



Università degli Studi di Pisa  
Dipartimento di Ingegneria dell'Informazione

---

Corso di Laurea Specialistica in Ingegneria  
dell'Automazione

Tesi di laurea specialistica

## Tecniche di guida innovative in scenari complessi

Candidato:  
Lorenzo Cancemi  
Matricola 407248

Relatori:  
Prof. Mario Innocenti

Correlatori:  
Prof. Lorenzo Pollini

Anno Accademico 2012–2013



# INDICE

1	INTRODUZIONE	1
2	SISTEMA DI GUIDA A WAYPOINTS CON APPROCCIO FUZZY	5
2.1	La logica e i sistemi fuzzy	5
2.2	Introduzione alla guida fuzzy	9
2.3	Design del sistema di guida fuzzy	11
2.4	Implementazione Real-Time	16
2.4.1	Semplificazione del Sistema Fuzzy	17
2.4.2	Confronto tra fuzzy e fuzzy approssimata	19
2.4.3	Guida per brevi e lunghe distanze	21
2.4.4	Simulazione hardware-in-the-loop con autopilota ICARO	23
2.5	Simulazioni e risultati sperimentali	27
3	STUDIO DI VISIBILITÀ DI UN TARGET IN UN AMBIENTE STRUTTURATO	29
3.1	Introduzione	29
3.2	Definizione dello spazio di lavoro	30
3.3	Definizione regione di incertezza del target	31
3.4	Definizione del livello di visibilità	32
3.5	Metodo per il calcolo della direzione di visibilità	34
4	POSSIBILI APPLICAZIONI DELLO STUDIO DI VISIBILITÀ	37
4.1	Introduzione	37
4.2	Task di sorveglianza	38
4.3	Task di intercettazione	39
4.3.1	La guida Proportional-Navigation	39
4.3.2	La guida Proportional-Navigation con contributo di visibilità	40
4.3.3	Simulazioni	41
4.4	Task di intercettazione: Target manovrato	45
4.4.1	Considerazioni sull'agilità del veicolo	48
4.5	Task di intercettazione: Considerazioni analitiche	52
4.6	Task di intercettazione: Molteplici veicoli inseguitori	59
4.6.1	Differenziazione delle zone di approccio	60
4.6.2	Simulazioni: 2 veicoli inseguitori vs. 1 target	63
4.7	Task di intercettazione: Molteplici target	65
4.7.1	Simulazioni: 1 veicolo inseguitore vs. 2 target	66
4.8	Task intercettazione: 3 Veicoli	70
4.8.1	Simulazioni	71
5	CONCLUSIONI	75

A	GUIDA A WAYPOINTS: SCHEMI A BLOCCHI E CODICE	77
A.1	Blocchi Simulink	77
A.2	Codice Matlab	80
B	VISIBILITÀ: SCHEMI A BLOCCHI E CODICE	87
B.1	Blocchi Simulink	87
B.2	Codice Matlab	95
	BIBLIOGRAFIA	111

## ELENCO DELLE FIGURE

Figura 1	Classificazione guide per processo	2
Figura 2	Classificazione guide per metodologia	3
Figura 3	Varie MF	6
Figura 4	Controllore fuzzy	7
Figura 5	Funzioni di uscita dell'inferenza fuzzy	9
Figura 6	Schema a blocchi path-planner	10
Figura 7	Anello di guida e controllo	11
Figura 8	Geometria del problema di guida	12
Figura 9	Percorso desiderato nel semipiano superiore	13
Figura 10	Percorso desiderato nel semipiano inferiore	13
Figura 11	Sistema fuzzy (FLC <sub>sup</sub> )	14
Figura 12	Sistema fuzzy (FLC <sub>inf</sub> )	15
Figura 13	Simulazione con 3 waypoints	16
Figura 14	Simulazione con 3 waypoints, vista 3D	16
Figura 15	Evoluzione errore dell'angolo di heading	17
Figura 16	Punto ( $e_X^w, e_Y^w$ ) in FLC <sub>sup</sub>	18
Figura 17	Confronto delle traiettorie con un singolo waypoint	19
Figura 18	Traiettorie con 3 waypoints	20
Figura 19	Errore angolo di heading	20
Figura 20	Schema di guida completo	22
Figura 21	Esempio di guida LD e SD	22
Figura 22	Autopilota ICARO II con GPS e IMU a 6-DOF	23
Figura 23	Diagramma a blocchi del simulatore completo HIL	24
Figura 24	Simulazione HIL con 4 waypoints	25
Figura 25	Evoluzione dell'angolo di beccheggio	25
Figura 26	Valori del TET	26
Figura 27	Valori di NP	26
Figura 28	Navigazione tra i palazzi, vista in pianta	27
Figura 29	Navigazione tra i palazzi, vista 3D	27
Figura 30	Intercettazione multipla, vista in pianta	28
Figura 31	Intercettazione multipla, vista 3D	28
Figura 32	Esempio di ambiente strutturato	30
Figura 33	Gaussiana bi-variata	31
Figura 34	Definizione livello di visibilità	33
Figura 35	Visibilità target fisso	33
Figura 36	Variazione di visibilità	34
Figura 37	Direzione suggerita	35
Figura 38	Catena di gradienti 1	36
Figura 39	Catena di gradienti 2	36
Figura 40	Esempio task di sorveglianza	38
Figura 41	Scenario PNG	39

Figura 42	Accelerazione con $\gamma$ e $\gamma_{vis}$	41	
Figura 43	Intercettazione 1	42	
Figura 44	Intercettazione 2	42	
Figura 45	Accelerazione e singoli contributi	43	
Figura 46	Singoli contributi in percentuale	43	
Figura 47	Livello di visibilità	44	
Figura 48	Traiettorie al variare di K	44	
Figura 49	Traiettorie, target manovrato	46	
Figura 50	Target manovrato	47	
Figura 51	Traiettorie, target manovrato 2	47	
Figura 52	Scenario Heading-off	49	
Figura 53	Zone per Heading-off	49	
Figura 54	Esempio Heading-off	50	
Figura 55	Intercettazione con Heading-off	51	
Figura 56	Target manovrato 2	51	
Figura 57	Visibilità per un target in (42, 73)	61	
Figura 58	Primo Clustering	62	
Figura 59	Secondo Clustering	62	
Figura 60	Intercettazione: 2 Intercettori vs. 1 Target	64	
Figura 61	Intercettazione: 2 Intercettori vs. 1 Target	64	
Figura 62	Curve di livello della visibilità	65	
Figura 63	Traiettorie di veicolo e target	67	
Figura 64	Livello di visibilità	68	
Figura 65	Accelerazioni veicolo	68	
Figura 66	Traiettorie di veicolo e target	69	
Figura 67	Livello di visibilità	70	
Figura 68	Accelerazioni veicolo	70	
Figura 69	Blending factor	71	
Figura 70	Traiettorie	72	
Figura 71	Accelerazione	72	
Figura 72	Traiettorie	73	
Figura 73	Livelli di visibilità	73	
Figura 74	Guida e Controllori	77	
Figura 75	Blocco Guidance	78	
Figura 76	Interno Blocco Guidance	79	
Figura 77	Simulatore intercettazione: veicolo e target manovrato	87	
Figura 78	Blocco guida veicolo	88	
Figura 79	Blocco visibility computing	89	
Figura 80	Blocco dinamica veicolo	90	
Figura 81	Blocco dinamica target	91	
Figura 82	Blocco guida, 2 veicoli inseguitori	92	
Figura 83	Blocco approach point	93	

Figura 84	Blocco guida, 2 target	94
-----------	------------------------	----

## ELENCO DELLE TABELLE

Tabella 1	Regole per il sistema fuzzy $FLC_{sup}$	15
Tabella 2	Regole per il sistema fuzzy $FLC_{inf}$	15



## SOMMARIO

La prima fase del lavoro tratta dello sviluppo, dell'implementazione e del test pratico di un sistema di guida basato su waypoints per veicoli aerei senza pilota. In una seconda parte viene invece fatto uno studio teorico e sperimentale riguardante la questione della visibilità di un target in un ambiente complesso. Nella prima parte il sistema di guida viene dapprima progettato come un sistema fuzzy, basato su due controllori di Takagi-Sugeno, in seguito vengono sostituiti da due mappe discrete, in quanto il sistema di guida iniziale viene adattato ai requisiti imposti dall'hardware utilizzato nelle fasi di testing. Vengono riportati i risultati di una simulazione con Hardware in the Loop e presentato un test di volo, in cui tutti i concetti sviluppati sono testati in uno scenario reale. Nella seconda parte, l'obiettivo è il calcolo di una strategia di moto che mantenga alto il livello di visibilità di un target in un ambiente complesso. Il target viene considerato dapprima a velocità costante e poi manovrato attraverso un sistema di guida. Questo trasformerà il problema in un cosiddetto gioco differenziale che viene risolto. Tra le applicazioni viene descritto brevemente un task di sorveglianza che utilizza la guida a waypoint descritta nella prima parte. Viene invece ampiamente trattato un task di intercettazione basato sulla legge di guida PNG, opportunamente modificata.

## ABSTRACT

The first part of this work presents the development, implementation and testing of a guidance scheme based on waypoints for unmanned aerial vehicles. In a second part is made a theoretical and experimental study on the question of the visibility of a target in a complex environment. In the first part the guidance system is first designed as a fuzzy system, based on two Takagi-Sugeno controllers, subsequently these are replaced by two discrete maps, in fact the initial guidance system is adapted to the requirements imposed by the autopilot hardware used in the test phase. The results of a simulation with Hardware in the Loop are reported, and a flight test is described, in which all the concepts presented are tested in a real scenario. In the second part, the purpose is to calculate a motion strategy which maintains high visibility of a target in a complex environment. The target is considered with constant velocity and subsequently maneuvered with a guidance system. This will transform the problem into a differential game that is solved. This result leads to different applications, among which is briefly described a possible surveillance-task, that uses the waypoint guidance described in the first part. Instead, is widely described an interception-task based on the PNG law, which is suitably modified.



Questo lavoro è costituito da due parti che inizialmente possono apparire molto differenti tra loro nei contenuti e nelle modalità con cui le tematiche vengono trattate, in realtà tali parti vengono poi ad incontrarsi in diversi punti che riguardano principalmente l'applicazione concreta degli argomenti sviluppati. Una prima fase del presente lavoro tratta dello sviluppo, dell'implementazione e del test pratico di un sistema di guida basato su waypoints per veicoli aerei senza pilota. In una seconda parte viene invece fatto uno studio teorico e sperimentale riguardante la questione della visibilità di un target mobile in un ambiente complesso, ovvero un ambiente caratterizzato da ostacoli fissi di forma poligonale e di posizione nota, che, tuttavia, ai fini dello studio si considera solo nel caso planare.

Un sistema di guida è un dispositivo (o un insieme di dispositivi) utilizzato per la navigazione di una nave, di un aereo, di un missile, di un razzo, di un satellite, o di altri mezzi. Di solito, si tratta di un sistema che naviga senza il supporto continuativo di assistenza umana. I sistemi che necessitano di interazione umana sono invece detti sistemi di navigazione. Uno dei primi esempi effettivi di sistema di guida è stato quello utilizzato nelle bombe V-1 tedesche nel corso della seconda guerra mondiale. Tale sistema consisteva di un semplice giroscopio per mantenere la direzione, di un sensore di velocità per valutare il tempo di volo, di un altimetro per mantenere l'altitudine, e di altri sistemi ridondanti. Un sistema di guida può essere suddiviso in tre sezioni principali, l'acquisizione dei dati, l'elaborazione, e l'output. La sezione di acquisizione include i sensori, la rotta, i collegamenti a dati, via radio o via satellite, e altre sorgenti di informazione. La sezione di elaborazione, formata da una o più unità di elaborazione, integra tali dati e determina quali azioni, se occorrono, sono necessarie per mantenere o per raggiungere una data rotta o puntamento. La risposta è poi inviata alla sezione di output che a sua volta va a controllare la velocità interagendo con dispositivi propri del mezzo come le turbine, le pompe di carburante, i flap o i timoni del mezzo da guidare. La guida è in sostanza quell'azione o quel sistema che continuamente calcola il riferimento di posizione, velocità e accelerazione del veicolo da fornire al sistema di controllo. Essa può essere implementata come un sistema a ciclo chiuso o aperto, ed ha come scopo quello di implementare una legge che mantenga uno specifico percorso e/o raggiunga uno o più target. Negli anni recenti, le guide sono strettamente associate a problemi di intercettazione o di rendezvous e gli strumenti che queste utilizzano derivano dalla teoria dei controlli. Ci sono vari criteri per classificare un sistema di guida:

- A seconda delle fasi del moto
- In termini di processo di guida

- In termini di metodologia

Il primo criterio è utile quando il moto del veicolo è suddiviso in più fasi e ciascuna di queste rappresenta una missione completamente diversa dalle altre, con una strategia di controllo che ha obiettivi completamente diversi da quelli delle altre fasi. Il secondo si basa principalmente sulle modalità di acquisizione dei dati utili al calcolo della strategia di guida, in Figura 1 è mostrato uno schema che illustra tale criterio.

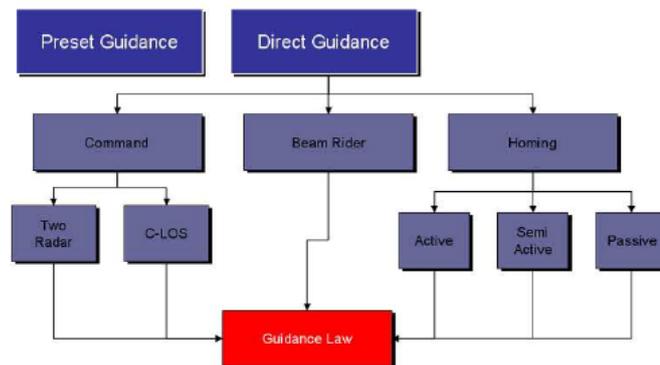


Figura 1.: Classificazione guide per processo

In particolare una Preset-Guidance è una guida in cui tutte le informazioni necessarie al processo di guida sono salvate sul veicolo prima del lancio, quindi non vi è un aggiornamento sullo stato del target; in una Direct-Guidance invece le informazioni riguardanti il target vengono inviate al lancio e aggiornate durante il volo. Di questa ultima tipologia fanno parte le guide di tipo Command, Beam-Rider e Homing. Le guide Command si basano principalmente sulle misure di distanza relativa e di velocità relativa tra veicolo e target, queste informazioni non vengono calcolate dal veicolo stesso ma esso le riceve dall'esterno in vari modi, di conseguenza il veicolo non necessita di alcun seeker a bordo. Nella guida Beam-Rider un fascio, generalmente laser, illumina costantemente il target e l'algoritmo di guida si occupa semplicemente di mantenere il veicolo sul fascio. Le guide Homing invece necessitano di un seeker, il veicolo ha un trasmettitore/ricevitore ed illumina il target ricevendo così informazioni di posizione e velocità direttamente da esso (Active e Semi-active Homing) oppure sfrutta delle proprietà intrinseche del target che emana esso stesso delle radiazioni, come calore o onde elettromagnetiche (Passive Homing). Questo criterio di classificazione è sicuramente il più diffuso, il terzo criterio è invece illustrato in Figura 2.

Lo schema riportato in figura fa una classificazione basata sull'algoritmo di guida. Le guide LOS-angle, LOS-rate e Advanced, per quanto simili, tutte necessitano di una conoscenza della Line-of-Sight (LOS), si differenziano rispetto a particolari aspetti dell'intercettazione: capacità di ottenere intercettazione in presenza di un target manovrato e/o con velocità superiore a quella del veicolo. L'obiettivo principale di queste guide è comunque quello di fornire un riferimento di accelerazione che garantisca l'intercettazione.

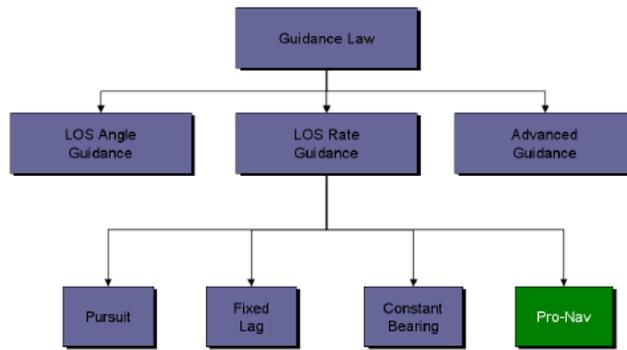


Figura 2.: Classificazione guide per metodologia

Il sistema di guida, trattato nella prima parte di questo lavoro, è basato sulla logica fuzzy e sui sistemi fuzzy. In particolare esso viene dapprima progettato come un sistema fuzzy, poi viene semplificato e adattato ai requisiti imposti dall'hardware utilizzato sia nelle fasi simulative che in quelle di testing. Esso suppone che il veicolo conosca la sua posizione attuale e la posizione dei waypoints. I waypoints sono descritti in uno spazio 4-dimensionale, la posizione in due dimensioni (come latitudine e longitudine), l'altitudine e l'angolo di attraversamento desiderato. Il sistema di guida fuzzy iniziale è basato su tre controllori fuzzy del tipo Takagi-Sugeno [17] che forniscono il riferimento dell'angolo di heading (l'imbardata in termini di angolo di rotazione) per l'autopilota, successivamente questo viene semplificato e vengono create due mappe discrete che lo sostituiscono. Questo nuovo sistema di guida risultante viene confrontato col precedente fuzzy e viene inoltre testato sia con simulazioni classiche, in cui sia il sistema di guida che la dinamica del veicolo vengono simulate, che con simulazioni hardware-in-the-loop (HIL), in cui resta simulata la dinamica del veicolo mentre l'algoritmo di controllo viene eseguito direttamente sull'hardware che verrà poi utilizzato in volo. Per concludere questa prima parte, vengono riportati i risultati di un test di volo, in cui tutti i concetti descritti vengono testati in uno scenario reale. La prima parte di questo lavoro (descritta nel Capitolo 2) ha quindi una caratterizzazione molto pratica, infatti gli algoritmi proposti vengono testati con simulazioni HIL, che prevedono l'uso di hardware connesso al simulatore, e infine confermati da un reale test di volo, operato con un veicolo quad-rotor. La seconda parte (descritta nei Capitoli 3 e 4) riguardante lo studio di visibilità, rimane invece prettamente simulativa, in quanto i concetti e gli algoritmi sviluppati vengono dapprima giustificati a livello teorico e in seguito vengono testati con diverse simulazioni.

Nella seconda parte viene intrapreso uno studio della visibilità di un target in un ambiente complesso, strutturato. Il problema che ci si pone è il calcolo di una strategia di moto che mantenga alto il livello di visibilità di un target, fisso o mobile, nel suddetto ambiente. Il target viene prima considerato come un oggetto fisso, in seguito avente moto rettilineo uniforme ed infine manovrato, con un'accelerazione opportunamente fornitagli da un sistema di guida. Tale accelerazione trasformerà il problema in un cosiddetto gioco differenziale in cui veicolo e target vanno a ottimizzare uno stesso indice di costo. Tutto ciò non

può prescindere dalla definizione di un livello di visibilità, che viene data e che, a differenza di quanto è solito trovare in letteratura, è basata sulla definizione di una regione di incertezza sulla posizione del target. Infatti la definizione di visibilità che è solito trovare in letteratura ha una caratterizzazione booleana, legata alla presenza o meno di una line-of-sight (LOS) tra il target e il veicolo, associando invece una incertezza alla posizione del target si ottiene una informazione di visibilità sfumata e più ricca. Viene quindi formulato un algoritmo che fornisce una direzione di riferimento che comporta un miglioramento del livello di visibilità del target. Questo studio apre la strada a diverse applicazioni, tra le principali vi è il task di osservazione, ampiamente discusso in letteratura, in cui il veicolo deve mantenere la visibilità del target portandosi, tuttavia, ad una distanza non inferiore ad una certa soglia. Questa applicazione è trattata brevemente, in quanto può essere ricondotta semplicemente al posizionamento opportuno di alcuni waypoints e all'utilizzo di una legge di guida che conduca il veicolo su di essi, infatti una volta individuate le zone ad alta visibilità e posizionati i waypoints i veicoli possono raggiungerli con semplici strategie di guida: una delle quali potrebbe essere la guida a waypoints fuzzy descritta in precedenza, o una semplice guida LOS. Un'altra possibile applicazione ampiamente discussa in questo lavoro, è rappresentata da un task di intercettazione. Viene elaborata una strategia di moto per il veicolo inseguitore che massimizzi la probabilità che il target si trovi in vista, in base ad una conoscenza probabilistica della posizione del target, che può essere interpretata anche come una predizione, entro un certo intervallo temporale, della sua posizione futura. La strategia applicata è dapprima risolta numericamente, vengono quindi riportate delle simulazioni in cui è presente un solo target e un solo veicolo inseguitore, vengono effettuate anche delle simulazioni in cui si hanno molteplici veicoli inseguitori, e delle simulazioni in cui un solo inseguitore deve intercettare più target. L'intercettazione viene implementata ispirandosi alla più classica delle guide per questi tipi di task, la Proportional-Navigation-Guidance (PNG), andando tuttavia a modificare questa per tenere in considerazione anche la visibilità del target. Inoltre viene data una impostazione teorica a tale applicazione, considerando l'intercettazione come un gioco differenziale tra Pursuer-Evader, viene quindi ricavato un funzionale di costo e viene risolto matematicamente il problema di ottimo.

Sebbene le due parti possano apparire disgiunte, esse presentano in realtà alcuni aspetti in comune. Infatti una applicazione dello studio di visibilità potrebbe consistere nell'utilizzo della guida fuzzy in cui un Waypoints Manager sceglierebbe posizione e angolo di attraversamento dei waypoint tenendo in considerazione quelle che sono le zone ad alta visibilità prodotte dall'algoritmo. Lo studio effettuato nella seconda parte è infatti comunque orientato alla produzione di una legge di guida per veicoli autonomi, e questo non può che essere compatibile con una guida a waypoint che fornisce e dettaglia un algoritmo che consente al veicolo di arrivare sul waypoint, ma che non si occupa di fornire dei criteri di scelta per la posizione e l'orientazione di tali waypoints.

# 2

## SISTEMA DI GUIDA A WAYPOINTS CON APPROCCIO FUZZY

### 2.1 LA LOGICA E I SISTEMI FUZZY

La logica fuzzy o logica sfumata o logica sfocata è una logica in cui si può attribuire a ciascuna proposizione un grado di verità compreso tra 0 e 1. È una logica polivalente, e pertanto un'estensione della logica booleana. È fortemente legata alla teoria degli insiemi sfocati e, già intuita da Cartesio, Bertrand Russell, Albert Einstein, Werner Karl Heisenberg, Jan Łukasiewicz e Max Black, venne concretizzata da Lotfi Zadeh. Con grado di verità o valore di appartenenza si intende quanto è vera una proprietà: questa può essere, oltre che vera (valore 1) o falsa (valore 0) come nella logica classica, anche pari a valori intermedi. Formalmente, questo grado di appartenenza è determinato da un'opportuna funzione di appartenenza  $\mu_F(x) = \mu$ , in cui  $x$  rappresenta dei predicati da valutare e appartenenti ad un insieme di predicati  $X$  e  $\mu$  rappresenta il grado di appartenenza del predicato all'insieme fuzzy considerato e consiste in un numero reale compreso tra 0 e 1. La teoria degli insiemi fuzzy costituisce un'estensione della teoria classica degli insiemi poiché per essa non valgono i principi aristotelici di non-contraddizione e del terzo escluso (detto anche "tertium non datur"). Si ricorda che, dati due insiemi  $A$  e  $\neg A$  (non- $A$ ), il principio di non-contraddizione stabilisce che ogni elemento appartenente all'insieme  $A$  non può contemporaneamente appartenere anche a non- $A$ ; secondo il principio del terzo escluso, d'altro canto, l'unione di un insieme  $A$  e del suo complemento non- $A$  costituisce l'universo del discorso. In altri termini, se un qualunque elemento non appartiene all'insieme  $A$ , esso necessariamente deve appartenere al suo complemento non- $A$ . Un insieme sfocato è caratterizzato da una funzione di grado di appartenenza, che mappa gli elementi di un universo in un intervallo reale continuo  $[0, 1]$ . Il valore 0 (zero) indica che l'elemento non è per niente incluso nell'insieme sfocato, il valore 1 (uno) indica che l'elemento è certamente incluso nell'insieme (questi due valori corrispondono alla teoria classica degli insiemi), mentre i valori tra zero e uno indicano il grado di appartenenza dell'elemento all'insieme sfocato in questione. Per un universo  $X$  e una data funzione del grado di appartenenza  $f : X \rightarrow [0, 1]$ , l'insieme sfocato  $A$  è definito come  $A = (x, f(x))_{x \in X}$ . Sugli insiemi fuzzy valgono gli operatori insiemistici: unione, intersezione e complementare. Vi sono vari modi possibili di generalizzare gli operatori della logica classica. L'operazione di unione su due insiemi fuzzy  $A$  e  $B$  si esegue applicando ad ogni elemento  $x$  di  $A$  e  $y$  di  $B$  una funzione chiamata s-norm; tipicamente si prende il massimo tra i due valori:

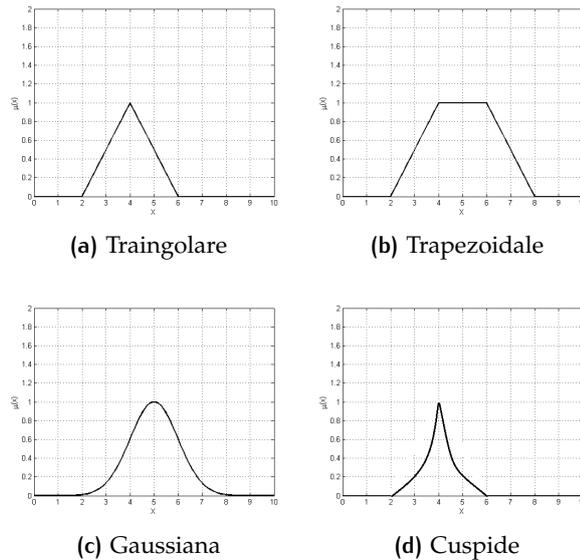


Figura 3.: Varie MF

$$\begin{aligned}
 A &= \left\{ \frac{0.2}{1} + \frac{0.3}{2} + \frac{0.6}{3} + \frac{0.3}{4} + \frac{0.4}{5} \right\} \\
 B &= \left\{ \frac{1}{1} + \frac{0.1}{2} + \frac{0.7}{3} + \frac{0.6}{4} + \frac{0.1}{5} \right\} \\
 A \cup B &= \left\{ \frac{1}{1} + \frac{0.3}{2} + \frac{0.7}{3} + \frac{0.6}{4} + \frac{0.4}{5} \right\}
 \end{aligned}$$

L'operazione di intersezione invece viene effettuata utilizzando funzioni t-norm, tipicamente la funzione minimo; per definire invece il complementare di un insieme, si calcola il nuovo grado di appartenenza di un elemento al nuovo insieme B come 1 meno il grado di appartenenza ad A, come nel seguente esempio:

$$\begin{aligned}
 A &= \left\{ \frac{0.4}{1} + \frac{0.6}{2} + \frac{0}{8} \right\} \\
 A^C &= \left\{ \frac{0.6}{1} + \frac{0.4}{2} + \frac{1}{8} \right\}
 \end{aligned}$$

Mentre un insieme classico possiede una funzione di appartenenza booleana, un insieme fuzzy ne ha una continua con valori compresi tra 0 e 1. Tale funzione esprime perciò il grado di appartenenza di una variabile reale ad un certo insieme fuzzy. La forma di queste funzioni non è fissa ma ne possono essere scelte di diversi tipi, le più comuni sono riportate in Figura 3.

Un controllore fuzzy è un sistema fuzzy che opera su variabili linguistiche. Per variabile linguistica si intende una variabile i cui valori sono parole e frasi

di un linguaggio naturale o artificiale. Per esempio "età" è una variabile linguistica se i suoi valori sono linguistici, ovvero "giovane", "non giovane", "molto giovane", "vecchio", "molto vecchio", oppure è una variabile numerica se i suoi valori sono numerici, ovvero 20, 21, 22 ecc. La dualità di queste descrizioni della stessa variabile è analoga alla dualità degli insiemi classici e fuzzy. Una descrizione numerica può essere tradotta in una linguistica (fuzzificazione) con l'uso delle funzioni di appartenenza. Una variabile linguistica è solitamente descritta da un certo numero di insiemi fuzzy. Quelli più comunemente usati sono:

- Negativo grande (NL)
- Negativo medio (NM)
- Negativo piccolo (NS)
- Circa zero (ZE)
- Positivo piccolo (PS)
- Positivo medio (PM)
- Positivo grande (PL)

Il controllore fuzzy vero e proprio opera su variabili linguistiche mediante una base di regole scritte dal progettista, ed estrapolate dalla cosiddetta base di esperienza. Risulta importante che il numero di regole sia un buon compromesso tra prestazioni e complessità. Tale controllore è inserito in sistemi reali, perciò questo viene completato da stadi di conversione verso il mondo fuzzy e poi dal ritorno a valori numerici (defuzzificazione). Le regole sono scritte in linguaggio naturale e sfruttano le operazioni elementari sugli insiemi fuzzy. In Figura 4 è rappresentato lo schema tipico di un controllore fuzzy.

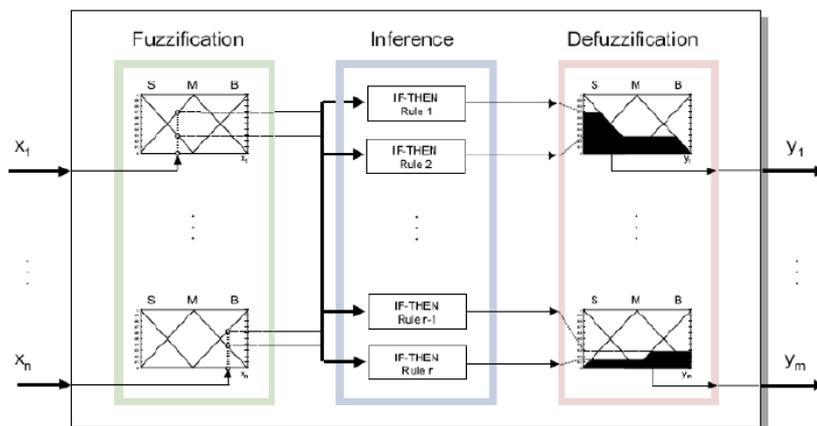


Figura 4.: Controllore fuzzy

La fuzzificazione è di fatto la quantizzazione di un segnale che però viene descritto con insiemi fuzzy anziché con livelli logici. Come già riportato i pre-

cedenza, le variabili numeriche vengono descritte in forma linguistica aggiornando ciascuno degli insiemi fuzzy scelti con il valore della propria funzione di appartenenza in corrispondenza del valore attuale della variabile numerica. Spesso la funzione di appartenenza ha un valore diverso da zero solo al più per due insiemi fuzzy, anche se questo dipende più in generale da come sono state definite le funzioni di appartenenza. Per quanto riguarda la fase di inferenza, questa è detta anche di attivazione delle regole. Le regole fuzzy sono rappresentazioni di inferenze logiche fatte su composizioni di proposizioni fuzzy. Il processo inferenziale è composto da una lista di regole e dall'operazione di somma logica: le regole possono essere suddivise in condizioni (che formano il blocco antecedente) e in conclusioni (il valore di verità di un blocco conseguente). La descrizione di un sistema con la logica fuzzy viene realizzato attraverso una serie di implicazioni fuzzy che nel loro insieme costituiscono un algoritmo fuzzy. Se  $X_i$  è una variabile linguistica di ingresso che assume come valore l'insieme fuzzy etichettato con  $LX_i$  e  $Y$  è una variabile linguistica di uscita che assume come valore l'insieme fuzzy etichettato con  $LY$ , allora una generica regola ha la forma seguente.

$$\text{Se } (X_1 \text{ è } LX_1) \text{ AND/OR } (X_2 \text{ è } LX_2) \text{ AND/OR } \dots \text{ Allora } (Y \text{ è } LY)$$

In pratica accade che se ogni condizione dell'antecedente sarà vera con un certo grado di verità; in base alle definizioni delle operazioni elementari presenti verrà assegnato un certo valore di attivazione di ogni regola. Nell'approccio fuzzy classico questo valore corrisponderà al valore di appartenenza della variabile di uscita  $Y$  all'insieme fuzzy  $LY$ . Questo tipo di regole, in altri termini, permette di definire in forma linguistica le condizioni in cui applicare un modello. L'utilizzo di queste regole non solo ha introdotto nuove possibilità di modellazione, ma ha anche permesso di provare delle proprietà importanti per i sistemi fuzzy, quali, ad esempio, la stabilità del controllore. In particolare questo tipo di regole sono spesso note come regole di inferenza di Mamdani e sono sicuramente le più utilizzate ma non le uniche. Un altro tipo di inferenza è detta di Takagi-Sugeno, introdotta nel 1985, e per molti aspetti è molto simile alla precedente. Infatti le prime due parti, la fuzzificazione e l'applicazione dell'operatore fuzzy sono esattamente identiche e la principale differenza è che, in Takagi-Sugeno, le funzioni di appartenenza dell'output possono essere lineari o costanti. Una regola tipica di Takagi-Sugeno ha la forma seguente.

$$\text{Se } (X_1 \text{ è } LX_1) \text{ AND/OR } (X_2 \text{ è } LX_2) \text{ AND/OR } \dots \text{ Allora } (Y = ax_1 + bx_2 + \dots + c)$$

Nel caso di un modello di ordine zero l'uscita è costante ( $a = b = 0$ ).

La fase di defuzzificazione è la fase successiva con cui si convertono i risultati ricavati dal processo di inferenza in un valore univoco. L'uscita dell'algoritmo fuzzy è ancora un insieme fuzzy, dal quale deve essere estratto un numero reale che deve rappresentare il contenuto informativo dell'algoritmo stesso. In questa fase le variabili linguistiche di uscita, e gli insiemi fuzzy ad esse associati, vengono interpretati per ricavare un valore numerico da assegnare alle variabili di uscita reali. E' possibile che le variabili di uscita abbiano un valore di appartenenza non nullo per diversi insiemi fuzzy, come può accadere alle variabili

di ingresso. Questo perché regole diverse possono agire sulla stessa variabile linguistica di uscita. Se ad esempio una certa variabile linguistica di uscita valesse ZE al 25% a causa della regola 1 e NS al 75% a causa della regola 2 e si avessero funzioni di appartenenza triangolari la situazione potrebbe essere rappresentata come in Figura 5.

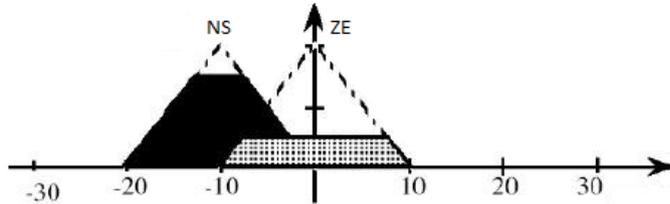


Figura 5.: Funzioni di uscita dell'inferenza fuzzy

Esistono molti metodi per risolvere questa ambiguità. I più comuni sono:

- Metodo del centroide: si assegna alla variabile numerica il valore in ascissa del baricentro delle aree evidenziate in figura.
- Baricentro dell'area massima: si assegna alla variabile numerica il valore in ascissa del baricentro dell'area massima tra quelle evidenziate in figura.
- Media dei massimi.
- Primo, ultimo o metà dei massimi.

In genere quello più utilizzato è il metodo del centroide, che ha il vantaggio di risultare abbastanza semplice nel caso in cui le funzioni di appartenenza dei conseguenti siano simmetriche. In tal caso infatti il baricentro di ciascun conseguente è indipendente dal grado di attivazione del relativo antecedente. In definitiva l'uscita può allora essere scritta come:

$$y = \frac{\int u_i \cdot \mu(u_i) du_i}{\int \mu(u_i) du_i} \quad (1)$$

Questa sequenza di operazioni, che descrivono per intero il comportamento di un sistema fuzzy, verrà utilizzata implicitamente nei controllori fuzzy trattati a seguire.

## 2.2 INTRODUZIONE ALLA GUIDA FUZZY

Varie leggi di guida che utilizzano un approccio fuzzy sono state proposte in letteratura, la maggior parte delle quali basate sul principio di navigazione proporzionale. In uno dei primi approcci fuzzy per la guida di un veicolo, un sistema fuzzy viene utilizzato per mettere insieme tre diverse leggi di guida:

PN, guida bang-bang e PNG aumentata, agendo basandosi su misure della distanza e della sua variazione [4]. Un altro approccio fuzzy utilizza uno schema di guida basato su un classico PID [11], in cui il PID utilizza l'errore di heading e l'errore di quota per calcolare i comandi di velocità angolari e la logica fuzzy è usata per generare e combinare le uscite delle parti PD e PI del controllore. Un approccio fuzzy è stato, inoltre, usato in una guida "commands to line-of sight" (CLOS) [5], in cui dapprima vengono presentati dei sistemi di controllo fuzzy e in seguito viene sviluppato un metodo per il design di un controllore adattativo sliding-mode di tipo fuzzy e applicato al design della legge di guida CLOS.

La progettazione di una traiettoria per un UAV deve tener conto dei suoi vincoli cinematici, il più importante dei quali è la curvatura massima del percorso sopportabile dal veicolo, che coincide col il minimo raggio di sterzata del veicolo stesso. Tale vincolo, se rispettato, rendono il percorso "flyable". Vi è un'ampia varietà di pianificatori di traiettoria (planners) per tali percorsi disponibili in letteratura. Le tecniche più utilizzate per calcolare il percorso tra una posa iniziale e una finale sono: il Dubins-path che ha archi di curvatura costanti, Clothoid-path che è simile al Dubins ma con archi di curvatura variabili e il Pythagorean-hodograph (PH) path [2]. Il Dubins-path è il percorso più breve tra due punti per cui è fissata l'orientazione iniziale e finale, col minimo raggio di sterzata, ma presenta delle discontinuità nella curvatura. I restanti due path-planners producono dei percorsi che sono continui nella curvatura ma sono computazionalmente più complessi. Nel primo caso, gli archi costanti vengono sostituiti con i "clothoid arcs", che forniscono una transizione graduale tra il percorso lineare e circolare. Infine, la curva PH è utilizzata soprattutto per le sue proprietà matematiche, infatti la curva è parametrizzata da polinomi in funzione della lunghezza del percorso ed è disegnata come un'unica curva invece che come un insieme di singole curve in sequenza.

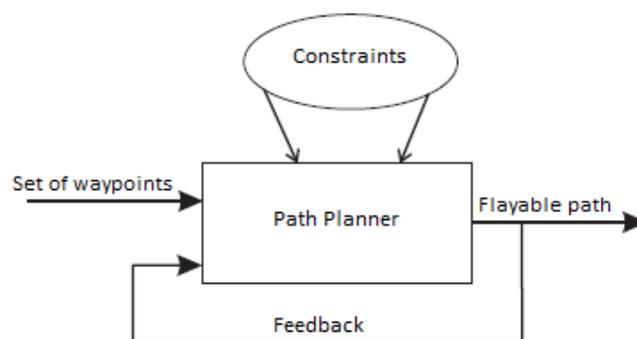


Figura 6.: Schema a blocchi path-planner

Questa sezione descrive lo sviluppo, l'implementazione e il collaudo di un sistema di guida basato sui sistemi fuzzy, che ha la caratteristica peculiare di poter disegnare il percorso di avvicinamento del veicolo verso il waypoint, e non richiede nessun altro path-planner tra i waypoints. L'approccio proposto

in questa sezione combina la pianificazione del percorso e le fasi di guida in uno, nel rispetto dei vincoli cinematici desiderati, che sono stati esplicitamente considerati durante la progettazione del sistema. Il sistema di guida fuzzy proposto (FGS) suppone che il veicolo conosca la sua posizione attuale e la posizione dei waypoints. I waypoints sono descritti in uno spazio 4-dimensionale, la posizione in due dimensioni (come latitudine e longitudine), l'altitudine e l'angolo di attraversamento desiderato. La FGS si basa su tre sistemi fuzzy di Takagi-Sugeno (TS) [17], che generano il riferimento dell'angolo di heading (l'imbardata, in termini di angoli di rotazione). L'uso di un sistema fuzzy deriva principalmente dalla necessità di specificare la direzione attraversamento sul waypoint e il percorso di approccio nelle zone circostanti. Gli obiettivi di progettazione sono, almeno in parte, gli stessi discussi in [10]:

- Possibilità di specificare l'angolo di attraversamento desiderato del waypoint
- Gestione di una sequenza di waypoints, che può richiedere manovre complesse a causa delle limitazioni del veicolo.

### 2.3 DESIGN DEL SISTEMA DI GUIDA FUZZY

Il sistema di guida proposto genera un percorso di riferimento in termini di angolo di heading desiderato; si assume che è disponibile un autopilota in grado di inseguire il riferimento di angolo fornito, come mostra lo schema riportato in Figura 7.

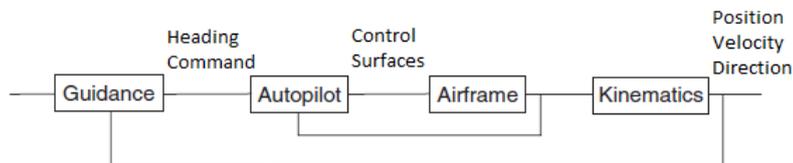


Figura 7.: Anello di guida e controllo

La traiettoria desiderata è specificata in termini di una sequenza di waypoints, senza alcun vincolo geometrico sul percorso tra due waypoints successivi. La posizione di un waypoint è specificata in coordinate cartesiane 3D  $(X_w, Y_w, H_w)$  nel sistema di riferimento Earth-fixed, e l'angolo di heading fornito  $(\chi_w)$  viene utilizzato per ottenere la direzione di approccio desiderata. Ciascun waypoint definisce un sistema di coordinate centrato nella posizione del waypoint stesso e ruotato dell'angolo  $\chi_w + (\pi/2)$  intorno all'asse Z rispetto al sistema di riferimento fisso, come mostrato in Figura 8. Come è solito fare in questi casi, un gestore di waypoint controlla la posizione del veicolo e aggiorna il waypoint desiderato quando quello precedente è stato raggiunto, considerando una certa tolleranza. Ai fini del nostro studio si considerano solo gli spostamenti sul

piano orizzontale, quindi un autopilota in grado di inseguire una altitudine di riferimento (cioè  $H_w$ ) è assunto a disposizione.

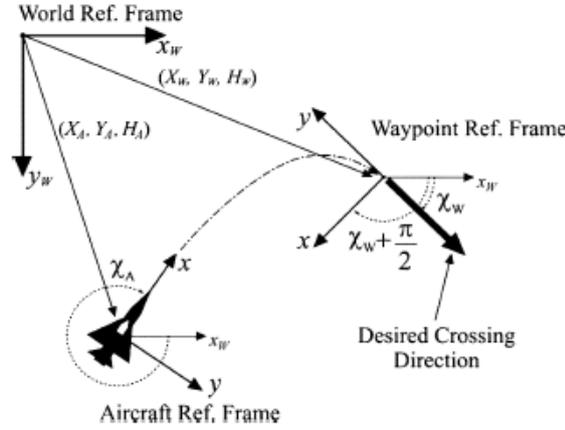


Figura 8.: Geometria del problema di guida

L'approccio desiderato al waypoint è ottenuto attraverso il controllore fuzzy  $f_\chi$  che genera l'angolo di heading desiderato  $\chi_d$  usando gli errori di posizione lungo gli assi X e Y del sistema di riferimento del waypoint corrente  $[e_X^w, e_Y^w]$ .

$$[e_X^w, e_Y^w]^T = R_z(\chi_w + (\pi/2)) \cdot [e_X, e_Y]^T \quad (2)$$

Dove  $e_X = X_w - X$  e  $e_Y = Y_w - Y$ , con  $(X, Y)$  posizione del veicolo in coordinate inerziali nel sistema di riferimento locale. Il sistema fuzzy  $f_\chi$  è stato progettato con il supporto di un fattore di scala  $S$  che lo rende indipendente dalla natura specifica del veicolo (terra, aria, veloce, lento, con ali fisse/girevoli, ecc). Il coefficiente  $S$  ridimensiona gli errori di posizione, quindi ridimensiona (ingrandendo o riducendo) le traiettorie desiderate del veicolo, aumentando o diminuendo il raggio di curvatura; questo coefficiente rappresenta di fatto un ulteriore grado di libertà nel processo di generazione della traiettoria.

$$\chi_d = f_\chi(e_X^w, e_Y^w, S) \quad (3)$$

Il controllore dell'angolo di heading  $f_\chi$  è costituito da due parti: un controllore fuzzy  $\delta_{\chi_w}$ , che genera l'angolo di heading desiderato nel piano orizzontale nel sistema di riferimento del waypoint, e un segnale feed-forward costituito dall'angolo di attraversamento desiderato per il waypoint corrente  $\chi_w$ .

$$\chi_d = \chi_w + \delta_{\chi_w} \quad (4)$$

Il sistema fuzzy genera il percorso desiderato espresso nel sistema di riferimento del waypoint. La sua progettazione dipende dalla traiettoria voluta, e

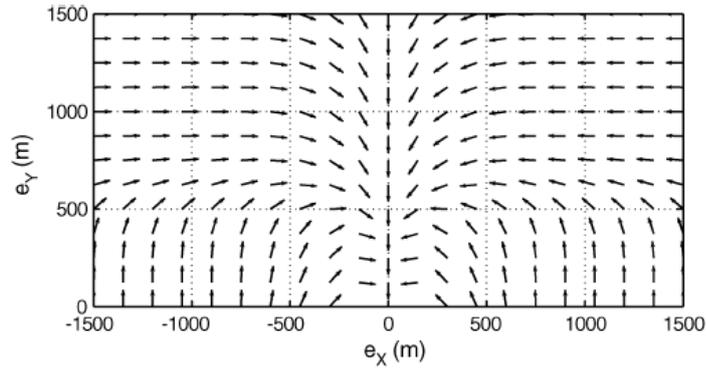


Figura 9.: Percorso desiderato nel semipiano superiore

tiene conto dei limiti del veicolo: la buona riuscita del sistema fuzzy si traduce in manovre morbide e in limitati segnali di controllo. Le traiettorie di riferimento nel piano orizzontale sono ottenute come interpolazione convessa degli angoli di heading in un numero finito di partizioni dell'intero piano. Questa interpolazione viene eseguita utilizzando un sistema fuzzy di Takagi-Sugeno, il sistema fuzzy conosce la posizione del veicolo nel sistema di riferimento del waypoint e genera l'angolo di heading desiderato per il veicolo in quel punto. L'autopilota del veicolo è responsabile della regolazione dell'angolo di imbardata al valore desiderato. Per evitare brusche transizioni nel percorso desiderato, soprattutto in prossimità dell'origine del sistema di riferimento del waypoint, la progettazione è stata suddivisa in due parti: la traiettoria nel semipiano superiore ( $e_y^w > 0$ ) e nel semipiano inferiore ( $e_y^w < 0$ ), ottenendo così due sistemi fuzzy:  $FLC_{sup}(e_x^w, e_y^w)$  e  $FLC_{inf}(e_x^w, e_y^w)$ , Figura 9 e Figura 10 mostrano le traiettorie desiderate risultanti definite in modo simile ad un campo vettoriale.

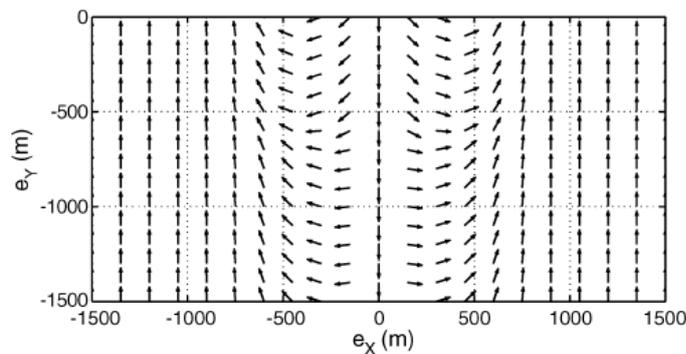


Figura 10.: Percorso desiderato nel semipiano inferiore

Per evitare singolarità, dovute al fatto che la misura dell'angolo di imbardata ha una rappresentazione discontinua per  $\pm\pi$ , l'uscita dei due controllori fuzzy deve essere corretta sommando  $\pm 2\pi$  a seconda dell'orientazione del veicolo, l'idea è quella di generare angoli di riferimento che non comportino un errore di angolo di heading superiore a  $\pm\pi$ . La mappa completa dei percorsi desiderati si ottiene unendo, attraverso un'apposita funzione, le traiettorie del semipiano

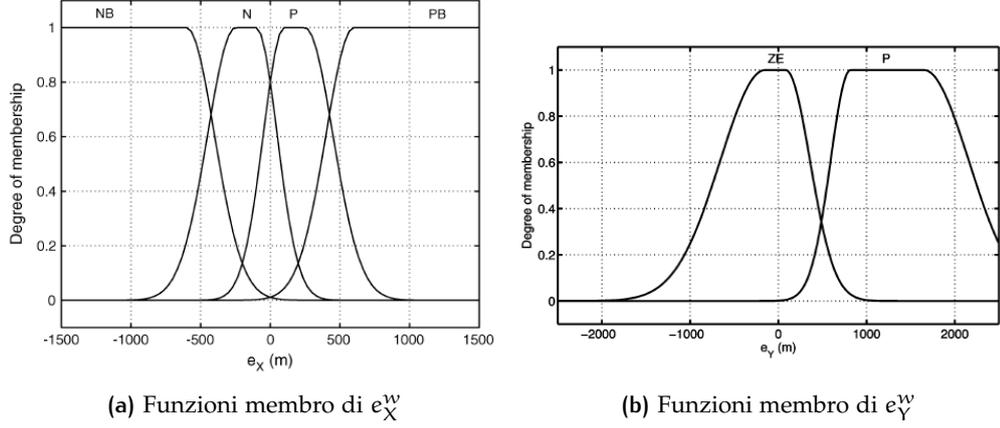


Figura 11.: Sistema fuzzy ( $FLC_{sup}$ )

superiore e inferiore, in tal modo le uscite dei due sistemi fuzzy sono fuse insieme dalla funzione di ponderazione  $\mu_Y(e_Y)$ , che dipende solo da  $e_Y$ :

$$\delta_{Xw} = \mu_Y \cdot [FLC_{sup}(S \cdot e_X^w, S \cdot e_Y^w) + c_{sup}] + (1 - \mu_Y) \cdot [FLC_{inf}(S \cdot e_X^w, S \cdot e_Y^w) + c_{inf}] \quad (5)$$

Con:

$$\mu_Y(e_Y^w) = \begin{cases} e^{-\frac{e_Y^w{}^2}{2\sigma^2}}, & e_Y^w < 0 \\ 1, & e_Y^w \geq 0 \end{cases} \quad (6)$$

Dove  $c_{sup} = \pm 2k\pi$  e  $c_{inf} = \pm 2h\pi$  ( $k, h = 0, \pm 1$ ) sono scelti per rispettare il vincolo angolare descritto prima. La funzione esponenziale  $\mu_Y(e_Y)$  definisce l'area in cui le uscite dei due sistemi fuzzy  $FLC_{sup}$  e  $FLC_{inf}$  sono fuse insieme, che si estende approssimativamente da  $e_Y^w = 0$  a  $e_Y^w = -3\sigma$ . Si noti che le aree in cui sono state definite le traiettorie di approccio desiderate sono larghe circa 3Km, e il raggio di curvatura minimo è di circa 250m, questi valori possono essere modificati facilmente, senza dover ri-progettare il sistema fuzzy utilizzando il fattore di scala  $S$ . Attraverso il fattore di scala  $S$ , il raggio di curvatura e la larghezza dell'area possono essere facilmente regolate per adattarsi ad eventuali caratteristiche del veicolo. Entrambi i sistemi fuzzy hanno due ingressi: l'errore di posizione ed una uscita: l'angolo di heading desiderato.

Il sistema fuzzy  $FLC_{sup}$  è interamente definito dalle otto regole descritte nella Tabella 1 [10] [6] (valori di output espressi in gradi); la Figura 11 mostra le funzioni membro: NB, N, P e PB che costituiscono gli insiemi fuzzy per  $e_X^w$  e le MF ZE e P per  $e_Y^w$ ; l'andamento dell'output risultante, ottenuto applicando il motore di inferenza di Takagi-Sugeno [17] alle funzioni membro con la tabella delle regole è riportato in Figura 9. Il sistema fuzzy  $FLC_{inf}$  è definito interamente dalle sette regole riportate nella Tabella 2 (valori di output espressi in

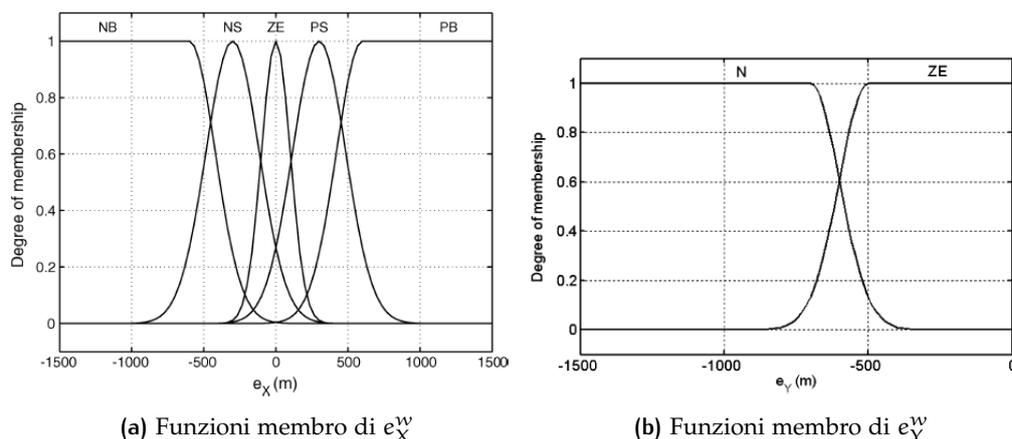


Figura 12.: Sistema fuzzy (FLC<sub>inf</sub>)

gradi); la Figura 12 mostra invece le funzioni membro: NB, NS, ZE, PS e PB che costituiscono gli insiemi fuzzy per  $e_x^w$  e le MF ZE e N per  $e_y^w$ ; l'andamento dell'output risultante è riportato in Figura 10.

Tabella 1.: Regole per il sistema fuzzy FLC<sub>sup</sub>

$e_Y/e_X$	NB	N	P	PB
ZE	90	45	-45	-90
P	18	13	-	-
	0	0	130	180

Tabella 2.: Regole per il sistema fuzzy FLC<sub>inf</sub>

$e_Y/e_X$	NB	NS	ZE	PS	PB
ZE	-180	-45	-20	45	180
N	-180	-90	*	90	180

L'asterisco (\*) nella Tabella 2 per la condizione  $e_x^w$  IS ZE AND  $e_y^w$  IS N, indica che non sono previste regole per questi valori degli ingressi fuzzy, ed il risultato viene ottenuto dall'interpolazione dei risultati delle regole vicine. In accordo con la descrizione dei sistemi fuzzy FLC fatta in precedenza, il valore del parametro della funzione peso  $\sigma$  viene scelto uguale a 100m.

La condizione patologica  $e_x(t) = 0 \forall t > t_0$ , con il veicolo che costantemente si allontana dal waypoint dopo un certo istante  $t_0$  rappresenta un equilibrio instabile dovuto alla scelta delle regole di inferenza; la Figura 10 mostra tuttavia che qualsiasi perturbazione nel sistema conduce ad una traiettoria di riferimento che, curvando in senso orario o antiorario, guida in veicolo verso il waypoint nuovamente.

Una simulazione è riportata in Figura 13 e in Figura 14, nel quale il veicolo procede percorrendo tre waypoints con differenti angoli di attraversamento. Tali waypoints hanno la stessa altitudine, e la velocità di avanzamento del veicolo è costante. L'angolo di attraversamento desiderato (in gradi) è di 135 per il primo waypoint (A), 270 per il secondo (B) e 90 per il terzo (C). La traiettoria

disegnata viene seguita e l'errore in valore assoluto tra l'angolo di heading comandato e quello reale è sempre minore di 3 gradi circa, come si evince dalla Figura 15.

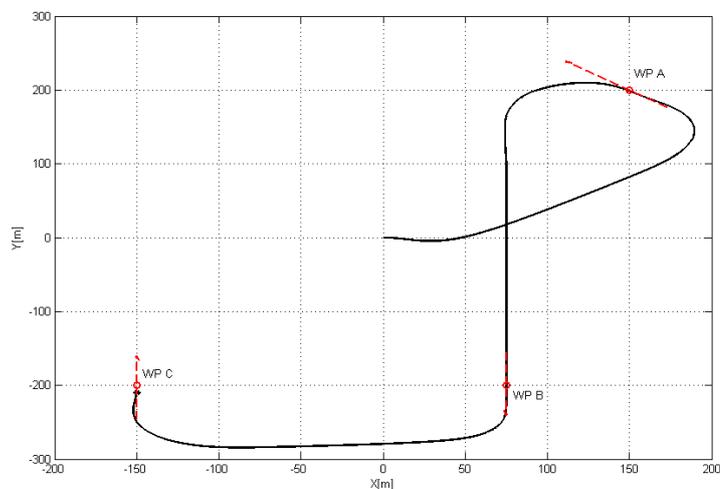


Figura 13.: Simulazione con 3 waypoints

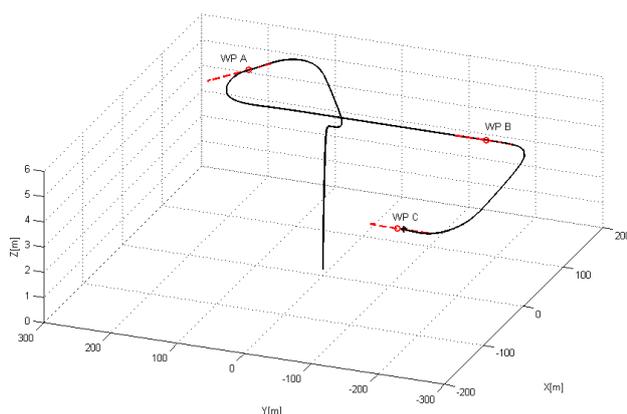


Figura 14.: Simulazione con 3 waypoints, vista 3D

## 2.4 IMPLEMENTAZIONE REAL-TIME

La simulazione presentata nel paragrafo precedente è stata fatta utilizzando dei sistemi fuzzy con motore di inferenza di Takagi-Sugeno [17]. Questo, per quanto sia matematicamente semplice, richiede un notevole numero di computazioni che possono non coincidere con le esigenze di una implementazione real-time su una piattaforma che ha ridotte risorse computazionali. Un sistema fuzzy di Takagi-Sugeno ha l'interessante proprietà di essere equivalente a una mappa ingresso/uscita che sia non-lineare e tempo-invariante. Tale proprietà è stata sfruttata in questo lavoro per ottenere una mappa ingresso/uscita discreta,

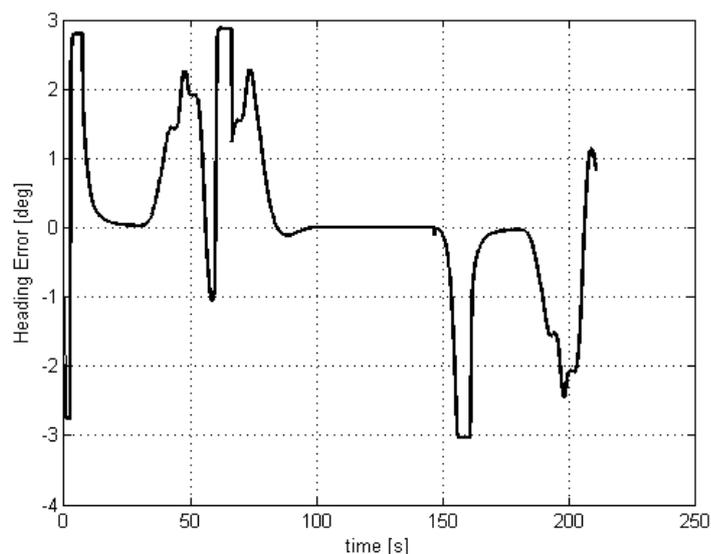


Figura 15.: Evoluzione errore dell'angolo di heading

che approssimi la caratteristica ingresso/uscita del nostro sistema fuzzy. Usando questo approccio, l'esecuzione del motore di inferenza fuzzy fatta fin'ora non è più necessaria.

#### 2.4.1 Semplificazione del Sistema Fuzzy

L'esigenza di avere un sistema di guida dal peso computazionale ridotto, ha portato ad una semplificazione del motore di inferenza fuzzy (FIS) che costituisce il controllore fuzzy del sistema di guida. La semplificazione è stata ottenuta attraverso una discretizzazione della caratteristica ingresso/uscita dei due controllori fuzzy ( $FLC_{sup}$  e  $FLC_{inf}$ ), che si occupano di generare l'angolo di heading di riferimento  $\chi_d$ . Entrambi i sistemi fuzzy utilizzano come ingressi l'errore di posizione lungo gli assi X e Y del sistema di riferimento del waypoint corrente  $[e_X^w, e_Y^w]$ . Le funzioni membro relative a  $e_X^w$  sono definite nell'intervallo  $[-3000 \ 3000]$  metri per entrambi i controllori fuzzy, quelle relative a  $e_Y^w$  sono definite nell'intervallo  $[-2500 \ 2500]$  metri per  $FLC_{sup}$  e in  $[-2000 \ 0]$  per  $FLC_{inf}$ . Quindi, si è operata una discretizzazione dei quattro intervalli, che contengono i valori di input dei due FIS, scegliendo una appropriata "distanza di campionamento"; questa scelta è stata fatta con l'intento di non tralasciare cambiamenti significativi del segnale di uscita (l'angolo di heading desiderato), come avviene nel tempo quando si campiona un segnale tempo-continuo.

Ranges for  $FLC_{sup}$

$$e_X^w \in [-3000 \ 3000]$$

$$e_Y^w \in [-2500 \ 2500]$$

Ranges for  $FLC_{inf}$

$$e_X^w \in [-3000 \ 3000]$$

$$e_Y^w \in [-2000 \ 0]$$

Si è deciso di campionare l'intervallo di  $e_X$  ogni 150 metri sia per  $FLC_{sup}$  che per  $FLC_{inf}$ , e l'intervallo di  $e_Y$  ogni 125 metri per  $FLC_{sup}$  ed ogni 100 metri per  $FLC_{inf}$ , cercando di trovare un compromesso tra accuratezza e occupazione di memoria. In tal modo viene a formarsi una griglia per ognuno dei due sistemi fuzzy. Il corrispondente valore del sistema fuzzy reale è associato ad ogni punto della griglia, ottenendo due mappe che rimpiazzano i controllori fuzzy: la prima mappa, relativa a  $FLC_{sup}$ , ha dimensione  $41 \times 41$ ; la seconda, relativa a  $FLC_{inf}$ , ha dimensione  $41 \times 21$ . Approssimando la caratteristica ingresso/uscita con una griglia si elimina l'esigenza di utilizzare un motore di inferenza fuzzy, ma di contro questo richiede una strategia per interpolare la mappa ottenuta. Quando il valore di ingresso della mappa non è un multiplo intero della "distanza di campionamento", ovvero quando il punto di coordinate  $(e_X^w, e_Y^w)$  non coincide con uno dei punti della griglia, deve essere usata una strategia di interpolazione: una semplice strategia è di usare come output il valore corrispondente al punto della griglia a distanza minima. Un'alternativa è di fare una media pesata dei quattro punti vicini, usando come pesi l'inverso della distanza di ognuno di questi quattro punti dal punto  $(e_X^w, e_Y^w)$ , come mostrato in Figura 16.

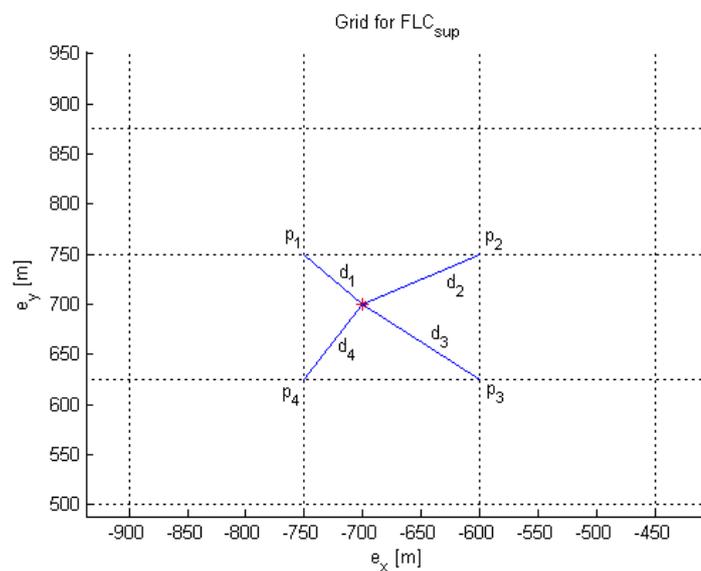


Figura 16.: Punto  $(e_X^w, e_Y^w)$  in  $FLC_{sup}$

Questo approccio è conveniente da un punto di vista computazionale, infatti le due mappe possono essere facilmente memorizzate come due matrici e il calcolo dell'angolo di heading di riferimento, nel caso più semplice, si riduce alla lettura di un elemento in una matrice. Quando è necessaria una interpolazione, l'output è calcolato come segue:

$$\text{FLC}_{\text{sup}} = \sum_{i=1}^4 w_{\text{norm}_i} \cdot v_i \quad \text{With: } w_{\text{norm}_i} = \frac{w_i}{\|w\|}, w_i = \frac{1}{d_i} \quad (7)$$

In cui  $v_i$  è l'output corrispondente al punto  $p_i$ .

#### 2.4.2 Confronto tra fuzzy e fuzzy approssimata

Con lo scopo di accertare la bontà e la realizzabilità di tale approssimazione del sistema fuzzy, viene presentato un reale confronto tra le traiettorie pianificate dai controllori fuzzy e quelle pianificate dalle mappe risultanti. La Figura 17 mostra quattro simulazioni in cui un singolo waypoint è posto in posizione  $(0, 0)$ , e con il fattore di scala  $S = 10$ . Il waypoint deve essere attraversato con un angolo di 270 gradi per le simulazioni nel semipiano superiore e con un angolo di 90 gradi per le simulazioni nel semipiano inferiore.

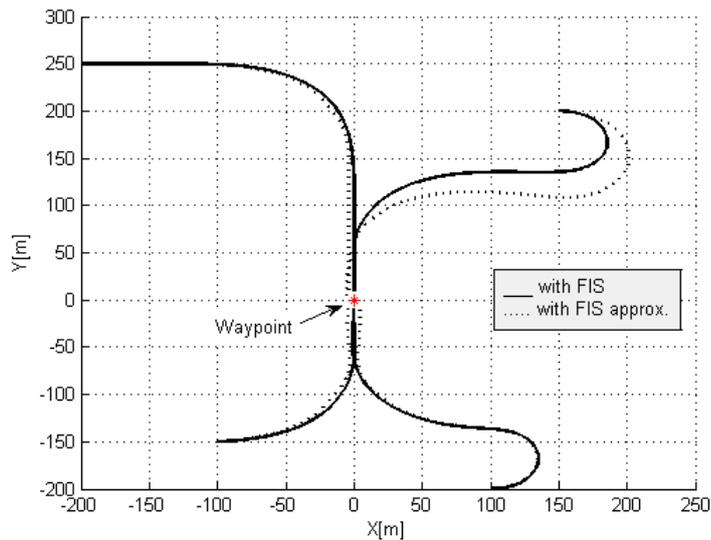


Figura 17.: Confronto delle traiettorie con un singolo waypoint

L'effetto principale di questa semplificazione può essere riassunta come un ritardo nella risposta del veicolo quando il veicolo si trova lontano dal target, e in un maggiore over-shoot quando il veicolo si trova nelle vicinanze del target, nella fase finale di allineamento, questo è più evidente nella simulazione del primo quadrante. Ciò risulta evidente anche dalla seconda simulazione, in cui si chiede al veicolo di attraversare 3 waypoint con differenti angoli di approccio, in particolare: 90 gradi per il waypoint A, 45 gradi per il waypoint B e 225 gradi per il waypoint C. Nella Figura 18 e nella Figura 19 vengono riportate rispettivamente la traiettoria e l'errore tra angolo di heading desiderato e reale. E' possibile notare che le traiettorie sono pressoché coincidenti e l'errore di heading si mantiene sempre sotto i 2 gradi, in entrambi i casi.

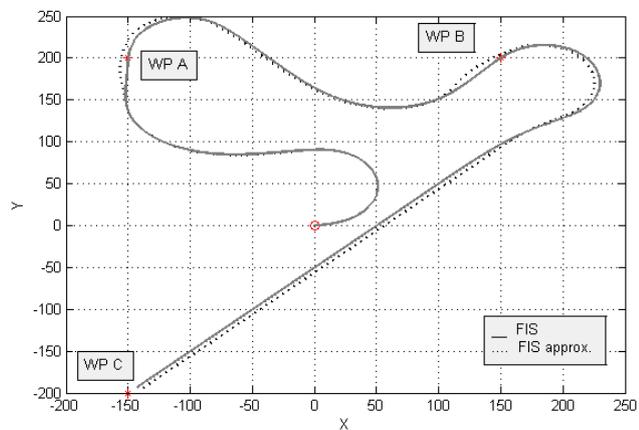


Figura 18.: Traiettorie con 3 waypoints

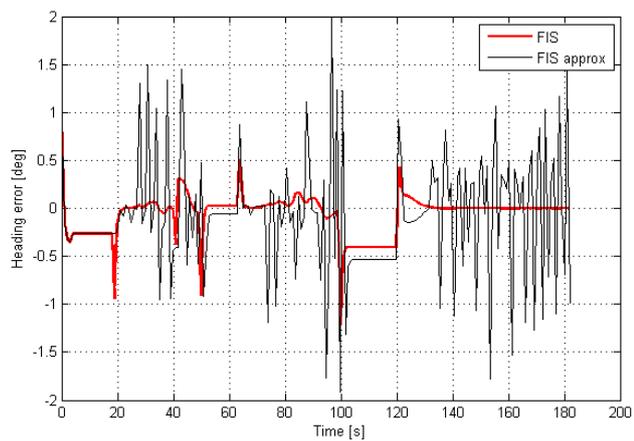


Figura 19.: Errore angolo di heading

### 2.4.3 Guida per brevi e lunghe distanze

Quando il veicolo è lontano dal waypoint, il vincolo sulla traiettoria dovuto alla direzione di approccio desiderata può essere tralasciato in favore di una traiettoria che segua il percorso a lunghezza minima. Tuttavia, il vincolo della direzione di approccio deve essere riconsiderato quando il veicolo è "abbastanza vicino" al waypoint, per permettere al veicolo di manovrare attorno ad esso. A questo scopo, è stata progettata una guida di lunga-distanza (LD), da utilizzare quando il waypoint è lontano. Di conseguenza, il controllore di heading  $f_X$  descritto in precedenza diventa la legge di guida per brevi distanze (SD)  $f_X^{SD}$ . La definizione di "lontano" è un parametro dipendente dal tipo di missione, questo valore deve essere compatibile con l'intervallo in cui la guida SD opera, per cui deve essere scelto come funzione del fattore di scala  $S$ . Lo switch tra guida LD e SD viene fatto in maniera morbida, fondendo gli output dei due controllori fuzzy attraverso la funzione  $\mu_R$ , che dipende dal valore di  $R_{AW}$ , definito come la distanza tra il veicolo e il waypoint:  $R_{AW} = \sqrt{e_X^2 + e_Y^2}$ , ottenendo così:

$$\chi_d = \mu_R(R_{AW}) \cdot f_X^{LD} + (1 - \mu_R(R_{AW})) \cdot f_X^{SD} \quad (8)$$

La guida LD utilizza l'angolo della line-of-sight (LOS)  $\theta_{LOS}$  per generare l'angolo di heading di riferimento per il veicolo. L'angolo di LOS è calcolato nel sistema di riferimento del waypoint ed è definito come segue

$$\theta_{LOS} = \arctan\left(\frac{-e_X^w}{e_Y^w}\right) \quad (9)$$

La funzione che fonde le due guide  $\mu_R(R_{AW})$  è definita come segue

$$\mu_R(R_{AW}) = \begin{cases} e^{-\frac{(R_{AW}-d)^2}{2\sigma^2}}, & R_{AW} < d \\ 1, & R_{AW} \geq d \end{cases} \quad (10)$$

questo garantisce che, quando il veicolo è lontano almeno  $d$  metri dal waypoint, si ha che  $\mu_R(R_{AW}) = 1$  e l'angolo di heading di riferimento  $\chi_d$  viene generato dalla guida LD.

$$f_X^{LD} = \chi_w + \theta_{LOS} \quad (11)$$

La Figura 20 mostra lo schema a blocchi completo del sistema di guida. La Figura 21 mostra invece un esempio in cui entrambe le guide vengono applicate, con  $\sigma = 5(\text{m})$ ,  $d = 35(\text{m})$  e  $S = 15$ .

Con tale scelta per  $\sigma$ , quando la distanza dal waypoint  $\sigma$  decresce sotto i 35 metri, la componente di guida LD riduce il suo effetto gradualmente finché sotto i 20 metri il suo effetto è praticamente nullo e la guida SD acquisisce piena autorità sul calcolo di  $\chi_d$ .

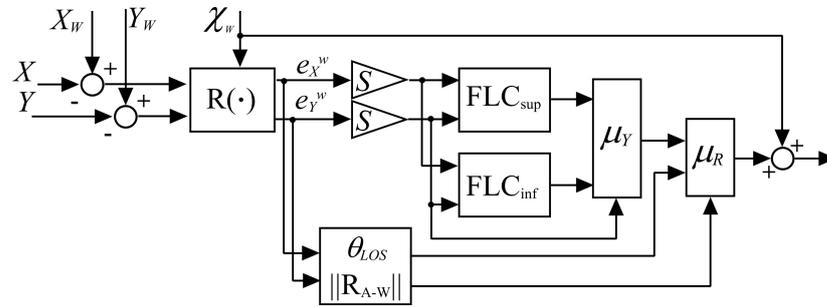


Figura 20.: Schema di guida completo

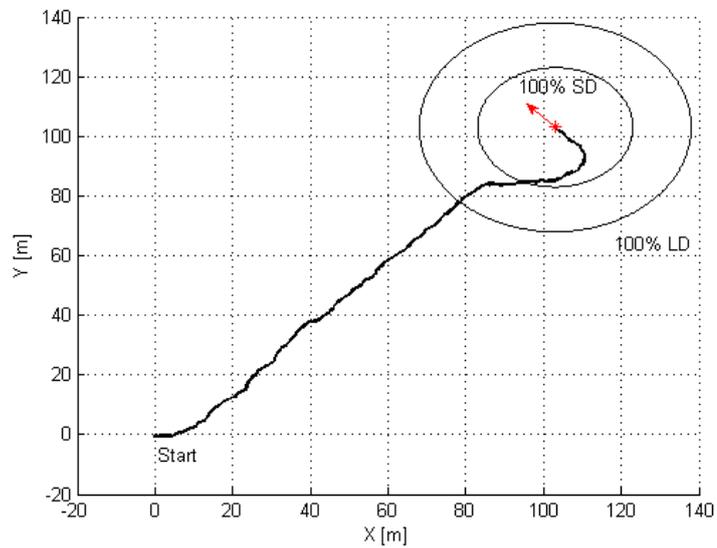


Figura 21.: Esempio di guida LD e SD

#### 2.4.4 Simulazione hardware-in-the-loop con autopilota ICARO

L'Università di Pisa iniziò il progetto ICARO nel 2007: un progetto interno che punta allo sviluppo di una piattaforma di calcolo per scopi generici, che sia low-cost ma affidabile e potente, e non dipendente in modo specifico da alcuna tipologia di sensori [8]. La versione utilizzata nei nostri test, che è mostrata in Figura 22, appartiene alla seconda famiglia di questi autopiloti, denominata ICARO II o ICARO-Light, poiché è stato progettato per essere piccolo e leggero. Le interfacce presenti in ICARO II sono: UART, SPI, I2C, Analog/Digital I/Os, PWM generation/capture, e, infine, interfaccia CANBUS, quest'ultima verrà utilizzata come canale di comunicazione ad alta velocità per lo scambio rapido dei dati necessari per l'implementazione del simulatore hardware-in-the-loop (HIL). La CPU ICARO II può essere programmata da qualsiasi compilatore che la supporta; linguaggio standard C è la scelta una programmazione di basso livello e per avere pieno accesso alle risorse hardware, in può essere utilizzato alternativa uno strumento di prototipazione rapida. La CPU che abbiamo utilizzato, per esempio, è completamente supportata dal MathWorks Embedded Coder. Con l'aiuto di tale strumento, gli algoritmi di GNC, sviluppati sotto Simulink, possono essere compilati, scaricati ed eseguiti in tempo reale sulla CPU.

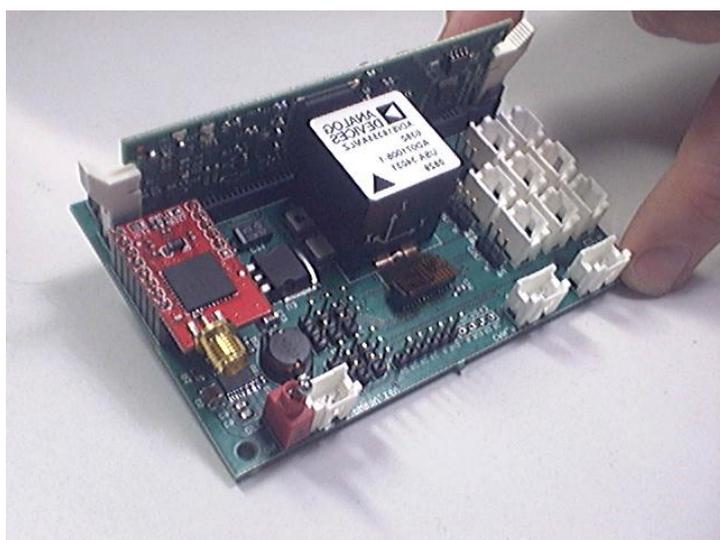


Figura 22.: Autopilota ICARO II con GPS e IMU a 6-DOF

La simulazione con hardware-in-the-loop (HIL) è un approccio molto utile per la validazione e la qualificazione delle funzionalità dell'hardware, o parti di esso, che si intende utilizzare (eventualmente con l'aggiunta del firmware) prima dell'installazione nell'ambiente operativo. I moderni veicoli senza pilota necessitano di funzionalità avanzate e un simulatore HIL costituisce una risorsa molto utile in quanto permette di testare la combinazione di hardware/software scelta, che verrà poi impiegata nelle operazioni. Un autopilota presenta sempre svariate interfacce per i sensori, in una simulazione HIL tutti i dati sensoriali possono essere simulati e inviati all'hardware dell'autopilota, che, dopo l'ela-

borazione dei dati, produce un'uscita per i suoi attuatori. Questa uscita può essere data agli attuatori reali e, in parallelo, inviata al simulatore in modo che esso esegua un altro passo di simulazione e produca nuovi dati sensoriali da inviare. Un altro problema rilevante per il funzionamento dei sistemi senza pilota è il design dell'interfaccia uomo-macchina. Per testare tutte le funzionalità del sistema in completa sicurezza, è necessario fornire al pilota le stesse interfacce di ingresso/uscita che verranno poi utilizzate nelle operazioni, e fornire in feedback al pilota un ambiente sintetico che sia realistico [7], così è necessario un altro sistema chiamato simulatore man-in-the-loop (MIL). Per l'ambiente sintetico, è molto importante utilizzare scenari che sono rappresentativi di luoghi reali sulla terra, al posto di luoghi inesistenti, puramente immaginari: una sensazione di full-immersion e la concentrazione sulle operazioni da eseguire è più elevata quando il pilota è portato in uno scenario 3D di un posto che conosce, e lo stesso luogo viene rappresentato in 2D (vista mappa) sulla schermata della stazione di terra. Così uno scenario sintetico realistico viene interfacciato in tempo reale con il nostro simulatore. Un diagramma a blocchi del ambiente di simulazione completo è riportato nella Figura 23.

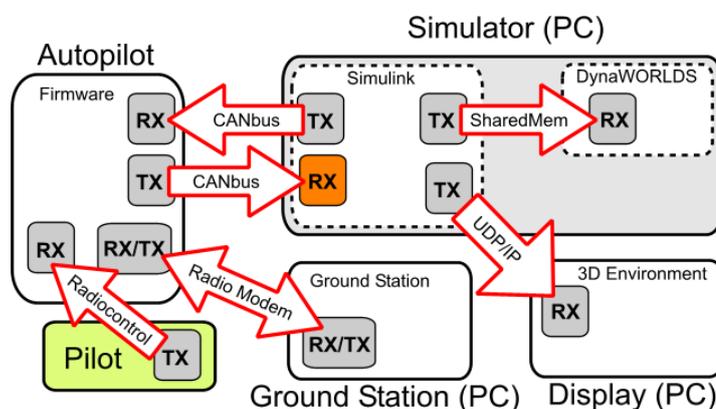


Figura 23.: Diagramma a blocchi del simulatore completo HIL

L'Università di Pisa ha una partnership con una società italiana (*Tecnodrone*, [www.tecnodrone.it](http://www.tecnodrone.it)), che progetta, produce e gestisce dei veicoli multi-rotore (quad-rotor, six-rotor, ecc.), per cui un simulatore completo HIL + MIL è stato sviluppato per uno dei loro quad-rotor. Il pilota automatico e il simulatore si scambiano pacchetti di tipo canbus alla frequenza di 100Hz, in particolare: 6 pacchetti vengono scambiati dal simulatore al pilota automatico per accelerometri, giroscopi, magnetometri, baro-altimetro e dati GPS, e 3 dal pilota automatico al simulatore per i comandi e vari dati di debug. Il verificarsi di errori di sincronizzazione o di ritardi nello scambio di dati comprometterebbe seriamente la stabilità del veicolo, quindi una simulazione di volo che sia stabile è già una buona indicazione di avvenuta sincronizzazione. Per convalidare l'efficacia del sistema di sincronizzazione e definire una metrica per la misurare la fedeltà della simulazione, sono identificati due parametri: la durata effettiva di un passo di simulazione (compresa l'attesa attiva) del simulatore o Task Execution Time (TET), e il numero di pacchetti (NP) che ad ogni passo temporale sono accodati nel buffer di ricezione canbus. Il valore ideale del TET è la velocità di

comunicazione dell'autopilota:  $1/100$  di secondo, mentre il valore desiderato di NP è 0, il che significa che il simulatore è abbastanza veloce da raggiungere il punto di sincronizzazione con l'autopilota prima che questo abbia generato nuovi dati.

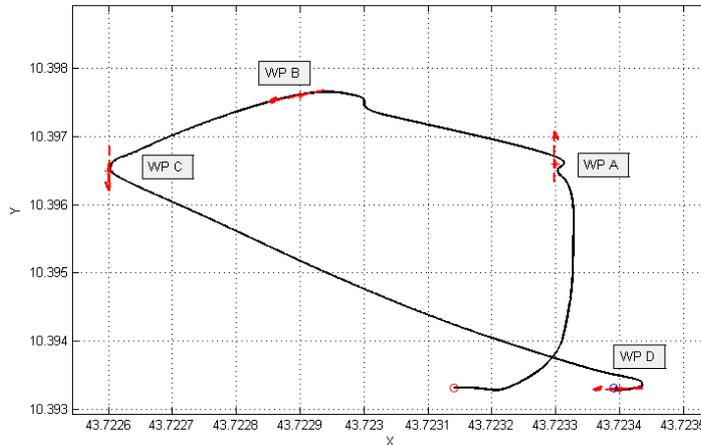


Figura 24.: Simulazione HIL con 4 waypoints

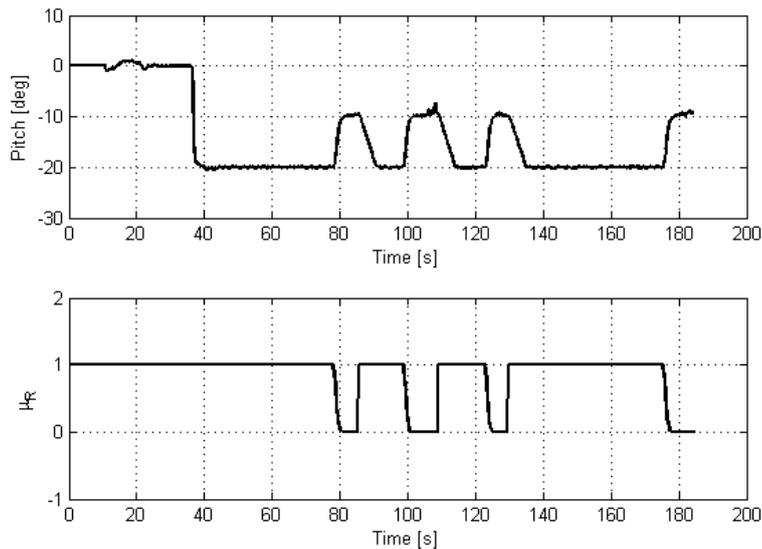


Figura 25.: Evoluzione dell'angolo di beccheggio

La Figura 24 mostra la traiettoria eseguita in una simulazione con HIL, nel quale il veicolo simulato (un quad-rotor) procede attraversando 4 waypoints con differenti angoli di approccio. Anche in questo caso tali waypoints hanno la medesima altitudine. In questa simulazione viene utilizzata la differenziazione della legge di guida in LD e SD come descritto in precedenza e viene usata una velocità differente nei due casi, come si evince dalla Figura 25 in cui è riportato anche il valore della funzione  $\mu_R(R_{AW})$  che fonde le due leggi di

guida: nello specifico il quad-rotor procede in LD con inclinazione di 20 gradi di beccheggio e in SD con inclinazione di 10 gradi, che corrisponde a una velocità minore. L'angolo di attraversamento desiderato (in gradi) è di 90 per il waypoint A, 225 per il waypoint B, 270 per il waypoint C e 180 per il D. Si noti che i valori di X e Y sono espressi in latitudine e longitudine e la simulazione si svolge a Pisa nei pressi di Piazza dei Miracoli. Le Figura 26 e Figura 27 riguardano la parte HIL della simulazione: esse mostrano che i valori di TET sono molto vicini a 10 ms, e la buona sincronizzazione si evince dai valori di NP, che sono praticamente sempre uguali al valore desiderato 0. Questo risultato garantisce che la performance ottenuta con la simulazione HIL dovrebbe essere coincidente con i risultati reali, se il modello del veicolo è affidabile.

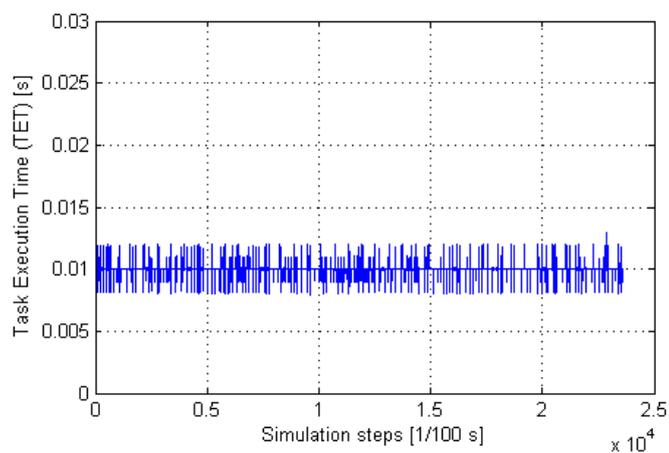


Figura 26.: Valori del TET

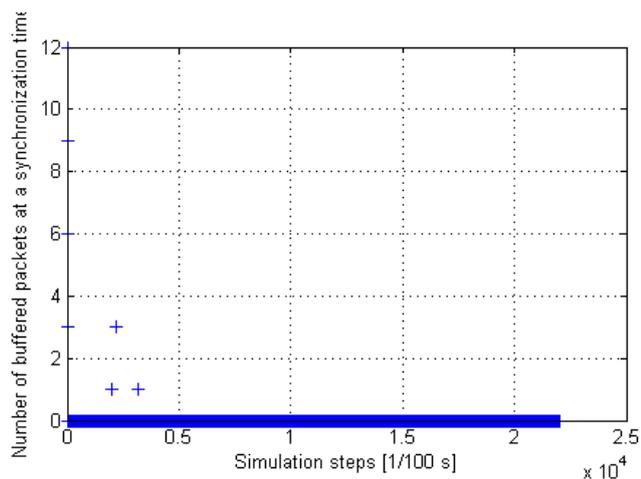


Figura 27.: Valori di NP

## 2.5 SIMULAZIONI E RISULTATI SPERIMENTALI

Con lo scopo di mostrare l'efficacia del sistema di guida proposto, due differenti scenari sono stati utilizzati nelle simulazioni: navigazione accurata tra i palazzi in un contesto cittadino, e intercettazione multipla dello stesso target (waypoint) con differenti direzioni di approccio.

Il primo scenario è mostrato in Figura 28 e in Figura 29, nel quale il veicolo deve raggiungere un target rappresentato dall'ultimo waypoint. Per questo scopo esso si muove attraversando 8 waypoint, posizionati in modo opportuno, in cui la direzione di attraversamento di ciascun waypoint è stata scelta per facilitare l'approccio a quello successivo.

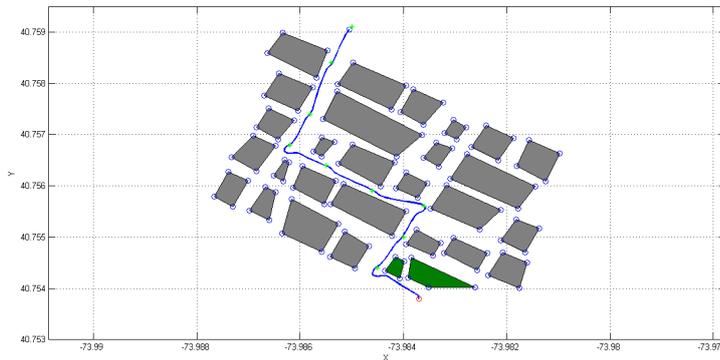


Figura 28.: Navigazione tra i palazzi, vista in pianta



Figura 29.: Navigazione tra i palazzi, vista 3D

Il secondo scenario rappresenta una situazione in cui è richiesto al veicolo un approccio multiplo allo stesso target. La Figura 30 e la Figura 31 mostrano che non è necessario utilizzare molteplici waypoint, ma un singolo waypoint è sufficiente, per il quale vengono tuttavia specificate diverse direzioni di attraversamento.

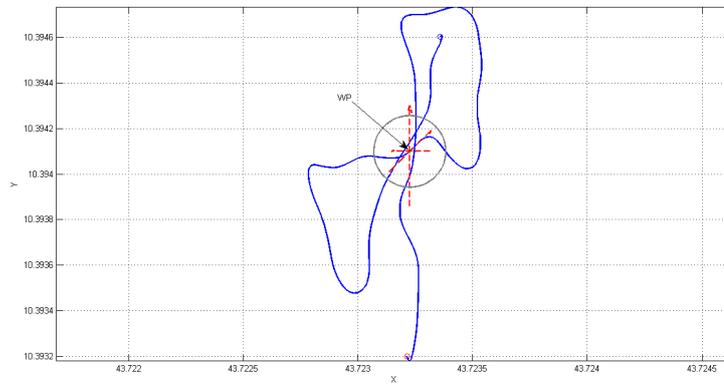


Figura 30.: Intercettazione multipla, vista in pianta

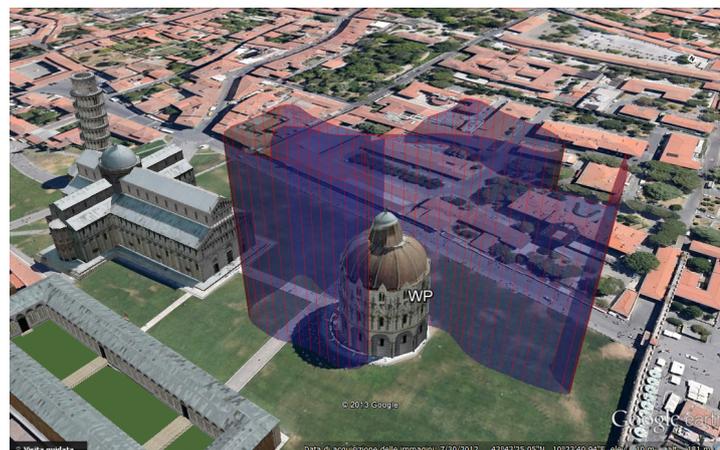


Figura 31.: Intercettazione multipla, vista 3D

# 3

## STUDIO DI VISIBILITÀ DI UN TARGET IN UN AMBIENTE STRUTTURATO

### 3.1 INTRODUZIONE

Il problema della visibilità di un target in un ambiente strutturato, caratterizzato da ostacoli fissi o mobili, è molto discusso in letteratura, soprattutto per applicazioni che riguardano l'osservazione del target, intesa come sorveglianza. In [15] viene presentato un algoritmo per il calcolo del moto di un veicolo che soddisfi i vincoli di visibilità di un target mobile che è prevedibile o parzialmente prevedibile, minimizzando la distanza totale percorsa dal veicolo stesso. In [13] viene invece implementato un task di osservazione vincolando il veicolo a mantenere una distanza fissa dal target e viene inoltre fornita una condizione per l'esistenza o meno della strategia di sorveglianza. Il problema dell'osservazione di un target è trattato anche in [12] e in [14], in cui i target da osservare possono essere mobili e molteplici. Essi sono assunti imprevedibili ma la posizione degli ostacoli fissi è nota a priori. L'algoritmo riportato calcola la strategia di moto che mantenga la visibilità, questa agisce massimizzando la "distanza minima di fuga", ovvero la distanza minima che serve al target per muoversi sfuggendo alla regione di visibilità dell'osservatore, in particolare in [14] l'algoritmo proposto viene applicato ad un robot reale in uno scenario uno vs. uno. Una interpretazione duale del problema viene data in [16], in cui viene ancora implementato un task di osservazione ma viene anche fornita al target una legge di moto ottima per fuggire dalla regione di visibilità. In questa sezione si descrive lo studio della visibilità di un target in un ambiente strutturato. Il problema che ci si pone è il calcolo della strategia di moto di un robot che mantenga la visibilità di un target, che in una prima fase è considerato fisso, nel suddetto ambiente. In questa fase non vengono considerati i vincoli di una normale pianificazione del moto ma solo quelli relativi alla visibilità devono essere soddisfatti. In particolare viene introdotta una incertezza associata alla posizione del target, infatti la definizione di visibilità che è solito trovare in letteratura ha una caratterizzazione booleana, legata alla presenza o meno di una line-of-sight (LOS) tra il target e il veicolo. Associando una incertezza alla posizione del target si ottiene una informazione di visibilità sfumata e più ricca, utile all'implementazione dell'algoritmo che fornisce una direzione di riferimento al veicolo. In particolare in questo capitolo viene dapprima definito lo spazio di lavoro, ovvero l'ambiente in cui il target risiede; in seguito viene definita una densità di probabilità che introduce l'incertezza di posizione sul target ed elaborata una definizione di "livello di visibilità"; ed infine viene riportato l'algoritmo che calcola la direzione di miglioramento di visibilità, che è accompagnato e validato da alcune simulazioni.

### 3.2 DEFINIZIONE DELLO SPAZIO DI LAVORO

L'obiettivo che ci si pone è di fare un'analisi del livello di visibilità di un generico oggetto (che può rappresentare un target, fisso o mobile) in un ambiente strutturato, che coincide con uno scenario tipicamente urbano, caratterizzato dalla presenza di ostacoli fissi. L'analisi ha principalmente l'obiettivo di individuare quali zone dell'ambiente sono caratterizzate da un alta visibilità di tale oggetto. Questo viene fatto attraverso una funzione che, per ogni punto dello scenario, calcola il livello di visibilità dell'oggetto in tale punto. Questo può permetterci, in una fase successiva, di mettere a punto un algoritmo che, data una posizione qualsiasi nello scenario, suggerisca ad un generico veicolo, che può aver interesse ad inseguire o osservare l'oggetto, una direzione che apporti un miglioramento del livello di visibilità dell'oggetto stesso. Il livello di visibilità di un oggetto in un ambiente complesso è una grandezza che verrà definita in seguito, essa tuttavia dipende da diversi fattori quali:

- Posizione del target
- Collocazione degli ostacoli
- Incertezza che affligge la posizione del target

L'ambiente strutturato utilizzato in questo studio è sempre rappresentato in pianta, ed è caratterizzato da una zona centrale che è ricca di ostacoli e da una zona esterna più libera, come si nota dalla Figura 32. La scelta di un ambiente così fatto, non eccessivamente pieno di ostacoli, è funzionale ad alcune applicazioni descritte in seguito.

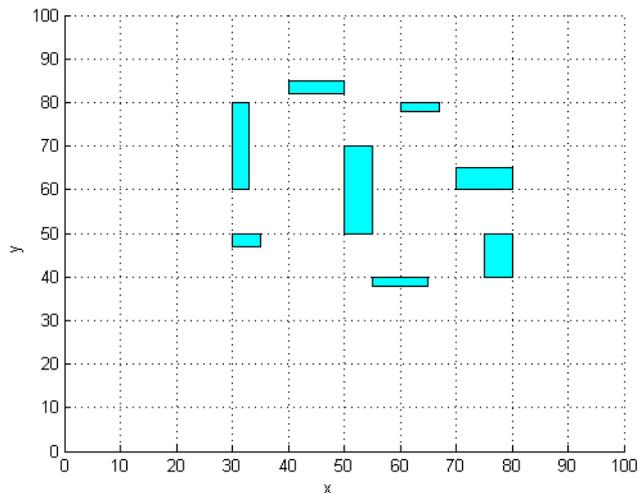


Figura 32.: Esempio di ambiente strutturato

### 3.3 DEFINIZIONE REGIONE DI INCERTEZZA DEL TARGET

Una prima definizione del livello di visibilità di un oggetto in un punto potrebbe coincidere con l'esistenza o meno di una congiungente, una line-of-sight (LOS), tra il punto in questione e l'oggetto. L'inesistenza di una LOS sarebbe causata dal fraporsi di un ostacolo nel percorso tra punto e oggetto. In tal modo però si otterrebbe una definizione booleana del livello di visibilità: oggetto in vista o non in vista. Ai fini del nostro studio è necessaria una definizione sfumata del livello di visibilità, non booleana, per tale scopo si considera una regione di incertezza attorno all'oggetto (target). La posizione del target è afflitta da un'incertezza descritta da una densità di probabilità di tipo gaussiana bi-variata.

$$f_T(x, y) = \frac{1}{\sigma_X \sigma_Y 2\pi} e^{-\frac{(x-x_T)^2}{2\sigma_X^2} - \frac{(y-y_T)^2}{2\sigma_Y^2}} \quad (12)$$

In cui  $\sigma_X$  e  $\sigma_Y$  sono le deviazioni standard lungo gli assi X e Y,  $x_T$  e  $y_T$  rappresentano invece la posizione esatta del target. Tale funzione, nel caso di  $(x_T, y_T) = (50, 50)$  e  $\sigma_X = \sigma_Y = 10$ , origina la curva mostrata in Figura 33

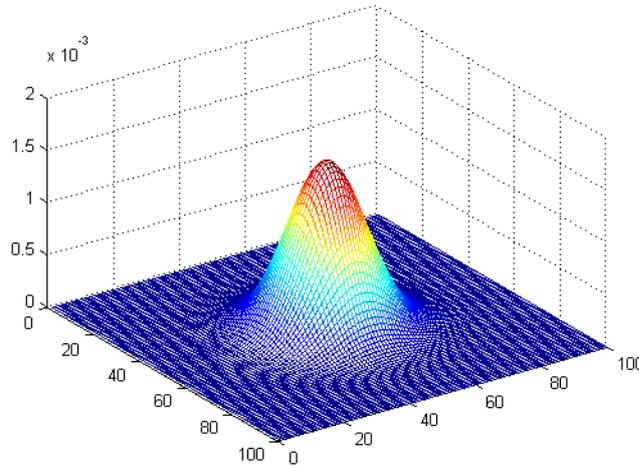


Figura 33.: Gaussiana bi-variata

Questa densità di probabilità, potrebbe essere intesa in maniera diversa rispetto ad un'incertezza sulla posizione esatta. Potrebbe infatti rappresentare una predizione della futura posizione del target, fatta entro un certo intervallo temporale, centrata nell'ultima posizione del target nota al veicolo. Infatti più ci si allontana da quel punto, meno è alta la probabilità che il target si trovi effettivamente lì, poiché dovrebbe percorrere un percorso più lungo nella stessa direzione.

### 3.4 DEFINIZIONE DEL LIVELLO DI VISIBILITÀ

La definizione che in questo studio viene data al livello di visibilità differisce da quello che è solito trovare in letteratura, infatti essa tiene conto anche dell'incertezza di posizione del target. Infatti qualunque sia l'interpretazione data alla densità di probabilità descritta nel paragrafo precedente, la definizione del livello di visibilità non può prescindere da essa. Il livello di visibilità di un target in un punto è inteso come la probabilità che il target si trovi effettivamente in una delle sue "possibili posizioni", moltiplicata per un fattore booleano che indica l'esistenza o meno di una congiungente (LOS) tra la probabile posizione del target e il punto stesso. Analizzando il solo caso planare, la probabilità di avere il target in vista, per un veicolo che si trova nel punto di coordinate  $(x^*, y^*)$  appartenente ad un certo dominio  $D$ , è data da:

$$p_{Tv}(x^*, y^*) = \int \int_D f_T(x, y) \cdot f_{LOS}(x^*, y^*, x, y) dx dy \quad (13)$$

In cui  $f_{LOS}$  è la funzione binaria rappresentante il fattore booleano sopracitato. Numericamente questo significa che per ogni possibile posizione in cui il target può trovarsi (a cui la d.d.p. descritta sopra associa una probabilità) ne sommo la probabilità moltiplicata per il fattore booleano. A livello numerico il calcolo del livello di visibilità viene infatti implementato come segue.

$$p_{Tv}(x^*, y^*) = \sum_{(x, y) \in I} f_T(x, y) \cdot f_{LOS}(x^*, y^*, x, y) \quad (14)$$

In cui  $I$  è un intorno circolare centrato nella posizione del target (sia  $r_t$ ), in particolare  $I = I(r_t, 3\sigma) = \{r^* : \text{dist}(r^*, r_t) \leq 3\sigma\}$ . La Figura 34 mostra tale intorno e le LOS che congiungono i punti appartenenti all'intorno con il veicolo. I punti in rosso sono quelli appartenenti all'intorno a cui la  $f_T$  associa una probabilità, tale probabilità verrà poi filtrata dalla  $f_{LOS}$ : verrà quindi moltiplicata per 1 se esiste una congiungente tra il punto e il veicolo (LOS verdi) per 0 altrimenti (LOS nere).

A questo punto, avendo dato una definizione del livello di visibilità di uno oggetto in un ambiente complesso rispetto ad un punto, che coinciderà con la posizione di un veicolo (osservatore o inseguitore), con le immagini che seguono, si vuole mostrare l'effetto di tale formula in tutti i punti dello scenario sopra descritto. La Figura 35 (a) mostra le LOS che congiungono il target ai diversi punti dell'ambiente, indicando quindi quali zone hanno visibilità del punto esatto in cui si trova il target (zone in verde) e quali no (zone in rosso), considerando tuttavia un raggio di visibilità massima pari a 30 metri.

In questo modo l'informazione di visibilità è solo booleana (target in vista o meno), infatti non si è considerata la regione di incertezza che caratterizza il target. Considerando adesso la regione di incertezza sul target descritta dalla

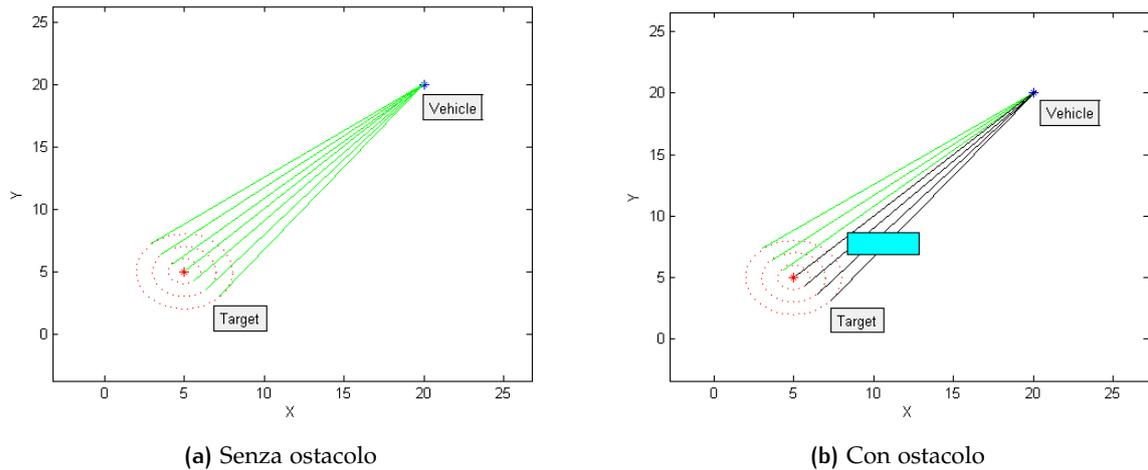


Figura 34.: Definizione livello di visibilità

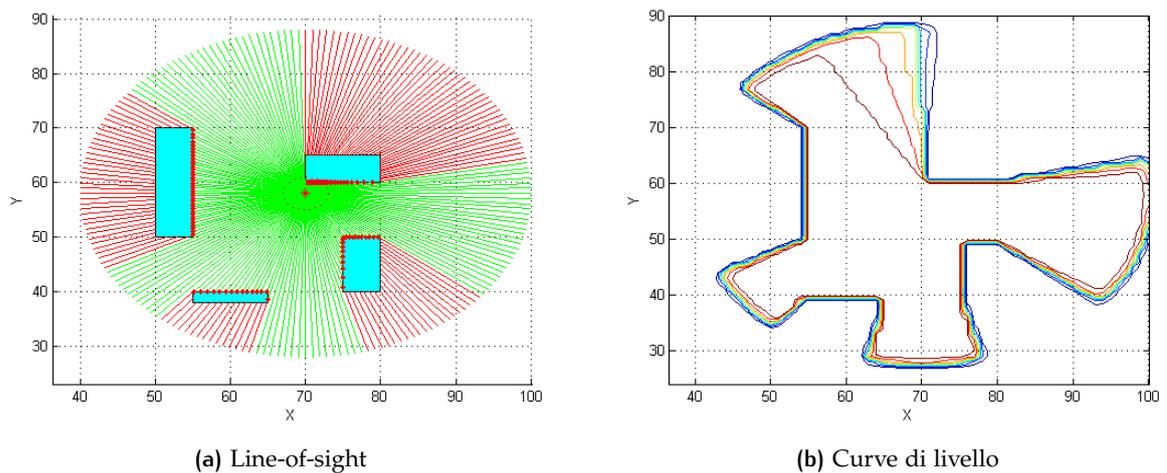


Figura 35.: Visibilità target fisso

gaussiana  $f_T(x, y)$ , otteniamo per ciascun punto dell'ambiente una informazione di visibilità sfumata, quindi più ricca, che va da un valore pari a zero, se nessuno dei punti della regione di incertezza è in vista, ad un valore pari a 1, se l'intera regione di incertezza è in visuale. Ciò è visibile dalla Figura 35 (b), in cui sono riportate le curve di livello della visibilità per il target nella medesima posizione, ottenute considerando la d.d.p.  $f_T(x, y)$  con  $\sigma_x = \sigma_y = 3$ , in cui le zone delimitate da bordi di colore più scuro sono quelle a visibilità più alta. È evidente che le zone di confine tra parte verde e rossa della Figura 35 (a) sono quelle in cui varia il livello di visibilità. Infatti il target potrebbe trovarsi, con una certa probabilità, in un punto qualsiasi della sua regione di incertezza, e il punto potrebbe essere non più visibile da tale zona di confine.

### 3.5 METODO PER IL CALCOLO DELLA DIREZIONE DI VISIBILITÀ

Si vuole, a questo punto, sfruttare la capacità di conoscere il livello di visibilità del target in qualsiasi punto dello spazio per mettere a punto un algoritmo che suggerisca una direzione che apporti un miglioramento alla visibilità attuale. Tale algoritmo potrebbe essere utilizzato da un veicolo che ha interesse sul target, per esempio il cui scopo potrebbe essere di inseguire o sorvegliare il target. L'algoritmo tiene conto della variazione della probabilità nello spazio e produce una direzione di moto, ciò comporta il calcolo di un vettore velocità caratterizzato da una sua orientazione e un suo modulo. Il calcolo del vettore velocità viene fatto attraverso il gradiente di  $p_{Tv}$ , calcolato al variare della posizione del veicolo.

$$\bar{v} = \left. \frac{\partial p_{Tv}}{\partial \bar{r}} \right|_{\bar{r}=\bar{r}_v} \quad (15)$$

Con  $\bar{r}_v \in X^n$ , generico vettore che descrive la posizione del veicolo. Limitandoci alle direzioni in cui tale gradiente è positivo, si ottengono le direzioni di crescita della probabilità. Per rendere possibile una implementazione numerica dell'espressione sopra riportata viene calcolato il gradiente entro un preciso intorno della posizione del veicolo, da un punto di vista numerico l'algoritmo viene quindi implementato come segue.

$$\max_{\bar{r}} [p_{Tv}(\bar{r}) - p_{Tv}(\bar{r}_v)], \quad \bar{r} \in B(\bar{r}_v, d) \quad (16)$$

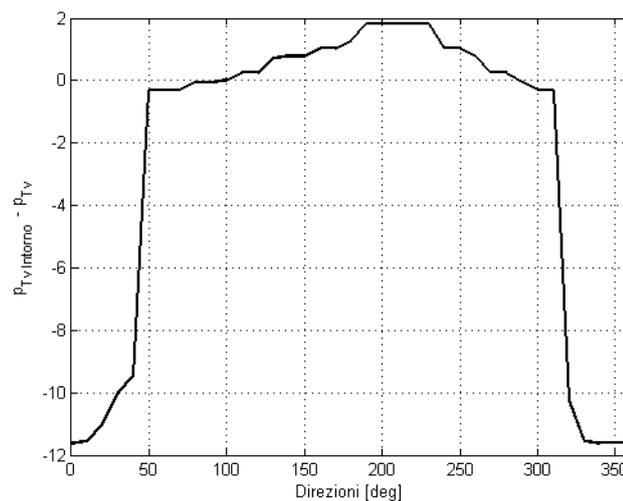


Figura 36.: Variazione di visibilità

In cui  $B(r_v, d)$  è un intorno circolare,  $B(r, d) = \{r^* \in X^n : \text{dist}(r^*, r) \leq d\}$ . Trovato  $r^* = [x^*, y^*]^T$ , il punto corrispondente al valore massimo nel semplice caso planare, la direzione è poi calcolata come  $\arctan(\frac{y^*}{x^*})$ . Nel caso in cui ci siano più punti corrispondenti al valore massimo, ovvero più direzioni che comportano lo stesso miglioramento di visibilità viene scelta la direzione che consente al veicolo di avvicinarsi maggiormente al target. Viene riportato di seguito un esempio in cui un veicolo si trova in posizione (68, 70), mentre il resto coincide con lo scenario visto in precedenza. La Figura 36 mostra la variazione della visibilità al variare della direzione attorno al veicolo, nella Figura 37 viene mostrata invece la direzione prodotta dall' algoritmo.

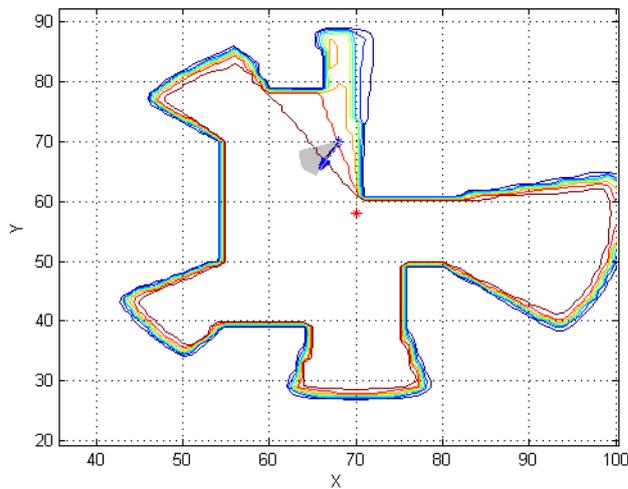


Figura 37.: Direzione suggerita

E' possibile notare le direzioni di massima crescita di visibilità sono comprese nell'intervallo [190, 230] gradi (riportate in grigio nell'immagine), la direzione selezionata tra queste (circa 230 gradi) è quella che più indirizza il veicolo verso il target. L'efficacia dell'algoritmo proposto è maggiormente visibile dalle simulazioni seguenti, in cui l'algoritmo viene applicato iterativamente, ed è possibile notare uno spostamento netto verso zone a più alta visibilità del target. La prima simulazione viene fatta con il veicolo che si trova in posizione iniziale (70, 70) ed il target nella stessa posizione dello scenario precedente. La Figura 38 mostra la traiettoria suggerita dall'algoritmo, inoltre riporta la stessa sulle curve di livello della visibilità.

Come è possibile notare, il veicolo si porta in una posizione in cui la visibilità è massima. Una seconda simulazione è mostrata nella Figura 39, in cui il veicolo ha posizione iniziale (90, 48), ed il target si trova nella posizione (74, 51). Dall'immagine che riporta le curve di livello, è possibile notare che il veicolo si porta in una zona ad alta visibilità (caratterizzata da una curva di livello più scura), in un punto in cui si assicura una visibilità completa del target e dell'intera regione di incertezza.

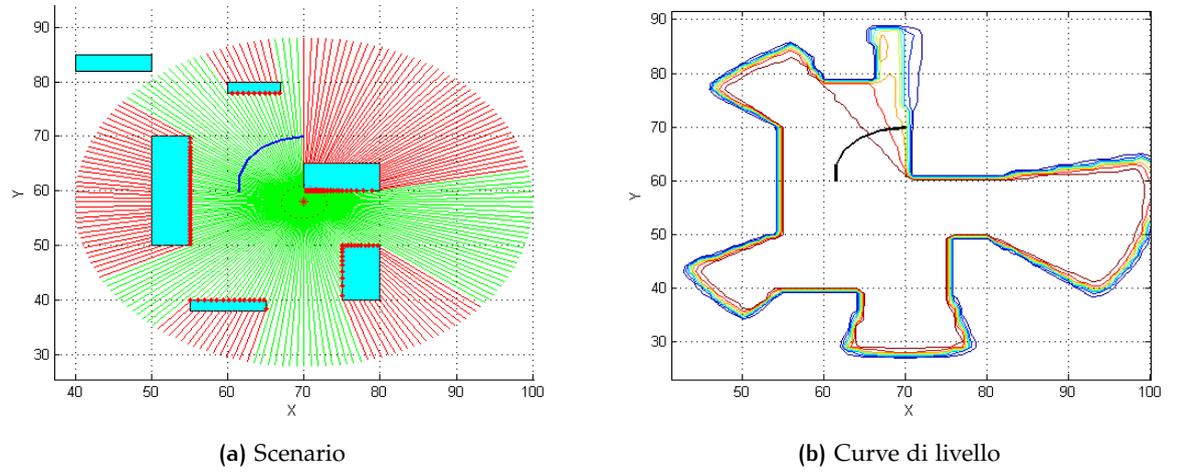


Figura 38.: Catena di gradienti 1

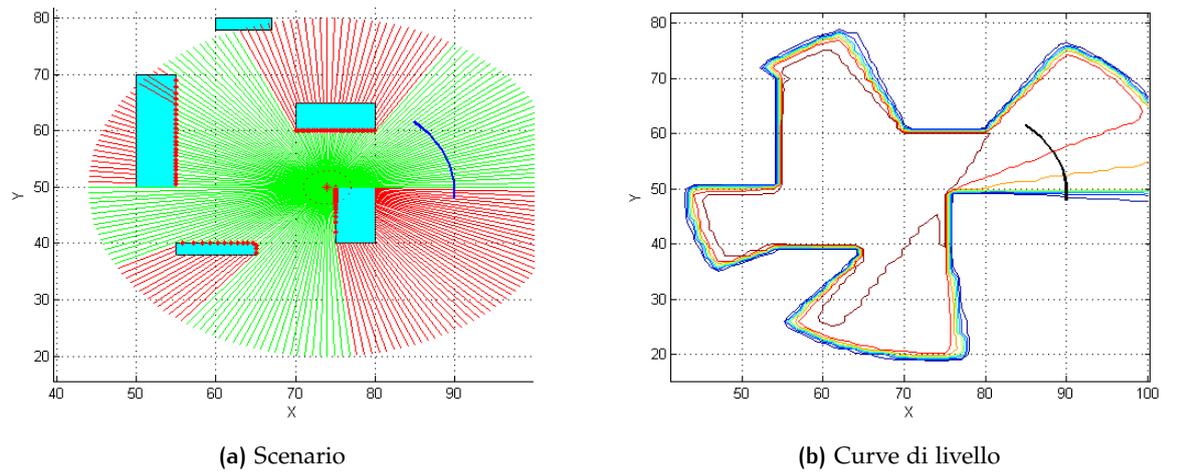


Figura 39.: Catena di gradienti 2

# 4

## POSSIBILI APPLICAZIONI DELLO STUDIO DI VISIBILITÀ

### 4.1 INTRODUZIONE

Lo studio di visibilità fatto nel capitolo precedente apre la strada a diverse applicazioni. Tra le principali applicazioni possibili di tale studio vi è il task di osservazione, ampiamente discusso in letteratura, in cui il veicolo deve mantenere la visibilità del target portandosi, tuttavia, ad una distanza non inferiore ad una certa soglia. Questa applicazione è trattata brevemente nel paragrafo successivo, in quanto si riconduce semplicemente al posizionamento opportuno dei waypoints, infatti una volta individuate le zone ad alta visibilità e posizionati i waypoints i veicoli possono raggiungerli con semplici strategie di guida: una delle quali potrebbe essere la guida a waypoints fuzzy descritta in precedenza, o una semplice guida LOS. In questa sezione viene descritta principalmente un'altra possibile applicazione dello studio di visibilità, rappresentata da un task di intercettazione, considerando un target non più fisso ma mobile in un ambiente strutturato. Il problema che ci si pone è il calcolo della strategia di moto di un veicolo che mantenga la visibilità del target mobile nel suddetto ambiente. Il generico problema è diviso in due categorie: in base alla presenza o meno di intelligenza nel target. Nel caso in cui il target non è intelligente si ha che esso si muove a velocità costante o con una certa accelerazione che tuttavia è nota al veicolo principale (osservatore o intercettore). Nel caso più complesso in cui il target è intelligente si ha che esso utilizza un algoritmo duale a quello utilizzato dal veicolo principale, ovvero il sistema di guida del target gli fornisce una direzione utile a fuggire dalla regione di visibilità del veicolo principale. La strategia utilizzata massimizza la probabilità che il target si trovi in vista, in base ad una conoscenza probabilistica della posizione del target, che può essere interpretata anche come una predizione, entro un certo intervallo temporale, della sua posizione futura. La strategia applicata è dapprima risolta numericamente, vengono riportate sia delle simulazioni in cui è presente un solo target e un solo veicolo principale che delle simulazioni in cui si hanno molteplici veicoli principali. Il task di intercettazione viene implementato ispirandosi alla più classica delle guide per questi tipi di task, la Proportional-Navigation-Guidance (PNG), e modificando questa per tenere in considerazione anche la visibilità del target. Inoltre viene data una impostazione teorica a tale applicazione, considerando l'intercettazione come un gioco differenziale tra Pursuer-Eavder, viene quindi ricavato un funzionale di costo e viene risolto matematicamente il problema di ottimo.

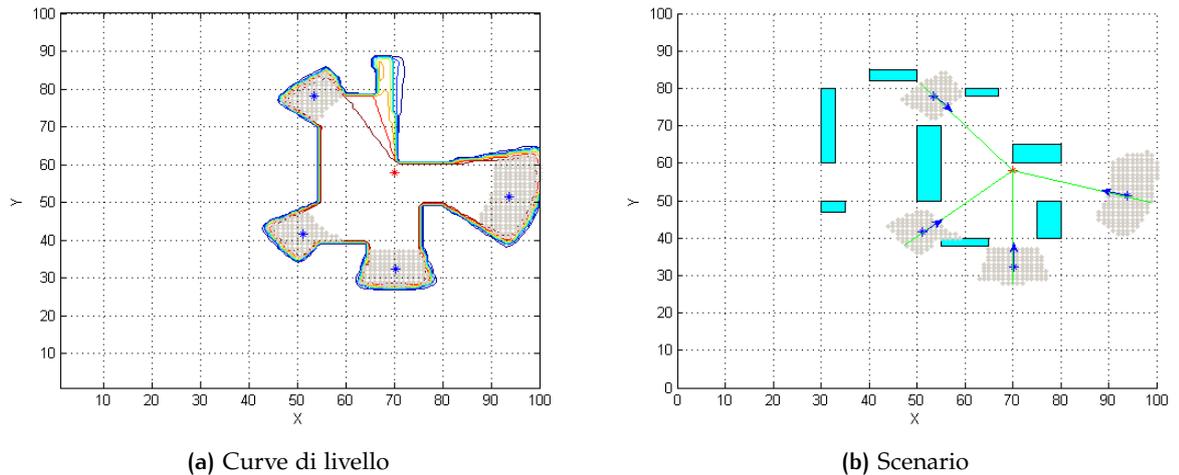


Figura 40.: Esempio task di sorveglianza

## 4.2 TASK DI SORVEGLIANZA

Un classica applicazione dello studio di visibilità di un oggetto o target in un ambiente strutturato consiste nel formulare una strategia di moto per un veicolo che ha il compito di osservare il target. Il veicolo in questo caso avrebbe semplicemente il compito di portarsi in un punto in cui il livello di visibilità del target è alto. Si potrebbe infatti sfruttare la capacità di individuare delle zone caratterizzate da un alto livello di visibilità del target, e delimitare queste con una circonferenza centrata nella posizione del target che indichi la distanza minima da mantenere, come tipicamente avviene in questi tipi di task. In seguito si potrebbe inserire banalmente uno o più waypoints per esempio nei centroidi di tali zone. Questo potrebbe esser fatto attraverso un meccanismo centralizzato che sceglie opportunamente le posizioni dei waypoints e le comunica ai veicoli osservatori che avrebbero semplicemente il compito di raggiungere e mantenere tali posizioni, oppure si potrebbe pensare che questo venga fatto da un veicolo "master" che esegue tali calcoli e comunica i risultati. Un esempio di ciò è mostrato in Figura 40, in cui le zone in grigio sono quelle caratterizzate da un'alta visibilità e da una distanza dal target non inferiore ad una certa soglia, come è tipico in un task di sorveglianza ed in blu sono visualizzati i waypoints. Una volta individuati i punti strategici ad alta visibilità del target, per cui è stata considerata anche la regione di incertezza, il veicolo osservatore potrebbe raggiungerli in diversi modi. Uno di questi consiste nell'utilizzo della guida a waypoint con approccio fuzzy, ampiamente descritta e testata nel primo capitolo, in questo modo si avrebbe anche la possibilità di specificare la direzione di approccio al waypoint, e si potrebbe quindi scegliere questa in modo che il veicolo sia rivolto verso il target. L'utilizzo di più waypoints aprirebbe inoltre la strada a uno scenario più complesso, peraltro trattato in questo capitolo, con più veicoli che si interessano dello stesso target.

### 4.3 TASK DI INTERCETTAZIONE

Uno dei possibili campi applicativi di tale studio riguarda l'intercettazione di un target da parte di un veicolo inseguitore. In un task di questo tipo una classica guida di tipo Proportional to Navigation potrebbe essere modificata con l'aggiunta di una accelerazione che tenga conto di una misura di visibilità del target.

#### 4.3.1 La guida Proportional-Navigation

La guida Proportional-Navigation (anche nota come PNG o Pro-Nav) è una legge di guida usata in molteplici ambiti, tra cui il principale è l'intercettazione di target aerei. Essa si basa sul fatto che due veicoli sono in rotta di collisione se la loro congiungente, la line-of-sight (LOS), non cambia direzione. La legge di guida PNG impone che il vettore velocità del veicolo intercettore ruoti ad una velocità proporzionale alla velocità di rotazione della line-of-sight (LOS rate), e nella stessa direzione. Nella Figura 41 è mostrato un classico scenario di intercettazione con PNG, in cui sono riportati i vettori di velocità e accelerazione di veicolo e target, gli angoli relativi alla direzione ( $\gamma_m$  e  $\gamma_t$ ) e la LOS ( $\sigma$ ).

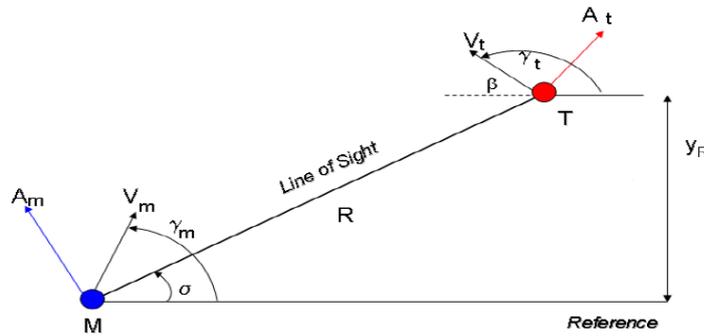


Figura 41.: Scenario PNG

Considerando per il veicolo e per il target il modello dinamico seguente

$$\begin{cases} A_x = A \cos(\gamma + \frac{\pi}{2}) = -A \sin(\gamma) = -u \sin(\gamma) \\ A_y = A \sin(\gamma + \frac{\pi}{2}) = A \cos(\gamma) = u \cos(\gamma) \end{cases} \quad (17)$$

In tale modello l'azione di controllo coincide con il modulo dell'accelerazione laterale ( $u = A$ ). Il principio che sta alla base della guida PNG è il controllo della direzione del vettore velocità, questo si ottiene utilizzando un comando di accelerazione che ha modulo proporzionale alla LOS rate e direzione che, nel caso di PNG di tipo "pure", è normale al vettore velocità.

$$u = A_{PNG} = N V_c \dot{\sigma} \quad (18)$$

In cui  $\dot{\sigma}$  è la LOS rate,  $N$  è una costante, adimensionale generalmente con un valore intero compreso nell'intervallo  $[3, 5]$

$$N = \frac{3}{1 - \frac{c_t}{c_p}} \quad (19)$$

In cui  $c_p$  e  $c_t$  sono le capacità energetiche del veicolo inseguitore e del target,  $N = 3$  corrisponde ad una situazione in cui il target non è manovrato ( $c_t = 0$ ) mentre  $N = 5$  corrisponde ad avere un rapporto  $c_t/c_p = 2/5$ .  $V_c$  è invece la velocità di chiusura (generalmente conosciuta come Closing Velocity)

$$V_c = V_m \cos(\gamma_m - \sigma) - V_t \cos(\gamma_t - \sigma) \quad (20)$$

La legge di guida PNG può essere classificata in base alla direzione del vettore accelerazione, infatti si distingue tra PNG di tipo "pure" se il vettore accelerazione è normale al vettore velocità, PNG di tipo "true" se il vettore accelerazione è normale alla line-of-sight, in questo lavoro si fa sempre riferimento ad una pure-PNG.

#### 4.3.2 La guida Proportional-Navigation con contributo di visibilità

Si vuole, a questo punto, introdurre una modifica all'accelerazione prodotta dalla legge di guida PNG, che tenga conto del livello di visibilità del target descritto nel capitolo precedente. Il problema che ci si pone è il calcolo della strategia di moto di un veicolo che mantenga la visibilità del target mobile in un ambiente strutturato. La strategia utilizzata massimizza la probabilità che il target si trovi in vista, in base ad una conoscenza probabilistica della posizione del target, che può essere interpretata anche come una predizione, entro un certo intervallo temporale, della sua posizione futura. Per tale scopo, è stata introdotta, nel capitolo precedente, una funzione di probabilità che associa ad ogni punto dell'ambiente la probabilità che il target si trovi effettivamente in quel punto, è stata data una definizione del livello di visibilità ed è stato implementato un algoritmo che produce la direzione di moto per cui si ha un incremento della visibilità del target. Una questione di primaria importanza che emerge a questo punto è come tradurre l'informazione fornita dall'algoritmo, che è espressa in termini di un angolo, in un contributo di accelerazione laterale che possa essere sommato all'accelerazione prodotta dalla guida Pro-Nav. Tale traduzione è stata fatta utilizzando l'errore tra l'angolo del vettore velocità ( $\gamma$ ) e l'angolo fornito dall'algoritmo sopra descritto ( $\gamma_{vis}$ ), nel modo seguente.

$$A_{vis} = k_{vis} \cdot \sin(\gamma_{vis} - \gamma) \quad (21)$$

Essendo  $k_{vis}$  un guadagno positivo, se la differenza  $\gamma_{vis} - \gamma$  è tale che:  $0 < (\gamma_{vis} - \gamma) < \pi$  si ha che la accelerazione  $A_{vis} > 0$ , se invece  $\pi < (\gamma_{vis} - \gamma) < 2\pi$  allora  $A_{vis} < 0$ , come mostrato in Figura 42.

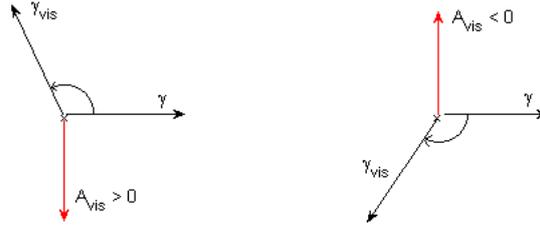


Figura 42.: Accelerazione con  $\gamma$  e  $\gamma_{vis}$

Il guadagno  $k_{vis}$ , come accennato in precedenza, è un guadagno proporzionale al miglioramento di visibilità che si otterrebbe se si andasse nella direzione ( $\gamma_{vis}$ ) indicata dall'algoritmo. Infatti esso è calcolato come la differenza tra la visibilità nella nuova direzione e quella attuale, tale differenza è poi normalizzata. In questo modo si ha che  $k_{vis} \in [0, 1]$ , valori di  $k_{vis}$  prossimi all'unità indicano un miglioramento massimo della visibilità, ovvero seguendo tale direzione si andrebbe da una situazione di visibilità zero del target a una di visibilità massima. Il contributo di accelerazione descritto fin qui, calcolato in base alle direzioni di incremento di visibilità, può essere sommato alla accelerazione della legge PNG.

$$\mathbf{u} = A_{PNG} + K \cdot A_{vis} = N V_c \dot{\sigma} + K \cdot k_{vis} \cdot \sin(\gamma_{vis} - \gamma) \quad (22)$$

In cui  $K > 0$  è un ulteriore guadagno che è utile a portare la componente di visibilità nel range di valori tipici della accelerazione di una PNG, per rendere efficace la somma dei due contributi.

### 4.3.3 Simulazioni

Per testare l'efficacia dell'approccio sopra descritto in un task di intercettazione sono presentate alcune simulazioni in cui un target, per il momento non manovrato, cerca di sfuggire al veicolo inseguitore. Quest'ultimo si muove con l'accelerazione, normale al vettore velocità, riportata in (22) ed entrambi sono caratterizzati dal modello dinamico descritto dalla (17). Nella Figura 43 viene mostrato il risultato ottenuto con la prima simulazione, in particolare viene presentato un confronto tra le traiettorie eseguite. A sinistra è riportata la traiettoria eseguita dal veicolo con legge di guida che coincide esclusivamente con la PNG, a destra invece è riportata la traiettoria eseguita con legge di guida (22).

Il veicolo inseguitore è caratterizzato da traiettoria blu, mentre il target da traiettoria rossa. Nella Figura 43 (b) si nota chiaramente come il veicolo, partendo da una situazione di difficile visibilità, si porti dapprima verso un punto in cui la visibilità del target è pressoché massima, e poi, come ci si aspetta da una guida PNG, si occupi di puntare ed intercettare il target.

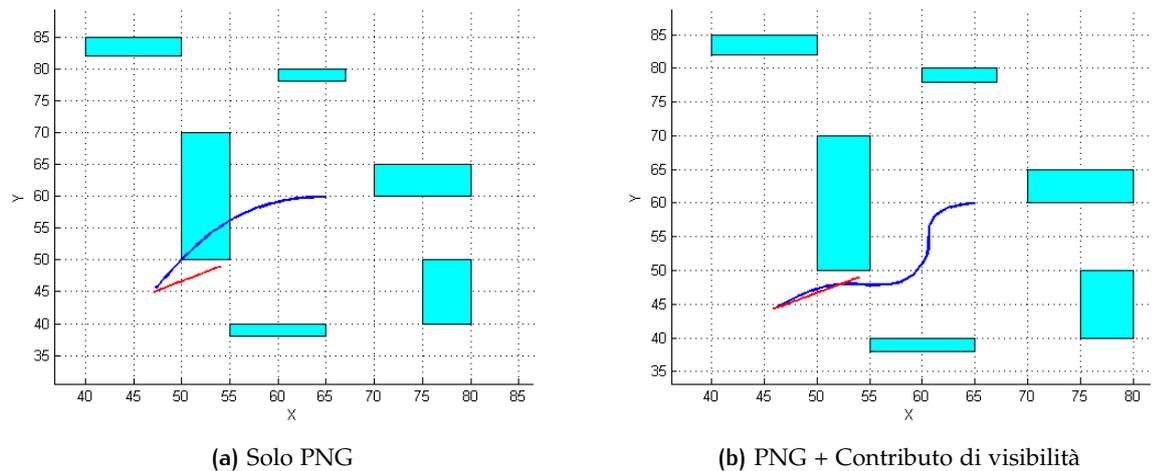


Figura 43.: Intercettazione 1

Ciò si nota anche dalla seconda simulazione, mostrata in Figura 44. Anche in questo caso il contributo di visibilità sull'accelerazione suggerisce al veicolo di aggirare l'ostacolo in una prima fase, e solo in seguito, quando ormai il target è in piena visibilità, viene puntato ed infine intercettato. Per valutare meglio l'efficacia dell'algoritmo di guida proposto è interessante mostrare l'andamento delle accelerazioni durante l'ingaggio, ciò viene fatto in Figura 45.

La Figura 45 mostra l'evoluzione nel tempo sia della accelerazione di PNG che del contributo di visibilità per la seconda simulazione, riportando anche quella effettiva, che è la somma delle due. È interessante notare come nella fase iniziale dell'ingaggio prevalga la componente di visibilità nell'accelerazione totale, in quanto in tale fase si ha ancora una situazione di scarsa visibilità del target. Infatti, il miglioramento apportato dall'algoritmo di visibilità nella

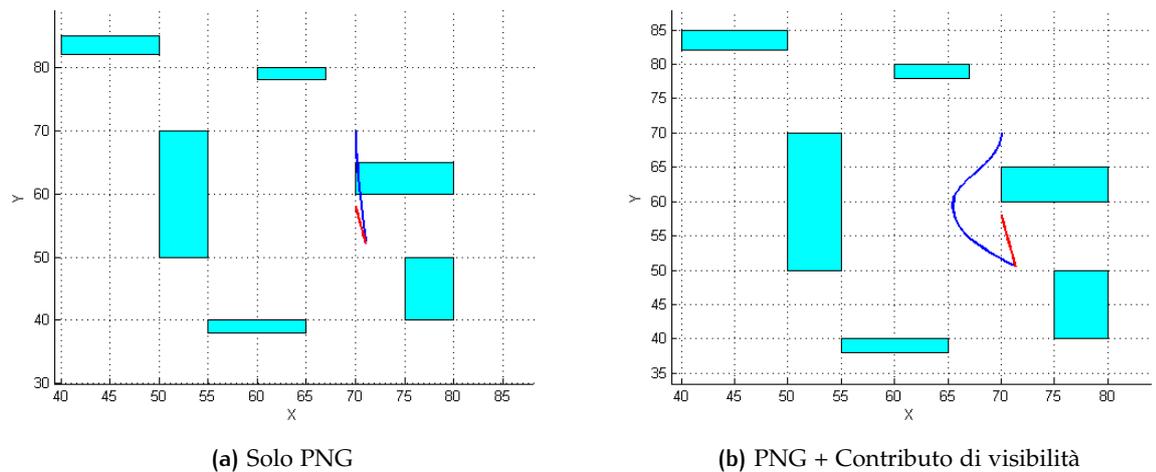


Figura 44.: Intercettazione 2

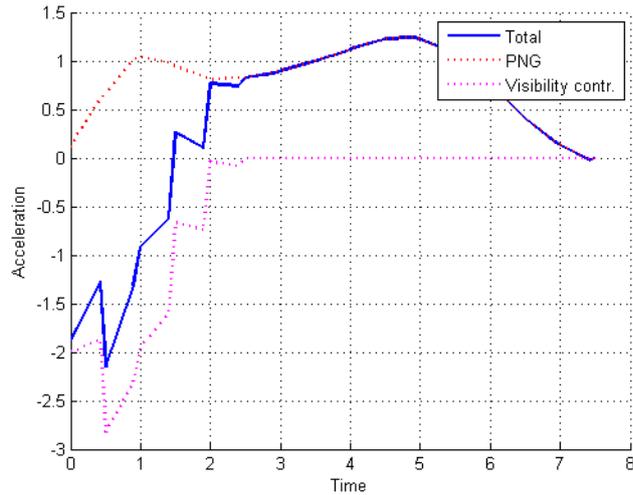
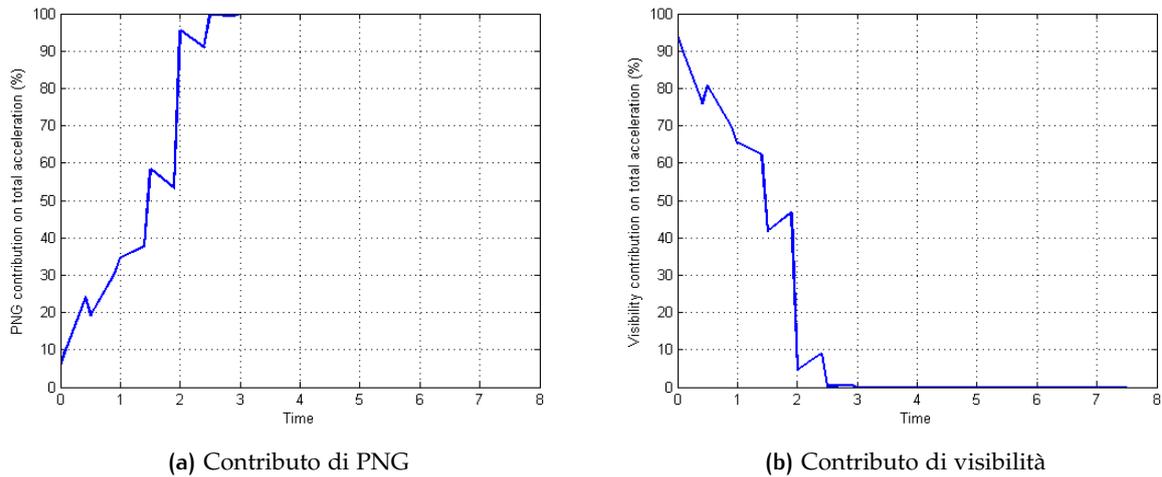


Figura 45.: Accelerazione e singoli contributi



(a) Contributo di PNG

(b) Contributo di visibilità

Figura 46.: Singoli contributi in percentuale

direzione da esso suggerita è notevole, rendendo il guadagno  $k_{vis}$  nell'equazione (22) prossimo a uno. Ciò rende la componente di visibilità prevalente sulla componente della PNG, come è facile notare anche dalla Figura 46.

Nella Figura 46 le due componenti sono espresse in percentuale sulla accelerazione totale, in tal modo è immediata la concezione del peso che queste hanno sull'ingresso del veicolo. Nella fase finale invece risulta prevalente il contributo di PNG, in linea con la traiettoria mostrata nelle immagini precedenti. La traiettoria eseguita con la legge di guida proposta, è determinata dal livello di visibilità del target. E' interessante notare come varia il livello di visibilità durante l'ingaggio, mostrato in Figura 47.

In linea con quanto ci si aspetta, questa misura raggiunge il massimo nella fase finale dell'ingaggio, questa fase coincide con la prevalenza della guida PNG sul contributo di visibilità, ed è la fase in cui il target è ormai in totale visibilità.

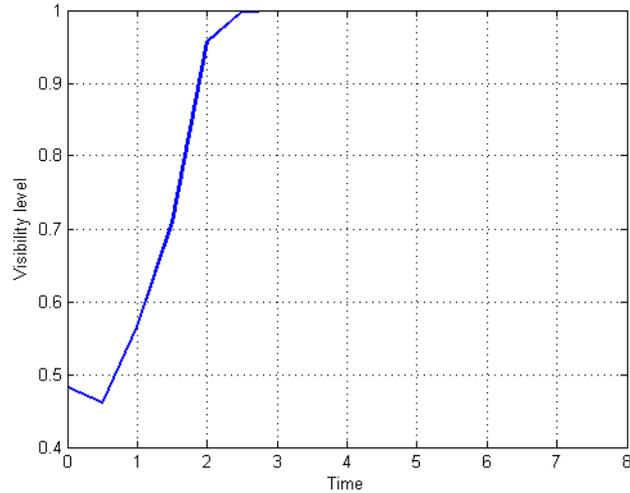


Figura 47.: Livello di visibilità

La Figura 48 invece mostra le traiettorie eseguite dal veicolo al variare del guadagno fisso  $K$  presente nella (22). Si noti come, all'aumentare di  $K$ , il veicolo esegua una traiettoria sempre più volta all'aggiramento dell'ostacolo, per acquisire e poi migliorare la visibilità del target. Infatti un incremento di  $K$  comporta un incremento del peso che la componente di visibilità ha nella accelerazione totale, e questo comporta una prevalenza più netta di tale componente sulla PNG.

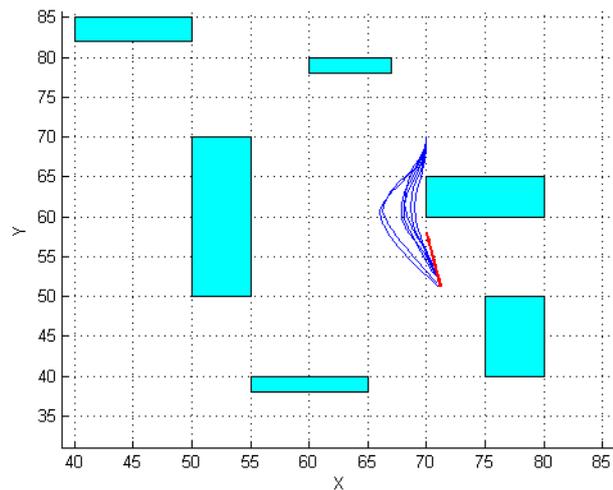


Figura 48.: Traiettoria al variare di  $K$

#### 4.4 TASK DI INTERCETTAZIONE: TARGET MANOVRATO

In questa sezione si estende il task di intercettazione visto nella sezione precedente. Una interessante modifica del task di intercettazione consiste nel fornire "intelligenza" al target, ovvero considerare una situazione in cui il target sia manovrato. In particolare al target si fornisce la capacità di poter sfuggire alla regione di visibilità del veicolo inseguitore, utilizzando un algoritmo di guida duale a quello utilizzato da quest'ultimo. Infatti al target viene adesso fornita una accelerazione laterale, anch'essa normale al vettore velocità. Tale accelerazione sarà però costituita non da due componenti (PNG e contributo di visibilità), come avviene per il veicolo principale, ma dalla sola componente di visibilità. Il target va a ricercare una direzione che comporti un peggioramento della sua visibilità rispetto al veicolo inseguitore, ovvero cerca una direzione che gli permetta di nascondersi sfruttando gli ostacoli. In maniera analoga a quanto visto nei capitoli precedenti, egli utilizza un algoritmo che tiene conto della variazione di probabilità nello spazio e produce una direzione di moto. Il calcolo di tale direzione viene fatto utilizzando il gradiente di  $p_{Tv}$ , sebbene in questo caso venga fatto al variare della posizione del target stesso.

$$\bar{v} = \left. \frac{\partial p_{Tv}}{\partial r} \right|_{\bar{r}=\bar{r}_t} \quad (23)$$

Con  $\bar{r}_t \in X^n$  generico vettore che descrive la posizione del target. Limitandoci alle direzioni in cui tale gradiente è negativo si ottengono le direzioni di decrescita della probabilità. Analogamente a quanto fatto prima, il gradiente viene in realtà calcolato in un preciso intorno della posizione del target, da un punto di vista numerico l'algoritmo viene quindi implementato come segue.

$$\min_r [p_{Tv}(r) - p_{Tv}(r_v)], \quad r \in B(r_t, d) \quad (24)$$

In cui  $B(r_t, d)$  è un intorno circolare,  $B(r, d) = \{r^* \in X^n : \text{dist}(r^*, r) \leq d\}$ . Trovato  $r^* = [x^*, y^*]^T$ , il punto corrispondente al valore minimo nel semplice caso planare, la direzione è poi calcolata come  $\arctan(\frac{y^*}{x^*})$ . Anche in questo caso, l'informazione prodotta dall'algoritmo, espressa in termini di un angolo, deve esser tradotta in una accelerazione laterale che costituisce l'ingresso di controllo per il modello del target, descritto dalla (17). Questo viene fatto in maniera analoga a quanto visto per il veicolo inseguitore.

$$A_{vis}^{targ} = k_{vis} \cdot \sin(\gamma_{vis}^{targ} - \gamma^{targ}) \quad (25)$$

In cui  $\gamma_{vis}^{targ}$  è la direzione di peggioramento di visibilità prodotta dall'algoritmo mentre  $\gamma^{targ}$  è la direzione del vettore velocità del target. Il guadagno  $k_{vis}$ , in maniera duale a quanto avviene per il veicolo inseguitore, è un guadagno proporzionale al peggioramento di visibilità, compreso tra  $[0, 1]$ . Il risultato di

questa estensione sarà una spinta sul target che lo porterà come a nascondersi in prossimità degli ostacoli, in zone in cui risulterà poco visibile dal veicolo inseguitore. Le immagini seguenti mostrano una simulazione in cui questo è visibile.

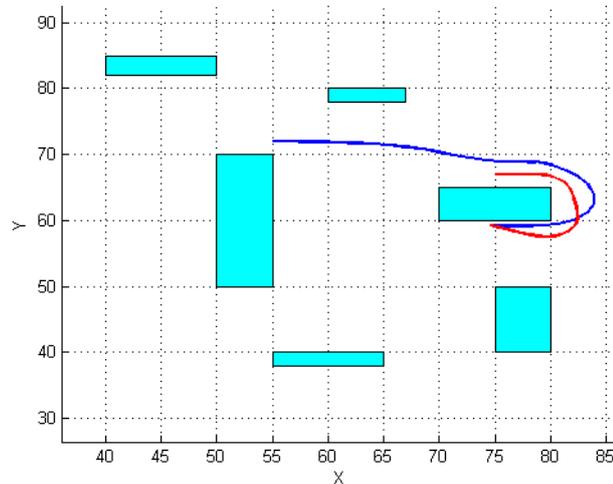


Figura 49.: Traiettorie, target manovrato

In particolare la Figura 49 mostra la traiettoria eseguita da veicolo e target, descritti dalle equazioni (17), in cui è evidente come il target, che parte da posizione (75, 67) e si muove con la legge di guida descritta in (25), provi a nascondersi sfruttando l'ostacolo e cercando di girarvi intorno. La Figura 50 (a) mostra invece il livello di visibilità del target rispetto al veicolo, si noti come questo presenti un minimo nella zona attorno ai 7 sec., intervallo di tempo in cui il target risultava nascosto dietro l'ostacolo. La Figura 50 (b) mostra l'evoluzione nel tempo dei contributi di accelerazione del veicolo inseguitore, è interessante notare come nella fase corrispondente al minimo del livello di visibilità si ha una prevalenza sulla accelerazione della componente di visibilità.

Nella seconda simulazione, di cui la Figura 51 mostra le traiettorie di veicolo e target, è emersa una questione di fondamentale importanza per la buona riuscita del task di intercettazione.

E' emersa infatti una limitazione caratteristica della legge di guida PNG, presente in prevalente misura nella seconda fase degli scenari di intercettazione visti fin'ora. Uno dei limiti principali di una guida classica quale la PNG, è legato al concetto di angles-of-attack [9], che si può riassumere come segue: data una certa posizione e orientazione del target, sono limitate le direzione iniziali del veicolo intercettore che permettono ad esso di concludere l'ingaggio con esito positivo. Questo limite, se non considerato, porta al fallimento di un gran numero di simulazioni. Infatti, in uno scenario come quello relativo a questa sezione in cui il target è manovrato, e cerca, spesso con manovre brusche, di nascondersi sfruttando gli ostacoli, accade che il veicolo inseguitore, appena terminata la fase in cui prevale in contributo di visibilità, si trova in una orientazione tale che la parte di guida PNG non è in grado di puntare il target. Questo

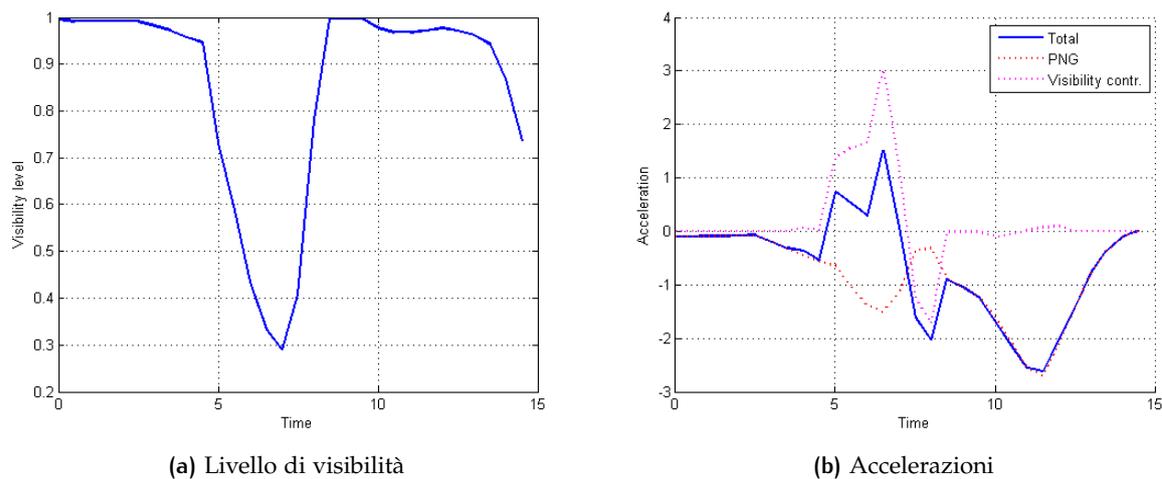


Figura 50.: Target manovrato

è visibile in Figura 51, in cui il punto cerchiato in rosso rappresenta proprio il momento in cui il contributo relativo alla visibilità si affievolisce, in quanto il target comincia ad essere in piena visibilità, e acquista autorità l'accelerazione di PNG. Come è facile notare, in quel momento l'orientazione del veicolo era al di fuori delle direzioni per cui la PNG riesce a puntare il target, e questo ha portato all'inevitabile fallimento dell'ingaggio. Si è pensato quindi ad una modifica della guida PNG introducendo un ulteriore contributo di accelerazione, detto di Heading-off, utile a riportare il veicolo in una configurazione per cui è possibile utilizzare una legge di guida classica di tipo Pro-Nav.

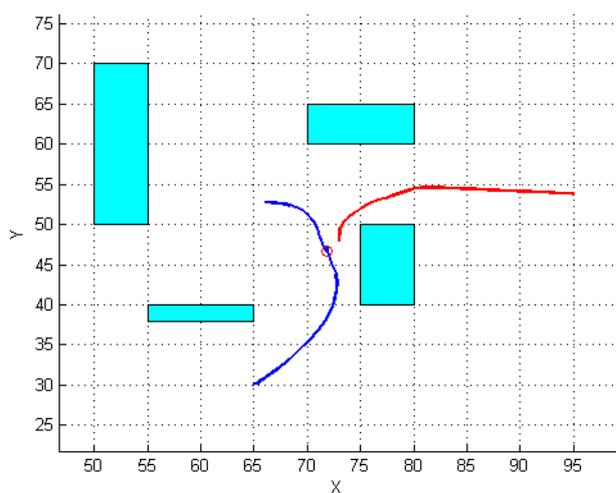


Figura 51.: Traiettorie, target manovrato 2

#### 4.4.1 Considerazioni sull'agilità del veicolo

Nelle simulazioni precedenti in cui, essendo il target guidato da una accelerazione che gli consente di diminuire la sua visibilità sfruttando gli ostacoli, accadeva che il veicolo inseguitore nel momento in cui cominciava a prevalere l'accelerazione di PNG sul contributo di visibilità, ovvero quando il target risultava ormai in piena visibilità, si trovava spesso orientato in maniera non compatibile con i limiti che la PNG ha sulle condizioni iniziali dell'ingaggio. Per ovviare a ciò bisogna ricorrere a delle considerazioni sull'agilità del veicolo inseguitore, questo comporta un'analisi dei suoi vincoli cinematici. Come già accennato nei capitoli precedenti il più importante tra tali vincoli è quello relativo alla massima curvatura supportata dal veicolo ovvero il minimo raggio di sterzata del veicolo stesso. Infatti se il veicolo ha un vincolo sulla curvatura massima, come avviene nel caso di aeroplani o missili, esso non potrà certamente ruotare sul posto per portarsi in condizione di intercettazione, ed allora è necessario pensare ad una azione correttiva per riportare il veicolo in condizione di intercettazione. Se invece il veicolo in questione è ad ali rotanti, come nel caso di un elicottero o un quad-rotor, allora non si hanno vincoli sulla curvatura massima, in quanto il veicolo è in grado di cambiare la sua orientazione senza cambiare posizione. Si è pensato quindi ad un contributo che permettesse anche ai veicoli della prima tipologia, che sono quelli considerati nelle simulazioni effettuate fin'ora ai quali è stata applicata la guida PNG e la guida PNG con contributo di visibilità, di concludere l'ingaggio con esito positivo. Un limite importante della guida proporzionale utilizzata è legato al concetto di angles-of-attack, limite che può essere riassunto come segue: data una certa posizione e orientazione del target sono limitate le direzioni iniziali del veicolo intercettore affinché l'ingaggio abbia esito positivo. Per tale motivo si è ricercata una soluzione che rendesse il sistema in grado di gestire anche angoli di attacco elevati, una delle quali è riportata in [9]. Il concetto che sta dietro l'introduzione di tale contributo è di dare al veicolo la capacità di generare repentine rotazioni del vettore che descrive la sua orientazione, con lo scopo di portarsi in una situazione in cui una classica legge di guida, quale la PNG, può essere applicata con successo. Ciò è ottenuto utilizzando comunque una accelerazione laterale, normale al vettore velocità come input di controllo. In particolare è necessario in primo luogo calcolare il "flight path angle" ideale del veicolo, ovvero l'angolo descritto dal vettore velocità per cui l'intercettazione avrebbe esito positivo.

Tale angolo, per lo scenario mostrato in Figura 52, è dato da:

$$\gamma_{mid} = \sigma + \arcsin \left( \frac{V_o \sin(\gamma_o - \sigma)}{V_m} \right) \quad (26)$$

Se durante l'ingaggio, si ha che  $\gamma_m$  è molto diverso da  $\gamma_{mid}$  allora è necessario un ulteriore contributo propulsivo per fare in modo che il veicolo riacquisti la condizione di intercettazione. Assumendo che le velocità siano costanti,

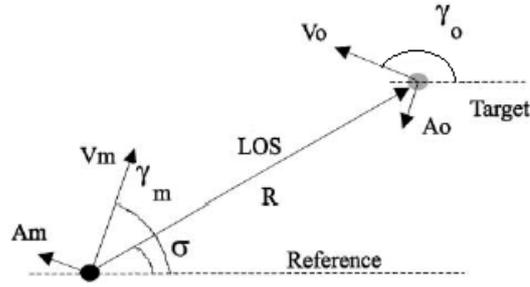


Figura 52.: Scenario Heading-off

con  $V_m > V_o$ , la Figura 53 mostra qualitativamente le situazioni che possono presentarsi.

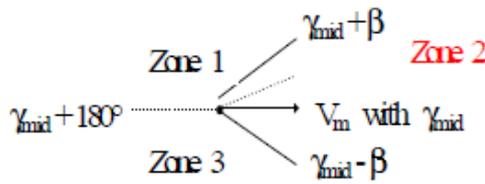


Figura 53.: Zone per Heading-off

Nel caso in cui la direzione del veicolo, descritta dal suo vettore velocità ( $\gamma_m$ ), è all'interno della Zona 2, l'intercettazione può essere portata a termine con una delle classiche leggi di guida. Nel caso in cui il veicolo si trova invece nella Zona 1 o nella Zona 3, una azione correttiva è necessaria per riportare il veicolo all'interno della Zona 2. La scelta della zona di separazione dipende dall'angolo  $\beta$  mostrato in figura, la scelta sul valore di tale angolo è stata fatta secondo quanto proposto in [3].

$$\beta > \arcsin\left(\frac{V_o}{V_m}\right) \quad (27)$$

L'azione correttiva è intesa con lo scopo di ridurre l'errore rispetto alla Line-of-sight, e deve essere espressa in termini di una accelerazione laterale normale al vettore velocità. Infatti tale accelerazione deve sostituire quella di PNG nel caso in cui il veicolo ha una direzione che è al di fuori della Zona 2, ed è calcolata come segue.

$$A_{hoff} = k \sin(\sigma - \gamma_m) \quad (28)$$

Anche in questo caso si utilizza un valore proporzionale al seno dell'errore tra LOS e direzione del veicolo.

$$A_{\text{PNG}} = \begin{cases} NV_c \dot{\sigma}, & \gamma_m \in [\gamma_{\text{mid}} - \beta, \gamma_{\text{mid}} + \beta] \\ A_{\text{h-off}}, & \text{altrimenti} \end{cases} \quad (29)$$

In Figura 54 è riportata una intercettazione in un ambiente aperto, in cui il contributo di heading-off è fondamentale nella fase iniziale. Il target è posizionato dietro il veicolo e orientato in direzione opposta si muove a velocità costante. Il veicolo, nella fase iniziale grazie al contributo di heading-off, si porta in una situazione in cui la guida di tipo PNG conclude con successo l'intercettazione.

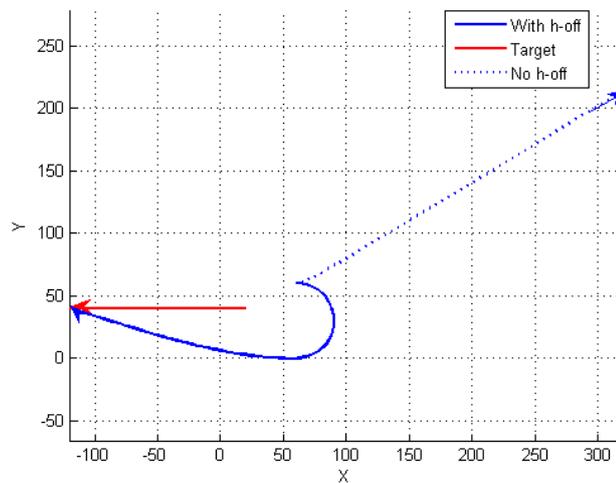


Figura 54.: Esempio Heading-off

E' interessante riproporre a questo punto la simulazione con esito negativo mostrata già in precedenza, con l'aggiunta, per la parte di accelerazione PNG, del contributo di heading-off.

In Figura 55 mostra la traiettoria eseguita dal veicolo e dal target, ed è visibile in tratteggio la traiettoria che verrebbe percorsa dal veicolo durante l'ingaggio se non ci fosse il contributo di heading-off. E' evidente come nel momento in cui la PNG acquista pieno potere con il target in totale visibilità, tale contributo sia fondamentale per riportare il veicolo in condizione di intercettazione. In Figura 56 si riportano inoltre il livello di visibilità del target e le accelerazioni ottenute. Il livello di visibilità, come previsto, presenta un minimo corrispondente all'intervallo di tempo in cui il target risulta nascosto dietro l'ostacolo, che coincide con [3, 7] sec. circa. Dal grafico delle accelerazioni si nota infatti che in tale intervallo vi è una prevalenza del contributo di visibilità sull'accelerazione di PNG. Dopo tale intervallo il target è in totale visibilità e l'accelerazione di PNG acquista maggior peso, tuttavia in una prima fase è proprio il contributo di heading-off a prevalere.

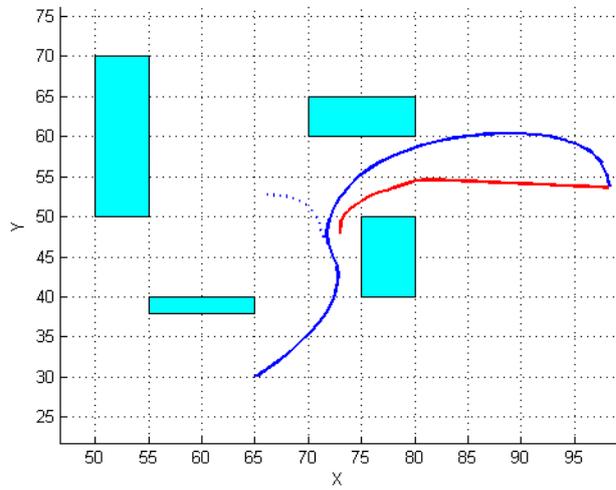
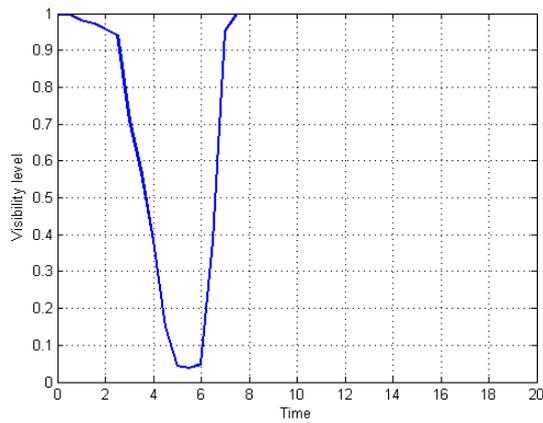
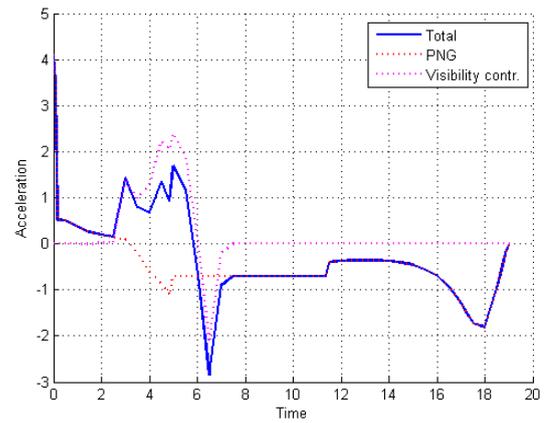


Figura 55.: Intercettazione con Heading-off



(a) Livello di visibilità



(b) Accelerazioni

Figura 56.: Target manovrato 2

#### 4.5 TASK DI INTERCETTAZIONE: CONSIDERAZIONI ANALITICHE

Si vuole, a questo punto, dare una impostazione teorica al problema dell'intercettazione di un target (evader) da parte di un veicolo inseguitore (pursuer). L'obiettivo di questa sezione è quindi quello di fornire all'algoritmo descritto in precedenza e testato nelle diverse simulazioni, che aggiunge un contributo di visibilità all'accelerazione tipica della legge di guida PNG, una solida giustificazione teorica. Come descritto in [1], un task di intercettazione tra veicolo e target può essere considerato come un gioco differenziale. Nella teoria dei giochi, i giochi differenziali sono un gruppo di problemi che interessano la modellazione e l'analisi di una situazione di conflitto, nel contesto di un sistema dinamico. Il problema solitamente consiste in due giocatori, un pursuer e un evader, con degli obiettivi non compatibili e contrastanti, le cui dinamiche sono modellate da sistemi di equazioni differenziali. I giochi differenziali sono dei problemi per alcuni aspetti molto simili a quelli di controllo ottimo. In un problema di controllo ottimo c'è un singolo giocatore, un singolo controllo ed un unico criterio da ottimizzare; la teoria dei giochi differenziali generalizza questo concetto considerando due giocatori, due controlli e due criteri. Ogni giocatore cercherà infatti di controllare lo stato del sistema per raggiungere i propri obiettivi ed il sistema risponderà agli input di entrambi i giocatori. In generale, un gioco differenziale consiste nel trovare un punto di sella per un funzionale di costo così fatto

$$J = \beta(x(T), T) + \int_{t_0}^T L(x, u, v, t) dt \quad (30)$$

Soggetto ai vincoli

$$\dot{x} = f(x, u, v, t) \quad x(t_0) = x_0 \quad (31)$$

In cui  $x$  è lo stato del gioco,  $u$  e  $v$  sono dette strategie e sono ristrette ad un certo insieme di strategie ammissibili. Un punto di sella è definito come una coppia  $(u^*, v^*)$  che soddisfi la relazione

$$J(u^*, v) \leq J(u^*, v^*) \leq J(u, v^*) \quad (32)$$

Tale relazione deve essere soddisfatta per scelte arbitrarie di  $u$  e  $v$ , se ciò avviene le strategie  $u^*$  e  $v^*$  sono dette strategie ottime. Se si considera un gioco tra target ed intercettore (pursuit-evasion game), il problema si riconduce alla determinazione di un punto di sella  $(u, v)$  per un funzionale di costo fatto come segue.

$$J = \frac{a^2}{2} \|x_p(T) - x_e(T)\|_{A^T A}^2 + \frac{1}{2} \int_{t_0}^T \left[ \|u(t)\|_{R_p(t)}^2 - \|v(t)\|_{R_e(t)}^2 \right] dt \quad (33)$$

Soggetto ai vincoli

$$\dot{x}_p = F_p(t)x_p + \bar{G}_p(t)u \quad (34)$$

$$\dot{x}_e = F_e(t)x_e + \bar{G}_e(t)v \quad (35)$$

Con

$$u(t), v(t) \in \mathbb{R}^m \quad (36)$$

In cui  $x_p$  è un vettore di dimensione  $n$  che descrive lo stato del pursuer,  $u(t)$  è un vettore di dimensione  $m$  che rappresenta il suo controllo,  $F_p(t)$  e  $\bar{G}_p(t)$  sono matrici di dimensione  $n \times n$  e  $n \times m$ , rispettivamente. Analogamente  $x_e$  è un vettore di dimensione  $n$  che descrive lo stato del evader,  $v(t)$  è un vettore di dimensione  $m$  rappresentante il suo controllo.  $R_p(t)$  ed  $R_e(t)$  sono matrici  $m \times m$  definite positive che pesano le azioni di controllo. La matrice  $A$  è definita come  $A = [I_k, 0]$ , dove  $I_k$  è una matrice identità di dimensione  $k$ , mentre la quantità positiva  $a^2$  viene introdotta per poter pesare l'energia di fine ingaggio.  $T$  rappresenta invece l'istante di fine ingaggio, infatti il gioco differenziale è di durata finita.

A questo punto, seguendo quanto fatto in [1], è possibile riconsiderare e riformulare il problema in termini di un vettore  $k$ -dimensionale, con  $1 \leq k \leq n$ , definito come segue.

$$z(t) = A \left[ \Phi_p(T, t)x_p(t) - \Phi_e(T, t)x_e(t) \right] \quad (37)$$

In cui  $\Phi_p$  e  $\Phi_e$  sono matrici di risposta impulsiva per i sistemi pursuer e evader rispettivamente.

$$\Phi_p(T, t) = e^{\int_t^T F_p(\tau) d\tau} \quad \Phi_e(T, t) = e^{\int_t^T F_e(\tau) d\tau} \quad (38)$$

Può essere riformulato un problema completamente equivalente utilizzando  $z(t)$ , e cambiando quindi lo stato del gioco, il problema diventa determinare un punto di sella per il seguente funzionale.

$$J = \frac{a^2}{2} \|z(T)\|^2 + \frac{1}{2} \int_{t_0}^T \left[ \|u(t)\|_{R_p(t)}^2 - \|v(t)\|_{R_e(t)}^2 \right] dt \quad (39)$$

Soggetto al vincolo

$$\dot{z} = G_p(T, t)u - G_e(T, t)v \quad (40)$$

Con

$$G_p = A\Phi_p(T, t)\bar{G}_p(t) \quad G_e = A\Phi_e(T, t)\bar{G}_e(t) \quad (41)$$

Tale trasformazione è inoltre facilmente reversibile. Per poter applicare la procedura standard di calcolo variazionale, utilizzata per i problemi di controllo ottimo, è necessario definire la funzione hamiltoniana.

$$H(\lambda, z, u, v, t) = \frac{1}{2}(\|u\|_{R_p}^2 - \|v\|_{R_e}^2) + \lambda^T(G_p u - G_e v) \quad (42)$$

Dal calcolo variazionale si ottengono le seguenti relazioni, conosciute come equazioni di Eulero-Lagrange.

$$\frac{\partial H(\lambda, z, u, v, t)}{\partial u} = 0 \quad \Rightarrow \quad u = -R_p^{-1}G_p^T \lambda(t) \quad (43)$$

$$\frac{\partial H(\lambda, z, u, v, t)}{\partial v} = 0 \quad \Rightarrow \quad v = -R_e^{-1}G_e^T \lambda(t) \quad (44)$$

$$\dot{z} = \frac{\partial H(\lambda, z, u, v, t)}{\partial \lambda} = G_p u - G_e v \quad (45)$$

$$\dot{\lambda} = -\frac{\partial H(\lambda, z, u, v, t)}{\partial z} = 0 \quad (46)$$

Essendo noto l'istante di fine ingaggio si ottiene la seguente condizione di trasversalità.

$$\lambda(T) = \frac{\partial \beta(z(T), T)}{\partial z(T)} = \alpha^2 z(T) \quad (47)$$

Integrando la (46) e sostituendo in (43) e in (44), si ottengono gli ingressi ottimi per pursuer e evader.

$$u(t) = -R_p^{-1}G_p^T K^{-1}(T, t)z(t) \quad (48)$$

$$v(t) = -R_e^{-1}G_e^T K^{-1}(T, t)z(t) \quad (49)$$

Ottenuti attraverso una reazione dello stato tramite  $K(T, t)$ , tale matrice è così definita.

$$K(T, t) = \left[ \frac{I_R}{a^2} + M_p(T, t) - M_e(T, t) \right] \quad (50)$$

Con

$$M_p(t, t_0) = \int_{t_0}^t G_p(T, t') R_p^{-1} G_p^T(T, t') dt' \quad (51)$$

Ed  $M_e$  è definita in maniera equivalente. Una condizione sufficiente per la verifica dell'esistenza del punto di sella utilizza la derivata seconda della funzione hamiltoniana rispetto ai controlli  $u$  e  $v$ . Tale condizione risulta essere verificata, ed è riportata di seguito.

$$\frac{\partial^2 H(\lambda, z, u, v, t)}{\partial u^2} = R_p > 0 \quad \frac{\partial^2 H(\lambda, z, u, v, t)}{\partial v^2} = -R_e < 0 \quad (52)$$

Il problema descritto e risolto fin'ora ha tuttavia una caratterizzazione molto generica, da questo punto a seguire verrà specificata la dinamica di pursuer ed evader, andando quindi a trattare un caso particolare della classe di problemi descritti fin qui. Per descrivere la dinamica del veicolo inseguitore e del target si utilizzano le equazioni riportate in (34) e (35), scegliendo un vettore di stato contenente posizione e velocità.

$$x_p = \begin{pmatrix} r_p \\ v_p \end{pmatrix} \quad x_e = \begin{pmatrix} r_e \\ v_e \end{pmatrix} \quad (53)$$

Con le seguenti matrici di stato.

$$F_p = F_e = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \bar{G}_p = \bar{G}_e = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (54)$$

In questo modo, si è scelto di avere  $n = 2$  ed  $m = 1$ , ciò comporta che posizione e velocità sono descritti in uno spazio a dimensione uno. Tale scelta è dettata dall'esigenza di semplificare i calcoli che verranno fatti in seguito, ma ne deriva una rappresentazione poco fedele della realtà, in quanto si dovrebbe avere posizione e velocità in uno spazio tri-dimensionale. Tuttavia i risultati ottenuti con questa descrizione sono perfettamente compatibili con quanto si otterrebbe con  $n = 6$ , sostituendo il valore 1 presente in  $F_p$  e  $F_e$  con l'identità  $I_3$ . A questo punto, lo stato del gioco descritto dal vettore  $k$ -dimensionale  $z(t)$  riportato in (37), essendo  $k = 1$ , diventa il seguente.

$$z(t) = (1 \ 0) \left[ \begin{pmatrix} 1 & T-t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} r_p \\ v_p \end{pmatrix} - \begin{pmatrix} 1 & T-t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} r_e \\ v_e \end{pmatrix} \right] \quad (55)$$

$$= (r_p - r_e) + (v_p - v_e)(T - t) \quad (56)$$

Considerando  $R_p(t) = \frac{1}{c_p}$  ed  $R_e(t) = \frac{1}{c_e}$ , in cui  $c_p$  e  $c_e$  rappresentano le capacità energetiche di veicolo e target, l'indice di costo e la funzione hamiltoniana, riportati in (39) e in (42), possono essere riscritti come segue.

$$J = \frac{a^2}{2} [r_p(T) - r_e(T)] [r_p(T) - r_e(T)]^T + \frac{1}{2} \int_0^T \left[ \frac{a_p a_p^T}{c_p} - \frac{a_e a_e^T}{c_e} \right] dt \quad (57)$$

$$H = \frac{1}{2} \left[ \frac{a_p a_p^T}{c_p} - \frac{a_e a_e^T}{c_e} \right] + \lambda(T-t)[a_p - a_e] \quad (58)$$

In quanto  $G_p$  e  $G_e$ , definite in (41), sono entrambe pari a  $(T-t)$ . Gli ingressi ottimi diventano i seguenti.

$$a_p = -c_p(T-t)K^{-1}(T,t)z(t) \quad (59)$$

$$a_e = -c_e(T-t)K^{-1}(T,t)z(t) = \frac{c_e}{c_p} a_p \quad (60)$$

In cui  $K$ , che in (50) è riportato in forma generica, assume nello specifico la seguente forma.

$$K(T,t) = \frac{1}{a^2} + \frac{1}{3}(T-t)^3(c_p - c_e) \quad (61)$$

Ottenendo in questo modo l'espressione completa dell'accelerazione ottima del pursuer.

$$a_p = \frac{-c_p(T-t)[r_p(t) - r_e(t) + (v_p(t) - v_e(t))(T-t)]}{\frac{1}{a^2} + \frac{(c_p - c_e)(T-t)^3}{3}} \quad (62)$$

Si evince che l'intercettazione è possibile solo se  $c_p > c_e$ , inoltre considerando il caso limite in cui  $a^2 \rightarrow \infty$ , e ricavando la costante di navigazione:

$$N = \frac{3}{1 - \frac{c_e}{c_p}} \quad (63)$$

Si ottiene la classica espressione della legge di guida PNG.

$$\mathbf{a}_p = NV_c \dot{\sigma} \quad (64)$$

A questo punto si vuole modificare il problema affinché tenga conto anche della probabilità che il pursuer abbia il target in vista, così da fornire all'algoritmo descritto in precedenza e testato nelle diverse simulazioni, che aggiunge un contributo di visibilità all'accelerazione di PNG, una giustificazione teorica. Per far ciò in primo luogo si è aumentato lo stato del gioco  $z(t)$ , che diventa di dimensione due e risulta essere così fatto.

$$\mathbf{x}(t) = \begin{pmatrix} z(t) \\ \dot{\mathbf{p}}_{Tv}(r_p, v_p, r_e, v_e) \end{pmatrix} \quad (65)$$

Si è creato un nuovo stato del gioco  $\mathbf{x}(t)$  introducendo la derivata rispetto al tempo del livello di visibilità del target, definito in (13), che è funzione della posizione e della velocità di pursuer ed evader, e utilizzando lo stesso  $z(t)$  descritto in precedenza. Di conseguenza si è riscritto il funzionale di costo (57), con l'introduzione di un termine che tenga conto del livello di visibilità.

$$J = \frac{\mathbf{a}^2}{2} \left[ \mathbf{r}_p(T) - \mathbf{r}_e(T) \right] \left[ \mathbf{r}_p(T) - \mathbf{r}_e(T) \right]^T + \frac{1}{2} \int_0^T \left[ \frac{\mathbf{a}_p \mathbf{a}_p^T}{c_p} - \frac{\mathbf{a}_e \mathbf{a}_e^T}{c_e} \right] dt - \bar{K} \int_0^T \dot{\mathbf{p}}_{Tv} dt \quad (66)$$

In particolare, anche per il nuovo funzionale di costo, si è scelto di utilizzare un termine proporzionale, attraverso la costante  $\bar{K}$ , alla derivata rispetto al tempo del livello di visibilità del target. In tal modo assumendo di partire da una condizione di buona visibilità del target si avrà che il veicolo intercettore giocherà per mantenere bassa tale variazione mentre il target cercherà di aumentarla. L'ipotesi di buona visibilità all'inizio dell'ingaggio è peraltro un'ipotesi molto plausibile. Infatti tale ipotesi è necessaria al veicolo intercettore per decidere una direzione iniziale, in attesa di manovre da parte del target che lo portino a nascondersi sfruttando gli ostacoli e di contro l'algoritmo lato pursuer cercherà di riportarlo in vista. Tale ipotesi è stata infatti considerata in tutte le simulazioni riportate in precedenza. La funzione hamiltoniana viene ridefinita come segue.

$$H = \frac{1}{2} \left[ \frac{\mathbf{a}_p \mathbf{a}_p^T}{c_p} - \frac{\mathbf{a}_e \mathbf{a}_e^T}{c_e} \right] - \bar{K} \dot{\mathbf{p}}_{Tv} + \lambda_1 (T - t) [\mathbf{a}_p - \mathbf{a}_e] + \lambda_2 \ddot{\mathbf{p}}_{Tv} \quad (67)$$

Con

$$\ddot{p}_{Tv} = \frac{d}{dt} \left( \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_p(t)} \dot{r}_p + \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_e(t)} \dot{r}_e \right) \quad (68)$$

$$\ddot{p}_{Tv} = \ddot{r}_p \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_p(t)} + \frac{d}{dt} \left( \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_p(t)} \right) \dot{r}_p + \quad (69)$$

$$+ \ddot{r}_e \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_e(t)} + \frac{d}{dt} \left( \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_e(t)} \right) \dot{r}_e \quad (70)$$

Le equazioni di Eulero-Lagrange diventano le seguenti.

$$\frac{\partial H(\lambda, x, u, v, t)}{\partial a_p} = \frac{a_p}{c_p} + \lambda_1(T-t) + \lambda_2 \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_p(t)} \quad (71)$$

$$\frac{\partial H(\lambda, x, u, v, t)}{\partial a_e} = -\frac{a_e}{c_e} - \lambda_1(T-t) + \lambda_2 \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_e(t)} \quad (72)$$

$$\dot{\lambda} = -\frac{\partial H(\lambda, x, u, v, t)}{\partial x} = \left( -\frac{\partial H(\lambda, x, u, v, t)}{\partial z} \quad -\frac{\partial H(\lambda, x, u, v, t)}{\partial \dot{p}_{Tv}} \right) = (0 \quad +\bar{K}) \quad (73)$$

Nel caso in questione, viene considerato noto l'istante di fine ingaggio  $T$ , e si ottiene la seguente condizione di trasversalità.

$$\lambda(T) = \frac{\partial \beta(x(T), T)}{\partial x(T)} = (a^2 z(T) \quad 0) \quad (74)$$

Per ricavare le leggi di controllo ottime per veicolo e target è sufficiente annullare le derivate della funzione Hamiltoniana (71) (72), ottenendo le espressioni delle accelerazioni.

$$\frac{\partial H(\lambda, x, u, v, t)}{\partial a_p} = 0 \Rightarrow a_p = -c_p \left( \lambda_1(T-t) + \lambda_2 \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_p(t)} \right) \quad (75)$$

$$\frac{\partial H(\lambda, x, u, v, t)}{\partial a_e} = 0 \Rightarrow a_e = -c_e \left( \lambda_1(T-t) - \lambda_2 \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_e(t)} \right) \quad (76)$$

A questo punto è necessario calcolare il valore dei moltiplicatori di Lagrange  $\lambda_1(t)$  e  $\lambda_2(t)$ . In particolare  $\lambda_1$  è lo stesso ricavato in precedenza per il gioco differenziale relativo alla sola legge PNG.

$$\lambda_1 = K^{-1}(T, t)z(t) \quad (77)$$

Con  $K(T, t)$  riportato in (61). Per calcolare  $\lambda_2(t)$  bisogna procedere integrando all'indietro, partendo dall'istante  $T$  e da  $\lambda_2(T)$ , ottenendo:

$$\lambda_2(t) = -\bar{K}(T-t) \quad (78)$$

L'espressione finale dell'accelerazione di pursuer ed evader risulta essere la seguente.

$$a_p = -c_p \left( K^{-1}(T,t)z(t)(T-t) - \bar{K} \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_p(t)} (T-t) \right) \quad (79)$$

$$a_e = -c_e \left( K^{-1}(T,t)z(t)(T-t) + \bar{K} \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_e(t)} (T-t) \right) \quad (80)$$

La condizione sufficiente per la verifica dell'esistenza del punto di sella, che utilizza la derivata seconda della funzione hamiltoniana rispetto ai controlli  $u$  e  $v$ , risulta anche in questo caso essere verificata, ed è riportata di seguito.

$$\frac{\partial^2 H(\lambda, x, u, v, t)}{\partial a_p^2} = \frac{1}{c_p} > 0 \quad \frac{\partial^2 H(\lambda, x, u, v, t)}{\partial a_e^2} = -\frac{1}{c_e} < 0 \quad (81)$$

Come ci si aspettava, il contributo di visibilità all'accelerazione sia del veicolo che del target è dato da un termine proporzionale alla variazione di probabilità rispetto alla variazione di posizione, il che è compatibile con l'algoritmo utilizzato nelle simulazioni.

$$c_p \bar{K} \frac{\partial p_{Tv}(r_p(t), r_e(t))}{\partial r_p(t)} (T-t) \quad (82)$$

Inoltre è presente il time-to-go, ciò indica che tale contributo si attenua nella parte finale dell'ingaggio compatibilmente con quanto visto nelle simulazioni in cui, nelle fasi finali, si notava una prevalenza del solo contributo di PNG.

#### 4.6 TASK DI INTERCETTAZIONE: MOLTEPLICI VEICOLI INSEGUITORI

L'algoritmo proposto, applicato ad un task di intercettazione, è stato ampiamente testato nelle sezioni precedenti dapprima considerando un target non manovrato, ovvero capace di muoversi solo di moto rettilineo uniforme, e in seguito considerando un target a cui veniva fornita, da un sistema di guida, una accelerazione laterale che gli permettesse di fuggire dall'inseguitore, cercando di diminuire la probabilità di essere visto. In questa sezione si vuole quindi complicare ulteriormente lo scenario prendendo in considerazione più veicoli inseguitori e un solo target, lasciando come in precedenza a quest'ultimo la capacità di nascondersi. I veicoli inseguitori sono pilotati dalla legge

di guida descritta in (22), comprendente sia l'accelerazione di PNG sia l'accelerazione dovuta al contributo di visibilità. In tale contesto è emersa l'esigenza di una coordinazione, seppur minima e limitata alla fase iniziale dell'ingaggio, tra i veicoli inseguitori. Se così non fosse, il vantaggio dovuto all'utilizzo di molteplici inseguitori verrebbe vanificato dal fatto che questi percorrerebbero la stessa traiettoria se partissero da configurazioni iniziali simili, essendo pilotati dalla stessa legge di guida e dovendo intercettare lo stesso target. Tale coordinazione è perciò riferita alla scelta di una differente regione di approccio all'ambiente strutturato in cui è presente il target, per ciascun veicolo inseguitore. In tal modo l'algoritmo proposto guiderebbe i veicoli attraverso traiettorie differenti e questi potrebbero avvicinarsi al target da punti differenti, sempre nel rispetto dei vincoli di visibilità, allo scopo di aumentare le probabilità che il target sia in vista da almeno uno di loro. Ciò aumenterebbe di conseguenza la probabilità che l'ingaggio abbia esito positivo.

#### 4.6.1 Differenziazione delle zone di approccio

La scelta di differenziare le zone di approccio permette ai veicoli inseguitori, anche partendo da condizioni iniziali simili, di percorrere traiettorie diverse e attaccare il target da punti differenti dell'ambiente strutturato, cercando di mantenere sempre alto il livello di visibilità. Questa differenziazione è basata su degli algoritmi di Clustering.

Per Clustering si intende un insieme di tecniche di analisi multivariata di un set di dati volte alla ricerca e al raggruppamento di elementi omogenei, ovvero con caratteristiche simili. Le tecniche di Clustering sono basate sulla definizione di una misura di somiglianza tra gli elementi, solitamente tale somiglianza è concepita in termini di distanza in uno spazio multidimensionale. La bontà delle analisi ottenute dagli algoritmi di Clustering dipende molto dalla scelta della metrica, e quindi da come la distanza viene calcolata. Gli algoritmi di Clustering raggruppano gli elementi sulla base della loro distanza reciproca, e quindi l'appartenenza o meno ad un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme stesso.

La procedura che calcola le differenti zone di approccio utilizza in una prima fase il livello di visibilità in ogni punto dell'ambiente, tale visibilità viene pesata con la distanza dal target. A questa grandezza, mostrata in Figura 57, viene applicato un primo Clustering utilizzando l'algoritmo K-means.

La procedura K-means permette, dato un insieme di  $X = \{X_1, \dots, X_N\}$  di elementi, di trovare una partizione, ovvero una famiglia di clusters, che minimizza un indice di costo. Una partizione è definita come  $P = \{A_1, \dots, A_C\}$  t.c.

$$\bigcup_{i=1}^C A_i = X \quad A_i \cap A_j = \emptyset, \forall i \neq j \quad \emptyset \subset A_i \subset X, \forall i \quad (83)$$

La procedura può essere brevemente descritta come segue:

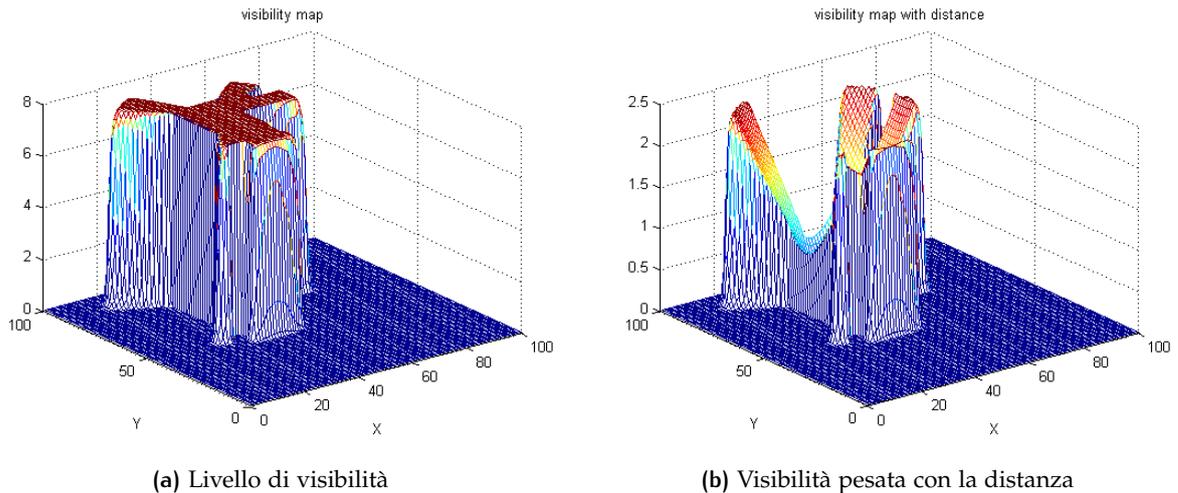


Figura 57.: Visibilità per un target in (42, 73)

- Generare in maniera casuale  $C$  centroidi o, equivalentemente, una partizione iniziale costituita da  $C$  cluster
- Assegnare ogni elemento al cluster il cui centroide è più vicino
- Se non si verifica nessun cambiamento nelle appartenenze degli oggetti ai clusters, terminare l'algoritmo; altrimenti calcolare i centroidi per i nuovi  $C$  cluster e tornare al punto precedente

La scelta di pesare il livello di visibilità in ogni punto con la distanza, viene fatta per escludere le zone a ridosso del target. Infatti l'obiettivo di questo primo Clustering è di individuare delle zone di approccio in cui la visibilità sia alta ma che si trovino ad una certa distanza dal target, in quanto zone d'approccio. Operando un Clustering con l'algoritmo K-means del livello di visibilità pesato con la distanza, che è associato a tutti i punti dell'ambiente, imponendo un numero di clusters pari a 4, si ottengono le zone mostrare in Figura 58.

In Figura 58 viene riportato sia lo scenario con le linee di visibilità del target (a), sia l'immagine dello stesso scenario con delle curve di livello (b), in cui ciascuna di queste delimita un cluster. Come si può notare i punti di uno stesso cluster sono disposti in modo concentrico, questo evidentemente è dovuto alla presenza in ogni punto del valore di distanza dal target. Nell'esempio riportato in figura si è selezionato tra questi 4 clusters quello più lontano dal target (in grigio), in realtà questa scelta può esser fatta basandosi sulla distanza dei veicoli dal target. Scegliendo quello più esterno si è ipotizzato che i veicoli siano abbastanza lontani, infatti sarebbe opportuno scegliere un cluster più interno se i veicoli si trovassero già nella zona in grigio, o comunque nelle vicinanze di questa. A questo punto, prendendo in considerazione solo i punti appartenenti al cluster scelto, viene fatto un secondo Clustering, il cui obiettivo è di creare  $N$  zone di approccio, dove  $N$  sono i veicoli inseguitori partecipanti all'ingaggio.

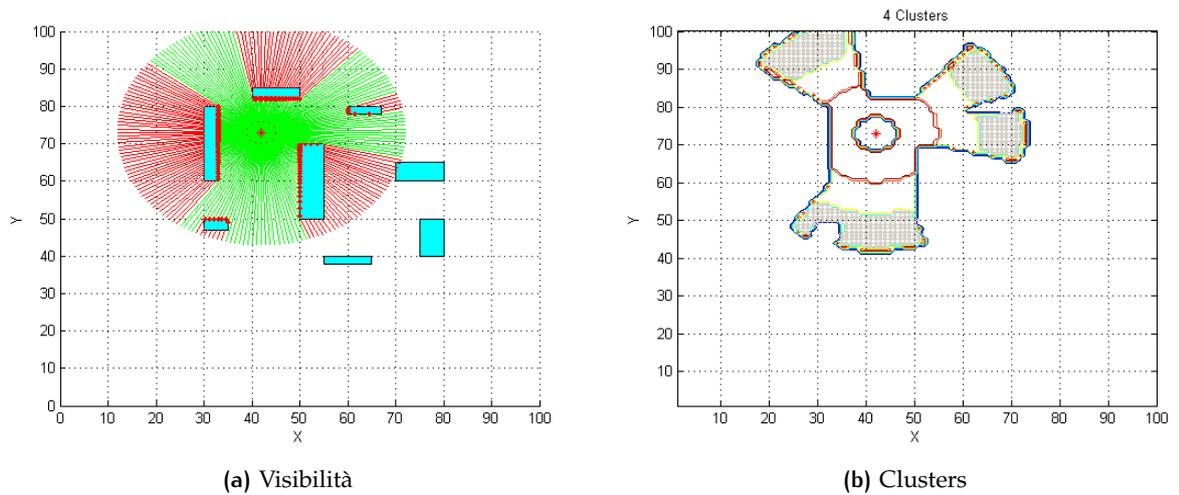


Figura 58.: Primo Clustering

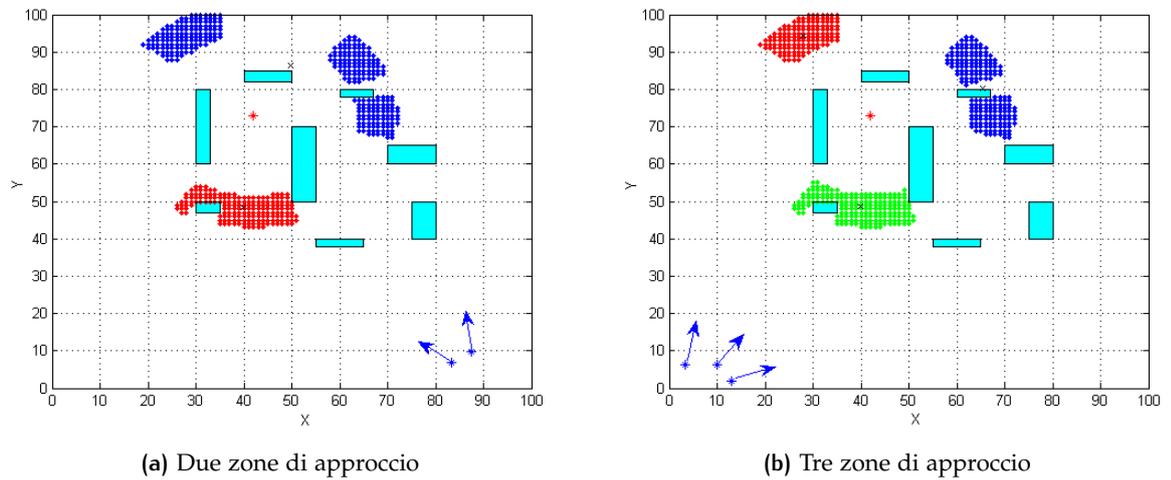


Figura 59.: Secondo Clustering

Questo secondo Clustering è, a differenza del primo, di tipo spaziale, ovvero i punti vengono raggruppati in base alla loro distanza nell'ambiente.

In Figura 59 sono riportate delle immagini in cui viene mostrato il risultato di questo secondo Clustering, che è applicato ai punti appartenenti ad uno solo dei clusters precedenti. I clusters risultanti sono evidenziati con colori diversi. In Figura 59 (a) vengono considerati  $N = 2$  veicoli inseguitori, vengono quindi create due zone di approccio distinte, mentre in Figura 59 (b) vengono considerati  $N = 3$  veicoli inseguitori, vengono quindi create tre zone di approccio distinte. In questo modo, si formano  $N$  regioni ben distinte tra loro, caratterizzate comunque da alta visibilità, in cui gli  $N$  veicoli possono dapprima portarsi, utilizzando per esempio una guida LOS oppure a Waypoints, e solo in seguito attivare la guida descritta in precedenza che permetta loro di intercettare il target.

#### 4.6.2 Simulazioni: 2 veicoli inseguitori vs. 1 target

In questa sezione sono riportate delle simulazioni in cui due veicoli cercano di intercettare un target, che è manovrato, ed è pilotato dalla legge di guida descritta in precedenza che lo rende capace di nascondersi sfruttando gli ostacoli. I due veicoli sono pilotati dalla legge di guida descritta in (22), comprendente sia l'accelerazione di PNG sia l'accelerazione dovuta al contributo di visibilità. In queste simulazioni i veicoli intercettatori hanno inizialmente delle configurazioni molto simili, in quanto si trovano vicini e orientati nella medesima direzione, quindi una coordinazione risulta indispensabile per ottenere delle traiettorie diverse e avvicinare, seppur con la stessa legge di guida, il target da zone differenti, incrementando quindi la probabilità che l'ingaggio si concluda con successo. La coordinazione è ottenuta in termini di una differenziazione delle zone di approccio al target, ed è basata sull'applicazione di due Clustering consecutivi. Il primo Clustering viene fatto sul livello di visibilità di ciascun punto dell'ambiente, pesato con la distanza dal target, il secondo è di tipo spaziale e viene fatto utilizzando la distanza euclidea tra le  $i$  punti selezionati attraverso il primo Clustering. Solo nel secondo Clustering viene imposto un numero di clusters pari al numero di veicoli inseguitori in gioco. In queste simulazioni viene operata una distinzione tra i due veicoli inseguitori, uno definito "principale" e l'altro "secondario" e diventa indispensabile instaurare un canale di comunicazione tra i due veicoli. Il veicolo principale ha il compito di eseguire gli algoritmi di Clustering e di individuare quindi le diverse regioni di approccio, il veicolo secondario deve semplicemente tenersi in ascolto e attendere le informazioni necessarie dal veicolo principale. Quest'ultimo, individuate le zone di approccio, sceglierà la propria selezionando tra tutte quella a lui più vicina e sceglierà inoltre la zona di approccio relativa al veicolo secondario, inviando tale informazione al veicolo interessato. Il veicolo principale viene scelto come quello più vicino al target, e tale distinzione viene fatta autonomamente dagli stessi veicoli. Una prima simulazione viene riportata in Figura 60 in cui il target ha (47, 74) come posizione iniziale. In particolare in Figura 60 (a) è mostrato il risultato del Clustering, ovvero le due zone di approccio evidenziate in rosso e in blu, i cui centroidi sono riportati in nero. Gli algoritmi di Clustering vengono eseguiti solo nella fase iniziale della simulazione. In Figura 60 (b) sono riportate le traiettorie eseguite da target (rosso) veicolo principale (blu tratteggiato) e secondario (blu).

Come si nota dall'immagine (a), nel caso in cui una zona di approccio risulti disgiunta (es. zona rossa), viene scelta come zona di approccio la porzione di tale zona più vicina al veicolo interessato. Dall'immagine (b) si evince che i due veicoli si portano dapprima nelle rispettive zone di approccio, cercate in rosso: il veicolo principale (traiettoria blu tratteggiata) si aggiudica la zona a lui più vicina (zona rossa) mentre quello secondario si sposta in una zona che è strategica per la visibilità del target. Dopo aver raggiunto le differenti zone di approccio e aver, in questo modo, diversificato le traiettorie, i due veicoli puntano il target, che cerca di nascondersi rispetto al veicolo principale ma viene da esso intercettato. Una seconda simulazione, che vede due veicoli impegnati ad intercettare un target, è riportata in Figura 61, in cui il target parte da posizione

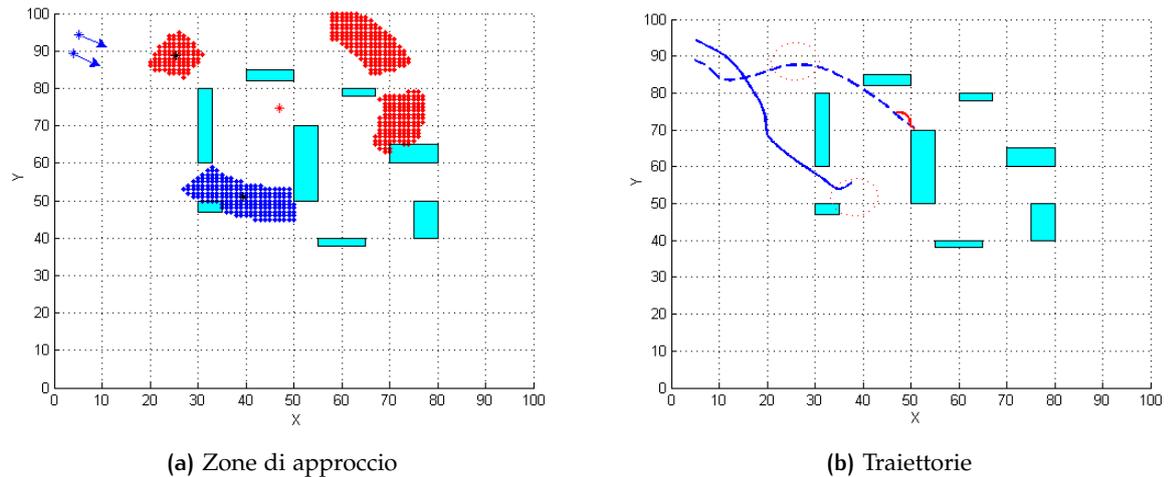


Figura 60.: Intercettazione: 2 Intercettori vs. 1 Target

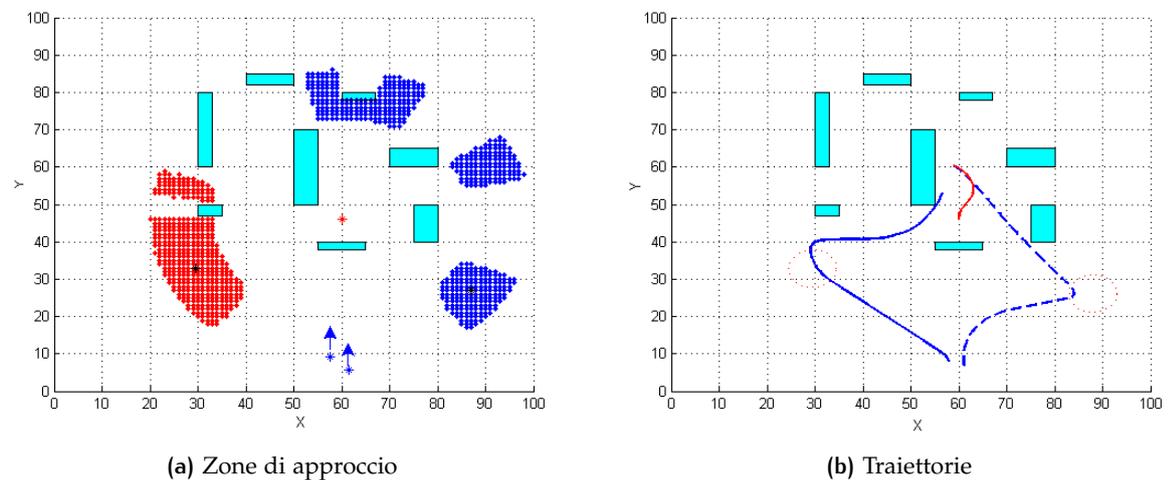


Figura 61.: Intercettazione: 2 Intercettori vs. 1 Target

(60, 46) e cerca di sfruttare gli ostacoli per diminuire la sua visibilità rispetto al veicolo principale. Anche in questo caso in Figura 61 (a) sono mostrate le regioni di approccio, evidenziate in rosso e in blu, con i centroidi in nero. Tali zone sono il risultato degli algoritmi di Clustering applicati al livello di visibilità pesato con la distanza dal target e coincidono con zone ad alta visibilità, come si evince dalla Figura 62 che mostra le curve di livello della visibilità per il target nella sua posizione iniziale. In figura Figura 61 (b) sono riportate le traiettorie di target (rosso), veicolo principale (blu) e secondario (blu tratteggiato).

Il veicolo principale si aggiudica la zona di approccio a lui più vicina (zona rossa). Dall'immagine (b) si nota come il target si muova nel tentativo di abbassare la sua visibilità rispetto al veicolo principale, portandosi in prossimità dell'ostacolo. Entrambi i veicoli, raggiunta la zona di approccio, puntano il target. In questa fase si noti come, mentre il veicolo secondario si muove guidato

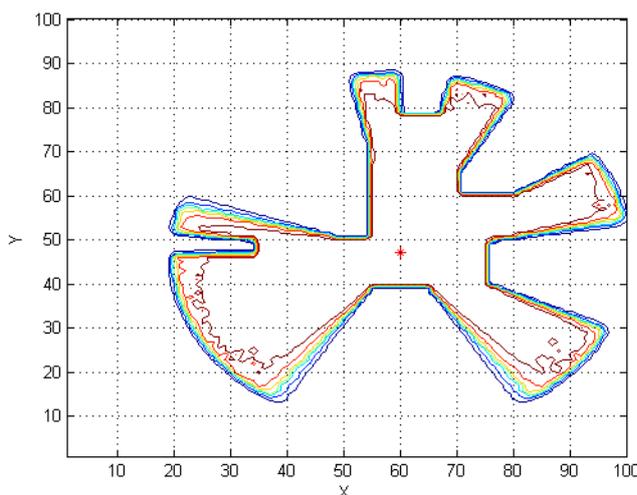


Figura 62.: Curve di livello della visibilità

solo da PNG, in quanto il target è per lui in piena visibilità, il veicolo principale è influenzato dalla manovra del target, infatti la guida lo porterà dapprima a spostarsi aggirando l'ostacolo e solo in seguito prevarrà l'accelerazione di PNG. In questo caso è proprio il veicolo secondario a completare l'intercettazione, avvantaggiato dalla piena visibilità del target.

#### 4.7 TASK DI INTERCETTAZIONE: MOLTEPLICI TARGET

In precedenza si è esteso il task di intercettazione classico in cui un solo veicolo si occupa di intercettare un solo target, con un task più complesso in cui due veicoli, con una iniziale coordinazione, inseguono uno stesso target, che tuttavia cerca di minimizzare la propria visibilità rispetto ad un solo veicolo, quello ad esso più vicino. Si vuole in questa sezione trattare invece il problema duale, in cui più target cercano di sfuggire ad un unico inseguitore. In particolare i target sono manovrati e pilotati dalla legge di guida riportata in (25), che permette ad essi di nascondersi rispetto all'unico veicolo inseguitore sfruttando gli ostacoli, il veicolo d'altra parte deve costantemente affrontare la questione della scelta di quale target inseguire. Esso è pilotato dalla legge di guida (22) che necessita, per la sua parte di PNG, della velocità e della posizione di un target e, per il suo contributo di visibilità, della direzione  $\gamma_{vis}$  prodotta dall'algoritmo, che è peraltro funzione della posizione di un target. La scelta del target potrebbe in alcuni casi essere fatta in maniera statica, stabilendo a priori l'ordine con cui intercettare i target, quindi effettuando la scelta prima dell'inizio dell'ingaggio. Questo potrebbe aver senso in tutte quelle situazioni in cui i target sono intrinsecamente caratterizzati da una diversa priorità; ovvero quando si ha un target è più importante degli altri, per esempio perché più pericoloso o più agile, quindi può nascere l'esigenza di intercettare prima uno e poi i restanti. Un'altra alternativa potrebbe essere quella di calcolare la distanza

iniziale dei due target rispetto al veicolo, quindi intercettare prima il target a distanza minima e ricalcolare le distanze dei rimanenti target, procedendo così fino a che tutti i target saranno intercettati.

Il metodo utilizzato in questa sezione è tuttavia differente dai precedenti approcci. Infatti, avendo come obiettivo il mantenimento di un buon livello di visibilità di tutti i target presenti, si vuole posticipare il più possibile la scelta di quale target seguire in modo esclusivo, scelta tuttavia necessaria per concludere l'intercettazione. Ovvero, finché vi sono le condizioni, si preferisce cercare di mantenere un buon livello di visibilità su tutti i target in gioco, piuttosto che occuparsi soltanto di uno trascurando totalmente gli altri. Questo si traduce in un algoritmo che calcola il livello di visibilità di tutti i target in gioco ed elabora una direzione di miglioramento per ciascun target, ma utilizza come direzione di visibilità  $\gamma_{vis}$  in (22), un'unica direzione che è funzione di queste.

$$\gamma_{vis} = f(\gamma_{T1}, \dots, \gamma_{Tn}) \quad (84)$$

In cui con  $\gamma_{Ti}$  si intende la direzione che apporta un miglioramento di visibilità per il target  $i$ -esimo. Anche per la parte di PNG si utilizzerà una posizione e una velocità non relativi ad un target specifico ma dipendente da quelle di tutti i target, creando così una sorta di target "virtuale" che verrà inseguito almeno nella prima fase dell'ingaggio. Infatti tale comportamento si protrarrà fino a che non sarà verificata una certa condizione che forzi il veicolo inseguitore ad occuparsi di un solo target per concludere l'intercettazione.

#### 4.7.1 Simulazioni: 1 veicolo inseguitore vs. 2 target

Vediamo un esempio di quanto descritto in due simulazioni, nelle quali un veicolo inseguitore deve intercettare due target. I target sono manovrati e pilotati dalla legge di guida che permette ad entrambi di nascondersi rispetto all'unico veicolo inseguitore sfruttando gli ostacoli, il veicolo è pilotato dalla legge di guida PNG con contributo di visibilità, come già visto tale algoritmo necessita della posizione di un solo target e di una direzione di miglioramento di visibilità. Il veicolo conosce le posizioni e le velocità di entrambi i target, e deve costantemente affrontare la scelta di quale tra i due target deve inseguire e incrementarne la visibilità. L'approccio proposto in queste simulazioni mira a mantenere un buon livello di visibilità di tutti i target presenti, non tralasciandone alcuno, almeno in una prima fase. Per fare ciò si utilizza un target fittizio la cui posizione e velocità sono dipendenti da quelle dei due target reali, un possibile modo è semplicemente quello di ricorrere alla media.

$$pos_{Tv} = (pos_{T1} + pos_{T2})/2 \quad vel_{Tv} = (vel_{T1} + vel_{T2})/2 \quad (85)$$

Tali posizioni e velocità sono date in ingresso alla parte PNG dell'algoritmo di guida fino a che non si verifica una certa condizione che obblighi il veicolo

a occuparsi soltanto di un target. Per quanto riguarda il contributo di visibilità dell'algoritmo di guida, anche in questo caso un semplice modo per generare una singola direzione di visibilità dipendente da quelle dei due target è rappresentato dalla media di queste ultime.

$$\gamma_{vis} = (\gamma_{T1} + \gamma_{T2})/2 \quad (86)$$

In tal modo, in una prima fase, il veicolo cercherà di mantenere ad un buon livello le visibilità di entrambi i target, fino a che il verificarsi di una data condizione non renderà necessario imporre un ordine ai target. Nelle simulazioni che seguono tale condizione coincide col fatto che le due direzioni di visibilità calcolate per i due target non devono differire più di una certa soglia (fissata a 120 gradi) oltre alla condizione che i due target devono essere ad una certa distanza (maggiore di 15, per lo scenario in questione), se questo avviene si eviterà di selezionare un target specifico e si utilizzerà quello fittizio, con lo scopo di mantenere in vista entrambi i target reali. Se ad un certo istante dell'ingaggio questa condizione non dovesse essere più verificata, si sceglierà uno dei due target con un certo criterio, che nelle simulazioni presentate coincide con quello a distanza minima.

In Figura 63 sono riportate le traiettorie eseguite da veicolo inseguitore e target.

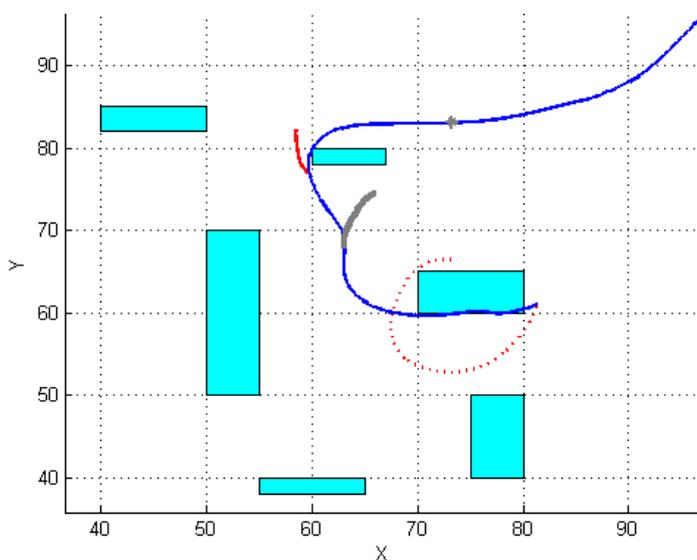


Figura 63.: Traiettorie di veicolo e target

In particolare in blu è riportata la traiettoria del veicolo, in rosso quella del primo target ed in rosso tratteggiato quella del secondo target. Si noti come nella prima fase il veicolo si muove cercando di mantenere un buon livello di visibilità di entrambi i target, ciò si evince meglio dalla Figura 64, che mostra i livelli di visibilità dei due target rispetto al veicolo.

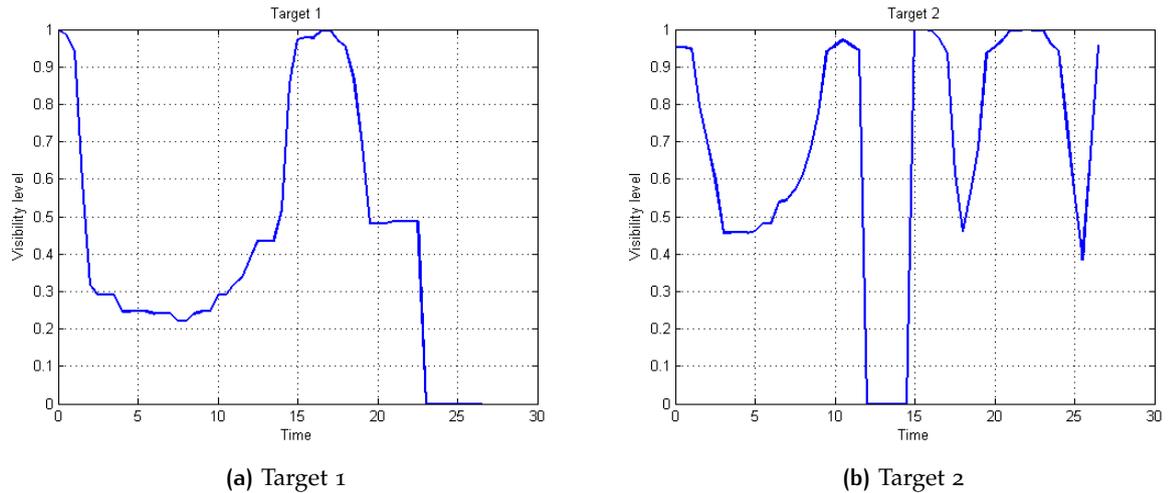


Figura 64.: Livello di visibilità

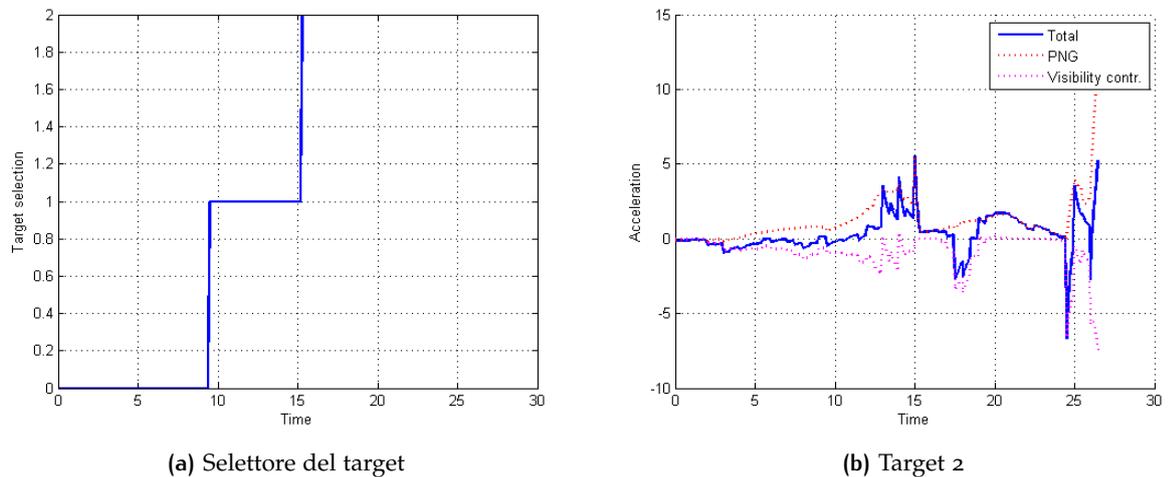


Figura 65.: Accelerazioni veicolo

Questi livelli si mantengono entrambi sufficientemente alti nella prima fase dell'ingaggio, fase in cui nessun target è stato ancora selezionato. Il target è selezionato tramite una variabile il cui andamento è riportato in Figura 65 (a). In particolare se il valore di tale variabile è zero, nessun target viene selezionato e si procede con posizione e velocità del target fittizio e con una direzione di visibilità che è la media delle due, altrimenti si ha che il primo target è stato selezionato se il valore di tale variabile è uno, il secondo target se è due. Si noti come la fase in cui nessuno dei due target prevale si protrae fino all'istante 9, istante in cui il veicolo si trova nel punto evidenziato in grigio nella traiettoria di Figura 63, inoltre, proprio in questa ultima, è possibile notare la posizione del target fittizio fino a quell'istante, che viene riportata in grigio.

In Figura 65 (b) è riportata l'accelerazione totale e le sue componenti di PNG e di visibilità, in cui è possibile notare come nei primi istanti di ogni fase (nes-

sun target selezionato, target 1, target 2) si ha sempre una prevalenza della componente di visibilità sulla PNG, mentre questa acquisisce peso negli istanti finali di tali fasi, questo ad indicare che l'algoritmo di guida opera dapprima per migliorare la visibilità del nuovo target selezionato e poi, una volta in vista, per intercettarlo.

Una seconda simulazione è viene riportata di seguito e la Figura 66 ne mostra le traiettorie di veicolo e target.

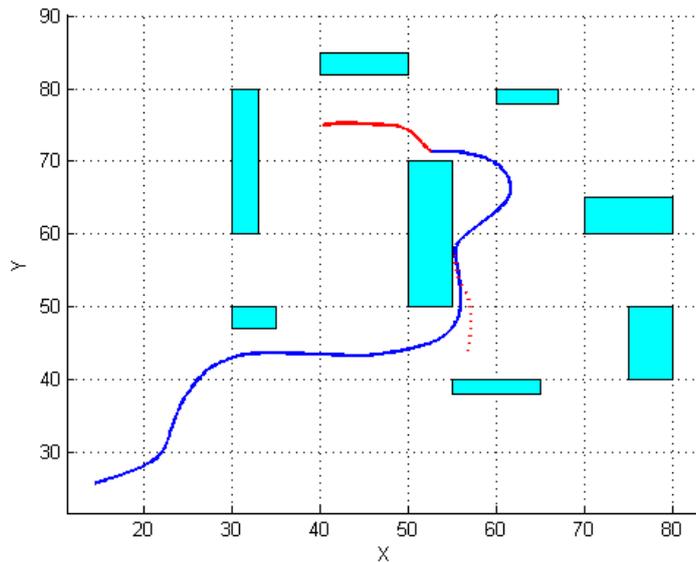


Figura 66.: Traiettorie di veicolo e target

Anche in questo caso in blu è riportata la traiettoria del veicolo, in rosso quella del primo target ed in rosso tratteggiato quella del secondo target. Come nella simulazione precedente, vi è una prima fase in cui nessun target è selezionato e viene creato un target fittizio, la cui velocità e posizione vengono utilizzate dalla componente PNG della legge di guida del veicolo, ed esso cercherà quindi di mantenere alta la visibilità di entrambi i target. A differenza di quanto avviene nella simulazione precedente, dopo questa prima fase viene selezionato dapprima il secondo target (traiettoria tratteggiata) in quanto più vicino, questa selezione avviene anche in questo caso all'istante 9. Infatti dopo tale istante, come si nota dai livelli di visibilità riportati in Figura 67, si ha un miglioramento graduale del livello di visibilità del secondo target.

All'istante 20 circa, il secondo target viene intercettato e il veicolo procede verso il primo target, infatti in seguito il livello di visibilità di tale target migliorerà notevolmente fino a conclusione dell'ingaggio. Anche per questa simulazione vengono riportati i contributi di accelerazione in Figura 68. È interessante notare che proprio a ridosso dell'istante 20 vi è un picco negativo di accelerazione di visibilità, e questa, avendo molto peso in quell'istante, coincide con l'accelerazione del veicolo. Questa accelerazione è riconoscibile anche analizzando la traiettoria del veicolo, infatti essa è responsabile della curva che, dopo aver intercettato il secondo target, porta il veicolo ad allontanarsi dall'ostacolo per

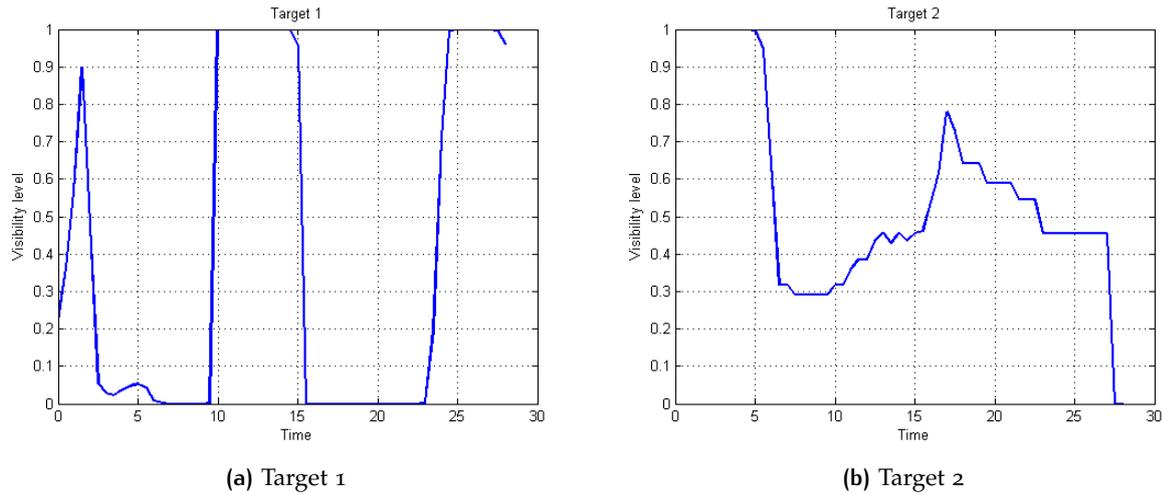


Figura 67.: Livello di visibilità

portare il primo target in visibilità e quindi concludere l'intercettazione.

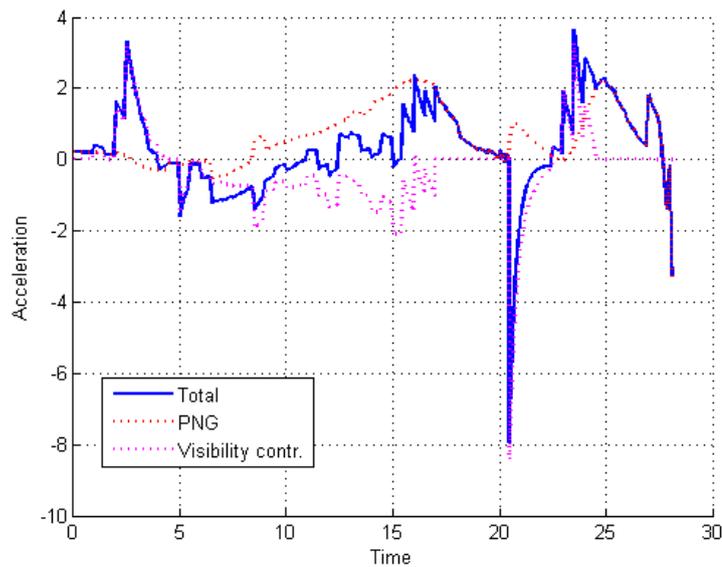


Figura 68.: Accelerazioni veicolo

#### 4.8 TASK INTERCETTAZIONE: 3 VEICOLI

Fin'ora si è esteso il task di intercettazione classico con un solo veicolo ed un solo target, con un task in cui due veicoli inseguono uno stesso target e si è trattato il problema duale, in cui più target cercano di sfuggire ad un unico inseguitore. Consideriamo adesso uno scenario particolare in cui sono presenti tre veicoli.

- Un veicolo inseguitore (Pursuer)
- Un target (Evader)
- Un veicolo terzo (Target Helper)

In cui tutti i veicoli hanno la stessa dinamica utilizzata negli scenari precedenti, riportata in (17). I tre veicoli hanno obiettivi ben diversi, in particolare il veicolo inseguitore ha il classico compito di intercettare il target e quest'ultimo cerca di sfuggire al primo. Il terzo veicolo è a disposizione del target, e viene attivato da quest'ultimo al verificarsi di una certa condizione, ovvero per le simulazioni in questione quando l'inseguitore è ad una distanza inferiore ad una certa soglia. Esso ha il compito di aiutare il target a sfuggire all'inseguitore e lo fa cercando di intercettare quest'ultimo. L'accelerazione per il veicolo inseguitore è la stessa riportata in (22), e per il target, che ha la capacità di nascondersi sfruttando gli ostacoli, è descritta in (25). Quando il veicolo terzo giunge abbastanza vicino all'inseguitore (Pursuer), esso si trasformerà da Pursuer ad Evader e cercherà di fuggire abbandonando l'inseguimento del target, passando ad utilizzare il controllo (25). Nello specifico, questo avviene gradualmente attraverso la seguente funzione.

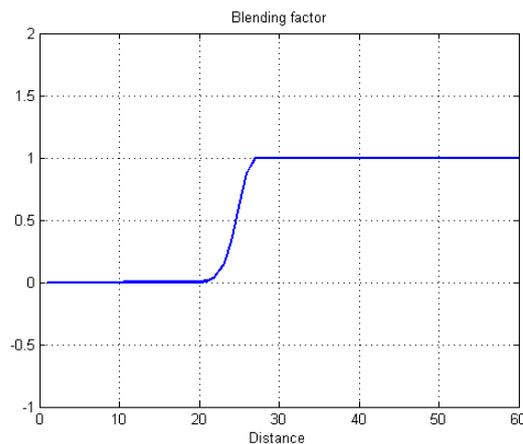


Figura 69.: Blending factor

Tale funzione ha in ingresso la distanza tra veicolo inseguitore e veicolo terzo, quando vale 1 il veicolo inseguitore si preoccuperà effettivamente di seguire il target, quando vale zero si preoccuperà invece di sfuggire al veicolo terzo.

#### 4.8.1 Simulazioni

In Figura 70 sono riportate le traiettorie di una simulazione in cui vengono confrontate le traiettorie con o senza veicolo terzo. Il veicolo inseguitore parte da posizione (13, 27), il target da (68, 29) mentre il veicolo terzo parte da (50, 6). Si noti come l'inseguitore nel caso in cui non è presente il veicolo terzo riesca a intercettare il target nonostante questo si sia nascosto aggirando l'ostacolo,

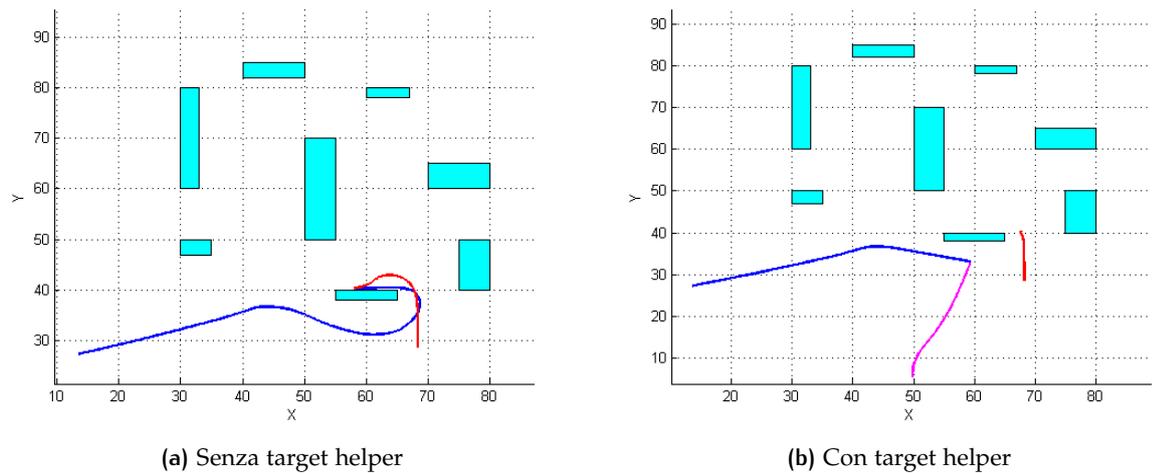


Figura 70.: Traiettorie

mentre, nel caso in cui il terzo veicolo viene attivato, il veicolo inseguitore non riesce a portare a termine l'intercettazione.

In Figura 71 è riportata l'evoluzione della accelerazione del veicolo. Si noti come questa sia dovuta, in una prima fase, ad un comportamento da Pursuer mentre in seguito gradualmente diventa Evader cercando di sfuggire al veicolo terzo.

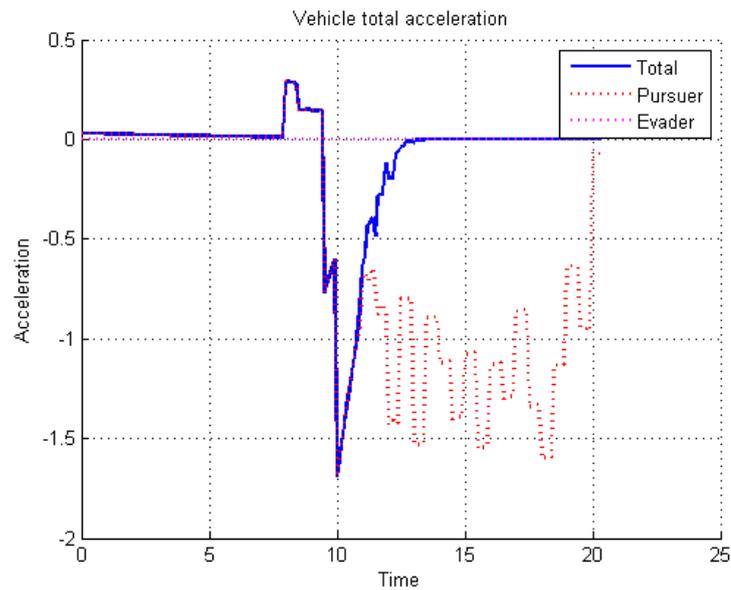


Figura 71.: Accelerazione

Una seconda simulazione ha prodotto delle traiettorie riportate in Figura 72, in cui il veicolo parte da posizione (30, 40), il target da (82, 54) mentre il veicolo terzo parte da (29, 35).

In questa simulazione si nota maggiormente come il veicolo inseguitore si

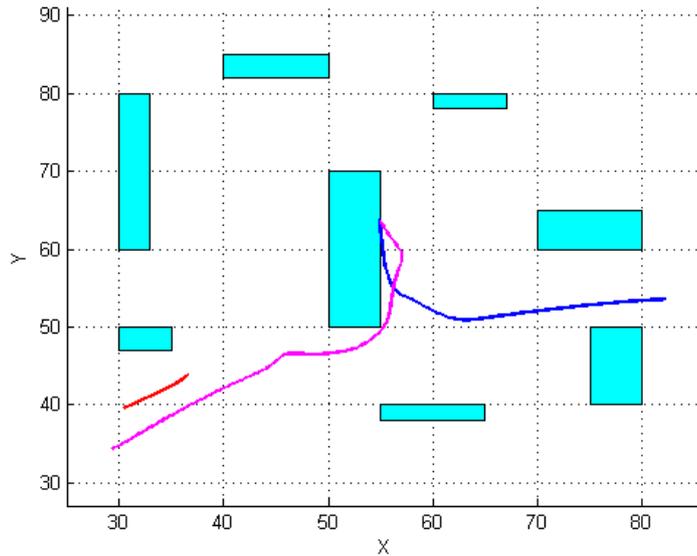
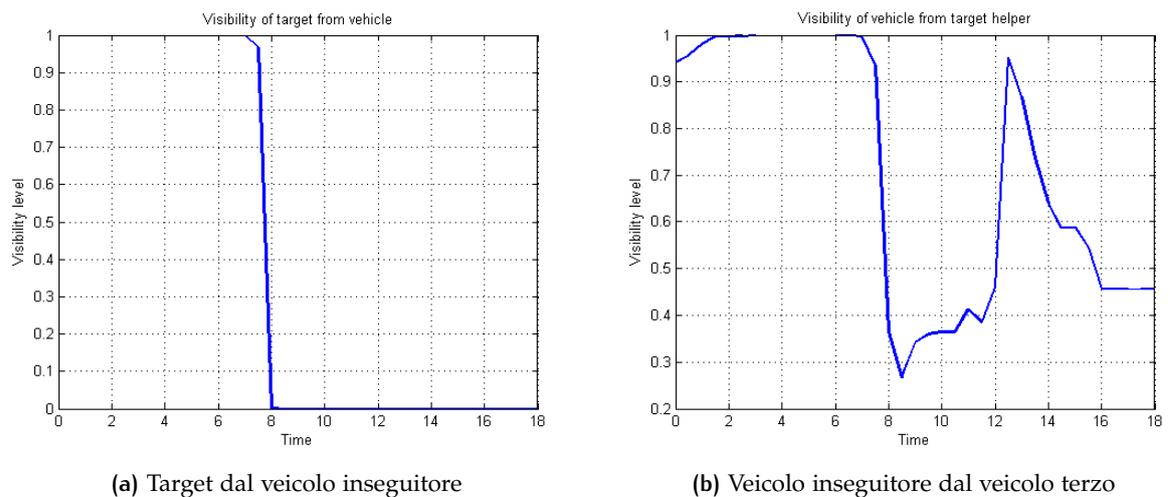


Figura 72.: Traiettorie

preoccupi soltanto in una prima fase si inseguire il target, mentre successivamente cerchi di nascondersi, sfruttando l'ostacolo, dal veicolo terzo ormai vicino. Ciò è confermato dai livelli di visibilità riportati in Figura 73.

Infatti la Figura 73 (a) mostra che, in una seconda fase, il livello di visibilità del target è zero e rimane tale in quanto questo non è più di interesse per il veicolo principale; mentre in Figura 73 (b) è evidente che il veicolo terzo ha interesse a portare l'inseguitore in visibilità per poi intercettarlo. Infatti la sua visibilità ha dapprima un minimo, poiché questo è riuscito a nascondersi, in seguito viene incrementata per poi concludere con l'intercettazione del veicolo inseguitore.



(a) Target dal veicolo inseguitore

(b) Veicolo inseguitore dal veicolo terzo

Figura 73.: Livelli di visibilità



# 5 | CONCLUSIONI

In questo lavoro si è cercato, nella sua prima parte, di dare una caratterizzazione prettamente pratica e concreta alla guida a waypoint con approccio fuzzy, che veniva decritta e simulata già in alcuni articoli riportati in bibliografia. Come in ogni approccio pratico, ci si è trovati di fronte a delle questioni e a delle problematiche che solo a tal punto emergono, ad esempio problemi riguardanti il peso computazionale dell'algoritmo e la gestione della memoria limitata che caratterizza il DSP utilizzato. I sistemi fuzzy utilizzati presentavano un motore di inferenza fuzzy di Takagi-Sugeno; questo, per quanto matematicamente semplice, richiede un notevole numero di computazioni che possono non coincidere con le esigenze di una implementazione real-time su una piattaforma dalle ridotte risorse computazionali. Sfruttando l'interessante proprietà di un sistema di Takagi-Sugeno di essere equivalente a una mappa ingresso/uscita non-lineare e tempo-invariante, si è ottenuta una mappa ingresso/uscita discreta approssimante la caratteristica ingresso/uscita del sistema fuzzy. Usando questo approccio, l'esecuzione del motore di inferenza fuzzy non risultava più necessaria e le due mappe sostitutive sono state agevolmente caricate nella memoria di cui il DSP disponeva. Inoltre, da un punto di vista computazionale, le due mappe sono state implementate come due matrici e il calcolo dell'angolo di heading di riferimento, nel caso più semplice, si riduceva alla lettura di un elemento in una matrice. Risolti tali problemi, si è potuto procedere con una fase, detta di simulazione hardware-in-the-loop (HIL), che solitamente si frappone tra le simulazioni classiche e i test operativi e che permette di ottenere maggiori garanzie sulla buona riuscita di questi ultimi. Le simulazioni (HIL) sono molto utili per la validazione e la qualificazione delle funzionalità dell'hardware, o di parti di esso, in relazione al firmware che si intende utilizzare, prima dell'installazione nell'ambiente operativo. L'autopilota utilizzato presentava svariate interfacce per i sensori, in una simulazione HIL tutti i dati sensoriali venivano simulati e inviati all'hardware dell'autopilota, che, dopo l'elaborazione, produceva un'uscita per i suoi attuatori. Questa uscita poteva esser data agli attuatori reali e, in parallelo, inviata al simulatore in modo da eseguire un altro passo di simulazione e produrre nuovi dati sensoriali da inviare. La simulazione HIL ha prodotto buoni risultati in termini di performance con miss-distance ridotte, e di sincronizzazione, infatti si sono ottenuti valori di TET (Task Execution Time) molto vicini a 10 ms, velocità di comunicazione dell'autopilota, e un numero di pacchetti persi sempre prossimo a zero. Sia la simulazione HIL che il test di volo sono stati fatti utilizzando l'autopilota ICARO, risultante da un progetto interno all'Università di Pisa. In particolare la versione utilizzata (ICARO II o ICARO-Light) è stata progettata per essere piccola e leggera, e presentava diverse interfacce quali: UART, SPI, I2C, Analog/Digital I/Os, PWM generation/capture, e CANBUS. Proprio quest'ultima è

stata utilizzata come canale di comunicazione ad alta velocità per lo scambio rapido dei dati necessari nelle simulazioni HIL.

Nella seconda parte si è intrapreso uno studio sulla visibilità di un target in un ambiente complesso, caratterizzato da ostacoli fissi, considerando il target dapprima fermo, poi avente velocità costante ed infine manovrato e cercando di ricavare una strategia di moto per un veicolo che ha interesse ad osservare o ad intercettare tale target. Il lavoro ha avuto inizio con la definizione un livello di visibilità del target, per far ciò si è fatto ricorso ad una regione di incertezza associata ad esso. Tale regione di incertezza poteva rappresentare sia una effettiva conoscenza non certa della posizione, sia una previsione, entro un certo intervallo temporale, della futura posizione del target. Con questo approccio si è potuto ottenere un livello di visibilità non booleano, ma sfumato, e con una potenza informativa maggiore. Si è quindi impostato un possibile task di sorveglianza, che facilmente ci riconduceva alla prima parte del presente lavoro, in quanto questo poteva essere implementato utilizzando la guida fuzzy con i waypoint posizionati in maniera opportuna, ovvero considerando la visibilità del target, e si è discusso inoltre di un task di intercettazione. Quest'ultimo è stato implementato dapprima considerando un solo target, ottenendo buoni risultati in termini di performance in quanto il target veniva intercettato dopo essere stato portato in una situazione di buona visibilità per il veicolo, e in termini energetici, in quanto i picchi di accelerazione ottenuti rimanevano vicini a valori normali, tipici di una guida PNG. In seguito lo scenario è stato complicato con l'aggiunta di più veicoli per intercettare un unico target. In questo caso si è implementato un meccanismo di coordinazione tra i veicoli, con lo scopo di aumentare le probabilità di mantenere alta la visibilità dell'unico target e di intercettarlo con successo, ed anche in questo caso le simulazioni hanno prodotto risultati positivi. Infine si è trattato il problema duale in cui più target fuggivano da un unico veicolo avente il compito di intercettarli tutti cercando, almeno in parte, di non compromettere la visibilità di nessuno, ottenendo buoni risultati e buone probabilità di concludere l'ingaggio con successo. Inoltre si è voluto fornire all'algoritmo descritto e testato con diverse simulazioni, che aggiunge un contributo di visibilità all'accelerazione tipica della legge di guida PNG, una solida giustificazione teorica, considerando il task di intercettazione tra veicolo e target come un gioco differenziale. Si è formulato un indice di costo e risolto matematicamente il problema di ottimo, ricavando una espressione della accelerazione ottima molto simile a quella utilizzata nelle simulazioni, caratterizzando quindi a livello teorico l'algoritmo di guida proposto. Lo studio di visibilità, caratterizzante la seconda parte del presente lavoro di tesi, è stato giustificato a livello teorico e supportato da diverse simulazioni, non è stato tuttavia accompagnato da test pratici. La validazione degli algoritmi proposti attraverso un test operativo potrebbe rappresentare la continuazione naturale di questo lavoro.

## A.1 BLOCCHI SIMULINK

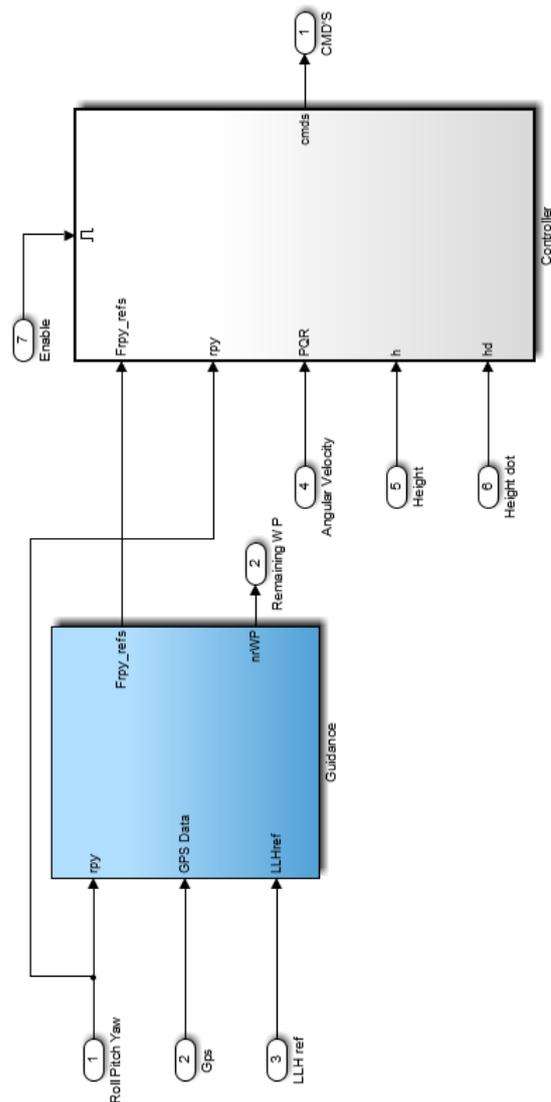


Figura 74.: Guida e Controllori

Il blocco Guidance fornisce i riferimenti di Roll Pitch e Yaw e il numero dei rimanenti waypoint ancora da attraversare. Il blocco Controllers, abilitato quando la guida viene selezionata dall'utente, contiene i PID per il tracking dei riferimenti forniti dalla guida e genera i comandi per i motori.

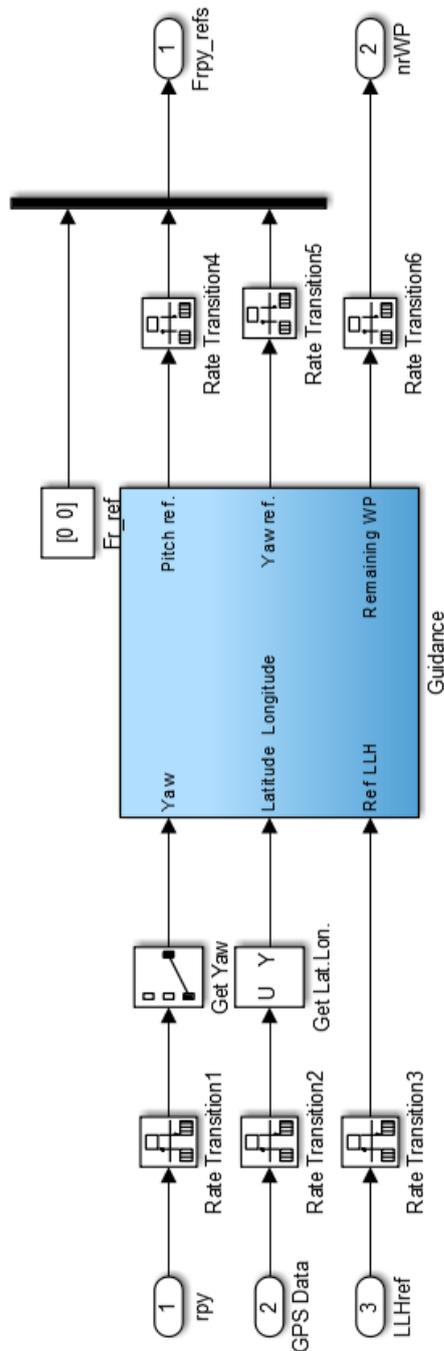


Figura 75.: Blocco Guidance

Tale blocco si occupa di selezionare tra i dati GPS la latitudine e la longitudine e di selezionare l'angolo di imbardata, tutte grandezze necessarie alla guida vera propria contenuta nel blocco in blu. Quest'ultimo produce in uscita i riferimenti di Yaw e Pitch completati con i riferimenti di Roll e Spinta a cui è assegnato il valore zero. Contiene inoltre i Rate-Transition utili ad adattare i dati alla frequenza in cui opera la guida (10Hz).

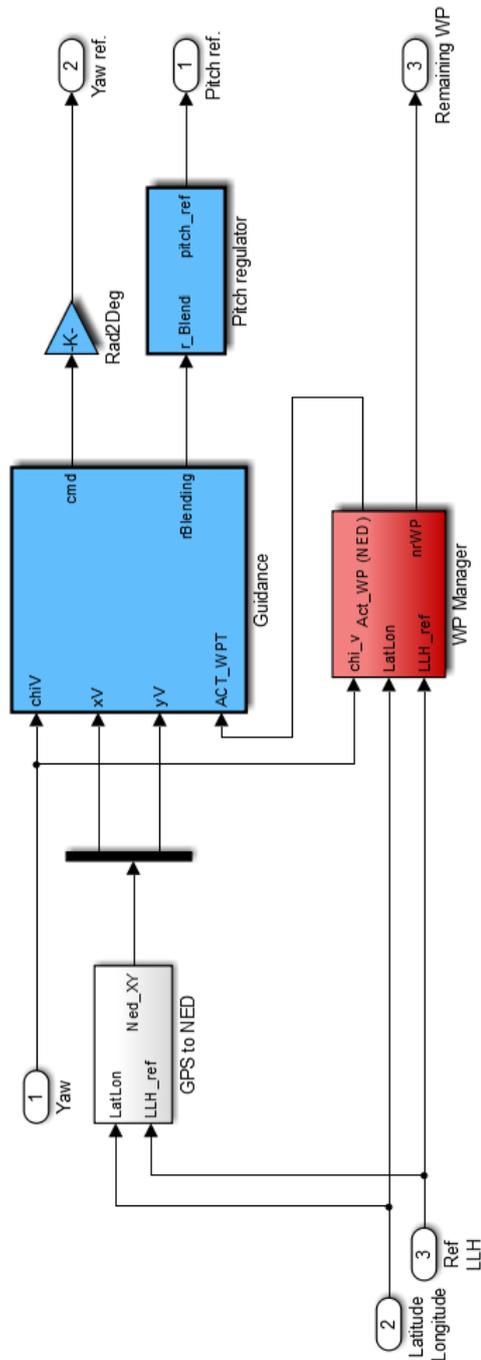


Figura 76.: Interno Blocco Guidance

La guida ha in ingresso il WP da raggiungere, posizione e orientazione del veicolo e produce il riferimento di heading ed il fattore  $r_{blending}$  indice del tipo di guida utilizzata (Long e Short Distance), utile a regolare due differenti livelli di Pitch. Il WP Manager stabilisce se un waypoint è stato raggiunto e indica il prossimo da raggiungere.

## A.2 CODICE MATLAB

Matlab function relativa al blocco Guidance in Figura 76.

```
function [cmd, rBlending] = fcn(chiV, xV, yV, ACT_WPT, Map_sup, Map_inf,
    gauss_r, gauss_y )

% INPUT
% [Current Heading, Current position (X,Y), Active Waypoint]
% OUTPUT
% [Yaw command, rBlending]

% Angle
chiWpt = ACT_WPT(3);

% Position error : E_XY* <=> eXY

% Error rotation matrix
R = [cos(chiWpt+pi/2), -sin(chiWpt+pi/2)
     sin(chiWpt+pi/2), cos(chiWpt+pi/2)];

eXY = [xV-ACT_WPT(1), yV-ACT_WPT(2)]*R;

eX = eXY(1);

eY = eXY(2);

% Long distance guidance

thetaLOS = atan2(-eXY(1), eXY(2));

[errChiLD, correzioneLD] = i_correggoAngolo(chiV, thetaLOS+chiWpt);

fLD = 0.5*errChiLD;

% Short distance guidance

s = 15; % Scaling factor

% Instead of Evalfis(..)
[echi1, echi2] = without_fis(s*eX, s*eY, Map_sup, Map_inf);

[errChiSup, correzioneSDsup] = i_correggoAngolo(chiV, chiWpt+echi1);

[errChiInf, correzioneSDinf] = i_correggoAngolo(chiV, chiWpt+echi2);

% Blending weights

yBlending = new_evalmf(eY+401, gauss_y);

rBlending = new_evalmf(norm(eXY), gauss_r);
```

```

% Heading Error for guidance

Delta_chi = rBlending*(fLD+thetaLOS+2*pi*correzioneLD)+(1-rBlending)*
    (yBlending*(echi1+2*pi*correzioneSDsup)+(1-yBlending)*(echi2+2*pi*
    correzioneSDinf));

% Output
cmd = (ACT_WPT(3)+Delta_chi);

```

```

function [err,correzione] = i_correggoAngolo(angolo,rif)

correzione = 0;

err = angolo - rif;

while (err<-pi),

    correzione = correzione -1;

    err= angolo -(rif+correzione*2*pi);

end

while (err>pi),

    correzione = correzione +1;

    err= angolo -(rif+correzione*2*pi);

end

```

```

function out = new_evalmf(ind, vett)

dim = size(vett);

if (ind < 1)

    ind = 1;

end

if (ind > dim(2))

    ind = dim(2);

end

out = vett(round(ind));

```

```

function [new_echi1, new_echi2]=without_fis(eX,eY, Map_sup, Map_inf)

% Function that replace the output of the FIS FLCRouteInf and FLCRouteSup
% with two map Map_inf and Map_sup, with some grid

% Map_sup:

% rangeX = [-3000 3000]    rangeY = [-2500 2500]

% grid : 150m in X e 150m in Y

% dim: 41x41

sx_sup = 150;

sy_sup = 125;

% Map_inf:

% rangeX = [-3000 3000]    rangeY = [-2000 0]

% grid : 150m in X e 100m in Y

% dim: 41x21

sx_inf = 150;

sy_inf = 100;

% [eX eY] -> [i j]

% SUP

i = ((eX+3000)/sx_sup)+1;

j = ((eY+2500)/sy_sup)+1;

%%%%%%%%%%

s = size(Map_sup);

if (i>=s(1))

    i=s(1)-0.1;

end

if (i<1)

    i=1;

```

```

end

if (j>=s(2))
    j=s(2)-0.1;
end

if (j<1)
    j=1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

P_dec_i=i-fix(i);

P_dec_j=j-fix(j);

i=fix(i);

j=fix(j);

dist = [norm( [P_dec_i, P_dec_j] ), norm( [1-P_dec_i, P_dec_j] ), norm( [P_dec_i, 1-P_dec_j] ), norm( [1-P_dec_i, 1-P_dec_j] )];

% Weighted mean

p=1-dist;

p=p/max(p);

new_echil = ( Map_sup(i,j)*p(1) + Map_sup(i+1,j)*p(2) + Map_sup(i,j+1)*p(3)
    + Map_sup(i+1,j+1)*p(4) )/ sum(p);

% INF

i = ((eX+3000)/sx_inf)+1;

j = ((eY+2000)/sy_inf)+1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s = size(Map_inf);

if (i>=s(1))
    i=s(1)-0.1;
end

```

```

if (i<1)
    i=1;
end

if (j>=s(2))
    j=s(2)-0.1;
end

if (j<1)
    j=1;
end

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

P_dec_i=i-fix(i);
P_dec_j=j-fix(j);

i=fix(i);
j=fix(j);

dist = [norm( [P_dec_i, P_dec_j] ), norm( [1-P_dec_i, P_dec_j] ), norm( [P_
    dec_i, 1-P_dec_j] ), norm( [1-P_dec_i, 1-P_dec_j] )];

% Weighted mean

p=1-dist;

p=p/max(p);

new_echi2 = ( Map_inf(i,j)*p(1) + Map_inf(i+1,j)*p(2) + Map_inf(i,j+1)*p(3)
    + Map_inf(i+1,j+1)*p(4) )/ sum(p);

```

Matlab function relativa al blocco Pitch Regulator in Figura 76.

```

function pitch_ref = fcn(r_Blending)

% Pitch for long distance guidance (deg)

pitch_1 = 20;

% Pitch for short distance guidance (deg)

pitch_2 = 10;

```

```
% Guidance Sample Time
G_ST = 0.1;

% Variation of pitch: deg/sec

pitch_v = 2;

persistent pitch
persistent en_short2long
persistent prec_r

if (isempty(pitch))
    pitch = (r_Blending * pitch_1) + ((1 - r_Blending) * pitch_2);
end

if (isempty(prec_r))
    prec_r = r_Blending;
end

if (isempty(en_short2long))
    en_short2long = 0;
end

if (r_Blending == 1) && (prec_r < 0.001)
    en_short2long = 1;
end

if (en_short2long)
    pitch = pitch + (pitch_v * G_ST);

    if (pitch >= pitch_1)
        en_short2long = 0;
    end
end

else
    pitch = (r_Blending * pitch_1) + ((1 - r_Blending) * pitch_2);
end
```

```
end  
  
prec_r = r_Blending;  
  
pitch_ref = -pitch;
```

# B

## VISIBILITÀ: SCHEMI A BLOCCHI E CODICE

### B.1 BLOCCHI SIMULINK

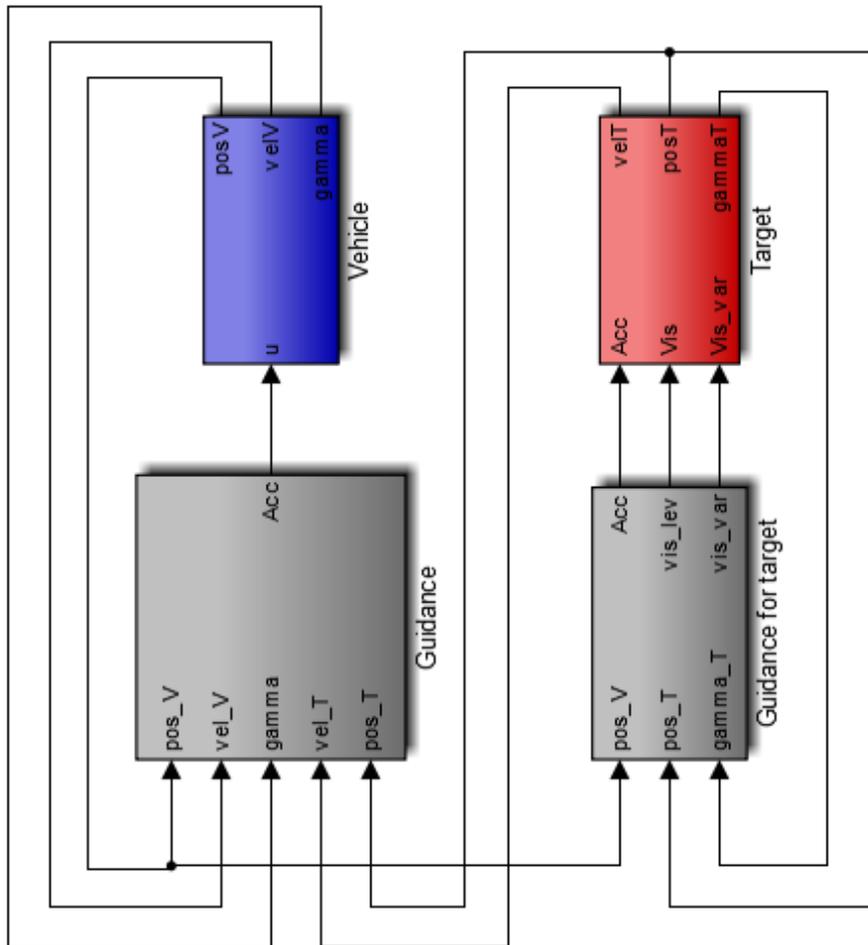


Figura 77.: Simulatore intercettazione: veicolo e target manovrato

Il blocco guida del veicolo inseguitore ha in ingresso, oltre alla propria posizione e velocità, la posizione e la velocità del target. Il blocco guida del target necessita in input, oltre alla propria posizione, solo della posizione del veicolo. In blu è mostrato il blocco relativo al modello del veicolo in rosso del target.

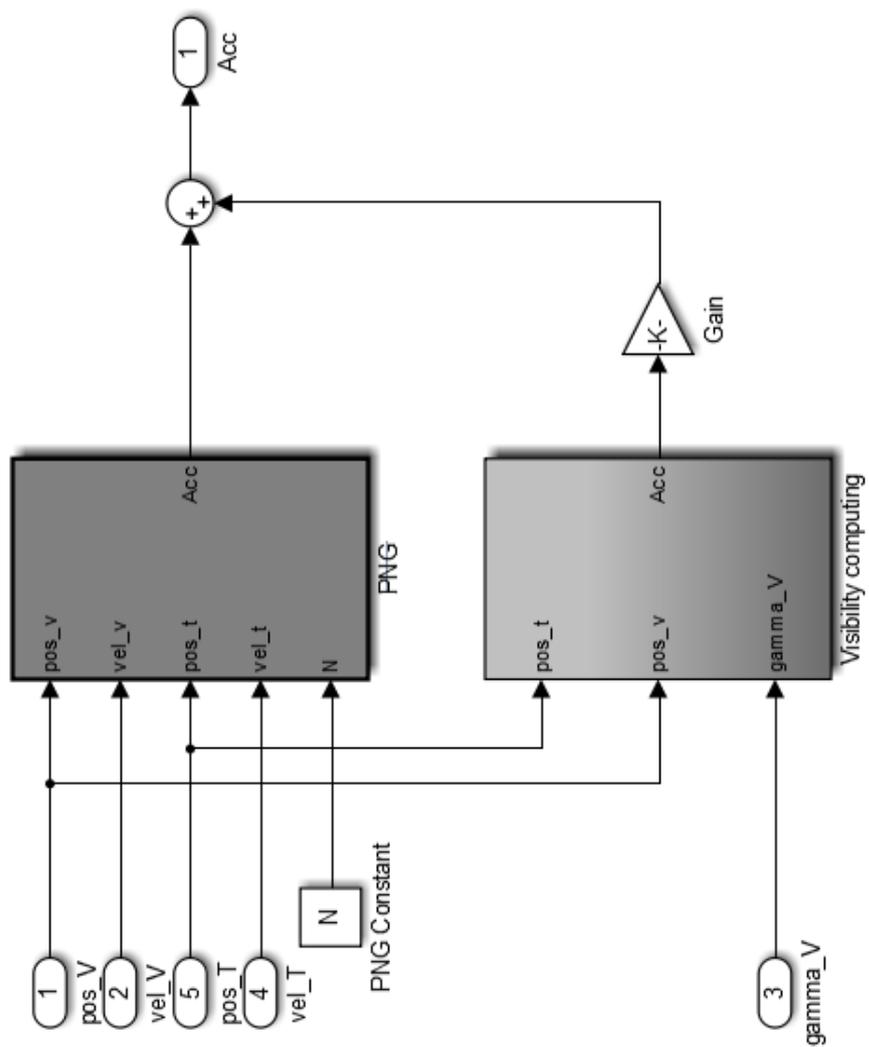


Figura 78.: Blocco guida veicolo

Si nota chiaramente come la legge di guida del veicolo sia risultante dalla somma dei due contributi di accelerazione: PNG e contributo di visibilità.

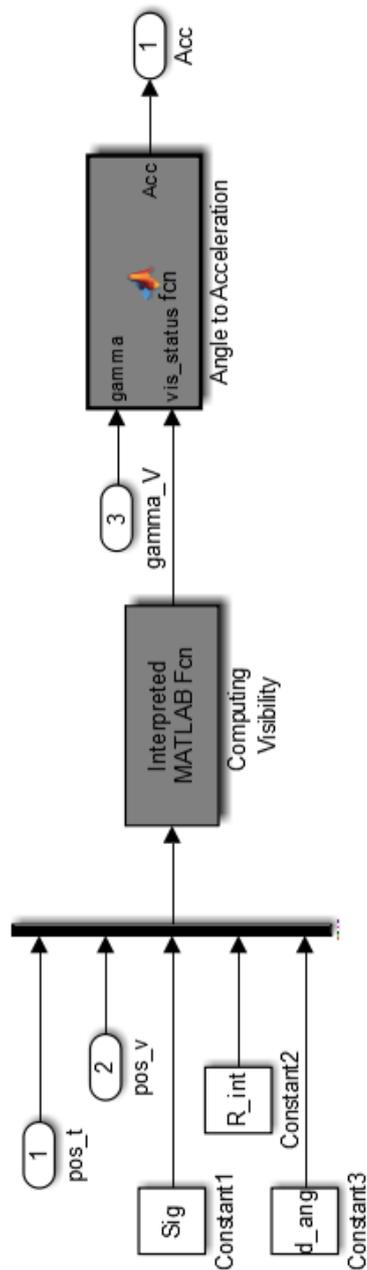


Figura 79.: Blocco visibility computing

Il blocco visibility computing ha in uscita un vettore di dimensione 3 contenente la visibilità attuale del veicolo, la direzione di miglioramento di visibilità e la variazione di visibilità. Inoltre è presente un blocco che traduce la direzione suggerita in accelerazione laterale.

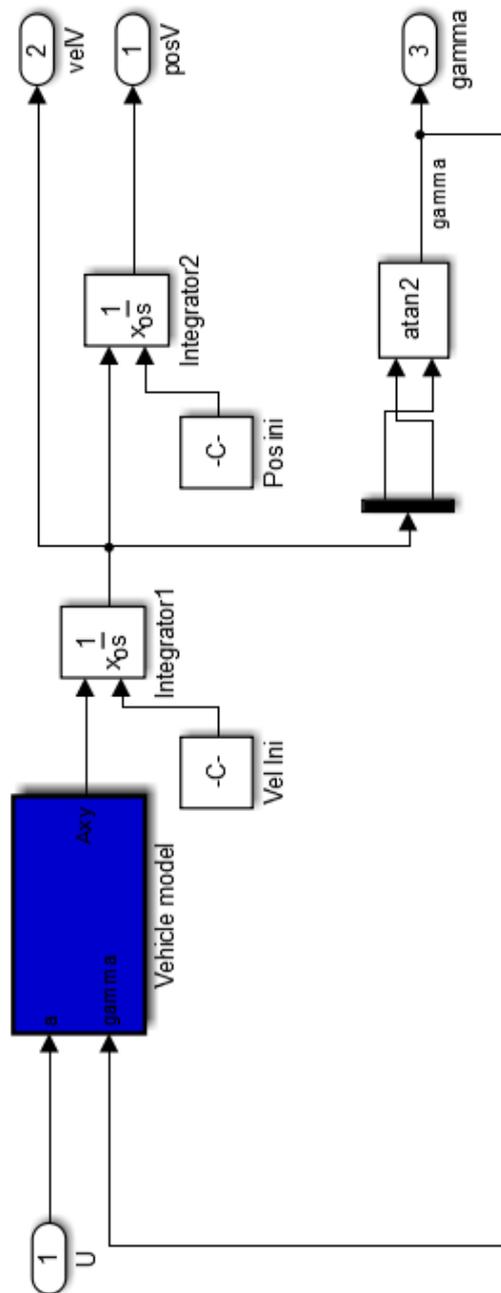


Figura 80.: Blocco dinamica veicolo

Il modello del veicolo ha in ingresso l'accelerazione laterale e produce in uscita un vettore accelerazione a dimensione 2 che viene integrato due volte.

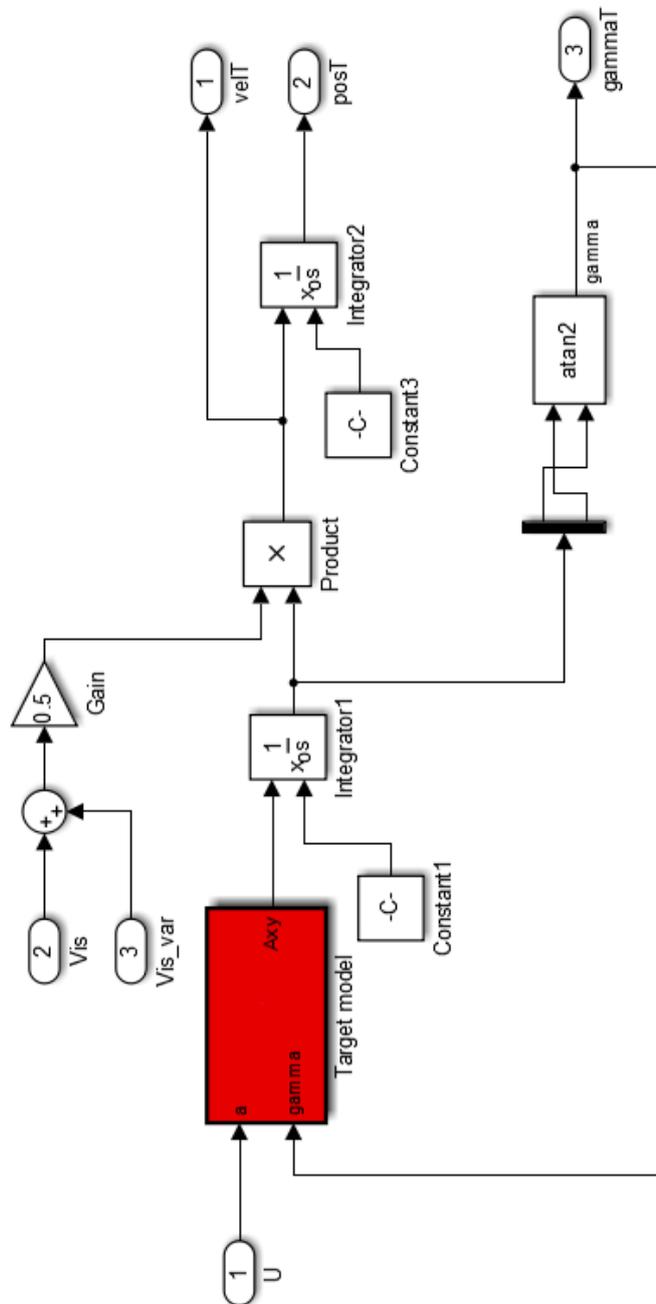


Figura 81.: Blocco dinamica target

Il modello del target è analogo a quello del veicolo con la differenza che alla velocità viene apportato un guadagno coincidente con la media tra la sua visibilità attuale rispetto al veicolo e la variazione che otterrebbe se andasse nella direzione suggerita dalla guida. In tal modo, se nascosto e senza alcuna possibilità di nascondersi meglio, il target rimarrà fermo.

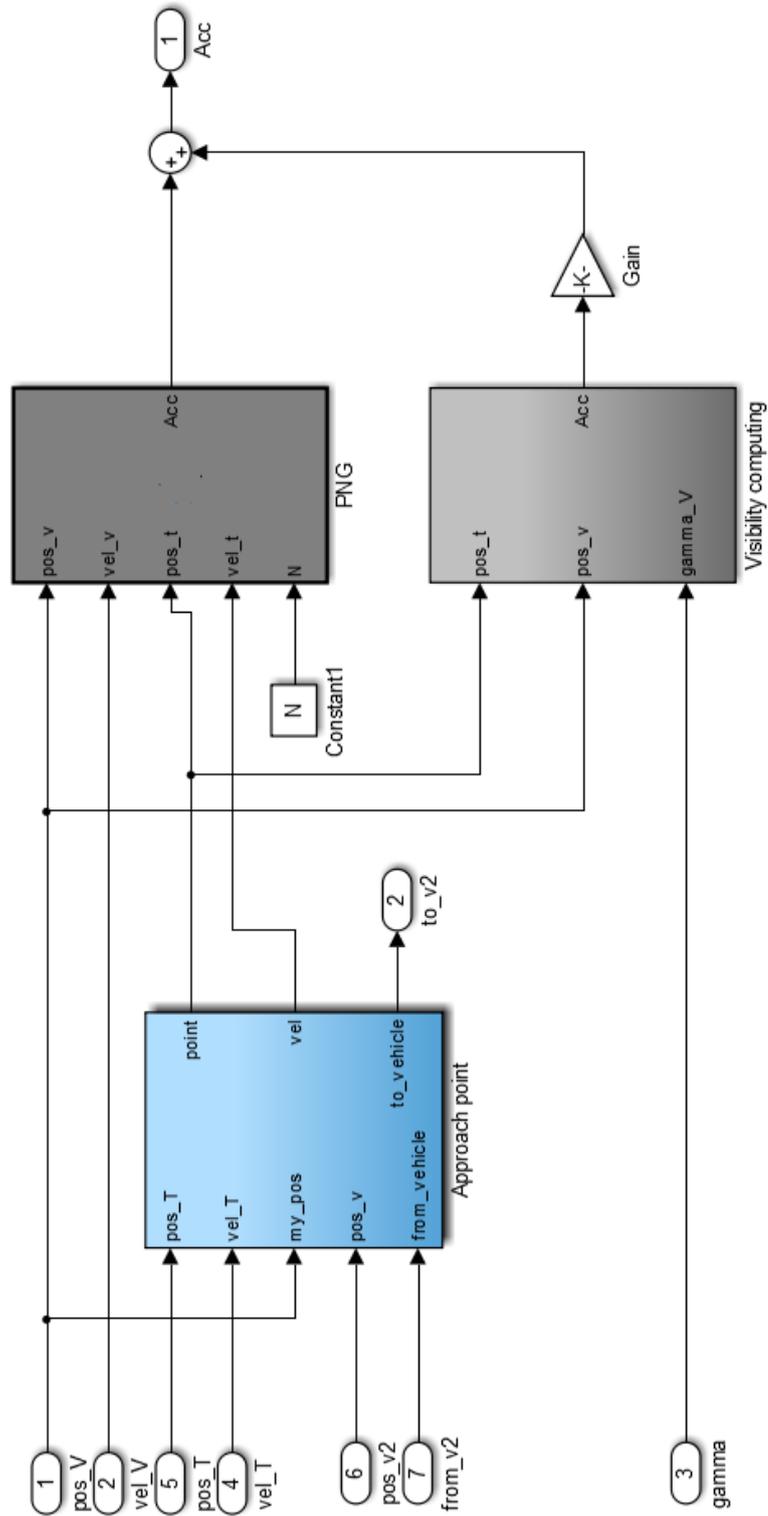


Figura 82.: Blocco guida, 2 veicoli inseguitori

In questo caso, nei blocchi di guida di entrambi i veicoli, si ha un blocco (approach point) che si occupa della coordinazione, della scelta della propria zona di approccio e di quella dell'altro veicolo.

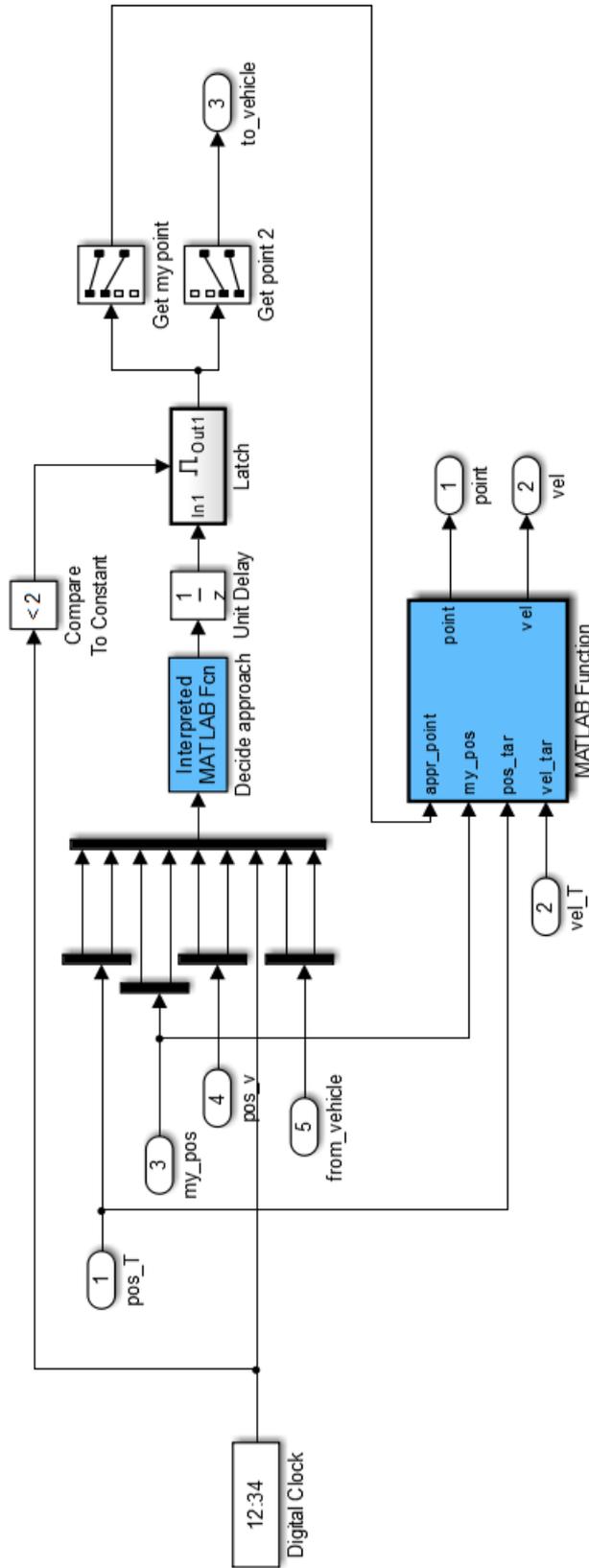


Figura 83.: Blocco approach point

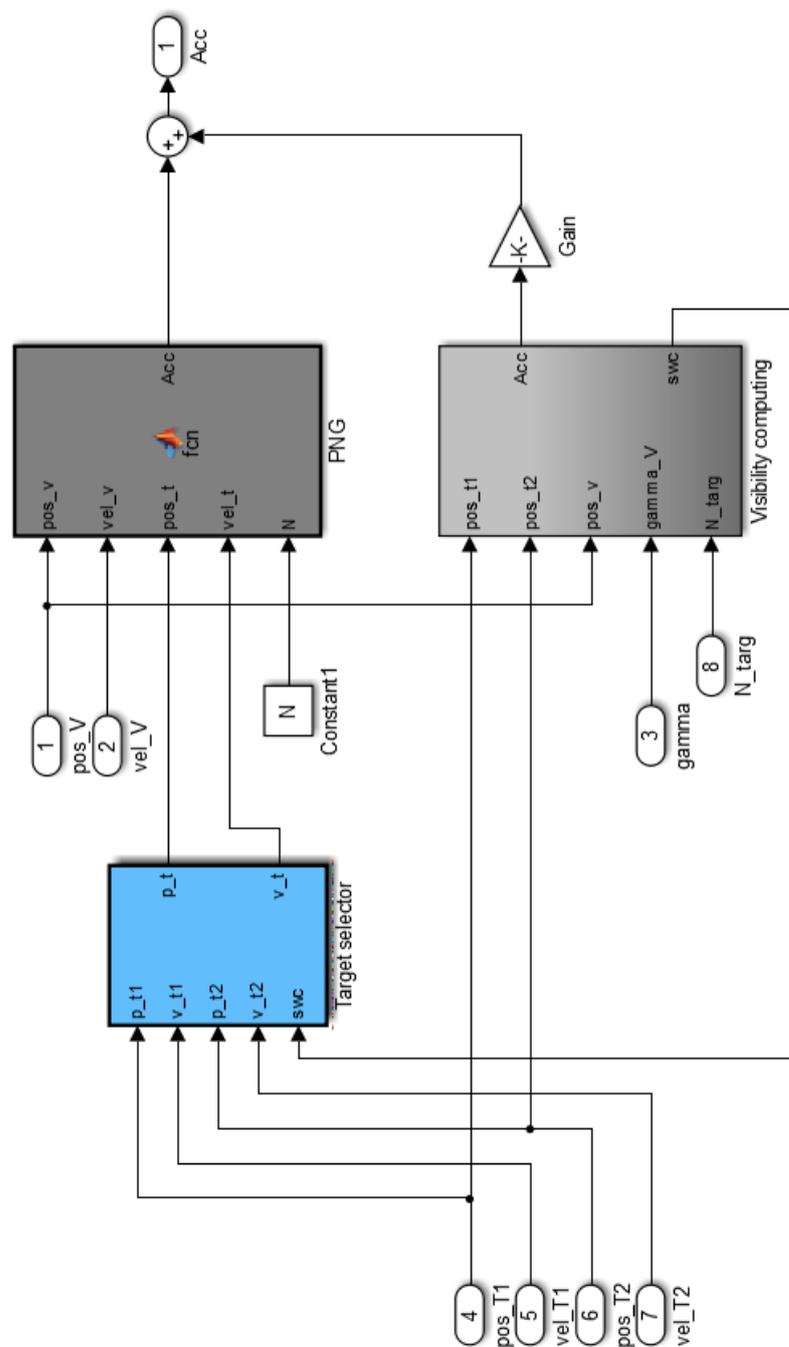


Figura 84.: Blocco guida, 2 target

Nel blocco di guida del veicolo, nel caso di due target da intercettare, è presente un blocco target selector che calcola posizione e velocità da inseguire, dipendenti da quelle dei due target.

## B.2 CODICE MATLAB

Funzione che implementa la guida PNG.

```
function Acc = fcn(pos_v, vel_v, pos_t, vel_t, N)

x_v = pos_v(1);
y_v = pos_v(2);
vel_vx = vel_v(1);
vel_vy = vel_v(2);
vel_tx = vel_t(1);
vel_ty = vel_t(2);
x_t = pos_t(1);
y_t = pos_t(2);
R = sqrt((x_t - x_v)^2 + (y_t - y_v)^2);
R_dot = (((x_t - x_v)*(vel_tx - vel_vx)) + ((y_t - y_v)*(vel_ty - vel_vy)))
/R;
sig_dot = (((vel_ty - vel_vy)*(x_t - x_v)) - ((vel_tx - vel_vx)*(y_t - y_v)
))/(R^2);
Acc_PNG = -R_dot*N*sig_dot;

% HEADING-OFF PART
sig = atan2((y_t - y_v),(x_t - x_v));
gamma_v = atan2(vel_vy, vel_vx);
gamma_t = atan2(vel_ty, vel_tx);
delta_gamma=sig-gamma_v;
if (sin(delta_gamma)<0.1)
    Acc_hoff = 0.1*sign(sin(delta_gamma));
else
    Acc_hoff = sin(delta_gamma);
end
```

```

gamma_mid = sig + real(asin(complex((norm(vel_t)*sin(gamma_t-sig))/norm(vel
_v)))));

beta = real(asin(complex(norm(vel_t)/norm(vel_v)))-0.3491);

if (gamma_v>(gamma_mid-beta)) && (gamma_v<(gamma_mid+beta)) % zone 2

    Acc = Acc_PNG;

else % zone 1 or 3

    Acc = 7*Acc_hoff;

end

```

Funzione che calcola il contributo di visibilità, Interpreted Matlab Function visibile in Figura 79

```

function y = angle_vis(u)

pos_tar = u(1:2);

pos_v = u(3:4);

sig = u(5);

R_int = u(6);

d_ang = u(7);

% Obstacles coordinates

obst=evalin('base','obst');

my_vis = compute_vis(pos_tar,pos_v,obst,sig);

% Consider a circle around the vehicle in which the visibility is computed

K=0;

my_pos=[];

for i=d_ang:d_ang:2*pi % 36 iterations

    K=K+1;

    my_pos(K,1) = pos_v(1) + R_int*cos(i);

    my_pos(K,2) = pos_v(2) + R_int*sin(i);

end

```

```

% Compute visibility for each point around the vehicle

vis_CirPoints=[];

for i=1:K
    vis_CirPoints(i) = compute_vis(pos_tar,my_pos(i,:),obst,sig);
end

% Decide_dir compute a direction for vehicle that maximizes target
% visibility

[des_ang, vis_diff] = decide_dir(vis_CirPoints, my_vis, d_ang, pos_v, pos_
tar);

% Desired angle [deg]

y(1) = des_ang*180/pi;

% Improvement of visibility, normalized

max_vis_diff = 4.4735; % con gauss_xy, senza media di pr ma solo somma

y(2) = (vis_diff/max_vis_diff);

y(3) = my_vis/max_vis_diff;

```

Funzione utilizzata dalla Angle\_vis().

```

function out=compute_vis(pos_tar, pos_v, pos_obst, sig)

% This function takes account of the target uncertainty and, for each point
% of uncertainty zone, computes the target visibility at the vehicle
% position

% Radius of target uncertainty circle

R = 3*sig;

% Vehicle coordinates

pos_vx = pos_v(1);

pos_vy = pos_v(2);

% Obstacles coordinates

x_pol = pos_obst(1,:);

y_pol = pos_obst(2,:);

```

```

all_pr = 0;

k = 0;

x = [pos_tar(1), pos_vx];
y = [pos_tar(2), pos_vy];

[xi, yi] = polyxpoly(x, y, x_pol, y_pol);

if (isempty(xi)) % There are no obstacles on the current LOS

    k = k + 1;

    % Probability of this point

    pr = gauss_xy(0, 0, sig, sig);

    % Update visibility

    all_pr = all_pr+pr;

end

% Consider a target uncertainty circle with radius R

for i = 0.1745:0.1745:2*pi % 10 deg

    for j = R/3:R/3:R % 2*36=72 points -> tot. 2592 + 36 iterations

        x = [pos_tar(1) + j*cos(i), pos_vx];
        y = [pos_tar(2) + j*sin(i), pos_vy];

        % Intersect the obstacles with the LOS

        [xi, yi] = polyxpoly(x, y, x_pol, y_pol);

        if (isempty(xi)) % There are no obstacles on the current LOS

            k = k + 1;

            % Probability of this point

            pr = gauss_xy(x(1) - pos_tar(1), y(1) - pos_tar(2), sig, sig);

            % Update visibility

            all_pr = all_pr+pr;

        end

    end

end

```

```

    end
end
out = all_pr;

```

Funzione utilizzata dalla Angle\_vis().

```

function [out1, out2]=decide_dir(vis, my_vis, d_ang, pos_v, pos_tar)

% This function compares target visibility at vehicle position with
% target visibility at points in a circumference around vehicle, and
% compute the angle that maximizes this difference

all_dir=[];

s=size(vis);

s=s(2);

for i=1:s
    all_dir(i) = diff([my_vis, vis(i)]);
end

% Find all the directions that correspond to max

[my_max, i] = max(all_dir);

s=size(all_dir);

s=s(2);

ind=[];

for i=1:s
    [m,in]=max(all_dir);

    if (m ~= my_max)
        break;
    end

    ind=[ind, in];

    all_dir(in)=-10;
end

```

```

s=size(ind);

ind

s=s(2);

d=[];

for i=1:s

    x=pos_v+3*cos(ind(i)*d_ang);

    y=pos_v+3*sin(ind(i)*d_ang);

    d(i)=norm([x - pos_tar(1), y - pos_tar(2)]);

end

[m_d,in_d]=min(d);

out1=ind(in_d)*d_ang;

out2 = my_max;

```

Funzione che calcola la direzione di visibilità, lato target.

```

function y = angle_vis_targ(u)

% This function computes the angle of target velocity vector that leads to
% a decline of the target visibility from vehicle

pos_tar = u(1:2);

pos_v = u(3:4);

sig = u(5);

% Use, for circle around the target, the same radius and d_ang of the
% circle around vehicle

R_int = u(6);

d_ang = u(7);

% Obstacles coordinates

obst=evalin('base','obst');

% Exchange pos_v and pos_tar -> compute visibility of vehicle from target!

my_vis = compute_vis(pos_v,pos_tar,obst,sig);

```

```

% Consider a circle around the target!

K=0;

my_pos=[];

in_obst=[];

for i=d_ang:d_ang:2*pi % 36 iterations

    K=K+1;

    my_pos(K,1) = pos_tar(1) + R_int*cos(i);

    my_pos(K,2) = pos_tar(2) + R_int*sin(i);

    in_obst(K)=test_point(obst, my_pos(K,1), my_pos(K,2));

end

% Compute visibility for each point around the target

vis_CirPoints=[];

for i=1:K

    vis_CirPoints(i) = compute_vis(pos_v, my_pos(i,:),obst,sig);

end

% Decide_dir_2_targ compute a direction for target that minimizes target
% visibility

[des_ang, vis_diff] = decide_dir_targ(vis_CirPoints, my_vis, d_ang);

% Desired angle [deg]

y(1) = des_ang*180/pi;

max_vis_diff = -4.4735;

out = (vis_diff/max_vis_diff);

y(2) = out;

y(3) = my_vis/-max_vis_diff;

```

Funzione utilizzata dalla `Angle_vis_targ()`.

```

function [out1, out2]=decide_dir_targ(vis, my_vis, d_ang)

% This function compares vehicle visibility at target position with

```

```
% vehicle visibility at points in a circumference around target, and
% compute the angle that minimizes this difference

all_dir=[];

s=size(vis);

s=s(2);

for i=1:s
    all_dir(i) = diff([my_vis, vis(i)]);
end

% Find all the directions that correspond to min

[my_min, i] = min(all_dir);

s=size(all_dir);

s=s(2);

ind=[];

for i=1:s
    [m,in]=min(all_dir);

    if (m ~= my_min)
        break;
    end

    ind=[ind, in];

    all_dir(in)=10;
end

% Compute mean of the directions

m_ind = mean(ind);

s=size(ind);

s=s(2);

d=[];

for i=1:s
```

```

        d(i)=abs(m_ind-ind(i));
end
[m,i]=min(d);
ang = ind(i)*d_ang;
out1 = ang;
out2 = my_min;

```

Funzione che calcola le diverse zone di approccio, nel caso in cui vi sono due veicoli inseguitori. Interpreted Matlab Function visibile in Figura 83.

```

function y=Initial_appr(u)

% Target Position
x_tar=u(1);
y_tar=u(2);

% My Position
my_x=u(3);
my_y=u(4);

% Vehicle2 Position
x_v2=u(5);
y_v2=u(6);

% Clock
clk=u(7);

% Vehicle2 data
from_vehic=[u(8), u(9)];

% Output
y(1)=0;
y(2)=0;
y(3)=0;

```

```

y(4)=0;

% Distances
d_targ=norm([my_x - x_tar, my_y - y_tar]);
d_v2=norm([x_v2 - x_tar, y_v2 - y_tar]);

% If is the primary vehicle
if (d_targ<d_v2) && (clk<2)

    % Compute my approach point and for the other vehicle
    % Load visibility map from Workspace

    vis_map=evalin('base','vis_map');

    % Compute distances from target for all point

    for i=1:100

        for j=1:100

            d(i,j) = norm([i - x_tar, j - y_tar]);

        end

    end

    d=d';

    d=d/max(max(d));

    % Visibility map weighted with the distance from target
    mod_vis_map=vis_map.*d;

    % Clustering mod_vis_map instead of vis_map

    [m, c]=my_kmeans(mod_vis_map,4);

    % Sort the clusters in ascending order

    [c_sort, ind]=sort(c);

    % Choose the cluster more away from the target

    s=size(ind);

    s=s(1);

    ind=ind(s);

```

```
% X contains all the points of the cluster

X=[];

for i=1:100

    for j=1:100

        if (m(i,j)==ind)

            X=[X;[j,i]];

        end

    end

end

end

% Clustering this points in the space
% Number of approach regions, coincides with number of vehicles

n=2;

[clidx, ctrs] = kmeans(X, n);

% Choose my cluster as the nearest

d_ctrs=zeros(n,1);

for i=1:n

    d_ctrs(i)=norm([my_x - ctrs(i,1), my_y - ctrs(i,2)]);

end

[min_dctrs,i]=min(d_ctrs);

my_cl=i;

% Choose cluster of the other vehicle

if (my_cl==1) % Only 2 vehicles

    v2_cl=2;

else

    v2_cl=1;

end

end
```

```

% m contains all the points of my cluster

m=zeros(100,100);

m2=zeros(100,100);

s=size(cidx);

s=s(1);

for i=1:s

    if (cidx(i)==my_cl)

        m( X(i,2), X(i,1) )=1;

    else

        m2( X(i,2), X(i,1) )=1;

    end

end

% I make sure I do not have a cluster with disjoint regions

c=contourc(m,1);

s=size(c);

s=s(2);

centr_point=[];

count=0;

i=1;

while(i<=s)

    % Number of the disjoint regions that are part of my cluster

    count=count+1;

    % Compute their centroids

    centr_point(count,:)= [mean(c(1,i+1:i+c(2,i))) mean(c(2,i+1:i+c(2,i)
        ))];

    i=i+c(2,i)+1;

end

```

```

% Choose the nearest centroid as approach point

d_centr_point=zeros(count,1);

for i=1:count

    d_centr_point(i)=norm([my_x - centr_point(i,1), my_y - centr_point(
        i,2)]);

end

% This is my approach point

[min_d,i]=min(d_centr_point);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% For other vehicle %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

c2=contourc(m2,1);

s=size(c2);

s=s(2);

centr_point2=[];

count=0;

j=1;

while(j<=s)

    % Number of the disjoint regions that are part of my cluster

    count=count+1;

    % Compute their centroids

    centr_point2(count,:)=mean(c2(1,j+1:j+c2(2,j))) mean(c2(2,j+1:j+c
        2(2,j)));

    j=j+c2(2,j)+1;

end

% Choose the nearest centroid as approach point

d_centr_point2=zeros(count,1);

for j=1:count

```

```

        d_centr_point2(j)=norm([my_x - centr_point2(j,1), my_y - centr_
            point2(j,2)]);

    end

    % This is the approach point of vehicle 2

    [min_d,j]=min(d_centr_point2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % The first two outputs are my approach point

    y(1)=centr_point(i,1);

    y(2)=centr_point(i,2);

    % Save in Workspace the approach point

    assignin('base','appr_point_1',[y(1),y(2)]);

    % The second two outputs are the approach point of vehicle 2

    y(3)=centr_point2(j,1);

    y(4)=centr_point2(j,2);

% If is the secondary vehicle
elseif (d_targ>=d_v2) && (clk==1) % 2 times

    % Approach point from primary vehicle

    y(1)=from_vehic(1);

    y(2)=from_vehic(2);

    y(3)=0;

    y(4)=0;

    % Save in Workspace the approach point

    assignin('base','appr_point_2',[from_vehic(1),from_vehic(2)]);

elseif (d_targ>=d_v2) && (clk==0) % 2 times

    y(1)=x_tar;

```

```

    y(2)=y_tar;

    y(3)=0;

    y(4)=0;

end

```

Matlab function visibile in Figura 83.

```

function [point, vel] = fcn(appr_point,my_pos,pos_tar, vel_tar)

persistent ok;

if (isempty(ok))

    ok=0;

end

% - If the vehicle has not reached the approach point: ok=0 (my target is
%   the approach point with zero velocity)
% - Else: ok=1

if (ok==0)

    d=norm([my_pos(1) - appr_point(1), my_pos(2) - appr_point(2)]);

    point=appr_point;

    vel=[0;0];

    if (d<5)

        ok=1;

    end

else

    point=pos_tar;

    vel=vel_tar;

end

```

Funzione all'interno del blocco Target selector, visibile in Figura 84.

```

function [p_t,v_t] = fcn(p_t1,v_t1,p_t2,v_t2, swc)

% swc = 0/1/2
% If swc remain equal to zero, a virtual target is calculated

```

```
% as a mean of the two target, else swc shows the current target
if (swc==0)
    p_t=0.5*(p_t1+p_t2);
    v_t=0.5*(v_t1+v_t2);
elseif (swc==1)
    p_t=p_t1;
    v_t=v_t1;
else
    p_t=p_t2;
    v_t=v_t2;
end
```

## BIBLIOGRAFIA

- [1] S. Baron A.E. Bryson. "Differential games and optimal Pursuit-Evasion strategies". In: *IEEE Transactions on Automatic Control* 10.4 (1965), pp. 385–389.
- [2] Brian A. White Antonios Tsourdos e Madhavan Shanmugavel. *Cooperative Path Planning of Unmanned Aerial Vehicles*. A cura di John Wiley e Llt Sons. 2011.
- [3] Gray W.S. Bezik S. "Guidance of a homing missile via nonlinear geometric control methods". In: *AIAA, Journal of Guidance Control and Dynamics* 18.3 (1995), pp. 441–448.
- [4] Menon P. K. e Iragavarapu V.R. "Blended homing guidance law using fuzzy logic". In: *Proc. AIAA Guid. Nav. Control Conf.* 1998, pp. 1–22.
- [5] Hsu Chun-Fei Lin Chih-Min. "Guidance law design by adaptive fuzzy sliding-mode control". In: *J. Guid., Control Dyn.* 25.2 (2002), pp. 248–256.
- [6] G.Ferri L.Pollini e M.Innocenti. "Fuzzy Guidance for Unmanned Ground Vehicles: Theory and Experiments". In: *AIAA Guidance Navigation and Control Conference*. 2007.
- [7] M.Innocenti L.Pollini. "A synthetic environment for dynamic systems control and distributed simulation". In: *IEEE Control Systems Magazine* 20.2 (2000), pp. 49–61.
- [8] M.Innocenti L.Pollini V.Parnenzini. "Distributed Real-Time Hardware- and Man-in-the-loop Simulation for the ICARO II Unmanned Systems Auto-pilot". In:
- [9] M.Innocenti. "Nonlinear guidance techniques for agile missiles". In: *Control Engineering Practice* 9 (2001), pp. 1131–1144.
- [10] L.Pollini M.Innocenti e D.Turra. "A Fuzzy Approach to the Guidance of Unmanned Air Vehicles Tracking Moving Targets". In: *IEEE Transaction on Control System Technology* 16.6 (2008).
- [11] Gonsalves P.G e Caglayan A.K. "Fuzzy logic PID controller for missile terminal guidance". In: *IEEE International Symposium on.* 1995, pp. 377–382.
- [12] B.Tovar R.Murrieta H.Gonzales. "A reactive motion planner to maintain visibility of unpredictable targets". In: *IEEE International Conference on Robotics and Automation*. 2002.
- [13] S.Bhattacharya R.Murrieta A.Sarmiento. "Maintaining visibility of a moving target at a fixed distance: the case of observer bounded speed". In: *IEEE International Conference on Robotics and Automation*. 2004.

- [14] S.Hutchinson R.Murrieta B.Tovar. "A simpling based motion planning approach to maintaing visibility of unpredictable targets". In: *Autonomous Robots* 19 (2005), pp. 285–300.
- [15] C.Backer S.LaValle H.Gonzales. "Motion strategies for maintaining visibility of a moving target". In: *IEEE International Conference on Robotics and Automation*. 1997.
- [16] R.Murrieta T.Muppirala S.Hutchinson. "Optimal motion strategies based on critical events to maintain visibility of a moving target". In: *IEEE International Conference on Robotics and Automation*. 2005.
- [17] T.Takagi e M.Sugeno. "Fuzzy identification of system and its applications to modeling and control". In: *IEEE Trans. Syst. Man Cybern.* SMC-15.1 (1985), pp. 116–132.