

---

**ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA**

SCUOLA DI INGEGNERIA E ARCHITETTURA  
SEDE DI FORLÌ

CORSO DI LAUREA IN  
**INGEGNERIA AEROSPAZIALE**

CLASSE L-9

ELABORATO FINALE DI LAUREA IN  
*MECCANICA DEL VOLO*

**MODELLAZIONE E SIMULAZIONE DI SCENARIO  
MULTIAGENTE CON VELIVOLI MULTIROTORE NON  
COOPERATIVI**

**CANDIDATO  
AUGUSTO MAZZEI**

**RELATORE  
FABRIZIO GIULIETTI**

*ANNO ACCADEMICO 2017/2018*

---



*«Un uccello posato su un ramo  
non ha mai paura che il ramo si rompa,  
perchè la sua fiducia non è nel ramo,  
ma nelle sue ali»*

## Sommario

INTRODUZIONE .....	1
DEFINIZIONE DEL PROBLEMA: L'INSEGUIMENTO .....	4
IL DRONE .....	7
IL MODELLO DELLA NAVIGAZIONE PROPORZIONALE .....	11
Un caso semplificato .....	11
Pure Proportional Navigation (PPN) .....	12
True Proportional Navigation (TPN) .....	13
Teoria del controllo ottimale.....	14
ALGORITMO RISOLUTIVO .....	16
Modello dinamico del drone .....	16
Modello del motore .....	18
Motor Mixer.....	19
Esacottero.....	20
Quadricottero .....	23
Pilota Automatico .....	26
Controllo dell'angolo di imbardata.....	27
Controllo dell'angolo di beccheggio .....	29
Controllo della spinta totale.....	30
Regolazione dei segnali in ingresso .....	32
SCHEMA GERARCHICO DELL'ALGORITMO .....	36
L'INTERFACCIA PER L'UTENTE .....	39
IL SOFTWARE IN AMBIENTE SIMULINK.....	43
RICOSTRUZIONE GRAFICA DELLA SIMULAZIONE .....	47
Il formato STL.....	47
La rappresentazione 3D in Matlab .....	47
CONCLUSIONI.....	51
BIBLIOGRAFIA & SITOGRAFIA .....	53



## INTRODUZIONE

Un sistema multiagente o (sistema ad agenti multipli) è un insieme di agenti situati in un certo ambiente ed interagenti tra loro mediante una opportuna organizzazione. Un agente è cioè un'entità caratterizzata dal fatto di essere, almeno parzialmente, autonoma, sia essa un programma informatico, un robot, un essere umano.

I sistemi multiagente possono essere visti come crocevia di diverse discipline:

- l'intelligenza artificiale, per gli aspetti decisionali relativi all'agente;
- i sistemi distribuiti per ciò che concerne le interazioni tra agenti;
- l'ingegneria del software, per quanto riguarda l'evoluzione verso componenti sempre più dotati di autonomia.

Si possono rilevare alcuni importanti problemi nei confronti della creazione di sistemi multiagente.

In primo luogo, il problema dell'azione: come può un insieme di agenti operare simultaneamente in un ambiente frammentato, e come interagisce tale ambiente in risposta agli agenti? Le questioni qui sottintese sono, tra le altre, quella della rappresentazione dell'ambiente per mezzo degli agenti, della collaborazione tra gli agenti, e della pianificazione dell'attività di una molteplicità d'agenti.

Inoltre, è da considerare il problema dell'agente e della sua relazione col mondo esterno, la quale è rappresentata dal modello cognitivo di cui dispone l'agente. Ogni singolo individuo in una società costituita da molteplici agenti deve essere in grado di mettere in opera le azioni più rispondenti ai suoi obiettivi.

I sistemi ad agenti multipli necessitano altresì dello studio della natura delle interazioni, come fonte di opportunità, da una parte, e di vincoli dall'altra. Il problema dell'interazione si interessa alle modalità di interazione (quale linguaggio? quale supporto?), all'analisi ed alla concezione delle forme d'interazione tra agenti. Le nozioni di collaborazione e di cooperazione (considerata come collaborazione + coordinamento delle azioni + risoluzione dei conflitti) sono in questo campo questioni fondamentali. Se si applicano le mere conoscenze dei sistemi a singolo agente a quelli multi agente si avranno degli agenti egoisti (*self-interested*) che non coopereranno in alcun modo, dunque l'obiettivo del sistema sarà raggiunto in minima parte.

Infine, rimane da discutere la questione della realizzazione effettiva e dell'implementazione dei sistemi multiagente, tramite la strutturazione dei linguaggi di programmazione in più tipologie, che vanno da un linguaggio di tipo L5, o linguaggio di formalizzazione e specifica, ad un linguaggio di tipo L1, che è quello dell'implementazione effettiva. Tra le due classi, si trovano i linguaggi per la comunicazione tra agenti, la descrizione delle leggi dell'ambiente e la rappresentazione delle conoscenze.

Definite queste problematiche sarà possibile delineare gli elementi dell'architettura d'un sistema ad agenti multipli come segue. Gli agenti devono essere dotati di svariati sistemi di decisione e pianificazione. La ricerca operativa, o teoria delle decisioni, è una disciplina completamente dedicata allo studio di questo soggetto. Nella categoria delle interazioni con l'ambiente, un altro problema ricorrente dei sistemi ad agenti è quello del pathfinding (ovvero la ricerca di un percorso determinati un punto di partenza e di arrivo, insieme al suo algoritmo più conosciuto, l'algoritmo A\* o anche A-star). Gli agenti necessitano inoltre di un modello cognitivo: anche qui esistono diversi modelli, tra cui, uno dei più classici, il modello BDI (*Beliefs-Desires-Intentions*, cioè *Convinzioni-Desideri-Intenzioni*). Esso considera da una parte l'insieme delle

credenze (*Beliefs*) dell'agente sull'ambiente in cui si trova ad operare, che sono il risultato delle sue conoscenze e delle sue percezioni, e dall'altra parte un insieme di obiettivi (*Desires*). Intersecando questi due insiemi, si ottiene un nuovo insieme di intenzioni (*Intentions*), che in seguito possono tradursi direttamente in azioni.

È possibile costruire una tabella che raggruppi i differenti tipi d'agente (compilata da Ferber, si veda la bibliografia):

Categorie di agenti

	<b>Agenti cognitivi</b>	<b>Agenti reattivi</b>
<b>Comportamento teleonomico</b>	Agenti intenzionali	Agenti impulsivi
<b>Comportamento riflesso</b>	Agenti "modulari"	Agenti tropici

*tab. 1 – caratterizzazione comportamentale generale di agenti capaci di intervenire in un ambiente sotto diversi fenomeni*

Gli agenti cognitivi sono il più delle volte intenzionali, cioè hanno scopi prefissati che tentano di realizzare. Si possono pure trovare talvolta degli agenti detti modulari i quali, anche se possiedono una rappresentazione del loro universo, non hanno tuttavia degli scopi precisi. Tali agenti potrebbero servire ad esempio a rispondere alle interrogazioni avanzate da altri agenti su detto universo. Gli agenti reattivi possono a loro volta essere suddivisi in agenti impulsivi e tropici. Un agente impulsivo avrà una missione ben determinata (per esempio, assicurarsi che un serbatoio resti sempre sufficientemente pieno) e si comporterà di conseguenza qualora percepisca che l'ambiente non risponde più allo scopo che gli è stato affidato (il livello del serbatoio è troppo basso). L'agente tropico, dal canto suo, reagisce solo allo stato locale dell'ambiente (c'è del fuoco, io fuggo). La fonte della motivazione è nel primo caso interna (agenti impulsivi che hanno una "missione"), nel secondo caso è legata esclusivamente all'ambiente.

## DEFINIZIONE DEL PROBLEMA: L'INSEGUIMENTO

Nel campo della ricerca, per i sistemi multiagente si distinguono generalmente tre tipi d'utilizzo: la simulazione di fenomeni complessi, la risoluzione di problemi, e la progettazione di programmi. Si utilizzano i sistemi multiagente per simulare le interazioni esistenti tra agenti autonomi. Si cerca di determinare l'evoluzione del sistema al fine di prevederne l'organizzazione risultante. Il fattore importante è il comportamento dell'insieme, non certo il comportamento individuale. Le applicazioni esistenti hanno riguardato la fisica delle particelle (agente = particella elementare), la chimica (agente = molecola), la biologia cellulare (agente = cellula), l'etologia (agente = animale), la sociologia e l'etnologia (agente = essere umano).

L'autonomia consente in questi casi di simulare il comportamento esatto di ogni entità. L'intelligenza artificiale distribuita è nata per risolvere i problemi di complessità dei grossi programmi monoblocco dell'intelligenza artificiale: l'esecuzione è pertanto non concentrata, ma distribuita, ma il controllo rimane centralizzato. Al contrario, nei sistemi ad agenti multipli, ciascun agente possiede un controllo totale sul suo comportamento. Per risolvere un problema complesso, talvolta è in effetti più semplice concepire programmi relativamente piccoli (gli agenti) interagenti con un unico grosso programma monoblocco. L'autonomia permette al sistema di adattarsi dinamicamente ai cambiamenti imprevisti che intervengono nell'ambiente.

Allo stesso tempo, l'ingegneria del software si è evoluta in direzione di componenti sempre più autonomi. I sistemi ad agenti multipli possono essere visti come un raccordo tra ingegneria del software ed intelligenza artificiale distribuita, con l'apporto rilevante dei sistemi distribuiti. In rapporto ad un oggetto, un agente può prendere iniziative, può rifiutarsi di obbedire ad una richiesta, può spostarsi, etc..

L'autonomia consente al progettista di concentrarsi sul lato *umanamente comprensibile* del software.

Oggetto di ricerca di questa tesi sarà la ricreazione di un sistema multiagente composto da due individui autonomi, rappresentati da due droni, situati in luoghi differenti dello spazio. Uno di questi si comporterà indifferentemente (Agente Modulare) nei confronti dell'altro, il quale dal canto suo, avrà l'obiettivo di raggiungere il primo, presentandosi quindi come agente intenzionale e impulsivo. Il problema dell'inseguimento sarà inoltre supportato da alcune condizioni:

- il drone inseguitore conosce esattamente la posizione e la velocità del suo obiettivo, quindi ha una conoscenza esatta e puntuale dell'ambiente che lo circonda.
- non ci sono elementi esterni presenti nell'ambiente che interferiscono con lo svolgimento della missione.

L'inseguimento è un tema che ricorre spesso anche nel mondo videoludico, dove alcuni elementi devono seguire il percorso di altri elementi capobranco, e la capacità di poter automatizzare il fenomeno, facendo a meno dell'intervento diretto dell'uomo, può portare grandi vantaggi nel settore tecnologico, in quanto vedrebbe la nascita di individui capaci di arrivare scientemente in un punto specifico, sia esso fisso o in movimento, ma non perché è stato insegnato loro un percorso preimpostato.

La ricerca di una strategia di guida sarà quindi il punto di svolta per la ricreazione dello scenario proposto, e sarà ampiamente affrontata nei successivi capitoli. Da un punto di vista matematico ci si limiterà a definire il problema in questo modo:

$$\iint_0^{\tau} \vec{a}_H(t) dt^2 = \iint_0^{\tau} \vec{a}_T(t) dt^2$$

Dove  $\vec{a}_H$  ed  $\vec{a}_T$  sono le accelerazioni di inseguitore e bersaglio (Hunter, Target)

$\tau$  è l'intervallo di tempo necessario al completamento della missione (parametro

incognito), e le condizioni iniziali dei due agenti  $\begin{bmatrix} x \\ y \\ z \end{bmatrix}_H$ ,  $\begin{bmatrix} u \\ v \\ w \end{bmatrix}_H$ ,  $\begin{bmatrix} x \\ y \\ z \end{bmatrix}_T$ ,  $\begin{bmatrix} u \\ v \\ w \end{bmatrix}_T$  sono distinte.

## IL DRONE

Un aeromobile a pilotaggio remoto o APR, comunemente noto come drone, è un apparecchio volante caratterizzato dall'assenza del pilota a bordo. Il suo volo è controllato dal computer a bordo del mezzo stesso oppure tramite il controllo remoto di un navigatore o pilota. Si è registrato un crescente utilizzo dei droni in applicazioni civili, ad esempio in operazioni di prevenzione e intervento in emergenza incendi, per usi di sicurezza, sorveglianza di oleodotti, o con finalità di telerilevamento e ricerca, ampliamento della rete internet, irrigazione nell'agricoltura, rilievo dell'architettura e, più in generale, in tutti i casi in cui tali sistemi possano consentire l'esecuzione di missioni "noiose, sporche e pericolose" (dull, dirty and dangerous) spesso con costi economici ed etici minori rispetto ai mezzi aerei tradizionali.



*Fig. 1 – Fotografia del Drone S800 EVO progettato dalla compagnia DJI, rinomata nel settore tecnologico. Caratteristiche presenti nel Data Sheet consultabile in Bibliografia & Sitografia.*

L'ICAO riconosce molte categorie di aeromobili con pilota a bordo (*manned*) o senza (*unmanned*), dando a tutte lo status di "aeromobile". Anche un APR è quindi un aeromobile e i concetti normativi non subiscono alterazioni di principio circa il mezzo, il pilota e l'operatore (valgono le stesse responsabilità e certificazioni, come

certificato di immatricolazione, certificato di aeronavigabilità, licenza di pilota, licenza di operatore). In particolare esiste un quadro normativo generale applicabile, tuttavia un insieme di norme implementative adeguate è in fase di definizione. Gli APR di peso inferiore ai 150 kg sono di pertinenza delle singole autorità aeronautiche nazionali, l'ENAC in Italia, come stabilito nei regolamenti europei.

A differenza degli aerei tradizionali, gli APR possono essere utilizzati in situazioni caratterizzate da un elevato pericolo per la vita umana e nelle aree inaccessibili o impervie, volando a bassa quota. Per questo motivo possono trovare impiego durante le fasi di monitoraggio di aree colpite da calamità naturali o da avvenimenti particolari (terremoti, esondazioni, incidenti stradali ecc.). Ne è un esempio il recente (Marzo 2011) terremoto del Tōhoku in Giappone che ha colpito la centrale nucleare di Fukushima Dai-ichi: in quell'occasione sono stati utilizzati dei Global Hawk, col fine di monitorare i reattori dopo le esplosioni che si erano verificate.

Un ulteriore vantaggio che deriva dall'uso degli APR per scopi civili è il contenuto costo di acquisizione e di esercizio di tali sistemi, rispetto ai tradizionali sistemi di ripresa aerea utilizzati fino ad ora. A questo aspetto si affianca la facilità di utilizzo e la loro versatilità.

Il progetto SMAT si propone di studiare e dimostrare un sistema di monitoraggio avanzato del territorio per la prevenzione e il controllo di una vasta gamma di eventi naturali (alluvioni, incendi, frane, traffico, urbanistica, inquinamento e coltivazioni). SMAT-F1 è focalizzato sulla dimostrazione dell'utilizzo integrato delle tre piattaforme UAV all'interno di uno scenario operativo, lo Sky-Y di Alenia Aermacchi (confluita in Leonardo), il Falco di SELEX Galileo (azienda di Leonardo) e il piccolo C-Fly della Nimbus e rappresenta un record europeo di grande rilievo in quanto è la prima volta che più sistemi senza pilota operano congiuntamente e contemporaneamente nello stesso spazio aereo non militare.

Un settore in piena espansione porta con sé la sfida di integrare migliaia di nuovi aeromobili nello spazio aereo nazionale e internazionale. Le loro ridotte dimensioni mettono in risalto seri problemi con il rilevamento radar, definendo quindi delle sfide di design per una gestione ottimale del traffico aereo (ATM). La NASA sta lavorando sul problema, in collaborazione con aziende come Google e Amazon. Il sistema di gestione del traffico progettato dalla NASA è probabile che lavori con l'uso di reti cellulari per le comunicazioni, mentre saranno utilizzati anche dei droni con GPS integrato per il relativo tracciamento. Il sistema basato su cloud sarebbe in gran parte automatico e auto-ottimizzante, il che significa che gli esseri umani potrebbero intervenire solo in caso di emergenza. Un mercato, quello dei droni, in rapida crescita con normative sempre più stringenti che devono stare al passo con lo sviluppo della tecnologia. Le normative costantemente aggiornate potranno contribuire a garantire il grande potenziale dei droni adeguatamente indirizzato.

Sul fronte della sicurezza, a preoccupare sono soprattutto le collisioni in volo e la perdita di controllo. Una collisione in volo potrebbe verificarsi qualora il pilota non riesca a vedere e quindi ad evitare in tempo gli altri velivoli, in particolare quelli che solitamente volano sotto i 150 metri. Lo scorso anno, i rapporti relativi agli avvistamenti di droni da parte di piloti, cittadini e forze dell'ordine sono quintuplicati negli USA, mentre in altre parti del mondo, come in Cina, a Dubai e nel Regno Unito, vi sono stati alcuni rischi di collisione. La perdita di controllo, invece, può avvenire se si verifica un guasto nel sistema o se il drone vola oltre la portata del segnale. AGCS prevede un rischio maggiore di perdita di controllo dovuta a interferenze di frequenza e ad altri fattori.

Ad esempio, se un pilota perde il controllo di un drone durante l'ispezione ad un edificio, può provocare una responsabilità totale che può superare facilmente i 5 milioni di dollari nel caso in cui il drone precipiti su un camion o all'interno di un negozio. Anche un piccolo drone può causare da solo un danno da 10 milioni di dollari

se entra in contatto con il motore di un aereo. Un altro pericolo emergente è la potenziale minaccia terroristica che potrebbe provenire da droni che mirano ad infrastrutture sensibili come impianti nucleari o a eventi dal vivo. Altri possibili scenari includono gli hacker che prendono il controllo di un volo aereo, provocando uno scontro, oppure che interferiscono con il segnale radio per trasmettere dati registrati importanti dall'aereo ad un'altra stazione di controllo ("spoofing").

Servono allora standard e normative coerenti che guidino l'utilizzo sicuro degli APR nel mondo e una formazione adeguata dal momento che l'inesperienza dei piloti è una causa principale della perdita delle attività.

## IL MODELLO DELLA NAVIGAZIONE PROPORZIONALE

La navigazione proporzionale è una legge di guida utilizzata per l'inseguimento di un obiettivo in movimento. Essa si basa sul fatto che due veicoli sono in rotta di collisione se la loro congiungente non cambia direzione al diminuire della distanza. La PN stabilisce che il vettore velocità dell'inseguitore (che definiamo come Hunter) dovrebbe ruotare in concomitanza con la rotazione della linea congiungente tra inseguitore ed obiettivo (che definiamo Target), linea che prende il nome di "LOS" (linea di vista, line of sight). Si noti che il vettore velocità è raramente parallelo alla LOS, e nei casi in cui ciò accade si assisterà ad uno scontro frontale o ad un raggiungimento alle spalle. Il modello PN si rivela molto utile poiché computazionalmente poco costoso, intuitivo, e può essere formulato in più varianti (come vedremo a breve) a seconda delle esigenze e delle informazioni che si hanno sul target. Non stupisce quindi che la PN sia alla base di quasi tutte le manovre di homing.

### Un caso semplificato

Per provare a capire il funzionamento della navigazione proporzionale proviamo a definire le variabili in gioco di una situazione elementare di inseguimento. In uno spazio bidimensionale vengono posti inseguitore (M) e obiettivo (T), i quali sono intesi come corpi puntiformi, e caratterizzati da una velocità costante in modulo. La distanza che li separa "R" è appunto la linea di vista, che rispetto ad un sistema di riferimento inerziale permette di descrivere l'angolo  $\vartheta$ . La velocità dell'inseguitore ha invece un orientamento  $\alpha_M$  sempre rispetto a tale sistema. Per far sì che il target venga raggiunto, l'hunter dovrà sviluppare una accelerazione tale da modificare la direzione

del vettore velocità con lo scopo di mantenersi costantemente rivolto verso la congiungente. Pertanto ogniqualvolta la LOS ha una variazione, descrivibile con  $\dot{\vartheta}$ , dovrà necessariamente corrispondere un  $\dot{\alpha}$ .

$$\dot{\alpha}_M = N\dot{\vartheta}$$

dove N è la costante di navigazione adimensionale, solitamente compresa tra 3 e 5

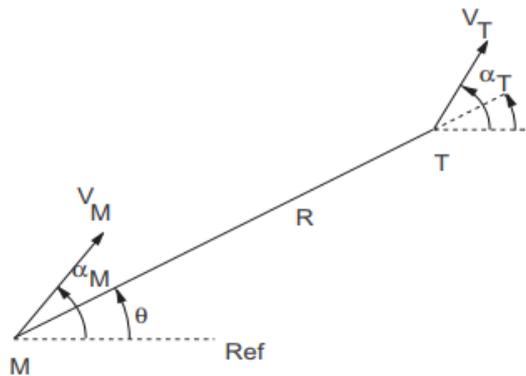


Fig. 2 – Schematizzazione preliminare della navigazione proporzionale. La modifica della linea di vista R implica una modifica nella direzione di  $V_M$ .

Col proseguire delle ricerche sono state evidenziate diverse modalità di navigazione proporzionale per gestire al meglio il comportamento dell'inseguitore.

### Pure Proportional Navigation (PPN)

La variante pura è quella più elementare, e si basa sul concetto che la rotazione del vettore velocità debba essere inversamente proporzionale alla velocità dell'inseguitore.

$$\dot{\alpha}_M = \frac{a_m}{V_m} \quad \text{che può essere riscritta come} \quad a_M = NV_M\dot{\vartheta}$$

Dove  $a_M$  risulta quindi essere l'accelerazione perpendicolare a  $V_M$  e che non si preoccupa della velocità del target. Il limite della navigazione pura in ambito missilistico nasce dal fatto che la portanza costituisca l'accelerazione di cui l'inseguitore ha bisogno, imponendo quindi un controllo molto stringente sull'angolo di attacco, tale da generare una portanza che, bilanciata con la spinta del missile, del Drag aerodinamico, e dell'accelerazione gravitazionale, dia come risultato la  $a_M$  richiesta.

### True Proportional Navigation (TPN)

Un'altra interessante variante della PN è quella che prende come riferimento la velocità di avvicinamento (ovvero la componente di  $V_M$  lungo la congiungente) anziché la  $V_M$ . La accelerazione sarà quindi costantemente perpendicolare alla LOS.

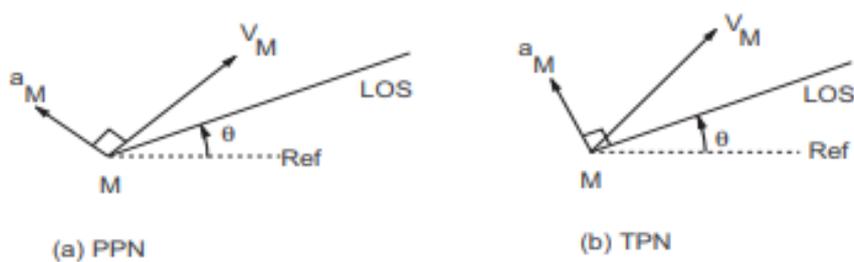


fig. 3 – la differenziazione tra PPN e TPN è facilmente distinguibile dalla perpendicolarità del vettore  $a_M$ .

$$\vec{a}_M = N\vec{V}_r \times \vec{\Omega} \qquad \vec{V}_r = \vec{V}_T - \vec{V}_M$$

con  $\Omega$  il vettore rotazione della congiungente (LOS) "R":  $\vec{\Omega} = \frac{\vec{R} \times \vec{V}_r}{R^2}$

Si può notare che, a differenza della PPN, una componente dell'accelerazione incrementerà il modulo della velocità dell'inseguitore. Tuttavia entrambi i modelli non pongono dei vincoli specifici sulla velocità dell'inseguitore, che può essere modificata in base alle esigenze della missione, liberandoci dalla condizione  $|V_M| = \text{costante}$ .

### Teoria del controllo ottimale

Ulteriori ottimizzazioni della PN sono state raggiunte tenendo conto del comportamento del target che raramente mantiene una velocità costante in modulo e direzione.

Definendo variabili come il "tempo all'impatto" e lo spostamento istantaneo previsto del target, che prende il nome di "Zero Effort Miss", la legge si arricchisce ulteriormente proprio per far fronte ad eventuali manovre evasive, con lo scopo di non mancare il bersaglio al primo tentativo.

$$t_{go} = \frac{R}{V_r}, \quad \overline{ZEM} = y + \dot{y} \cdot t_{go},$$

(dove  $y$  è la coordinata del target, ed  $\dot{y}$  la sua velocità)

$$\vec{a}_M = N \cdot \frac{\overline{ZEM}}{t_{go}^2} + N \frac{\vec{a}_{TPN}}{2}$$

Il risultato più evidente della variante TCO è un "appiattimento" della traiettoria percorsa dall'inseguitore. La accelerazione definita dalla PPN, sempre perpendicolare alla  $V_M$ , era l'artefice di un percorso sinuoso reso ancor più evidente quando l'hunter, mancando il target, si riposizionava dietro di esso compiendo una curva di 360° molto stretta.

La TPN è stata poi adottata nel modello risolutivo proposto in questa tesi e, vedendo i droni come protagonisti dello scenario, si è preferito ancorarsi al modello 2D anziché estenderlo alla navigazione proporzionale in 3 dimensioni a causa dei fenomeni di instabilità che si verificano quando l'accelerazione implica un assetto pericoloso per il drone.

## ALGORITMO RISOLUTIVO

### Modello dinamico del drone

Prima di descrivere la soluzione del problema proposto in questa tesi è opportuno introdurre alcune semplificazioni che sono state formulate per il drone agente nel sistema:

- Angolo diedro nullo
- Corpo rigido

Indipendentemente dal numero di eliche che il drone possiede, il modello cinematico generale resta invariato. I diversi valori di manetta danno luogo ad una forza, la quale contribuisce alla forza risultante ed al momento risultante del corpo. Il momento risultante modifica nel tempo le velocità angolari, relativamente facili da calcolare in un sdr body, e le stesse, convertite in un sistema di riferimento globale integrate nel tempo, modificano l'assetto stesso del drone. La forza risultante, combinata con l'assetto, modifica la velocità (body) del baricentro del drone, e anche questa, convertita in un sdr globale, essendo noto l'assetto precedentemente ricavato, ed integrata nel tempo, ci fornisce informazioni sulla posizione nel sistema di riferimento globale. Il sistema di riferimento locale è stato scelto per il calcolo di variabili dinamiche proprio per sfruttare alcuni valori costanti nel tempo, come il tensore baricentrico d'inerzia, ed il vettore del coefficiente di resistenza aerodinamica.

Equazioni differenziali delle velocità angolari nel sistema Body:

$$\begin{cases} \frac{J_Y - J_Z}{J_X} Q \cdot R + \frac{T_{Roll}}{J_X} = \dot{P} \\ \frac{J_Z - J_X}{J_Y} P \cdot R + \frac{T_{Pitch}}{J_Y} = \dot{Q} \\ \frac{J_X - J_Y}{J_Z} P \cdot Q + \frac{T_{Yaw}}{J_Z} = \dot{R} \end{cases}$$

Abbreviando poi per praticità le funzioni goniometriche, esprimeremo:

$$s = \text{"sin}(x)\text{"}, \quad c = \text{"cos}(x)\text{"}, \quad t = \text{"tan}(x)\text{"}$$

Equazioni differenziali dell'assetto, angoli di Eulero<sub>ZYX</sub>:

$$\begin{cases} P + (s\phi)(t\theta)Q + (c\phi)(t\theta)R = \dot{\phi} \\ (c\phi)Q - (s\phi)R = \dot{\theta} \\ \frac{(s\phi)}{(c\theta)} Q + \frac{(c\phi)}{(c\theta)} R = \dot{\psi} \end{cases}$$

Equazioni differenziali della velocità del baricentro nel sistema body:

$$\begin{cases} Rv - Qw - g(s\theta) - A_X C_{D_X} \rho \frac{|u|u}{2m} = \dot{u} \\ Pw - Ru + g(c\theta)(s\phi) - A_Y C_{D_Y} \rho \frac{|v|v}{2m} = \dot{v} \\ Qu - Pv + g(c\theta)(c\phi) - \frac{F_{TOT}}{m} - A_Z C_{D_Z} \rho \frac{|w|w}{2m} = \dot{w} \end{cases}$$

Equazioni differenziali della posizione del baricentro nel sistema di riferimento NED:

$$\begin{cases} (c\theta c\psi)u + v(s\phi s\theta c\psi) - c\phi s\psi + w(c\phi s\theta s\psi) + s\phi s\psi = \dot{x} \\ (c\theta s\psi)u + v(s\phi s\theta s\psi) + c\phi c\psi + w(c\phi s\theta s\psi) - s\phi c\psi = \dot{y} \\ -u(s\theta) + v(s\phi c\theta) + w(c\phi c\theta) = \dot{z} \end{cases}$$

Le variabili ottenute dal sistema saranno poi il feedback necessario al modello di navigazione proporzionale descritto successivamente.

## **Modello del motore**

Ogni drone è equipaggiato con motori identici, alimentati da una certa potenza elettrica. Si suppone che:

- il motore raggiunga istantaneamente i valori di manetta assegnati.
- la totalità della potenza fornita contribuisca alla produzione di coppia, senza perdite.

Analizzando al banco prova le prestazioni del motore che si intende assegnare ad ogni elica del proprio drone, sarà possibile delineare un *modello approssimato* da introdurre all'interno del software. Ogni motore può essere descritto osservando l'andamento della Tensione, Corrente, RPM e Spinta in funzione della manetta. Aumentando il grado di approssimazione della polinomiale, si ottiene un modello sempre più vicino alle prestazioni reali rilevate. Tuttavia, per non rinunciare ad una rappresentazione matriciale, si deve accettare una situazione di compromesso, scegliendo una approssimazione del primo ordine, forti del fatto che un buon controllo in retroazione (ad anello chiuso) garantirà un funzionamento efficace.

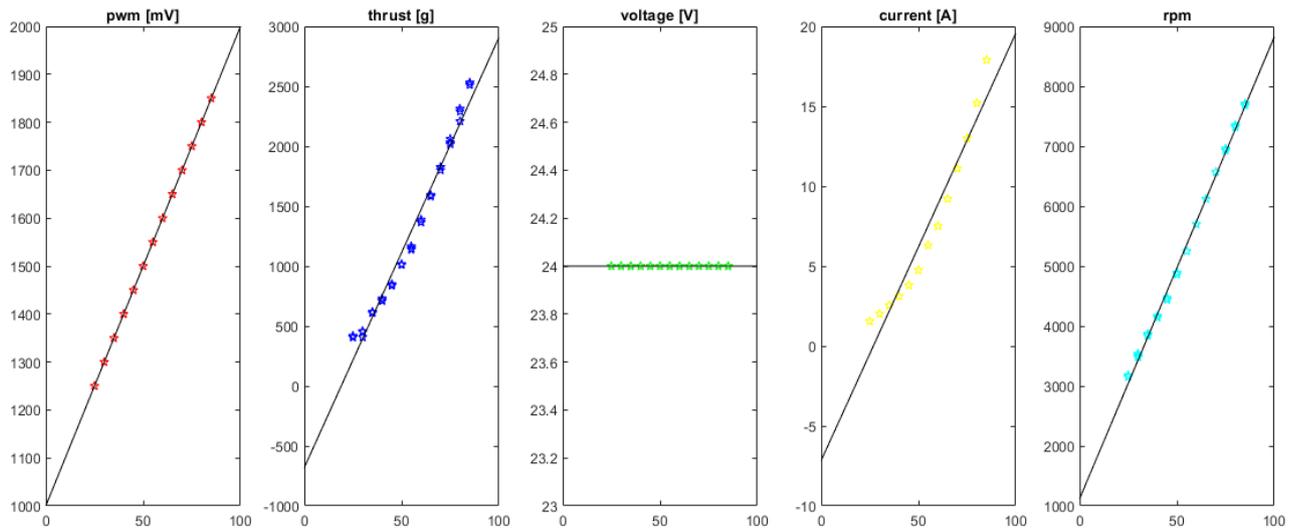


Fig. 4 – analisi delle prestazioni del motore standard equipaggiato sul drone S800 EVO, ottenuti al banco prova.

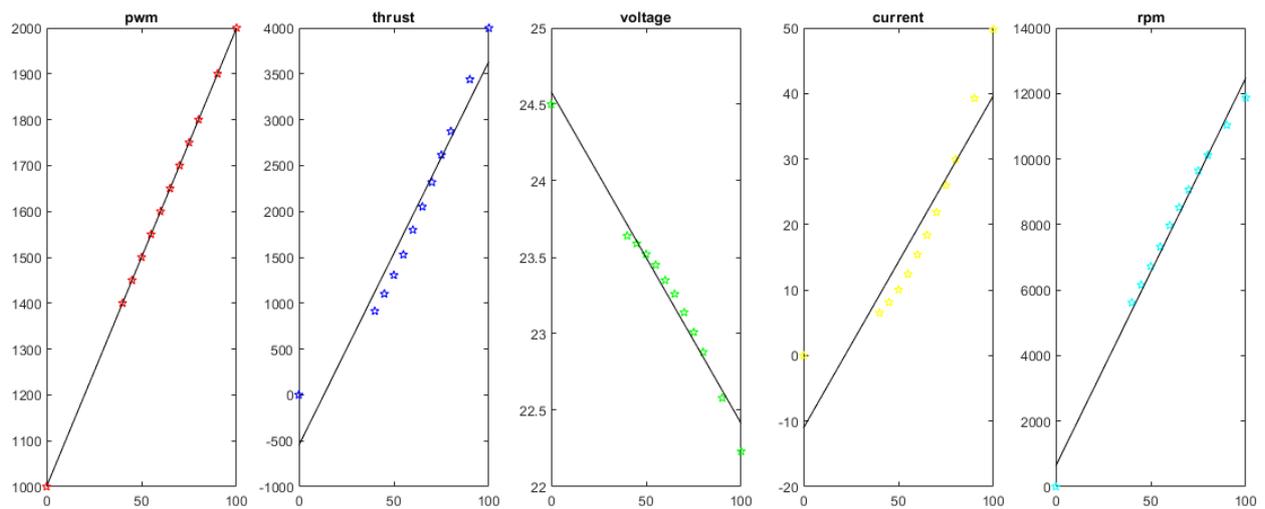


Fig. 5 – analisi delle prestazioni del motore F1000-Long Shaft KV635 per prestazioni più sportive, ottenute dal data-sheet (si veda bibliografia & sitografia).

## Motor Mixer

La sezione del motor mixer ha lo scopo di determinare, noto il numero di motori e la geometria a disposizione del drone, quale sia il set di manette necessario al raggiungimento di una specifica richiesta del pilota. In questa sezione ha poca rilevanza se la richiesta, caratterizzata da 4 valori, “Spinta Totale, Coppia in Roll, in

Pitch, in Yaw” sia stata formulata direttamente dal pilota con comando manuale o se sia intervenuto un software. Poiché solitamente i droni sono costituiti da 4 o da 6 motori, nel corso della trattazione si dovrà prendere come riferimento queste due casistiche.

## Esacottero

Per ogni motore si ottengono i valori di Spinta e Coppia in funzione della manetta:

$$f(throttle_i) = ax_i + b$$

$$\tau(throttle_i) = Ax_i + B$$

con la coppia esprimibile in funzione di Tensione, Corrente, e numero di giri al minuto, ipotizzando che l’energia con cui il motore è alimentato non vada incontro a forti dissipazioni di cui tener conto, data la semplicità del modello, e la rotazione coassiale dell’elica.

$$\tau_i = \frac{V_{(throttle)} \cdot I_{(throttle)}}{RPM_{(throttle)} \frac{2\pi}{60}}$$

Ipotizzando che l’angolo diedro delle braccia del drone sia trascurabile, tutte le eliche, puntando nella stessa direzione, contribuiscono alla spinta totale.

$$\sum_{i=1}^N f(x_i) = F_{TOT}$$

ed esplicitando, si ottiene:

$$a \sum_{i=1}^N (x_i) + 6b = F_{TOT}$$

quindi:

$$\sum_{i=1}^6 (x_i) = \frac{F_{TOT} - 6b}{a} = F'_{tot}$$

Mentre per la coppia in Roll e Pitch si deve tener conto anche della disposizione spaziale su cui il motore è installato.

$$\sum_{i=1}^N f(x_i) \wedge b_i = Torque_{ROLL}$$

Nel classico modello esagonale, il prodotto vettoriale lungo l'asse  $X_{body}$  ed  $Y_{body}$ , rispettivamente per il calcolo della coppia in Roll e Pitch, si può semplificare con il prodotto con un coefficiente scalare. Lo schema dei motori esibito offre delucidazioni a riguardo, in cui tutte le eliche sono distanziate di  $60^\circ$ .

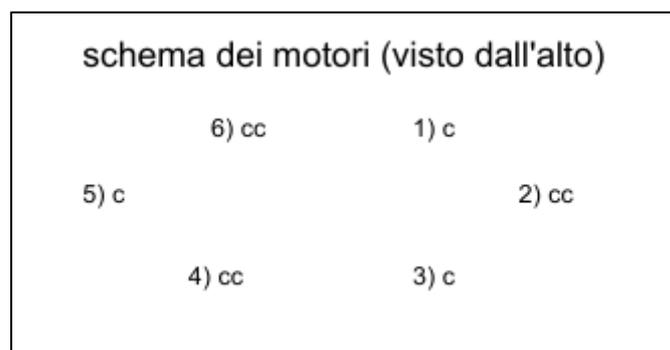


fig. 6 – rappresentazione geometrica della disposizione dei motori orari ed antiorari.

Equazione della coppia in Roll, ovvero lungo l'asse  $X_{body}$ :

$$-L \cos\left(\frac{\pi}{3}\right) \cdot f_1 - L \cdot f_2 - L \cdot \cos\left(\frac{\pi}{3}\right) f_3 + L \cdot \cos\left(\frac{\pi}{3}\right) f_4 + L \cdot f_5 + L \cdot \cos\left(\frac{\pi}{3}\right) f_6 = T_{roll}$$

Equazione della coppia in Pitch, ovvero lungo l'asse  $Y_{body}$ :

$$+L \sin\left(\frac{\pi}{3}\right) \cdot f_1 - 0 \cdot f_2 - L \cdot \sin\left(\frac{\pi}{3}\right) f_3 - L \cdot \sin\left(\frac{\pi}{3}\right) f_4 + 0 \cdot f_5 + L \cdot \sin\left(\frac{\pi}{3}\right) f_6 = T_{pitch}$$

In forma più compatta:

$$L \sum_{i=1}^N (C_i \cdot f_i) = T_{roll} \quad L \sum_{i=1}^N (C'_i \cdot f_i) = T_{pitch}$$

Con  $f_i = ax_i + b$

Osservando la simmetria dal punto di vista geometrico, che si riflette poi anche nei coefficienti  $C_i$  e  $C'_i$ , è evidente che il contributo complessivo del termine noto "b" è nullo, e si ottengono così la seconda e terza equazione che definiscono il modello del motor mixer.

$$\sum_{i=1}^N C_i x_i = \frac{T_{roll}}{aL} = T'_{roll} \quad \sum_{i=1}^N C'_i x_i = \frac{T_{pitch}}{aL} = T'_{pitch}$$

Ultima equazione per la ricerca del set dei sei valori  $X_i$  di manetta è l'equazione della coppia lungo l'asse di yaw. La coppia di ogni motore è determinata da un segno, il quale dipende dal senso di rotazione, che può essere orario (clockwise,  $K= +1$ ) oppure antiorario (counter-clockwise,  $K= -1$ ). Ancora una volta interviene la simmetria del modello, che stabilisce la presenza di 3 motori orari e 3 antiorari per minimizzare le vibrazioni prodotte dal drone, simmetria che annulla il contributo del termine noto della polinomiale che approssima la coppia del motore in funzione della manetta.

$$\sum_{i=1}^N \tau_i = \text{Torque}_{yaw}$$

In forma estesa si ottiene:

$$A(x_1 - x_2 + x_3 - x_4 + x_5 - x_6) + (0) = T_{yaw}$$

Ottenute così le quattro equazioni contenenti le 6 incognite del problema è possibile procedere alla risoluzione. Richiamando tali equazioni in forma matriciale otteniamo una rappresentazione compatta del modello, e risolviamo il sistema calcolando la matrice pseudoinversa.

$MAT$  = (matrice dei coefficienti del modello esacottero)

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -\cos\frac{\pi}{3} & -1 & -\cos\frac{\pi}{3} & \cos\frac{\pi}{3} & 1 & \cos\frac{\pi}{3} \\ \sin\frac{\pi}{3} & 0 & -\sin\frac{\pi}{3} & -\sin\frac{\pi}{3} & 0 & \sin\frac{\pi}{3} \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ \vdots \\ x_6 \end{bmatrix} = x = pinv(MAT) * \begin{bmatrix} F'_{tot} \\ T'_{roll} \\ T'_{pitch} \\ T'_{yaw} \end{bmatrix}$$

## Quadricottero

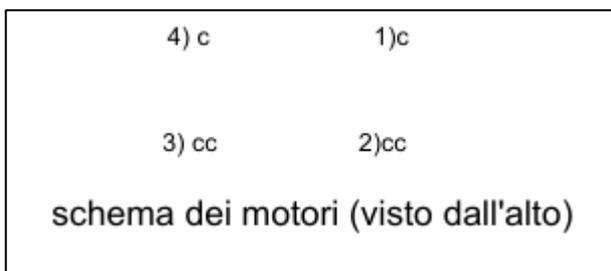


fig. 7 – disposizione geometrica dei motori in un quadricottero.

fig. 8 – fotografia del drone Phantom 4 PRO V2.0 della DJI.

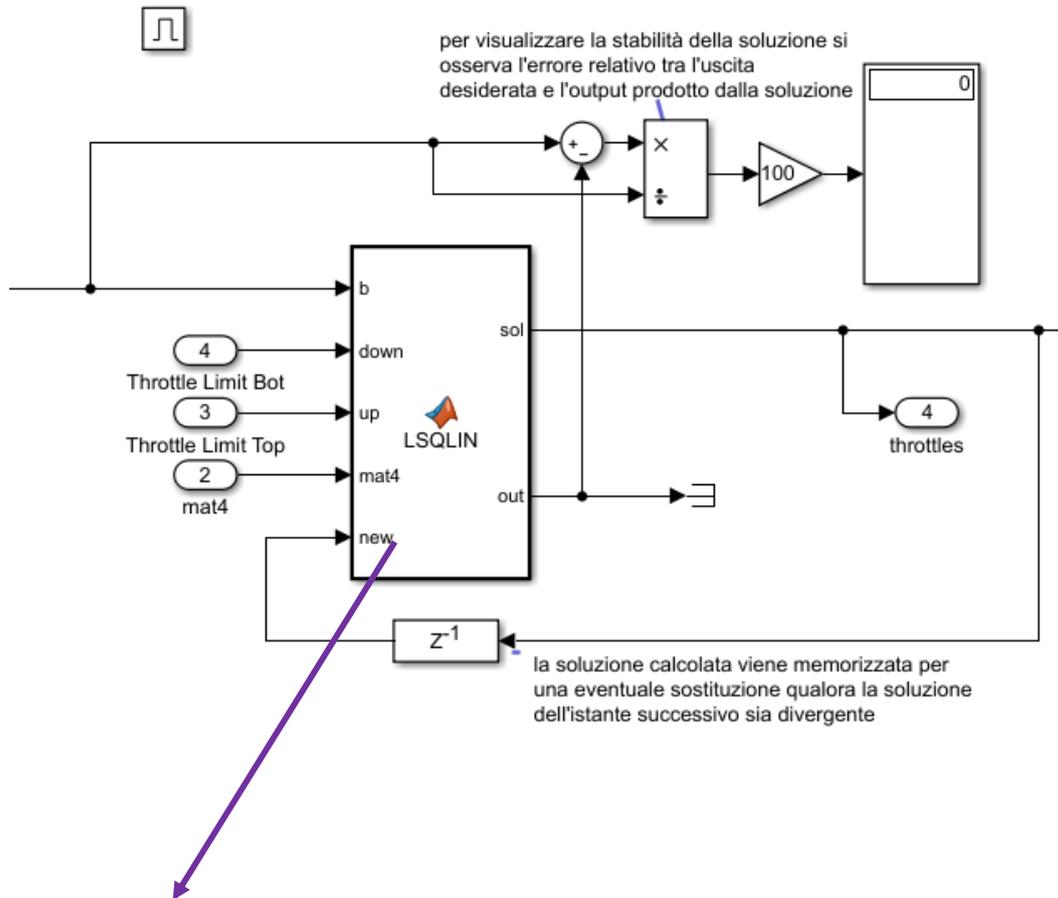
Sebbene fosse auspicabile una soluzione simile a quella descritta nel modello dell'esacottero per rispondere alla domanda: "quali valori di manetta assegnare ai 4 motori del drone per ottenere Forza e Coppie desiderate?", ciò non è stato possibile a causa di un problema sviluppatosi all'interno della matrice dei coefficienti ottenuta in modo analogo.

$MAT$  = (matrice dei coefficienti del modello quadricottero)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -\cos\left(\frac{\pi}{4}\right) & -\cos\left(\frac{\pi}{4}\right) & +\cos\left(\frac{\pi}{4}\right) & +\cos\left(\frac{\pi}{4}\right) \\ +\sin\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) & +\sin\left(\frac{\pi}{4}\right) \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

Tale matrice infatti non ha rango massimo, e trovandoci quindi di fronte ad un problema ad alto malcondizionamento è necessario percorrere altri iter risolutivi, poiché una minima variazione nel vettore dei termini noti provocherebbe fortissime oscillazioni nel set delle manette. Sono proprio le forti oscillazioni della soluzione a suggerire il metodo di cui avvalersi. Dovendo infatti rispettare i limiti fisici che il valore di manetta può assumere (dallo 0% al 100% nei casi più ideali, anche se nella realtà si rimane intorno al 30% con deboli oscillazioni) il modello deve essere definito attuando un processo di ottimizzazione lineare, in cui lo spazio dimensionale della soluzione è ben definito in  $[\mathcal{R}^4 \text{ con estremi } 0 - 100]$ , la cui variabile da ottimizzare è lo scarto quadratico medio tra il vettore dei termini noti (le richieste fornite al Motor Mixer) e l'uscita prodotta dal sistema lineare. In altre parole, si deve trovare la soluzione  $[x_1, x_2, x_3, x_4]$ , entro i limiti del dominio, che produce in uscita il vettore più simile a quello costituito da  $[F'_{tot}, T'_{roll}, T'_{pitch}, T'_{yaw}]$ .

Esistono diversi procedimenti per l'ottimizzazione lineare, alcuni di questi già implementati in MATLAB, come l'algoritmo "Trust Region Reflective" che si avvicina rapidamente alla soluzione del sistema lineare sfruttando il gradiente, similmente al metodo di Newton N-dimensionale.



```
sol=zeros(4,1);
coder.extrinsic('lsqlin')
coder.extrinsic('optimoptions')

a=zeros(4,1);
opt=optimoptions('lsqlin','Algorithm',...
    'trust-region-reflective','OptimalityTolerance',0.0001,'Display','off');
a =lsqlin(mat4,b,[],[],[],[], repmat(down,4,1), repmat(up,4,1), [], opt);
```

fig 9. – dettaglio del software progettato in Simulink relativo al problema di ottimizzazione lineare risolto con la funzione LSQLIN implementata nella libreria di Matlab.

Per attenuare ulteriormente le oscillazioni sul vettore delle manette è stato ritenuto opportuno “sporcare” la matrice dei coefficienti, modificando la disposizione geometrica dei motori per restituire alla matrice il rango massimo (4), pur non potendo diminuire significativamente il malcondizionamento del problema:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ -\cos(30^\circ) & -\cos(40^\circ) & +\cos(40^\circ) & +\cos(30^\circ) \\ +\sin(30^\circ) & -\sin(40^\circ) & -\sin(40^\circ) & +\sin(30^\circ) \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

## **Pilota Automatico**

Il complesso di formule inerenti alla navigazione proporzionale descritto nel paragrafo precedente procurava, indipendentemente dal tipo di veicolo, la accelerazione necessaria per correggere la propria traiettoria per il raggiungimento dell’obiettivo. Come già anticipato in precedenza, il modello PN adottato è rimasto ancorato ad un inseguimento bidimensionale, senza fornire vincoli lungo la terza dimensione, la quota, con la prospettiva di ricreare il comportamento di un’automobile capace di “sterzare” nel piano XY e di traslare lungo la dimensione Z. Nell’espone quindi l’elenco di canali sui quali effettuare il controllo della traiettoria, si ipotizzerà:

- Una modifica dell’angolo di beccheggio per modulare la velocità nel piano  $XY_{NED}$
- Una modifica dell’angolo di imbardata per indurre una deviazione della velocità nel piano  $XY_{NED}$
- Una modifica della spinta complessiva dei motori per determinare la velocità uscente dal piano  $XY_{NED}$

Questo comportamento favorisce infatti, per qualunque tipo di drone, un assetto stabile nel tempo, preservandolo da inclinazioni di Roll o Pitch troppo pericolose che lo avrebbero portato a ribaltarsi e a perdere il controllo.

Ogni modello deve essere capace di comunicare con il Motor Mixer, al quale fornire in ingresso un vettore contenente la Spinta complessiva dei motori e le Coppie lungo gli assi body. Vi è da precisare che il drone, con le sue braccia disposte radialmente, e l'angolo diedro quasi nullo, produce una spinta complessiva solidale all'asse  $Z_{body}$ , e, di conseguenza, spesso parallela all'asse  $Z_{NED}$ . All'atto pratico ciò significa che non esiste alcun metodo per il drone di ricreare fedelmente l'accelerazione suggerita dalla PN, essendo questa suscettibile a variazioni di modulo e direzione, ma sempre complanare al piano XY. L'accelerazione dalla PN tuttavia fornisce un chiaro spunto per un controllo automatico del drone: avendo questa lo scopo di deviare la traiettoria del drone, sembra plausibile l'idea di modificare opportunamente l'angolo di imbardata per ricreare il comportamento dell'automobile, dove l'angolo di Yaw riveste la funzione del volante (il quale non sempre fa coincidere l'assetto dell'auto con la direzione della traiettoria, basti pensare alla manovra di derapata, e che quindi va gestito opportunamente per un pilotaggio sicuro ed efficiente) mentre l'angolo di beccheggio gestisce la direzione di avanzamento del velivolo come il pedale di un acceleratore.

Tutto questo si traduce, come riportato, partendo quindi dalla conoscenza della accelerazione calcolata dal modello PN e nota la velocità, la posizione e l'assetto dei droni Hunter e Target.

#### **Controllo dell'angolo di imbardata**

Prima di tutto è necessario tradurre il vettore accelerazione in coordinate body, e ciò è possibile grazie all'assetto del drone Hunter  $[\varphi, \theta, \psi]$ .

$$\vec{a}_{PN_{body}} = M_{body}^{NED} \cdot \vec{a}_{PN_{NED}}$$

La 2° componente del vettore ricavato è quella predominante, essendo diretta lungo l'asse  $y_{Body}$ . Da questa seconda componente si ricava un angolo di deviazione, " $d^\circ$ " con il quale modificare, istante per istante, l'angolo iniziale di imbardata. L'angolo di deviazione è determinato dal rapporto tra l'intensità dell'accelerazione e la norma della velocità di avanzamento:

$$d^\circ = -k \frac{a_{PNybody}}{|V_{2D}|} \quad \text{con} \quad V_{2Dbody} = (V_{xNED}^2 + V_{yNED}^2)^{1/2}, k = 100$$

Per evitare sterzate troppo brusche, le deviazioni che superano una determinata soglia, " $\delta^\circ$ ", vengono ridotte appositamente.

$$|d^\circ| = \min\left(\delta^\circ + \frac{d^\circ}{\delta^\circ}, d^\circ\right) \quad \text{con} \quad \delta = 15$$

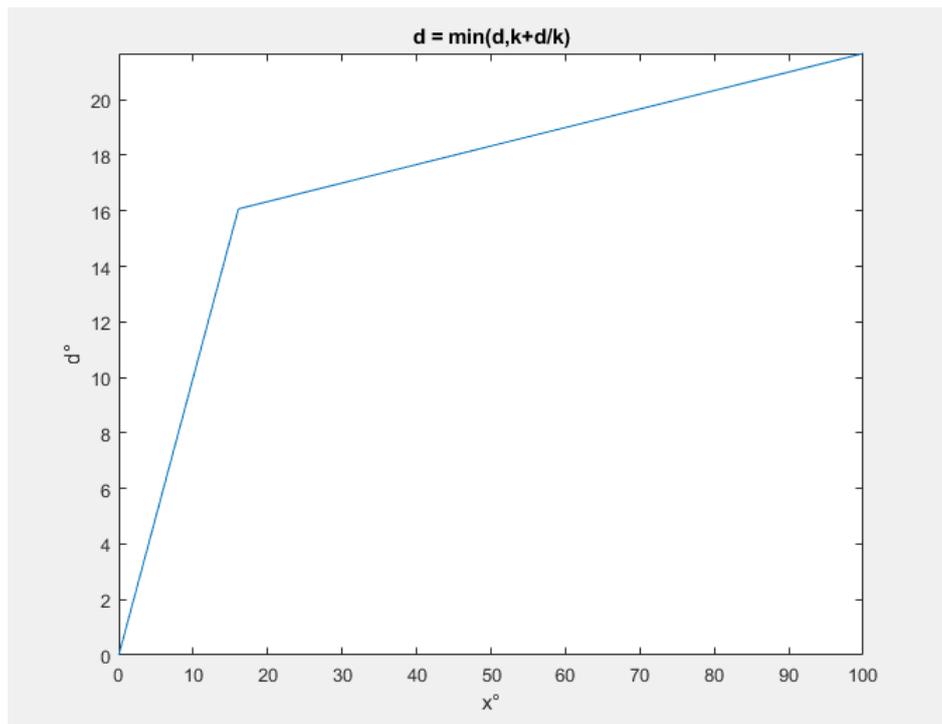


fig. 10 – andamento della riduzione dell'angolo di deviazione ideale.

Si ottiene, integrando, l'angolo ideale di imbardata che il drone ha l'obiettivo di assumere.

$$\psi_{id}^{\circ} = \int_{\psi_{0H}^{\circ}} d^{\circ} dt$$

### Controllo dell'angolo di beccheggio

In un primo momento si può prendere come fattore principale per l'angolo ideale di beccheggio la distanza tra cacciatore (H) e obiettivo (T), ottenendo empiricamente:

$$P^{\circ} = \max(-2^{\circ}, -(0,3^{\circ} + \log_{10}(Dist + 1)))$$

con  $Dist = ((x_H - x_T)^2 + (y_H - y_T)^2)^{\frac{1}{2}}$

Si noti come all'aumentare della Distanza vi sia un aumento graduale del modulo dell'angolo di Pitch (angolo che deve essere negativo per l'avanzamento del drone). Tuttavia, in determinati casi, se la distanza tra Hunter e Target è troppo elevata, è opportuno arricchire il modello con altri vincoli: infatti se la velocità del drone cacciatore fosse inferiore a quella del suo obiettivo, non ci sarebbe possibilità di recupero. Sia " $\beta$ " il rapporto desiderato tra velocità di avanzamento del drone H rispetto al drone T. Imporremo:

$$\beta(Dist) = \min(1.3, 1 + 0.005 \cdot D)$$

Tale rapporto va diminuendosi quando la distanza tra i due velivoli si riduce: lo scopo è quello di dotare il cacciatore di una velocità simile a quella del suo obiettivo per consentirgli una maggiore precisione nella manovra. Infatti sarebbe molto difficile dover centrare un obiettivo quando non si conosce il comportamento di questo, ed ogni possibile deviazione all'ultimo momento da parte del target pregiudicherebbe gravemente il successo della missione. Lo scarto prodotto sarà quindi il secondo fattore influenzante l'angolo di pitch.

$$\Delta S = |\beta \cdot \vec{V}_{T2D} - \vec{V}_{H2D}| \quad \text{con } \vec{V}_{H2D} = \begin{bmatrix} U_H \\ V_H \end{bmatrix}_{NED}$$

La distanza di scarto tra velocità ideale richiesta dal drone (dipendente dalla velocità reale del target) e la velocità di avanzamento reale del drone appena descritta genera un nuovo angolo di beccheggio unitamente a quello precedentemente imposto.

$$\theta^{\circ}_{id} = P^{\circ} + \int_{\theta_{0H}^{\circ}} (-k \cdot \Delta S) dt \quad \text{con } k_{empirico} = 0.1$$

Ovviamente, dovendo prevenire possibili ribaltamenti del drone, si può inoltre imporre un limite all'angolo massimo di beccheggio:

$$|\theta^{\circ}_{id}| = \min(\theta^{\circ}_{max}, \theta^{\circ}_{id})$$

### Controllo della spinta totale

Il problema del raggiungimento di una quota assegnata può essere affrontato dal punto di vista monodimensionale. Nota infatti la quota del target e la quota e velocità del cacciatore, otteniamo:

$$\Delta Z = Z_T - Z_H$$

La differenza di quota andrà quindi ridotta attingendo ad un profilo di velocità verticale caratterizzato da un andamento 'morbido', quindi logaritmico che tenderà asintoticamente al valore nullo. Un profilo che, infatti, predisponga un *cambio di segno* qualora il cacciatore raggiungesse la quota bersaglio con una velocità residua troppo elevata. La rapidità con cui la quota bersaglio viene raggiunta può essere modulata empiricamente modificando la pendenza globale della logaritmica grazie ad un coefficiente "k".

$$V_{zid} = k \cdot \log_{10}(1 + |\Delta Z|) \quad \text{con } k = 2$$

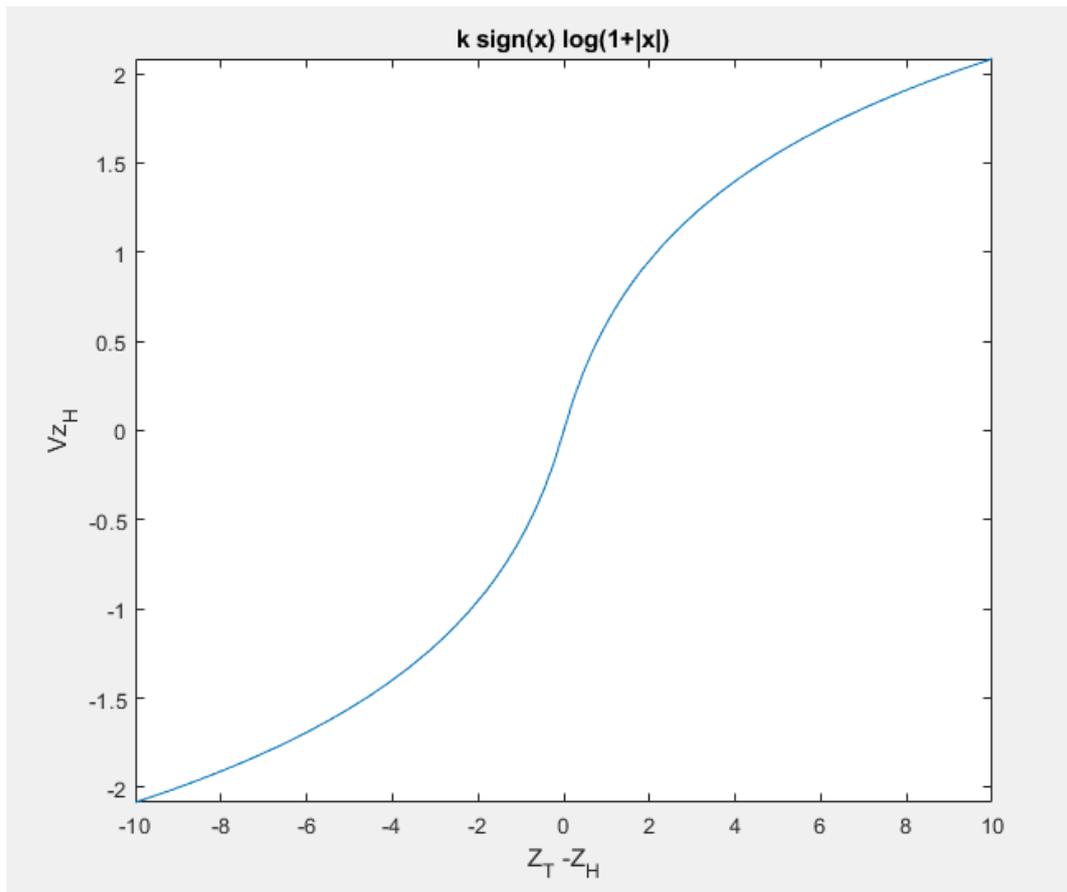


fig. 11 – profilo di velocità verticale adottato per il raggiungimento della quota bersaglio.

L'accelerazione a cui il drone sarà soggetto dovrà quindi vincere la differenza di velocità ideale-reale istantanea oltre al fattore costante dell'accelerazione gravitazionale.

$$a_z = g + V_{zid} - V_{zreale}$$

Mutando da un problema monodimensionale a uno tridimensionale, per ricavare la forza complessiva sarà sufficiente effettuare un cambio di coordinate body ->NED, e moltiplicare la accelerazione per la massa del drone.

$$F_{tot} = \frac{m_H \cdot a_z}{\cos(\varphi)\cos(\vartheta)}$$

## Regolazione dei segnali in ingresso

Uno dei vantaggi connessi all'utilizzo dei regolatori PID consiste nella possibilità di effettuare la taratura dei parametri sulla base di semplici prove sperimentali, prescindendo dalla formulazione matematica, non sempre agevole, del sistema sotto controllo. Tra i diversi metodi empirici per la sintonizzazione dei regolatori PID, regolatori già implementati nella libreria di simulink e facilmente modificabili, è doveroso accennare alla **taratura di Ziegler e Nichols** da cui si è tratto spunto per modulare i valori in uscita dalla sezione PILOTA AUTOMATICO prima di sottoporli alla sezione MOTOR MIXER.

Consideriamo un regolatore che esercita un'azione di controllo dipendente dall'errore attraverso la seguente legge:

$$u(t) = K_P \cdot e(t) + K_I \int_0^t e(\tau) d\tau + K_D \cdot \frac{d e(t)}{dt}$$

La legge di controllo è quindi composta da:

- un'azione Proporzionale all'errore;
- un'azione Integrale sull'errore;
- un'azione Derivativa sull'errore.

Questo tipo di regolatori prende quindi il nome di PID. I tre guadagni che compaiono nella legge di controllo vengono chiamati:

$K_P$  : guadagno proporzionale. All'aumentare di tale valore, il sistema riduce la durata del proprio transitorio.

$K_I$  : guadagno integrale. Più lo si aumenta, più il sistema è restio a passare velocemente al valore desiderato.

$K_D$  : guadagno derivativo. Più lo si aumenta, più le oscillazioni nel transitorio riportano il sistema al suo valore asintotico desiderato, e quindi aumentano di intensità.

Il metodo si articola nei seguenti passi, ed è stato operato su ogni canale, silenziando gli altri:

1) Si chiude l'anello di controllo con il regolatore PID (i cui parametri devono essere sintonizzati), imponendo nulle le azioni integrale e derivativa:  $K_I = 0$ ,  $K_D = 0$ .

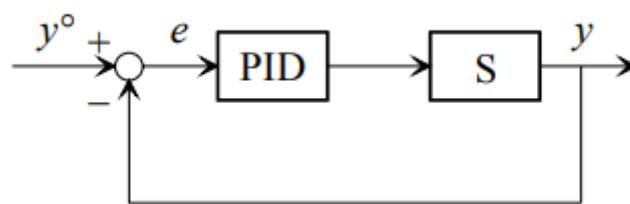


fig. 12 – schema elementare di un sistema regolato, oggetto classico di studio dei controlli automatici.

2) Partendo da valori molto piccoli di  $K_P$  si effettua un semplice esperimento, consistente nell'applicare un piccolo gradino al segnale di riferimento.

3) Si aumenta progressivamente  $K_P$  ripetendo di volta in volta l'esperimento finché non si instaura nell'anello un'oscillazione permanente, come in figura.

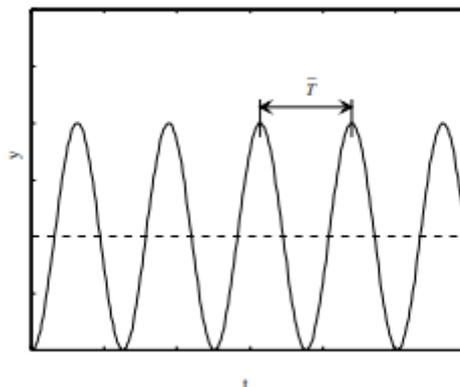


fig. 13 – andamento dell'uscita di un sistema con Guadagno Critico. L'andamento è di tipo sinusoidale e non accenna a modificare tale condizione.

4) Detto  $K_P$  il valore del guadagno proporzionale corrispondente all'oscillazione permanente (guadagno critico) e  $T$  il periodo di tale oscillazione, si tarano i parametri di un regolatore P, PI o PID sulla base della seguente tabella:

	$K_P$	$T_I$	$T_D$
<b>P</b>	$0.5 \bar{K}_P$	-	-
<b>PI</b>	$0.45 \bar{K}_P$	$\frac{\bar{T}}{12}$	-
<b>PID</b>	$0.6 \bar{K}_P$	$\frac{\bar{T}}{2}$	$\frac{\bar{T}}{8}$

tab. 2 – coefficienti evidenziati dalla taratura di Ziegler Nichols una volta ricavato il guadagno critico e il periodo dell'uscita del sistema.

Seguono i valori dei controllori PID per i canali del PILOTA AUTOMATICO, valori di cui è bene sottolineare la forte connessione con il tipo di motore utilizzato nella simulazione: al variare infatti delle prestazioni del motore, sarà necessario aggiornare i regolatori con nuovi parametri per garantire la massima efficienza durante la missione.

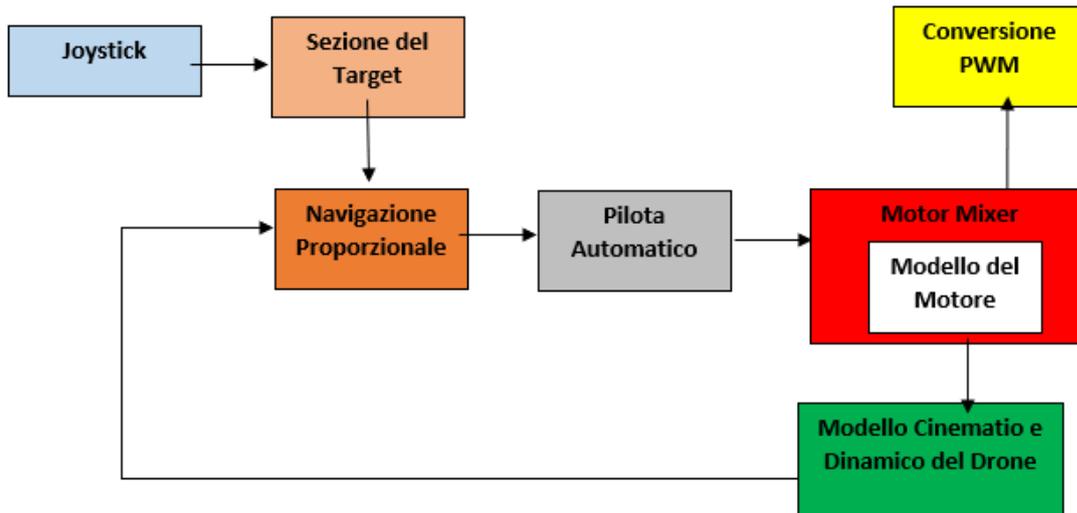
	$F_{tot}$	$T_{roll}$	$T_{pitch}$	$T_{yaw}$
<i>Proportional (P)</i>	1	25	250	0.01
<i>Integral (I)</i>	0	1	25	0.0005
<i>Derivative (D)</i>	0	15	1	1

Tab. 3 – coefficienti dei controllori PID implementati nel software progettato in Simulink associati al motore F100 per prestazioni sportive.

Il metodo non è sempre applicabile per tutti i tipi di sistemi: ci sono infatti alcuni che non generano oscillazioni, anche con guadagni proporzionali elevati. Altre volte può essere pericoloso, o comunque sconsigliabile, portare il sistema al limite di stabilità.

## SCHEMA GERARCHICO DELL'ALGORITMO

Lo schema presentato in questo paragrafo dovrebbe riassumere i concetti esposti finora nel corso della trattazione per favorire una visione d'insieme e valorizzare quanto proposto.



*fig. 14 – rappresentazione a blocchi della struttura della simulazione sviluppata in Simulink.*

Con una tabella si descrive sinteticamente la funzionalità di ogni sezione:

Nome della sezione	Input	Output	Descrizione
Joystick	I valori di tensione di un comune controller analogico disponibile nel mercato elettronico.	Un vettore di 4 elementi con valori compresi tra -1 ed 1.	Al sistema può essere aggiunto un joystick per controllare la posizione del drone bersaglio e quindi dirigere la simulazione in modo autonomo.

Sezione del Target	I valori in uscita da un joystick <i>oppure</i> Nessun input.	La posizione del drone Target e la sua velocità nel sdr NED.	Le forze che agiscono sul target vengono integrate nel tempo ottenendo così la dinamica del velivolo bersaglio. In assenza di input, simulink genera un segnale soggetto ad oscillazioni casuali con un valor medio determinato dall'utente, e lo integra nel tempo.
Navigazione Proporzionale	Posizione e Velocità di drone Hunter e Target.	Un vettore che esprime la accelerazione che l'hunter deve adottare per correggere la traiettoria.	L'intensità dell'accelerazione imponibile all'Hunter è proporzionale all'angolo di vista tra I due velivoli.
Pilota Automatico	Posizione, velocità, assetto del drone H e posizione e velocità del drone T.	Un vettore con 4 elementi contenente I valori di: [Forza totale, Coppie] da indurre nel drone H.	Questa sezione determina quale comando il drone deve tentare di eseguire, è qui che inoltre si applica la regolazione PID per ogni canale in uscita.
Motor Mixer	Un vettore contenente 4 valori (ideali).	Un vettore contenente N valori compresi tra 0 e 100, simboleggianti I valori di manetta che gli N motori devono assumere; Un vettore contenente I 4 valori (reali), frutto dell'inseguimento del vettore in ingresso.	Questa sezione traduce la richiesta del modello della Navigazione Proporzionale in una richiesta pratica basata sui valori di manetta imponibili al drone.

Modello del Motore	Un valore di manetta possibilmente compreso tra 0 e 100.	Un vettore contenente la Spinta, la Corrente, la Tensione, gli RPM	Assegnato un insieme di parametri iniziali, assegnato un valore di manetta si otterranno tutti i valori delle variabili fisiche più significative caratterizzanti un motore elettrico.
Conversione PWM	Un set con N valori di manetta.	Un vettore con N elementi.	In questa sezione i valori di manetta normalizzati da 0 a 100% vengono convertiti come valori di tensione che la centralina del drone deve realmente assegnare ai motori.
Modello Cinematico e Dinamico del Drone	Spinte e Coppie dei motori;	Posizione, Velocità e Assetto del drone H;	In questa sezione si procede al calcolo delle variabili cinematiche inerenti al Drone, inteso come corpo rigido ideale.

Tab. 4 – descrizione dei sottosistemi del software.

## L'INTERFACCIA PER L'UTENTE

Con la finalità di facilitare l'utente nella preparazione di una simulazione personalizzata, è stata creata un'interfaccia grafica con cui egli potesse interagire, anziché intervenire, modificando i parametri delle condizioni iniziali, direttamente all'interno del codice informatico. L'interfaccia non è caratterizzata da una complessità elevata e possiede dei "tool" la cui funzionalità si presenta particolarmente intuitiva. L'utente potrà intervenire manualmente premendo pulsanti, selezionando un'opzione proposta da menu a tendina, compilando caselle di testo con la tastiera, o semplicemente spuntando alcuni box. La modifica di ogni impostazione sarà poi riscontrabile confrontando i dati nel workspace di matlab.

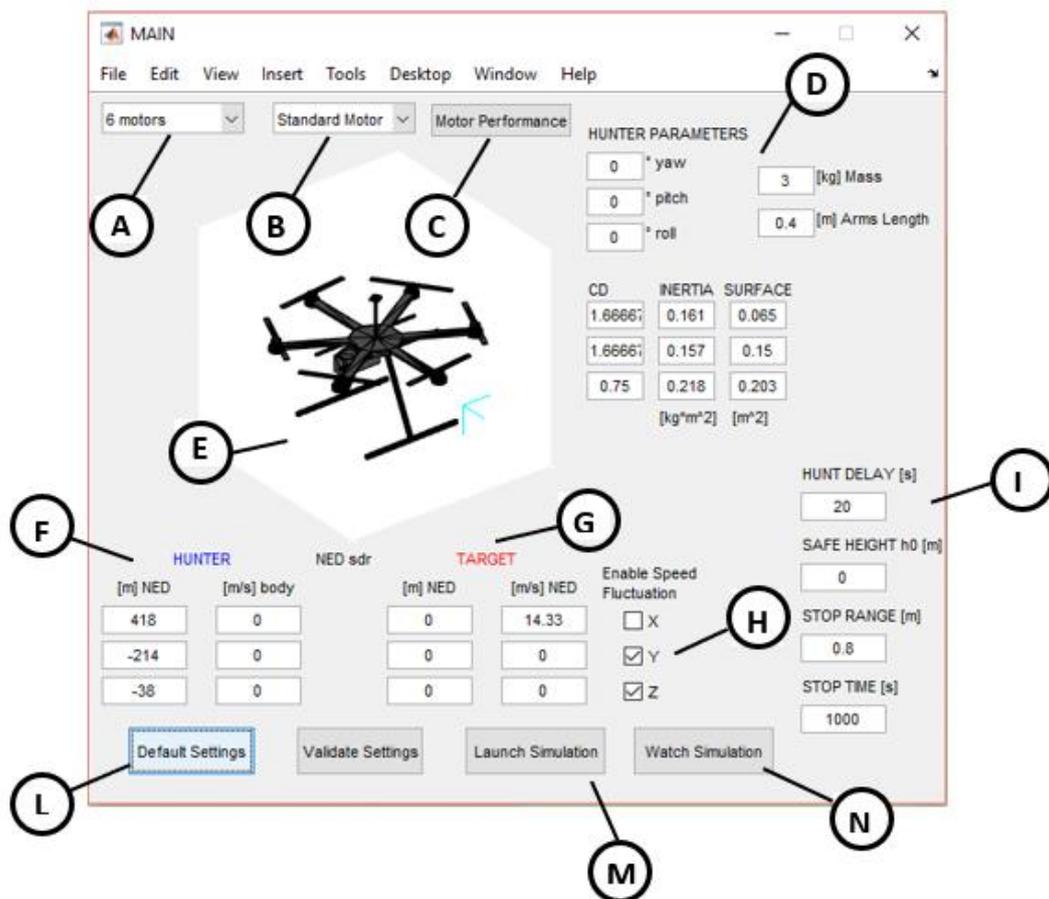


fig. 15 – immagine dell'interfaccia ricreata in Matlab (ver 2018a) a disposizione dell'utente.

Presentiamo qui l'immagine dell'interfaccia accompagnata da un elenco di annotazioni sulle funzionalità che propone.

A) **Menu a Tendina:** l'utente può selezionare la tipologia di drone scegliendo tra l'esacottero ed il quadricottero.

B) **Menu a Tendina:** una volta definito il numero di eliche e motori, è possibile selezionare il tipo di motore (e di conseguenza i parametri con le prestazioni ad esso associate) con cui equipaggiare il velivolo.

C) **Pulsante:** alla pressione del pulsante l'utente potrà visualizzare, in una finestra esterna che comparirà all'occorrenza, i parametri associati al motore selezionato.

D) **Insieme di caselle di testo compilabili:** l'utente può determinare le caratteristiche inerenti al drone inseguitore.

E) **La Figura del modello 3D del drone:** l'orientamento spaziale del drone si aggiorna automaticamente quando l'utente modifica gli angoli di assetto iniziali inclusi nell'insieme descritto al punto D.

F) **Insieme di caselle di testo compilabili:** l'utente può disporre il drone cacciatore nello spazio assegnando una posizione, in coordinate (NED), ed una velocità iniziale, in coordinate (Body), qualsiasi.

G) **Insieme di caselle di testo compilabili:** il discorso si ripropone anche per il drone

bersaglio come descritto al punto precedente. Si noti che non è possibile impostare l'assetto iniziale del drone bersaglio poiché si ipotizza, per semplicità, che assuma un assetto solidale, durante tutta la simulazione, alla propria velocità.

H) **Insieme di box a spunta:** qualora l'utente non avesse intenzione di controllare manualmente la traiettoria del drone bersaglio si attiverebbe un sistema per la modifica casuale dei valori impostati descritti al punto precedente. Più in particolare, questo elenco di box chiede all'utente lungo quali assi il drone bersaglio è autorizzato ad oscillare: per esempio, se l'utente spunta per esempio la casella "Z" allora il drone modificherà casualmente la propria quota durante la simulazione.

I) **Insieme di caselle di testo compilabili:** terminato l'insieme dei parametri e delle condizioni iniziali assegnabili, l'utente può anche agire in merito al comportamento della simulazione stessa, determinando:

- un intervallo di tempo iniziale entro il quale il drone cacciatore non attiva il sistema di inseguimento ma si limita a conservare le sue condizioni iniziali.
- il valore di una altezza "extra", che il drone cacciatore deve mantenere rispetto alla quota del drone bersaglio. Se tale altezza è settata sul valore nullo, vuol dire che l'utente desidera che i due velivoli collidano.
- il valore massimo con cui la cattura può considerarsi verificata: essendo i due velivoli ben diversi da due punti materiali, la distanza dei due baricentri ammissibile ha un range più ampio.
- il limite massimo di tempo oltre il quale la simulazione non può proseguire qualora la cattura non sia ancora stata soddisfatta.

L) **Pulsante:** l'utente può anche compilare con un solo click tutte le impostazioni dell'interfaccia prendendo da un set predefinito di valori, e da lì modificare manualmente solo quei parametri a lui congeniali.

M) **Pulsante**: questo pulsante lancia la simulazione in ambiente simulink, e richiede alcuni istanti di tempo prima che l'utente possa tornare a controllare l'interfaccia. Se la convalida dei parametri immessi in input ha prodotto qualche errore la simulazione non può essere lanciata. Se la simulazione si conclude con successo allora il pulsante descritto al punto successivo viene abilitato.

N) **Pulsante**: terminata la simulazione, alla pressione di questo strumento, si aprirà una finestra esterna all'interfaccia in cui poter visualizzare graficamente il comportamento dei due velivoli mediante una breve animazione in 3D.

## IL SOFTWARE IN AMBIENTE SIMULINK

Il software implementato in ambiente simulink riproduce l'algoritmo descritto in precedenza rispettando la suddivisione delle macrosezioni. I principali fattori che hanno portato all'utilizzo di tale linguaggio sono stati:

- Semplificazione del codice, per mezzo della gestione dei calcoli con la rappresentazione a blocchi, in larga parte già implementati nella libreria interna di Simulink.
- Integrazione con Matlab, linguaggio di alto livello con cui risulta conveniente organizzare grandi moli di dati.

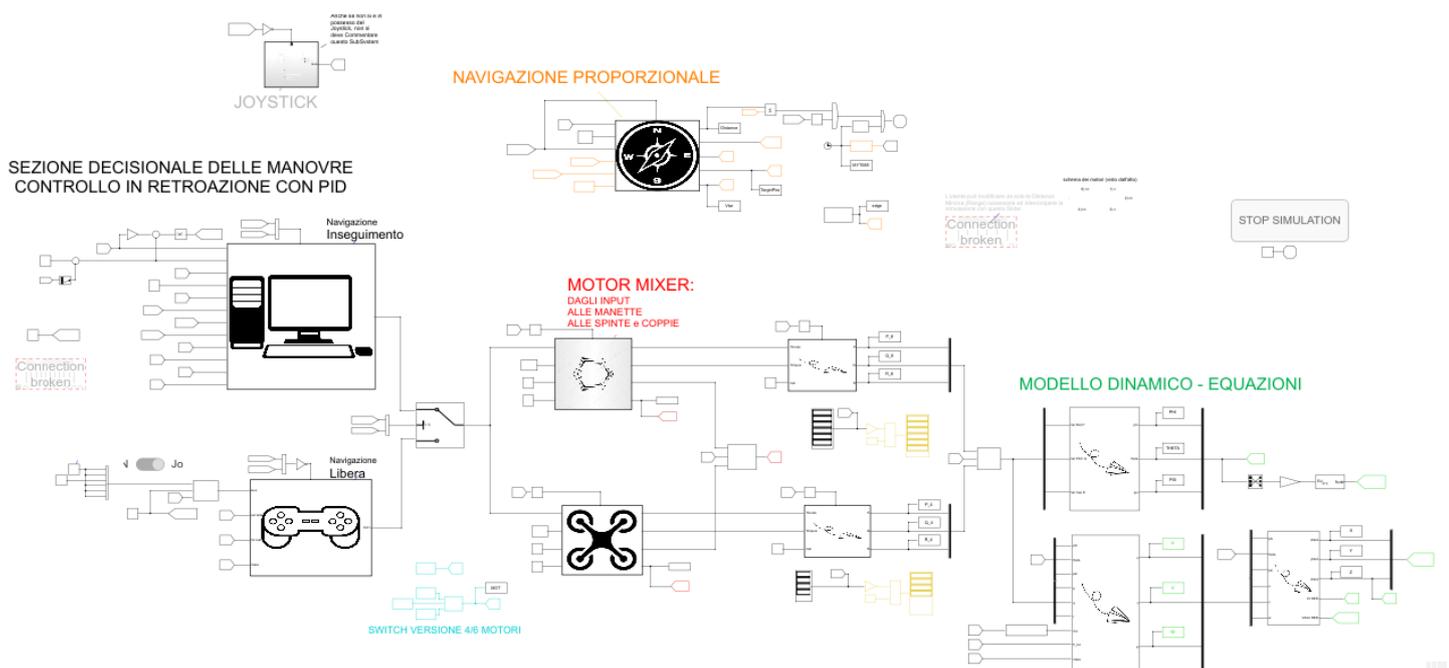


fig. 16 – panoramica generale del software di simulazione implementato in Simulink. La distinzione dei sottosistemi è facilitata dall'immagine contenuta nel proprio blocco.

Alcuni accorgimenti che hanno semplificato la programmazione in simulink: la ciclicità con cui le variabili vengono computate, vestendo la duplice funzione di

input e output del sistema ha richiesto un metodo elegante per collegare blocchi del programma collocati in posizioni distanti fra loro. L'utilizzo dei blocchi "From.m" & "GoTo.m", caratterizzati dalla tipica forma ad *etichetta*, ha risolto questo problema prevenendo la comparsa di lunghissime linee di collegamento all'interno del sistema.

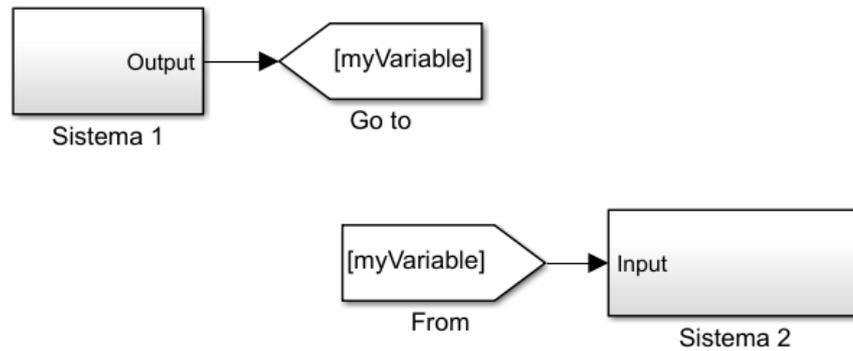


fig. 17 – esempio della retroazione gestita con l'ausilio dei blocchi GOTO e FROM presenti in libreria.

Solo alcune funzioni di Matlab sono state implementate nella libreria di simulink, ma questo non costringe il programmatore a ricreare ogni funzione di cui necessita con dei sistemi a blocchi: infatti è possibile impiantare all'interno del programma in simulink alcune functions o delle pagine di script.m da far eseguire al programma come se si fosse in presenza dell'ambiente di programmazione Matlab. Ciò di cui l'utente ha bisogno è il blocco "MATLAB function".

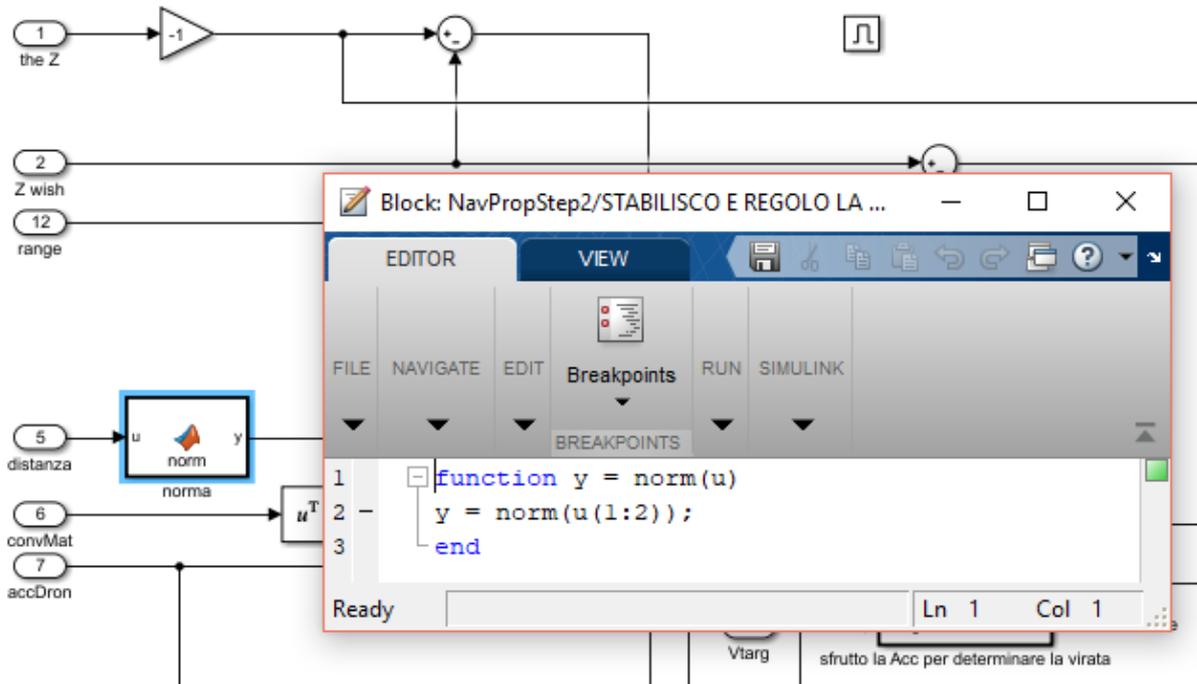


fig. 18 – esempio del richiamo di uno script Matlab all'interno del software progettato in Simulink.

Simulink risponde perfettamente all'esigenza di gestire un discreto numero di equazioni differenziali da integrare istante per istante grazie al blocco 'integrator' con il quale è possibile impostare il valore iniziale ed anche la condizione di reset del blocco stesso.

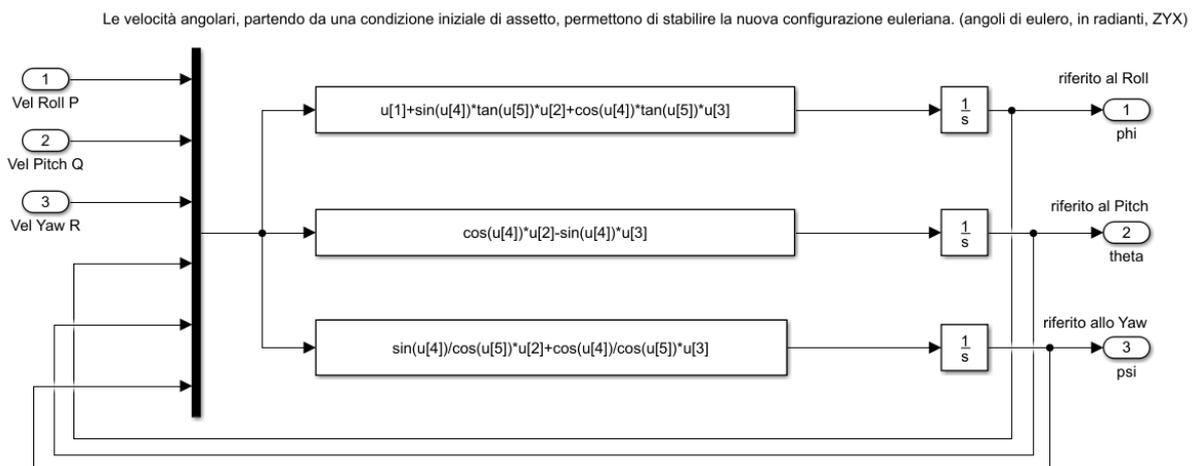
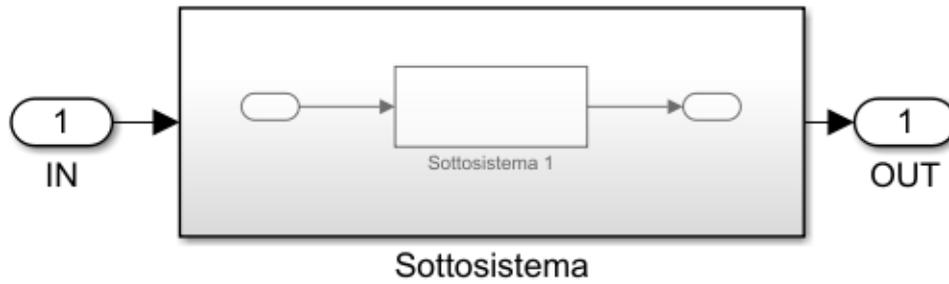


fig. 19 – dettaglio del software di simulazione, rappresentante il sistema delle equazioni differenziali inerenti all'assetto del drone Hunter. I tre blocchi integratori posti a seguito dei comandi permettono una soluzione numerica efficace e rapida per lo svolgimento della simulazione.

Ultima importante funzionalità di Simulink è quella di poter scomporre la programmazione di un sistema molto complesso in sottosistemi di complessità inferiore tra loro comunicanti e connotati da una gerarchia. Molto frequentemente infatti l'utente troverà sottosistemi contenuti all'interno di altri sottosistemi.



*fig. 20 – esempio di schema a blocchi innestati, dalla tipica struttura a Matrioska.*

## **RICOSTRUZIONE GRAFICA DELLA SIMULAZIONE**

I droni sono velivoli di dimensioni modeste facilmente schematizzabili per quanto riguarda la rappresentazione di un modello fisico-matematico, ma, come tutti gli oggetti prodotti dalle grandi industrie, sono artefatti geometricamente complessi dal punto di vista del design. Sarà necessario dunque accennare qualcosa sulla rappresentazione delle geometrie 3D, dal momento che l'interfaccia si preoccupa di computare e inscenare la simulazione di due velivoli multirotores.

### **Il formato STL**

Il formato .STL (Standard Triangulation Language) è un formato di file, binario o ASCII, nato per i software CAD, ed è forse il più comunemente usato, ed è utilizzato nella prototipazione rapida, ed anche i macchinari per la stampa 3D ne fanno ricorso.

Un file di questo formato rappresenta un solido la cui superficie è stata discretizzata in triangoli. Esso consiste di un elenco di coordinate [ X Y Z ] per ogni vertice di ogni triangolo che compone la superficie. I calcolatori pertanto rappresentano solamente l'involucro di un solido che viene considerato "pieno", dotato di massa.

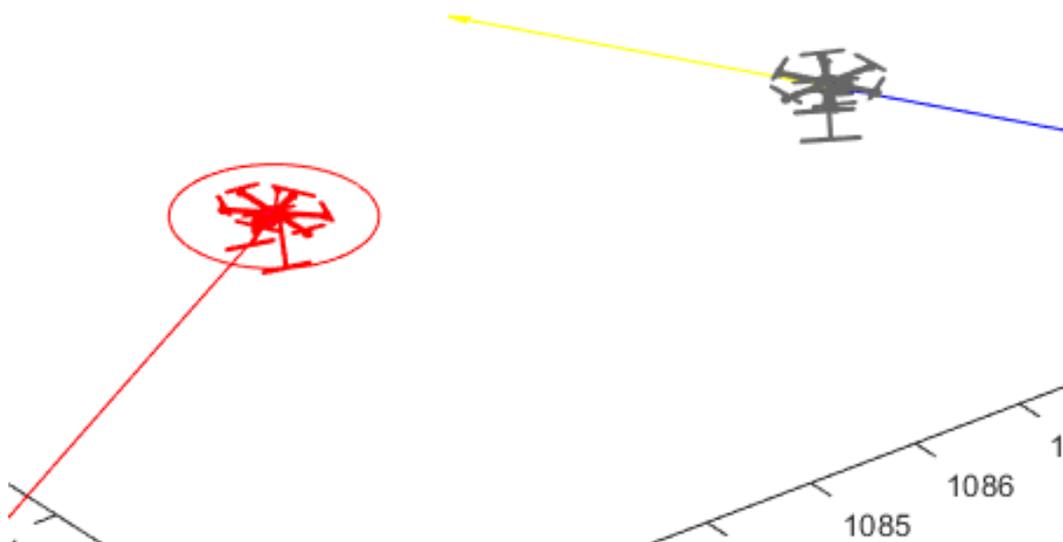
Questo si conferma appunto un metodo con molti vantaggi, quali la semplicità per essere generato e processato, mentre il suo principale sfavore sta nel fatto che la sua è una geometria approssimata.

### **La rappresentazione 3D in Matlab**

Un potente strumento presente nella libreria di Matlab è la funzione "patch.m", la quale appunto ricrea la superficie 3D di un oggetto se sono note le coordinate dei vertici e l'organizzazione dei triangoli (esempio: Triangolo 1: [vertice 23, vertice 4,

vertice 12]; Triangolo 2: [vertice 12, vertice 23, vertice 45]; ... ). Questi dati sono stati ottenuti estrapolandoli da un modello CAD di elicottero ricreato con il programma CATIA, più precisamente il drone DJI S800 EVO.

Per ricreare l'assetto è stato utilizzato il prodotto matriciale, effettuando il passaggio di coordinate dal sdr Body (quelle su cui la geometria si basa di default) a Cartesiano. Per inscenare l'animazione è stato poi sufficiente salvare la geometria elaborata dalla funzione Patch nella memoria, e, durante il corso della simulazione, traslarla in modo solidale al baricentro del velivolo.



*fig. 21 – istantanea di simulazione generata in ambiente Matlab. Il drone Hunter, caratterizzato da superficie nera, si appresta a raggiungere il drone Target, con superficie rossa.*

La simulazione viene ricreata interamente in ambiente Matlab, e consta di un grande loop in cui vengono mostrati a video elementi 3D, dati numerici e diagrammi. I dati riportati sono ottenuti dai risultati della simulazione in ambiente Simulink, e la struttura della finestra d'animazione si suddivide in 3 regioni.

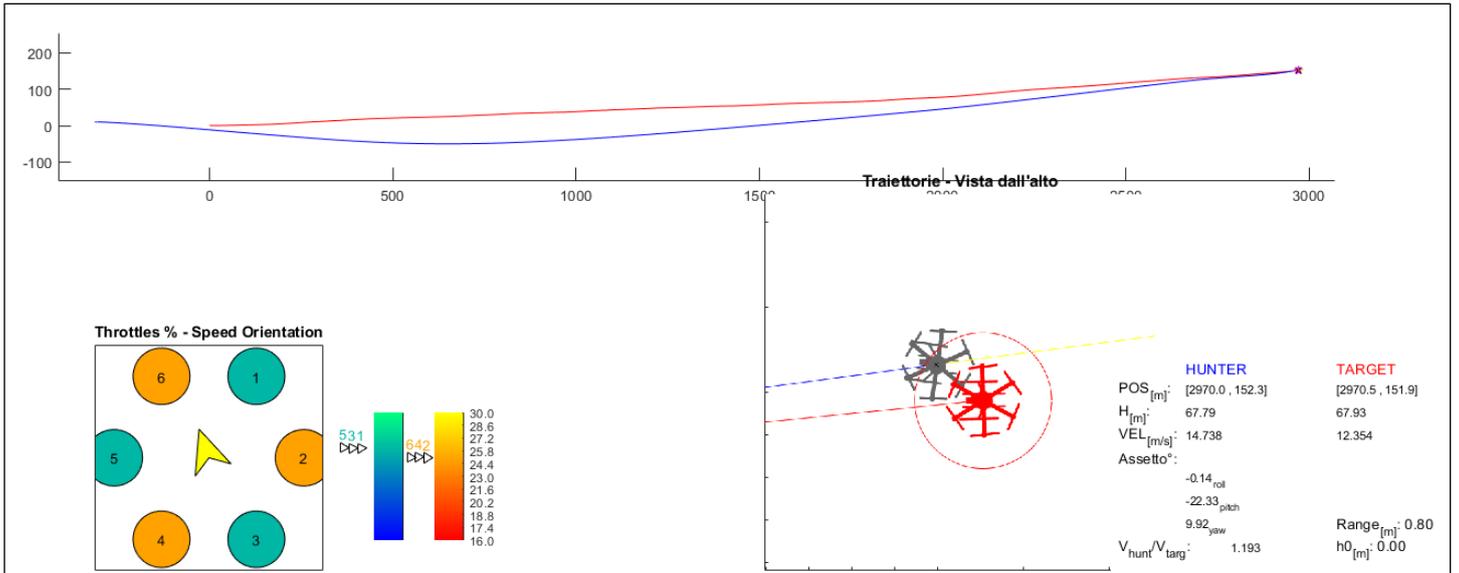


fig. 22 – panoramica sull’interfaccia grafica, in ambiente Matlab, per l’osservazione della simulazione pre-computata dal software in Simulink.

- Nel plot superiore si osserva la riproduzione, istante dopo istante, della missione. Il drone Hunter (superficie di color nero) descrive una traiettoria evidenziata dalla linea blu, e tenta di raggiungere il suo Target (in rosso) la cui traiettoria è descritta nel tempo dalla linea rossa. Attorno al drone Target è collocata una circonferenza che evidenzia l’area entro cui il raggiungimento può considerarsi compiuto.
- Nel plot inferiore a sinistra si osserva una panoramica sui valori di manetta dei motori del drone Hunter. Essendo i motori suddivisi per rotazione, oraria o antioraria, è stato opportuno suddividere cromaticamente i due gruppi, dedicando due sfumature ben distinte.

Nel plot si osserva la vista dall’alto dei motori dell’esacottero, rappresentati da “led circolari” capaci di cambiare colore, passando da una sfumatura più accesa a una più scura, a seconda del proprio livello di manetta. Solitamente i valori di manetta rimangono tra il 20% ed il 50%.

Ogni motore ha un proprio cursore triangolare che aiuta l'utente con la lettura dei valori di manetta, scorrendo lungo la legenda posta sul lato destro del plot. Al centro di questo plot, circondato dai sei cerchi numerati del drone, si muove un indicatore a punta. Questo indicatore mostra all'utente la direzione della velocità (nel riferimento Body, ovviamente) del drone Hunter, chiarificando quindi se il drone sta volando dritto o se è presente una componente di drift laterale.

- Nell'ultimo plot, in basso a destra, è presente una tabella con i valori fisici più significativi, i quali si aggiornano fotogramma per fotogramma, mettendo in evidenza fattori come la posizione e velocità dei due velivoli, ed il rapporto tra le velocità nel piano XY.

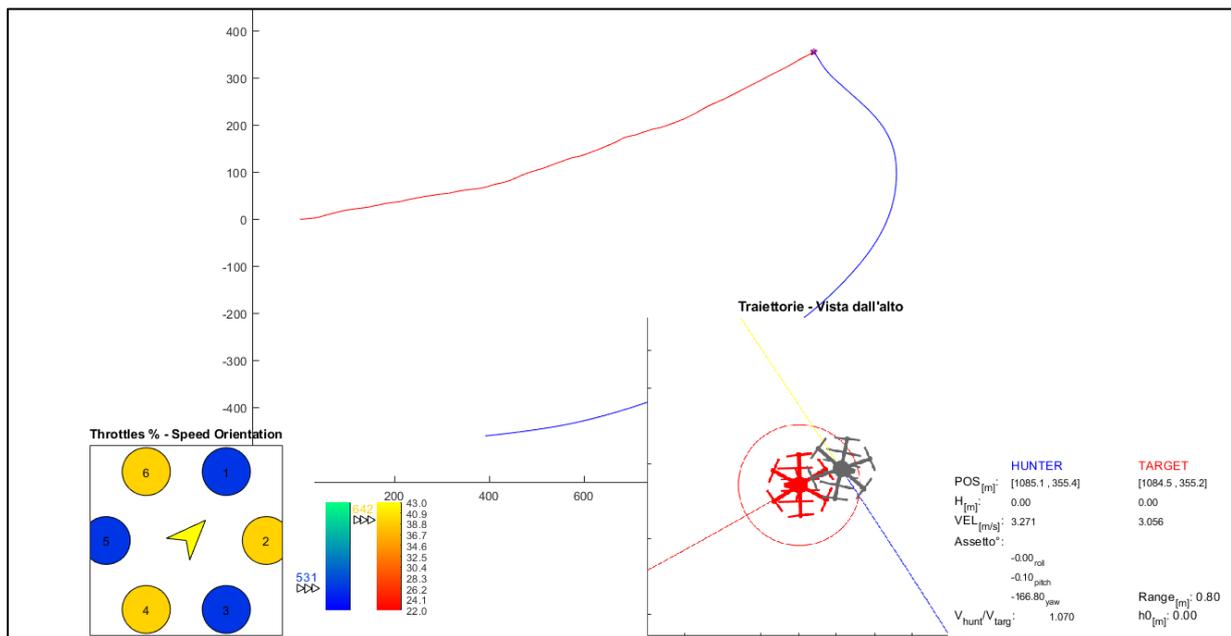


fig. 23 – panoramica dell'interfaccia relativa ad un'altra simulazione, determinata da condizioni iniziali differenti.

## CONCLUSIONI

L'applicazione del modello della navigazione proporzionale per questa tipologia di velivoli si presenta interessante, e ci sono tutti i presupposti per ottenere miglioramenti. Il modello si concretizza con facilità proprio grazie alla semplicità delle formule che lo compongono, fornendo, in molte simulazioni, ma non tutte, ottimi risultati, riuscendo a raggiungere velivoli con velocità che variano dai 5 ai 20 m/s lungo una direzione preferenziale.

Un difetto del simulatore è stato riscontrato proprio nelle simulazioni in cui la missione non viene portata a termine: solitamente il fattore principale appartiene al campo dell'instabilità del velivolo. Infatti, a seguito di una virata troppo forte, spesso causata da una manovra troppo repentina del target a pochi secondi dalla collisione, il drone Hunter raggiunge un assetto critico e perde il controllo. Probabilmente il rimedio a questo problema deve essere ricercato in una taratura migliore dei regolatori PID, i cui valori però, variando da modello a modello di velivolo, non suscitano particolare interesse per un miglioramento del software. Sarebbe molto più ragionevole agire solamente in prospettiva di una simulazione reale, in cui il drone utilizzato è noto, ed i suoi parametri rimangono invariati e di facile misurazione.

La navigazione proporzionale contraddistingue la traiettoria di ogni simulazione con una tipica forma sinuosa, e più il coefficiente adimensionale (che ricordiamo essere compreso solitamente tra 3 e 5) aumenta, più il velivolo si appresta a disporsi parallelo alla velocità del suo bersaglio, chiudendo in poco tempo (in certi casi si potrebbe ritenere anche troppo poco) la manovra di virata.

Non è rara la presenza di qualche "scarto dell'ultimo secondo": il drone Target devia improvvisamente e l'Hunter, che è stato scartato, esegue una virata a 360°

ridisponendosi in coda ad esso per un nuovo riavvicinamento. Anche in questo fenomeno il coefficiente adimensionale ha una sua influenza: le curve quasi-circolari descritte dall'hunter sono più strette all'aumentare di  $N$ , ma c'è il rischio che il drone sia costretto a dover recuperare più scarti in sequenza, poiché è difficile eseguire una virata perfetta senza drift laterali, e ciò gli impedisce di accodarsi con la dovuta precisione all'obiettivo.

### **Sviluppi futuri**

La messa a punto di una simulazione reale potrebbe essere l'opzione più soddisfacente per questo lavoro, il quale potrebbe essere arricchito dall'aggiunta di sensori per il rilevamento della posizione GPS, la misurazione delle forze inerziali ed inoltre di strumenti per la telecomunicazione con una ground station. Una alternativa altrettanto interessante potrebbe essere lo sviluppo di un programma ausiliario che preveda la sostituzione del sottosistema "Motor Mixer" precedentemente descritto con un pilota automatico, creato grazie ad una rete neurale addestrata, capace di apprendere metodi più sicuri ed efficienti di pilotaggio e che impari autonomamente dalle simulazioni che esegue.

## BIBLIOGRAFIA & SITOGRAFIA

- Per la messa a punto del sistema per il controllo dinamico del drone inseguitore sono state adottate tutte le indicazioni presenti nelle dispense del Laboratorio di Meccanica del Volo 2018 – F. Giulietti docente Unibo nonché relatore di questa Tesi.

Motor Mixer (1).

[https://iol.unibo.it/pluginfile.php/183297/mod\\_resource/content/1/motor%20mixer.pdf](https://iol.unibo.it/pluginfile.php/183297/mod_resource/content/1/motor%20mixer.pdf)

Motor Mixer (2).

[https://iol.unibo.it/pluginfile.php/181320/mod\\_resource/content/1/Hexa.pdf](https://iol.unibo.it/pluginfile.php/181320/mod_resource/content/1/Hexa.pdf)

Equazioni differenziali.

[https://iol.unibo.it/pluginfile.php/180103/mod\\_resource/content/1/Multirotors\\_kinematics\\_and\\_dynamics GR.pdf](https://iol.unibo.it/pluginfile.php/180103/mod_resource/content/1/Multirotors_kinematics_and_dynamics_GR.pdf)

- È inoltre possibile assistere ad una dimostrazione sullo svolgimento del software, osservando una registrazione caricata online a supporto degli utenti meno esperti ma comunque interessati alla visione della simulazione.

1) Come impostare i parametri dell'interfaccia:

<https://vimeo.com/303830838>

2) Lancio ed Osservazione della simulazione generata:

<https://vimeo.com/303829710>

- Data sheet del motore F100-Long Shaft KV635 con cui poter personalizzare la simulazione. <http://store-en.tmotor.com/goods.php?id=477>

- Tabella relativa alla classificazione degli agenti autonomi presente nell'opera: Jacques Ferber, Multi-Agent Systems: An Introduction to Artificial Intelligence, Addison- Wesley, 1999.
- Differenziazione delle tipologie di navigazione proporzionale presso un articolo divulgativo Introduction to Proportional Navigation (Part I): <https://www.moddb.com/members/blahdy/blogs/gamedev-introduction-to-proportional-navigation-part-i>
- Utilizzo di immagini e formule relative al modello di navigazione proporzionale nella dispensa Guidance of Missiles, NPTEL,2012,D. Ghose, capitolo 9 <https://nptel.ac.in/courses/101108054/module8/lecture22.pdf>
- Altre tipologie di problemi relativi al pathfinding con approccio matematico disponibili sul sito “Base Cinque – Appunti di Matematica Ricreativa” <http://utenti.quipo.it/base5/geopiana/quattroform.htm>
- Data sheet del modello esacottero DJI Spreading Wings S800 EVO facilmente reperibile su siti di acquisto online: <https://www.dji.com/spreading-wings-s800-evo>
- Informazioni e chiarimenti sulla taratura dei regolatori PID con metodo di Ziegler Nichols evidenziati, anche con esempi numerici, nelle dispense di Automatica P. Rocco, docente del Politecnico di Milano.

<http://home.deib.polimi.it/rocco/leonardo/lez12.pdf>

- Informazioni generali inerenti alla progettazione e pilotaggio amatoriale del drone per una maggior comprensione e giustificazione dell'algoritmo utilizzato:

“Droni & Aerodinamica: Ecco come dovrebbe essere un drone” – Team di RecensioniDroni presenziato da R. Masutti.

<https://recensionidroni.com/droni-aerodinamica-ecco-come-dovrebbe-essere-un-drone/>

D. Pironi – “Come Pilotare un Drone: la Guida Definitiva”

<https://dariopironi.com/come-pilotare-un-drone-la-guida-definitiva/>

- Documentazione redatta dalla MathWorks inc. su cui sono stati sviluppati il software e le interfacce (rispettivamente in Simulink e in Matlab versione 2018a) in merito al funzionamento della funzione LSQLIN.m per la risoluzione di sistemi lineari con vincoli sul dominio.

<https://it.mathworks.com/help/optim/ug/lsqlin.html>

- La struttura dell'algoritmo di simulazione è stata creata trovando spunto da una precedente simulazione adattata su missili in ambiente Simulink redatta da P. Costello: “Simulink simulation of proportional navigation and command to line of sight missile guidance” – California, Calhoun: The NPS Institutional Archive Dspace Repository.

[https://calhoun.nps.edu/bitstream/handle/10945/31532/95Mar\\_Costello.pdf;sequence=1](https://calhoun.nps.edu/bitstream/handle/10945/31532/95Mar_Costello.pdf;sequence=1)