel tempo nel caso di movimentazione tramite segnale sinusoidale e tramite segnale del moto ondoso, evidenziando dei picchi molto grandi indicati dalle frecce viola e seguiti da istanti in cui le coordinate rimangono entrambe a zero che rappresentano gli istanti di perdita del target precedenti la successiva nuova selezione del target da parte dell'operatore;
- Dalle figure 69 e 71, che rappresentano l'andamento delle coordinate x-y del target nel piano stesso, notando la presenza di punti aventi coordinate che differiscono notevolmente dai valori attorno allo zero ed indicati dalle frecce viola.

In conclusione a queste prove si può dire che il sistema di controllo progettato consente di effettuare il tracking di un particolare oggetto di interesse selezionato dall'operatore con la sicurezza che il target venga mantenuto sempre al centro del piano immagine senza ulteriori interventi esterni, mentre nel caso sia presente il solo controllo video è necessaria la costante presenza di un operatore che selezioni nuovamente il target quando questo sta per uscire dal piano immagine.

Per completezza di trattazione sono inserite qui di seguito anche le prove effettuate movimentando la base col segnale di riferimento di moto ondoso, target posto a 3.60m e inserimento di un fattore di zoom di 3x. In questo caso i miglioramenti ottenuti sono minori per effetto dei problemi introdotti nel paragrafo precedente.

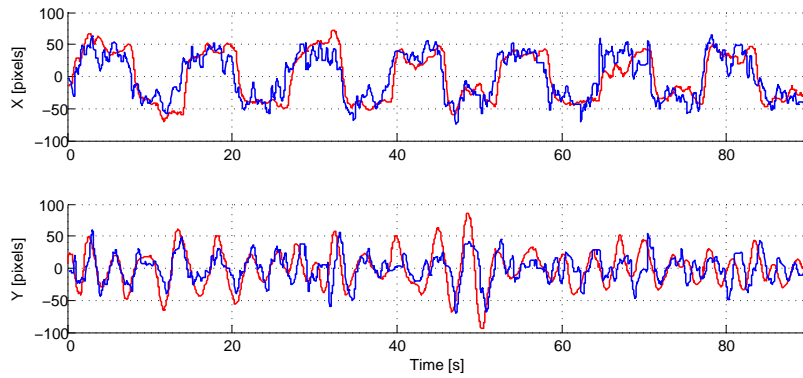


Figura 72: Andamenti delle coordinate del target mobile nel caso di solo controllo video (linea rossa) e nel caso sia presente anche l'anello di stabilizzazione inerziale (linea blu) con segnale di riferimento di moto ondoso e zoom 3x

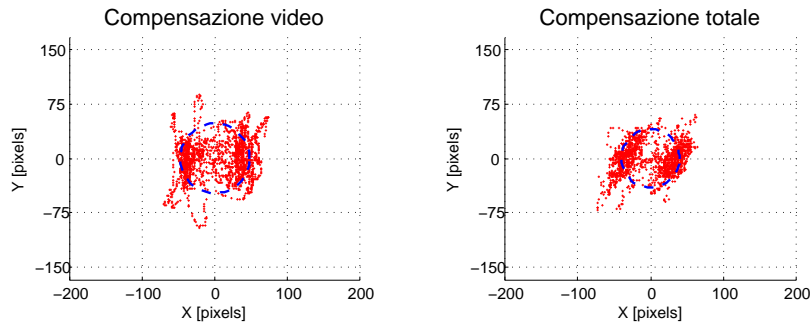


Figura 73: Andamenti delle coordinate del target mobile nel piano X,Y nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con segnale di riferimento di moto ondoso e zoom 3x

Direzione	Simbolo	Video comp. [pixels]	Full comp. [pixels]
asse x	σ_x	38.19	34.98
asse y	σ_y	26.34	20.21
radiale	σ_r	15.93	13.34

Tabella 14: Deviazioni standard delle coordinate del target mobile nel caso sia presente il solo controllo video e nel caso sia presente anche l'anello di stabilizzazione inerziale con segnale di riferimento di moto ondoso e zoom 3x

CONCLUSIONI

Gli obiettivi posti all'inizio del lavoro erano di effettuare la stabilizzazione delle immagini compensando i movimenti di yaw, pitch e roll che si verificano alla base del brandeggio, e di sviluppare un sistema di controllo che consenta di effettuare il tracking di un oggetto che si muove a velocità e distanze arbitrarie. La soluzione proposta in questo lavoro si differenzia da quelle già presenti sul mercato per il fatto che, per effettuare la stabilizzazione dell'immagine, utilizza solamente le misure derivanti da un sensore di misura inerziale (IMU).

Dopo un'attenta fase preliminare di analisi del sistema, si è giunti alla conclusione che solamente due dei tre movimenti che si manifestano alla base del brandeggio possono essere stabilizzati elettromeccanicamente fornendo opportuni segnali di controllo ai motori presenti all'interno del brandeggio stesso; questo perché esso possiede solamente due gradi di libertà, yaw e pitch. La stabilizzazione del movimento di roll è stata effettuata a livello software, applicando all'immagine una rotazione di ugual valore ma verso opposto al valore del movimento stesso rilevato tramite il sensore di misura inerziale. Una volta ottenuto il disaccoppiamento totale delle immagini dal movimento della base, è stato progettato un sistema di controllo in grado di effettuare il tracking di un particolare oggetto presente nello scenario, ed in grado di mantenerlo sempre al centro del piano immagine per poterne monitorare il comportamento.

Attraverso le prove sperimentali effettuate in laboratorio è stato possibile verificare la bontà del controllo progettato. A tal fine sono state svolte prove con il target fisso volte a verificare la reale compensazione totale delle immagini, e prove con il target mobile per valutare la correttezza del sistema di tracking preposto. Le prove sono state svolte cercando di simulare al meglio le condizioni di lavoro reali attraverso un'opportuna movimentazione del simulatore di moto ondoso.

I risultati raggiunti potrebbero essere il punto di partenza di lavori futuri volti a migliorare le prestazioni del sistema ottenuto soprattutto per quanto riguarda l'anello di stabilizzazione inerziale che risente della presenza di segnali di disturbo introdotti nel sistema dal sensore di misura inerziali. Ulteriori applicazioni future potrebbero riguardare l'utilizzo di algoritmi di predizione al fine di predire la traiettoria eseguita da un oggetto in movimento per poterne effettuare il monitoraggio anche in mancanza di informazioni estratte dalle immagini.

APPENDIX

A.1 CODICE C++ NECESSARIO PER L'IDENTIFICAZIONE DEL TARGET FISSO.

```

1  #include "highgui.h"
   #include <cv.h>
   #include "cvaux.h"
   #include <iostream>
   #include <time.h>
6  #include "imgproc.hpp"
   #include "v4l2_input.h"
   #include "watch_win.h"
   #include "fixed_target.h"
   #include <stdio.h>
11 #include <sys/time.h>
   #include <iostream>
   #include <fstream>
   #include <stdlib.h>
   #include <sys/socket.h>
16 #include <arpa/inet.h>
   #include <stdlib.h>
   #include <string.h>
   #include <fcntl.h>
   #include <sys/file.h>
21 #include <signal.h>
   #include <errno.h>
   #define ECHOMAX 8

   using namespace cv;
26 using namespace std;

   IplImage *src;
   int width = 720;
   int height = 576;
31
   Mat rotateImage(const Mat& source, double angle)
   {
       Point2f src_center(source.cols/2.0F, source.rows/2.0F);
       Mat matrix_rotate = getRotationMatrix2D(src_center, angle,
36         1.0);
       Mat img_rotate;
       warpAffine(source, img_rotate, matrix_rotate, source.size());
       return img_rotate;
   }

41 bool cmp_wrt_theta(const Vec2f &a, const Vec2f &b)

```

```

{
    return a[1] < b[1];
}

46 bool cmp_wrt_absrho(const Vec2f &a, const Vec2f &b)
{
    return abs(a[0]) < abs(b[0]);
}

51 void draw_line(float rho, float theta, CvScalar color, Mat &
    select_area)
{
    double a = cos(theta), b = -sin(theta);
    double x0 = a*rho, y0 = b*rho;

56     Point pt1( cvRound(width/2 + x0 + 1000*(-b)), cvRound(height
        /2 - y0 - 1000*(a)) );
    Point pt2( cvRound(width/2 + x0 - 1000*(-b)), cvRound(height
        /2 - y0 + 1000*(a)) );
    line(select_area, pt1, pt2, color, 10, 8);
}

61 void find_fixed_target()
{
    v4l2_input_layer_t layah;
    char name[20] = "/dev/video0";

66     layah.device_name = name;
    layah.input_index = 0;
    layah.n_buffers = 0;
    layah.enable_debug_time = false;
    layah.lower_layer = 0; /* quello che mi elimina il layer
        inferiore*/

71     Rect frameSize (0,0,width,height);
    frame_t fram;
    fram.full_height = height;
    fram.full_width = width;

76     layah.frame_info = &fram;
    v4l2_input_start_layer(&layah);
    v4l2_input_process(&layah);

81     namedWindow( "Immagine originale", CV_WINDOW_AUTOSIZE);

    src = cvCreateImage(cvSize(width,height), IPL_DEPTH_8U, 3);

    unsigned char mypixel[width * height * 3] ;

86     Mat frame(src);
    Mat dst, cdst, img_rotate, rot_matrix;
    vector<Vec2f> lines;

```

```

91  struct sockaddr_in echoServAddr2;
    int sendMsgSize;
    echoServAddr2.sin_family = AF_INET;
    echoServAddr2.sin_addr.s_addr = inet_addr("192.168.1.11");
    echoServAddr2.sin_port = htons(12000);
96  char echoString[8];
    sprintf(echoString, "%d", 10);
    int sock;

    struct sockaddr_in data_from, client_add;
101  unsigned int cliAddrLen;
    int recvMsgSize;
    memset(&data_from, 0, sizeof(data_from));
    data_from.sin_family = AF_INET;
    data_from.sin_addr.s_addr = inet_addr("192.168.1.10"); //mio
        indirizzo
106  data_from.sin_port = htons(12001);
    char recString[ECHOMAX];
    int sock1;

    double angle =0;

111  //CREAZIONE SOCKET NECESSARIA PER LA COMUNICAZIONE
    if ((sock=socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0)
        printf("socket() failed.");

116  if ((sock1=socket(PF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0)
        printf("socket() failed.");

    if(bind(sock1, (struct sockaddr *)&data_from, sizeof(
        data_from)) < 0)
        printf("bind() failed");

121  for(;;)
    {
        v4l2_input_process(&layah);

126  //CONVERSIONE DEL FRAME IN FORMATO BGR
        UYVY2BGR(layah.frame, width * height * 2 , mypixel );

        memcpy(src->imageData, mypixel, width * height *3 );

131  Mat select_area = frame;

        cliAddrLen = sizeof(client_add);
        if((recvMsgSize=recvfrom(sock1, recString, ECHOMAX, 0, (
            struct sockaddr *)&client_add, &cliAddrLen)) < 0)
        {
136  cout<<"ERROR " <<recvMsgSize<<endl;
        }
        else

```

```

    {
        angle = *(float *)recString;
141    }

    //ROTAZIONE IMMAGINE PER COMPENSARE MOVIMENTO DI ROLL
    select_area = rotateImage(select_area, angle-4);
    cvtColor(select_area, dst, CV_BGR2GRAY);
146

    //ALGORITMO DI RICONOSCIMENTO DELLE LINEE
    Canny(dst, cdst, 40, 50);
    HoughLines(cdst, lines, 1, CV_PI/180, 100);

151    int i=0;
    while(i < lines.size())
    {
        double rho = lines[i][0], theta = lines[i][1];
        rho = rho - (width/2)*cos(theta) - (height/2)*sin(
            theta);
156

        if(abs(rho) > 200)
        {
            lines.erase(lines.begin()+i);
        }
161        else
        {
            lines[i][0] = rho;
            i++;
        }
166    }

    float rhoh = 0, thetah = 0, rhov = 0, thetav = 0;
    int Nh = 0, Nv = 0;

171    for(int k=1; k<lines.size(); k++)
    {
        if(lines[k][1] > CV_PI/4 && lines[k][1] < 3*CV_PI/4
            )
        {
            Nh++;
            rhoh += lines[k][0];
            thetah += lines[k][1];
        }
        else
181        {
            Nv++;
            if(lines[k][1] > 3*3.14/4)
            {
                rhov -= lines[k][0];
                thetav += (lines[k][1]-3.14);
186            }
            else
            {

```

```

        rhov += lines[k][0];
        thetav += lines[k][1];
191     }
    }
}
if(Nh > 0)
{
196     rhoh = rhoh/Nh;
    thetah = thetah/Nh;
    draw_line(rhoh, thetah, Scalar(0,0,255), select_area)
        ;
}
if(Nv > 0)
201 {
    rhov = rhov/Nv;
    thetav = thetav/Nv;
    if(thetav<0)
    {
206         rhov = -rhov;
        thetav += 3.14;
    }
    draw_line(rhov,thetav,Scalar(0,0,255), select_area);
}
211
i=0;

//RICERCA DEL PUNTO DI INCONTRO DELLE RETTE INDIVIDUATE
float coord_x = ( sin(thetav)*rhoh - sin(thetah)*rhov )/(
    cos(thetah)*sin(thetav) - sin(thetah)*cos(thetav));
216 float coord_y = -( -cos(thetav)*rhoh + cos(thetah)*rhov )
    /(cos(thetah)*sin(thetav) - sin(thetah)*cos(thetav));

//INVIO DEI DATI AL SISTEMA DI CONTROLLO
sprintf(echoString, "%04d%04d%04d",
    (int)((width/2+coord_x)*10), (int)((height/2-
        coord_y)*10),
221 (int)(thetah*1000));
cout<<echoString<<endl;
if((sendMsgSize = sendto(sock, echoString, strlen(
    echoString), 0, (struct sockaddr *)&echoServAddr2,
    sizeof(echoServAddr2))) < 0)
{
226     cout<<"ERROR " <<sendMsgSize<<endl;
}

//TRACCIAMENTO CERCHIO CON CENTRO NEL CENTRO DELLA CROCE
circle(select_area, cvPoint(width/2+coord_x,height/2-
    coord_y), 50, Scalar (0,255,0),5);

231 //TRACCIAMENTO CROCE NEL CENTRO IMMAGINE
line(select_area, cvPoint(width/2,height/2-50),cvPoint(
    width/2,height/2+50),Scalar (0,255,255),5);

```

```

        line(select_area, cvPoint(width/2-50,height/2),cvPoint(
            width/2+50,height/2),Scalar (0,255,255),5);

        imshow( "Immagine originale", select_area);
        waitKey(20);
    }
}

```

A.2 S-FUNCTION PER LEGGERE DATI DALLA IMU

```

#define S_FUNCTION_NAME  sfun_parser_bin
2 #define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include <malloc.h>
#include <string.h>
7
/* S-function defs */

#define NUM_INPUTS        1
#define NUM_OUTPUTS       3
12 #define NUM_PARAMS      0

#define SAMPLE_TIME_0      INHERITED_SAMPLE_TIME
#define NUM_DISC_STATES    1
#define DISC_STATES_IC     [0]
17 #define NUM_CONT_STATES  0
#define CONT_STATES_IC     [0]

#define INPUT_0_WIDTH      1
#define INPUT_0_FEEDTHROUGH 1
22

#define OUTPUT_0_WIDTH     3
#define OUTPUT_1_WIDTH     1
#define OUTPUT_2_WIDTH     1

27 static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, NUM_PARAMS); /* Number of expected
        parameters */
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        /* Return if number of expected != number of actual
            parameters */
32     return;
    }

    ssSetNumContStates(S, NUM_CONT_STATES);
    ssSetNumDiscStates(S, NUM_DISC_STATES);
37
    if (!ssSetNumInputPorts(S, NUM_INPUTS)) return;

```

```

    if (!ssSetNumOutputPorts(S, NUM_OUTPUTS)) return;

42  /*Input Port 0*/
    ssSetInputPortWidth(S, 0, INPUT_0_WIDTH);
    ssSetInputPortDataType(S, 0, SS_UINT8);
    ssSetInputPortDirectFeedThrough(S, 0, INPUT_0_FEEDTHROUGH);
    ssSetInputPortRequiredContiguous(S, 0, 1);

47  /*Output Port 0*/
    ssSetOutputPortWidth(S, 0, OUTPUT_0_WIDTH);
    ssSetOutputPortDataType(S, 0, SS_DOUBLE);

52  /*Output Port 1*/
    ssSetOutputPortWidth(S, 1, OUTPUT_1_WIDTH);
    ssSetOutputPortDataType(S, 0, SS_DOUBLE);

    /*Output Port 2-debug*/
57  ssSetOutputPortWidth(S, 2, OUTPUT_2_WIDTH);
    ssSetOutputPortDataType(S, 0, SS_DOUBLE);

    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S, 0);
62  ssSetNumIWork(S, 0);
    ssSetNumPWork(S, 3);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);

67  ssSetOptions(S, 0);
}

static void mdlInitializeSampleTimes(SimStruct *S)
{
72  ssSetSampleTime(S, 0, SAMPLE_TIME_0);
    ssSetOffsetTime(S, 0, 0.0);
}

#define MDL_INITIALIZE_CONDITIONS
77 #if defined(MDL_INITIALIZE_CONDITIONS)
    static void mdlInitializeConditions(SimStruct *S)
    {
        real_T *x = ssGetRealDiscStates(S);

82     x[0] = 0;
    }
#endif /* MDL_INITIALIZE_CONDITIONS */

#define MDL_START /* Change to #undef to remove function */
87 #if defined(MDL_START)
    static void mdlStart(SimStruct *S)
    {
        void **PWork = ssGetPWork(S);
        PWork[0] = (void *) calloc(1, sizeof(uint8_T));
    }

```

```

92     PWork[1] = (void *) calloc(3, sizeof(real_T));
    PWork[2] = (void *) calloc(12, sizeof(uint8_T));
    }
#endif

97 static void mdlOutputs(SimStruct *S, int_T tid)
{
    real_T    *y = ssGetOutputPortSignal(S,0);
    real_T    *data_ready = ssGetOutputPortSignal(S,1);
    real_T    *output_debug = ssGetOutputPortSignal(S,2);
102    real_T    *x = ssGetRealDiscStates(S);

    uint8_T *bcnt = (uint8_T *)ssGetPWorkValue(S,0);

    real_T    *yprev = (real_T *)ssGetPWorkValue(S,1);
107    uint8_T *buff = (uint8_T *)ssGetPWorkValue(S,2);

    if( (x[0] == 4) && (*bcnt == 11) )
    {
112        y[0] = (real_T) (*(real32_T *)buff);
        y[1] = (real_T) (*(real32_T *)buff+4);
        y[2] = (real_T) (*(real32_T *)buff+8);

        memcpy(yprev, y, 3*sizeof(real_T));
117        *data_ready = 1;
    }

    else
    {
122        memcpy(y, yprev, 3*sizeof(real_T));
        *data_ready = 0;
    }
    /*output_debug = *x;
    *output_debug = (real_T) *bcnt;
127 }

#define MDL_UPDATE /* Change to #undef to remove function */
#if defined(MDL_UPDATE)
    static void mdlUpdate(SimStruct *S, int_T tid)
132 {
        real_T    *x = ssGetRealDiscStates(S);
        const uint8_T *u = (const uint8_T*) ssGetInputPortSignal(S
            ,0);

        uint8_T *bcnt = (uint8_T *)ssGetPWorkValue(S,0);
137        uint8_T *buff = (uint8_T *)ssGetPWorkValue(S,2);

        switch((uint8_T)x[0])
        {
142            case 0:

```



```

        if(*u == 0xFF)
            x[0] = 1;
        else
            x[0] = 0;
        break;
147
    case 1:
        if(*u == 0x00)
            x[0] = 2;
        else
            x[0] = 0;
        break;
152
    case 2:
        if(*u == 0xFF)
            x[0] = 3;
        else
            x[0] = 0;
        break;
157
    case 3:
        if(*u == 0x00)
            x[0] = 4;
        else
            x[0] = 0;
        break;
162
    case 4:
        buff[*bcnt] = *u;
        if(*bcnt < 11)
        {
            *bcnt= (*bcnt)+1;
            x[0] = 4;
        }
        else
        {
            *bcnt = 0;
            x[0] = 0;
        }
        break;
167
    }
172
}
177
}
182
#endif /* MDL_UPDATE */
187
#undef MDL_DERIVATIVES /* Change to #undef to remove function */
#if defined(MDL_DERIVATIVES)
    static void mdlDerivatives(SimStruct *S)
    {
192    }
#endif /* MDL_DERIVATIVES */

```

```

static void mdlTerminate(SimStruct *S)
{
197     if (ssGetPWork(S) != NULL)
        {
            free(ssGetPWorkValue(S,0));
            ssSetPWorkValue(S,0,NULL);

202         free(ssGetPWorkValue(S,1));
            ssSetPWorkValue(S,1,NULL);

            free(ssGetPWorkValue(S,2));
            ssSetPWorkValue(S,2,NULL);
207     }
}

#ifdef MATLAB_MEX_FILE
#include "simulink.c"
212 #else
#include "cg_sfun.h"
#endif

```

A.3 FILE MATLAB PER IL DESIGN DEL CONTROLLORE INERZIALE

```

1 %% Build mex files

if exist('sfun_parser_bin','file') ~= 3,
    fprintf('Build sfun_parser_bin MEX file ... ');
    mex sfun_parser_bin.c Winio.lib;
6    fprintf('Done.\n');
end;

%% Get plant model

11 load idres_PAN;

% Add double integrator input
sysI = zpk([], 1, Ts, Ts);
sysP = sysPi*sysI^2;
16

%% Controller design (Bode's method)

% crossover freq + phase margin defs
wgc = 2*pi*1;
21 phim = pi/180*80;

% compute required gain + phase adjustments
[magP, phaP] = bode(sysP, wgc);
phaP = pi/180*(phaP-360);
26

dmag = 1/magP;
dphi = phim-pi-phaP;

```

```

% compute controller params
31 tau = Ts*tan(dphi)/(sin(wgc*Ts)-(cos(wgc*Ts)-1)*tan(dphi));
   sysC1 = 1 + zpk(1, [], tau/Ts, Ts);

   [magC1, phaC1] = bode(sysC1, wgc);

36 Kp_PAN = dmag/magC1;
   Kd_PAN = tau*Kp_PAN;

%% Get plant model

41 load idres_TILT;

% Add double integrator input
   sysI = zpk([], 1, Ts, Ts); % discrete-time integrator
   sysP = sysPi*sysI^2;

46

%% Controller design (Bode's method)

% crossover freq + phase margin defs
51 wgc = 2*pi*0.2;
   phim = pi/180*80;

% compute required gain + phase adjustments
   [magP, phaP] = bode(sysP, wgc);
56 phaP = pi/180*(phaP-360);

   dmag = 1/magP;
   dphi = phim-pi-phaP;

61 % compute controller params
   tau = Ts*tan(dphi)/(sin(wgc*Ts)-(cos(wgc*Ts)-1)*tan(dphi));
   sysC1 = 1 + zpk(1, [], tau/Ts, Ts);

   [magC1, phaC1] = bode(sysC1, wgc);

66 Kp_TILT = dmag/magC1;
   Kd_TILT = tau*Kp_TILT;

```


BIBLIOGRAFIA

- [1] Scanning the Issue/Technology - Special Issue on Video Communications, Processing, and Understanding for Third Generation Surveillance Systems. *Proceedings of the IEEE*, 89(10), October 2001.
- [2] Z. Hurak and M. Rezac. Image-Based Pointing and Tracking for Inertially Stabilized-Airborne Camera Platform. *Control System Technology, IEEE Transactions on*, 20(5):1146–1159, September 2012.
- [3] Jonathan Connell Ahmet Ekin Norman Haas Max Lu Hans Merkl Sharath Pankanti Andrew Senior Chiao-Fe Shu Arun Hampapur, Lisa Brown and Ying Li Tian. Smart Video Surveillance - Exploring the concept of multiscale spatiotemporal tracking. *IEEE Signal Processing Magazine*, 20(5):38–51, 2005.
- [4] S. Ghidoni B. Lain and E. Menegatti. *Robust Object Tracking in a dual camera sensor*. Human System Interactions ,2001 4th International Conference, may 2011. pp. 150–157.
- [5] Gary Bradski and Adrian Kaehler. *Learning OpenCV- Computer Vision with the OpenCv library*. 2008.
- [6] Riccardo Crestani. *Sistema di visione ad alte prestazioni per il controllo di fluidi in movimento*. 2012.
- [7] Cristian Fontani. *Sviluppo di algoritmi per asservimenti visivi basati su geometria epipolare con libreria open source intel opencv*. pages 27–28, 2004.
- [8] Brian Hall. *Beenj's Guide To Network Programming*. 2011.
- [9] Francois Chaumette Seth Hutchinson. *Visual Servo Control Part I-Basic Approaches*. 2007.
- [10] M. Ulvklo J. Nygards, P.Skoglar and T. Hogstrom. Navigation Aided Image Processing in Uav Surveillance: Preliminary Results and Design of an Airborne Experimental System. *Journal of Robotic Systems*, 21(2):63–72, 2004.
- [11] Davide Pilastro. *Stabilizzazione dell'immagine in brandeggi mobili per videosorveglianza: Applicazione di tecniche basate sull'uso di sensori inerziali*. 2012.
- [12] Roberto De Rossi. *Studio e applicazione della trasformata di hough per il riconoscimento di target umani*. 2011.

- [13] Mattia Schiesaro. Analisi di prestazione di un brandeggio per applicazioni di videosorveglianza. 2012.
- [14] Inc Techwell. TW6805-66MH PCI Video Decoder with Audio and TS Input Support. August 2007.
- [15] J. L Thompson and R.A.N. Research Laboratory. Ship Motion Measurements Made on an Attack Class Patrol Boat. Report, 2002.
- [16] Christian Tobaldo. Software per la gestione remota di un sistema di teleriabilitazione. pages 21–26, 2010.
- [17] Simone Viola. Stabilizzazione dell'immagine in brandeggi mobili per videosorveglianza: Applicazione di tecniche basate sull'elaborazione dell'immagine. 2012.
- [18] Prof. Mauro Zigliotto. Dispense del corso di macchine e azionamenti elettrici. *Part 10.0 - Azionamenti con motore a passo.*