



Università di Genova

Scuola Politecnica

Corso di Laurea Magistrale in Yacht Design

TESI DI LAUREA

Studio e sviluppo di un sistema di ormeggio automatico

RELATORI

Prof. Ing M. Martelli

Ing C. Fruzzetti

CANDIDATO

Vincenzo Ruocco

Contents

1. INTRODUZIONE.....	9
1.1. Literature review.....	11
2. SVILUPPO DEL SISTEMA AUTOBERTHING.....	14
2.1. Algoritmo A*	15
2.2. Implementazione del codice matlab.....	16
2.3. Legge della velocità.....	20
2.4. Ottimizzazione della rotta.....	22
3. GUIDANCE.....	26
3.1. Starter	28
3.2. Waypoint management	29
3.3 path planning	29
4. SISTEMA DI CONTROLLO PER POSIZIONAMENTO DINAMICO.....	30
4.1. Controller	31
4.2. FAL.....	33
4.3. TAL.....	35
5.TUNING	37
5.1. Prova statica di puro avanzo.....	37
5.2. Prova statica di pura deriva	48
5.3. Prova statica di pura imbardata.....	59
5.4. Prova statica completa	68
5.5. Prova di avanzo a velocità richiesta non nulla.....	69
6. RISULTATI.....	76
6.1. Geometria portuale semplice	76
6.2. Geometria portuale intermedia.....	80
6.3. Geometria portuale complessa.....	85
7. CONCLUSIONI E SVILUPPI FUTURI	92

Lista delle figure

Figura 1 - Sinistri di navi di qualunque bandiera dal 2010 al 2019 [fonte].....	9
Figura 2 - Sinistri di navi di qualunque bandiera differenziati in base alla categoria[fonte]	10
Figura 3 - Schema a blocchi del sistema di autoberthing	14
Figura 4 - Schema rappresentativo.....	15
Figura 5 - Geometrie portuali	17
Figura 6 - Approssimazione della figura di galleggiamento.....	18
Figura 7 - Diagramma di flusso dell’algoritmo.....	19
Figura 8 - Intorno quadrato ed orientazione figura di galleggiamento.....	19
Figura 9 - Profilo di velocità trapezoidale	21
Figura 10 - Input di velocità e direzione	22
Figura 11 - Diagramma di flusso dell'ottimizzazione del numero dei waypoints.....	23
Figura 12 - Rotta ottimizzata	24
Figura 13 - Controllo collisioni con figura di galleggiamento	25
Figura 14 - Diagramma funzionamento autoberthing.....	26
Figura 15 - Diagramma di flusso guidance.....	27
Figura 16 - Schema a blocchi starter.....	28
Figura 17- Waypoint management.....	29
Figura 18 - Modellino Tito neri	30
Figura 19 - Layout sistema di posizionamento dinamico	31
Figura 20 - Allocazione propulsori	34
Figura 21 - Plotter prima prova puro avanzo.....	38
Figura 22 - Grafico velocità prima prova puro avanzo	39
Figura 23 - Grafico posizione prima prova puro avanzo.....	39
Figura 24 - Grafico tensione propulsore di sinistra prima prova puro avanzo.....	40
Figura 25 - Grafico tensione propulsore di dritta prima prova puro avanzo	40
Figura 26 - Grafico tensione bow thuster prima prova puro avanzo	41
Figura 27 - Plotter seconda prova puro avanzo.....	42
Figura 28 - Grafico velocità seconda prova puro avanzo	43
Figura 29 - Grafico posizione seconda prova puro avanzo.....	42
Figura 30 - Grafico tensione propulsore di sinistra seconda prova puro avanzo.....	44
Figura 31 - Grafico tensione propulsore di dritta seconda prova puro avanzo.....	43
Figura 32 - Grafico tensione bow thruster seconda prova puro avanzo	44
Figura 33 - Grafico posizione terza prova puro avanzo	46
Figura 34 - Plotter terza prova puro avanzo	45
Figura 35 - Grafico tensione propulsore di dritta terza prova puro avanzo.....	47
Figura 36 - Grafico velocità terza prova puro avanzo.....	46
Figura 37 - Grafico tensione bow thruster terza prova puro avanzo	48
Figura 38 - Grafico tensione propulsore di sinistra terza prova puro avanzo	47
Figura 39 - Plotter prima prova pura deriva	49
Figura 40 - Grafico velocità prima prova pura deriva	50
Figura 41 - Grafico posizione prima prova pura deriva	49

Figura 42 - Grafico tensione propulsore di sinistra prima prova pura deriva	51
Figura 43 - Grafico tensione propulsore di dritta prima prova pura deriva	50
Figura 44 - Grafico tensione bow thruster prima prova pura deriva	51
Figura 45 - Grafico posizione seconda prova pura deriva	53
Figura 46 - Plotter seconda prova pura deriva	52
Figura 47 - Grafico tensione propulsore di dritta seconda prova pura deriva	54
Figura 48 - Grafico velocità seconda prova pura deriva	53
Figura 49 - Grafico tensione bow thruster	55
Figura 50 - Grafico tensione propulsore di sinistra seconda prova pura deriva	54
Figura 51 - Plotter terza prova pura deriva	56
Figura 52 - Grafico posizione terza prova pura deriva.....	56
Figura 53 - Grafico tensione propulsore di dritta terza prova pura deriva	57
Figura 54 - Grafico velocità terza prova pura deriva	57
Figura 55 - Grafico tensione bow thruster terza prova pura deriva.....	58
Figura 56 - Grafico tensione propulsore di sinistra terza prova pura deriva.....	58
Figura 57 - Plotter prima prova pura imbardata	59
Figura 58 - Grafico posizione prima prova pura imbardata.....	60
Figura 59 - Grafico velocità prima prova pura imbardata	60
Figura 60 - Grafico tensione propulsore di sinistra prima prova pura imbardata.....	61
Figura 61 - Grafico tensione propulsore di dritta prima prova pura imbardata	61
Figura 62 - Grafico tensione bow thruster prima prova pura imbardata.....	62
Figura 63 - Grafico posizione seconda prova pura imbardata.....	63
Figura 64 - Plotter seconda prova pura imbardata.....	63
Figura 65 - Grafico tensione propulsore di dritta seconda prova pura imbardata	64
Figura 66 - Grafico velocità seconda prova pura imbardata	64
Figura 67 - Grafico tensione bow thruster seconda prova pura imbardata.....	65
Figura 68 - Grafico tensione propulsore di sinistra seconda prova pura imbardata.....	65
Figura 69 - Grafico velocità terza prova pura imbardata.....	66
Figura 70 - Grafico posizione terza prova pura imbardata	66
Figura 71 - Grafico tensione propulsore di sinistra terza prova pura imbardata	67
Figura 72 - Grafico tensione bow thruster	67
Figura 73 - Spostamento in x e y con prua a 0°	68
Figura 74 - Spostamento in x e y con prua a 45°	69
Figura 75 - Plotter prima prova avanzo a velocità richiesta non nulla	70
Figura 76 - Grafico velocità prima prova avanzo a velocità richiesta non nulla	71
Figura 77 - Grafico velocità prima prova avanzo a velocità richiesta non nulla	71
Figura 78 - Grafico tensione propulsore di dritta prima prova avanzo a velocità richiesta non nulla	72
Figura 79 - Grafico posizione seconda prova avanzo a velocità richiesta non nulla	73
Figura 80 - Plotter seconda prova di avanzo a velocità richiesta non nulla	73
Figura 81 - Grafico velocità seconda prova di avanzo a velocità richiesta non nulla	74
Figura 82 - Grafico tensione propulsore di dritta seconda prova di avanzo a velocità richiesta non nulla	74
Figura 83 - Plotter prova finale geometria semplice	77

Figura 84 - Confronto x prova finale geometria semplice	78
Figura 85 - Grafico posizione prova finale geometria semplice	77
Figura 86 - Confronto y prova finale geometria semplice	78
Figura 87 - Confronto psi prova finale geometria semplice	79
Figura 88 - Grafico velocità prova finale geometria semplice.....	79
Figura 89 - Confronto velocità u prova finale geometria semplice	80
Figura 90 - Plotter prova finale geometria intermedia.....	81
Figura 91 - Confronto x prova finale geometria intermedia	82
Figura 92 - Grafico posizione prova finale geometria intermedia.....	82
Figura 93 - Confronto y prova finale geometria semplice	83
Figura 94 - Confronto psi prova finale geometria intermedia.....	83
Figura 95 - Confronto velocità u prova finale geometria intermedia.....	84
Figura 96- Grafico velocità prova finale geometria intermedia	84
Figura 97 - Plotter prova finale geometria complessa	85
Figura 98 - Grafico posizione prova finale geometria complessa.....	86
Figura 99 - Confronto x prova finale geometria complessa	86
Figura 101 - Confronto psi prova finale geometria complessa	88
Figura 100 - Confronto y prova finale geometria complessa	88
Figura 102 - Grafico velocità prova finale geometria complessa	88
Figura 104 - Confronto velocità u prova finale geometria complessa	89
Figura 103 - Grafico tensione propulsore di dritta prova finale geometria complessa	89
Figura 106 - Grafico tensione bow thruster prova finale geometria complessa.....	90
Figura 105 - Grafico tensione propulsore di sinistra prova finale geometria complessa....	90

Lista delle tabelle

Tabella 1 – Confronto tra gli algoritmi.....	12
Tabella 2 - Coefficienti seconda prova puro avanzo.....	41
Tabella 3 - Coefficienti seconda prova pura deriva	52
Tabella 4 - Coefficienti seconda prova pura imbardata	62
Tabella 5 – Coefficienti seconda prova avanzo a velocità richiesta non nulla	72

Lista delle equazioni

Equazione 1 – Equazione A*	16
Equazione 2 – Geometria ellittica.....	18
Equazione 3 – Velocità desiderata.....	21
Equazione 4 – Calcolo vettori errori	31
Equazione 5 – Matrice di rotazionee	32
Equazione 6 – Calcolo vettore forza	32
Equazione 7 – Vettore forza in base nave	33
Equazione 8 – Sistema di equilibrio.....	33
Equazione 9 – angolo dei propulsori	35
Equazione 10 - Adimensionalizzazione delle forze di spinta	35
Equazione 11 - Definizione dei coefficienti PID	37

1. INTRODUZIONE

La continua espansione del settore della nautica ha di fatto incrementato il traffico marittimo, specialmente quello portuale e costiero. Questo ha suscitato un aumento di interesse per quanto riguarda i sistemi di ausilio alla navigazione, al fine di avere una maggiore sicurezza ed un approccio facilitato al problema che permette, ad utenti con limitata esperienza, di concludere ugualmente la manovra al meglio.

In base ai dati pubblicati dal MIT [1] sui sinistri marittimi, si evince che il maggior numero di collisioni avvengono nell'ambito del diporto e sono riconducibili maggiormente a errori umani o condizioni ambientali. Anche dai dati pubblicati dall'EMSA [2] si può notare come nonostante ci sia una globale diminuzione di collisioni, essi siano riconducibili principalmente (89,5%) ad errori umani e avvengano maggiormente (41%) nelle acque interne in corrispondenza delle aree portuali.

Un'altra pubblicazione sui sinistri marittimi in ambito diportistico è quella emessa da U.S. Coast Guard's Boating Safety Division [3] che analizza e cataloga tutti i dati relativi ai vari incidenti avvenuti durante l'anno. In particolare, anche in questo caso si misura una complessiva diminuzione del 15,7% sui sinistri marittimi rispetto agli altri anni.

Di seguito nella Figura 1 e Figura 2 sono riportati alcuni dati presi dalla pubblicazione [1] dove si può vedere l'elaborazione dei dati relativi a tutti i sinistri marittimi sia in ambito commerciale che diportistico.

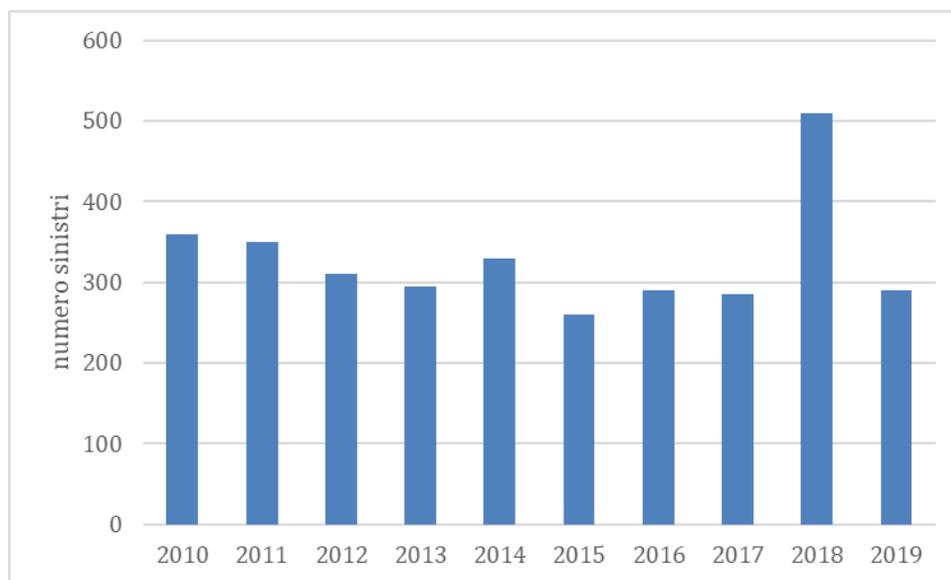


Figura 1 - Sinistri di navi di qualunque bandiera dal 2010 al 2019 [fonte]

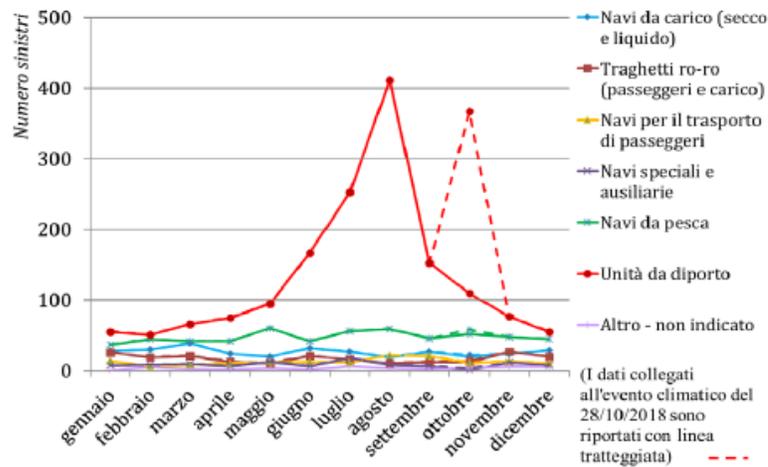


Figura 2 - Sinistri di navi di qualunque bandiera differenziati in base alla categoria[fonte]

In generale, come descritto in [4], tra i sistemi utili all'assistenza alla navigazione abbiamo lo smart pilot (ovvero l'insieme di autopilota, capace di mantenere un determinato angolo di prua, e del controllore di velocità, capace di assicurare all'imbarcazione una velocità stabilita) e il sistema di posizionamento dinamico (DP) che è capace di mantenere l'imbarcazione ferma con una data orientazione e, nel caso di errore fa seguire tratti a bassa velocità. Questi sono solo alcuni di quelli installati al giorno d'oggi su un gran numero di mezzi. Esistono altre funzionalità più avanzate che sono spesso dedicate al settore dell'offshore.

Per evitare eventuali sinistri marittimi, facilitando la gestione della manovra, vengono sempre più studiati, progettati e installati, impianti ausiliari che, integrati con l'impianto di propulsione e di manovra, permettono di avere un completo controllo sulla navigazione e movimentazione dell'imbarcazione.

In generale, si può dire che durante la fase progettuale degli impianti ausiliari, al fine di ottimizzare al meglio la progettazione, vengono effettuati test su sistemi che possono essere reali o virtuali.

Come riportato nelle dispense del corso di Impianti Navali [5], e in [6], nel caso di modelli virtuali risulta opportuno raffigurare i fenomeni fisici attraverso opportuni modelli che ci aiutano a schematizzare i processi reali.

Questi sono tipicamente sviluppati sulla base di tre criteri principali [5]:

- Generalità
- Realismo
- Fedeltà.

Tali modelli, vengono sviluppati al fine di comprendere la realtà e cercare di fare previsioni in modo da poter testare un opportuno sistema di controllo.

Tramite le tecniche di programmazione è possibile quindi creare un modello virtuale che tiene fede ai fenomeni fisici permettendo di studiare sia il comportamento dei sistemi sia di effettuare le opportune simulazioni, così da ipotizzare quello che sarà il risultato finale.

È utile utilizzare dunque un ambiente per il calcolo numerico come, per esempio, Matlab o software per la modellazione e simulazione come Simulink.

L'obiettivo di questo elaborato, quindi, è quello di sviluppare un sistema di supporto decisionale per ormeggio o auto-attracco e nelle sue caratteristiche più avanzate quello di sviluppare anche l'attuazione della manovra autonomamente senza l'ausilio del comandante.

Tale sistema sarà formato inizialmente da una parte decisionale, necessaria per effettuare la pianificazione della rotta tenendo conto dei vincoli ambientali, e avrà poi come uscita il percorso designato, che sarà l'input del sistema di controllo come descritto in [4], capace di gestire i vari attuatori nelle diverse condizioni.

Il sistema di supporto oggetto di questo elaborato permetterà ad una qualsiasi nave o imbarcazione di raggiungere autonomamente l'ormeggio, considerando eventuali ostacoli in base a quella che è la geometria portuale.

1.1. Literature review

Il problema dell'attracco automatico è molto discusso in letteratura e sono presenti numerosi studi che propongono diverse soluzioni al problema. Alcuni esempi sono gli studi condotti da Nguyen e Im [7] dove propongono soluzioni che prevedono l'utilizzo delle Fuzzy logic [8] o Artificial Neural Network [9] per risolvere il problema dell'attracco a velocità ridotta, cercando così di eliminare il più possibile il fattore umano, prendendo in considerazione sia la geometria portuale che le condizioni ambientali. Va tenuto conto che questi studi sono ancora in fase di sviluppo e si addicono maggiormente a situazioni in cui la manovra di ormeggio è pressoché sempre la stessa (come, per esempio, per le navi traghetto o crociere). Tra aspetti che sono coinvolti nei sistemi di attracco automatico, un focus è stato posto nel problema della pianificazione della rotta e nella selezione di un appropriato algoritmo per la sua risoluzione. Il problema verrà successivamente trattato nel paragrafo 2 e risulta essenziale per assicurare che il processo di gestione della navigazione funzioni in modo ottimale.

In letteratura ci sono diversi studi che analizzano l'utilizzo dei vari algoritmi per la pianificazione della rotta, come quello condotto da Sariff e Buniyamin [10] dove vengono confrontati gli algoritmi. Di seguito in

Tabella 1 sono riportati alcuni dati estratti dalla pubblicazione [10] :

Workspace	Type of Algorithm	Strength & Weakness
Polyhedral object moving to avoid polyhedral objects in a static environment	Visibility Graph algorithm + A* search algorithm	Strength: Visibility graph will find collision free path while A* search algorithm will find optimum path and allow use of efficient heuristic information Weakness: Not necessary for dynamic obstacles that always changes
Static Environment cluttered with obstacles	Quadtree Representation + A* search algorithm	Strength: Dynamic path planning. Weakness: Does not localize the effect of obstacles on the representation, Algorithms are time consuming because A strategy expands a lot of nodes.
Static Environment with obstacles cluttered	Modified A* Search algorithm by using a trial vectors that span several cells to help in the planning	Strength: Speed of searching algorithm is increased dramatically compare to traditional A* search algorithm
Time varying and unknown environment	Genetic algorithm	Strength: Works for complex problem that traditional algorithms cannot find a satisfactory solution. Can work in time varying and unknown environment.

Tabella 1 – Confronto tra gli algoritmi

Questi dati permettono di capire quali sono i punti di forza e debolezza di ogni algoritmo nelle loro applicazioni da cui emerge che per un ambiente statico con ostacoli disordinati, la scelta dell'algoritmo A* [11] risulta essere la migliore in termini di costo della traiettoria con l'inconveniente che, nel caso si abbiano ambienti di lavoro molto vasti, il tempo computazionale aumenti notevolmente. Questo verrà utilizzato in questo elaborato e quindi trattato in modo approfondito nel paragrafo 2.1.

Da citare inoltre, l'algoritmo Dijkstra [12], questo risulta simile all'A* ma si basa su una scelta casuale di punti rendendo il tutto meno stabile e con un costo computazionale maggiore.

Altri tipi di algoritmi possono essere "The visibility graph" [13] che mostra i punti visibili da ogni vertice generando il percorso con l'unione di questi segmenti o il "vector based

approach” [14] che si basa sull’utilizzo di vettori per la creazione della traiettoria: se un vettore passa per una zona non consentita allora viene creato un altro arrivo intermedio e viene collegato con nuovi vettori.

Il focus è stato poi spostato sui sistemi di assistenza alla manovra presenti sul mercato. Attualmente, diversi sistemi sono stati presentati dai vari produttori e generalmente permettono di controllare l’imbarcazione non fornendo direttamente i setpoint per attuatori e motori, ma controllando direttamente i 3 gradi di libertà tramite il cosiddetto “joystick”. Ad esempio, una soluzione è quella presentata da Volvopenta [15] dove grazie all’installazione del sistema “*joystick control*” si è capaci di manovrare con una mano l’imbarcazione riuscendo a regolarne anche la velocità.

Un'altra soluzione non ancora sul mercato viene proposta da Rolls-royce/MTU [16] dove con “*mtu-nautiq-copilot*” si propone un sistema in grado di dare assistenza alla navigazione grazie alla localizzazione di oggetti, ostruzioni e traffico utilizzando una serie di sensori marini e carte nautiche digitali. In particolare, questo sistema permette di scegliere tra tre diverse modalità:

- modalità guida: viene emessa un’indicazione visiva e un avviso acustico di potenziali conflitti di transito con imbarcazioni o altri ostacoli (come, per esempio, la bassa profondità);
- modalità di guardia: oltre le indicazioni visive e gli avvisi acustici, nel momento in cui non viene eseguita nessuna azione dagli operatori, il sistema assume il controllo per evitare situazioni pericolose;
- modalità viaggio: c’è un controllo autonomo di un viaggio pianificato dall’inizio alla fine con molti comportamenti e funzionalità integrati sviluppati appositamente per supportare una serie diversificata di operazioni in acqua;

Durante questa fase di ricerca sono emersi anche altri progetti degni di nota e portati avanti da industrie ed università. Uno di questi è il progetto “X-Self” [17] proposto da Xenta. Questo consiste in un sistema d’ormeggio autonomo in grado di far entrare e uscire un’imbarcazione dal suo posto barca senza alcun intervento umano.

Un altro progetto è quello portato avanti dalla Norwegian University of Science and Technology (NTNU) [18] dove è stata costruita un’imbarcazione totalmente autonoma per il trasporto passeggeri [18]. Questa imbarcazione, chiamata milliAmpere 2 è completamente elettrica ed è stata messa in prova nei canali della città di Trondheim lo scorso settembre.

In base ai dati analizzati finora, non è presente sul mercato un sistema di assistenza all’ormeggio che sia in grado di effettuare la manovra in modo autonomo e in sicurezza, andando così a ridurre quello che è il rischio di collisione dovuto al fattore umano.

Pertanto, si ritiene utile fare uno studio per approfondire l’argomento e progettare un sistema di assistenza alla manovra autonomo da un punto di vista sia esecutivo, ossia dell’esecuzione della manovra, sia decisionale, cioè scegliere qual è la manovra migliore da effettuare, così da poter ridurre ulteriormente il numero di sinistri marittimi.

2. SVILUPPO DEL SISTEMA AUTOBERTHING

Quando ci si avvicina ad un sistema di *autobertthing*, la prima difficoltà che si incontra è quella di creare, in modo autonomo, una traiettoria descritta da una polilinea descritta tramite waypoint, opportunamente posizionati in modo da evitare collisioni tenendo conto di quelle che sono le caratteristiche di manovra dell'unità e capace di unire un punto di partenza ad un punto di arrivo.

In Figura 3 è riportato uno schema a blocchi di quello che sarà il lavoro complessivo di questo elaborato in modo da comprendere al meglio quali sono i temi sviluppati.

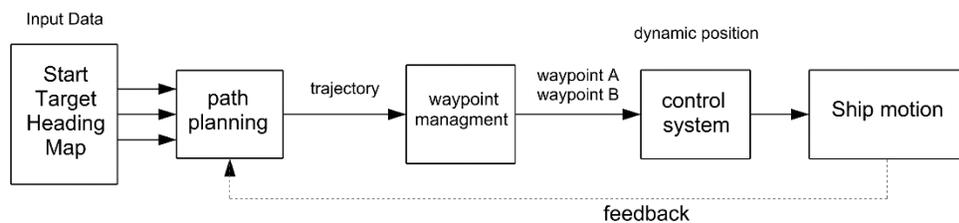


Figura 3 - Schema a blocchi del sistema di autobertthing

Come si è detto la prima difficoltà riguarda il *path planning*, ossia un sistema per cui avendo determinati dati in input come posizione iniziale, finale, angolo di prua e mappa, è in grado di produrre una traiettoria che, tramite waypoint, è in grado di collegare la partenza con la destinazione. Successivamente poi, vengono scelti in modo progressivo, i punti da seguire e di conseguenza, tramite il sistema di controllo, vengono fornite tutte le informazioni ai sistemi di manovra per poter seguire la traiettoria.

A titolo di esempio dunque, è riportata in Figura 4 uno schema rappresentativo che permette di capire qual è il nostro problema riguardo il *path planning* e come fare per risolverlo. Il grigliato rappresenta uno spazio generico in cui vogliamo, partendo da una data posizione, arrivare alla destinazione scelta. Il rettangolo blu rappresenta il robot, ossia il nostro "veicolo" o "imbarcazione" che deve raggiungere l'ovale rosso (target) evitando tutti gli ostacoli (cerchi neri).

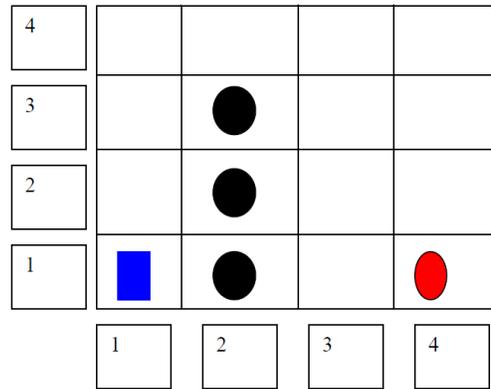


Figura 4 - Schema rappresentativo

Il problema, quindi, prevede la ricerca di una traiettoria ottimale che permette di collegare il punto di partenza a quello di arrivo, per fare ciò è necessario dover raffigurare lo spazio fisico con uno spazio virtuale, a tale scopo si è scelta una schematizzazione di tipo matriciale, trattata poi nel paragrafo 0 che permette di raffigurare per ogni cella una data superficie di spazio reale.

L'idea è quella di esaminare un intorno quadrato del nostro veicolo dove è possibile spostarci fino a raggiungimento del target.

È importante inoltre tener conto non solo della posizione, ma anche dell'orientazione della nave e di una opportuna distanza di sicurezza che ci permette di non collidere con banchina o ostacoli fissi.

2.1. Algoritmo A*

Il sistema di autoberthing, quindi, ha come punto di partenza l'algoritmo A* [19], ossia un algoritmo di ricerca formulato sui grafi pesati, progettato per risolvere la traiettoria più breve, in base alla schematizzazione utilizzata, tra il punto iniziale e quello finale. Questo è possibile grazie ad una diramazione di traiettorie che hanno origine nel punto iniziale e che le estende, una per volta, fino a che il criterio non è soddisfatto.

Numerosi studi, infatti, utilizzano l'A* per la pianificazione del percorso di sistemi automatizzati, ne è un esempio quello condotto da Tang G [20], dove viene proposto un sistema autonomo, in grado di generare una traiettoria che sia il più breve possibile da affiancare ad un veicolo a guida automatizzata (AGV), così da evitare problemi di elevato numero di nodi, grandi distanze e ampio angolo di sterzata.

Ad ogni iterazione, l'A* necessita di determinare quale traiettoria estendere. Ciò è possibile grazie al computo del costo della traiettoria e del costo necessario per raggiungere il target dove per costo si intende una funzione che si vuole minimizzare (per esempio spazio, tempo, spesa etc.).

Dunque, l'algoritmo seleziona la traiettoria che minimizza la funzione in Equazione 1:

Equazione 1 – Equazione A*

$$f = g + h$$

dove si hanno conseguentemente tre parametri:

- g : costo effettivo dello spostamento dalla cella iniziale alla cella in esame, è la somma di tutte le celle esaminate da quando si è lasciata quella di partenza.
- h : conosciuto anche come valore euristico, è il costo stimato del movimento tra la cella in esame e quella finale.
- f : è la somma dei due valori g e h

L'algoritmo dunque, prende le decisioni tenendo conto del valore f : seleziona le celle con il più basso valore di f e si espande da quel punto. Il processo poi continua fino a quando non si raggiunge l'arrivo.

Una volta raggiunta la destinazione, come conseguenza, avremo che il valore euristico di h sarà uguale a 0 e dunque f non sarà altro che g ossia il costo effettivo del percorso.

2.2. Implementazione del codice matlab

Per la stesura del codice è stata utilizzata come base di partenza un'implementazione dell'algoritmo A* disponibile presso il *Matlab central* [21] che permette di ottenere il tragitto più breve per raggiungere il punto di arrivo (target).

Il file di partenza è stato opportunamente modificato in modo ottenere risultati congruenti a quelli richiesti dal problema di questo elaborato.

La funzione f quindi è stata valutata in ordine di distanza, ciò vuol dire che, come risultato, si avrà la traiettoria più breve in termini di spazio. Si avrà dunque che il valore g sarà calcolato come la somma dei singoli spostamenti, mentre il valore euristico h , sarà calcolato come distanza tra due punti.

Nel problema sono state prese in considerazione anche le geometrie e quindi gli ingombri, in modo da poter controllare eventuali "collisioni", inoltre è stata implementata anche l'orientazione dell'imbarcazione.

Per effettuare tutti i controlli quindi, ed avere una maggiore accuratezza del problema reale, è necessario considerare le geometrie sia portuali che dell'imbarcazione.

Per quanto riguarda la geometria portuale, in Figura 5 sono raffigurate tre geometrie quadrata di lato 5,2 m dove sono stati inseriti degli ostacoli in accordo con la vasca disponibile in laboratorio per effettuare le prove.

Tale geometria è stata schematizzata come una matrice dove, per ogni coppia di indici riga-colonna, viene scritto un valore a seconda di cosa rappresenta:

- -1 per gli ostacoli
- 0 per il target
- 1 per il veicolo

Le celle inoltre sono la rappresentazione dello spazio, quindi, a parità di dimensioni complessive, minore sarà lo spazio fisico rappresentato dalla singola cella, maggiore sarà il numero di celle necessarie e di conseguenza maggiore sarà l'accuratezza della rappresentazione.

A tale scopo infatti, ogni geometria è stata schematizzata con 3 diverse grandezze: quella più grossolana nella quale ogni casella rappresenta uno spazio di 10 cm^2 , una seconda dove ogni casella è 5 cm^2 e infine una terza per cui ogni casella raffigura 1 cm^2 . Queste diverse discretizzazioni sono utile al fine di valutare la relazione tra precisione, tempi di calcolo e dimensione della cella.

In questo spazio sono state create inizialmente tre raffigurazioni geometriche diverse, in modo da avere una più ampia possibilità di scenari.

Nello specifico, a titolo d'esempio in Figura 5 è raffigurato:

- uno scenario sgombro, chiamato geometria semplice, ossia senza alcun tipo di ostacolo in modo da poter avere un primo approccio al problema.
- una geometria intermedia con qualche ostacolo che può simulare un pontile o molo
- con una geometria complessa con un maggior numero di ostacoli che possono raffigurare sia pontili o sia delle imbarcazioni all'ormeggio.

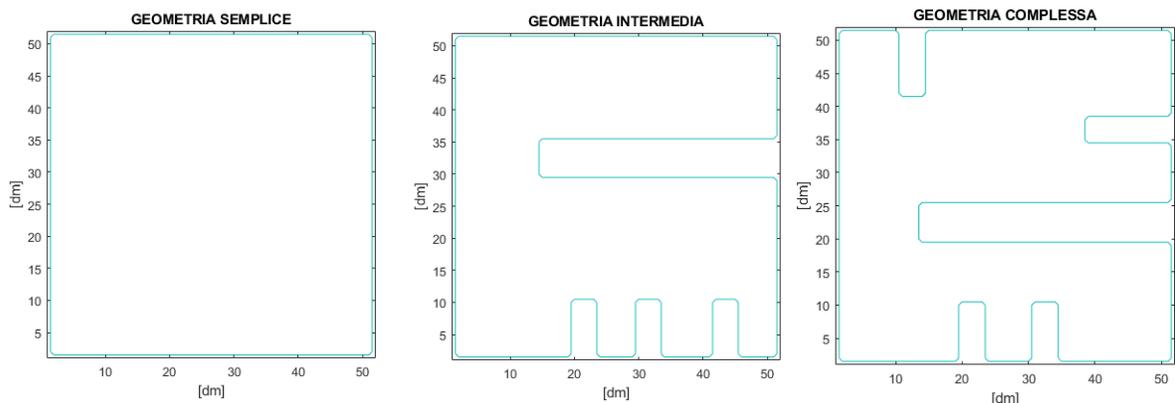


Figura 5 - Geometrie portuali

Anche per il veicolo, raffigurante l'imbarcazione, è stato necessario considerare la sua reale geometria in quanto l'algoritmo, prendendo in esame una sola cella per volta, non riesce a tenere conto dell'intero spazio occupato dall'imbarcazione.

A tale scopo è stata creata una geometria ellittica che meglio rappresenta la figura della linea d'acqua.

È stata utilizzata la formula in equazione:

Equazione 2 – Geometria ellittica

$$\begin{cases} x = B \cos(\alpha)/2 \\ y = L \sin(\alpha)/2 \end{cases}$$

dove B e L rappresentano rispettivamente la lunghezza al galleggiamento (Lwl) e larghezza al galleggiamento (Bwl) mentre α è la variabile dell'angolo e va da 0 a 2π .

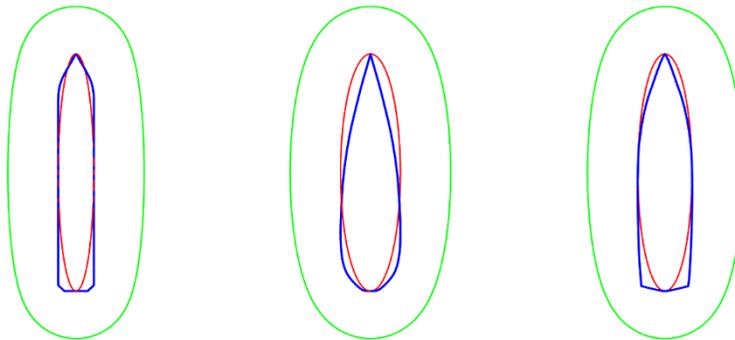


Figura 6 - Approssimazione della figura di galleggiamento

In Figura 6 è riportata la sovrapposizione di un'ellisse a varie linee d'acqua: partendo da sinistra, si ha la figura di galleggiamento di una nave da crociera, una barca a vela e infine un motoscafo. In blu abbiamo il contorno della figura reale mentre in rosso quella dell'ellisse costruito con le dimensioni principali. In verde invece si ha invece il profilo ellittico maggiorato della distanza di sicurezza d_{safe} pari al 20% della lunghezza al galleggiamento.

L'ellisse viene poi suddiviso in cento punti con le rispettive coordinate x e y in modo da convertire una geometria continua in una discreta.

In Figura 7 viene riportato un diagramma di flusso che raffigura lo sviluppo dell'algoritmo A* nel codice Matlab.

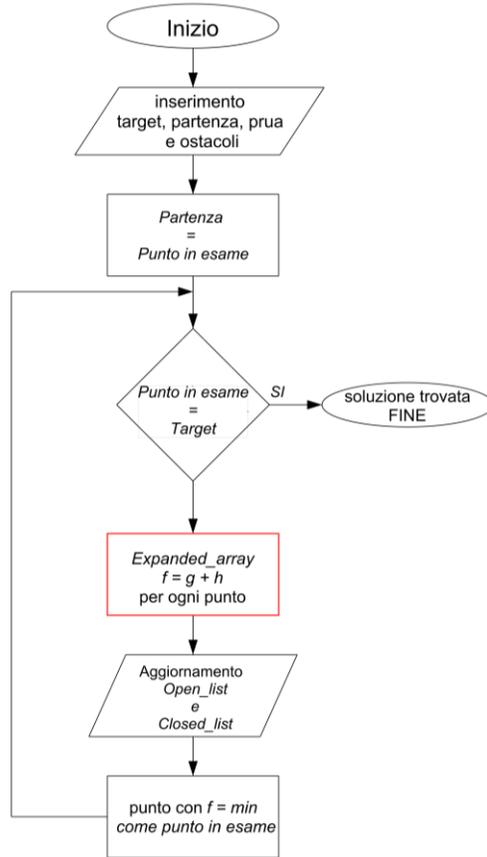


Figura 8 - Diagramma di flusso dell' algoritmo

Il codice necessita come dati di input le coordinate dei punti di arrivo, partenza e angolo di prua insieme ad eventuali ostacoli. Inizialmente l' algoritmo analizza la cella corrispondente al nodo iniziale, considerandolo quindi come "punto in esame".

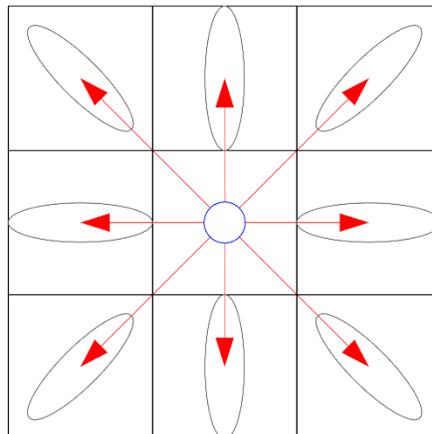


Figura 7 - Intorno quadrato ed orientazione figura di galleggiamento

Da qui vengono analizzati tutti i punti presenti nell'intorno quadrato del punto in esame come rappresentato in Figura 8, ovvero con angolazioni che partono da 0 e variano ogni 45 gradi.

Una volta ottenute le coordinate dei punti, bisogna controllare se corrispondono con quelle degli ostacoli e nel caso scartarli. Il codice Matlab è stato modificato opportunamente nel processo logico evidenziato in rosso della Figura 7, dove, in ogni punto candidato, viene implementata la geometria ellittica, considerando sia le dimensioni dell'imbarcazione sia l'orientazione del segmento che congiunge i due punti (Figura 8) in modo da tener conto non solo della singola cella, ma anche di un intorno che considera la figura al galleggiamento dell'imbarcazione. È stata stabilita poi una distanza di sicurezza d_{safe} che permette di verificare per ogni punto dell'intorno se l'imbarcazione si avvicina troppo ad un ostacolo in modo da scartarlo.

I punti dell'intorno poi, una volta eliminati quelli non idonei, vengono trascritti insieme ai rispettivi valori di g, h e quindi f nell'*expanded_array* ossia una matrice che contiene i dati dei punti di tutti i possibili collegamenti dal punto in esame.

Successivamente, i punti vengono copiati nell' *OPEN_list* tenendo conto della cella "parente", ossia la cella a cui sono collegati. L' *OPEN_list* sarà dunque una matrice che contiene tutte le celle da esaminare e i rispettivi parenti. Nel caso in cui una cella dovesse risultare ripetuta più volte, allora verrebbe scartata quella con il valore f maggiore.

Una volta analizzato il "punto in esame" e aggiornato l'*OPEN_list*, esso viene spostato nella *CLOSED_list* ossia la lista di tutti i punti già presi in esame.

Il passo successivo è quindi quello di analizzare il punto presente nell' *OPEN_list* con il valore di f minore e aggiornare l'*expanded_array*, tenendo presente però di quelli già analizzati e quindi presenti nella *CLOSED_list*. Se il punto in esame è quello di arrivo, allora abbiamo la soluzione, altrimenti si continua in modo iterativo.

I risultati finali poi sono stati opportunamente modificati, come illustrato nel paragrafo seguente, in modo tale da ottenere il minor numero possibile di punti di controllo (waypoints) così da ridurre il tempo di esecuzione della manovra.

2.3. Legge della velocità

In base al funzionamento dell'autoberthing e nello specifico della parte di controllo degli attuatori, per questa applicazione è stato ipotizzato che l'imbarcazione debba raggiungere tutti i waypoints con un profilo di velocità trapezoidale e un angolo di prua (heading) costante per ogni tratta che collega due waypoint consecutivi.

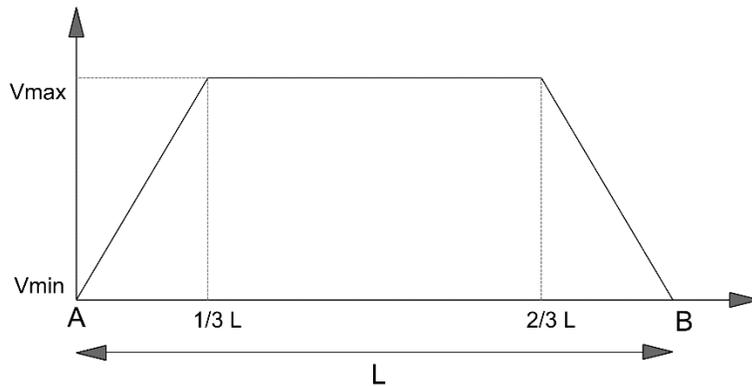


Figura 9 - Profilo di velocità trapezoidale

Il profilo di velocità trapezoidale come descritto in Figura 9 ci permette di percorrere il tragitto da un punto iniziale (A) ad un punto finale (B) variando la velocità in funzione della distanza percorsa tra i due punti. Una volta arrivati a destinazione, se il punto B corrisponde a quello di arrivo allora la velocità sarà impostata su 0.

Dunque, avremo una legge di velocità così come descritta in equazione:

Equazione 3 – Velocità desiderata

$$\left\{ \begin{array}{l} v_{set} = (v_{max} - v_{min}) \frac{L_i - d}{L_i} + v_{max} \quad \text{per il primo tratto} \\ v_{set} = v_{max} \quad \text{per il secondo tratto} \\ v_{set} = v_{max} \left(\frac{-d + L_f}{L - L_f} + 1 \right) \quad \text{per il terzo tratto} \end{array} \right.$$

Dove:

v_{set} è la velocità desiderata

v_{max} è il valore di velocità massimo che si assume per questa applicazione

L_i è 1/3 della distanza totale

L_f è 2/3 della distanza totale L

L è la distanza tra i due waypoint.

È stata aggiunto inoltre un vincolo per cui se la lunghezza del tratto da percorrere è inferiore a 2 volte la lunghezza nave allora il valore v_{max} sarà diminuito del 30% in modo evitare cambiamenti bruschi di segnale in ingresso.

In Figura 10 è riportato un esempio di input di velocità inviato poi al sistema.

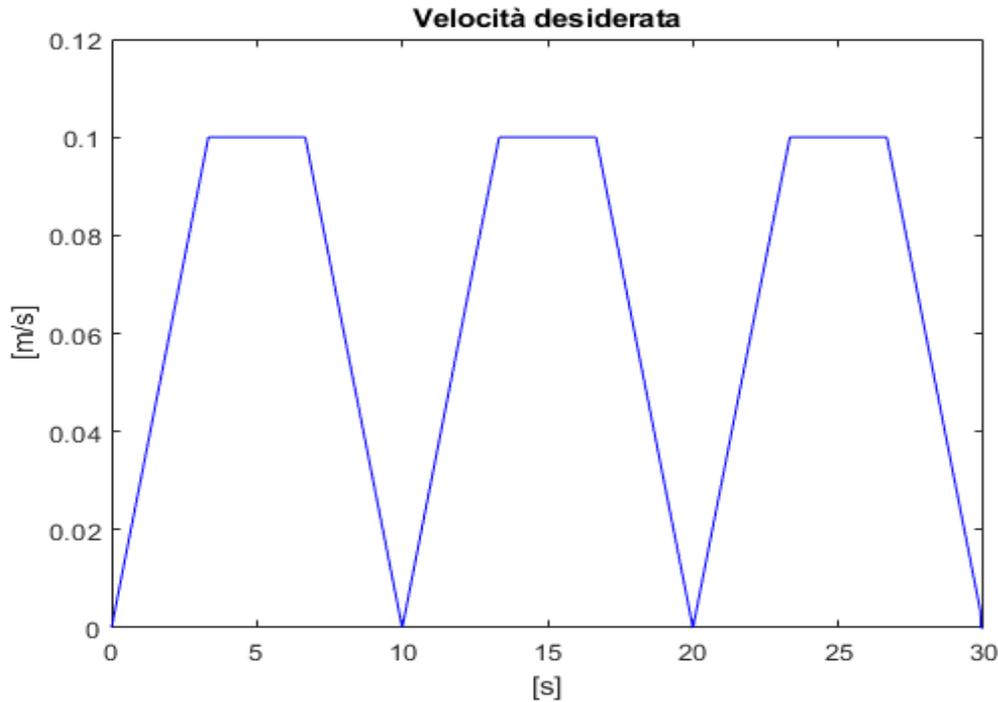


Figura 10 - Input di velocità

Il sistema riceve in ingresso la velocità di avanzamento desiderata, la quale segue la legge descritta in Equazione 3. Nello specifico, all'inizio della navigazione la velocità cresce gradualmente fino a raggiungere un valore costante. Successivamente, quando l'imbarcazione si avvicina al waypoint, la velocità comincia a diminuire progressivamente fino a quando non raggiunge una distanza definita come d_{skip} , ovvero pari al 10% della lunghezza della nave. La variabile d_{skip} rappresenta il valore per cui si considera raggiunto il waypoint. Una volta raggiunta tale distanza l'imbarcazione passerà al waypoint successivo del percorso predefinito.

2.4. Ottimizzazione della rotta

Come risultato dell'algoritmo A* si hanno una serie di punti che permettono di arrivare, passo dopo passo, alla generazione di un percorso senza collisioni con ostacoli fissi.

D'altro canto però c'è da considerare che per il principio di funzionamento dell'algoritmo si possono avere spostamenti solo lungo gli assi cardinali e le loro bisettrici. Si hanno dunque 9 possibili direzioni diverse contro tutte le direzioni possibili che può assumere una qualsiasi imbarcazione. Infatti, è da notare che l'ottimizzazione della rotta risultante dal codice non è effettivamente la descrizione della rotta più breve in assoluto ma quella che rispetta le condizioni di movimento imposte dalla schematizzazione della geometria e dall'algoritmo.

Quindi, concordemente con la legge delle velocità, si prevede che, minore sia il numero di punti da seguire e più veloce sarà l'esecuzione della manovra. Per questo motivo è opportuno ottimizzare la rotta, eliminando tutti i punti superflui e per quanto possibile riducendo il percorso.

È stato perciò implementato, come descritto in Figura 11, il codice Matlab con un algoritmo che, partendo dal punto di inizio, si collega a quello successivo considerando la geometria ellittica dell'imbarcazione e l'angolo di prua che c'è tra i due punti. Qualora nel tratto esaminato non ci siano collisioni, allora si passa al punto successivo e si effettua di nuovo il controllo. L'algoritmo si interrompe nel momento in cui si ha una collisione, posizionando il waypoint al punto precedente rispetto a quello per cui si ha la collisione. L'algoritmo riparte poi posizionando il waypoint appena aggiunto come punto iniziale e prosegue collegandolo a quelli successivi.

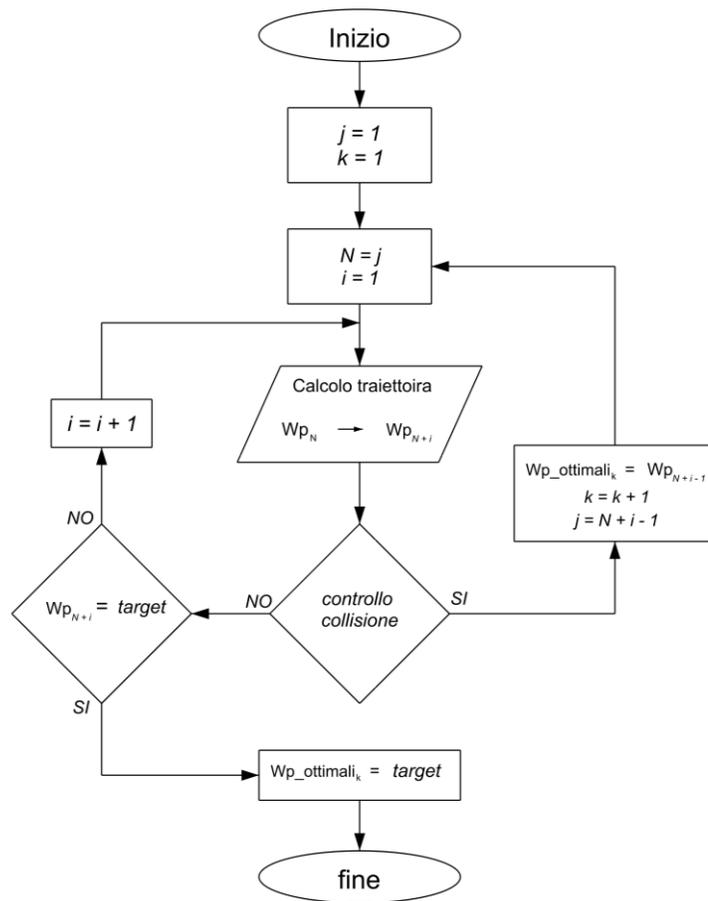


Figura 11 - Diagramma di flusso dell'ottimizzazione del numero dei waypoints

Come già anticipato nella Figura 11 è rappresentato il flusso logico che gestisce il percorso dei *waypoint*. l'indice "*j*" viene utilizzato per inizializzare il processo mentre il contatore dei *waypoint* è indicato con "*i*" e viene incrementato ad ogni ciclo finché non si verifica una collisione. In tal caso, il contatore viene reimpostato a 1. La variabile "*N*" tiene traccia del *waypoint* raggiunto ogni volta che si verifica una collisione. Il contatore "*k*" viene utilizzato per assegnare il *waypoint* nella matrice *Wp_ottimale* e viene aggiornato ogni volta che si verifica una collisione.

Questo controllo non è però sufficiente, in quanto non assicura che non ci siano collisioni dovute alla rotazione, dato che controlla la posizione della nave solo nello stato di arrivo e partenza dal *waypoint*. È stata quindi implementata un'ulteriore funzione per controllare se, in quel determinato *waypoint*, è possibile ruotare l'imbarcazione dall'angolo di prua con cui si arriva, fino all'angolo di prua con cui si parte per il punto successivo. Qualora questo requisito non fosse soddisfatto, allora l'algoritmo prenderà il *waypoint* precedente e effettuerà nuovamente i controlli.

A titolo d'esempio in Figura 12 è riportato quanto detto prima. In viola è raffigurato il percorso suggerito dall'algoritmo (*path*), ossia il percorso più breve tenendo conto dei vincoli di movimento imposti dalla geometria e dall'algoritmo, mentre in nero quello ottimizzato. Nella figura sono riportati in rosso anche le ellissi rappresentativi della linea d'acqua dell'imbarcazione.

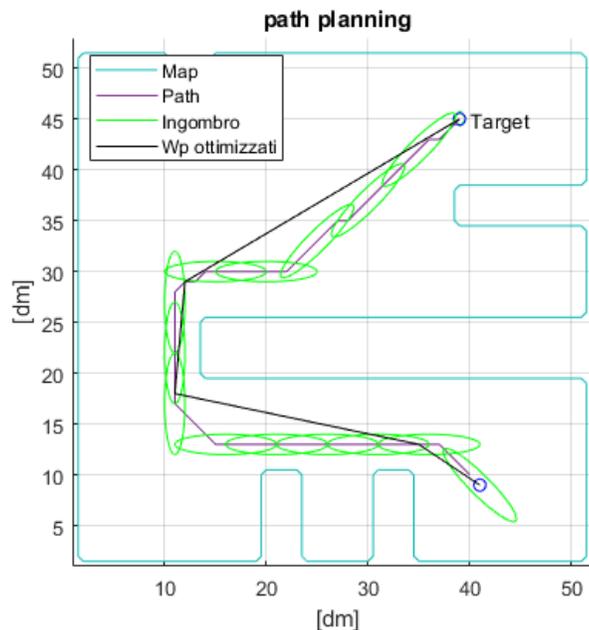


Figura 12 - Rotta ottimizzata

Dopo aver ottenuto la rotta ottimizzata e ipotizzando di avere la figura di galleggiamento effettiva della nostra imbarcazione, è possibile effettuare un ulteriore controllo in modo da essere sicuri che la manovra venga eseguita correttamente rispettando le geometrie.

Di seguito, in Figura 13 ne è riportato un esempio.

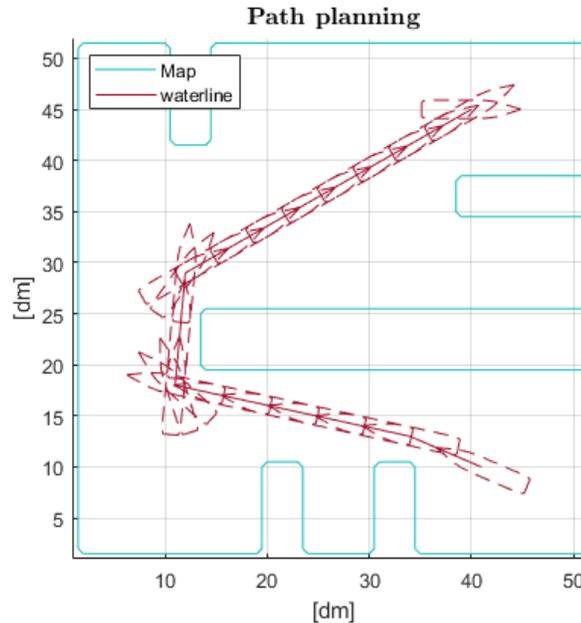


Figura 13 - Controllo collisioni con figura di galleggiamento

Una volta ottenuti quindi i *waypoints* da seguire sono stati inizialmente scritti in forma matriciale dove ogni riga contiene le coordinate x e y del punto desiderato espresse nel sistema di riferimento solidale con il porto e il valore, espresso in radianti, dell'angolo di prua anch'esso espresso rispetto al sistema di riferimento fisso.

Il sistema di *path planning* descritto, con una schematizzazione della geometria dove ogni casella raffigura $1 dm^2$ impiega, con un normale calcolatore, circa 2 secondi per trovare una soluzione nella geometria scelta. Volendo aumentare l'accuratezza del problema e passare ad una schematizzazione a $5 cm^2$, il tempo di calcolo dello stesso problema risulta essere circa 30 secondi, mentre per la schematizzazione di $1 cm^2$ il tempo sale a circa 20 minuti. Questo significa che il tempo impiegato dal sistema nel trovare un percorso per raggiungere una determinata destinazione aumenta esponenzialmente in base al tipo di accuratezza della rappresentazione della geometria.

Schematizzazione	Tempo computazionale
$1 dm^2$	$\sim 2 s$
$5 cm^2$	$\sim 30 s$
$1 cm^2$	$\sim 20 min$

3. GUIDANCE

Il sistema di *autoberthing*, in generale, è formato dalla *guidance*, dal *control* e dal modello nave così come descritto in Figura 14.

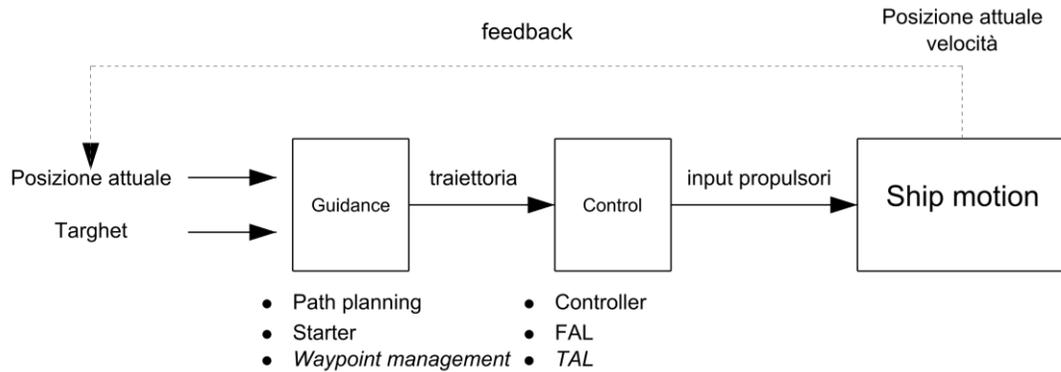


Figura 14 - Diagramma funzionamento autoberthing

Così come descritto in [22], il sistema di *guidance* calcola continuamente la posizione e la velocità desiderate di un'imbarcazione da utilizzare dal sistema di controllo del movimento (*control*). Questi dati vengono solitamente forniti al sistema di navigazione.

In molti casi, vengono utilizzate tecniche avanzate di ottimizzazione per calcolare la traiettoria o il percorso ottimale per l'imbarcazione da seguire. Ciò potrebbe includere funzioni sofisticate come l'ottimizzazione del carburante, la navigazione a tempo o spazio minimo, l'indirizzamento meteorologico, l'evitamento di collisioni.

Quindi il compito principale della *guidance* è quello di garantire che il sistema di controllo della nave segua la traiettoria desiderata. Questo viene fatto attraverso l'utilizzo di tre componenti principali: il *path planning*, lo *starter* e il *waypoint management*.

Il *path planning*, come abbiamo visto, è il processo di pianificazione della rotta da seguire, che tiene conto di eventuali ostacoli e limitazioni.

Lo *starter* è responsabile dell'avvio, quando necessario, del *path planning* in modo da non ripetere processi inutili e dispendiosi.

Il *waypoint management*, infine, si occupa di trasferire in modo adeguato le informazioni di rotta e velocità al sistema di controllo del movimento, in modo che quest'ultimo possa regolare la posizione della nave in base alle istruzioni ricevute. Per farlo, il *waypoint management* riceve *feedback* dalla posizione attuale della nave, che vengono utilizzate per adattare la rotta e, in caso di necessità, tramite segnale da parte dello *starter*, ripianificarla.

Come output della *guidance* si avrà il vettore η_{des} , un vettore colonna espresso come coordinate x e y della posizione desiderata e angolo di prua ψ nel sistema di riferimento

inerziale e il vettore v_{des} ossia il vettore descritto con le velocità espresse nel sistema di riferimento solidale con la nave.

Come output si avranno quindi due vettori principali, η_{des} , e v_{des} , che vengono utilizzati dal controllore per regolare la posizione e la velocità della nave. Il vettore η_{des} è un vettore colonna che descrive la posizione desiderata della nave in termini di coordinate x e y , nonché l'angolo di prua ψ . Queste informazioni vengono poi utilizzate dal controllore per regolare la posizione della nave in modo da seguire la traiettoria desiderata. Il vettore v_{des} , d'altra parte, descrive le velocità desiderate della nave espresse nel sistema di riferimento solidale con la nave.

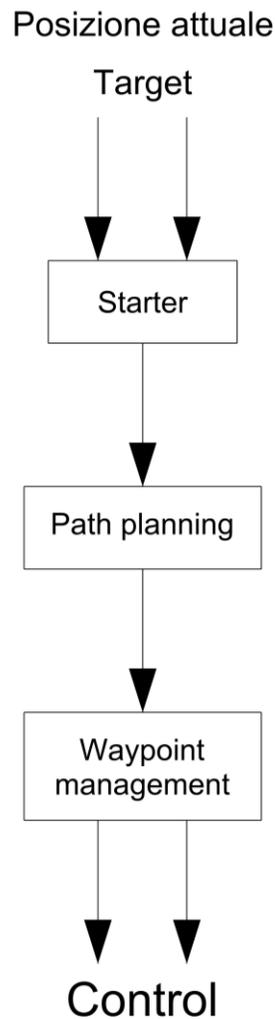


Figura 15 - Diagramma di flusso guidance

In sintesi, come mostrato in Figura 15 la *guidance* ha il compito di comunicare al sistema di controllo del movimento le coordinate e la velocità da seguire in modo da poter avanzare lungo la traiettoria desiderata in funzione della posizione attuale ricevuta come *feedback*.

3.1. Starter

Il sistema di *starter*, come già anticipato, ha la funzione di avviare inizialmente il processo di *path planning* una volta inserite le coordinate del punto di destinazione. Inoltre, il sistema include un ulteriore controllo per verificare se ci sono eventuali scostamenti significativi dalla traiettoria pianificata, utilizzando i *feedback* ricevuti dal modello nave sulla posizione attuale. In caso di scostamenti significativi, il sistema riavvierà il processo di pianificazione del percorso utilizzando la posizione aggiornata della nave. Questo è importante per garantire che la nave si mantenga sulla traiettoria pianificata, evitando possibili ostacoli o pericoli lungo il percorso. Il sistema di *starter* è essenziale per la sicurezza e l'efficienza delle manovre della nave.

Di seguito, in Figura 16, è raffigurato lo schema a blocchi di quanto descritto, dove per d_max si intende la distanza massima che si può avere come scostamento dalla traiettoria desiderata.

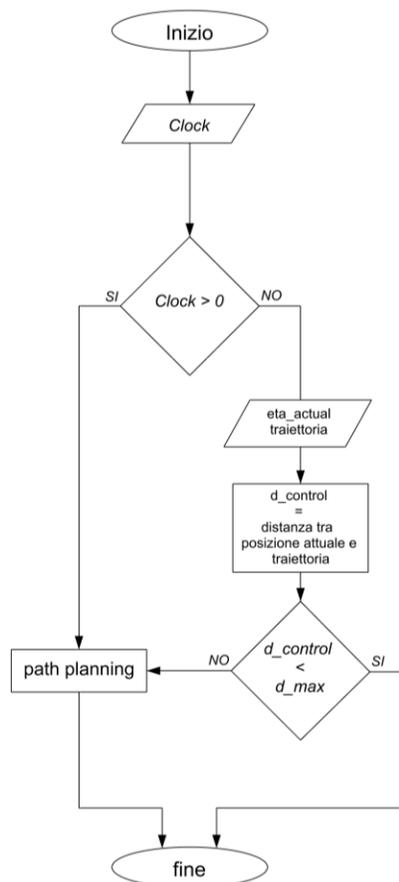


Figura 16 - Schema a blocchi starter

3.2. Waypoint management

Il *waypoint management* ha il compito di gestire e coordinare le informazioni relative all'avanzamento durante la manovra d'ormeggio. In particolare, esso è responsabile di comunicare con il resto del sistema le coordinate del waypoint da seguire e la velocità desiderata V_{set} . Questo sistema, utilizzando i *feedback* del modello nave, sceglie il waypoint da raggiungere tenendo in considerazione quello precedente, in modo da poter tracciare una linea retta tra i due che rappresenta la traiettoria da seguire e fornisce informazioni per stabilire la velocità desiderata come indicato nel paragrafo 3.3. Quando la nave sarà entro una certa distanza dal waypoint considerato, il sistema passerà automaticamente a quello successivo.

In Figura 17 è mostrato come avviene la gestione del *waypoint management*, in particolare abbiamo in blu la linea d'acqua dell'imbarcazione, in nero la traiettoria da seguire, in verde il profilo di velocità desiderato ed in rosso è raffigurata la distanza d_{skip} per cui si passa al waypoint successivo.

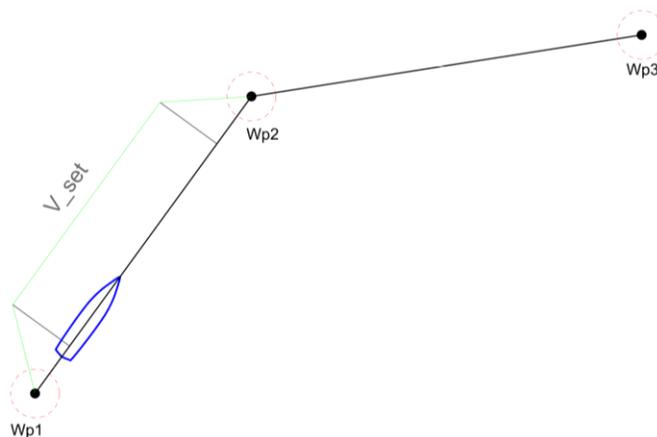


Figura 17- Waypoint management

3.3 path planning

Il path planning invece è il processo per cui, una volta stabilite le condizioni iniziali e finali, calcola la traiettoria da seguire così come detto nel capitolo 2.

4. SISTEMA DI CONTROLLO PER POSIZIONAMENTO DINAMICO

Una volta ottenuti i dati in uscita dalla *guidance*, ossia i vettori η_{des} e v_{des} , è necessario elaborarli in modo da ottenere i *setpoint* per il sistema di propulsione e per gli organi di manovra.



Figura 18 - Modellino Tito neri

In questo elaborato, si è scelto di utilizzare un modellino del rimorchiatore "Tito neri" (Figura 18) fornito di due propulsori azimutali a poppa e un *bow thruster* a prua. Tutti i propulsori sono azionati da motori elettrici, per cui all'uscita del controllore si ha bisogno di un valore di tensione e l'orientamento dei propulsori.

Inoltre, è importante specificare l'allocazione scelta per il modellino in esame, come descritto in [23] dove si ha il propulsore di dritta che provvede alle spinte in senso longitudinale mentre quello di sinistra si occupa, insieme al *bow thruster*, delle spinte laterali. Ciò significa che il propulsore di dritta fornirà la potenza necessaria per muovere il rimorchiatore in avanti o indietro, mentre quello di sinistra e il *bow thruster* forniranno la potenza necessaria per muovere il rimorchiatore lateralmente e permettere la rotazione. In questo modo, il controllore genererà i valori di tensione e orientamento dei propulsori per controllare la velocità e la direzione del rimorchiatore in base ai dati di uscita della *guidance*.

In Figura 19 è schematizzato il layout del sistema di posizionamento dinamico

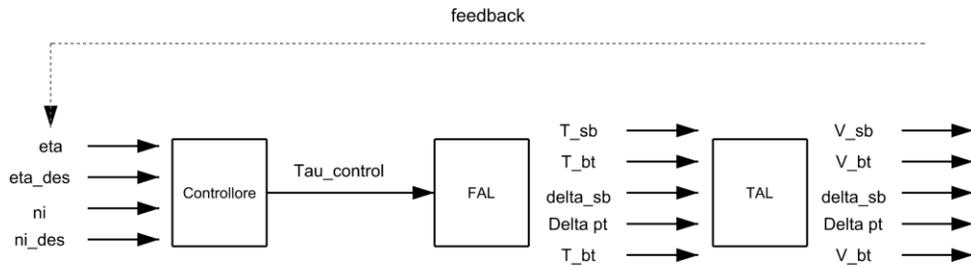


Figura 19 - Layout sistema di posizionamento dinamico

Il sistema di controllo delle navi è composto da diverse sezioni che lavorano insieme per garantire un funzionamento ottimale. In questa applicazione, è presente il *controller*, che elabora i dati in ingresso e genera un segnale che rappresenta le forze necessarie per mantenere la nave in posizione o portarla nella posizione e alla velocità desiderata; un sistema di allocazione delle forze FAL (*force allocation logic*), che elabora il segnale proveniente dal *controller* per ottenere le forze effettive da applicare ai sistemi di propulsione; e un sistema di allocazione spinte TAL (*thrust allocation logic*), che partendo dalle spinte trovate dalla FAL, calcola i corretti *set-points* per ogni attuatore.

4.1. Controller

Nel controllore, come abbiamo detto, vengono elaborati i dati uscenti dalla *guidance*, ossia η_{des} e v_{des} , insieme ai segnali di *feedback* ricevuto dal modello nave.

In particolare, questi dati, insieme ai *feedback* ricevuti dal modello nave di posizione attuale (η) e velocità attuale ($\dot{\eta}$) permettono di calcolare i vettori errore sia per quanto riguarda la posizione che per quanto riguarda la velocità secondo le formule in Equazione 4 nel sistema di riferimento inerziale n .

Equazione 4 – Calcolo vettori errori

$$[\bar{\eta}] = [\eta] - [\eta_{des}]$$

$$[\dot{\bar{\eta}}] = [\dot{\eta}] - [\dot{\eta}_{des}]$$

È importante notare che nel caso della velocità, bisogna effettuare un cambio di base in quanto essa risulta definita nel sistema di riferimento solidale con l'imbarcazione. Per questo motivo, è necessario moltiplicare l'errore per la matrice di rotazione R, che è descritta nell'Equazione 5

Equazione 5 – Matrice di rotazione

$$R(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Una volta che sono stati calcolati gli errori di posizione e di velocità, è possibile calcolare il vettore forza τ così come descritto in [24] con l'Equazione 6:

Equazione 6 – Calcolo vettore forza

$$[\tau_{control}] = K_P \bar{\eta} + K_D \dot{\bar{\eta}} + K_I \int \bar{\eta} - K_{IAW}(\tau_{control} - R^{-1}(\psi)\tau_{del}) d\tau$$

Dove:

$$- [K_P] = \begin{bmatrix} K_{P_x} & 0 & 0 \\ 0 & K_{P_y} & 0 \\ 0 & 0 & K_{P_\psi} \end{bmatrix} \text{ è definito come guadagno proporzionale;}$$

$$- [K_D] = \begin{bmatrix} K_{D_x} & 0 & 0 \\ 0 & K_{D_y} & 0 \\ 0 & 0 & K_{D_\psi} \end{bmatrix} \text{ è definito come guadagno derivato;}$$

$$- [K_I] = \begin{bmatrix} K_{I_x} & 0 & 0 \\ 0 & K_{I_y} & 0 \\ 0 & 0 & K_{I_\psi} \end{bmatrix} \text{ è definito come guadagno integrale;}$$

- $[K_{IAW}] = \begin{bmatrix} K_{IAW_x} & 0 & 0 \\ 0 & K_{IAW_y} & 0 \\ 0 & 0 & K_{IAW_\psi} \end{bmatrix}$ è definito come guadagno dell'antiwindup;

- τ_{del} è un feedback del modello nave e rappresenta la somma delle forze fornite dagli attuatori all'istante precedente.

Una volta che è stato calcolato il vettore $\tau_{control}$, bisogna trasformarlo in modo che le forze siano espresse nel sistema di riferimento solidale con la nave. Per fare questo, bisognerà moltiplicare $\tau_{control}$ per la matrice di rotazione R come in Equazione 7:

Equazione 7 – Vettore forza in base nave

$$[\tau_{control\ nave}] = R(\psi) \tau_{control} = \begin{bmatrix} X_R \\ Y_R \\ N_R \end{bmatrix}$$

4.2.

II

Questo

:

Equazione 8 – Sistema di equilibrio

$$\begin{cases} X_R + T_{SB} = 0 \\ Y_R + T_{PT} + T_{BT} = 0 \\ N_R + x_{PT}T_{PT} - y_{SB}T_{SB} + x_{BT}T_{BT} = 0 \end{cases}$$

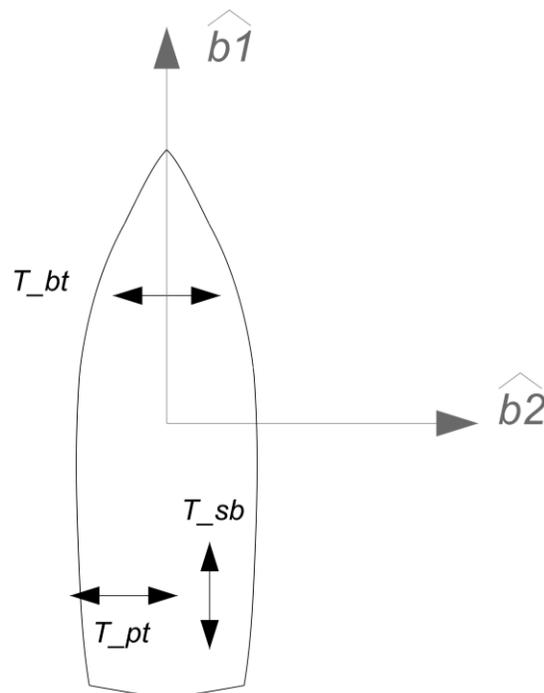


Figura 20 - Allocazione propulsori

Dove

- T_{SB} è la spinta del propulsore di dritta;
- T_{PT} è la spinta del propulsore di sinistra;
- T_{BT} è la spinta del bow thruster;
- x_{PT} è la distanza longitudinale che intercorre tra l'origine del sistema b e il propulsore di sinistra;
- x_{BT} è la distanza longitudinale che intercorre tra l'origine del sistema b e il bow thruster;
- y_{SB} è la distanza trasversale che intercorre tra l'origine del sistema b e il propulsore di dritta;

Risolvendo il sistema avremo quindi i valori di T_{SB} , T_{PT} e T_{BT} che rappresentano le spinte richieste rispettivamente al propulsore di dritta, di sinistra e del bow thruster da mandare al blocco successivo (FAL). Tuttavia, questi valori possono essere sia positivi che negativi a seconda che siano rivolti verso avanti, indietro o destra o sinistra in base ai versori \hat{b}_1 e \hat{b}_2 . Pertanto, per una corretta comunicazione, questi valori vengono modificati in modo che per i propulsori principali di poppa il valore di spinta sia sempre positivo e l'allocazione e quindi l'angolo di orientamento delle eliche cambino.

La selezione dell'angolo di azimutale avviene secondo le relazioni in Equazione 9

Equazione 9 – angolo dei propulsori

$$\delta_{SB} \begin{cases} 180^\circ & \text{se } T_{SB} < 0 \\ 0^\circ & \text{se } T_{SB} > 0 \end{cases}$$

$$\delta_{PT} \begin{cases} 90^\circ & \text{se } T_{PT} < 0 \\ -90^\circ & \text{se } T_{PT} > 0 \end{cases}$$

Dove:

- δ_{SB} è l'angolo di rotazione del propulsore di dritta;
- δ_{PT} è l'angolo di rotazione del propulsore di sinistra;

In questo modo, δ_{SB} sarà di 180° se T_{SB} è negativo e 0° se T_{SB} è positivo. Analogamente, δ_{PT} sarà di 90° se T_{PT} è negativo e -90° se T_{PT} è positivo.

4.3. TAL

Il sistema di allocazione spinte TAL (*thrust allocation logic*) è un sistema che elabora le spinte richieste ad ogni propulsore trovate dal sistema di allocazione delle forze (FAL). Il TAL trasforma poi questi valori in tensioni da dare ai relativi motori elettrici.

Prima di questa trasformazione, bisogna adimensionalizzare il valore della spinta, al fine di semplificare una futura taratura delle forze. Ciò avviene tramite la relazione in Equazione 10:

Equazione 10 - Adimensionalizzazione delle forze di spinta

$$T_{PTAD} = T_{PT} / T_{MAXPT}$$

$$T_{SBAD} = T_{SB} / T_{MAXSB}$$

$$T_{BTAD} = T_{BT} / T_{MAXBT}$$

dove T_{MAX} rappresenta i rispettivi valori massimi di spinta che i propulsori possono fornire. Successivamente, tramite interpolazione lineare viene stabilito il valore della tensione e si ha come output del TAL:

V_{PT} [volt] : tensione motore di sinistra;

V_{SB} [volt] : tensione motore dritta;

V_{BT} [volt]: tensione *bow thruster*.

δ_{PT} [deg]: angolo di allocazione del propulsore di poppa sinistra;

δ_{BT} [deg]: angolo di allocazione del propulsore di poppa dritta;

questi valori vengono poi collegati direttamente al modello nave.

5.TUNING

Per garantire un funzionamento corretto del modello della nave e ottenere una traiettoria desiderata, è fondamentale che il controllo e il modello comunichino in modo impeccabile. In particolare, i set point scelti dal controllore devono essere in grado di guidare gli attuatori della nave affinché segua il percorso prestabilito. Come accennato, il controllore è composto da una prima parte (controller) che utilizza i vettori di errore per generare una forza che rappresenta la forza necessaria per muovere la nave nella direzione desiderata. Tale forza dipende sia dall'errore che dal PID, il che significa che è necessario regolare i coefficienti K_P , K_D e K_I in modo opportuno. In altre parole, i valori delle costanti devono essere adeguati ad assicurare una buona regolazione della variabile in uscita.

È da ricordare, come descritto in [5], che il PID è un regolatore ad azione Proporzionale, Integrata, Derivativa utilizzato anche in ambito navale. Si compone di tre parti: la parte proporzionale (K_P), che agisce proporzionalmente all'errore attuale, la parte integrale (K_I), che agisce in modo proporzionale all'accumulo di errore passato, e la parte derivativa (K_D), che agisce in modo proporzionale alla velocità di cambiamento dell'errore. Le tre parti vengono moltiplicate per i rispettivi errori e sommate per generare la variabile di controllo τ .

Per regolare correttamente la variabile di controllo, è essenziale modificare e impostare adeguatamente i valori delle costanti del PID. Per questo motivo, inizialmente è stata disattivata la *guidance*, consentendo di effettuare prove statiche di avanzamento, traslazione e rotazione, dove è stata impostata una velocità richiesta pari a zero. Successivamente, sono state condotte ulteriori prove con una velocità richiesta diversa da zero.

La messa a punto delle costanti è stata effettuata utilizzando il metodo del "*trial & error*", che prevede la sperimentazione di diverse impostazioni, l'analisi dei risultati e la correzione degli errori riscontrati.

5.1. Prova statica di puro avanzo

In questo elaborato per prova statica si vuole intendere una prova per cui viene richiesta una velocità di esecuzione nulla e si chiede al modello di restare fermo in posizione una volta raggiunto il target.

È stata impostata la posizione di partenza coincidente con l'origine degli assi ed è stato chiesto al modello di raggiungere il punto di coordinate $[3,0,0]$, dove i primi due valori rappresentano le coordinate x e y espresse in metri mentre il terzo valore rappresenta l'angolo di prua espresso in radianti.

Inizialmente sono stati presi dei coefficienti di primo tentativo come descritto in Equazione 11:

Equazione 11 - Definizione dei coefficienti PID

$$[K_P] = \begin{bmatrix} K_{P_x} & 0 & 0 \\ 0 & K_{P_y} & 0 \\ 0 & 0 & K_{P_\psi} \end{bmatrix} = \begin{bmatrix} 3,5531429 & 0 & 0 \\ 0 & 3,785057 & 0 \\ 0 & 0 & 106,1085505 \end{bmatrix}$$

$$[K_I] = \begin{bmatrix} K_{I_x} & 0 & 0 \\ 0 & K_{I_y} & 0 \\ 0 & 0 & K_{I_\psi} \end{bmatrix} = \begin{bmatrix} 0,01 & 0 & 0 \\ 0 & 0,01 & 0 \\ 0 & 0 & 0,1 \end{bmatrix}$$

$$[K_D] = \begin{bmatrix} K_{D_x} & 0 & 0 \\ 0 & K_{D_y} & 0 \\ 0 & 0 & K_{D_\psi} \end{bmatrix} = \begin{bmatrix} 0,1026844022 & 0 & 0 \\ 0 & 0,1398035832 & 0 \\ 0 & 0 & 5,3478503873 \end{bmatrix}$$

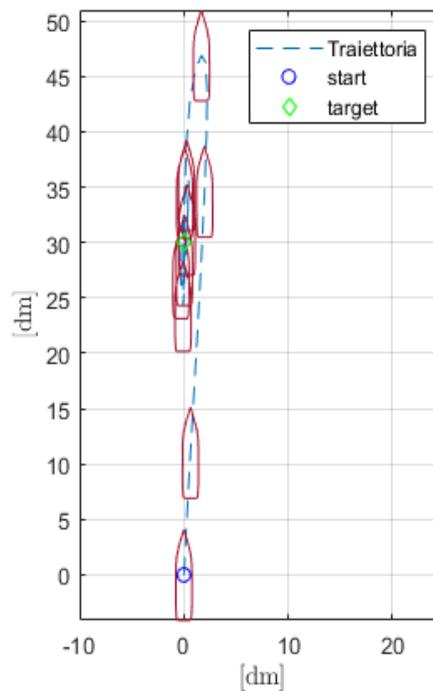


Figura 21 - Plotter prima prova puro avanzo

Questi valori sono stati poi moltiplicati per ulteriori coefficienti per permetterne la regolazione.

Di seguito sono riportati i risultati della prima simulazione ottenuti con i coefficienti di partenza.

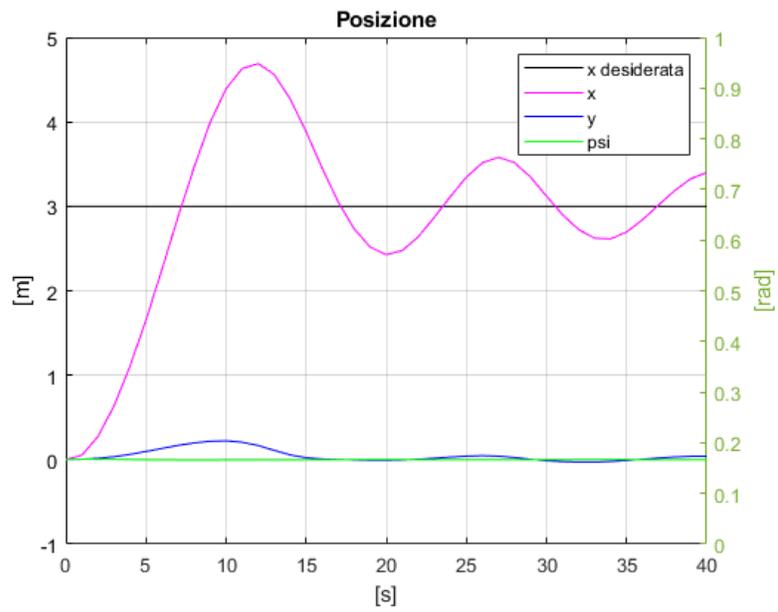


Figura 22 - Grafico posizione prima prova puro avanzo

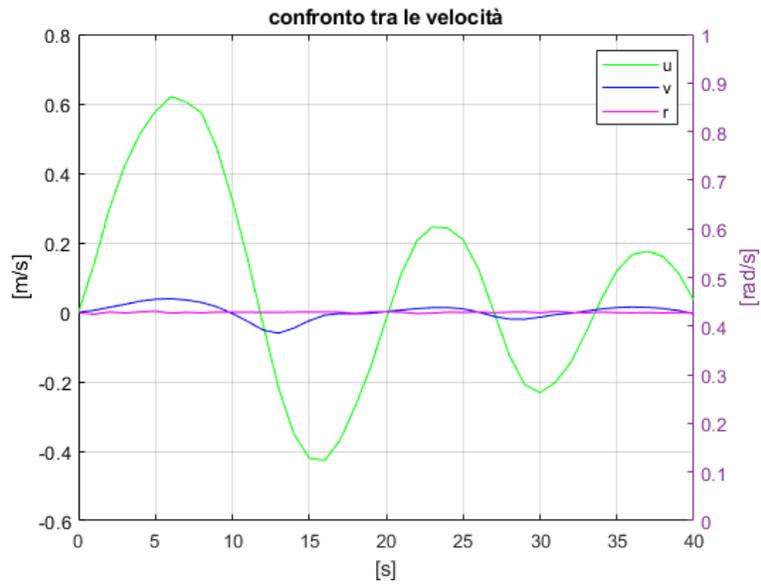


Figura 23 - Grafico velocità prima prova puro avanzo

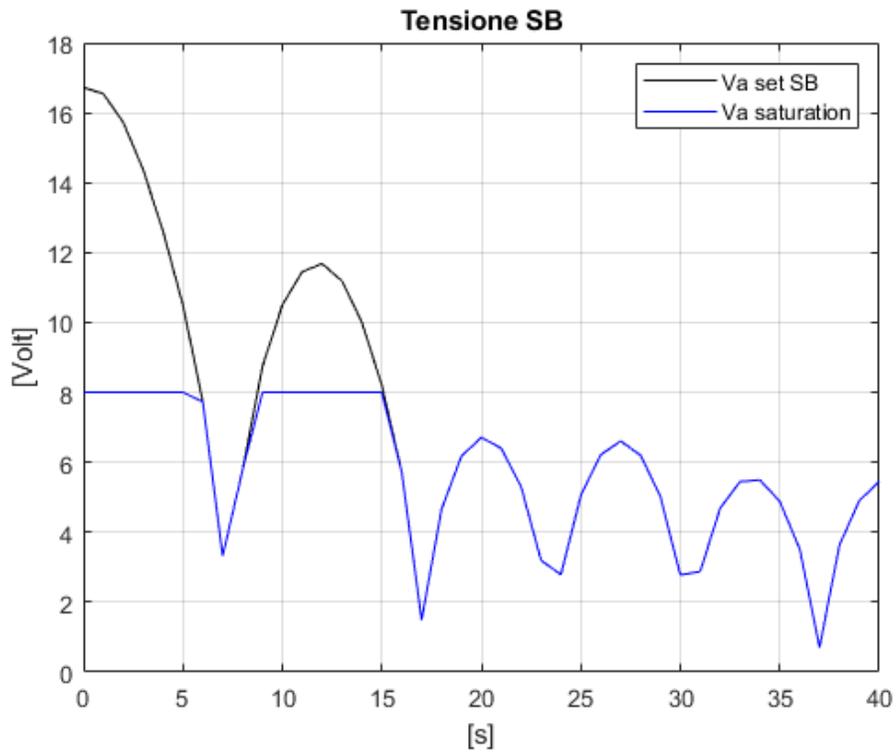


Figura 24 - Grafico tensione propulsore di dritta prima prova puro avanzo

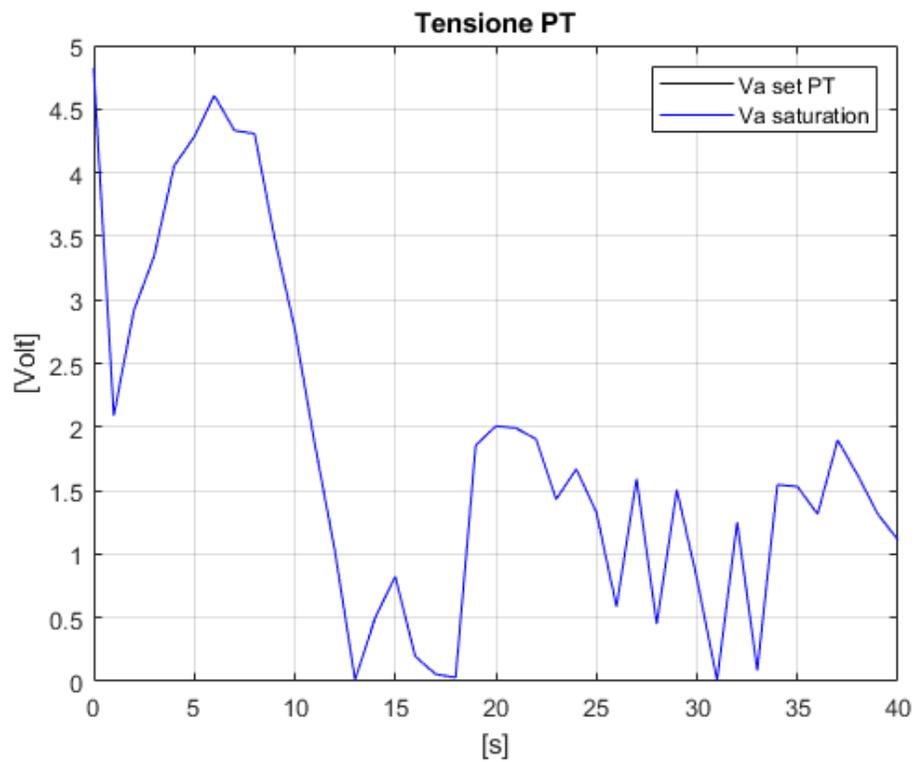


Figura 25 - Grafico tensione propulsore di sinistra prima prova puro avanzo

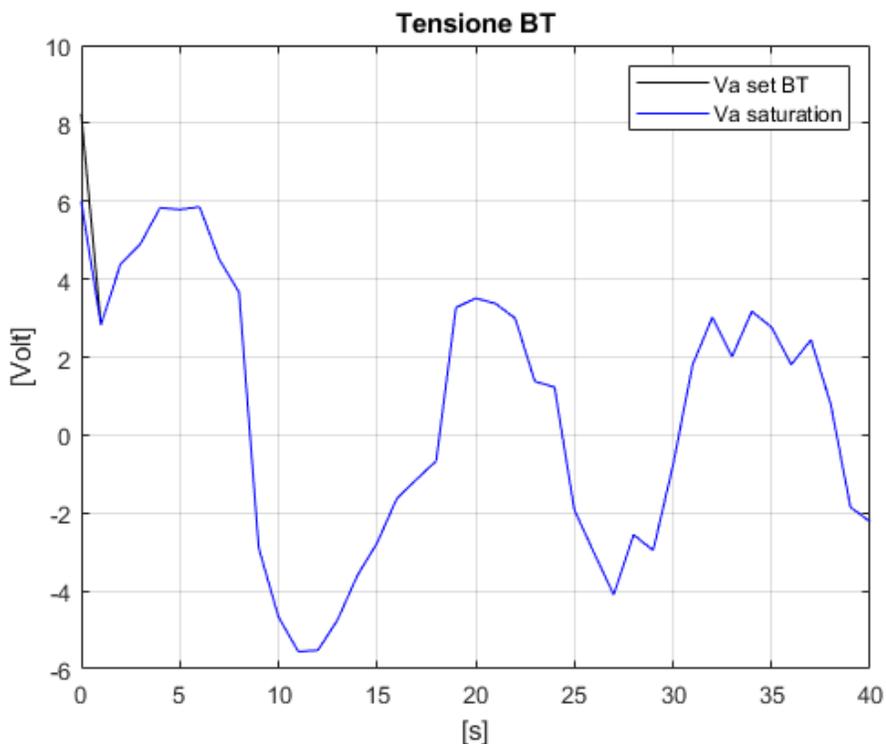


Figura 26 - Grafico tensione bow thuster prima prova puro avanzo

Come si può vedere in Figura 21 e Figura 23 il modello raggiunge la posizione desiderata ma a causa delle inerzie supera la posizione desiderata trovandosi costretta a tornare indietro ed invertire il verso della velocità.

Si osserva che il modello oscilla intorno al valore target a causa dei continui *overshoot*.

È da notare inoltre che il valore di tensione richiesto dal sistema (Figura 24), per il propulsore di dritta, è per la maggior parte del tempo superiore al valore massimo ammesso dai macchinari, portandoli quindi a saturazione. Risulta necessario ridurre il valore di K_p responsabile dell'azione proporzionale e diminuire di conseguenza il valore di tensione evitando così anche l'eccessivo *overshoot*. In seguito a questa prima prova si è deciso anche di aumentare i valori di K_d in modo da poter intervenire tempestivamente riducendo il valore di *overshoot*.

I nuovi coefficienti sono descritti di seguito in Tabella 2

Tabella 2 - Coefficienti seconda prova puro avanzo

	K_P	K_I	K_D
x	$0,5 \cdot K_{P_x}$	$1 \cdot K_{I_x}$	$70 \cdot K_{D_x}$
y	$0,5 \cdot K_{P_y}$	$1 \cdot K_{I_y}$	$70 \cdot K_{D_y}$
z	$1 \cdot K_{P_z}$	$1 \cdot K_{I_z}$	$1 \cdot K_{D_z}$

È stata quindi effettuata una nuova simulazione e di seguito sono riportati i risultati ottenuti:

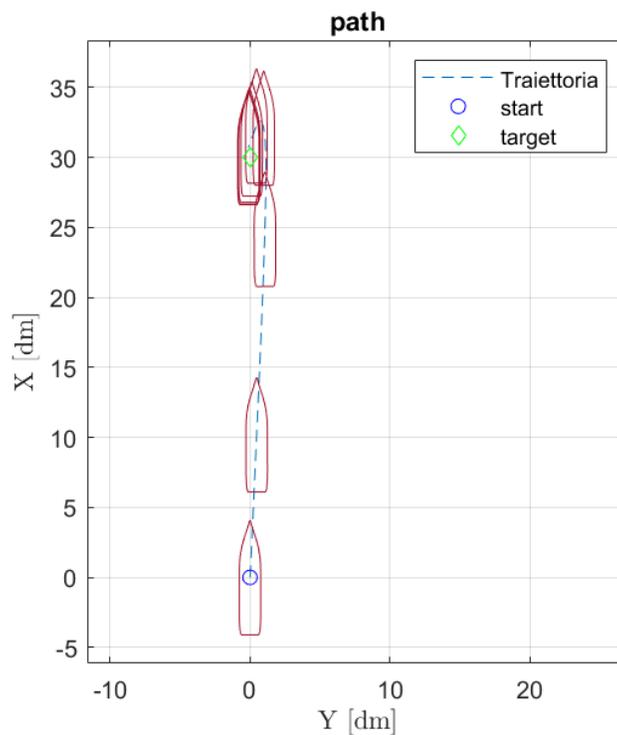


Figura 28 - Plotter seconda prova puro avanzo

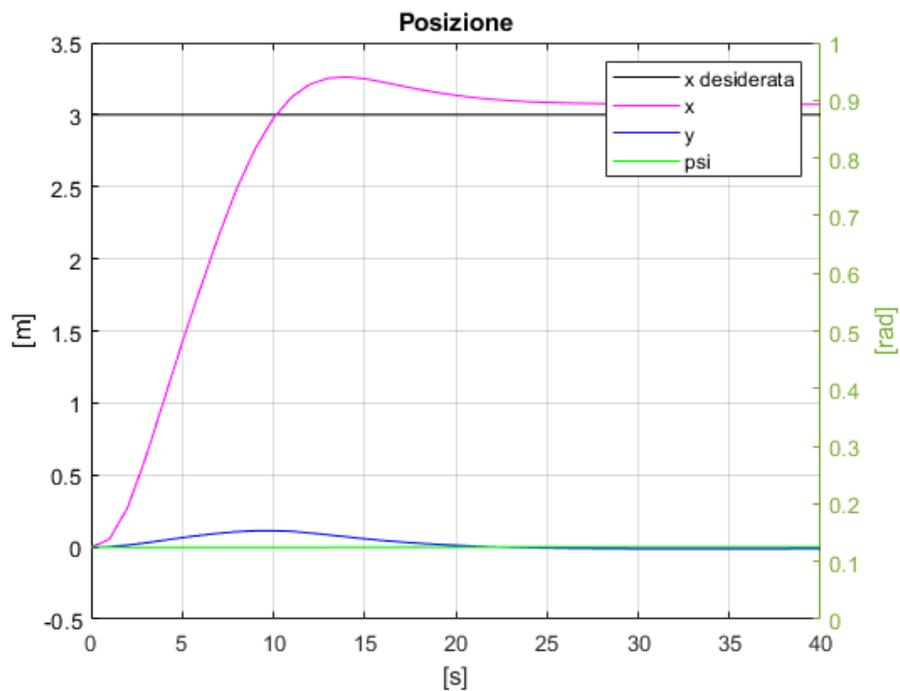


Figura 27 - Grafico posizione seconda prova puro avanzo

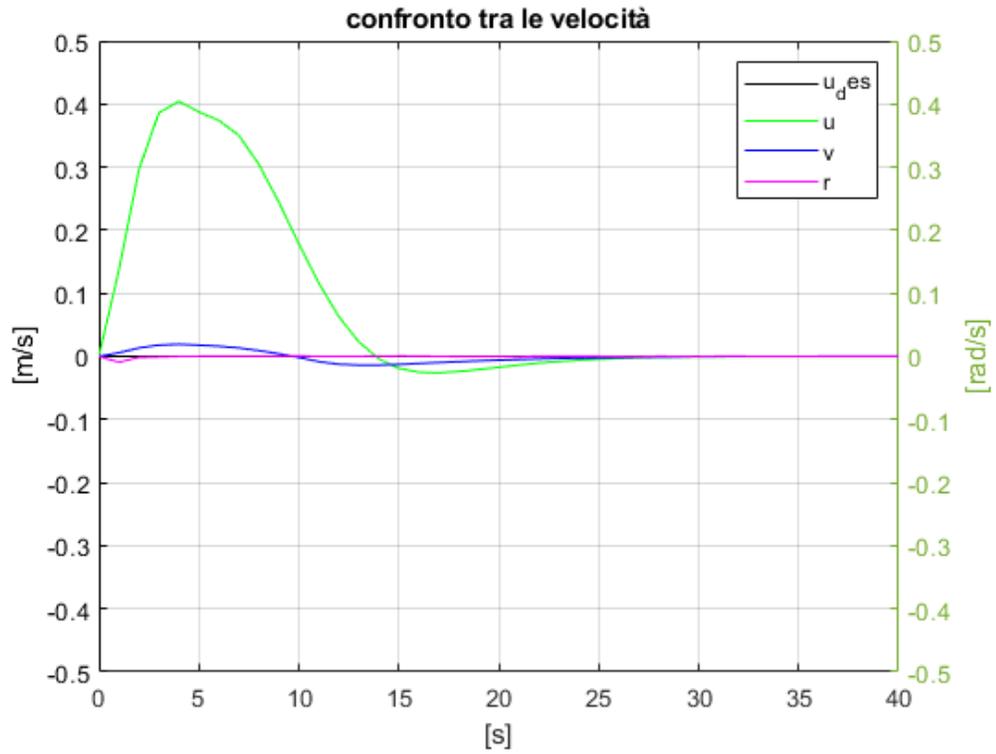


Figura 30 - Grafico velocità seconda prova puro avanzo

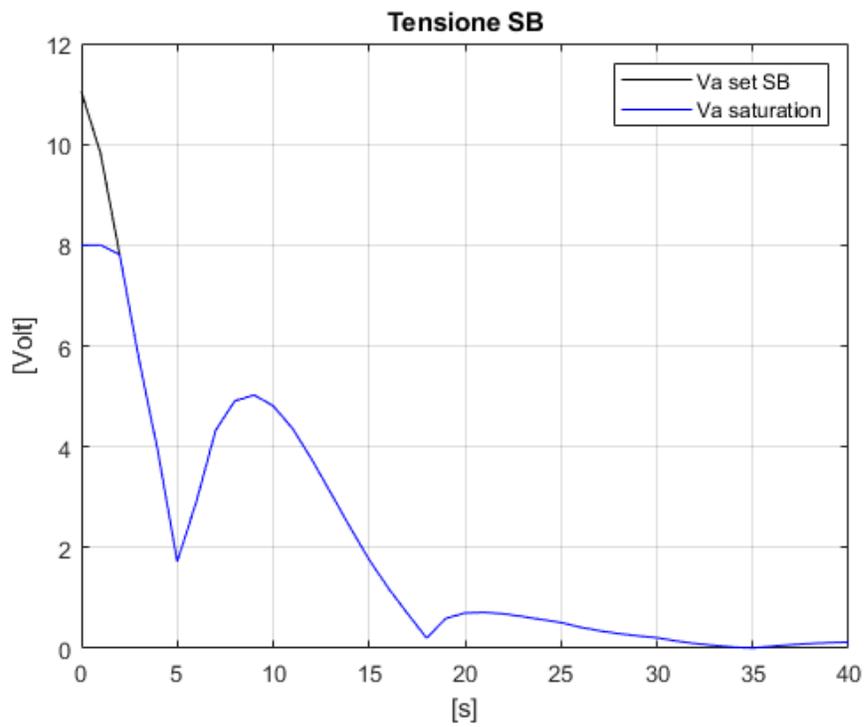


Figura 29 - Grafico tensione propulsore di dritta seconda prova puro avanzo

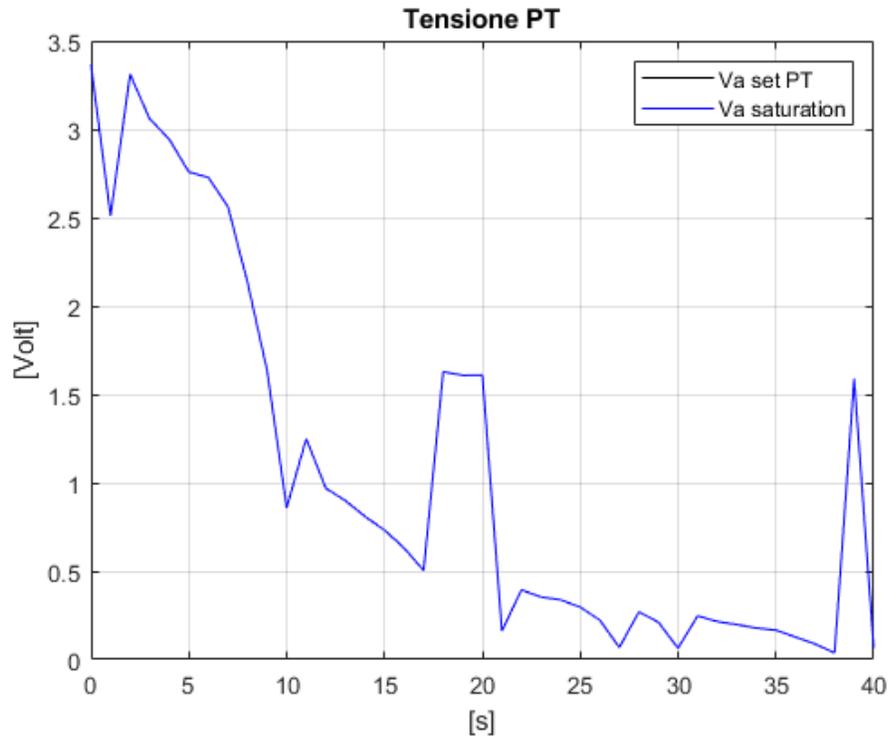


Figura 32 - Grafico tensione propulsore di sinistra seconda prova puro avanzo

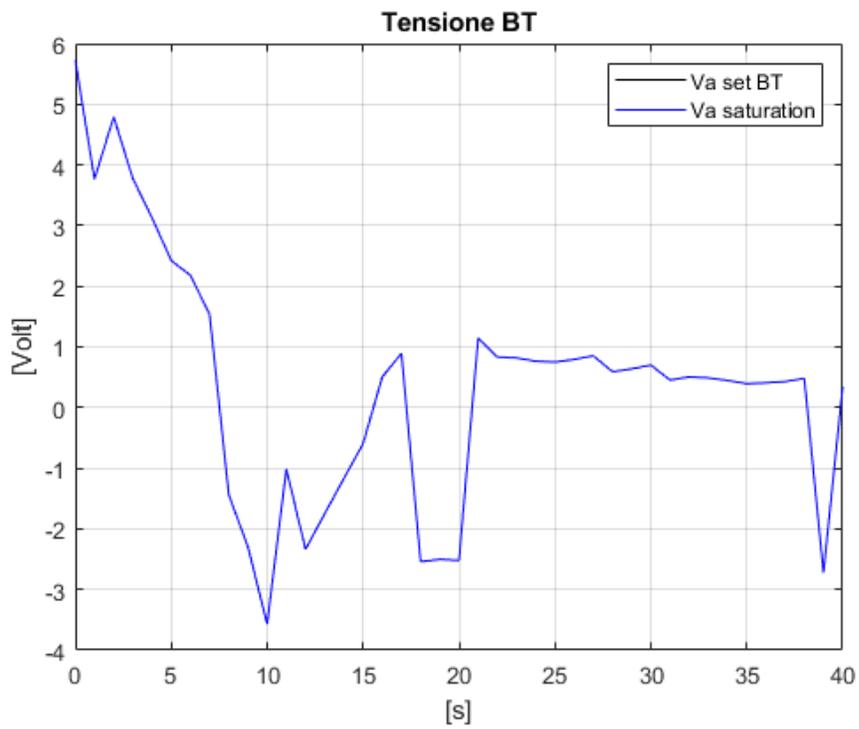


Figura 31 - Grafico tensione bow thruster seconda prova puro avanzo

Si può notare dalla Figura 27 che il modello raggiunge la posizione desiderata con un overshoot inferiore a quello iniziale. Inoltre dalla Figura 29 si può vedere che il valore di tensione richiesto è inferiore a quello massimo per la maggior parte del tempo. Infine, si può notare che non c'è oscillazione intorno al target dovuto ad un aumento del coefficiente K_d .

È stata poi fatta una prova, usando i medesimi coefficienti, per far raggiungere l'imbarcazione il punto di coordinate $[-3,0,0]$.

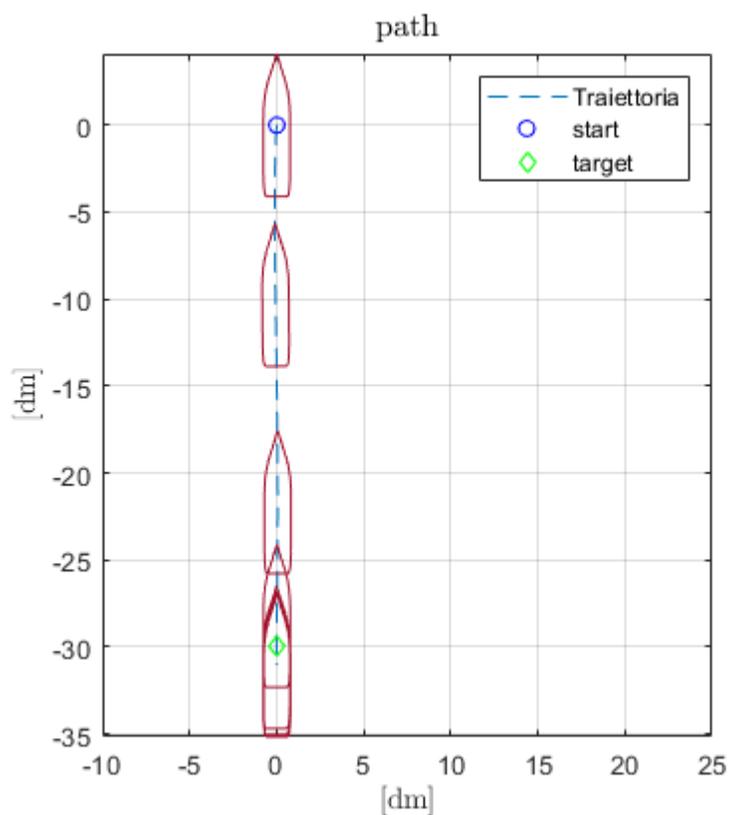


Figura 33 - Plotter terza prova puro avanzo

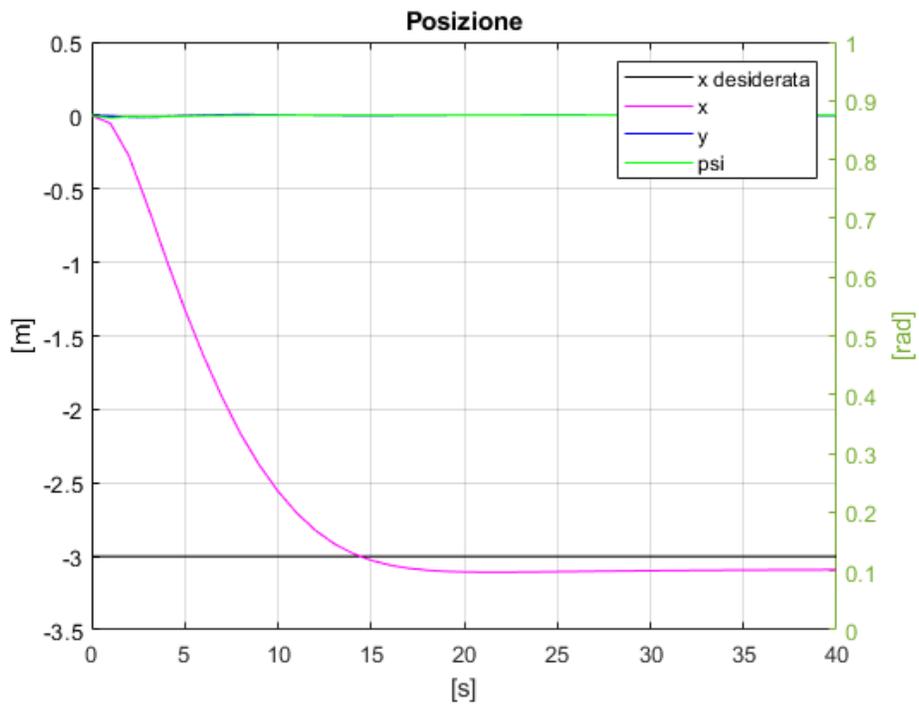


Figura 34 - Grafico posizione terza prova puro avanzo

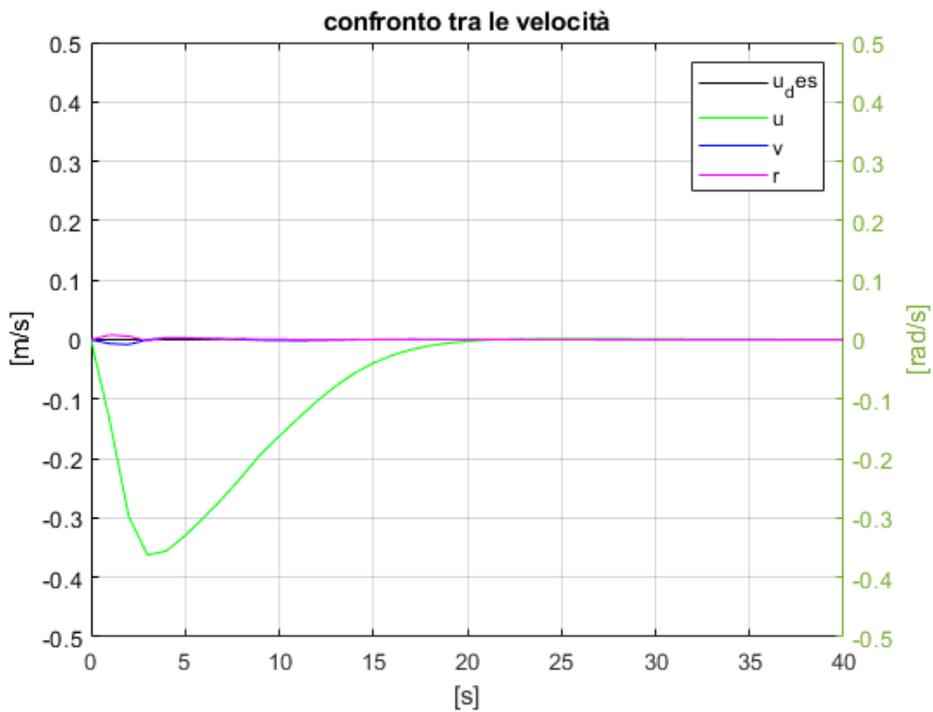


Figura 35 - Grafico velocità terza prova puro avanzo

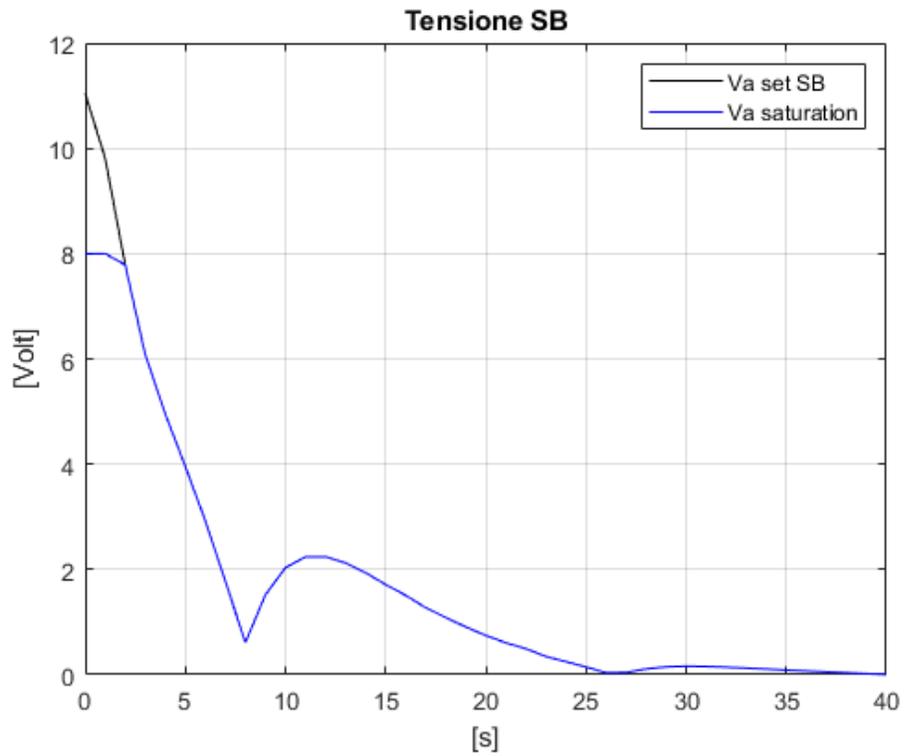


Figura 37 - Grafico tensione propulsore di dritta terza prova puro avanzo

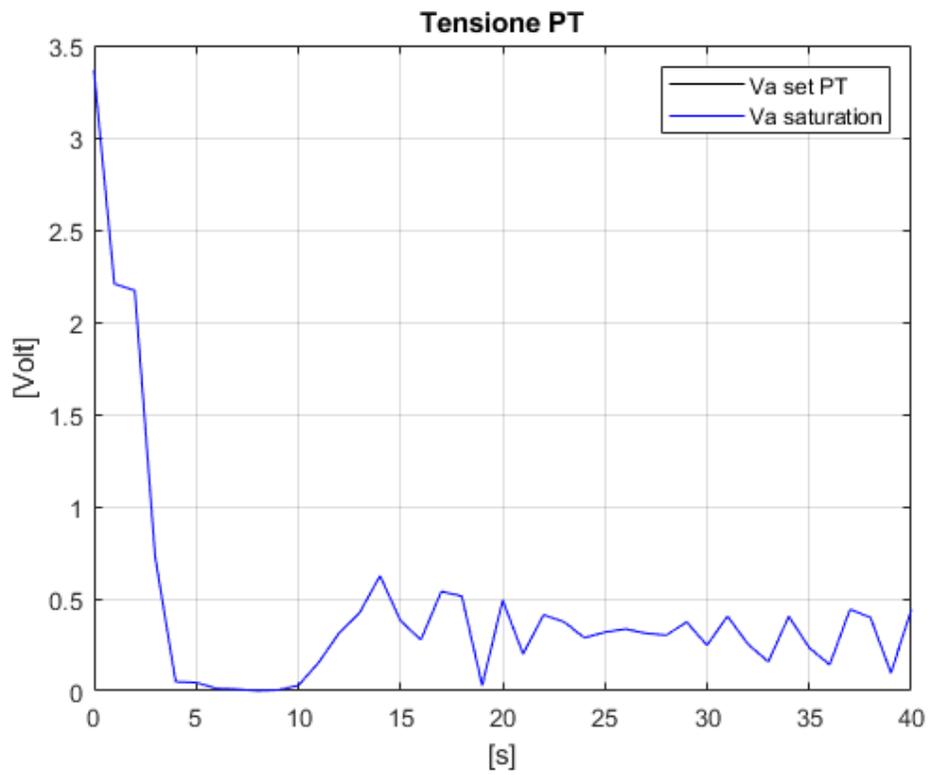


Figura 36 - Grafico tensione propulsore di sinistra terza prova puro avanzo

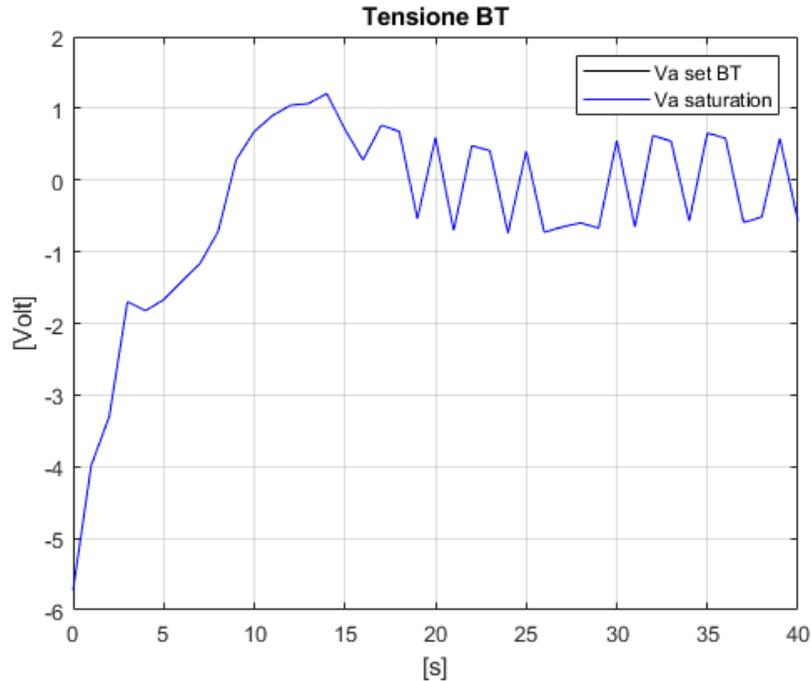


Figura 38 - Grafico tensione bow thruster terza prova puro avanzo

Come si può notare in tutte le prove c'è un valore di tensione richiesto al *bow thruster*, questo è dovuto all'allocazione scelta in quanto, essendo il propulsore di dritta l'unico a soddisfare le spinte lungo l'asse longitudinale, genera un momento dovuto al fatto che è decentrato, per compensare quindi, vengono azionati *bow thruster* e propulsore di sinistra.

5.2. Prova statica di pura deriva

Successivamente alla prova di puro avanzo, è stata effettuata la prova di pura deriva, in cui si chiede al modello di raggiungere il punto di coordinate $[0,3,0]$ partendo dall'origine degli assi. Anche in questo caso, i primi due valori rappresentano le coordinate x e y espresse in metri, mentre il terzo valore rappresenta l'angolo di prua espresso in radianti. Come primo tentativo, sono stati utilizzati gli stessi coefficienti utilizzati per la prova di puro avanzo, al fine di comprendere come funziona il modello anche nel caso di pura deriva.

Di seguito sono riportati i risultati:

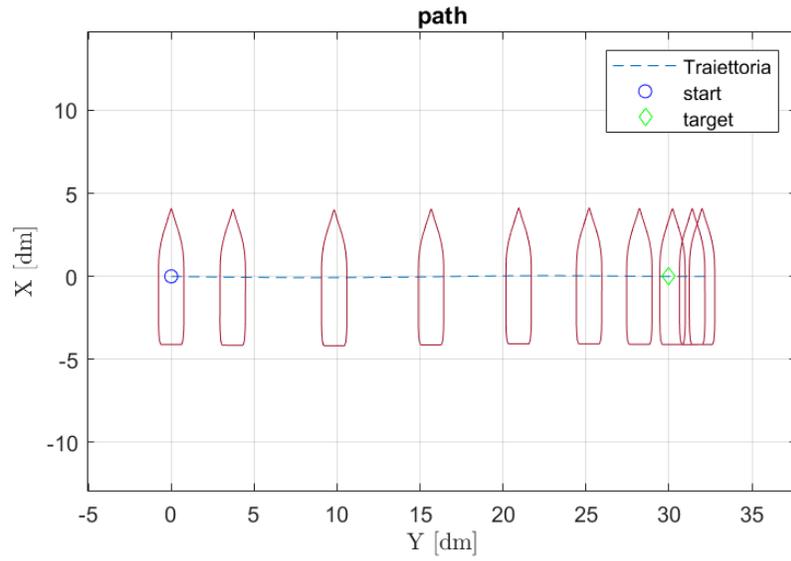


Figura 39 - Plotter prima prova pura deriva

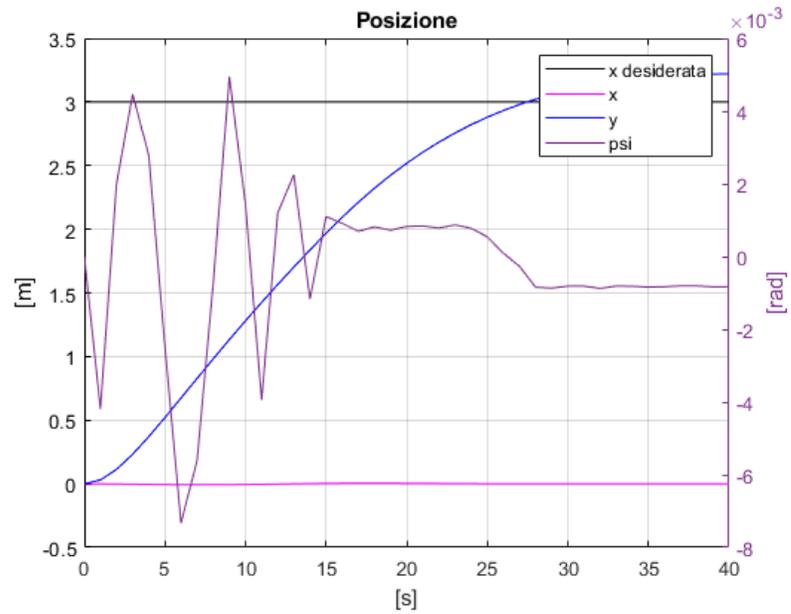


Figura 40 - Grafico posizione prima prova pura deriva

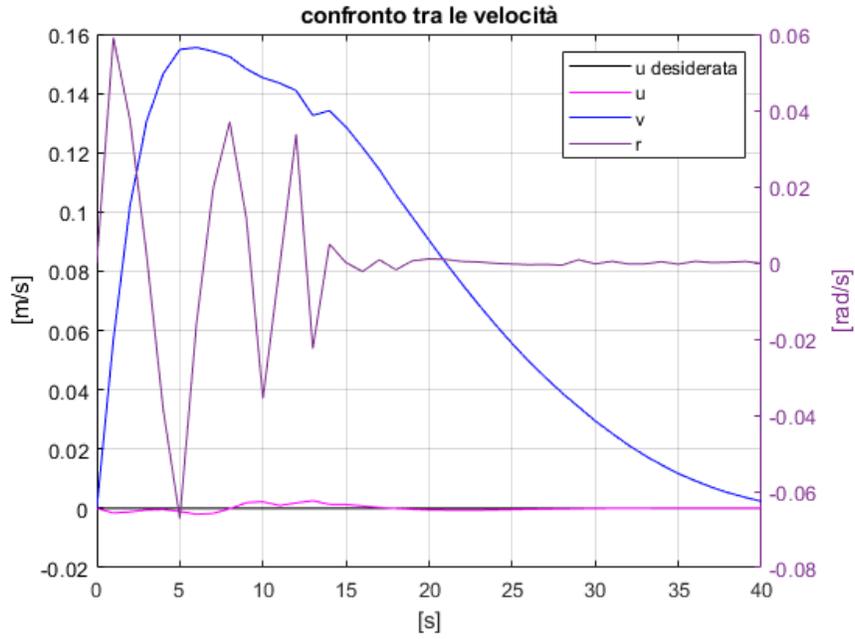


Figura 41 - Grafico velocità prima prova pura deriva

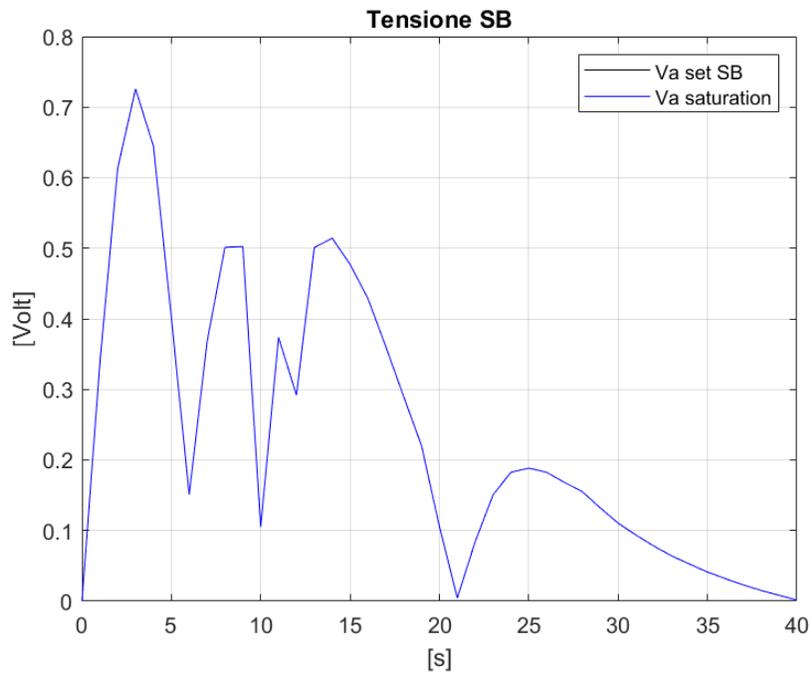


Figura 42 - Grafico tensione propulsore di dritta prima prova pura deriva

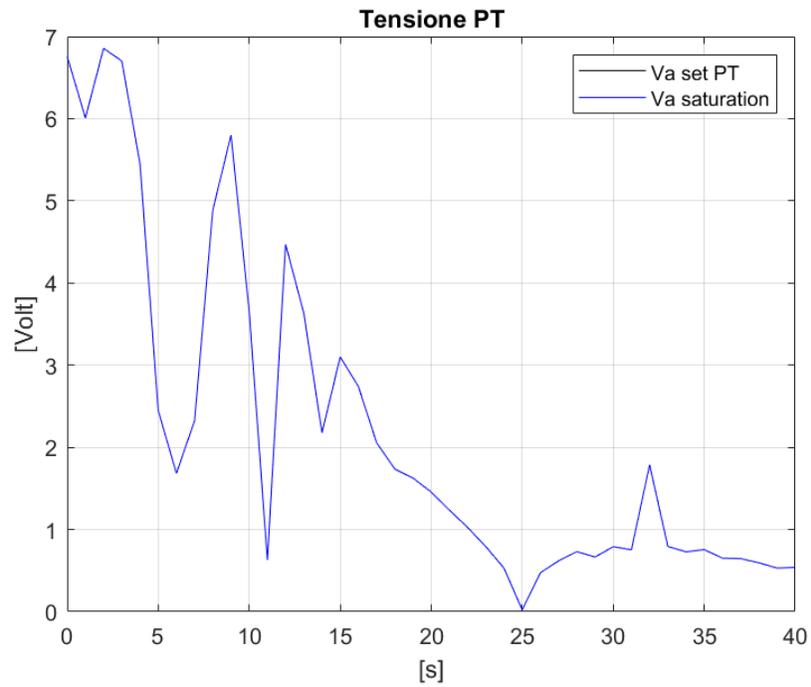


Figura 44 - Grafico tensione propulsore di sinistra prima prova pura deriva

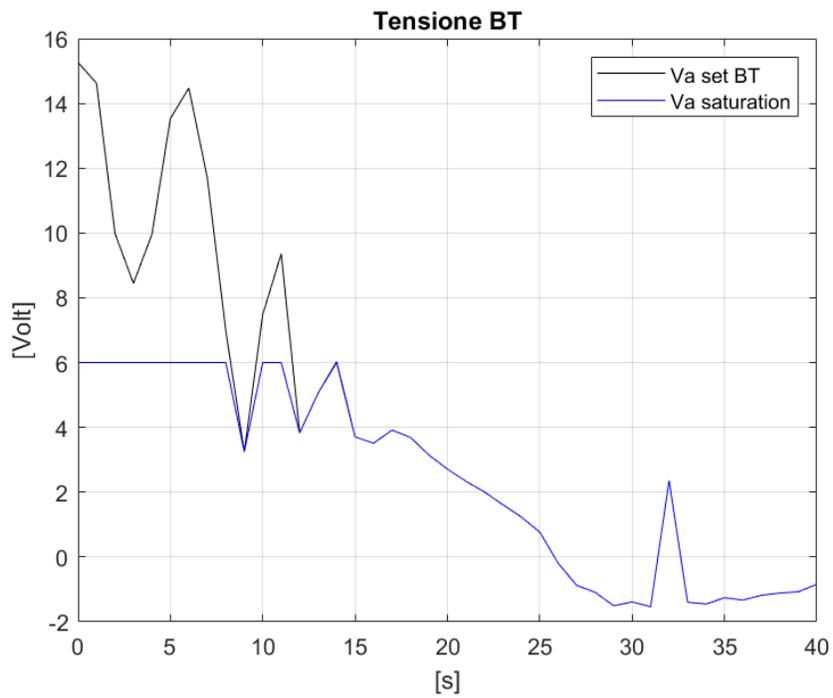


Figura 43 - Grafico tensione bow thruster prima prova pura deriva

Come si può notare dalla Figura 39 e Figura 40, c'è un errore statico che può essere corretto aumentando i coefficienti K_I . È interessante notare che la posizione desiderata viene raggiunta in circa 27 secondi, mentre nella prova di puro avanzo ci sono voluti circa 10 secondi. Questo è dovuto al fatto che la resistenza alla deriva è maggiore rispetto alla resistenza all'avanzo. È importante sottolineare che i coefficienti K_x e K_y sono espressi nel sistema di riferimento inerziale; pertanto, agiscono allo stesso modo sia per l'avanzo che per la deriva. Tuttavia, per cercare di accelerare la manovra, si è scelto di aumentare i valori di K_p e K_I rispetto agli assi x e y .

Di seguito in Tabella 3 vengono riportati i coefficienti aggiornati utilizzati per la manovra successiva

Tabella 3 - Coefficienti seconda prova pura deriva

	K_P	K_I	K_D
x	$0,6 \cdot K_{P_x}$	$1,5 \cdot K_{I_x}$	$90 \cdot K_{D_x}$
y	$0,6 \cdot K_{P_y}$	$1,5 \cdot K_{I_y}$	$90 \cdot K_{D_y}$
z	$1 \cdot K_{P_z}$	$1 \cdot K_{I_z}$	$1 \cdot K_{D_z}$

È stata quindi effettuata una seconda prova di pura deriva. Di seguito i risultati:

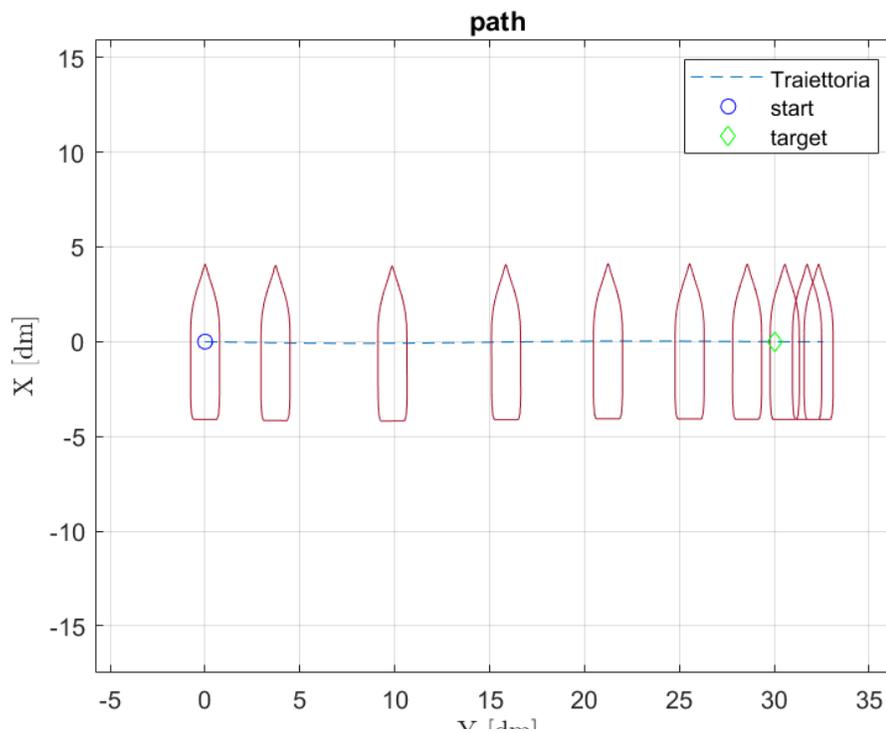


Figura 45 - Plotter seconda prova pura deriva

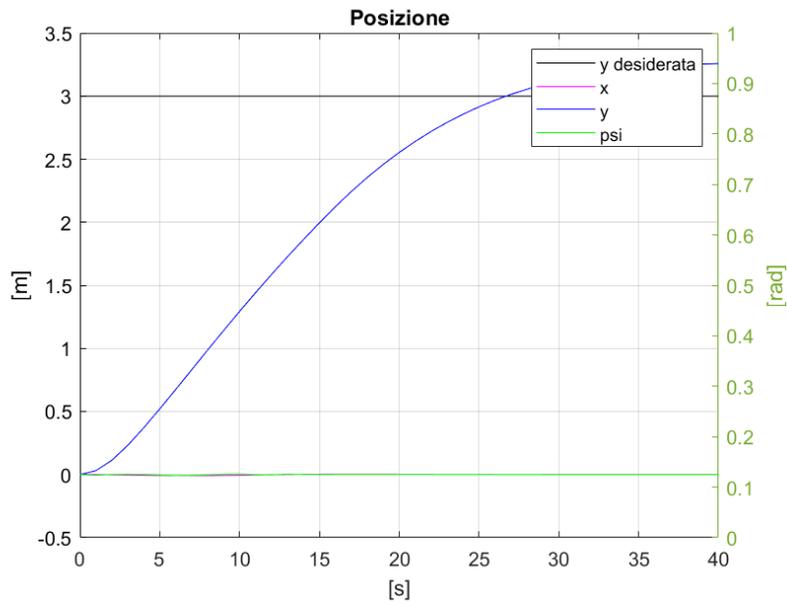


Figura 47 - Grafico posizione seconda prova pura deriva

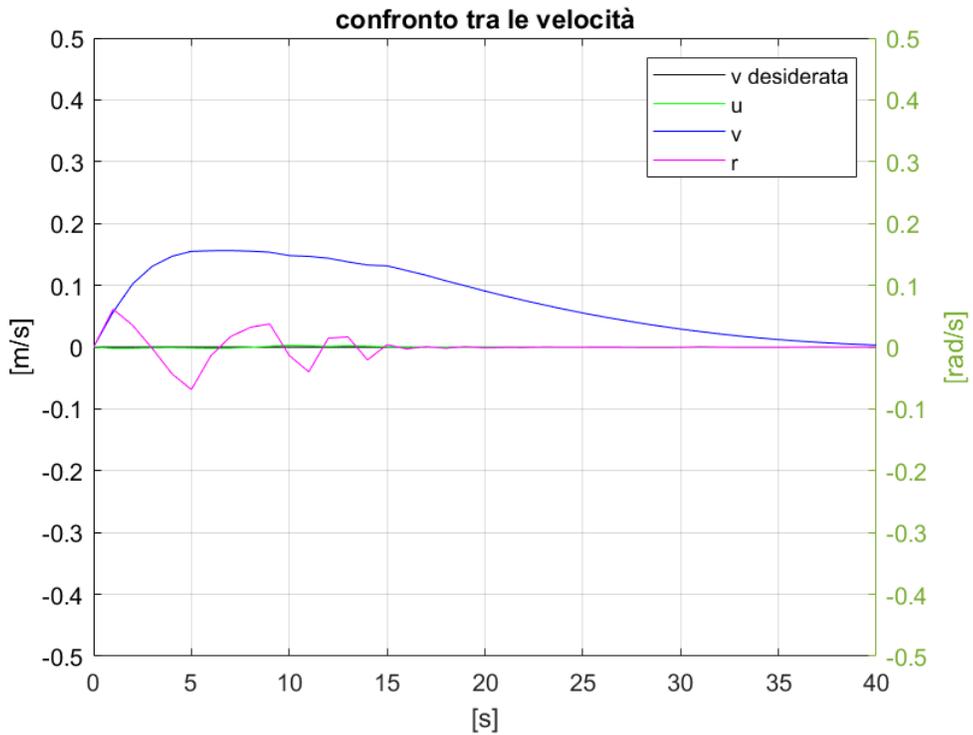


Figura 46 - Grafico velocità seconda prova pura deriva

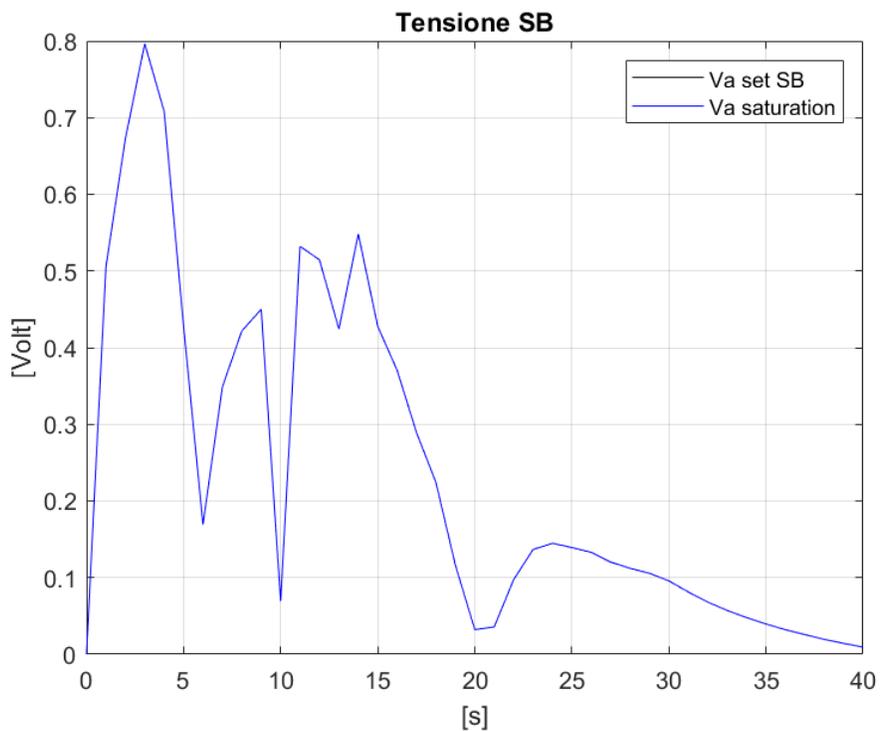


Figura 48 - Grafico tensione propulsore di dritta seconda prova pura deriva

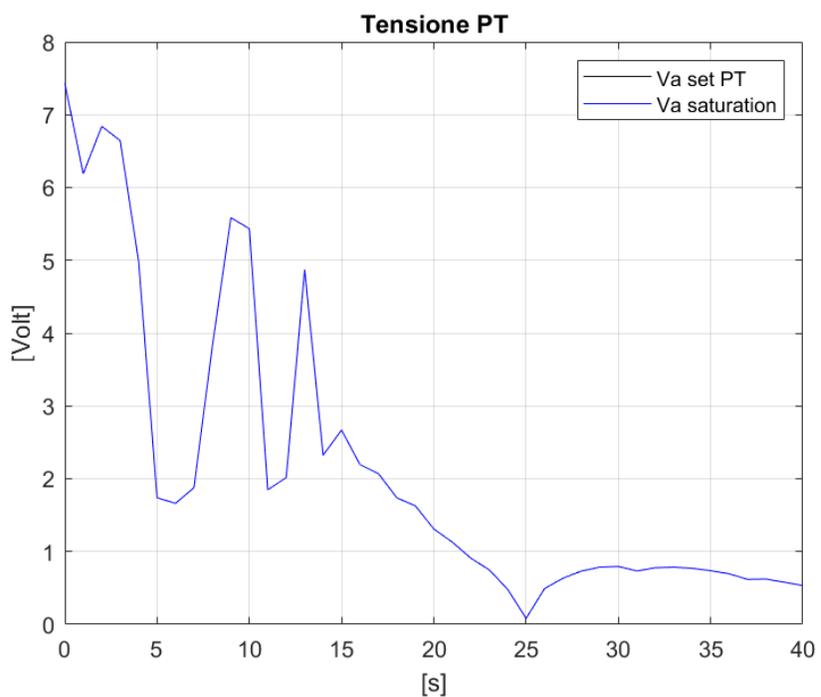


Figura 49 - Grafico tensione propulsore di sinistra seconda prova pura deriva

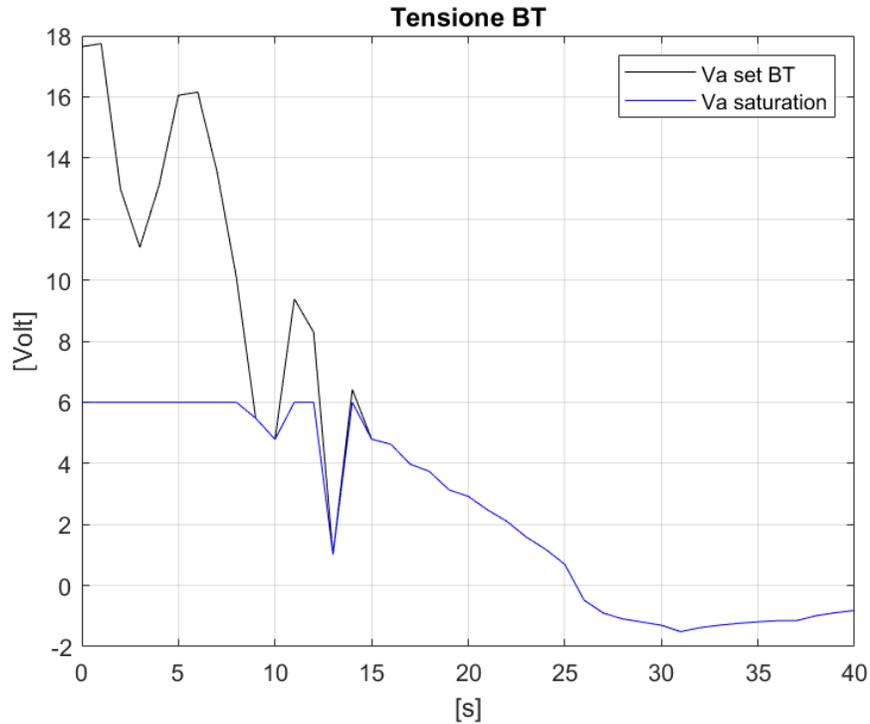


Figura 50 - Grafico tensione bow thruster

Dalla Figura 45 e Figura 46 si nota una lieve riduzione del tempo impiegato a raggiungere il target. Questo è dovuto ad un minimo aumento dei coefficienti k_p ; è probabile che anche l'aumento di k_I abbia influenzato il tempo di esecuzione negativamente.

È stata poi effettuata una ulteriore prova in cui è stato richiesto al modello di raggiungere il punto di coordinate $[0,-3,0]$ ovvero di effettuare una traslazione di 3 metri verso sinistra. Di seguito i risultati.

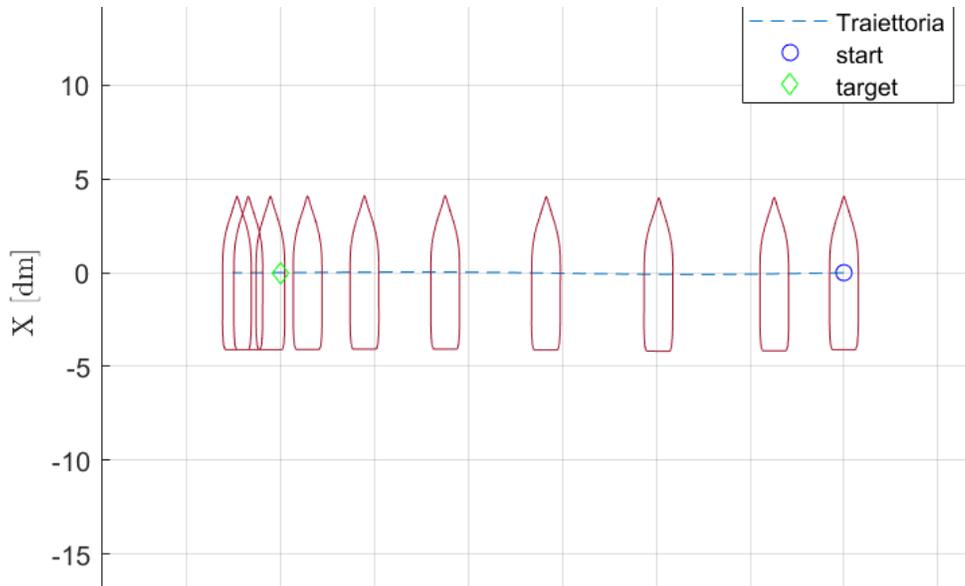


Figura 52 - Plotter terza prova pura deriva

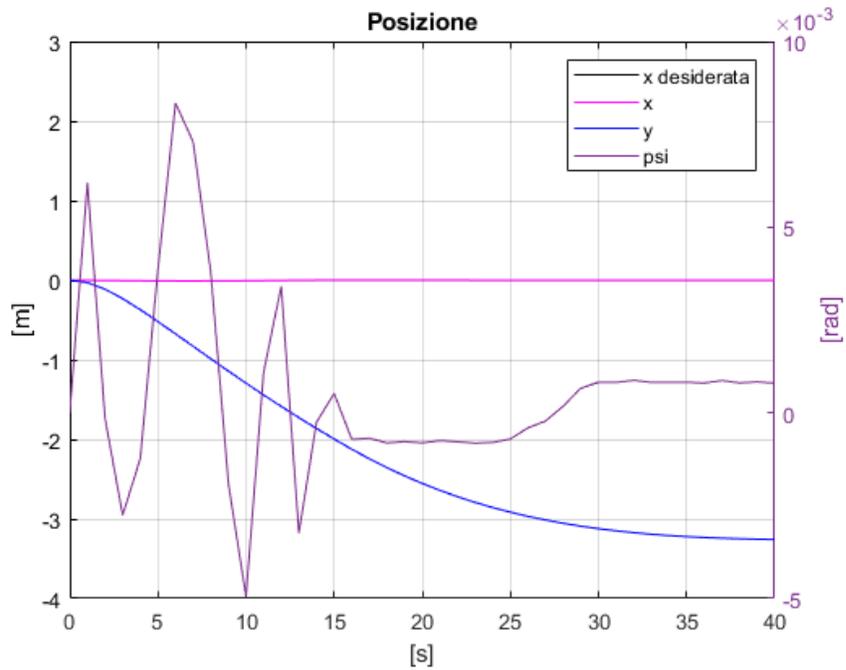


Figura 51 - Grafico posizione terza prova pura deriva

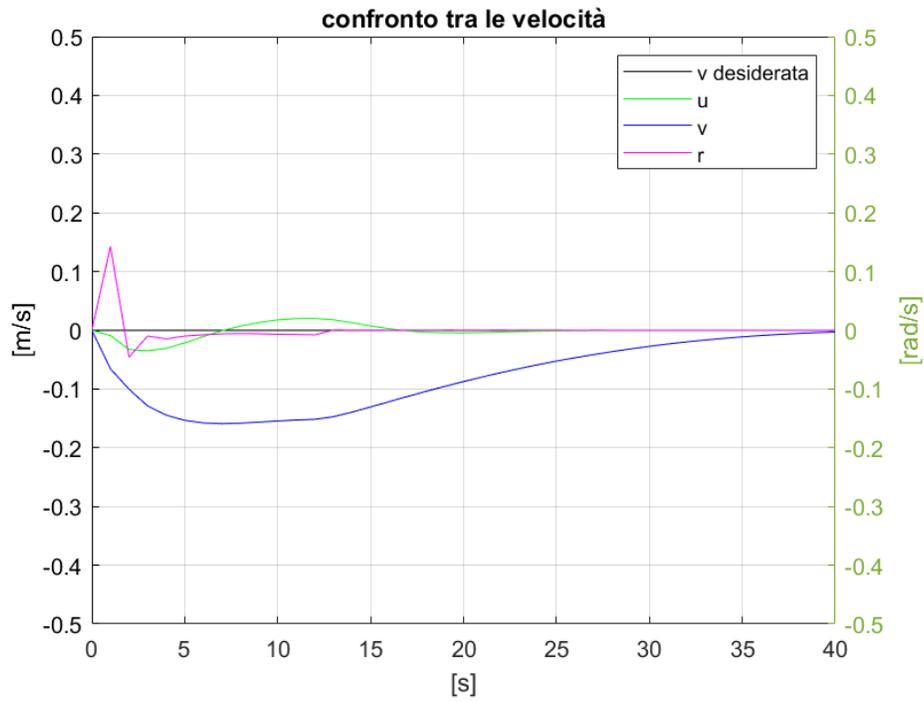


Figura 54 - Grafico velocità terza prova pura deriva

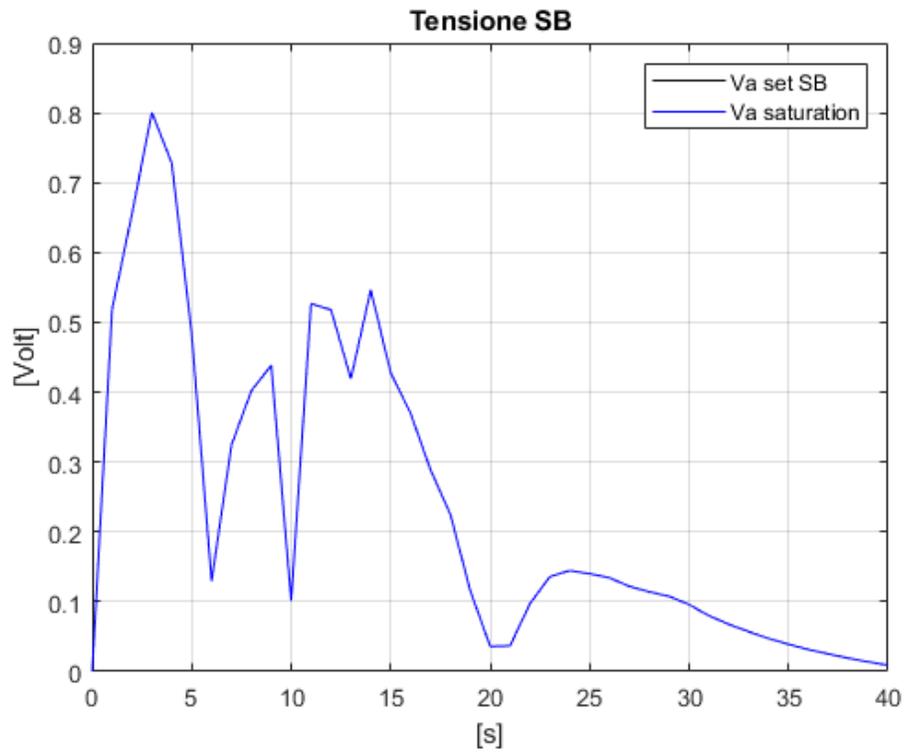


Figura 53 - Grafico tensione propulsore di dritta terza prova pura deriva

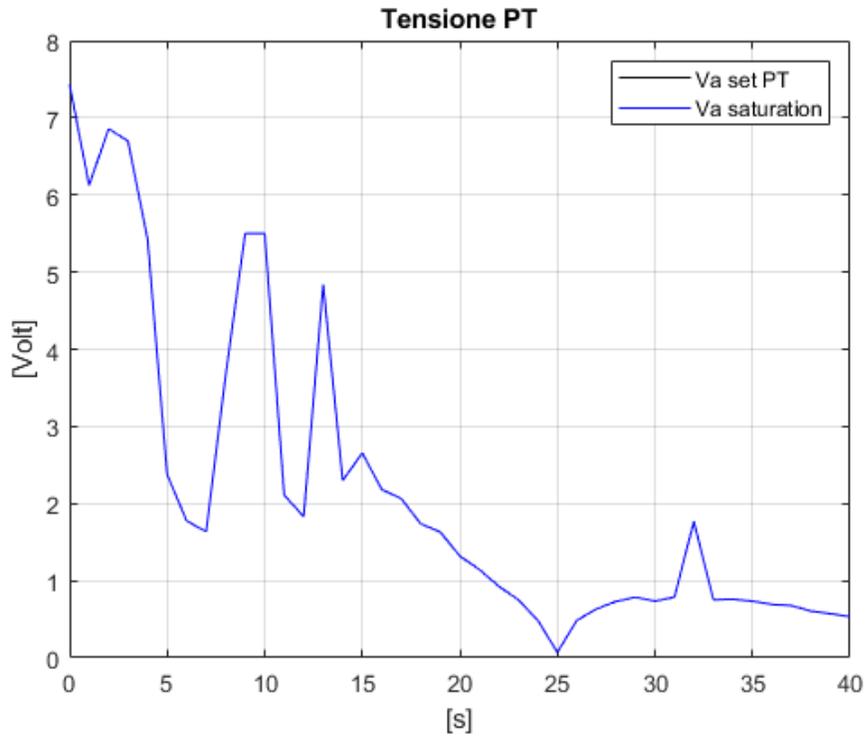


Figura 56 - Grafico tensione propulsore di sinistra terza prova pura deriva

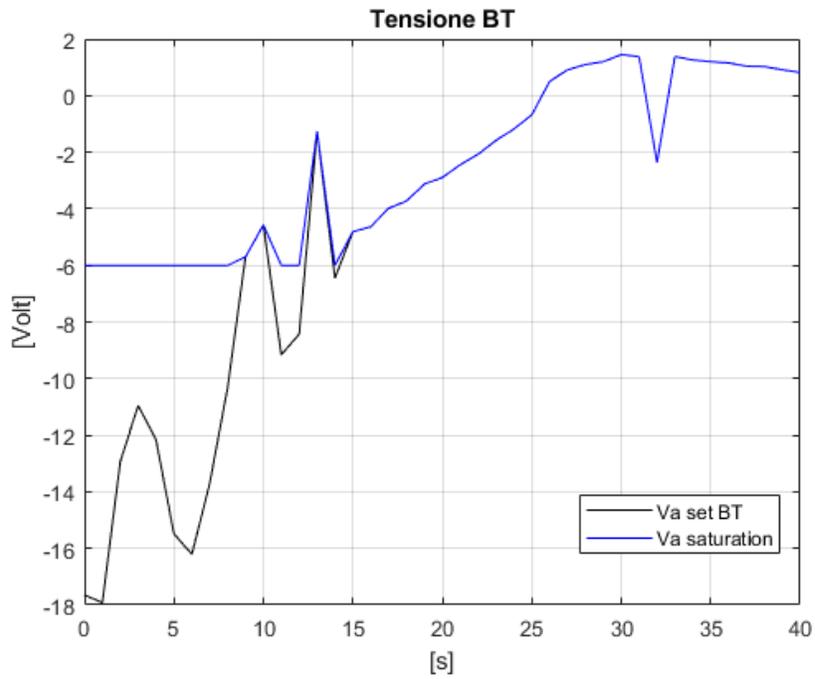


Figura 55 - Grafico tensione bow thruster terza prova pura deriva

5.3. Prova statica di pura imbardata

L'ultima prova effettuata a velocità richiesta nulla è stata quella di pura imbardata, ossia è stato chiesto al modello di ruotare su se stesso di 90° verso dritta raggiungendo così il punto di coordinate $[0,0,1.57]$.

Come primo tentativo sono stati utilizzati gli stessi coefficienti adoperati per l'ultima prova di pura deriva.

Di seguito sono riportati i risultati ottenuti:

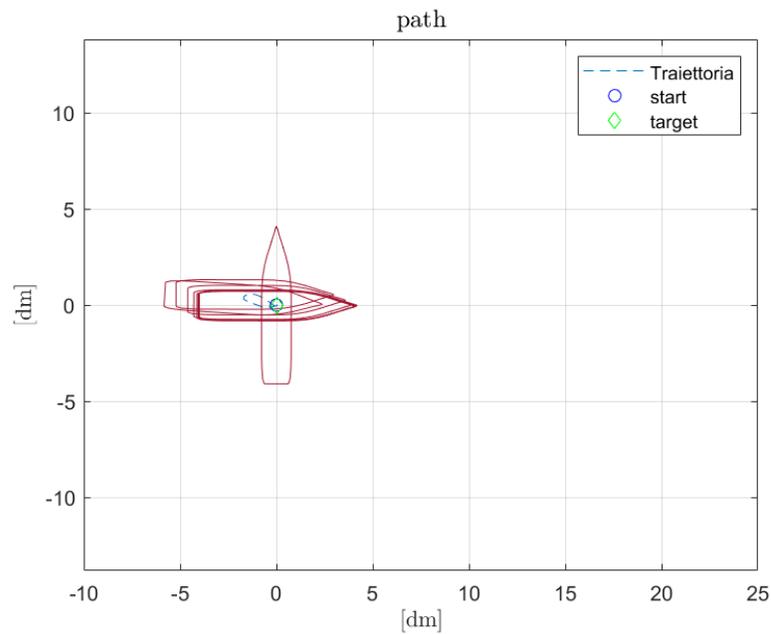


Figura 57 - Plotter prima prova pura imbardata

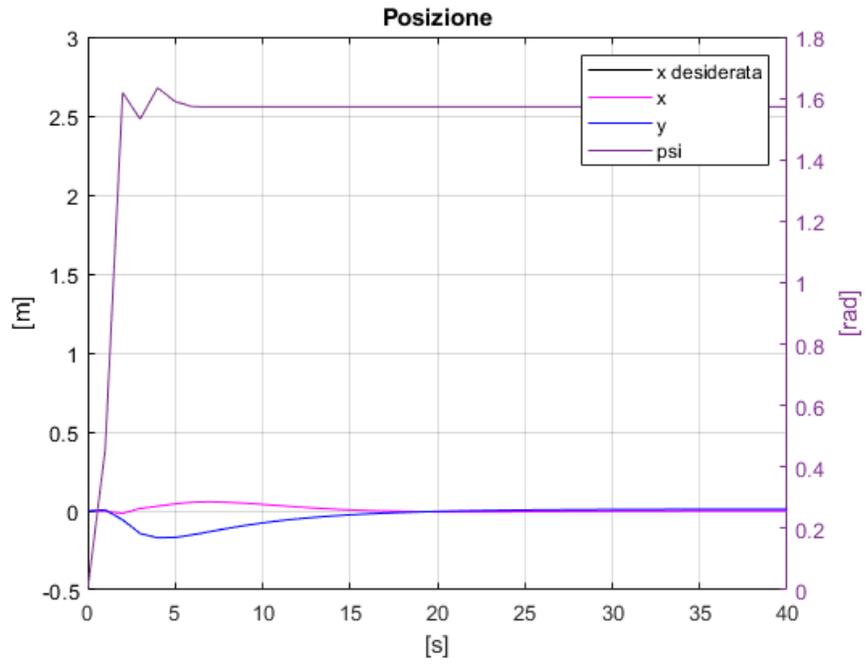


Figura 58 - Grafico posizione prima prova pura imbardata

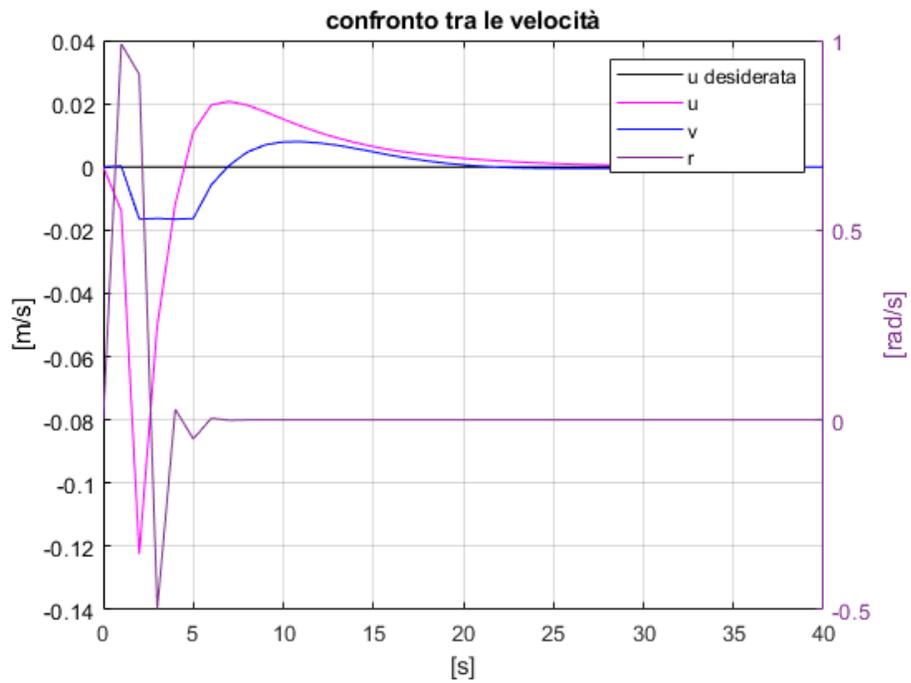


Figura 59 - Grafico velocità prima prova pura imbardata

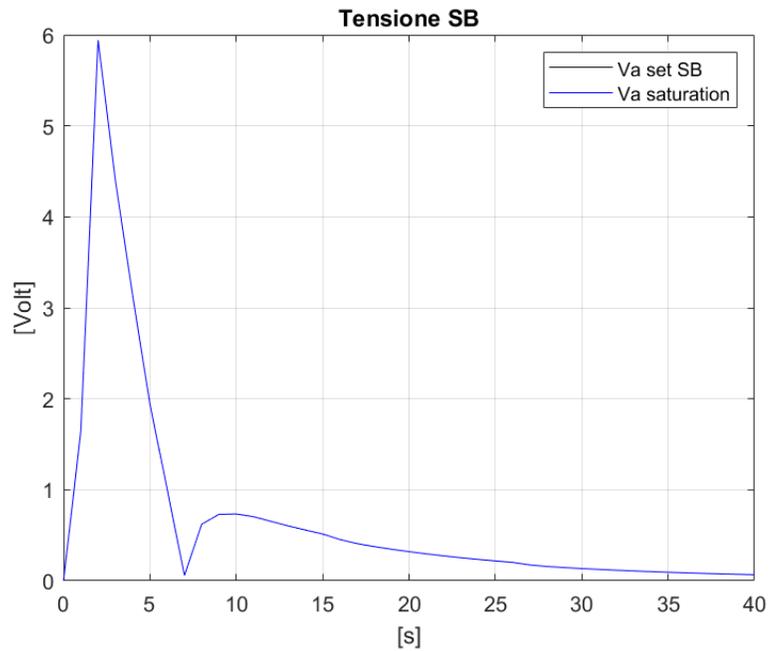


Figura 61 - Grafico tensione propulsore di dritta prima prova pura imbardata

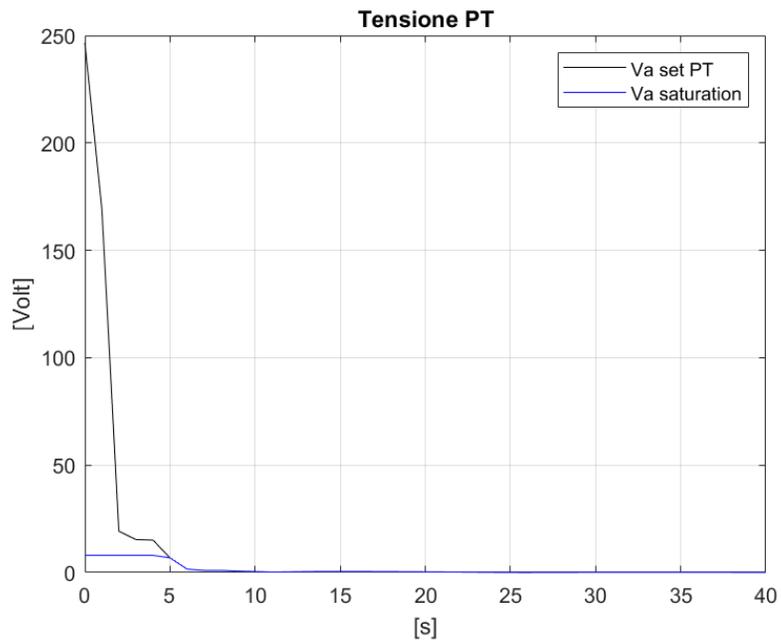


Figura 60 - Grafico tensione propulsore di sinistra prima prova pura imbardata

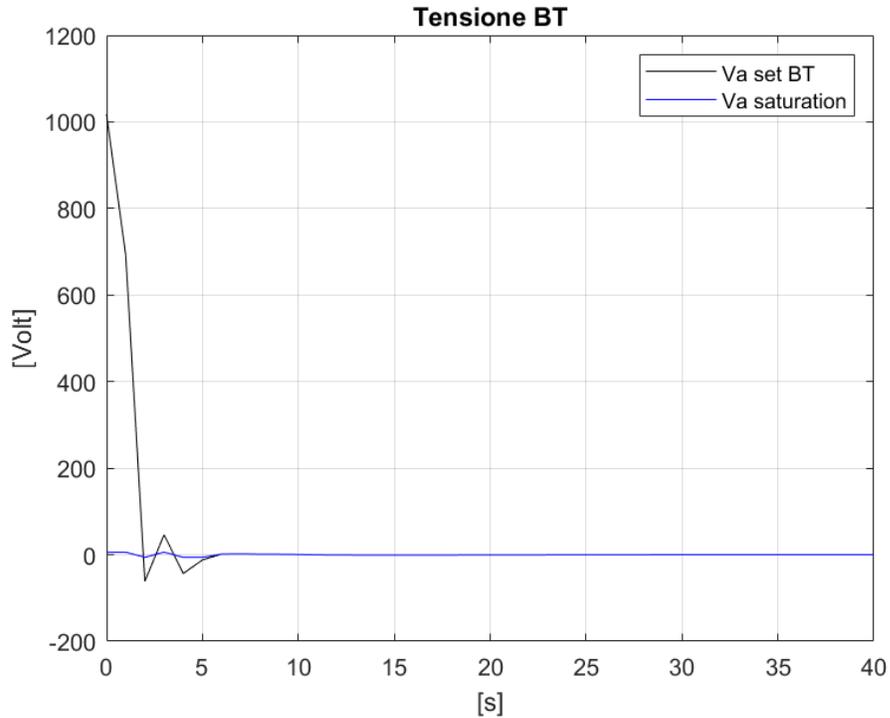


Figura 62 - Grafico tensione bow thruster prima prova pura imbardata

come si può notare dalla Figura 58 il modello raggiunge la posizione desiderata in pochi secondi. Per il funzionamento del sistema di *autoberthing* di questo elaborato è importante che la rotazione venga effettuata in tempi brevi in modo da permettere all'imbarcazione di effettuare lo spostamento tra i due *waypoint* con l'angolo di prua desiderato, se così non fosse l'imbarcazione ruoterebbe durante lo spostamento andando a occupare posizioni non previste dalla *guidance* rischiando così possibili collisioni. È da notare però che i valori di tensione del *bow thruster* e del propulsore di sinistra sono estremamente alto, andando così in saturazione per tutta la manovra. Risulta dunque necessario andare a ridurre il coefficiente K_{P_z} senza andare però aumentare troppo il tempo di esecuzione della manovra.

In Tabella 4 i coefficienti adoperati per la seconda prova di pura imbardata:

Tabella 4 - Coefficienti seconda prova pura imbardata

	K_P	K_I	K_D
x	$0,6 \cdot K_{P_x}$	$1,5 \cdot K_{I_x}$	$90 \cdot K_{D_x}$
y	$0,6 \cdot K_{P_y}$	$1,5 \cdot K_{I_y}$	$90 \cdot K_{D_y}$
z	$1 \cdot K_{P_z}$	$1 \cdot K_{I_z}$	$1 \cdot K_{D_z}$

Di seguito i risultati:

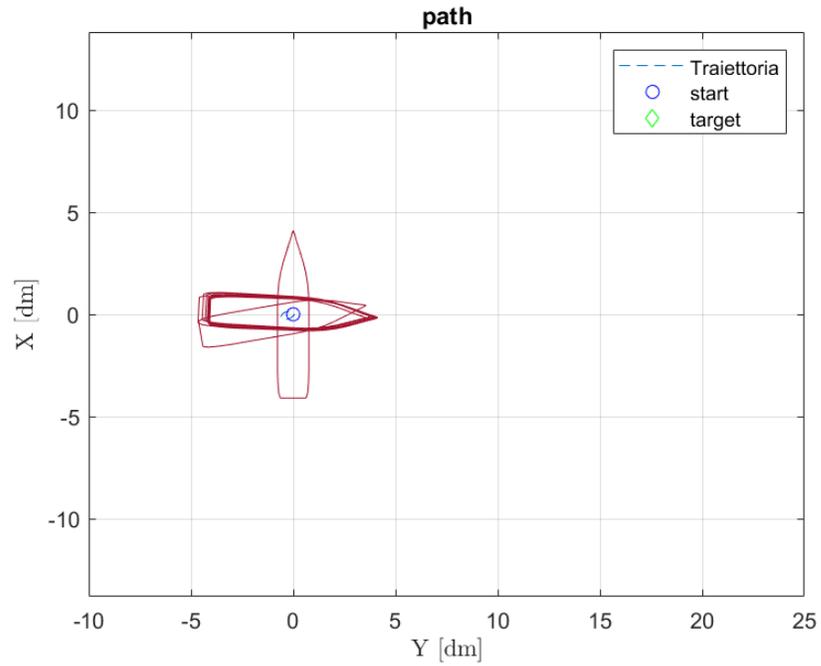


Figura 64 - Plotter seconda prova pura imbardata

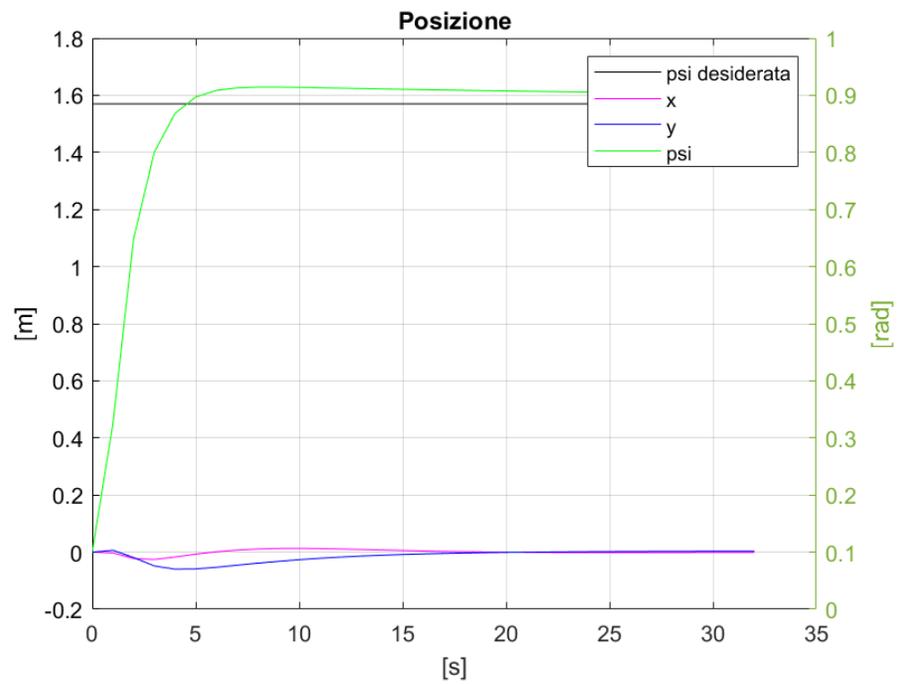


Figura 63 - Grafico posizione seconda prova pura imbardata

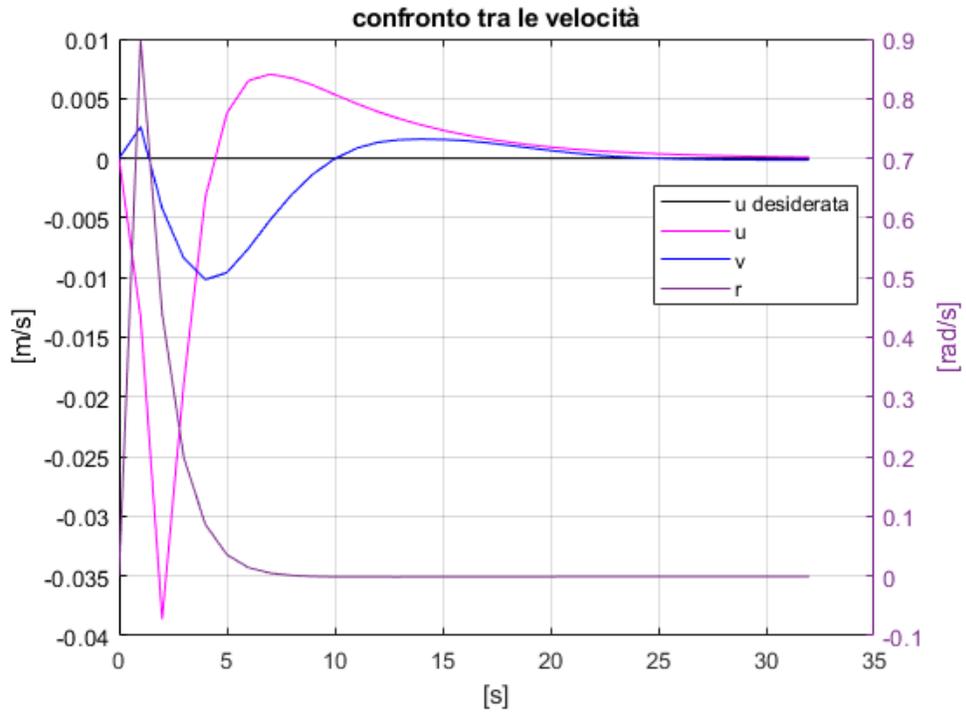


Figura 66 - Grafico velocità seconda prova pura imbardata

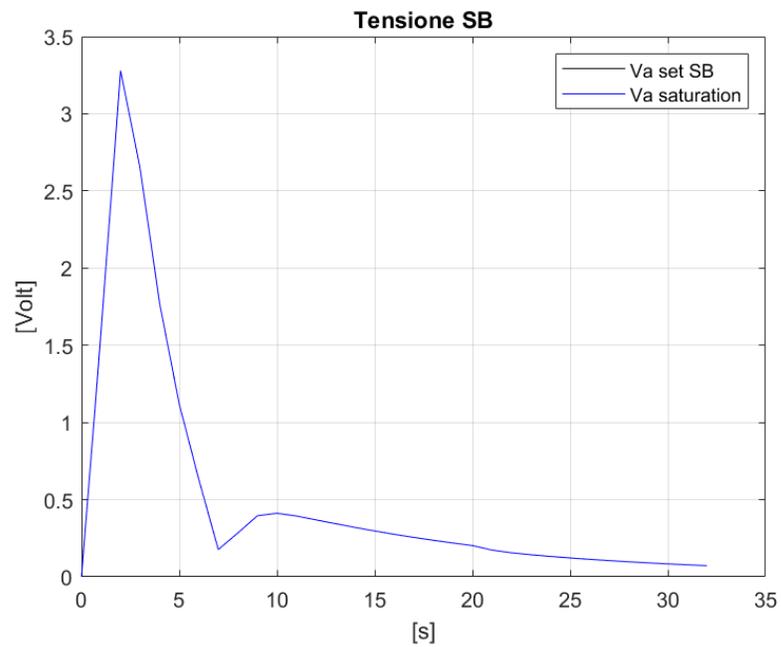


Figura 65 - Grafico tensione propulsore di dritta seconda prova pura imbardata

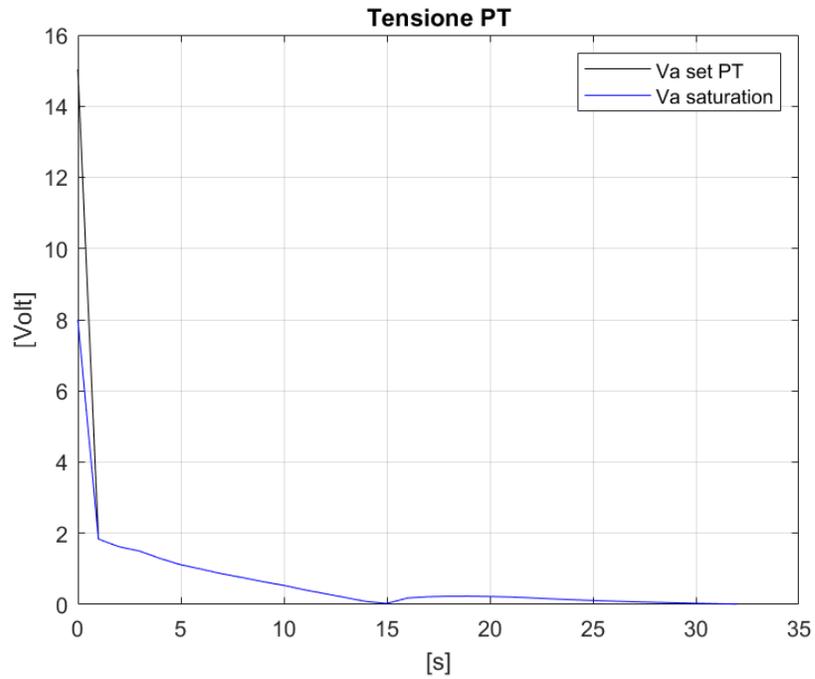


Figura 68 - Grafico tensione propulsore di sinistra seconda prova pura imbardata

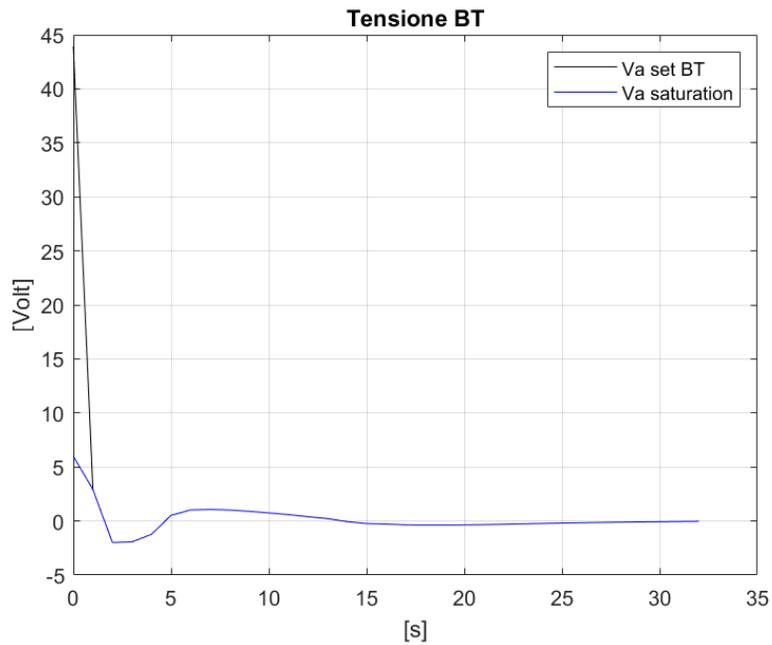


Figura 67 - Grafico tensione bow thruster seconda prova pura imbardata

Come si può notare dalla Figura 67 e Figura 68 la tensione del bow thruster e del propulsore di sinistra è notevolmente diminuita sia in termini di valori sia in termini di tempo.

È stata poi fatta una prova di controllo con una rotazione di 90° a sinistra.
 Di seguito i risultati in termini di posizione, velocità e tensione del propulsori:

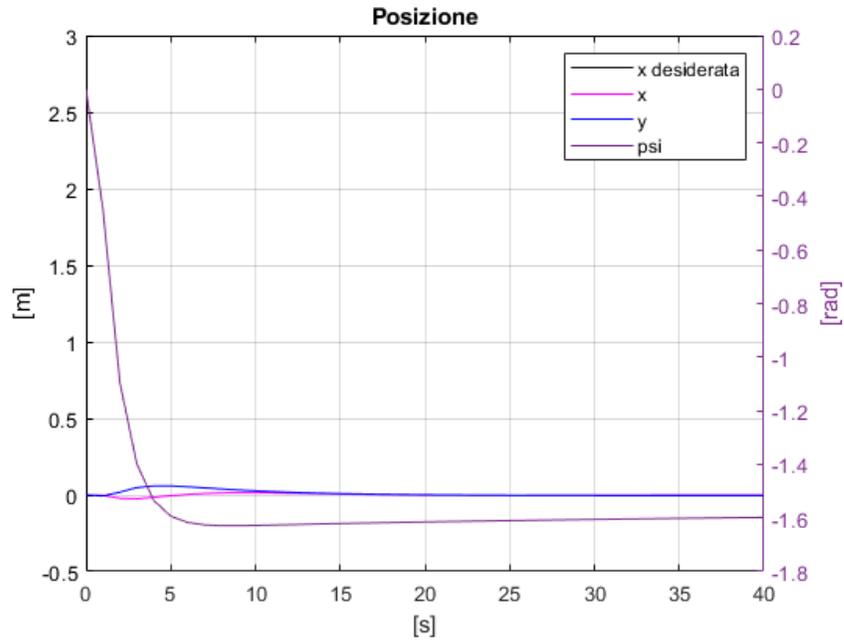


Figura 70 - Grafico posizione terza prova pura imbardata

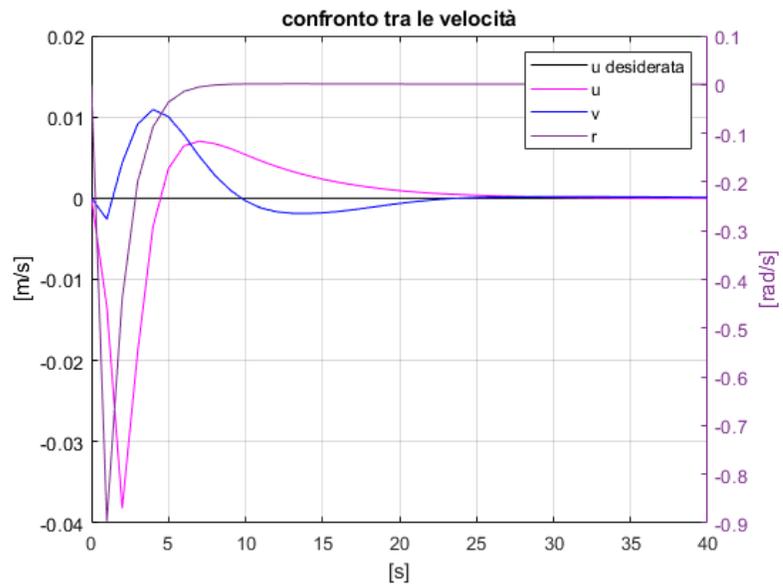


Figura 69 - Grafico velocità terza prova pura imbardata

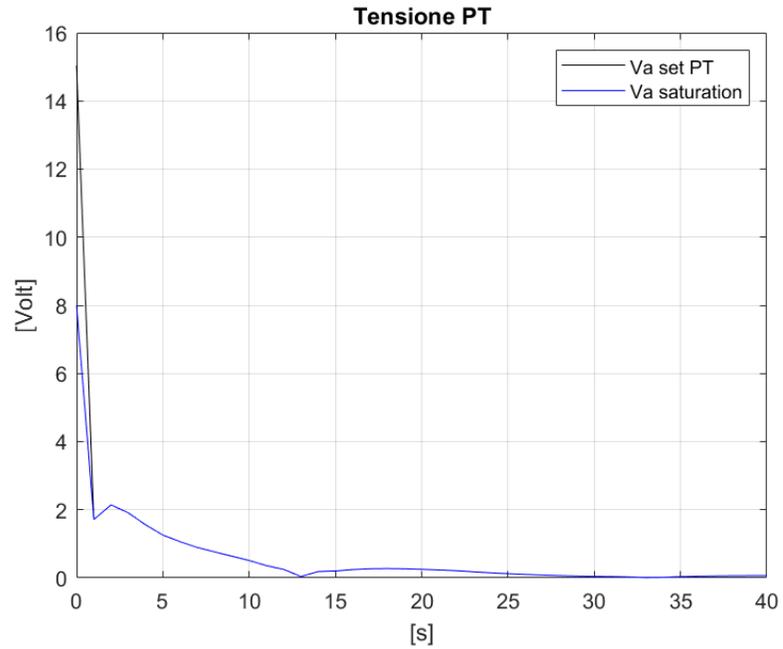


Figura 71 - Grafico tensione propulsore di sinistra terza prova pura imbardata

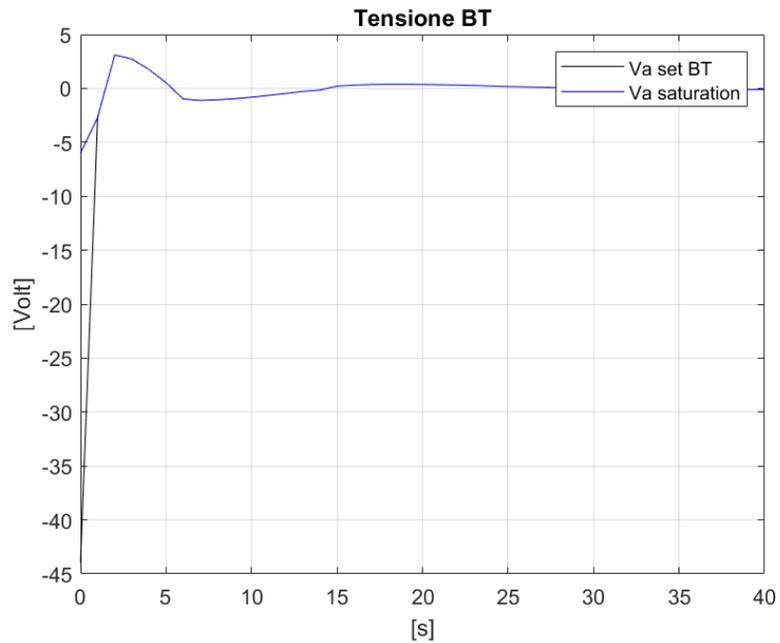


Figura 72 - Grafico tensione bow thruster

È interessante notare come nelle manovre di imbardata il modello è soggetto ad un arretramento, ossia l'imbarcazione durante la rotazione ha una velocità di avanzo "negativa".

5.4. Prova statica completa

Una volta terminate tutte le prove statiche, sono state effettuate delle prove in cui è richiesto uno spostamento sia lungo l'asse x che lungo l'asse y . È stato quindi chiesto al modello di raggiungere il punto di coordinate $[3,3]$ sia con un angolo di prua di 0° che con un angolo di prua di 45° , concorde alla direzione del vettore congiungente i due punti.

Riassumendo, sono state fatte due prove: una dove si raggiunge il punto $[3,3,0]$ e una dove si raggiunge il punto $[3,3,0.78]$. Di seguito è riportata la traiettoria risultante delle due manovre:

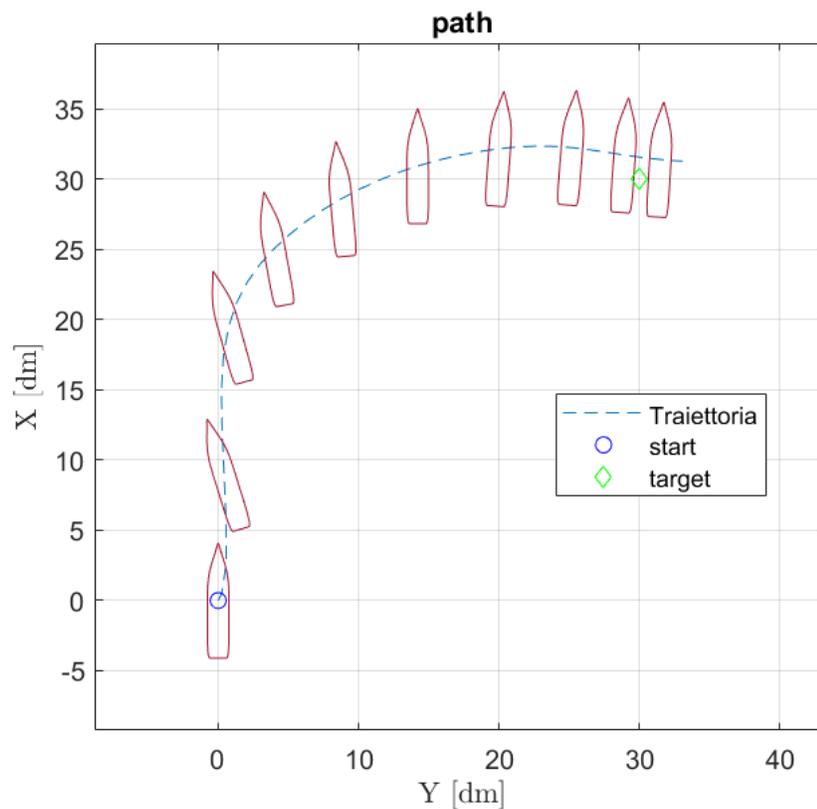


Figura 73 - Spostamento in x e y con prua a 0°

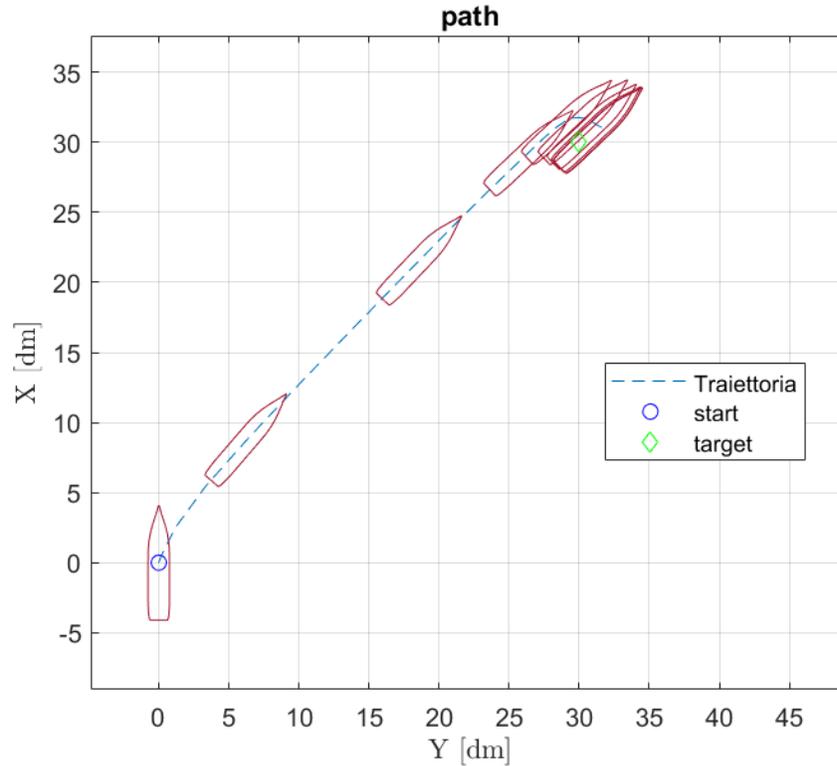


Figura 74 - Spostamento in x e y con prua a 45°

Come si può vedere nel caso in cui la prua richiesta è 0° allora la traiettoria si allontana da quella che è la rotta desiderata, ossia il segmento che congiunge il punto di partenza e il punto di arrivo, raffigurato in Figura 73 e Figura 74 in rosso, mentre nel caso della prova con angolo di prua concorde con il target allora il percorso effettuato dal modello meglio segue il segmento.

Questo è dovuto al fatto che la resistenza all'avanzo è minore di quella dovuta alla deriva come già accennato nel capitolo 6.2. in più i coefficienti vengono riferiti rispetto al sistema di riferimento inerziale, per tanto in questo elaborato si è data la preferenza al percorrere la traiettoria indicata con un angolo di prua concorde al segmento da seguire così come suggerisce anche la normale navigazione.

5.5. Prova di avanzo a velocità richiesta non nulla

Così come per le prove statiche è stata impostata la posizione di partenza coincidente con l'origine degli assi ed è stato chiesto al modello di raggiungere il punto di coordinate [3,0,0] con una velocità di avanzo $u = 0,1 \text{ m/s}$.

I coefficienti sono stati presi inizialmente come in Tabella 4 ed è stata avviata una simulazione di cui vengono riportati di seguito i risultati:

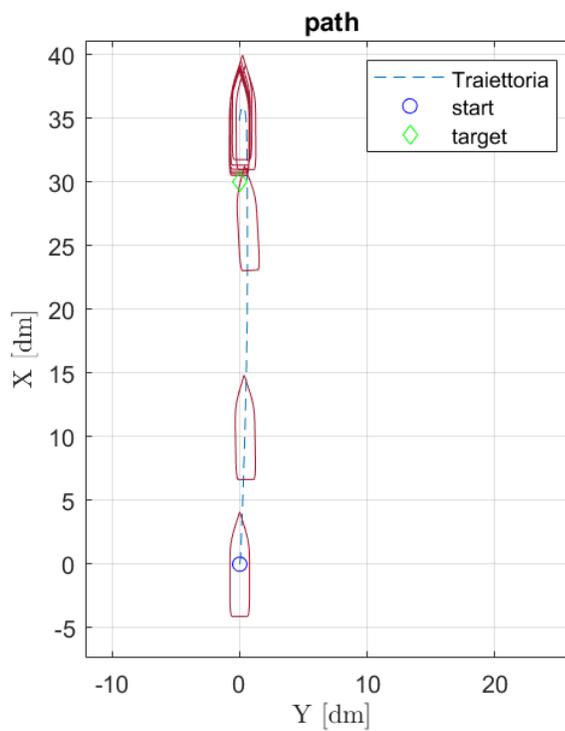


Figura 75 - Plotter prima prova avanzo a velocità richiesta non nulla

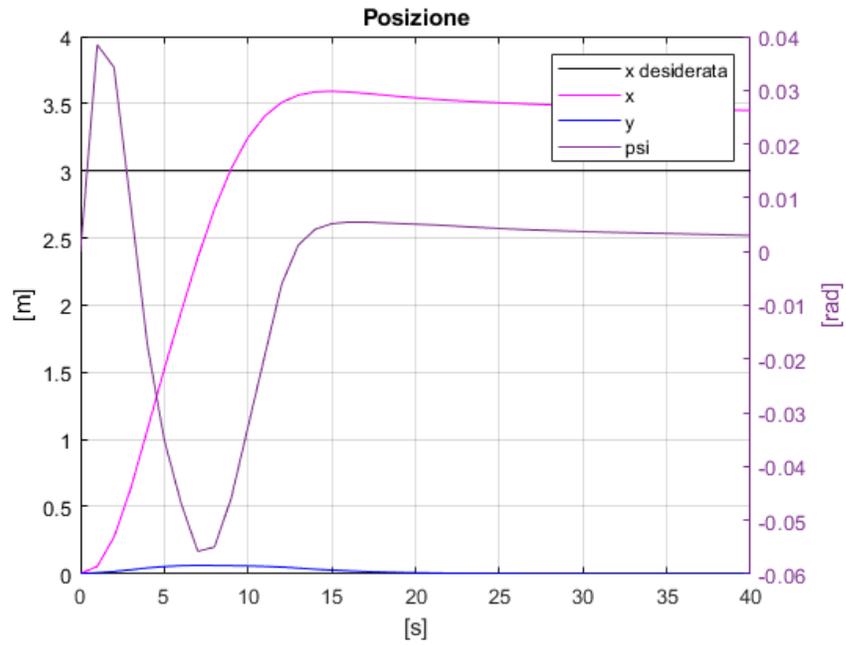


Figura 76 - Grafico velocità prima prova avanzo a velocità richiesta non nulla

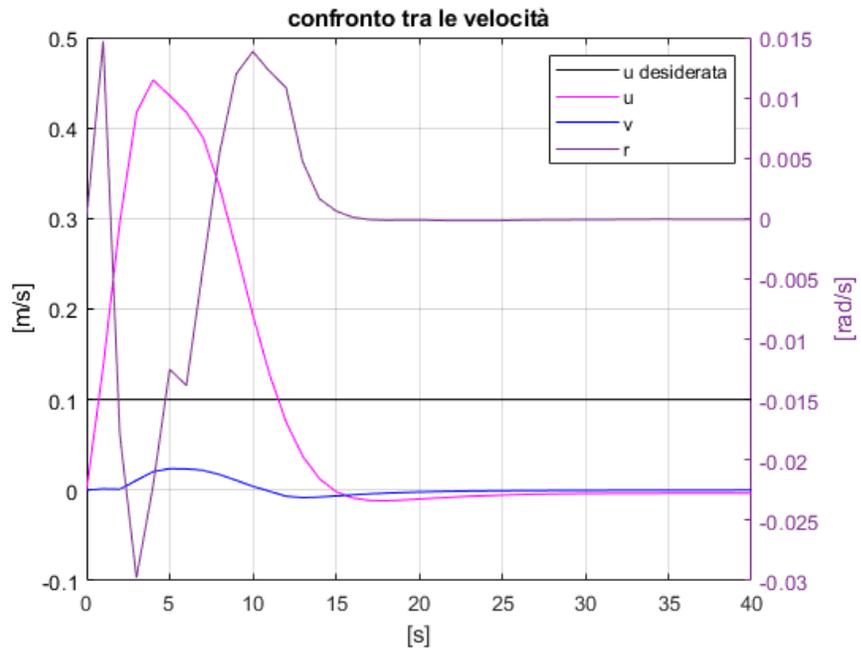


Figura 77 - Grafico velocità prima prova avanzo a velocità richiesta non nulla

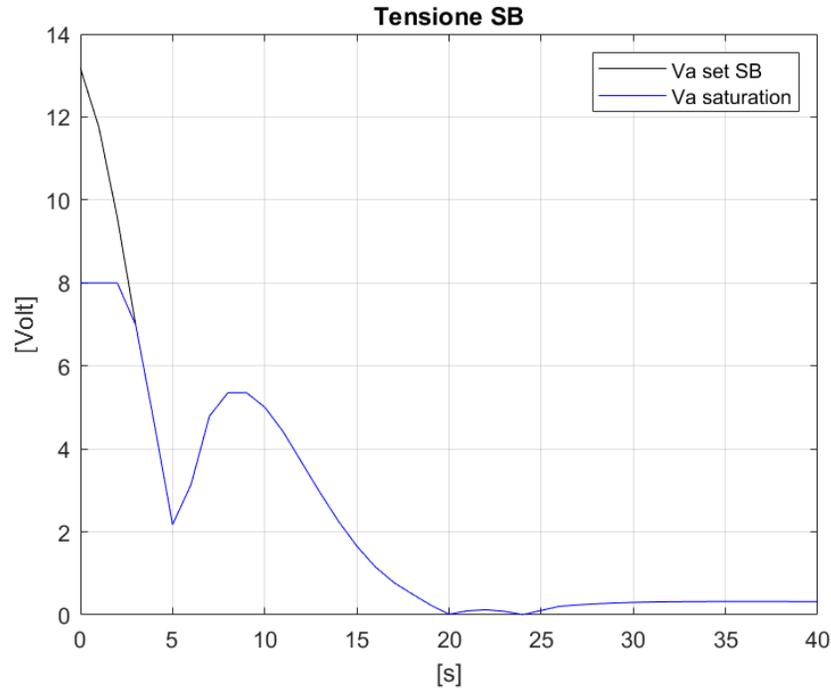


Figura 78 - Grafico tensione propulsore di dritta prima prova avanzo a velocità richiesta non nulla

Come si può vedere dalla Figura 77, la velocità di avanzo u supera il valore richiesto, pertanto è necessario abbassare i valori dei coefficienti K_p e aumentare i K_I in modo da aumentare l'influenza dell'errore di velocità ossia la differenza che c'è tra velocità attuale e quella desiderata così come descritto in Equazione 6.

Inoltre si può notare che anche il valore di tensione del propulsore di dritta richiesto all'inizio è superiore a quello massimo ammesso facendo andare il motore in saturazione.

Sono stati quindi modificati i coefficienti così come descritto in Tabella 5

Tabella 5 – Coefficienti seconda prova avanzo a velocità richiesta non nulla

	K_p	K_I	K_D
x	$0,01 \cdot K_{P_x}$	$1 \cdot K_{I_x}$	$100 \cdot K_{D_x}$
y	$0,01 \cdot K_{P_y}$	$1 \cdot K_{I_y}$	$100 \cdot K_{D_y}$
z	$0.05 \cdot K_{P_z}$	$1 \cdot K_{I_z}$	$1 \cdot K_{D_z}$

È stata effettuata un'ulteriore simulazione e disegnato sono esposti i risultati ottenuti:

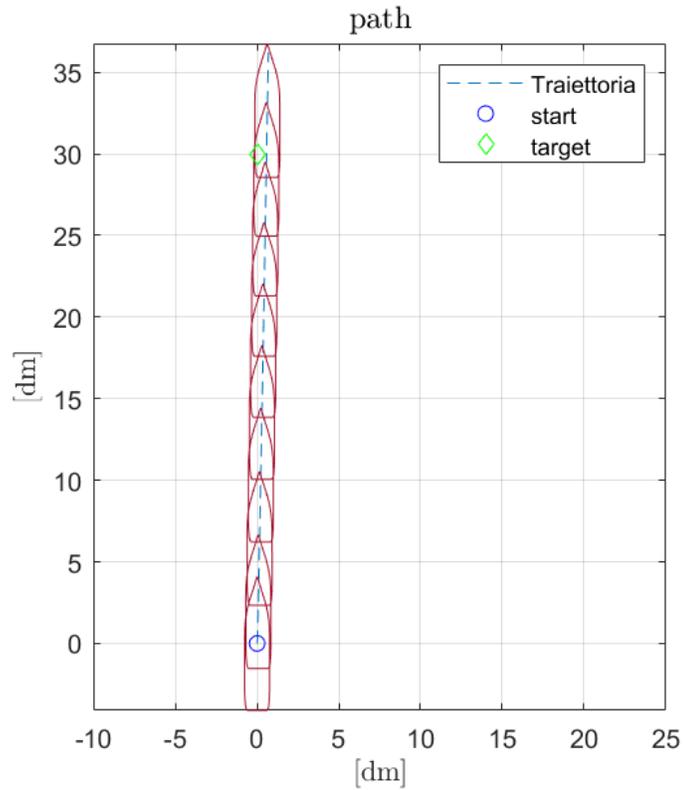


Figura 80 - Plotter seconda prova di avanzo a velocità richiesta non nulla

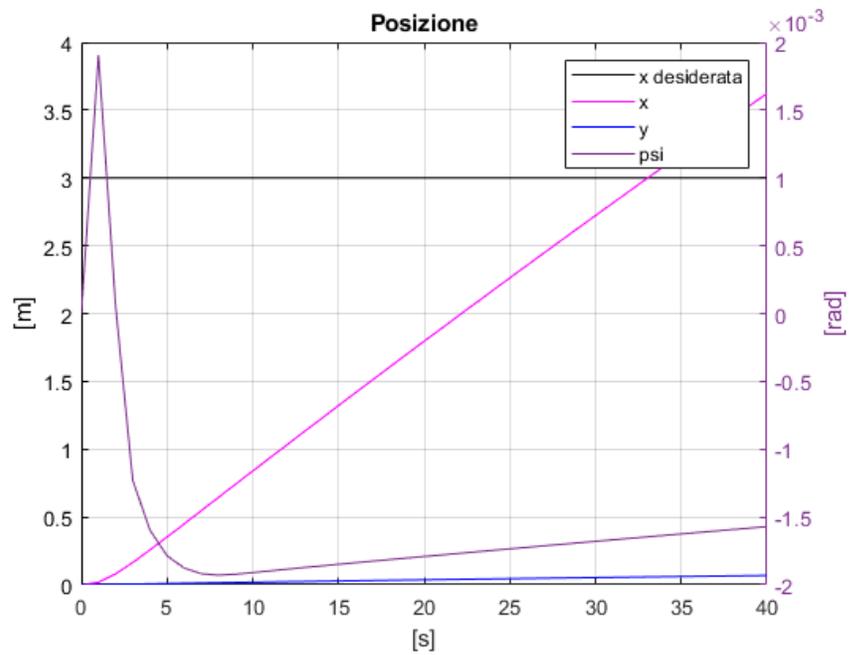


Figura 79 - Grafico posizione seconda prova avanzo a velocità richiesta non nulla

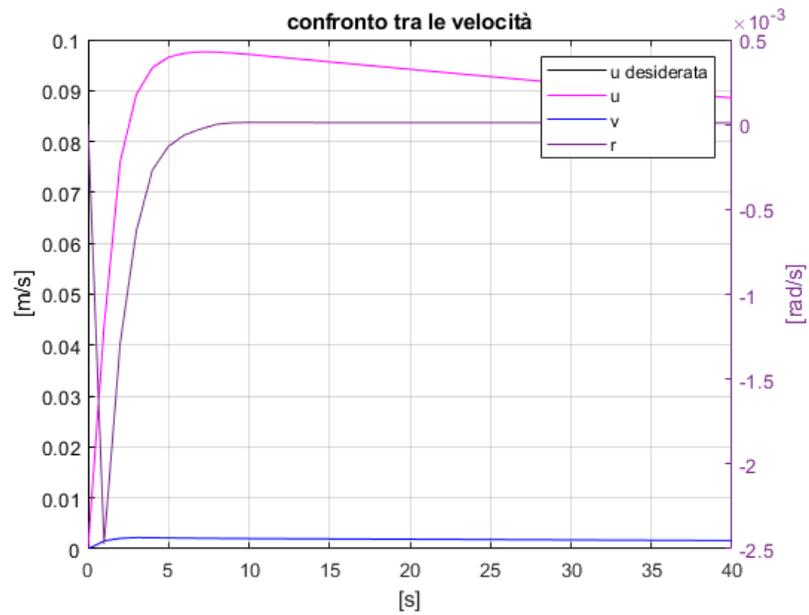


Figura 81 - Grafico velocità seconda prova di avanzo a velocità richiesta non nulla

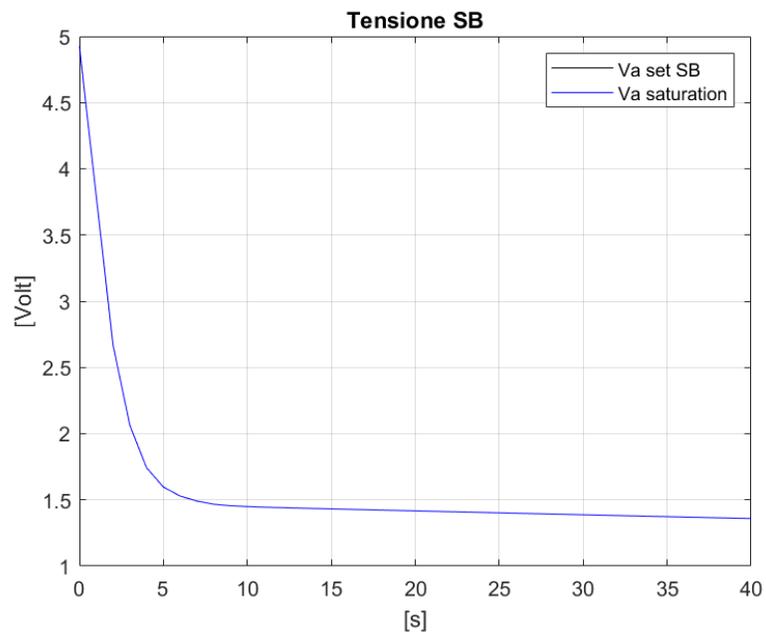


Figura 82 - Grafico tensione propulsore di dritta seconda prova di avanzo a velocità richiesta non nulla

Come si può notare nella Figura 80 e Figura 79 il modello raggiunge il target superandolo e continuando ad avanzare, questo perché avendo aumentato l'influenza dei coefficienti K_D e diminuito il valore di K_P , il modello darà più peso all'errore dovuto alla velocità $\dot{\eta}$, rispetto a quello dovuto alla posizione η . Nonostante questo però si può notare come l'influenza di η porta ad un rallentamento sempre più significativo all'aumentare dell'errore.

Dalla Figura 82 invece si può notare che il valore di tensione richiesto è sempre inferiore a quello ammissibile.

6. RISULTATI

Una volta impostati i vari coefficienti, riportati di seguito in Tabella 6 si sono eseguite le prove finali.

Tabella 6 – Coefficienti per prove finali

	K_P	K_I	K_D
x	$0,01 \cdot K_{P_x}$	$0 \cdot K_{I_x}$	$100 \cdot K_{D_x}$
y	$0,01 \cdot K_{P_y}$	$0 \cdot K_{I_y}$	$100 \cdot K_{D_y}$
z	$0.05 \cdot K_{P_z}$	$0 \cdot K_{I_z}$	$1 \cdot K_{D_z}$

Come si può notare i coefficienti sono simili a quelli utilizzati per la prova di avanzo a velocità richiesta non nulla, ad eccezione dei coefficienti del guadagno integrale che sono stati posti nulli in quanto non si è interessati alla storia temporale dell'errore in quanto per il principio di funzionamento del *waypoint management* è presente in tutta la prova.

Si è scelta una geometria specifica e si è definita la posizione di partenza e di arrivo, considerando non solo le coordinate del punto, ma anche l'angolo di prua desiderato. Si è quindi chiesto al modello di eseguire la simulazione, verificando che la manovra fosse effettuata correttamente senza collisioni.

Inizialmente, si è scelta la geometria più semplice, chiedendo al modello di raggiungere il punto di destinazione e di orientarsi secondo un angolo di prua specifico. Successivamente, si è proceduto alla geometria intermedia, richiedendo al modello di eseguire una manovra di attracco tra due ostacoli. Infine, si è passati alla geometria portuale più complessa, che meglio può rappresentare un porto reale con i relativi ostacoli.

6.1. Geometria portuale semplice

In questa simulazione è stato richiesto al modello, partendo dal punto di coordinate $[1,1,0]$, di raggiungere il punto $[4.5,4.5,3.14]$. Durante la manovra quindi il modello dovrà inizialmente orientarsi in direzione della traiettoria desiderata poi procedere fino al target e infine dovrà effettuare un'ulteriore rotazione affinché raggiunga un valore di angolo di prua di $3,14$ *radianti* equivalenti a 180° .

Di seguito sono riportati i risultati ottenuti:

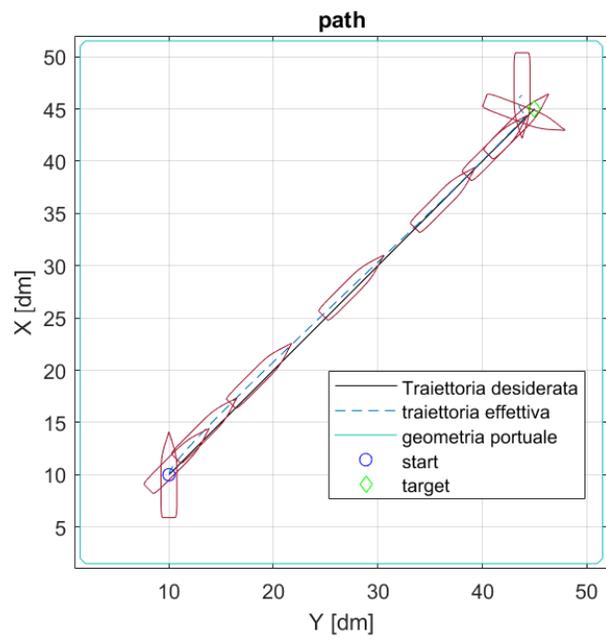


Figura 83 - Plotter prova finale geometria semplice

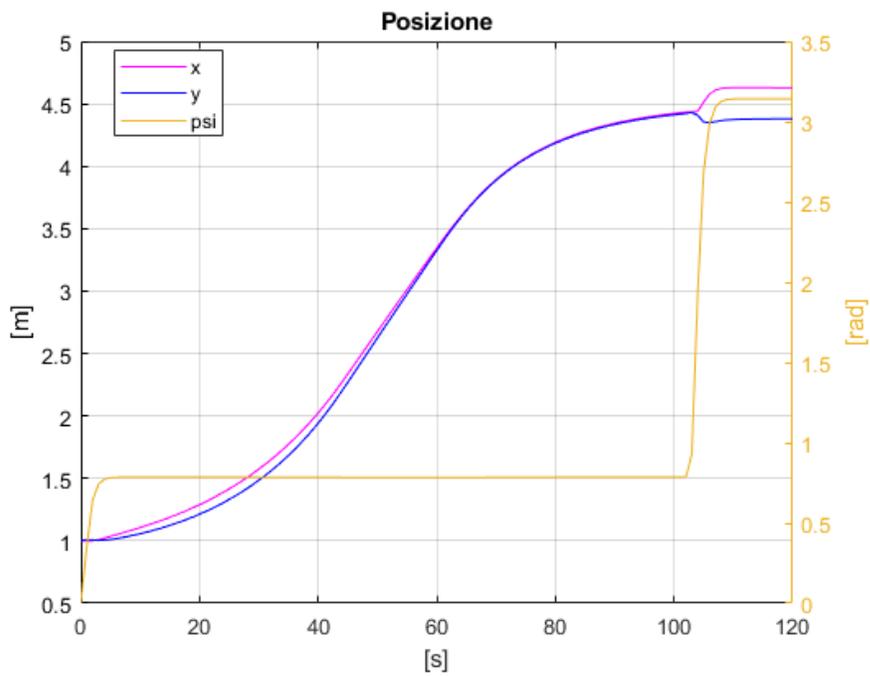


Figura 84 - Grafico posizione prova finale geometria semplice

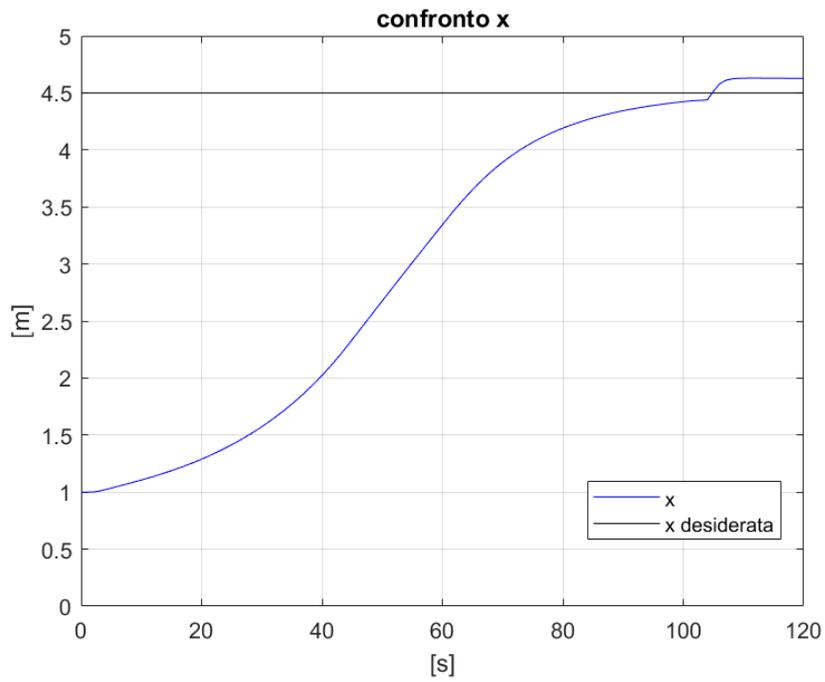


Figura 86 - Confronto x prova finale geometria semplice

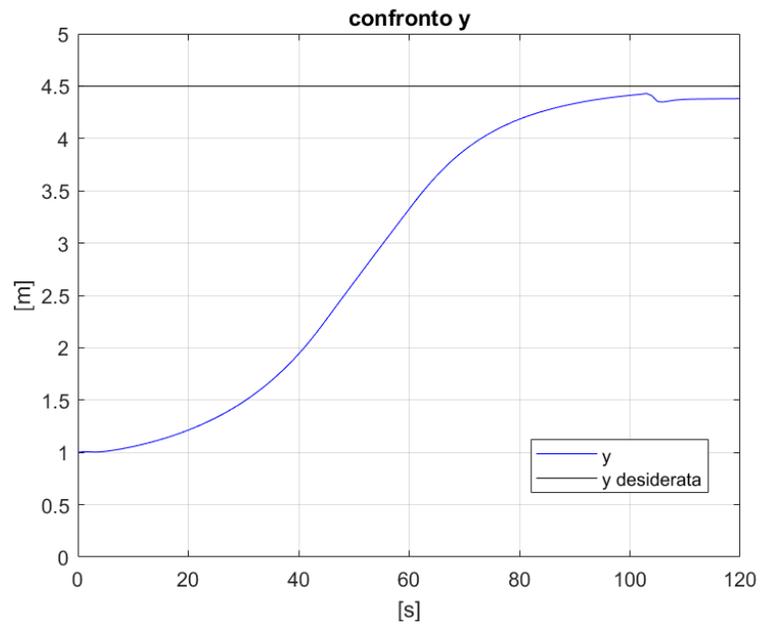


Figura 85 - Confronto y prova finale geometria semplice

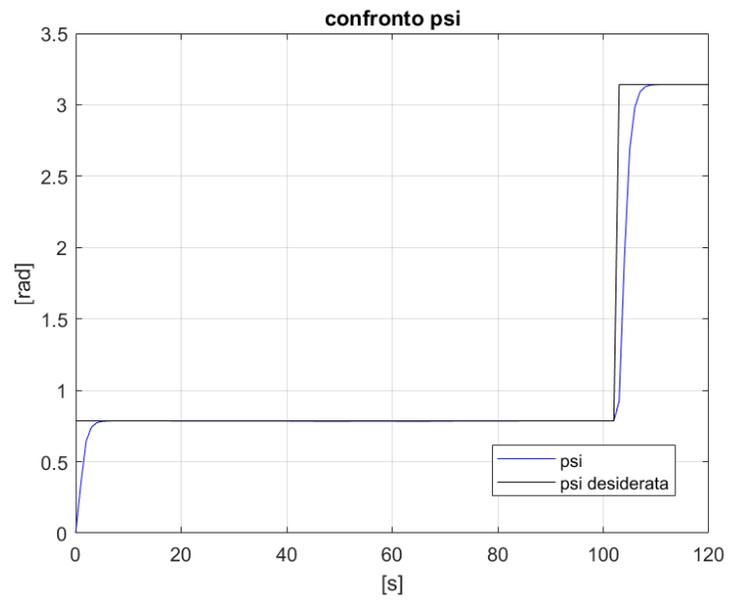


Figura 87 - Confronto ψ prova finale geometria semplice

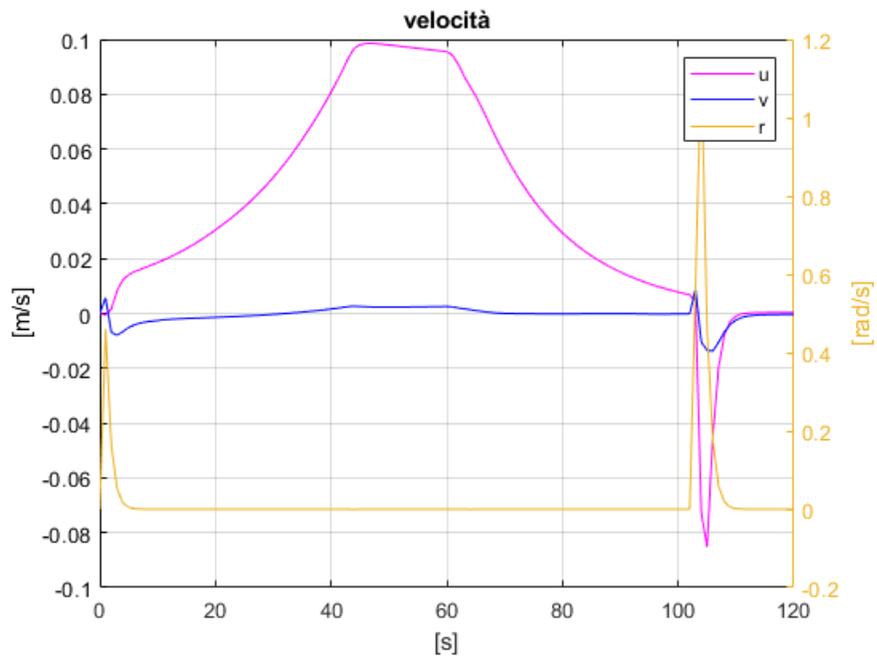


Figura 88 - Grafico velocità prova finale geometria semplice

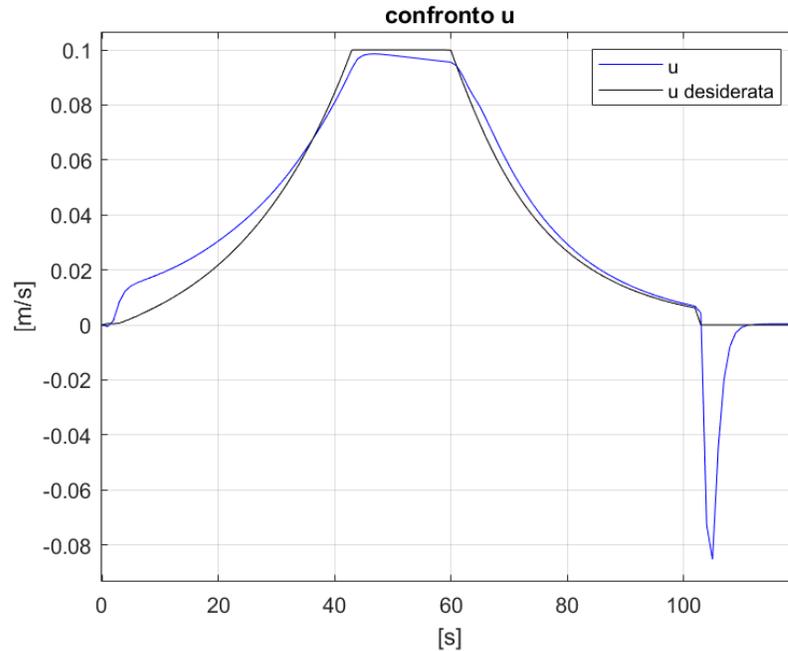


Figura 89 - Confronto velocità u prova finale geometria semplice

I grafici mostrano che il modello segue abbastanza fedelmente la traiettoria e la velocità richiesta. Tuttavia, in Figura 88 e Figura 89, si può notare un picco negativo di velocità u durante la rotazione finale. Questo fenomeno è dovuto alla rapida rotazione sul posto che genera un leggero arretramento e una traslazione.

6.2. Geometria portuale intermedia

In questa simulazione, si è richiesto al modello, partendo dal punto di coordinate [4.5,4.5,0], di raggiungere il punto [0.8, 3.8, 3.14], evitando gli ostacoli che rappresentano la geometria portuale e inserendosi tra due ostacoli che potrebbero rappresentare due moli o imbarcazioni ormeggiate. L'obiettivo era di valutare il comportamento del modello nei vari waypoint della traiettoria, seguendo le specifiche descritte nel paragrafo **Errore. L'origine riferimento non è stata trovata.** Si richiedeva infatti che il modello accelerasse inizialmente fino a raggiungere la velocità v_{set} e successivamente decelerasse man mano che ci si avvicina al *waypoint*, fino a raggiungere la velocità minima v_{min} .

Una volta che la distanza dal punto di arrivo fosse inferiore alla distanza d_{skip} , il sistema dovrebbe indicare le nuove coordinate. L'obiettivo era verificare che in queste situazioni il modello fosse in grado di rallentare adeguatamente, effettuare un'imbardata fino al raggiungimento della nuova direzione e avanzare fino al successivo *waypoint*.

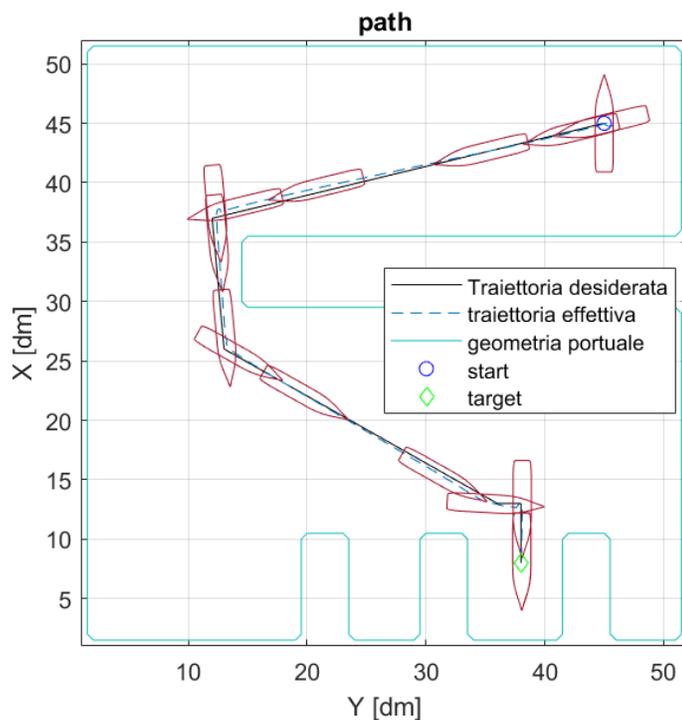


Figura 90 - Plotter prova finale geometria intermedia

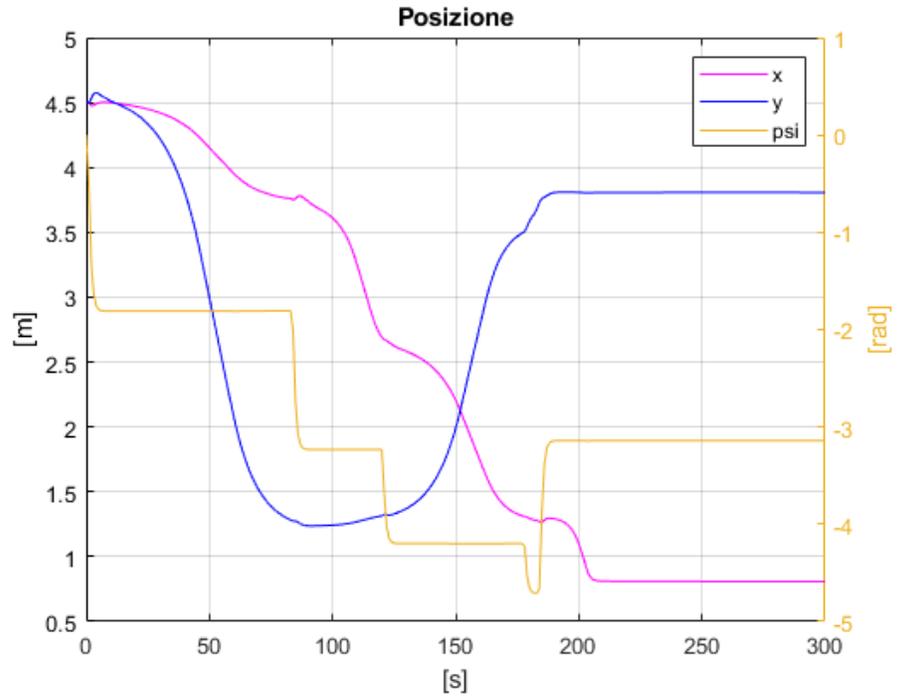


Figura 92 - Grafico posizione prova finale geometria intermedia

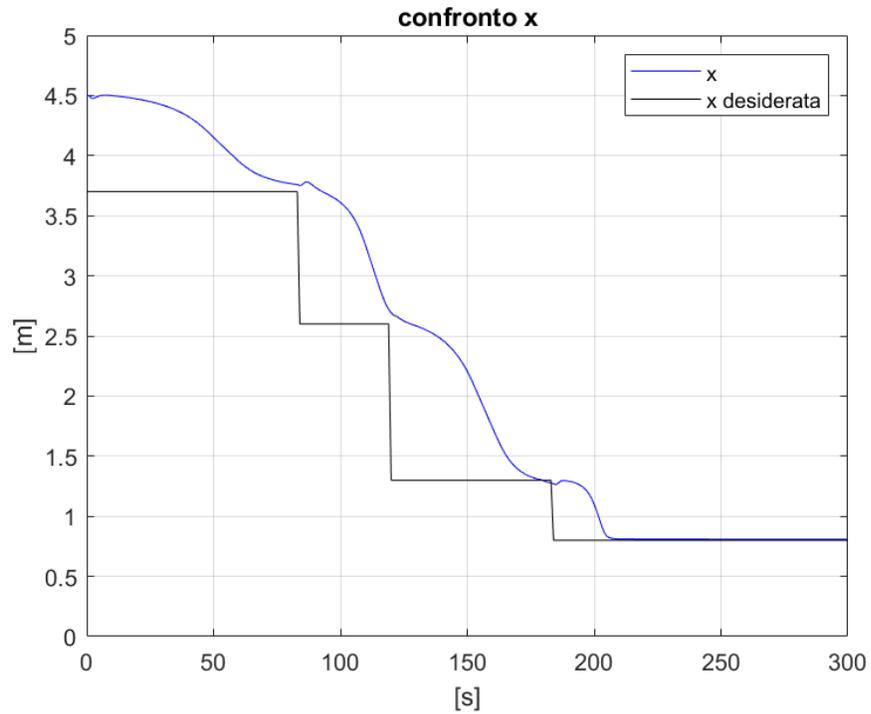


Figura 91 - Confronto x prova finale geometria intermedia

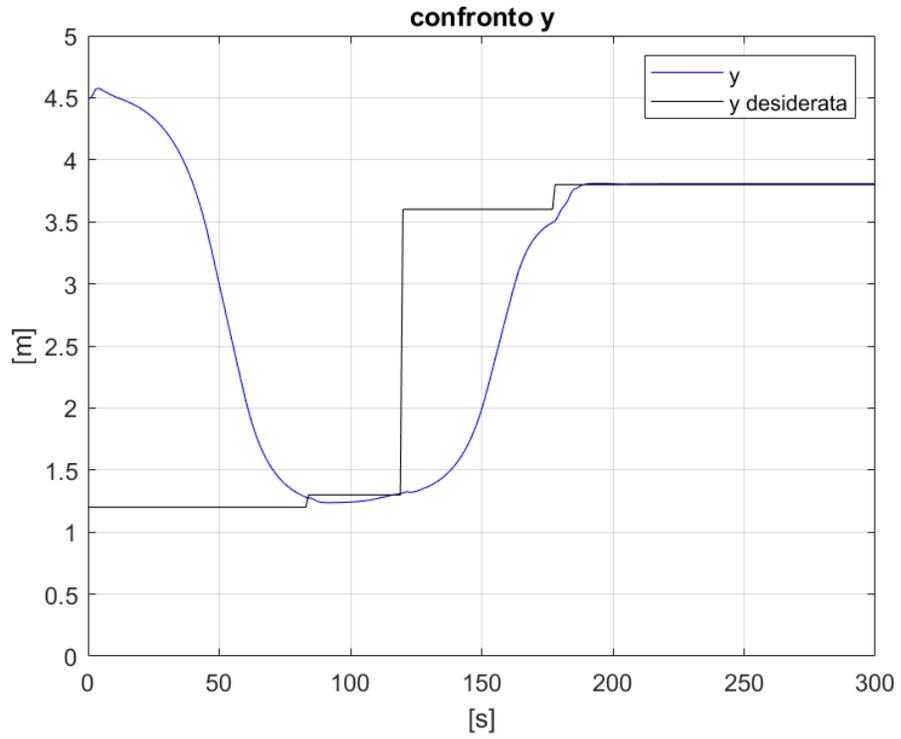


Figura 93 - Confronto y prova finale geometria semplice

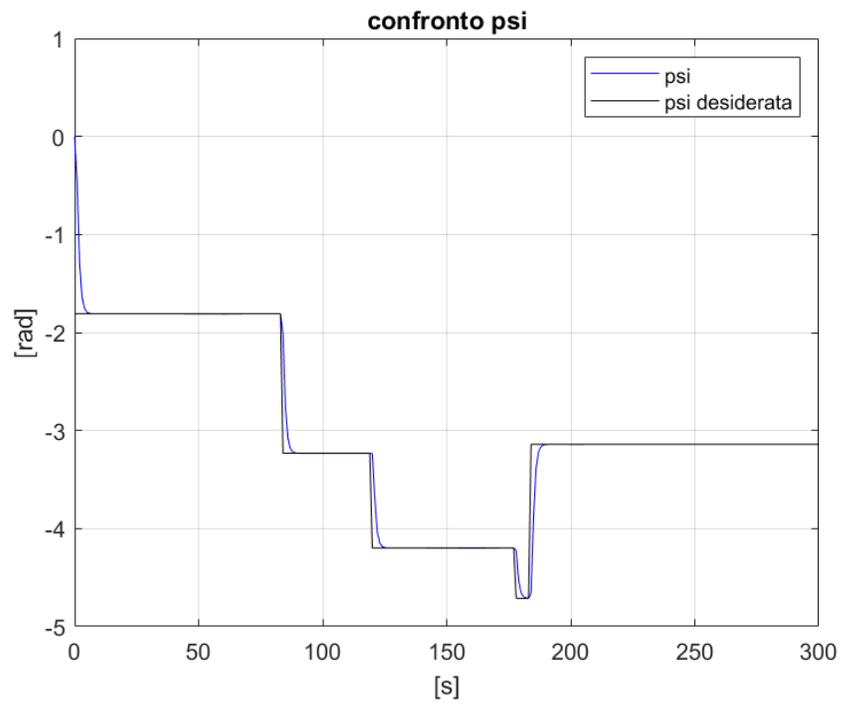


Figura 94 - Confronto psi prova finale geometria intermedia

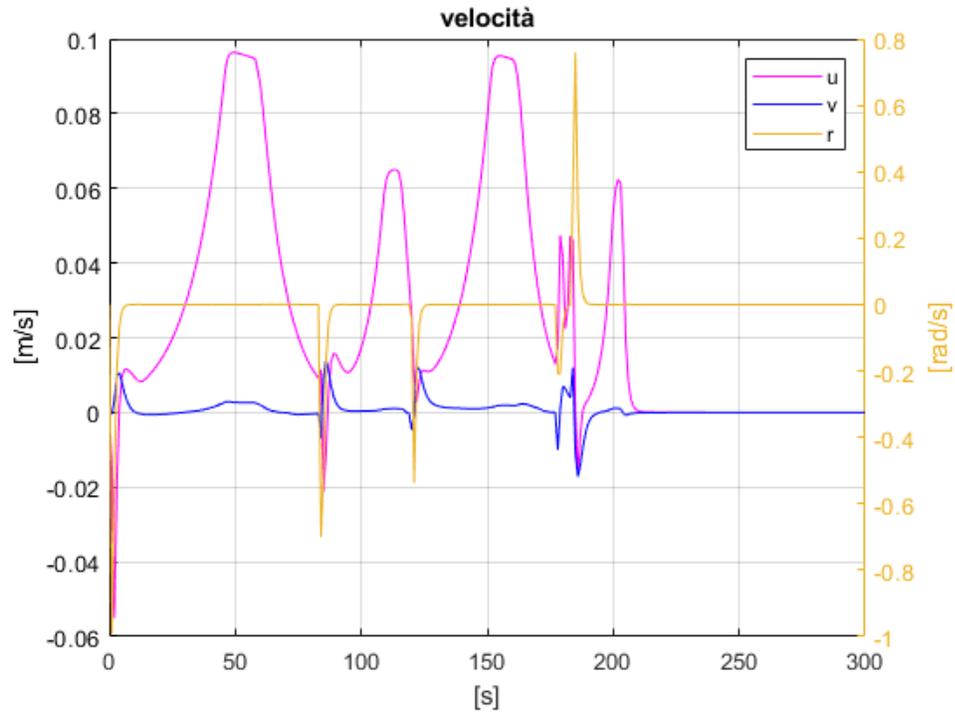


Figura 96- Grafico velocità prova finale geometria intermedia

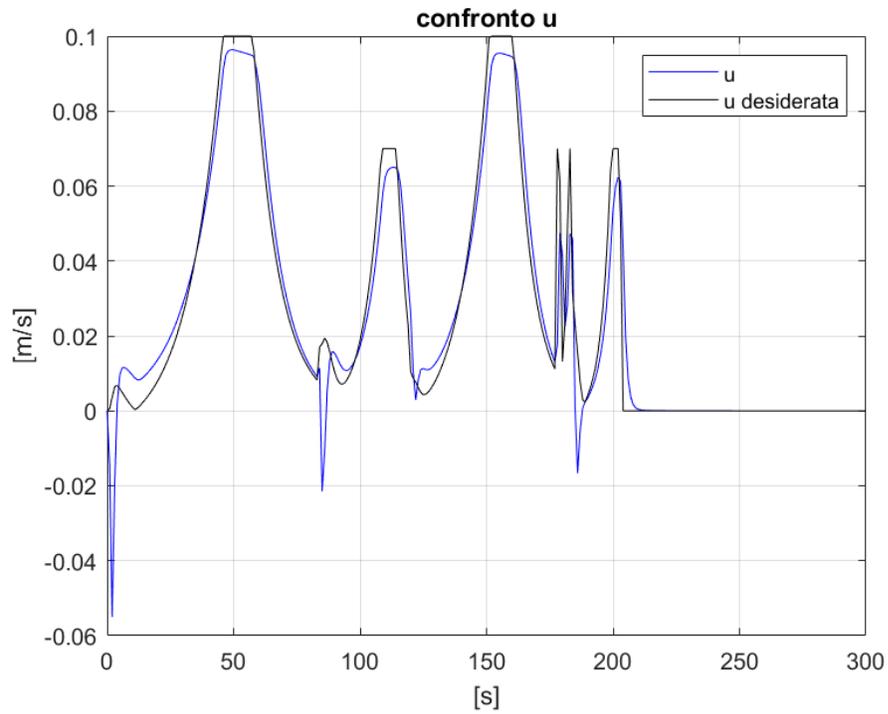


Figura 95 - Confronto velocità u prova finale geometria intermedia

Come si può notare dalla Figura 95 anche in questo caso, si verificano dei picchi di velocità u negativi in prossimità dei *waypoint*, a causa delle imbardate del modello. Tuttavia, è importante sottolineare che tali imbardate sono di bassa intensità e di durata molto breve. Come si può vedere dalla Figura 90 e Figura 92, tali picchi non influiscono significativamente sulla posizione del modello.

6.3. Geometria portuale complessa

In questa simulazione si è chiesto al modello di partire dal punto di coordinate $[4.5, 4.5, 0]$, di raggiungere il punto $[0.8, 4.3, 3.14]$, evitando gli ostacoli presenti lungo il percorso e iniziando e terminando il percorso da una posizione simile a quella di ormeggio. Questa prova ha simulato un percorso completo che ha incluso la navigazione in prossimità degli ostacoli e una rotazione finale sul posto, al fine di verificare se il modello è in grado di seguire la traiettoria designata anche in queste condizioni. Di seguito sono riportati i risultati ottenuti:

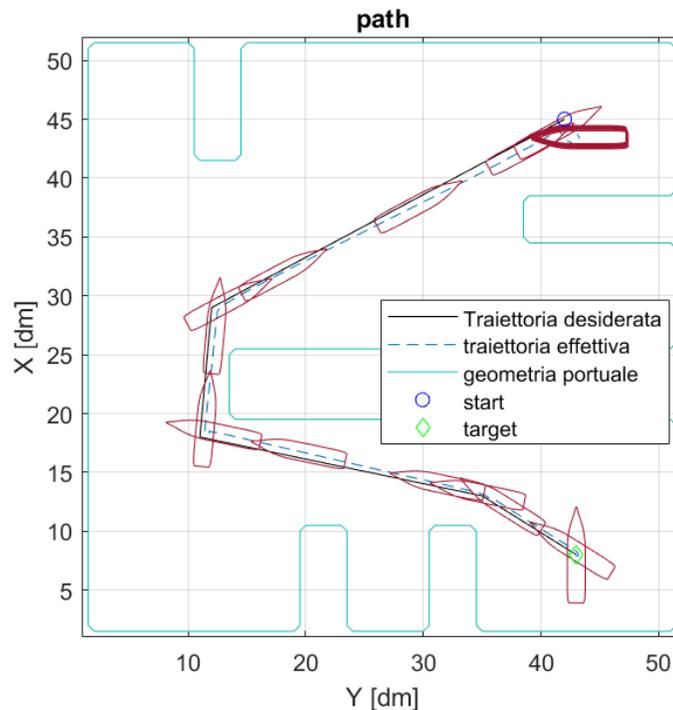


Figura 97 - Plotter prova finale geometria complessa

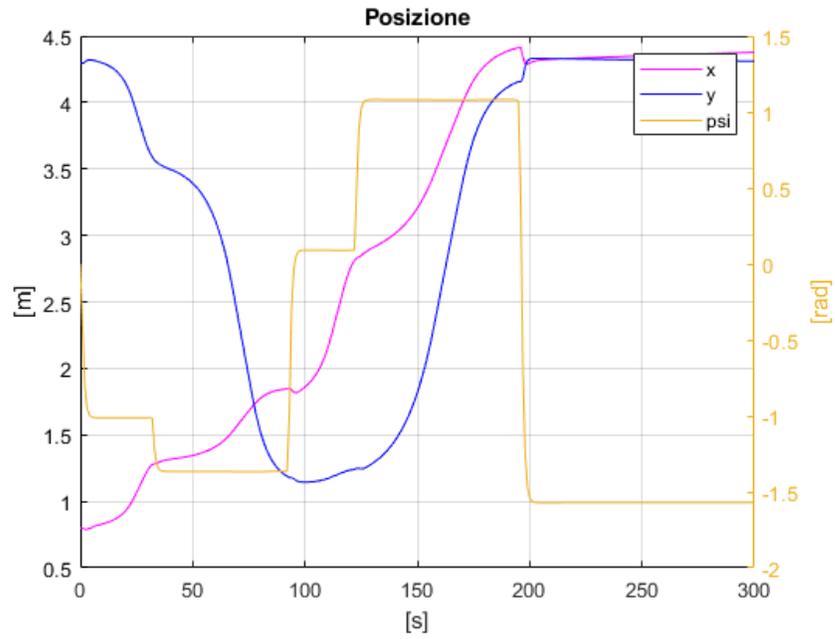


Figura 98 - Grafico posizione prova finale geometria complessa

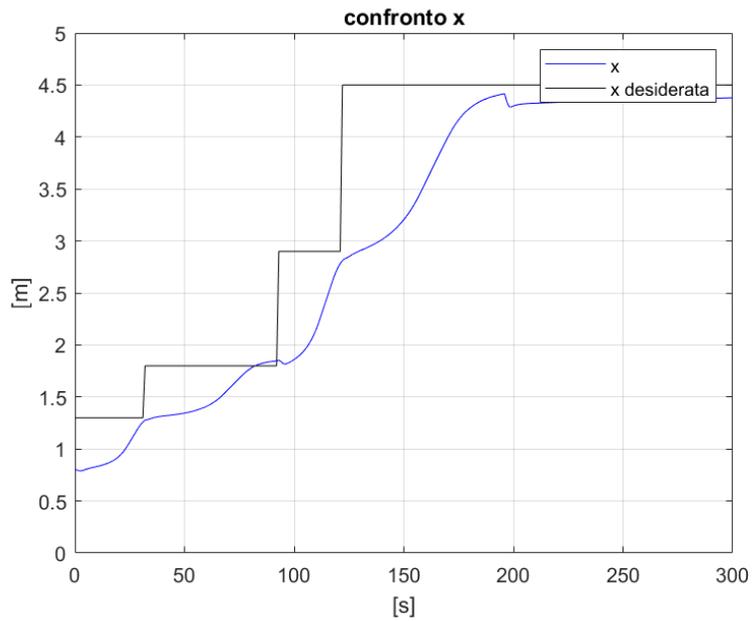


Figura 99 - Confronto x prova finale geometria complessa

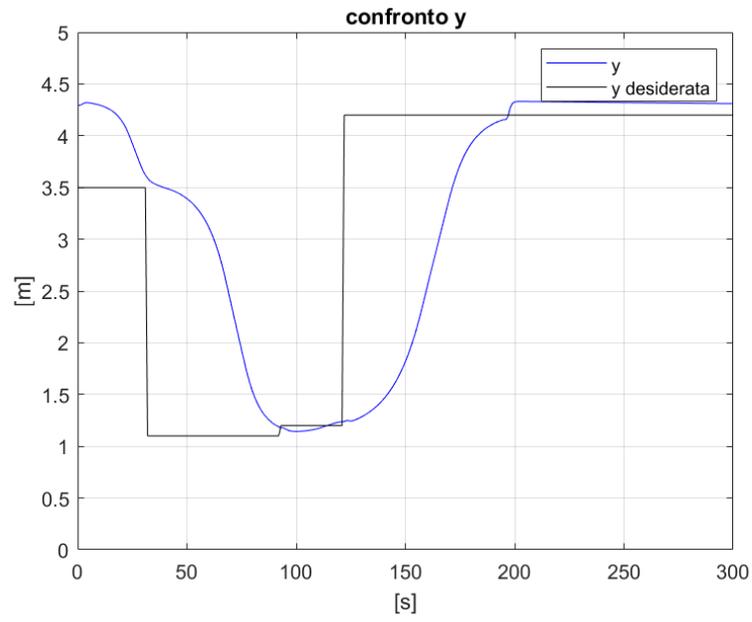


Figura 100 - Confronto y prova finale geometria complessa

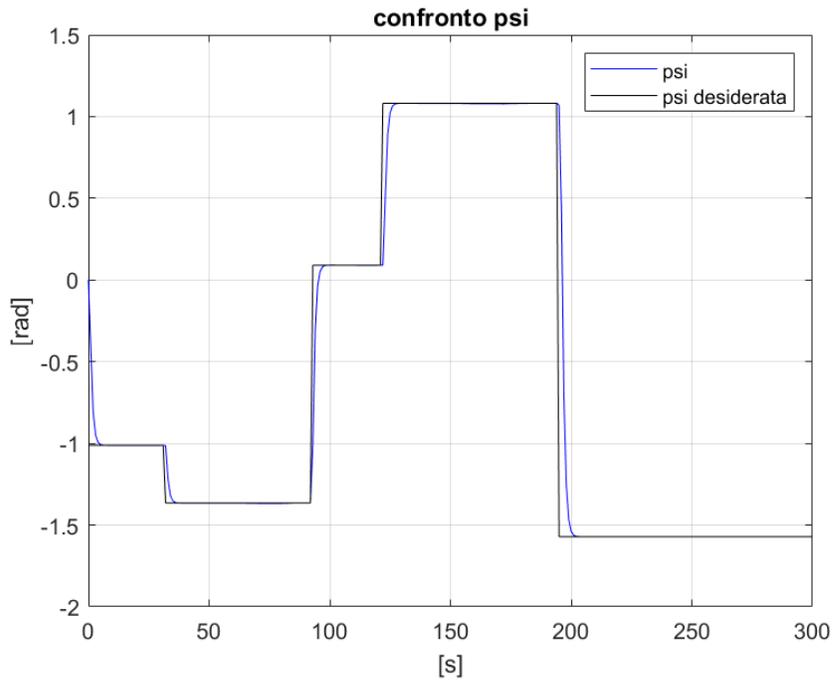


Figura 101 - Confronto psi prova finale geometria complessa

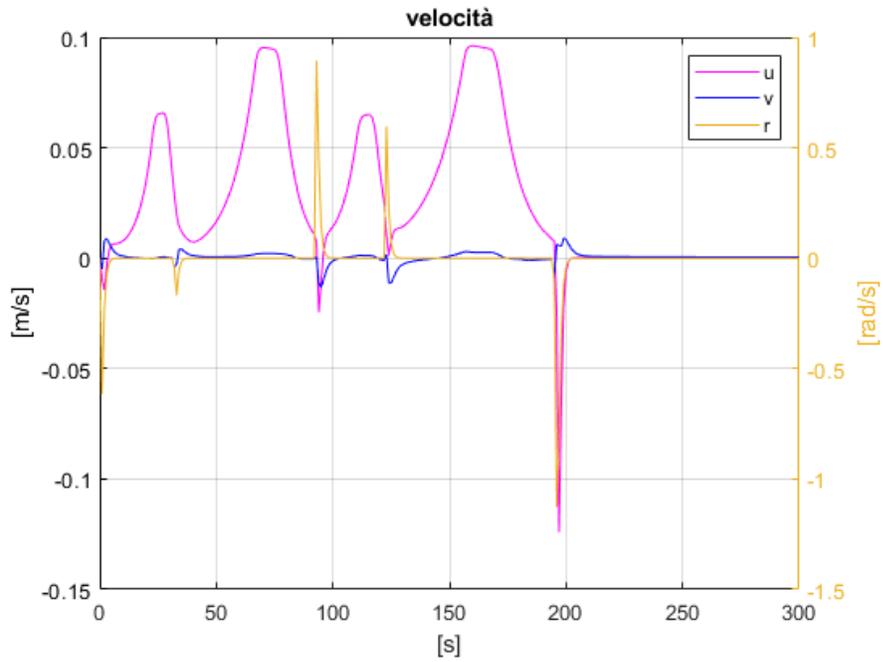


Figura 103 - Grafico velocità prova finale geometria complessa

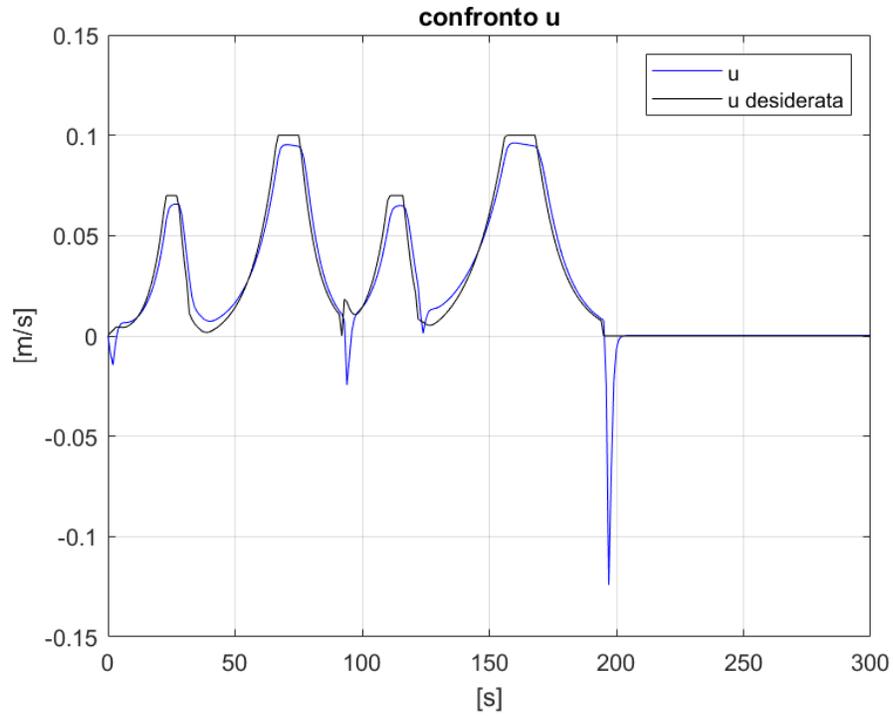


Figura 104 - Confronto velocità u prova finale geometria complessa

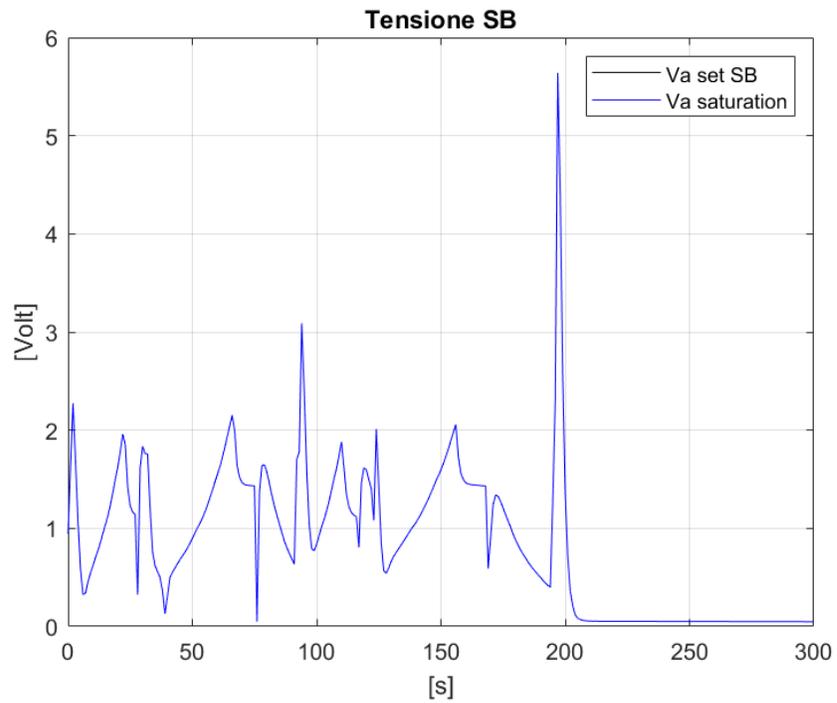


Figura 105 - Grafico tensione propulsore di dritta prova finale geometria complessa

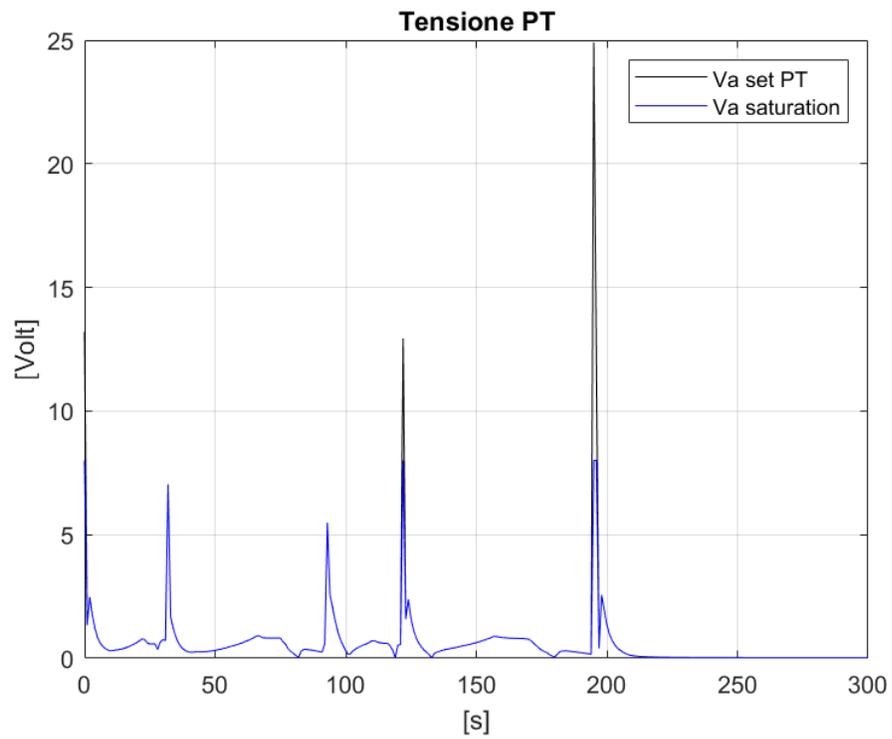


Figura 106 - Grafico tensione propulsore di sinistra prova finale geometria complessa

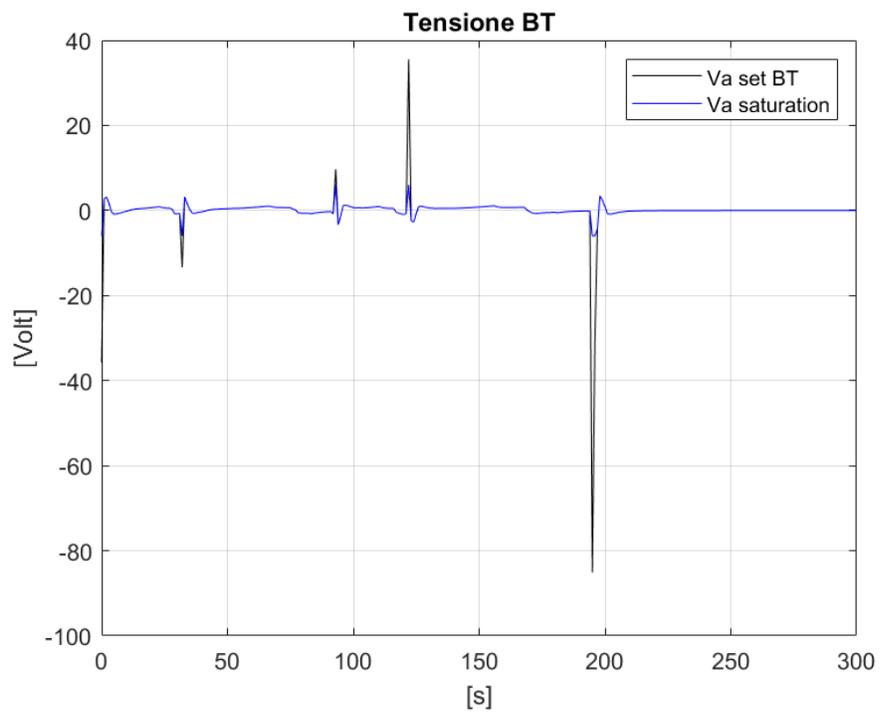


Figura 107 - Grafico tensione bow thruster prova finale geometria complessa

Come si può notare dalla Figura 97 la manovra viene eseguita correttamente ad eccezione dell'arrivo, dove il modello si allontana leggermente dal target per effettuare la rotazione sul posto. Questo problema può essere risolto in due modi: aumentando i coefficienti K_{p_x} e K_{p_y} in modo che l'errore di posizione influisca maggiormente sul vettore forza risultante descritto nell'Equazione 6 $\tau_{control}$ oppure riducendo il valore di K_{p_z} in modo da effettuare la rotazione più lentamente. Tuttavia, la seconda soluzione comporterebbe un ritardo nell'esecuzione della manovra di imbardata, che deve essere effettuata rapidamente per evitare che il modello segua una traiettoria con un angolo di prua diverso da quello richiesto. Inoltre, va notato che ridurre il valore di K_{p_z} comporterebbe un abbassamento del valore di tensione richiesta dal bow thruster e dal propulsore di sinistra, come si può vedere dalle Figura 106 e Figura 107 presentano picchi isolati con valori ben oltre quelli consentiti.

Un'altra soluzione possibile per evitare questo problema potrebbe essere quella di aggiungere ulteriori *waypoint* per imporre la rotazione in un punto specifico e verificare l'angolo di prua, in modo che il modello effettui la rotazione sul posto e solo una volta raggiunto l'angolo di prua desiderato possa quindi procedere verso le nuove coordinate.

Inoltre, durante i tratti rettilinei si notano valori non nulli della tensione del propulsore di dritta. Questo è dovuto all'allocazione del modello, che presenta il propulsore di dritta leggermente decentrato per soddisfare la spinta necessaria all'avanzamento. Questo comporta un momento che tende a far ruotare l'imbarcazione e genera quindi un errore dell'angolo di prua.

7. CONCLUSIONI E SVILUPPI FUTURI

In questo elaborato è stato sviluppato un sistema di auto ormeggio basilare, in grado di creare un percorso libero da ostacoli che collega il punto di partenza a quello di arrivo, e di guidare il modello in modo opportuno per eseguire correttamente la manovra. Sulla base dei risultati ottenuti dalle numerose prove, si può concludere che il modello così impostato costituisce una buona base di partenza per effettuare prove in vasca al fine di controllare il corretto funzionamento e tarare ulteriormente i coefficienti.

Dopo aver collegato e tarato al meglio il modello, per rendere il sistema il più efficace possibile, sarebbe auspicabile effettuare alcune migliorie. Ad esempio, è fondamentale che il sistema tenga in considerazione le varie condizioni meteomarine che possono perturbare l'equilibrio dell'imbarcazione e implementi un sistema di controllo in grado di individuare eventuali ostacoli mobili o non segnalati per evitare collisioni con altre imbarcazioni.

Inoltre, sarebbe possibile migliorare la scelta di schematizzazione, come spiegato in 0, rappresentando lo spazio fisico mediante uno spazio virtuale rappresentato da una matrice. Tuttavia, come visto in **Errore. L'origine riferimento non è stata trovata.**, il tempo computazionale aumenta esponenzialmente con l'aumentare del grado di dettaglio o della porzione di spazio che si vuole raffigurare. Si potrebbe quindi pensare di effettuare una schematizzazione che permetta di trovare una soluzione, come descritta in [25], che permetta di utilizzare come punti da inserire nell'*expanded_array* non solo i punti strettamente adiacenti ma anche quelli presenti in un intorno di una certa distanza, similmente al metodo del "*vector based approach*" descritto in [14]. Nel caso in cui si è vicini al target, si potrebbe tornare al metodo descritto ed utilizzato in questo elaborato. Sarebbe quindi interessante effettuare uno studio sui vari algoritmi applicabili con le relative schematizzazioni per identificare l'approccio migliore sia in termini di peso computazionale sia in termini di sicurezza ed affidabilità del risultato.

Un'altra possibile miglioria riguarda la distanza di sicurezza d_{safe} , che potrebbe essere modificata in funzione della posizione e della velocità dell'imbarcazione, come avviene nella navigazione classica. Nel caso di ormeggio o disormeggio, la distanza di sicurezza potrebbe essere ridotta data la scarsità di spazio di manovra così come anche la velocità di esecuzione, mentre durante lo spostamento si potrebbe aumentare questa distanza per garantire una maggiore sicurezza dagli ostacoli.

Infine, sarebbe interessante testare il sistema su imbarcazioni con diverse configurazioni (numero e tipologie di propulsori) e orientazioni per valutarne le prestazioni in diverse condizioni di navigazione. È importante notare che l'allocazione propulsori scelta si è dimostrata efficace per gli spostamenti minimi a basse velocità tipici del DP, ma non è adatta per una navigazione a lungo raggio.

BIBLIOGRAFIA

- [1] “Ministero delle Infrastrutture e dei Trasporti RAPPORTO SUI SINISTRI MARITTIMI e sugli infortuni a bordo delle navi.”
- [2] “ANNUAL OVERVIEW OF MARINE CASUALTIES AND INCIDENTS 2021.”
- [3] “Recreational-Boating-Statistics-2021”.
- [4] A. Alessandri, S. Donnarumma, M. Martelli, and S. Vignolo, “Motion control for autonomous navigation in blue and narrow waters using switched controllers,” *J Mar Sci Eng*, vol. 7, no. 6, 2019, doi: 10.3390/jmse7060196.
- [5] M. Martelli and D. D. Belvisi, “Dispense del Corso di Impianti di Propulsione Navale.”
- [6] “Michele Martelli Marine Propulsion Simulation: methods and results,” 2015.
- [7] V. S. Nguyen and N. K. Im, “Automatic ship berthing based on fuzzy logic,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 19, no. 3, pp. 163–171, 2019, doi: 10.5391/IJFIS.2019.19.3.163.
- [8] L. A. Zadeh, “Fuzzy Logic,” *Computer (Long Beach Calif)*, vol. 21, no. 4, pp. 83–93, 1988, doi: 10.1109/2.53.
- [9] N. K. Im and V. S. Nguyen, “Artificial neural network controller for automatic ship berthing using head-up coordinate system,” *International Journal of Naval Architecture and Ocean Engineering*, vol. 10, no. 3, pp. 235–249, May 2018, doi: 10.1016/j.ijnaoe.2017.08.003.
- [10] N. Sariff and N. Buniyamin, “An overview of autonomous mobile robot path planning algorithms,” in *SCORED 2006 - Proceedings of 2006 4th Student Conference on Research and Development “Towards Enhancing Research Excellence in the Region,”* 2006, pp. 183–188. doi: 10.1109/SCORED.2006.4339335.
- [11] P. E. Hart, N. J. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968, doi: 10.1109/TSSC.1968.300136.
- [12] H. Wang, Y. Yu, and Q. Yuan, “Application of Dijkstra algorithm in robot path-planning,” in *2011 2nd International Conference on Mechanic Automation and Control Engineering, MACE 2011 - Proceedings*, 2011, pp. 1067–1069. doi: 10.1109/MACE.2011.5987118.
- [13] T. Lozano-P6rez and M. A. Wesley, “An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles,” 1979.
- [14] C. W. Warren, “A vector based approach to robot path planning,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 1991, vol. 2, pp. 1021–1026. doi: 10.1109/robot.1991.131725.
- [15] “[https://www.volvopenta.com/marine/accessories/joystick-control/.](https://www.volvopenta.com/marine/accessories/joystick-control/)”
- [16] “[https://www.mtu-solutions.com/eu/en/automation-solutions/ship-automation-systems/mtu-nautiq-copilot.html.](https://www.mtu-solutions.com/eu/en/automation-solutions/ship-automation-systems/mtu-nautiq-copilot.html)”

- [17] "<https://www.xentas.com/xenta-at-cannes-and-genoa-boat-shows-with-two-innovations-x-aid-and-x-self-the-future-of-boating-is-now/>."
- [18] "<https://norwegianscitechnews.com/2022/09/ntnu-trials-worlds-first-urban-autonomous-passenger-ferry/>."
- [19] W. Zeng and R. L. Church, "Finding shortest paths on real road networks: the case for A*," 2007.
- [20] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment," *IEEE Access*, vol. 9, pp. 59196–59210, 2021, doi: 10.1109/ACCESS.2021.3070054.
- [21] "<https://it.mathworks.com/matlabcentral/fileexchange/26248-a-a-star-search-for-path-planning-tutorial>."
- [22] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. Wiley, 2011.
- [23] M. Martelli, N. Faggioni, and S. Donnarumma, "A time-domain methodology to assess the dynamic positioning performances," *Ocean Engineering*, vol. 247, Mar. 2022, doi: 10.1016/j.oceaneng.2022.110668.
- [24] A. Alessandri *et al.*, "Dynamic positioning system of a vessel with conventional propulsion configuration: Modeling and simulation," in *Maritime Technology and Engineering - Proceedings of MARTECH 2014: 2nd International Conference on Maritime Technology and Engineering*, 2015, pp. 725–734. doi: 10.1201/b17494-97.
- [25] C. W. Warren, "Fast path planning using modified a method," in *Proceedings - IEEE International Conference on Robotics and Automation*, 1993, vol. 2, pp. 662–667. doi: 10.1109/robot.1993.291883.

Desidero esprimere la mia sincera gratitudine a tutte le persone che mi hanno supportato durante il mio percorso accademico e nella stesura di questa tesi.

In primo luogo, ringrazio il professor Martelli e l'ingegner Fruzzetti, miei relatori, per avermi guidato con, pazienza e professionalità in questo lungo percorso. Grazie alla loro competenza e alla loro dedizione, ho potuto approfondire la mia formazione e sviluppare le mie capacità.

Un pensiero speciale va alla mia famiglia, che mi ha sempre sostenuto con affetto e comprensione, supportandomi in ogni scelta e facendomi sentire il loro amore incondizionato. Ringrazio di cuore mio padre, la nonna, Angelo e zio che sono stati e restano il mio punto di riferimento più prezioso.

Grazie Nicola, perché alla fine non stacchi mai alle 18:00. Grazie Sole, che sei riuscita a far uscire il lato bello di ciò che ho fatto.

Grazie Cri, per la tua preziosa RAM e grazie Davide perché quando c'era da lavorare alla tesi eri sempre disposto a portarmi a pesca.

Grazie a tutti i miei amici, che durante questi anni, hanno reso La Spezia un posto meno piovoso.

Portami a ballare

