

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

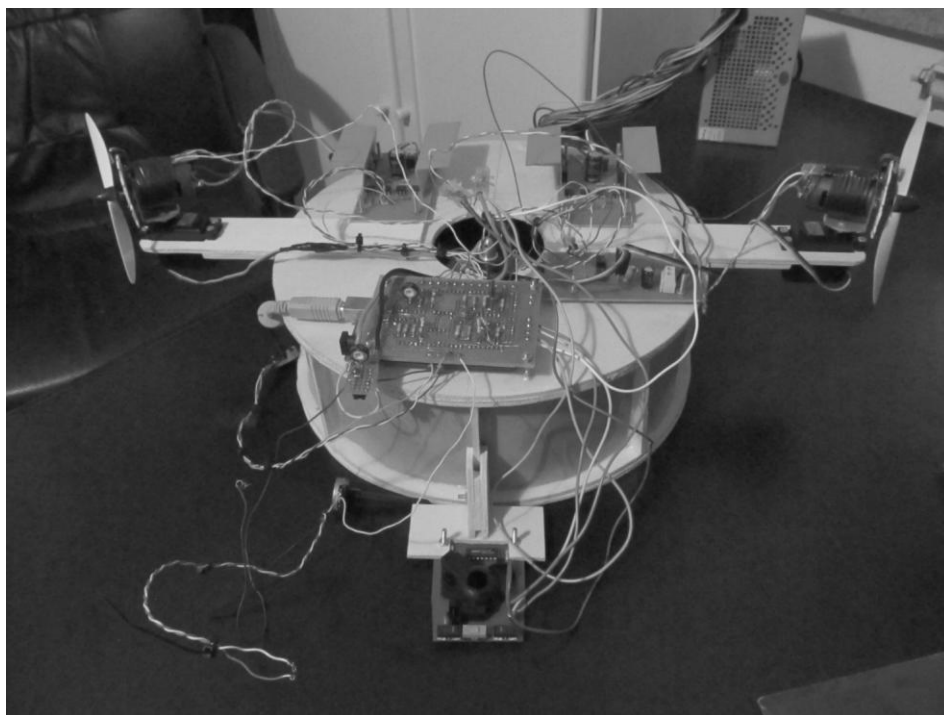


Facoltà di Ingegneria

CORSO DI LAUREA IN
INGEGNERIA DELL'AUTOMAZIONE

Corso di Sistemi di Controllo Multivariabile

Controllo di stabilità di un modellino hovercraft



MATTEO DEL CASTELLO
MATR. M58/29

ROSARIO MELITO
MATR. M58/34

ANNO ACCADEMICO 2011/2012

INDICE

1. Introduzione	pag.3
2. Hovercraft, principio di funzionamento	pag.4
3. Elettronica di bordo e attuatori	pag.7
4. Derivazione modello matematico	pag.17
5. Progetto del controllore	pag.25
6. Appendice A	pag.37
7. Appendice B	pag.43

1. Introduzione

Il lavoro presentato consiste nel progetto di un regolatore allo scopo di stabilizzare e controllare in traiettoria un piccolo modello di hovercraft. Un hovercraft è un mezzo anfibio che sostenuto da un cuscino d'aria e quindi non a contatto diretto con il terreno sottostante è in grado di spostarsi su superfici di vario genere tra cui acqua, fango, asfalto ecc.. L'idea risale al sedicesimo secolo anche se, causa l'impossibilità di sviluppare potenze sufficienti al sostentamento e alla propulsione e le scarse conoscenze di fluidodinamica possedute, le prime realizzazioni pratiche risalgono alla metà del secolo scorso.

2. Hovercraft, principio di funzionamento

Come detto in precedenza l'hovercraft è un mezzo sostenuto da un cuscino d'aria per cui, non essendo a diretto contatto con la superficie terrestre potrebbe essere considerato un aeromobile anche se la spinta di sustentamento è generata in modo radicalmente diverso dai comuni oggetti volanti quali aerei o elicotteri. Con riferimento alla figura 1 nel caso dell'hovercraft la spinta è generata da un compressore (3) posto sulla base del veicolo e il flusso d'aria (2) da esso generato è confinato, tramite dei condotti che sfogano lungo il perimetro, sotto la base dello stesso. Nelle realizzazioni industriali è solitamente presente una robusta gonna di gomma (4) rinforzata da materiali con elevata resistenza a trazione, similmente a quanto avviene negli pneumatici automobilistici, allo scopo di contenere l'aria ad alta pressione al variare delle condizioni della superficie sottostante. La spinta necessaria al movimento del veicolo è generata invece dalle eliche (1). La direzione viene controllata da un operatore o variando la direzione di spinta delle eliche o disponendo un sistema di timoni che indirizzino la spinta secondo il loro angolo di rotazione.

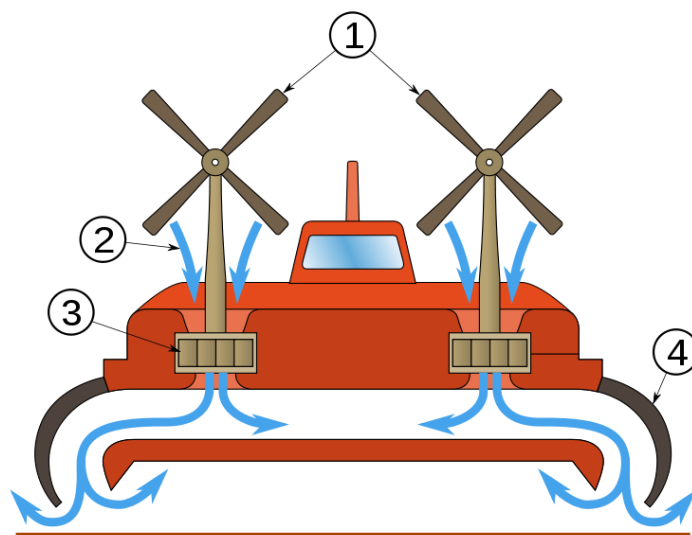


Figura 1 - Schematizzazione delle varie componenti di un hovercraft

I vantaggi di tale sistema di propulsione e di sostentamento derivano dal fatto che l'hovercraft, realizzabili in dimensioni che vanno dai pochi metri alle diverse decine di metri può muoversi di fatto su qualunque superficie sufficientemente regolare, indipendentemente dalla consistenza, (gli hovercraft, anche di grandi dimensioni, esercitano una pressione al suolo paragonabile a quella esercitata da un essere umano) a notevoli velocità.



Figura 2 - Hovercraft militare

La manovrabilità del mezzo è però piuttosto scarsa se si considera che, trascurando gli inevitabili fenomeni di attrito, la forza centripeta necessaria a percorrere una traiettoria curva è fornita unicamente dalle componenti radiali delle forze di spinta generata dalle eliche, normalmente governate da esseri umani.

Il problema di cui ci si occupa è quello di progettare un'unità di controllo per un piccolo modello di hovercraft realizzato in compensato e con attuatori comunemente utilizzati nell'ambito del modellismo aeronautico. La struttura è semplificata rispetto al caso reale principalmente a causa dell'assenza della gonna sopra descritta e alla disposizione dei propulsori. La presenza della gonna infatti è necessaria unicamente per adattarsi alla

disomogeneità della superficie e aggiungerebbe, per l'elasticità del cuscino d'aria ulteriori gradi di libertà al modello che non potrebbero essere trascurati. La scelta effettuata di una struttura completamente rigida rende invece trascurabili le rotazioni lungo gli assi di rollio e beccheggio nonché le traslazioni lungo l'asse verticale permettendo di pervenire ad un più semplice modello matematico e di conseguenza ad un più semplice controllore. La disposizione degli attuatori rende più agevole la generazione di forze e coppie necessarie al controllo.

3. Elettronica di bordo e attuatori

Il sistema di controllo è basato sulla piattaforma open source Arduino mega 2560. Tale dispositivo è a sua volta basato sul microcontrollore Atmel AVR ATmega 2560 a 8 bit che dispone di numerose periferiche di input/output sia analogiche che digitali oltre ai preziosi contatori e timer utili alla generazione di segnali modulati in ampiezza di impulsi, temporizzazione e conteggio di eventi esterni.

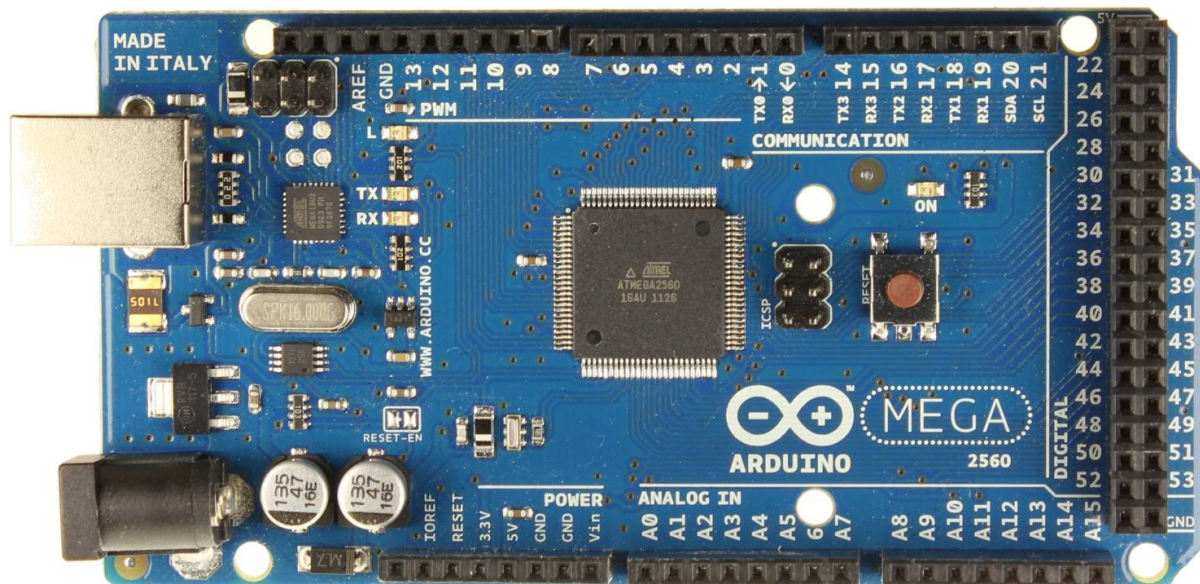
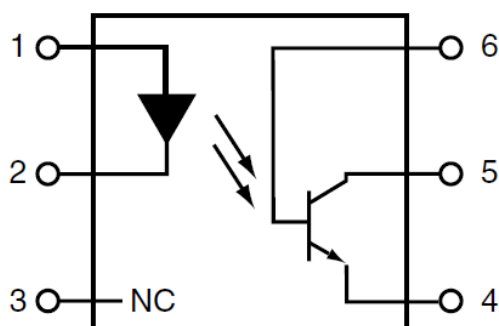


Figura 3 - Scheda Arduino 2560

Il vantaggio derivante dall'impiego di Arduino deriva dall'immediata disponibilità di un'unità di programmazione del microcontrollore e di un circuito stampato che rende semplice l'accesso alle periferiche sopra elencate, senza considerare le difficoltà che deriverebbero dalla saldatura dell'integrato non disponendo di adeguate attrezzature. Inoltre l'ambiente di sviluppo mette anche a disposizione un'interfaccia di comunicazione seriale utile al trasferimento di dati provenienti dalle periferiche di acquisizione del microcontrollore al PC. Per quanto riguarda tutto il resto dell'elettronica necessaria all'acquisizione delle

grandezze di stato provenienti dai sensori e la distribuzione dei segnali di potenza agli attuatori i circuiti stampati sono stati realizzati autonomamente. In particolare si sono resi necessari due driver H-bridge per il comando dei motori a corrente continua che movimentano le eliche direzionabili posti sui lati dell'hovercraft e una scheda per il comando dei servomotori che provvedono alla rotazione dei propulsori. La complessità di tali circuiti è piuttosto modesta per cui ci si limiterà a descriverne superficialmente le caratteristiche. Il compressore per il sostentamento è collegato direttamente al sistema di alimentazione dovendo funzionare a potenza costante. Esso non è altro che un motore CC sul cui asse è calettato l'ogiva che ospita un elica da modellismo. L'utilità delle schede di controllo deriva anche dalla necessità di disaccoppiare la circuiteria di potenza da quella di segnale in primo luogo perché funzionanti a livelli di tensione differenti e poi per ridurre i disturbi di massa del circuito di controllo vero e proprio. I circuiti integrati necessari a fare ciò sono i fotoaccoppiatori 4n35.



PIN 1. ANODE
 2. CATHODE
 3. NO CONNECTION
 4. EMITTER
 5. COLLECTOR
 6. BASE

Figura 4 - Schematizzazione del circuito fotoaccoppiatore

Un altro circuito stampato è quello che alloggia direttamente sulla scheda Arduino ed ospita i sensori inerziali e i circuiti di condizionamento necessari a quest'ultimi oltre a rendere possibile l'interfacciamento con gli altri sensori necessari. In particolare per la misurazione della velocità delle eliche necessarie a generare le forze trainanti e alla chiusura di un anello di controllo in velocità dei motori elettrici ad esse collegate si sono resi necessari degli encoder incrementali ricavati da un vecchio mouse. Per essere conteggiati, gli impulsi provenienti dalle fotocellule sono preventivamente squadrati da un comparatore di tensione ad isteresi che alloggia sulla scheda suddetta. Oltre alle misure inerziali è disponibile una misura dello spostamento effettuata interfacciando Arduino con un vecchio mouse Ps2 a pallina.

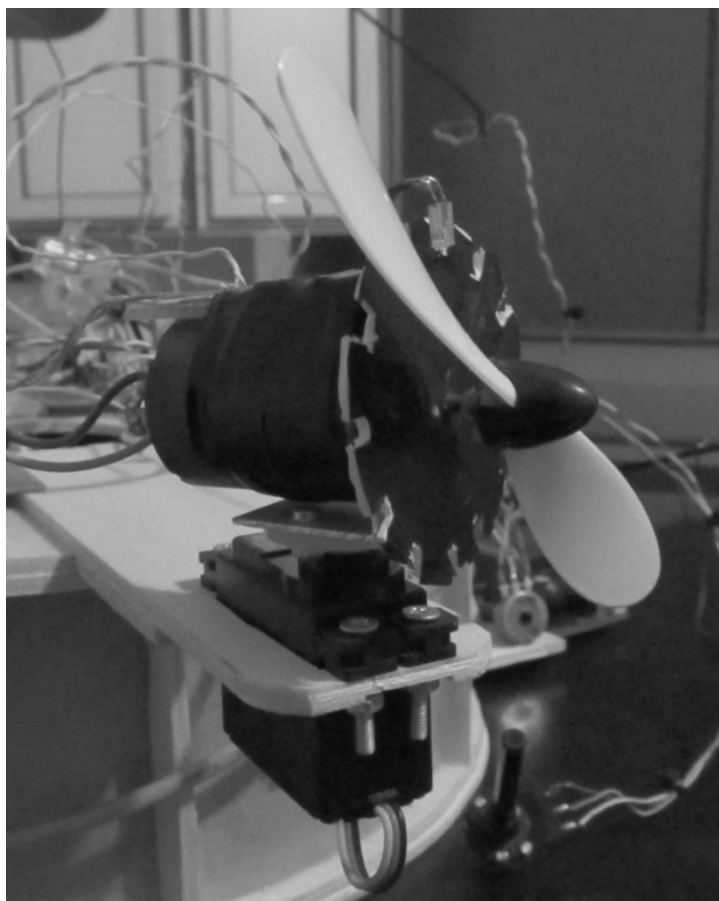


Figura 5 - Dettaglio elica, servomotore ed encoder incrementale

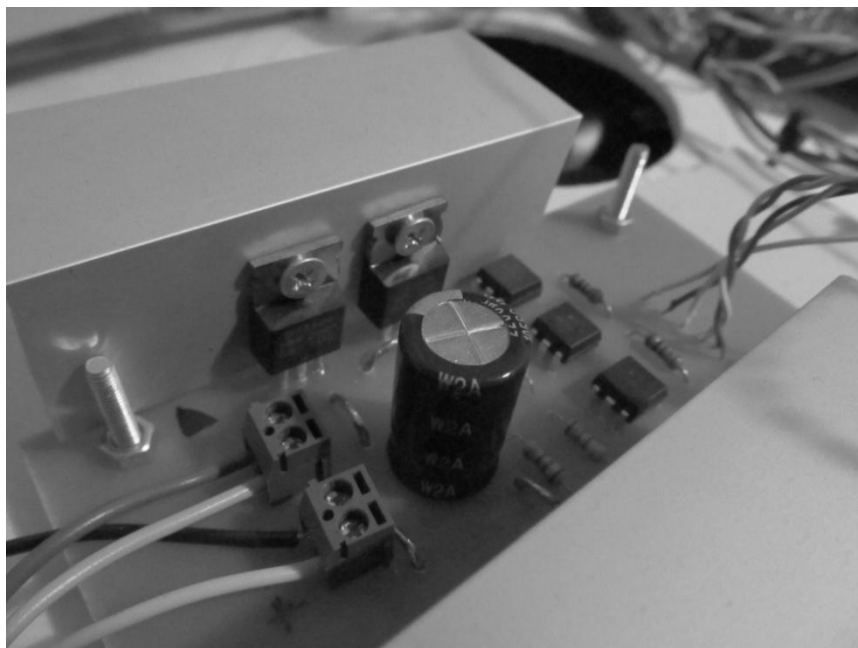


Figura 6 - Scheda controllo motori (H-Bridge)



Figura 7 - Dettaglio sensore di posizione (PS/2 Mouse)

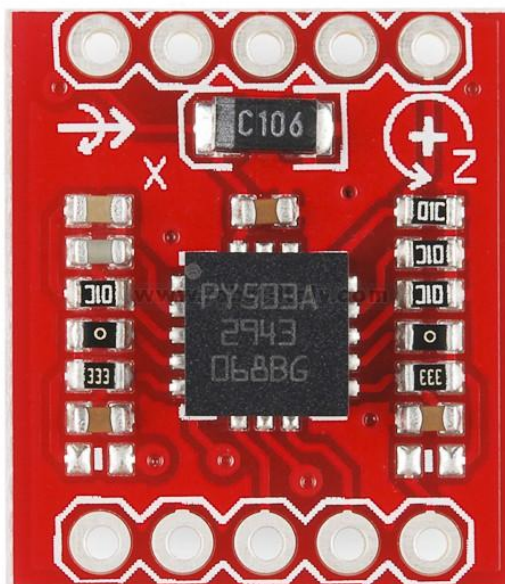


Figura 8 – Giroscopio



Figura 9 - Dettaglio elica di sostentamento

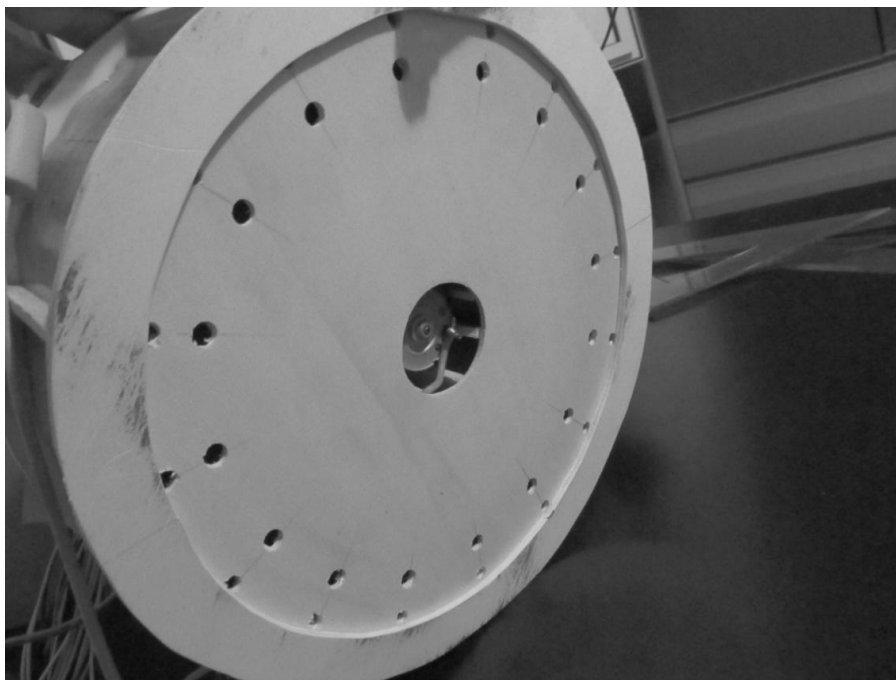


Figura 10 - Fondo per la generazione del cuscino d'aria

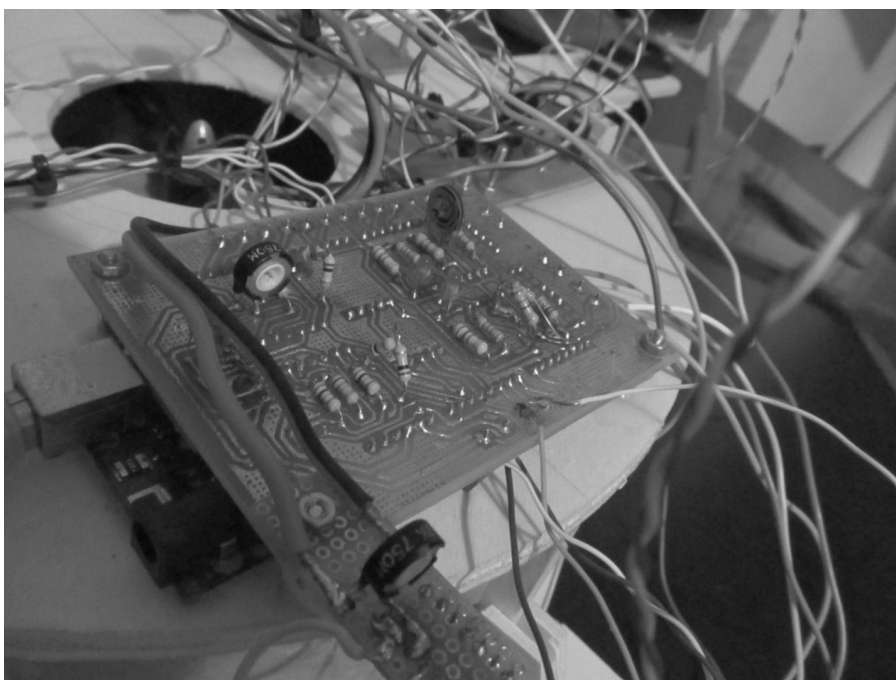


Figura 11 - Dettaglio scheda sensori e distribuzione segnali



Figura 12 - Sistema di alimentazione

Prima di procedere è il caso di fare alcune considerazioni sugli attuatori. Innanzitutto per quanto riguarda i servomotori, comunemente adottati nell'ambito del aeromodellismo, della robotica di consumo ecc. va detto che nonostante sia già implementato al loro interno un sistema di controllo a controreazione che garantisce entro certi limiti la correttezza del posizionamento non è possibile esprimere il legame tra riferimento e uscita tramite un'equazione differenziale lineare in quanto il loro sistema di controllo è basato su relè. Dal punto di vista del sistema di controllo complessivo inoltre i servomotori pongono un limite sulla frequenza di campionamento utilizzabile. Ciò è dovuto al fatto che il riferimento deve essere modulato con la tecnica PWM con una portante a 50Hz. Inoltre da considerazioni basate sull'esperienza deriva che è il caso che il periodo di campionamento del sistema di controllo sia in relazione armonica con la portante del segnale di riferimento.

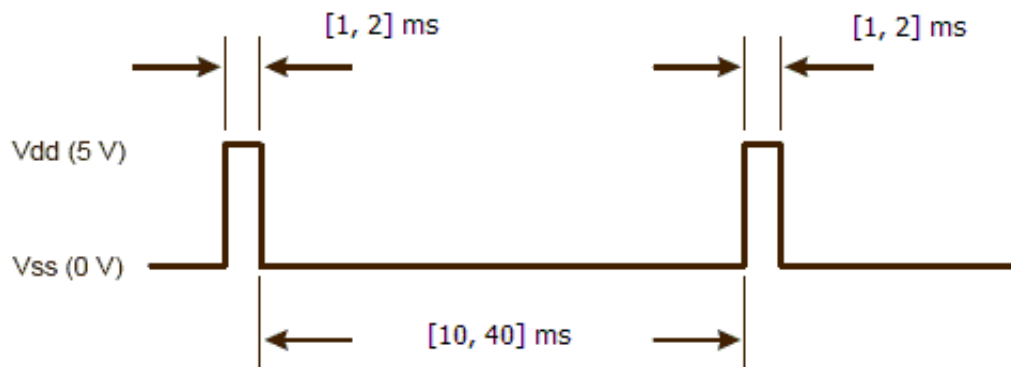


Figura 13 - Segnale di riferimento servomotore

Ciononostante la modellazione del servomotore sarà comunque effettuata tramite funzione di trasferimento considerata per semplicità del primo ordine.

Per quanto riguarda i motori elettrici che movimentano le eliche si è effettuato un processo di identificazione basato sull'apposito pacchetto messo a disposizione da Matlab, system identification tool, a partire dall'acquisizione di un segnale di riferimento casuale e dall'uscita ad esso corrispondente. Di seguito sono riportati gli andamenti temporali di tali grandezze.

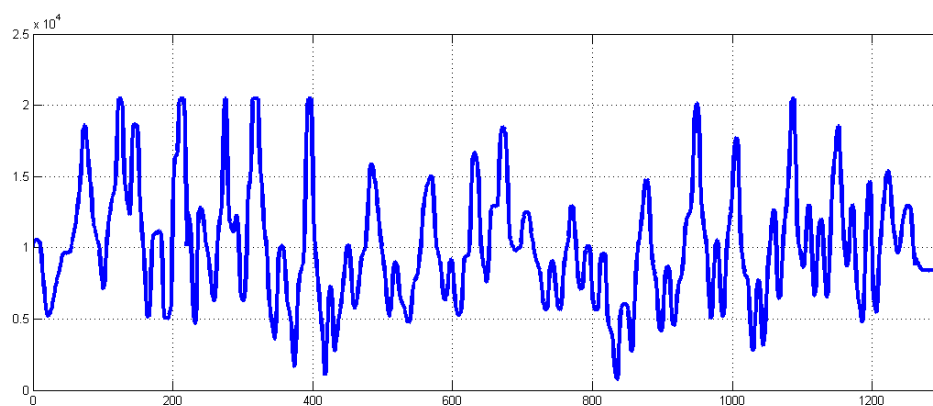


Figura 14 - Andamento del riferimento

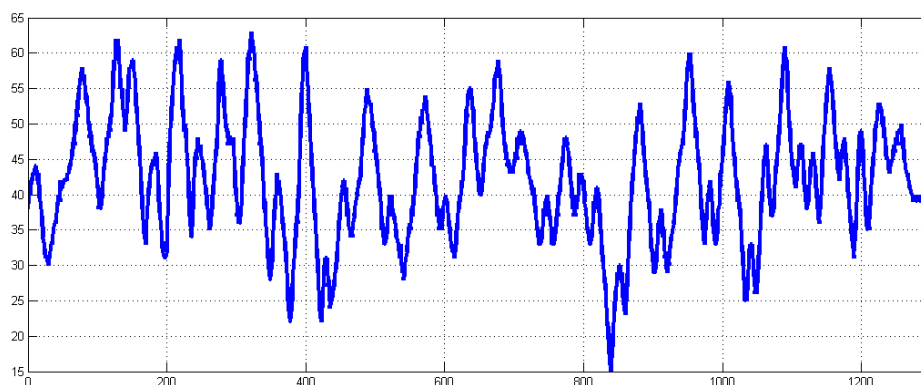


Figura 15 - Andamento dell'uscita misurata

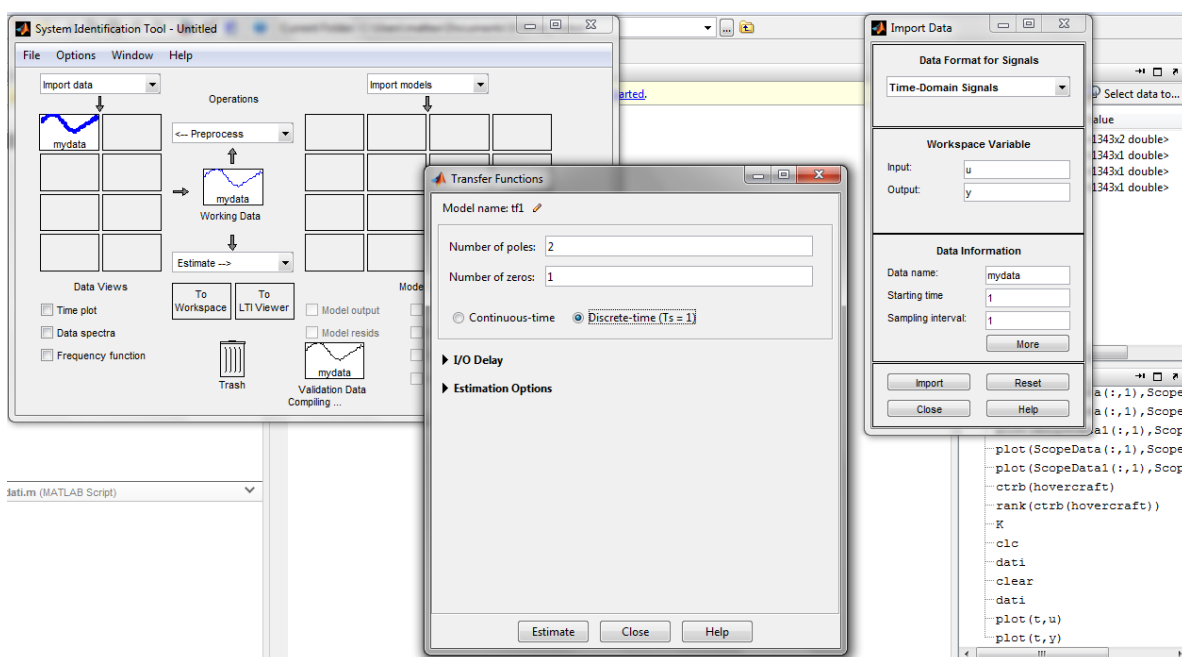


Figura 16 - Tool di identificazione

Procedendo come mostrato in Figura 16 si giunge all'identificazione della funzione di trasferimento TD seguente:

$$\frac{N(z^{-1})}{R(z^{-1})} = \frac{0.0004013 - 0.0004 z^{-1}}{1 - 1.843 z^{-1} + 0.8431 z^{-2}}$$

Che rappresenta il legame differenziale esistente tra segnale di alimentazione e numero di giri (scalati opportunamente) che caratterizza i propulsori dell'hovercraft. Il tempo di campionamento coincide con quello del sistema di controllo. Nella figura successiva è riportato il risultato di una simulazione effettuata sul sistema identificato confrontato con il segnale proveniente dai sensori. Le differenze che si notano nel primo tratto del grafico sono da attribuire anche alla modalità di simulazione che non tiene conto delle condizioni iniziali diverse da zero.

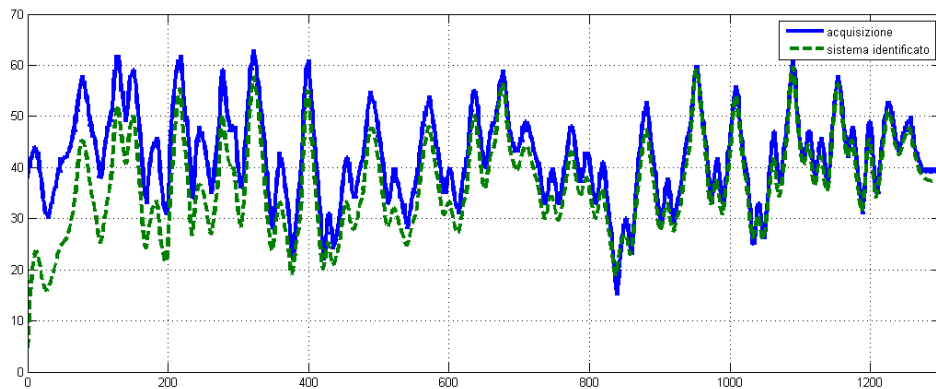


Figura 17 - Simulazione sul sistema identificato

Sulla base del modello identificato è stato progettato un regolatore PI in grado di garantire il corretto numero di giri delle eliche e modificarne la dinamica. Si omettono i dettagli del progetto poiché le tecniche utilizzate non risultano interessanti per questo corso. Infine è il caso di dire che così come i servomotori, all'interno delle simulazioni atte alla validazione del controllore le dinamiche di tali attuatori sono state considerate del primo ordine e lineari.

4. Derivazione del modello matematico

Con riferimento alla figura deriviamo il sistema di equazioni che governano la dinamica dell'hovercraft.

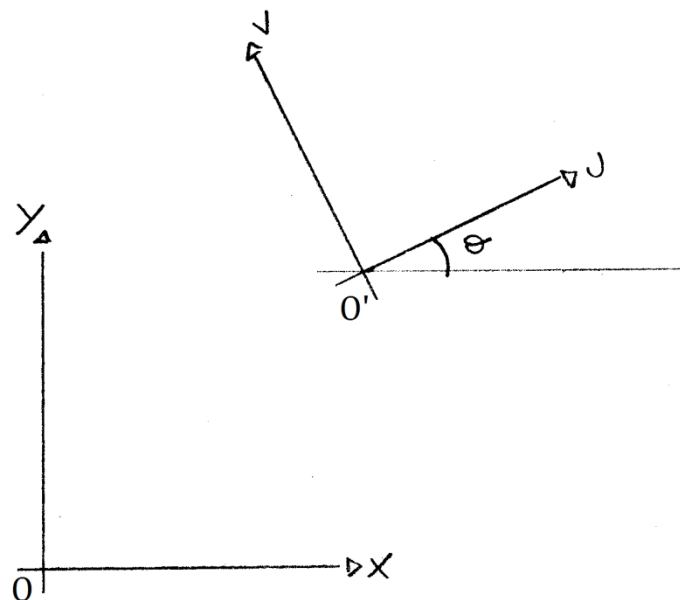


Figura 18 - Sistemi di coordinate fisso e solidale

Se consideriamo il sistema OXY di riferimento inerziale, indicando con F_u e F_v le componenti della forza sugli assi del riferimento O'UV solidale all'hovercraft, e trascurando fenomeni dovuti alle forze d'attrito, si ottiene il seguente sistema di equazioni:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} F_u \\ F_v \end{bmatrix}$$

$$\ddot{\vartheta} = \frac{M}{I}$$

Dove con m si è indicata la massa del sistema, con I si è indicato il momento di inerzia di massa, con M il momento torcente agente sul sistema, mentre, come si può dedurre dalla figura, ϑ indica l'angolo di rotazione del sistema di riferimento solidale rispetto al fisso.

Il modello matematico ricavato in precedenza, esprime i valori del sistema in un riferimento inerziale, ma poiché le grandezze in gioco vengono misurate nel sistema solidale, sorge la necessità di trovare un legame che permetta di riportare le informazioni da un sistema all'altro.

L'equazione successiva rappresenta le componenti di velocità del sistema solidale rispetto a quello fisso. Nel sistema di riferimento fisso leggo lo stesso vettore velocità ma con componenti differenti. Il legame esistente tra le componenti di uno stesso vettore nei due sistemi di riferimento è il seguente:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}$$

Derivando ambo i membri si ottiene:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} \ddot{u} \\ \ddot{v} \end{bmatrix} - \dot{\vartheta} \begin{bmatrix} \sin \vartheta & \cos \vartheta \\ -\cos \vartheta & \sin \vartheta \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}$$

Considerando la prima equazione, si ricava:

$$\begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} \ddot{u} \\ \ddot{v} \end{bmatrix} - \dot{\vartheta} \begin{bmatrix} \sin \vartheta & \cos \vartheta \\ -\cos \vartheta & \sin \vartheta \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} F_u \\ F_v \end{bmatrix}$$

Indicando con R la matrice di rotazione tra i due sistemi di riferimento

$$R = \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix}$$

Moltiplicando per l'inversa di R a primo e secondo membro si ottiene:

$$R^{-1} \left(\begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} \ddot{u} \\ \ddot{v} \end{bmatrix} - \dot{\vartheta} \begin{bmatrix} \sin \vartheta & \cos \vartheta \\ -\cos \vartheta & \sin \vartheta \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \right) = \frac{1}{m} R^{-1} \begin{bmatrix} \cos \vartheta & -\sin \vartheta \\ \sin \vartheta & \cos \vartheta \end{bmatrix} \begin{bmatrix} F_u \\ F_v \end{bmatrix}$$

E quindi si giunge a:

$$\begin{bmatrix} \ddot{u} \\ \ddot{v} \end{bmatrix} - \dot{\vartheta} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_u \\ F_v \end{bmatrix}$$

Che unita alla 2 dà il sistema di equazioni espresso nel sistema di riferimento solidale.

Poniamo ora il sistema in forma di stato. Il vettore di stato è costituito dalla posizione e velocità angolare, dalla posizione e dalla velocità lungo u e lungo v .

Indicando con $x = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ il vettore di stato, dove le variabili di stato rappresentano:

x_1 la posizione lungo u

x_2 la posizione lungo v

x_3 la posizione angolare

x_4 la velocità lungo u

x_5 la velocità lungo v

x_6 la velocità angolare

considerando puramente algebrica la funzione di trasferimento degli attuatori si ottiene il seguente modello matematico:

$$\dot{x}_1 = x_4$$

$$\dot{x}_2 = x_5$$

$$\dot{x}_3 = x_6$$

$$\dot{x}_4 = x_5 x_6 + \frac{F_u}{m}$$

$$\dot{x}_5 = -x_4 x_6 + \frac{F_v}{m}$$

$$\dot{x}_6 = \frac{M}{I}$$

Come si vede il modello presenta dei termini di accoppiamento non lineare. Effettuando la linearizzazione attorno al punto di equilibrio definito dal vettore nullo i termini suddetti spariscono giungendo alla consueta forma ISU per modelli LTI.

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Dove:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & \frac{1}{I} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Risulta evidente, quindi, che il modello linearizzato è tanto più fedele al modello reale quanto più contenuta è la velocità angolare attorno l'asse perpendicolare al piano del moto dell'hovercraft. Banalmente per velocità angolari nulle l'atto di moto sarebbe di pura traslazione e le equazioni scritte nei due riferimenti sarebbero coincidenti.

Ci si pone ora il problema di come vengano generate le risultanti delle forze, F_u e F_v , e la risultante del momento M tramite il sistema di attuazione scelto per il controllo dell'hovercraft e rappresentato schematicamente nella figura successiva.

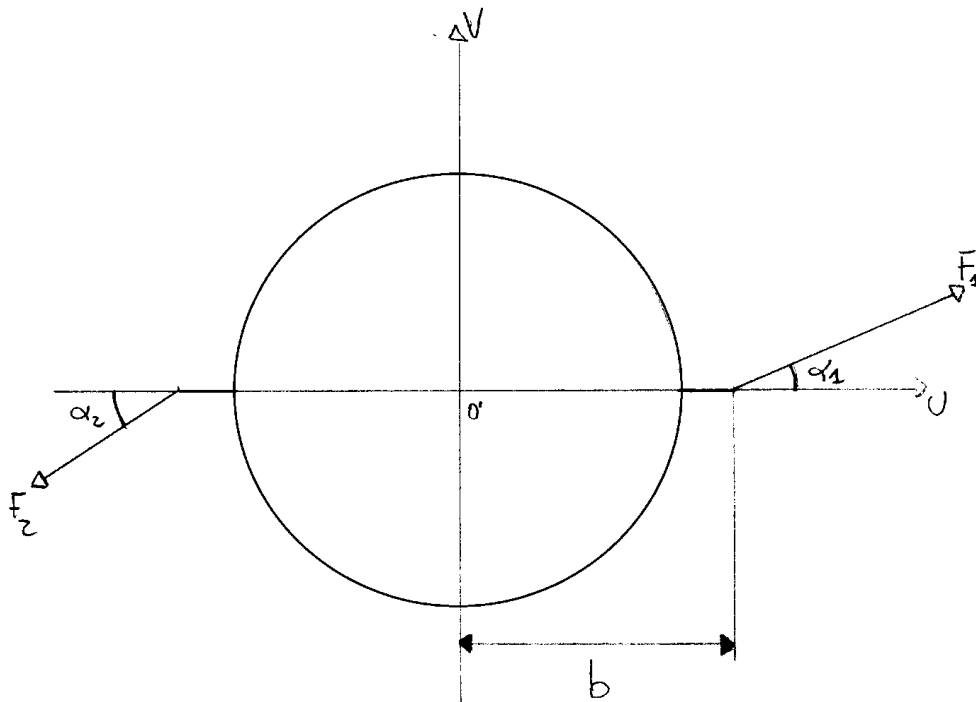


Figura 19 - Schematizzazione del sistema di forze agente

F_1 e F_2 sono le forze di spinta esercitate dalle eliche, mentre α_1 e α_2 indicano gli angoli che formano gli assi delle eliche con l'asse di riferimento u .

I parametri indicati, quindi, rappresentano le grandezze sulle quali agire per controllare il sistema. Partendo dalla conoscenza dei valori di F_u e F_v ed M , forniti dal sistema di controllo occorre scegliere opportunamente i valori delle variabili indicate in figura affinché sul sistema agiscano le azioni di controllo necessarie. È il caso di dire che per quanto riguarda le eliche il legame che intercorre tra velocità di rotazione e forza esplicita non è di tipo lineare. Nonostante dalla letteratura, note le caratteristiche costruttive dell'elica, sia possibile ricavare un'espressione in forma chiusa che ben approssimi tale legame, si è preferito identificare sperimentalmente la caratteristica di trasferimento delle eliche scelte. Il processo di identificazione, al quanto rudimentale, si è svolto utilizzando una bilancia elettronica da cucina come dinamometro e imponendo manualmente diversi valori della velocità di rotazione delle eliche e registrando manualmente i valori di forza corrispondenti. I dati così raccolti sono stati elaborati tramite il curve fitting tool di matlab ricavando i coefficienti di un

polinomio di secondo grado anche se per convenienza, sia il termine costante che il termine lineare sono stati imposti nulli, in modo da ottenere l'espressione del legame forza-velocità di rotazione più semplice possibile. La possibilità di imporre valori di velocità di rotazione precisi deriva dal fatto che, come già anticipato, i propulsori sono controllati a ciclo chiuso secondo uno schema PI.

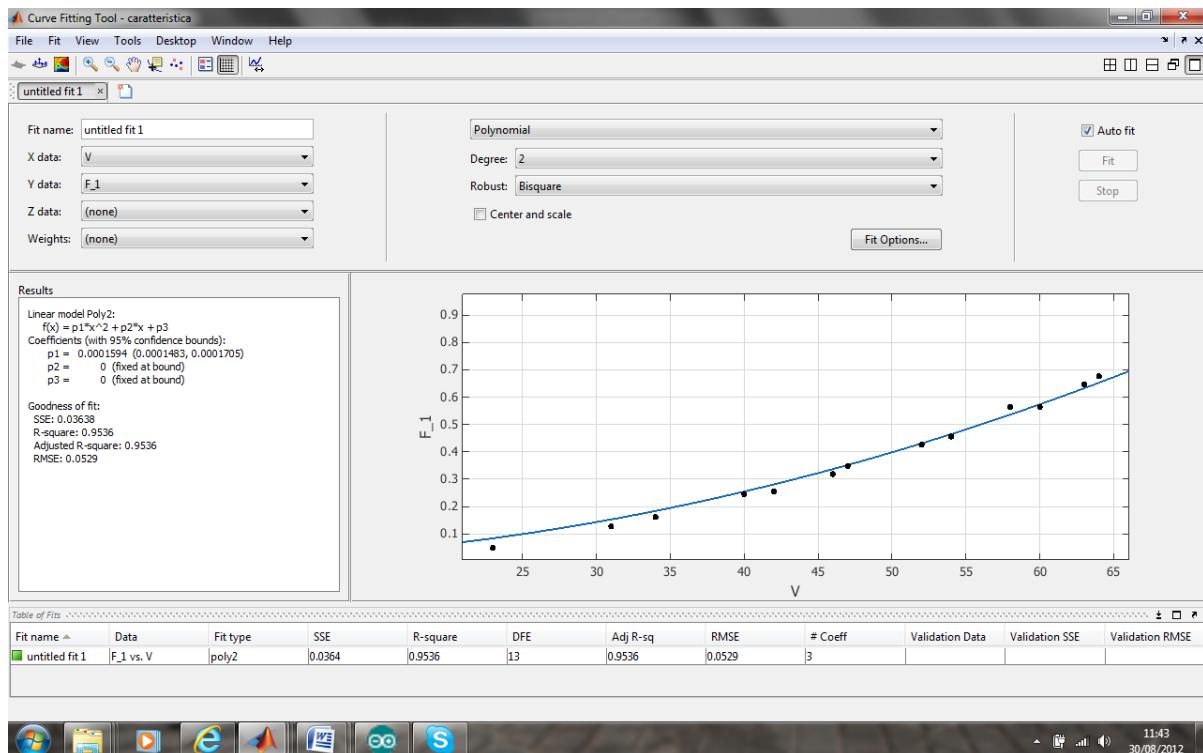


Figura 20 - Schermata del Curve Fitting Tool per l'identificazione sperimentale della caratteristica dell'elica

Ritornando al problema della generazione delle azioni di controllo si intuisce che ci si trova da un sistema di tre equazioni in quattro incognite:

$$F_u = F_1 \cos \alpha_1 - F_2 \cos \alpha_2$$

$$F_v = F_1 \sin \alpha_1 - F_2 \sin \alpha_2$$

$$M = b(F_1 \sin \alpha_1 + F_2 \sin \alpha_2)$$

Volendo evitare rotazioni negative delle eliche e volendo mantenere gli angoli α_1 e α_2 in un range limitato, una delle possibili soluzioni è quella di mantenere anche in condizioni di equilibrio attive le eliche generando le F_1 e F_2 uguali ed opposte. Così facendo si perviene alla seguente soluzione:

$$F_1 = \sqrt{\left(C + \frac{F_u}{2}\right)^2 + \left(\frac{F_v}{2} + \frac{M}{2}\right)^2}$$

$$F_2 = \sqrt{\left(C - \frac{F_u}{2}\right)^2 + \left(-\frac{F_v}{2} + \frac{M}{2}\right)^2}$$

$$\alpha_1 = \tan^{-1} \frac{\left(\frac{F_v}{2} + \frac{M}{2}\right)}{\left(C + \frac{F_u}{2}\right)}$$

$$\alpha_2 = \tan^{-1} \frac{\left(-\frac{F_v}{2} + \frac{M}{2}\right)}{\left(C - \frac{F_u}{2}\right)}$$

Nella trattazione si è solo accennato alla dinamica degli attuatori e di fatto nel modello matematico la loro caratteristica è stata considerata puramente algebrica. In realtà sia le forze di spinta che le loro direzioni sono governate da motori in corrente continua la cui dinamica può essere considerata in prima approssimazione lineare e del primo ordine anche se, come si vede dalle equazioni delle risultanti delle forze e dei momenti, la loro combinazione dà luogo ad una funzione non lineare delle variabili di stato. Tuttavia se si considera che le dinamiche degli attuatori risultano più veloci della dinamica del sistema controllato allora si commette un errore trascurabile nel ritenere algebrico il legame ingresso uscita. Nel modello simulink in ogni caso si include il modello matematico degli attuatori per pervenire a simulazioni più fedeli alla realtà.

5. Progetto del controllore

Come si vede dal modello matematico non lineare, le equazioni presentano dei termini di accoppiamento non lineare che spariscono con la linearizzazione del sistema attorno all'origine dello spazio di stato. Per questo motivo il problema che ci si presenta può in realtà essere visto come tre problemi di controllo SISO, e di conseguenza affrontato con le tecniche di progetto mono variabile note dai corsi di controlli automatici oppure risolto con tecniche multi variabile quali LQR, allocazione dei poli, controllori di disaccoppiamento. Si vuole sottolineare comunque che il disaccoppiamento delle equazioni è però solo frutto della linearizzazione per cui il problema non è intrinsecamente disaccoppiato.

Per quanto detto precedentemente a proposito della dinamica, e quindi dal ritardo introdotto all'interno del ciclo di reazione dagli attuatori non modellati nel sistema di equazioni ricavato, per le sue caratteristiche di robustezza, il controllore che si è scelto di utilizzare è l'LQR. Il control system toolbox di Matlab mette a disposizione numerose funzioni per la sintesi del controllore, tra cui quelle per il progetto di LQR e LQR PI, quest'ultimo in grado di annullare asintoticamente gli effetti dovuti a disturbi costanti agenti sul sistema.

Prima di affrontare il progetto del controllore è il caso di analizzare le proprietà strutturali del sistema e cioè verificarne la controllabilità e l'osservabilità. Quest'operazione richiede in realtà poche righe di codice Matlab e dato l'ingombro della matrice di controllabilità è superfluo riportarne l'espressione numerica.

$$\text{rank}([B \ AB \ A^2B \ A^3B \ A^4B \ A^5B]) = 6$$

Nel progetto del controllore lo stato del sistema è considerato interamente misurabile. In effetti i sensori presenti a bordo del modellino forniscono tutte le misure necessarie a meno

dei loro coefficienti i amplificazione. Supponendo quindi la matrice C proporzionale all'identità, o comunque diagonale con elementi non nulli l'osservabilità è garantita. In realtà nell'implementazione fisica del controllore alcune delle misure sono ottenute per integrazione o per derivazione numerica poiché l'implementazione di un osservatore avrebbe aggiunto un ulteriore carico computazionale i cui effetti avrebbero potuto degradare le prestazioni del sistema a ciclo chiuso.

Solamente per rendere più lineare la trattazione si riporta l'impostazione matematica di un problema di controllo ottimo LQR:

$$\left\{ \begin{array}{l} \min_{u(\cdot)} \int_0^{+\infty} (x^t Q x + u^t R u) dt \\ \text{s. t.} \\ \dot{x} = Ax + Bu \\ x(0) = x_0 \end{array} \right.$$

Dove A e B rappresentano le matrici del sistema e Q e R rappresentano le matrici di peso rispettivamente dello stato e del controllo. Si intuisce quindi che il punto cruciale del progetto di un controllore LQR consiste nella scelta delle matrici di peso. La matrice dei guadagni K ottenuta dalla soluzione dell'equazione di Riccati è ricavata direttamente attraverso il comando Matlab `lqr()`.

Nel seguito verranno riportati i risultati del progetto e delle simulazioni effettuate sul sistema linearizzato e con attuatori privi di dinamica. Lo schema di controllo realizzato è quello riportato in figura:

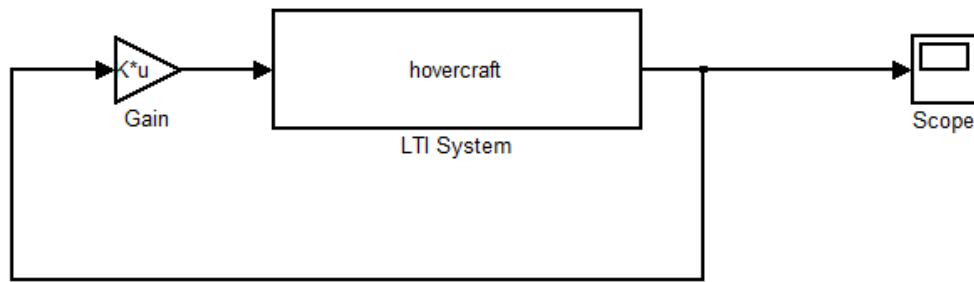


Figura 21 - Schema di controllo Simulink

La scelta delle matrici di peso è la seguente:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$$

$$R = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

Il significato fisico della scelta effettuata è da ricercare nel fatto che senza ulteriori considerazioni lo scostamento dello stato del sistema dall'origine dello spazio di stato ha peso identico per ogni sua componente. Stesso discorso si potrebbe ripetere per la matrice R che esprime il costo dell'azione di controllo. Inoltre non esistono specifiche di progetto se non quelle della stabilizzazione e della reiezione dei disturbi. Il motivo di tutto ciò è da ricercare nel fatto che il modello sviluppato contiene numerose fonti di incertezza ed è costruito con tecniche rudimentali. Il sistema di controllo possiede una potenza di calcolo piuttosto limitata.

Molti dei sensori utilizzati nascono per scopi differenti. La lista sarebbe ancora lunga e il progetto certamente non esaurisce le possibilità di miglioramento. Di seguito sono riportati i risultati del progetto del controllore e gli andamenti di velocità, posizione e azioni di controllo a partire da uno stato iniziale non nullo del sistema controllato secondo lo schema precedente.

$$K = \begin{bmatrix} 3.1623 & 0 & 0 & 4.4144 & 0 & 0 \\ 0 & 3.1623 & 0 & 0 & 4.4144 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.1921 \end{bmatrix}$$

Dalla struttura della matrice dei guadagni K si evince che il controllore agisce sul sistema come se fosse costituito da tre sottosistemi completamente disaccoppiati. Un indubbio vantaggio derivante è la semplicità di implementazione del controllore all'interno del microcontrollore.

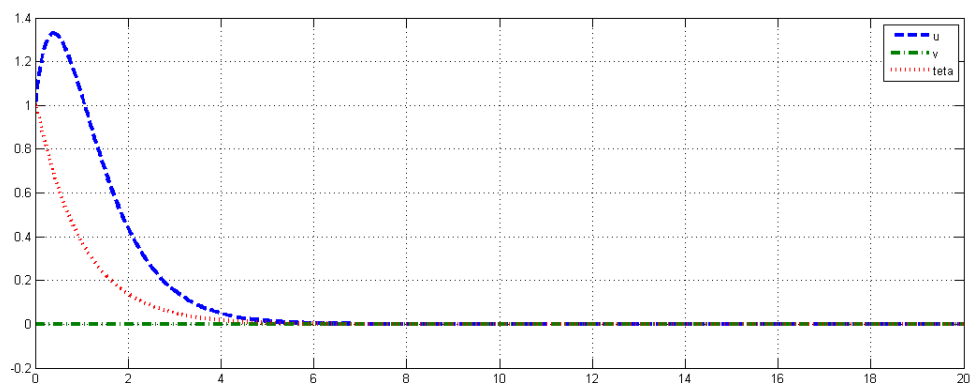


Figura 22 - Andamento posizione lineare ed angolare

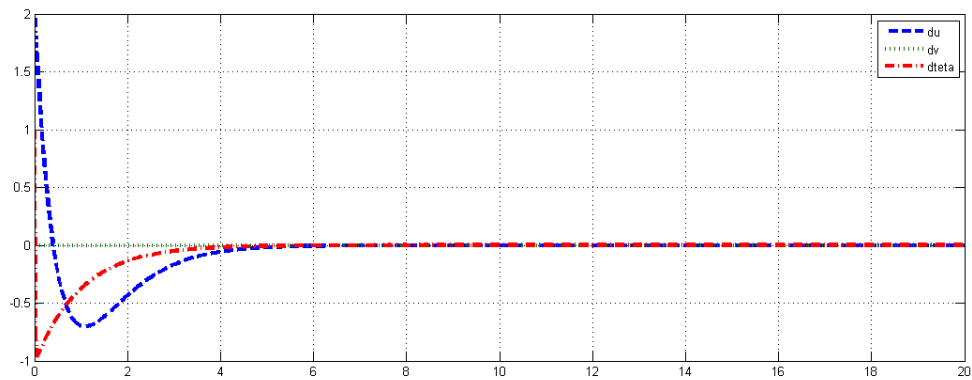


Figura 23 - Andamento velocità lineare ed angolare

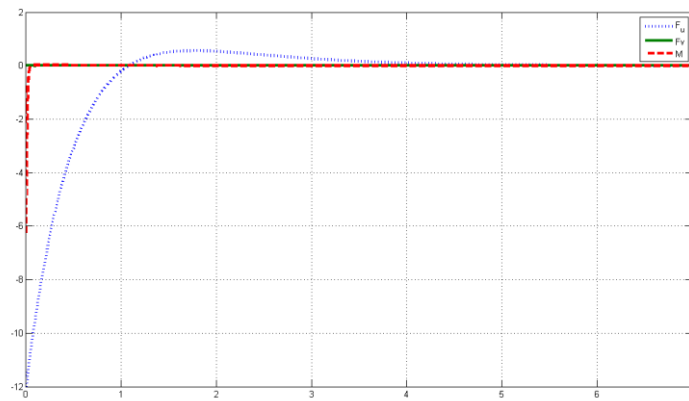


Figura 24 - Andamento delle azioni di controllo

Il progetto del controllore potrebbe essere considerato ultimato se non si avessero presenti gli aspetti implementativi, come ad esempio la necessità di discretizzare il sistema di controllo o la presenza inevitabile di saturazioni ed altri tipi di non linearità. In altre parole data, come già anticipato, l'assenza di specifiche ben definite sulle prestazioni non è facile giudicare l'adequatezza dei risultati.

Per validare il progetto inseriamo innanzitutto il controllore all'interno di un ciclo di controllo che contenga il modello non lineare oltre al sistema di attuazione comprensivo di un suo modello matematico così come mostrato di seguito.

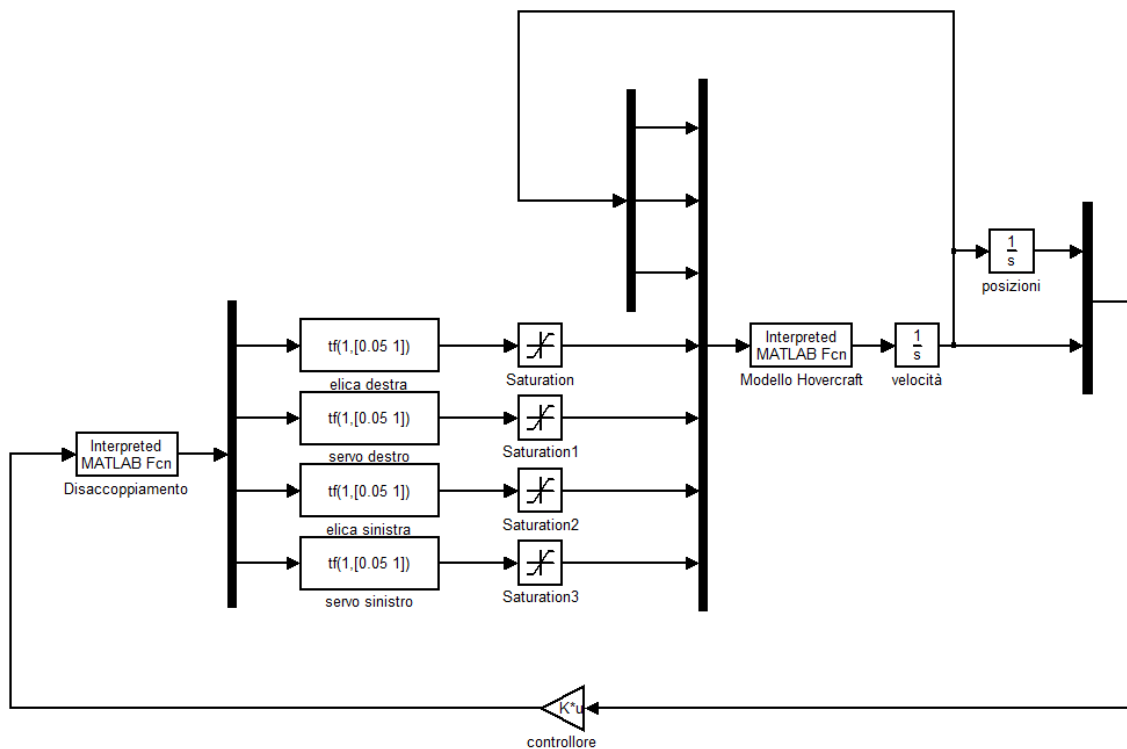


Figura 25 - Schema di controllo completo di attuatori e modello non lineare

Di seguito sono riportati gli andamenti delle variabili di stato del sistema sottoposto ad un disturbo sulle condizioni iniziali. A differenza del caso lineare la variabile che partiva dallo stato di equilibrio subisce degli scostamenti a causa dei termini di accoppiamento di cui si è precedentemente discusso. Per confronto sono riportate anche le simulazioni sul sistema linearizzato.

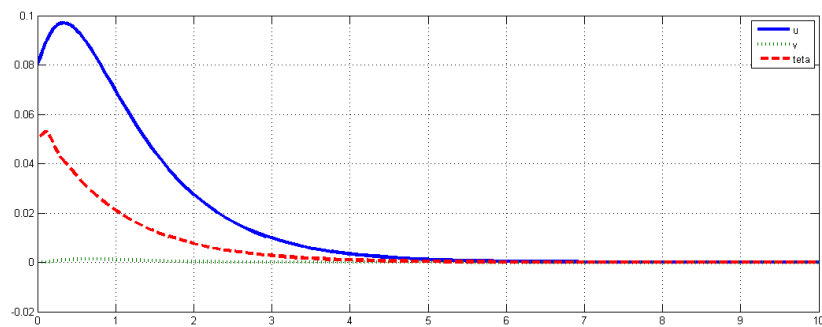


Figura 26 - Andamento delle posizioni (non lineare)

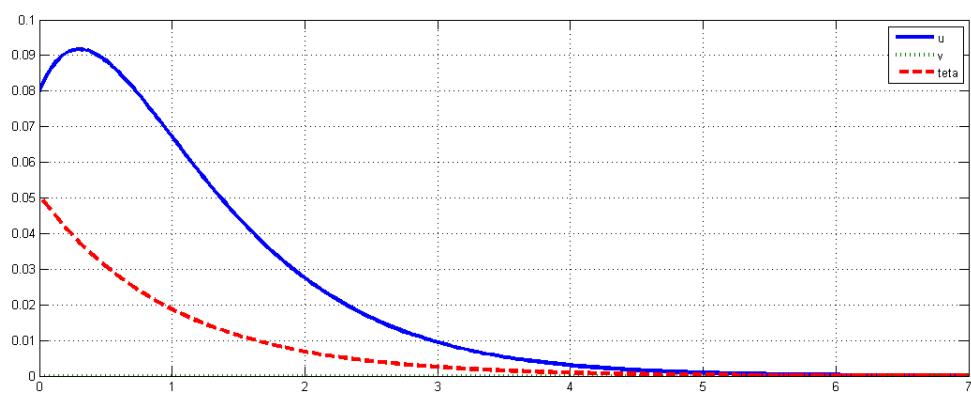


Figura 27 - Andamento delle posizioni (lineare)

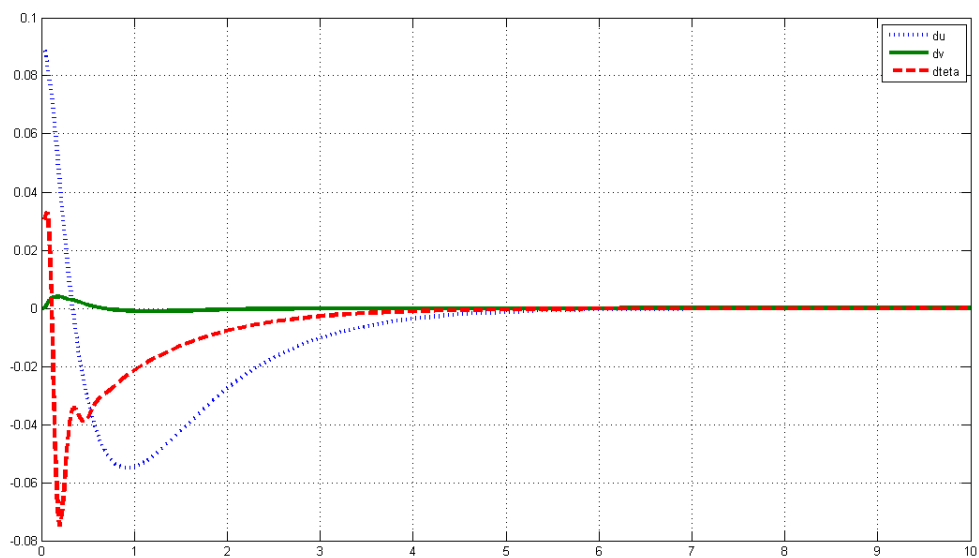


Figura 28 - Andamento delle velocità (non lineare)

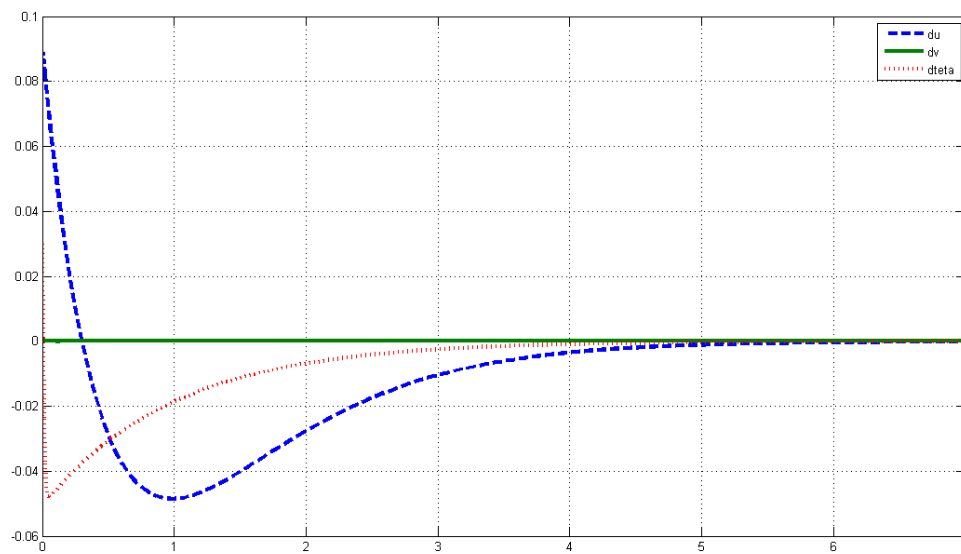


Figura 29 - Andamento delle velocità (lineare)

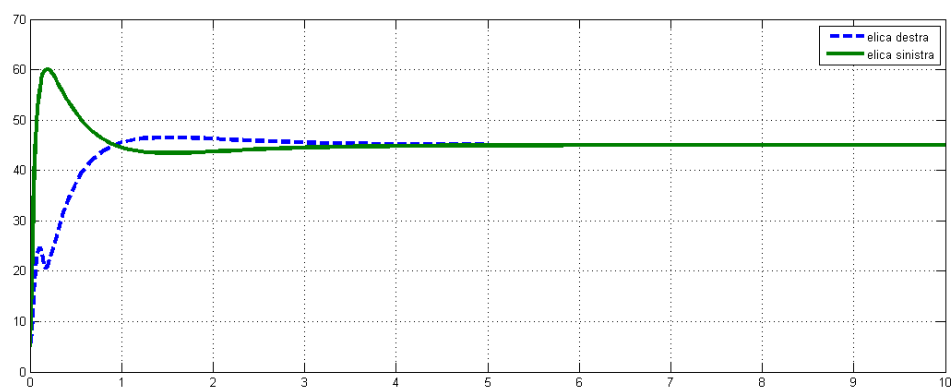


Figura 30 - Azione di controllo effettiva (giri elica)

È da tenere presente che le grandezze riportate sull'asse verticale del precedente grafico sono solo proporzionali al numero di giri al minuto dell'elica. In realtà rappresentano il numero di impulsi contati in un periodo di campionamento.

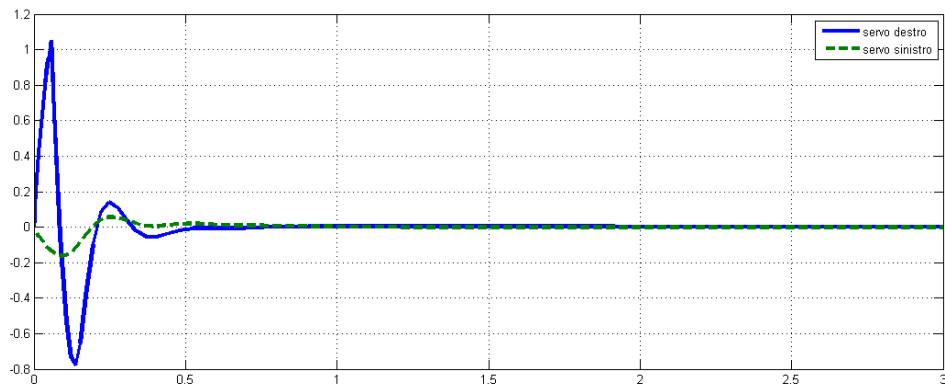


Figura 31 - Azione di controllo effettiva (angoli servomotori)

Infine si vuole realizzare un controllo PI che, come detto in precedenza è in grado di effettuare una più efficace reiezione dei disturbi costanti o lentamente variabili. Lo schema di controllo è mostrato in figura.

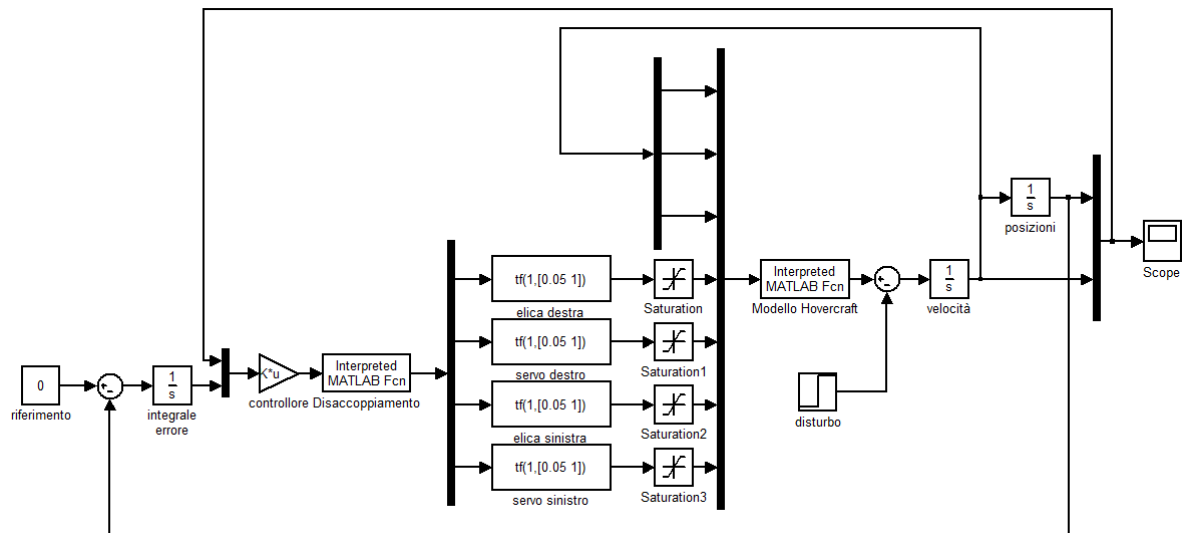


Figura 32 - Schema di controllo PI

Ancora una volta le matrici di peso sono selezionate empiricamente e la conferma della validità della scelta è data più che altro dalla capacità del controllore di stabilizzare il sistema in presenza di dinamiche non modellate e di non linearità.

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}$$

$$R = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

Di seguito sono riportati, come fatto in precedenza gli andamenti temporali delle variabili di stato e di controllo del sistema controllato e sottoposto ad un disturbo a gradino.

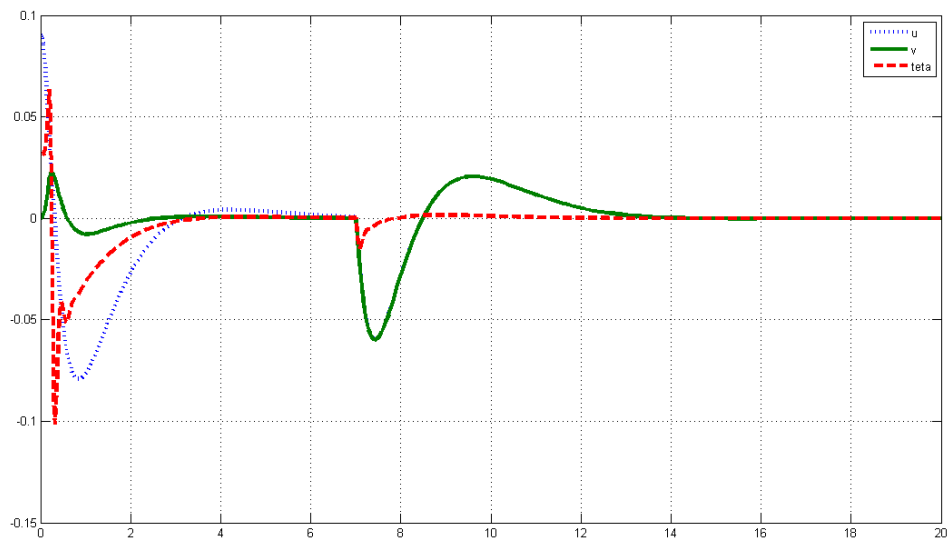


Figura 33 - Andamento posizioni e reiezione del disturbo (PI)

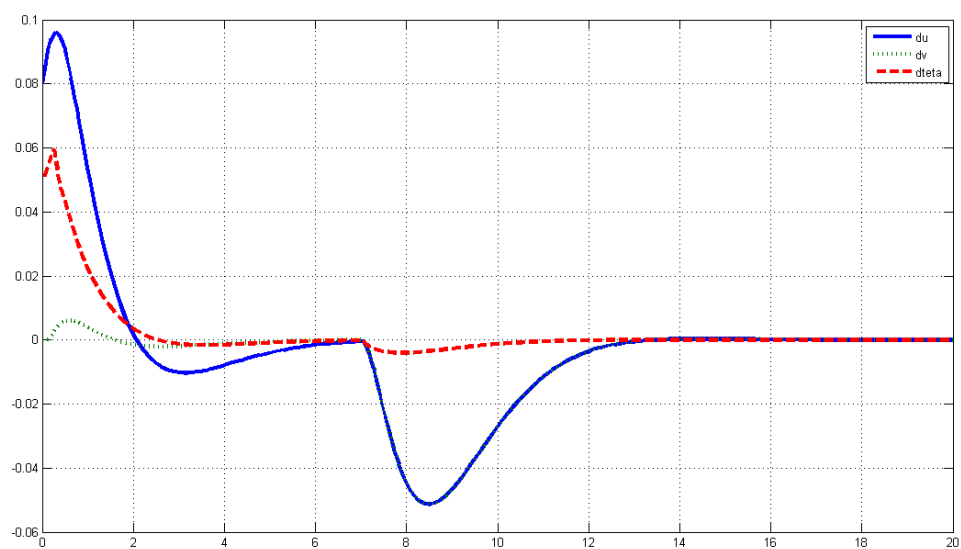


Figura 34 - Andamento delle velocità (PI)

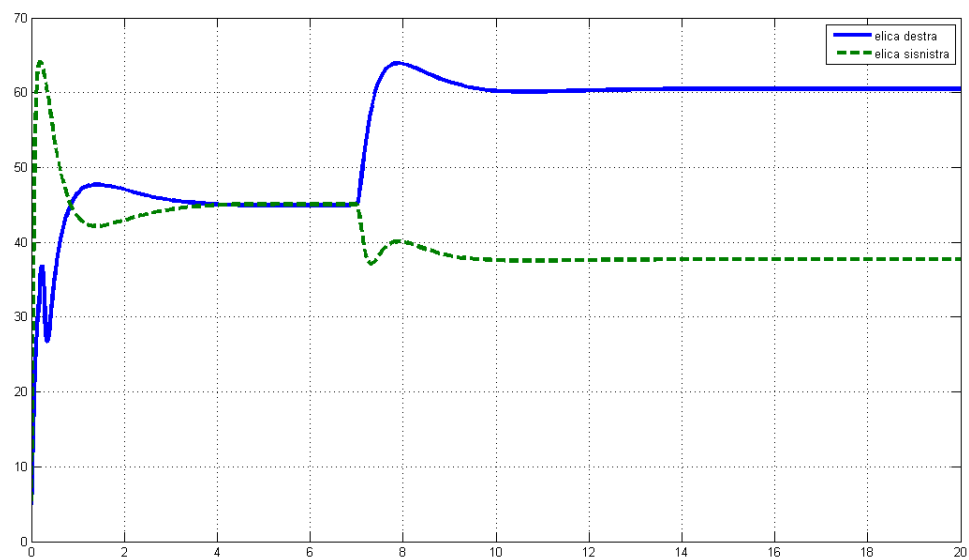


Figura 35 - Azione di controllo (giri elica PI)

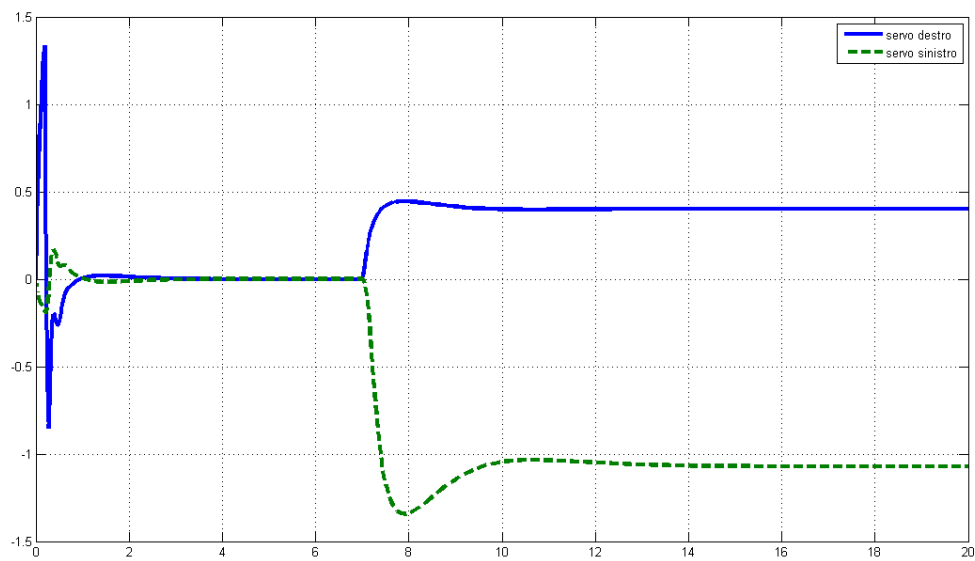


Figura 36 - Azione di controllo (angolo servomotori PI)

A questo punto il progetto del controllore può considerarsi esaurito. I problemi derivanti dall'implementazione fisica del controllore quale discretizzazione del modello e del controllore sono omessi. In appendice è riportato il codice Matlab per il progetto dei controllori e la codifica del controllore nell'ambiente Arduino.

Appendice A

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <math.h>
#include <ps2.h>

PS2 mouse(31,33);

//peso dell'apparato 1425g
//momento di inerzia circa 0.03

void aziona_motore_s(unsigned int duty, boolean dir);
void aziona_motore_d(unsigned int duty, boolean dir);
int regola_motore_d(int r);
int regola_motore_s(int r);
unsigned int leggi_contatore(void);
void scrivi_contatore(unsigned int i);
void corpo(void);

int r_s, r_d, alfa_s, alfa_d;
float omega;
float teta;

int x,y;
void mouse_init()
{
    mouse.write(0xff); // reset
    mouse.read(); // ack byte
    mouse.read(); // blank
    mouse.read(); // blank
    mouse.write(0xf0); // remote mode
    mouse.read(); // ack
    delayMicroseconds(100);
}
void setup()
{
    Serial.begin(115200);
    analogReference(INTERNAL2V56);
    mouse_init();

    //uscite pwm servomotori
    pinMode(2,OUTPUT);
```

```

pinMode(3,OUTPUT);

//uscita pwm elica sostentamento
pinMode(9,OUTPUT);
//uscite pwm ventola direzionale sinistra
pinMode(11,OUTPUT);
pinMode(12,OUTPUT);
//uscite controllo direzione ventola destra
pinMode(14,OUTPUT);
pinMode(15,OUTPUT);
//uscite pwm ventola direzionale destra
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
//uscite controllo direzione ventola sinistra
pinMode(16,OUTPUT);
pinMode(17,OUTPUT);
//clock contatore 5
pinMode(47,INPUT);
pinMode(20,INPUT);
pinMode(21,INPUT);
//tutti i mosfet lato basso degli H-Bridge sono interdetti
digitalWrite(14,HIGH);
digitalWrite(15,HIGH);
digitalWrite(16,HIGH);
digitalWrite(17,HIGH);
digitalWrite(9,HIGH);

//configurazione modulo pwm servomotori
//frequenza portante 50Hz
TCCR3A = 0b11111110;
TCCR3B = 0b00011010;
ICR3 = 40000;
TIMSK3 = 0b00000001;
OCR3A=3000;
OCR3B=3000;
OCR3C=3000;

//configurazione modulo pwm elica direzionale sinistra
//frequenza portante 244Hz
TCCR1A = 0b10101010;
TCCR1B = 0b00011001;
ICR1 = 65535;
//configurazione modulo pwm elica direzionale destra
//frequenza portante 244Hz
TCCR4A = 0b10101010;
TCCR4B = 0b00011001;
ICR4 = 65535;

```

```

    //configurazione contatore timer 5 associato al motore
    destro
    TCCR5A = 0b00000000;
    TCCR5B = 0b00000111;
    scrivi_contatore(0);

    //configurazione contatore timer 0 associato al motore
    sinistro
    TCCR0A = 0b00000000;
    TCCR0B = 0b00000110;
    TCNT0 = 0;
}
void loop()
{
}
void corpo()
{
    char mstat;
    char mx;
    char my;
    float u_z, u_x, u_y;
    float G;
    unsigned int tempo;

    static float m, F_x, F_y;
    static float x, y;
    float v_x, v_y;
    static int x_1;
    static float I_x=0, I_y=0;
    static float I_teta;
    int acc_x, acc_y;

    mouse.write(0xeb); // give me data!
    mouse.read();      // ignore ack
    mstat = mouse.read();
    mx = -mouse.read();
    my = -mouse.read();
    v_x = 25*float(mx)-2.2045*omega;
    v_y = 25*float(my);
    x+=0.04*v_x;
    y+=my;
    x_1+=mx;

    I_x=-0.04*x+I_x;
    I_y=-0.04*y+I_y;
    I_teta=-0.04*teta+I_teta;
    u_x=-double(0.00043*v_x+0.00061*x-0.0004*I_x);
    u_y=-double(0.00043*v_y+0.00061*y-0.0004*I_y);
    u_z=float(0.0002526*omega+0.0003074*teta-0.0028*I_teta);

```

```

disaccoppia(u_x,u_y,u_z);
tempo=TCNT3;
Serial.print("alfa_s ");
Serial.print(alfa_s);
Serial.print("\talfa_d ");
Serial.print(alfa_d);
Serial.print("\tX ");
Serial.print(x);
Serial.print("\tY ");
Serial.print(y);
Serial.print("\tteta ");
Serial.print(teta);
Serial.print("\tr_s ");
Serial.print(r_s);
Serial.print("\tr_d ");
Serial.print(r_d);
Serial.println();
}
ISR(TIMER3_OVF_vect)
{
    static int cont=0;
    cont++;

    omega=float(analogRead(A4)-491);
    teta+=float(omega*0.02);
    if(cont==2){
        cont=0;
        corpo();
    }
}
int regola_motore_d(int r)
{
    int r_k;
    boolean dir;
    static int cont_0;
    static int u_k=0,r_old=0;
    cont_0=TCNT0;
    TCNT0=0;
    if(r>85) r=85;
    if(r<5) r=5;

    r_k=(r-cont_0);
    u_k=616*r_k-520*r_old+u_k;
    if(u_k>35000) u_k=35000;
    if(u_k<-35000) u_k=-35000;

    if(u_k>0)dir=false;
    else dir=true;
    aziona_motore_d(u_k,dir);
}

```



```

    r_old=r_k;
    return cont_0;
}
int regola_motore_s(int r)
{
    int r_k;
    boolean dir;
    static int cont_5;
    static int u_k=0, r_old=0;
    cont_5=TCNT5;
    TCNT5=0;
    if(r>85) r=85;
    if(r<5) r=5;
    r_k=(r-cont_5);
    u_k=616*r_k-520*r_old+u_k;
    if(u_k>35000) u_k=35000;
    if(u_k<-35000) u_k=-35000;

    if(u_k>0) dir=false;
    else dir=true;
    aziona_motore_s(u_k, dir);
    r_old=r_k;
    return cont_5;
}
void aziona_motore_d(unsigned int duty, boolean dir)
{
    static int stab=0;
    //motore sinistro
    if(dir==true)
    {
        stab++;
        if(duty>10000)
            duty=10000;
        digitalWrite(16, LOW);
        digitalWrite(17, HIGH);
        OCR1A=0;
        if(stab>20)
            OCR1B=0;
        else OCR1B=duty;
    }
    else
    {
        stab--;
        if(stab<0) stab=0;
        if(duty>35000)
            duty=35000;
        digitalWrite(16, HIGH);
        digitalWrite(17, LOW);
        OCR1A=duty;
    }
}

```

```

        OCR1B=0;
    }
}
void aziona_motore_s(unsigned int duty, boolean dir)
{
    static int stab;
    //motore destro

    if(dir==true)
    {
        stab++;
        if(duty>10000)
            duty=10000;
        digitalWrite(14,LOW);
        digitalWrite(15,HIGH);
        OCR4A=0;
        if(stab>20)
            OCR4B=0;
        else OCR4B=duty;
    }
    else
    {
        stab--;
        if(stab<0) stab=0;
        if(duty>35000)
            duty=35000;
        digitalWrite(14,HIGH);
        digitalWrite(15,LOW);
        OCR4A=duty;
        OCR4B=0;
    }
}
unsigned int leggi_contatore( void )
{
    //la lettura del contatore avviene in due cicli macchina,
    disattivando gli interrupt atomizzo l'operazione
    unsigned int i;
    noInterrupts();
    i = TCNT5;
    interrupts();
    return i;
}
void scrivi_contatore(unsigned int i)
{
    unsigned char sreg;
    noInterrupts();
    TCNT5 = i;
    SREG = sreg;
    interrupts();
}

```

```

}
void disaccoppia(float r_x, float r_y, float m)
{
    //static float c=0.323;
    static float c=0.4822;

    r_d=int(sqrt(6273.525721*sqrt(square(c+r_x)+square(r_y+m))));
    r_s=int(sqrt(6273.525721*sqrt(square(c-r_x)+square(m-
r_y))));
    alfa_d=int(1120*atan2(r_y+m,c+r_x));
    alfa_s=int(1120*atan2(m-r_y,c-r_x));
    if(alfa_d>1000) alfa_d=1000;
    if(alfa_d<-1000) alfa_d=-1000;
    if(alfa_s>1000) alfa_s=1000;
    if(alfa_s<-1000) alfa_s=-1000;
    if(r_d>85) r_d=85;
    if(r_d<0) r_d=0;
    if(r_s>85) r_s=85;
    if(r_s<0) r_s=0;
    regola_motore_s(r_s);
    regola_motore_d(r_d);
    OCR3C=3000+alfa_s;
    OCR3B=3000+alfa_d;
}

```

Appendice B

```

m=1.5;
I=0.03;
b=0.3;

```

```

A=[0    0    0    1    0    0
    0    0    0    0    1    0
    0    0    0    0    0    1
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0];

```

```

B=[0    0    0

```

```

0      0      0
0      0      0
1/m    0      0
0      1/m    0
0      0      1/I];

C=eye(6);

D=zeros(6,3);

hovercraft=ss(A,B,C,D);

Q=diag([100 100 100 100 100 100]);
R=diag([10 10 10]);

[K,S,e]=lqr(hovercraft,Q,R,[]);

C_1=[eye(3),zeros(3,3)];
D_1=zeros(3,3);
hovercraft_int=ss(A,B,C_1,D_1);

Q_i=diag([100 100 100 100 100 100 100 100 100]);
R_i=diag([10 10 10]);

[K_i,S_i,e_i]=lqi(hovercraft_int,Q_i,R_i,[]);

```