



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

Implementazione di sistemi per auto-localizzazione e map-building di auto a guida autonoma

**Implementation of self-localization and map-building systems of
autonomous driving cars**

Candidato:
Andrea Faccenda

Relatore:
Andrea Bonci

Anno Accademico 2025-2026



UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

Implementazione di sistemi per auto-localizzazione e map-building di auto a guida autonoma

**Implementation of self-localization and map-building systems of
autonomous driving cars**

Candidato:
Andrea Faccenda

Relatore:
Andrea Bonci

Anno Accademico 2025-2026

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
CORSO DI LAUREA IN INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE
Via Brezze Bianche – 60131 Ancona (AN), Italy

Ringraziamenti

Giunto al termine di questo percorso, guardando indietro, mi rendo conto che il traguardo della laurea magistrale non è solo il frutto del mio impegno, ma il risultato di un viaggio condiviso con le persone che mi sono state vicine. A tutti loro va la mia più profonda gratitudine.

Un ringraziamento speciale va ai miei genitori e alla mia famiglia. Siete stati il mio punto di riferimento, la certezza su cui ho potuto contare in ogni momento. Grazie per aver creduto in me, per i vostri sacrifici, per avermi spronato a non mollare nei momenti di stanchezza e per aver festeggiato con me ogni successo. Se oggi sono qui, lo devo immensamente a voi e al vostro sostegno incrollabile.

Un posto d'onore spetta a Giulia. Grazie per essere stata la mia ancora, la mia complice e una fonte inesauribile di energia. In questo percorso non mi hai mai permesso di darmi per vinto; hai saputo ascoltare i miei dubbi, sopportare i momenti di stress e trasformare ogni difficoltà in uno stimolo per andare avanti. La tua presenza costante, la tua pazienza e il tuo esserci sempre, nei fatti, sono stati il motore più potente di questi mesi. Questo traguardo ti appartiene profondamente. Ai miei amici, sia a quelli di sempre che a quelli incontrati all'università, va un grazie immenso. Grazie per aver alleggerito il peso dello studio con le risate, i caffè, i confronti e le serate distensive. Siete stati la boccata d'aria fresca necessaria a non perdere la testa e a ricordarmi l'importanza di goderci il viaggio. Il vostro tifo e la vostra presenza hanno reso questi anni indimenticabili.

Infine, desidero ringraziare i parenti e tutte le persone che, anche solo con una parola di incoraggiamento o un pensiero affettuoso, mi hanno dimostrato vicinanza e stima durante questo cammino.

Ancona, Luglio 2026

Andrea Faccenda

Abstract

This work has been carried out within the framework of the Bosch Future Mobility Challenge (BFMC), an international competition that involves university students in the development of autonomous driving and intelligent mobility solutions. In this context, the thesis addresses the problem of vehicle localization and environment mapping, which are essential aspects for the proper operation of advanced autonomous driving functionalities.

The main objective is to define and implement a methodology that, starting from sensor fusion based on nonlinear Gaussian filters, improves odometric estimation through noise reduction techniques. On this basis, the global localization phase has been developed by means of simultaneous localization and mapping (SLAM) algorithms for the construction of the environmental map. The generated map is subsequently employed by the adaptive Monte Carlo localization (AMCL) package in order to enhance the accuracy and robustness of the vehicle pose estimation.

Experimental results have shown an average localization error below 0.10 m and improved system stability under dynamic conditions, confirming the effectiveness of the proposed approach. The main contribution of this work lies in the design and validation of a complete and modular localization architecture, fully integrated within the ROS 2 environment, capable of coherently combining sensor fusion, SLAM, and AMCL into a single operational framework.

Sommario

Il presente elaborato si colloca nell'ambito della Bosch Future Mobility Challenge (BFMC), competizione internazionale che coinvolge studenti universitari nello sviluppo di soluzioni per la guida autonoma e la mobilità intelligente. In tale contesto, il lavoro affronta il problema della localizzazione del veicolo e della mappatura dell'ambiente di test, aspetti fondamentali per il corretto funzionamento delle funzionalità avanzate di guida autonoma.

L'obiettivo è definire e implementare una metodologia che, a partire dalla fusione sensoriale basata su filtri gaussiani non lineari, migliori la stima odometrica mediante tecniche di riduzione del rumore. Su tale base viene sviluppata la fase di localizzazione globale, realizzata attraverso algoritmi di simultaneous localization and mapping (SLAM) per la costruzione della mappa dell'ambiente. La mappa generata è successivamente impiegata dal pacchetto adaptive Monte Carlo localization (AMCL), con lo scopo di incrementare la precisione e la robustezza della stima di posa del veicolo.

I risultati sperimentali hanno evidenziato una riduzione dell'errore medio di localizzazione al di sotto di 0.10 m e una maggiore stabilità del sistema in condizioni dinamiche, confermando l'efficacia dell'approccio proposto. Il contributo principale del lavoro consiste nella progettazione e validazione di un'architettura di localizzazione completa, modulare e pienamente integrata in ambiente ROS 2, capace di combinare in modo coerente fusione sensoriale, SLAM e AMCL in un unico framework operativo.

Indice

I. Introduzione	1
1. Introduzione	2
1.1. Bosch Future Mobility Challenge (BFMC)	2
1.2. Obiettivo della tesi	3
1.3. Struttura del documento	4
II. Hardware e architettura del sistema	5
2. Hardware	6
2.1. Chassis	8
2.2. Motore	8
2.3. Servomotore	9
2.4. IMU	10
2.5. STM32 Nucleo-G491RE	11
2.6. Jetson Orin NX	12
2.7. Telecamera	13
2.8. LiDAR	14
2.9. Sistema di localizzazione UWB	15
2.10. Schema delle connessioni elettriche	16
3. Architettura del sistema	19
3.1. Overview dell'architettura	19
3.2. Architettura modulare e pipeline di elaborazione	19
III. Fondamenti teorici e metodologie di localizzazione autonoma	22
4. Localizzazione locale	23
4.1. Filtri gaussiani	24
4.1.1. Filtro di Kalman (KF)	25
4.1.2. Filtro di Kalman esteso (EKF)	28
4.1.3. Filtro di Kalman unscented (UKF)	30
4.2. Modello cinematico del veicolo	32
4.2.1. Sistemi di riferimento	33
4.2.2. Modello a bicicletta	34

Indice

4.2.3.	Equazioni cinematiche	34
4.3.	Modellazione dei sensori	35
4.3.1.	Modellazione matematica dell'IMU BNO055	36
4.3.2.	Modellazione matematica dell'encoder ruota	37
4.3.3.	Modellazione matematica del sensore UWB DWM1001	37
4.3.4.	Validazione statistica delle misure	38
4.4.	Filtri bayesiani per la stima dello stato	39
4.4.1.	Stato, ingressi e modello di transizione	39
4.4.2.	EKF: jacobiano del modello	40
4.4.3.	UKF: propagazione unscented	40
4.4.4.	Modello di osservazione e aggiornamento	40
4.4.5.	Accorgimenti numerici e strategie di filtraggio	40
5.	Localizzazione globale	43
5.1.	Localizzazione globale	43
5.2.	Percezione e rappresentazione spaziale	44
5.2.1.	Mappatura a griglia di occupazione	45
5.3.	Simultaneous Localization and Mapping (SLAM)	47
5.3.1.	GraphSLAM	48
5.4.	Adaptive Monte Carlo Localization (AMCL)	53
5.4.1.	Formalismo probabilistico	54
5.4.2.	Filtro a campioni	54
5.4.3.	Adattività del numero di particelle	55
5.4.4.	Modelli probabilistici	55
5.4.5.	Algoritmo	56
5.5.	Pipeline globale	57
IV.	Implementazione	58
6.	Architettura ROS 2 Localizzazione	59
6.1.	Middleware ROS 2	60
6.1.1.	Distribuzione utilizzata	61
6.1.2.	Architettura ROS 2 Localizzazione	61
6.2.	Modello URDF	63
6.2.1.	Modello URDF/Xacro del veicolo	64
6.3.	Architettura di comunicazione ROS 2 per la localizzazione	66
6.3.1.	Sorgenti dati e interfacce di comunicazione	67
6.3.2.	Sistema di posizionamento UWB e comunicazione V2X	68
6.3.3.	Fusione e propagazione delle stime	69
6.3.4.	Gestione delle trasformazioni e sincronizzazione temporale	70

Indice

7. Implementazione del sistema di fusione sensoriale	72
7.1. Analisi dei sensori e determinazione delle covarianze	73
7.1.1. Encoder	73
7.1.2. IMU BNO055	76
7.1.3. Modulo di localizzazione UWB DWM1001	78
7.1.4. Sintesi delle covarianze di misura	81
7.2. Implementazione software del filtro di fusione	81
7.3. Validazione sperimentale del sistema di localizzazione	82
7.3.1. Validazione nel circuito dell'Ateneo UNIVPM	83
7.3.2. Validazione presso il Bosch Engineering Center	86
7.3.3. Analisi di consistenza del filtro	90
7.3.4. Analisi complessiva	91
8. Implementazione del sistema di localizzazione globale	93
8.1. Implementazione dello SLAM	93
8.1.1. Parametrizzazione e tuning	93
8.1.2. Risultati operativi	95
8.2. Implementazione dell'AMCL	98
8.2.1. Configurazione e tuning dei parametri	98
8.2.2. Risultati sperimentali	102
8.2.3. Comportamento in condizioni di guasto sensoriale	106
8.2.4. Risultati complessivi e considerazioni finali	107
V. Conclusione	109
9. Conclusione	110

Elenco delle figure

2.1. Configurazione hardware finale del veicolo.	7
2.2. Chassis Reely TC-04 Onroad 1:10.	8
2.3. Motore DC 36GP-555.	8
2.4. Servomotore Reely RS610WP.	9
2.5. Sensore IMU Bosch BNO055.	10
2.6. Scheda STM32 Nucleo-G491RE.	11
2.7. Scheda NVIDIA Jetson Orin NX da 8 GB.	12
2.8. Raspberry Pi Camera Module 3, versione grandangolare.	13
2.9. Sensore LiDAR LD19.	15
2.10. Modulo QORVO DWM1001-DEV.	16
2.11. Schema delle connessioni elettriche e relativa legenda.	18
3.1. Schema dell'architettura modulare del sistema.	21
4.1. Schema funzionale del processo di localizzazione locale.	24
4.2. Schema qualitativo del funzionamento del filtro di Kalman.	27
4.3. Rappresentazione dei sistemi di riferimento	33
4.4. Schema cinematico Ackermann	34
4.5. Schema del sistema di fusione sensoriale.	42
5.1. Modello grafico per la mappatura con pose note.	45
5.2. Esempio di risultato del processo di GraphSLAM.	53
5.3. Pipeline globale di localizzazione.	57
6.1. Flusso dei dati dal mondo reale all'attuazione.	59
6.2. Albero dei frame $\mathbf{tf}$ con le principali trasformazioni dinamiche e statiche.	64
6.3. Modello URDF del veicolo visualizzato in RViz.	66
6.4. Pipeline di comunicazione tra i nodi del sistema ROS 2 di localizzazione.	67
7.1. Segnale grezzo dell'encoder velocità angolare ω_m	74
7.2. Confronto tra misura grezza e stima filtrata della velocità angolare ω_m	74
7.3. Errore residuo tra misura e stima filtrata della velocità angolare ω_m	75
7.4. Segnale grezzo della velocità angolare $\dot{\psi}_m$	76
7.5. Deviazione standard della velocità angolare $\dot{\psi}_m$	76
7.6. Distribuzione dei residui della velocità angolare $\dot{\psi}_m$ con densità gaussiana stimata.	77
7.7. Andamento temporale delle coordinate misurate dal modulo UWB.	79

Elenco delle figure

7.8. Distribuzione dei campioni UWB lungo le due coordinate spaziali considerate.	79
7.9. Dispersione spaziale dei campioni UWB nel piano xy rispetto alla posizione teorica.	79
7.10. Schema planimetrico del circuito realizzato presso l'Ateneo UNIVPM.	84
7.11. Confronto tra la traiettoria di riferimento e le stime EKF e UKF. . .	84
7.12. Andamento temporale dell'errore di localizzazione per EKF e UKF.	85
7.13. Distribuzione cumulativa dell'errore per odometria grezza, EKF e UKF.	85
7.14. Distribuzione di frequenza dell'errore di localizzazione per odometria grezza, EKF e UKF.	86
7.15. Schema planimetrico del tracciato presso il Bosch Engineering Center di Cluj-Napoca.	87
7.16. Confronto tra la traiettoria di riferimento e le stime EKF e UKF. . .	88
7.17. Andamento temporale dell'errore di localizzazione per EKF e UKF.	88
7.18. Distribuzione cumulativa dell'errore per odometria grezza, EKF e UKF.	89
7.19. Distribuzione di frequenza dell'errore per odometria grezza, EKF e UKF.	89
8.1. Mappa di occupazione generata nel circuito dell'Ateneo con il grafo delle pose e i vincoli probabilistici.	96
8.2. Mappa di occupazione generata presso il Bosch Engineering Center, con evidenza delle pose e dei vincoli probabilistici.	97
8.3. Confronto tra traiettoria stimata da AMCL e traiettoria di riferimento.	103
8.4. Andamento temporale dell'errore di localizzazione dell'AMCL. . . .	103
8.5. Funzione di distribuzione cumulativa dell'errore di localizzazione dell'AMCL.	104
8.6. Confronto tra traiettoria stimata da AMCL e traiettoria di riferimento.	105
8.7. Errore di localizzazione stimato dall'AMCL.	105
8.8. Funzione di distribuzione cumulativa dell'errore di localizzazione dell'AMCL.	106

Elenco delle tabelle

2.1. Kit hardware fornito da Bosch.	6
2.2. Confronto tra Jetson Orin NX e Raspberry Pi 5.	13
4.1. Confronto tra filtro di Kalman e filtro di Kalman esteso.	30
4.2. Valori soglia del test χ^2 per diversi numeri di misure.	39
6.1. Moduli software dedicati alla localizzazione.	62
6.2. Parametri geometrici e di massa del modello URDF/Xacro.	65
6.3. Trasformazioni statiche dei sensori rispetto a base_link (giunti fissi).	66
7.1. Metriche di valutazione del filtro applicato ai dati dell'encoder.	75
7.2. Risultati dell'analisi statistica del giroscopio.	77
7.3. Statistiche delle coordinate UWB in condizioni statiche LOS.	80
7.4. Covarianze di misura dei sensori impiegate nel filtro di fusione.	81
7.5. Metriche di accuratezza della localizzazione nel circuito UNIVPM.	86
7.6. Metriche di accuratezza della localizzazione nel tracciato Bosch Engineering Center.	90
7.7. Indicatori di consistenza dell'EKF sul recording finale ($N \approx 2600$ aggiornamenti UWB, ~ 280 s).	91
8.1. Parametri principali di configurazione del nodo SLAM Toolbox.	94
8.2. Parametri principali di configurazione del nodo AMCL.	100
8.3. Metriche di accuratezza della localizzazione nel circuito dell'Ateneo.	107
8.4. Metriche di accuratezza della localizzazione presso il Bosch Engineering Center di Cluj-Napoca.	107

Parte I.

Introduzione

Capitolo 1.

Introduzione

Negli ultimi anni, la mobilità autonoma ha assunto un ruolo centrale nella ricerca ingegneristica, costituendo uno dei settori più rilevanti per l'innovazione tecnologica e industriale. La presente tesi si colloca in tale ambito, con particolare riferimento alla Bosch Future Mobility Challenge (BFMC), competizione internazionale dedicata allo sviluppo di veicoli autonomi in scala. L'elaborato analizza in modo specifico gli aspetti relativi alla localizzazione e alla mappatura dell'ambiente, elementi essenziali per garantire l'affidabilità e la precisione dei sistemi di guida autonoma.

1.1. Bosch Future Mobility Challenge (BFMC)

La BFMC è una competizione internazionale organizzata annualmente dal Bosch Engineering Center di Cluj, in Romania, a partire dal 2017. L'iniziativa coinvolge team universitari provenienti da diversi Paesi, impegnati nello sviluppo di algoritmi e architetture software per la guida autonoma e la connettività veicolare.

La competizione utilizza veicoli in scala 1:10 forniti da Bosch, progettati per muoversi in un ambiente che riproduce una città intelligente in miniatura, cioè un contesto urbano dotato di infrastrutture digitali e sensori interconnessi. L'obiettivo è rendere il veicolo capace di affrontare in modo completamente autonomo scenari realistici di mobilità, integrando capacità di percezione, pianificazione e controllo. Ogni edizione introduce nuove sfide tecniche, con l'intento di aumentare progressivamente la complessità delle prove e favorire l'adozione di soluzioni sempre più avanzate.

Durante la fase di preparazione, i team collaborano con ingegneri Bosch e docenti universitari, beneficiando di attività di mentoring e supporto tecnico. Il percorso di sviluppo è articolato in tappe intermedie caratterizzate dalla consegna di rapporti di avanzamento e da verifiche periodiche, finalizzate al monitoraggio dello stato dei progetti e alla pianificazione strutturata delle attività.

Le principali funzionalità richieste ai veicoli comprendono:

- controllo della velocità e mantenimento della traiettoria;
- attraversamento autonomo di incroci e rotatorie;
- riconoscimento della segnaletica stradale e dei semafori;

- rispetto della distanza di sicurezza e gestione delle interazioni con pedoni e altri veicoli;
- esecuzione di manovre di parcheggio e sorpasso in condizioni controllate;
- adattamento della guida a differenti condizioni ambientali e di illuminazione.

L'integrazione simultanea di tali funzionalità costituisce la principale sfida della competizione, poiché anche un singolo errore può compromettere l'affidabilità complessiva del sistema.

La BFMC riveste inoltre un rilevante valore formativo e applicativo. Da un lato, offre agli studenti un contesto sperimentale in cui applicare conoscenze teoriche e sviluppare competenze nella gestione di progetti complessi; dall'altro, rappresenta per Bosch un'opportunità di individuazione di giovani talenti e di sperimentazione di tecnologie emergenti nel campo della mobilità intelligente.

Il progetto descritto in questa tesi si inserisce in tale contesto, essendo stato sviluppato nell'ambito della partecipazione diretta alla competizione.

1.2. Obiettivo della tesi

L'obiettivo principale del presente lavoro consiste nello studio e nello sviluppo di un sistema di localizzazione e mappatura per un veicolo autonomo in scala, realizzato nell'ambito della partecipazione del team universitario alla BFMC. L'attenzione è rivolta alla progettazione di un'architettura hardware e software in grado di stimare con precisione la posizione del veicolo e di ricostruire l'ambiente circostante, garantendo robustezza e affidabilità in condizioni operative variabili.

Il progetto è articolato in più fasi, comprendenti l'analisi dei sensori e dei flussi informativi, la realizzazione e la validazione del sistema di localizzazione. Particolare attenzione è dedicata alla fusione dei dati provenienti da sensori eterogenei e all'integrazione coerente delle informazioni spaziali, al fine di ottenere una rappresentazione accurata dell'ambiente e migliorare la capacità decisionale del veicolo.

Una localizzazione precisa costituisce la base di tutte le funzionalità di guida autonoma, quali la pianificazione del percorso, l'evitamento degli ostacoli e l'esecuzione di manovre complesse. Una stima della posizione non sufficientemente robusta può compromettere l'intero processo decisionale, rendendo inefficaci anche algoritmi di controllo e pianificazione avanzati.

Il lavoro non si propone di introdurre nuove metodologie di localizzazione, ma di analizzare e adattare tecniche consolidate presenti in letteratura al contesto specifico di un veicolo autonomo in scala, valutandone le prestazioni e i limiti in un ambiente sperimentale realistico.

I risultati ottenuti vengono infine discussi in modo critico, evidenziando i principali punti di forza del sistema sviluppato e delineando possibili direzioni di miglioramento per sviluppi futuri.

1.3. Struttura del documento

L'elaborato è organizzato come segue:

- Il Capitolo 2 descrive l'architettura hardware del veicolo autonomo, illustrandone i componenti principali, i sensori di bordo e le caratteristiche meccaniche e di alimentazione;
- Il Capitolo 3 presenta l'architettura del sistema software, delineando i moduli principali e la loro integrazione funzionale nel sistema di controllo del veicolo;
- Il Capitolo 4 introduce i fondamenti teorici relativi alla localizzazione odometrica e alla fusione sensoriale, descrivendo i modelli cinematici e i filtri probabilistici utilizzati per la stima incrementale della posizione;
- Il Capitolo 5 approfondisce i principi della localizzazione globale e del Simultaneous Localization and Mapping (SLAM), illustrando gli approcci basati su mappa e i metodi di ottimizzazione associati;
- Il Capitolo 6 descrive l'architettura complessiva del sistema sviluppato in ambiente ROS 2, con particolare attenzione ai protocolli di comunicazione tra i nodi, alla struttura del grafo dei processi e alla gestione dei flussi di dati tra i moduli di percezione, stima e controllo;
- Il Capitolo 7 riporta i risultati implementativi della localizzazione relativa basata sull'odometria e sulla fusione dei sensori, analizzando il comportamento del sistema in diverse condizioni operative e confrontando le prestazioni dei diversi filtri di stima;
- Il Capitolo 8 presenta i risultati implementativi della mappatura e della localizzazione globale mediante SLAM, includendo le prove sperimentali condotte in ambienti reali e la valutazione quantitativa dell'accuratezza complessiva del sistema;
- Il Capitolo 9 conclude il lavoro, sintetizzando i risultati ottenuti e delineando le prospettive di miglioramento e sviluppo futuro.

Parte II.

Hardware e architettura del sistema

Capitolo 2.

Hardware

In questo capitolo vengono descritte le componenti hardware che costituiscono la base del sistema impiegato nella competizione. Dopo una panoramica del kit originario fornito dall'azienda, vengono illustrate le modifiche introdotte per migliorarne le prestazioni e adattarlo ai requisiti specifici delle diverse prove.

La piattaforma iniziale, fornita da Bosch, rappresentava una base solida, ma presentava alcune limitazioni in termini di capacità di calcolo, precisione e percezione. Essa comprendeva i componenti riportati in Tabella 2.1.

Tabella 2.1.: Kit hardware fornito da Bosch.

Funzione	Componente
Controllo	STM32 Nucleo F401RE
Unità di calcolo	Raspberry Pi 5 (8 GB RAM)
Motorizzazione	Motore brushless Quickrun Fusion SE 1800KV
Sterzo	Servomotore dedicato
Sensore inerziale	BNO055 (IMU)
Alimentazione	Batteria standard fornita a corredo
Sistema visivo	Raspberry Pi Camera Module v2.1

Per superare le limitazioni del sistema di base e garantire prestazioni adeguate alle prove più impegnative, il veicolo è stato oggetto di una serie di aggiornamenti mirati. Il motore brushless originario è stato sostituito con un motore a corrente continua da 36 mm, alimentabile a 12/24 V e dotato di encoder, scelta che consente un controllo più accurato della velocità e della posizione, migliorando la stabilità della locomozione. Contestualmente, la scheda di controllo STM32 Nucleo F401RE è stata sostituita con la versione Nucleo STM32G491RE, caratterizzata da una maggiore capacità di elaborazione e affidabilità nella gestione delle logiche di controllo a basso livello.

L'unità di calcolo è stata aggiornata con l'integrazione di una Jetson Orin NX da 8 GB, capace di eseguire in tempo reale algoritmi di visione artificiale e intelligenza artificiale ad elevato carico computazionale.

Capitolo 2. Hardware

Il sistema di percezione è stato oggetto di una revisione sostanziale. La Raspberry Pi Camera Module v2.1 è stata sostituita da una Raspberry Pi Camera Module 3 con ottica grandangolare, al fine di ottenere un campo visivo più ampio, requisito essenziale per gli algoritmi di mantenimento della traiettoria. È stato inoltre aggiunto un sensore di luminosità Adafruit VEML7700, impiegato per rilevare condizioni di scarsa illuminazione e gestire automaticamente l'accensione dei fari, incrementando l'adattabilità del veicolo a diverse condizioni ambientali.

Per la percezione tridimensionale dell'ambiente circostante, è stato integrato un LiDAR LD19 in sostituzione dei sonar a basso costo, ottenendo una rappresentazione più accurata e affidabile dello spazio. Tale scelta ha apportato benefici diretti in termini di localizzazione, pianificazione e gestione degli ostacoli.

Il sistema di alimentazione è stato adeguato mediante la sostituzione della batteria originaria con una batteria agli ioni di litio 4S da 3500 mAh, in grado di fornire la potenza necessaria al corretto funzionamento della Jetson Orin NX, caratterizzata da un assorbimento energetico superiore rispetto alla Raspberry Pi.

L'insieme delle modifiche ha permesso di ottenere una configurazione finale caratterizzata da maggiore precisione nel controllo, superiori risorse di calcolo e una percezione ambientale più completa e robusta.

La configurazione risultante è mostrata in Figura 2.1.

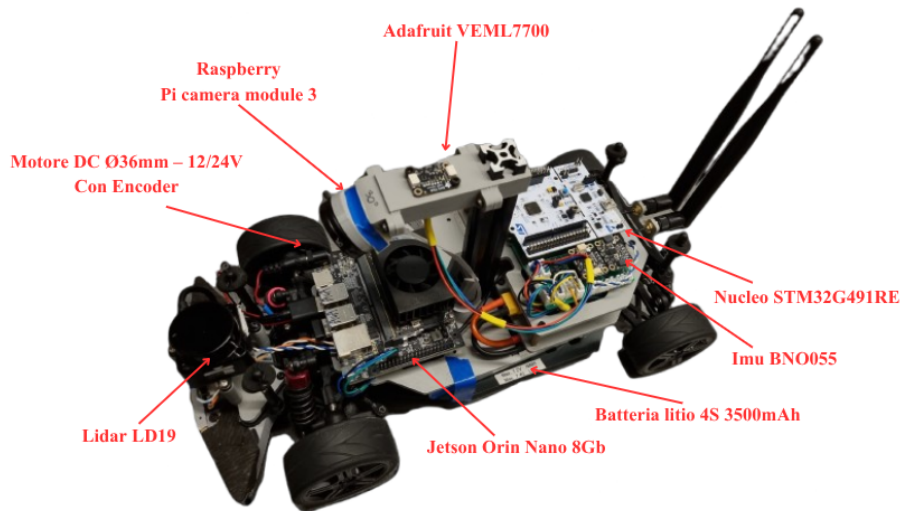


Figura 2.1.: Configurazione hardware finale del veicolo.

2.1. Chassis

Il veicolo impiega come base meccanica il modello Reely TC-04 Onroad Chassis 1:10, Electric Road Version 4WD ARR (Figura 2.2). Si tratta di uno chassis per automodelli elettrici in scala 1:10, dotato di telaio in alluminio che garantisce elevata rigidità e stabilità strutturale.



Figura 2.2.: Chassis Reely TC-04 Onroad 1:10.

Il sistema integra sospensioni regolabili e trazione integrale con differenziali anteriore e posteriore, caratteristiche che assicurano un'adeguata aderenza anche a velocità elevate. Lo chassis funge inoltre da supporto strutturale per l'installazione delle componenti elettroniche ed elettromeccaniche del prototipo, tra cui pacco batterie, schede di controllo, motori e sensori, costituendo la base meccanica per l'integrazione dell'intero sistema hardware. Le specifiche tecniche del telaio sono state ricavate dal datasheet [15] rilasciato dalla casa madre.

2.2. Motore

Il motore impiegato per la trazione del veicolo è un modello DC 36GP-555 dotato di riduttore planetario e di encoder incrementale integrato.



Figura 2.3.: Motore DC 36GP-555.

La sostituzione dell'unità brushless originaria con un motore a corrente continua è stata motivata dall'esigenza di ottenere un controllo più preciso della velocità e più stabile della posizione angolare, requisito fondamentale per la localizzazione autonoma e per l'esecuzione di manovre automatiche in spazi ristretti.

Il riduttore interamente metallico assicura una trasmissione meccanica rigida e priva di giochi significativi, migliorando la risposta dinamica e la ripetibilità del moto. L'adozione di un sistema di alimentazione a 12/24 V consente una gestione flessibile della coppia erogata, permettendo di adattare le prestazioni del veicolo alle diverse condizioni operative previste dalla competizione.

L'integrazione dell'encoder a quadratura direttamente nel corpo motore costituisce un vantaggio ingegneristico rilevante: la misura in tempo reale della velocità e della posizione relativa dell'albero motore consente l'implementazione di controlli in retroazione chiusa (PID) caratterizzati da elevata stabilità. Tale approccio riduce gli effetti di deriva tipici dei sistemi open-loop e consente di ottenere una locomozione più regolare e prevedibile, condizione necessaria per l'efficacia degli algoritmi di localizzazione basati su odometria.

Dal punto di vista dell'integrazione hardware, la scelta del modello 36GP-555 ha permesso di semplificare il cablaggio e ridurre gli ingombri rispetto a soluzioni con encoder esterni, incrementando l'affidabilità complessiva del sistema e la facilità di manutenzione. Nel complesso, il motore selezionato rappresenta un compromesso equilibrato tra precisione, compattezza e robustezza meccanica, risultando idoneo all'impiego su un veicolo autonomo in scala ad alte prestazioni. I dati tecnici e le curve caratteristiche di coppia e assorbimento sono stati consultati nel datasheet [16] fornito dal produttore.

2.3. Servomotore

Per la gestione del meccanismo di sterzo del veicolo è stato impiegato un servomotore modello Reely RS610WP, progettato per applicazioni che richiedono movimenti angolari precisi e controllo diretto della posizione.



Figura 2.4.: Servomotore Reely RS610WP.

Il servomotore fornisce una coppia di stallo pari a $7.08 \text{ kg} \cdot \text{cm}$ con alimentazione a 4.8 V, raggiungendo una velocità di circa 0.37 s per compiere un angolo di 60° . A

6.0 V la coppia di stallo aumenta fino a $8.5 \text{ kg} \cdot \text{cm}$, mentre la velocità si riduce a 0.3 s per 60° . L'incremento prestazionale ottenibile con una tensione superiore risulta tuttavia marginale rispetto al maggiore consumo energetico, che riduce l'autonomia complessiva del veicolo. Per tale motivo, il servomotore è stato alimentato a 4.8 V, configurazione che rappresenta un compromesso equilibrato tra prestazioni e durata operativa.

Il dispositivo impiega una tecnologia analogica e integra un potenziometro di retroazione, che consente di mantenere un movimento regolare e stabile. Il controllo è realizzato mediante un segnale PWM in ingresso, generato dagli algoritmi implementati sul microcontrollore, i quali determinano in tempo reale la posizione dell'albero di uscita.

Nel complesso, le caratteristiche del modello Reely RS610WP lo rendono idoneo all'applicazione considerata, assicurando reattività e affidabilità nella gestione del sistema di sterzo del veicolo. I dati tecnici presentati derivano dal datasheet [17] rilasciato dal costruttore.

2.4. IMU

L'unità di misura inerziale (IMU) costituisce un componente essenziale per la navigazione e il controllo del veicolo, in quanto fornisce le grandezze cinematiche fondamentali, quali accelerazioni, velocità angolari e orientamento nello spazio, necessarie per la stima dello stato dinamico e per la correzione dei modelli odometrici.

Nel presente progetto è stato impiegato il sensore Bosch BNO055, appartenente alla categoria delle IMU NDOF (Nine Degrees of Freedom). Il dispositivo misura nove gradi di libertà grazie all'integrazione di tre sottosistemi principali:

- accelerometro, che rileva le accelerazioni lineari lungo i tre assi del sensore;
- giroscopio, che fornisce la velocità angolare di rotazione rispetto agli assi;
- magnetometro, che misura il campo magnetico terrestre e consente la determinazione dell'orientamento assoluto.



Figura 2.5.: Sensore IMU Bosch BNO055.

L'impiego combinato di tali sensori consente di ottenere una rappresentazione ridondante e coerente dello stato dinamico del veicolo. I dati del giroscopio risultano accurati nel breve periodo ma soggetti a deriva, mentre quelli del magnetometro

garantiscono maggiore stabilità nel lungo termine; la loro fusione incrementa la robustezza complessiva della misura. Il sensore BNO055 integra un microcontrollore dedicato che elabora i segnali mediante algoritmi di fusione sensoriale basati su tecniche di filtraggio di Kalman, producendo grandezze derivate come angoli di Eulero e versioni filtrate delle misure grezze. Questa architettura riduce la complessità computazionale richiesta all'unità di calcolo principale, rendendo il dispositivo adatto a piattaforme embedded con risorse limitate.

Il collegamento con l'unità di controllo è stato realizzato tramite il protocollo I²C, scelto per la semplicità di implementazione e la piena compatibilità con la scheda STM32 Nucleo. Tra i principali vantaggi del sensore si evidenziano la calibrazione automatica all'accensione, la ridotta sensibilità al rumore e la stabilità numerica delle grandezze derivate. Tali caratteristiche garantiscono una stima affidabile dell'assetto e una maggiore consistenza nella localizzazione del veicolo, anche in presenza di vibrazioni o variazioni repentine della traiettoria. I dati tecnici e le caratteristiche funzionali sono stati ottenuti dal datasheet [18] messo a disposizione dal produttore.

2.5. STM32 Nucleo-G491RE

La scheda Nucleo-G491RE, prodotta da STMicroelectronics, appartiene alla serie STM32G4 e integra un microcontrollore Arm[®] Cortex[®] M4 a 32 bit con unità a virgola mobile (FPU), capace di raggiungere una frequenza operativa di 170 MHz. Il microcontrollore dispone di 512 KB di memoria Flash e 112 KB di RAM, risorse che consentono l'esecuzione di applicazioni complesse e la gestione in tempo reale di più periferiche. Rispetto alla precedente STM32 F401RE, la versione G491RE garantisce maggiore stabilità del segnale PWM, tempi di elaborazione ridotti e migliore efficienza energetica, risultando quindi più adatta alla gestione dei controlli a bassa latenza richiesti dal veicolo autonomo.

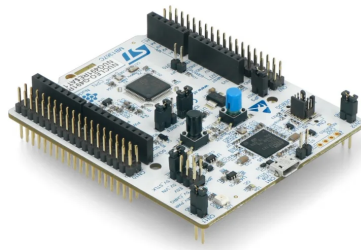


Figura 2.6.: Scheda STM32 Nucleo-G491RE.

La scheda offre diverse interfacce di comunicazione, tra cui GPIO, I²C, SPI, UART e CAN, che ne agevolano l'integrazione con sensori, attuatori e moduli esterni. In particolare, include:

- timer avanzati a 16 e 32 bit, utilizzati per la generazione di segnali PWM ad alta risoluzione;

- convertitori ADC a 12 bit ad alta velocità, necessari per la lettura accurata di segnali analogici;
- interfacce seriali multiple, che assicurano una comunicazione stabile con l'unità di calcolo principale;
- periferiche dedicate al controllo diretto di motori e segnali di potenza.

Dal punto di vista hardware, la Nucleo-G491RE include pulsanti e LED di stato programmabili, oltre al debugger e programmatore ST-LINK/V2-1 integrato, che consente la programmazione e il debug diretto tramite connessione micro USB, senza necessità di dispositivi esterni. La possibilità di alimentazione sia via USB sia da sorgenti esterne offre ulteriore flessibilità nelle fasi di test e integrazione.

Nel contesto del progetto, la scheda svolge la funzione di interfaccia tra i moduli di percezione e gli attuatori, gestendo il controllo in tempo reale del motore di trazione e del servosterzo. Coordina inoltre la ricezione dei setpoint generati dalla Jetson Orin NX tramite comunicazione seriale e l'acquisizione dei dati provenienti dai sensori di bordo, come l'IMU. La combinazione di elevate prestazioni e compatibilità con un'ampia gamma di periferiche ha reso la STM32 Nucleo-G491RE una piattaforma affidabile per l'implementazione dei controlli a basso livello richiesti dalla competizione. Le informazioni relative alle caratteristiche hardware e alle specifiche di comunicazione sono state ricavate dal datasheet [19] pubblicato dal produttore.

2.6. Jetson Orin NX

La Jetson Orin NX è un computer a scheda singola sviluppato da NVIDIA per applicazioni di intelligenza artificiale e robotica avanzata. La piattaforma integra un'architettura ad alte prestazioni che consente l'elaborazione in tempo reale di modelli di visione artificiale, apprendimento automatico e pianificazione del movimento. La versione utilizzata nel progetto è dotata di 8 GB di memoria LPDDR5 e raggiunge fino a 70 TOPS di potenza di calcolo, mantenendo un consumo energetico compreso tra 10 W e 20 W.



Figura 2.7.: Scheda NVIDIA Jetson Orin NX da 8 GB.

Rispetto alla Raspberry Pi 5 precedentemente impiegata, la Jetson Orin NX offre un incremento significativo della capacità di calcolo e della larghezza di banda della

memoria, risultando più adatta all'esecuzione di algoritmi di percezione e localizzazione in tempo reale. La Tabella 2.2 riassume i principali parametri comparativi tra le due piattaforme. Nel progetto, la scheda svolge la funzione di unità di calcolo principale, gestendo la percezione e la pianificazione del veicolo. Elabora in tempo reale le immagini provenienti dal sistema visivo per il mantenimento della traiettoria, esegue la localizzazione mediante fusione dei dati sensoriali e pianifica le traiettorie in presenza di ostacoli o intersezioni.

Tabella 2.2.: Confronto tra Jetson Orin NX e Raspberry Pi 5.

Caratteristica	Jetson Orin NX (8 GB)	Raspberry Pi 5 (8 GB)
CPU	6 core Arm Cortex-A78AE, 2.0 GHz	4 core Arm Cortex-A76, 2.4 GHz
GPU	NVIDIA Ampere (1024 CUDA cores)	VideoCore VII integrata
Prestazioni AI	Fino a 70 TOPS	Circa 1 TOPS
RAM	8 GB LPDDR5, 102.4 GB/s	8 GB LPDDR4X, 51.2 GB/s
Acceleratori dedicati	2 NVDLA v2 per AI	Nessuno
Supporto software	JetPack, CUDA, TensorRT, ROS 2	Linux standard

I comandi di controllo generati vengono trasmessi al microcontrollore STM32 per l'attuazione fisica del moto. L'elevato parallelismo di elaborazione e la disponibilità di librerie ottimizzate per l'intelligenza artificiale rendono la Jetson Orin NX una piattaforma affidabile per la gestione delle funzioni di guida autonoma del veicolo. Le caratteristiche tecniche sono state ricavate dal datasheet [20] rilasciato dal produttore.

2.7. Telecamera

Per la percezione visiva del veicolo è stato impiegato il modulo Raspberry Pi Camera Module 3, evoluzione della versione 2.1 e caratterizzato da prestazioni superiori in termini di risoluzione e sensibilità luminosa. Il dispositivo integra un sensore Sony IMX708 da 12 megapixel con formato ottico da 1/2.43 pollici, in grado di acquisire immagini ad alta definizione e operare in modo affidabile anche in condizioni di scarsa illuminazione.

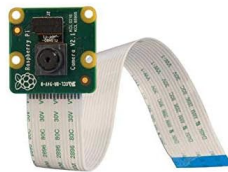


Figura 2.8.: Raspberry Pi Camera Module 3, versione grandangolare.

Una delle principali innovazioni rispetto alla versione precedente è l'introduzione dell'autofocus, che consente la regolazione automatica della messa a fuoco in funzione della distanza dell'oggetto ripreso, migliorando la nitidezza e la stabilità della visione durante il movimento del veicolo. Il modulo è disponibile in due varianti, con campo visivo standard o grandangolare, a seconda delle esigenze applicative.

Le principali caratteristiche tecniche del modulo sono sintetizzate di seguito:

- sensore: Sony IMX708, 12 megapixel;
- risoluzione massima: 4608×2592 pixel;
- acquisizione video: fino a 1080p a 120 fps oppure 1536p a 30 fps;
- ottica: autofocus integrato con apertura $f/1.8$;
- campo visivo: versione standard (66°) o wide (102°);
- interfaccia: connettore CSI compatibile con le schede Raspberry Pi;
- alimentazione e controllo: forniti direttamente dal connettore CSI, senza circuiti esterni aggiuntivi.

L'integrazione della telecamera nel sistema di bordo risponde all'esigenza di acquisire dati visivi accurati per le funzioni di percezione e interpretazione dell'ambiente. Le immagini raccolte costituiscono la base per gli algoritmi di rilevamento della corsia e di riconoscimento degli oggetti presenti nello scenario stradale, inclusi veicoli, pedoni e segnaletica. La qualità del sensore, unita alla possibilità di acquisire immagini ad alta frequenza e con ampio campo visivo, contribuisce a migliorare la robustezza dei sistemi di visione e a garantire un funzionamento stabile in contesti dinamici. Il Camera Module 3 rappresenta un componente chiave per le capacità percettive del veicolo autonomo, offrendo un compromesso efficace tra qualità d'immagine, compattezza e semplicità di integrazione hardware. Le specifiche tecniche sono state desunte dal datasheet [21] pubblicato dal produttore.

2.8. LiDAR

Nel sistema di percezione del veicolo è stato integrato un sensore LiDAR planare modello LD19, prodotto da LDROBOT, caratterizzato da un campo visivo di 360 gradi e da dimensioni compatte che ne agevolano l'integrazione in piattaforme embedded. Il dispositivo offre un equilibrio adeguato tra prestazioni, costo e ingombro, risultando idoneo per applicazioni di robotica mobile e navigazione autonoma.

Il LD19 effettua scansioni fino a una distanza massima di 12 m. L'errore di misura, dell'ordine di ± 2 cm, consente di ottenere rappresentazioni bidimensionali dell'ambiente precise e aggiornate in tempo reale, permettendo al veicolo di rilevare con

affidabilità ostacoli e strutture circostanti. La regolarità e la densità delle informazioni acquisite rendono il sensore adatto a compiti di percezione spaziale e mappatura, in cui è richiesta una descrizione accurata dello spazio operativo.



Figura 2.9.: Sensore LiDAR LD19.

La scelta del LD19 rispetto ai sensori a ultrasuoni impiegati in precedenza è stata motivata dall'esigenza di migliorare la continuità e la qualità delle misurazioni. I sensori a ultrasuoni, pur rappresentando una soluzione economica e semplice da implementare, risultano limitati in termini di risoluzione angolare, campo visivo e affidabilità in presenza di superfici inclinate o materiali con elevata capacità di assorbimento acustico. Il LiDAR, al contrario, fornisce una sequenza continua di punti spaziali che descrivono l'ambiente in modo uniforme, garantendo una percezione coerente e precisa dello spazio circostante.

Nel presente progetto il sensore è stato collegato alla piattaforma Jetson Orin NX tramite interfaccia USB e integrato nel sistema ROS 2 mediante i driver forniti dal produttore. I dati di scansione vengono pubblicati come nuvole di punti bidimensionali e utilizzati dai moduli di percezione per generare una rappresentazione geometrica dell'ambiente. In tale contesto, il LiDAR riveste un ruolo centrale nel sistema percettivo, fornendo le informazioni necessarie per individuare ostacoli, stimare distanze e supportare la pianificazione autonoma del movimento in condizioni operative variabili. Le informazioni di riferimento sono state ottenute dal datasheet [22] pubblicato dal produttore.

2.9. Sistema di localizzazione UWB

Con l'edizione 2024 della BFMC è stato introdotto un sistema di localizzazione in tempo reale (Real-Time Locating System, RTLS) basato su tecnologia a banda ultra larga (Ultra WideBand, UWB). L'hardware impiegato nel progetto è costituito dal kit di sviluppo QORVO MDEK1001, composto da dodici moduli DWM1001-DEV configurabili in due modalità operative:

- ancora: nodo installato in posizione fissa che funge da ricetrasmittitore e, insieme alle altre ancore, partecipa al calcolo della posizione dei tag mediante tecniche di multilaterazione;

- tag: nodo mobile installato sul veicolo, che comunica con le ancore per ricevere la propria posizione stimata.



Figura 2.10.: Modulo QORVO DWM1001-DEV.

Il principio di funzionamento si basa sulla misurazione dei tempi di propagazione dei segnali radio tra il tag e le ancore circostanti. Conoscendo la velocità di trasmissione del segnale e la posizione delle ancore, è possibile determinare la distanza del tag da ciascuna di esse e stimare la posizione assoluta del veicolo con elevata precisione. Il sistema fornisce aggiornamenti di posizione a una frequenza di 10 Hz, con un'accuratezza tipica inferiore a 10 cm in ambienti privi di interferenze significative. Il modulo UWB fornisce una misura assoluta di riferimento che consente di correggere le derive accumulate dai sensori inerziali durante il movimento, migliorando la precisione complessiva della localizzazione.

Questa funzione di correzione è fondamentale per garantire che il veicolo mantenga un tracciamento affidabile del proprio percorso e possa interagire con l'ambiente circostante in modo autonomo e sicuro. Le strategie di fusione dei dati e il loro impatto sulle prestazioni di localizzazione verranno approfonditi nei capitoli successivi. Le informazioni tecniche e le modalità di utilizzo sono state ricavate dal manuale utente [23] messo a disposizione dal produttore.

2.10. Schema delle connessioni elettriche

Dopo aver descritto i principali componenti hardware del sistema, in questa sezione viene analizzato il sistema di connessioni elettriche che integra e collega tra loro le diverse unità funzionali. L'obiettivo è fornire una visione complessiva della distribuzione della potenza, dei segnali di controllo e delle linee di comunicazione che costituiscono l'infrastruttura fisica del veicolo autonomo.

L'architettura elettrica è stata progettata per garantire un flusso stabile e affidabile sia per la distribuzione della potenza sia per lo scambio dei segnali digitali tra i moduli. Il sistema si articola su due livelli principali: un livello di potenza, dedicato all'alimentazione dei motori e dei componenti ad alto assorbimento, e un livello logico, destinato alla gestione dei segnali di controllo, delle comunicazioni seriali e dell'alimentazione a bassa corrente dei circuiti di calcolo e sensorizzazione.

La configurazione complessiva prevede il collegamento della piattaforma Jetson Orin NX con il microcontrollore STM32 Nucleo-G491RE, al quale fanno capo i sensori e gli attuatori di bordo. La Jetson Orin NX costituisce il nodo centrale di elaborazione, eseguendo gli algoritmi di visione, localizzazione e pianificazione, mentre la Nucleo svolge la funzione di unità di controllo a basso livello, responsabile della gestione diretta del motore e del servosterzo.

Il sistema di alimentazione si basa su due batterie indipendenti agli ioni di litio da 3500 mAh, collegate a una scheda di distribuzione della potenza. Quest'ultima gestisce la ripartizione delle tensioni operative verso le varie sezioni del veicolo, separando le linee di potenza destinate ai motori da quelle di alimentazione logica. Le connessioni di potenza, rappresentate in rosso nello schema, alimentano i componenti ad alto assorbimento, mentre le linee di massa (GND), in nero, forniscono un riferimento comune e riducono il rischio di disturbi dovuti a differenze di potenziale. Le linee di comunicazione sono organizzate in due categorie principali: le linee standard, indicate in verde, destinate allo scambio di dati tra Jetson Orin NX e camera tramite interfacce come USB o MIPI; e le linee dedicate, in giallo, riservate ai collegamenti tra microcontrollore, sensori e attuatori mediante protocolli I²C e UART. Le connessioni fisiche, in azzurro, comprendono i collegamenti diretti che trasmettono i segnali di comando e di retroazione necessari al corretto funzionamento del sistema di controllo. La comunicazione tra Jetson Orin NX e Nucleo avviene tramite collegamento seriale a frequenza costante, con meccanismi di verifica dell'integrità dei pacchetti per assicurare coerenza e affidabilità dei dati.

Dal punto di vista ingegneristico, la disposizione dei collegamenti è stata progettata per minimizzare le interferenze elettromagnetiche mediante la separazione fisica tra linee di potenza e linee di segnale, ottimizzare la gestione termica e lo spazio disponibile sullo chassis, e garantire modularità e semplicità di manutenzione. L'organizzazione ordinata del cablaggio contribuisce a ridurre la suscettibilità del sistema ai disturbi elettrici, assicurando stabilità anche durante il funzionamento a carichi elevati.

In Figura 2.11 è riportato lo schema generale delle connessioni elettriche, corredato dalla relativa legenda che consente di identificare con chiarezza le diverse tipologie di collegamento e la funzione di ciascun elemento all'interno del sistema.

La configurazione elettrica così definita costituisce la base fisica su cui si innesta l'architettura software e di controllo presentata nel capitolo successivo, garantendo coerenza tra la struttura hardware e la logica di funzionamento complessiva del veicolo autonomo.

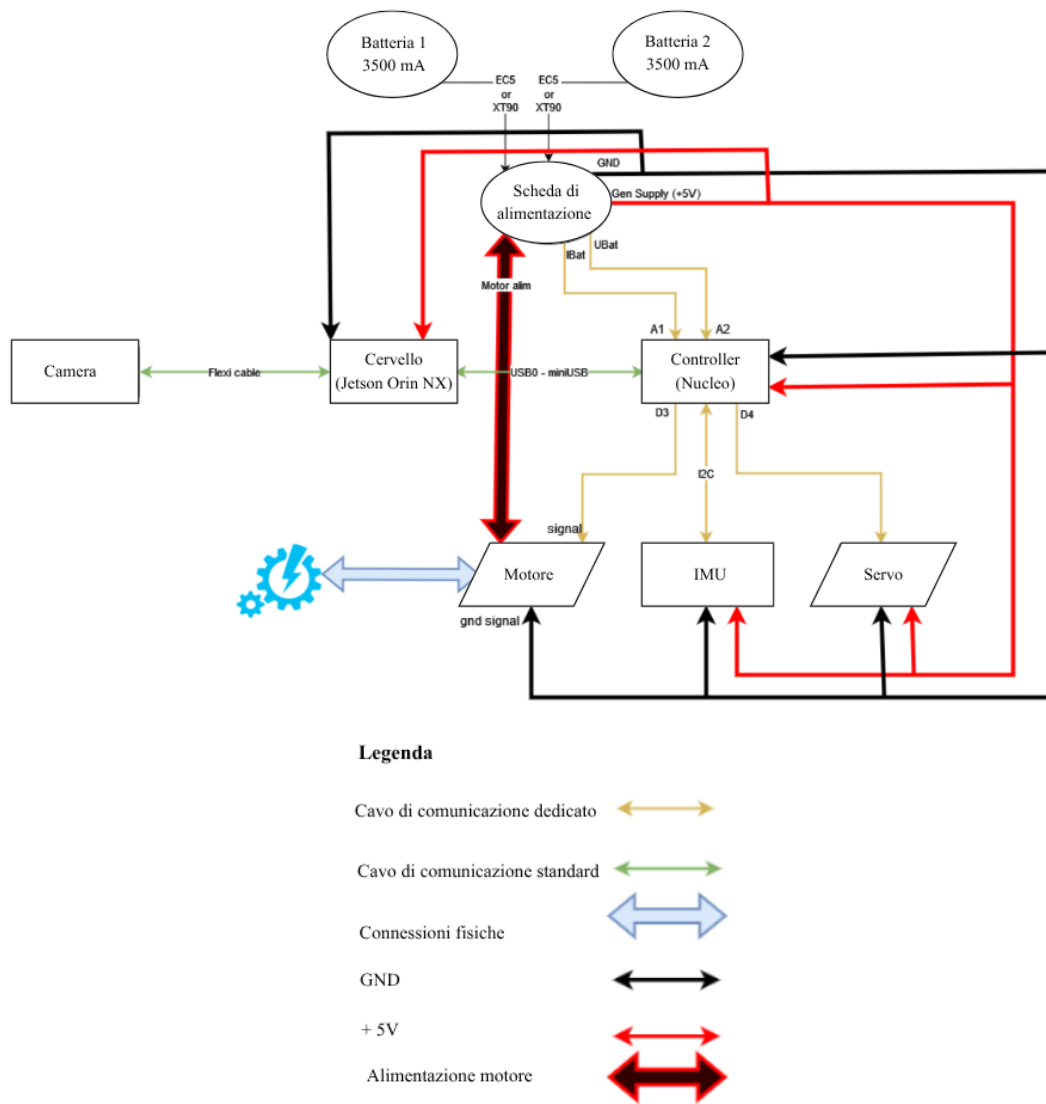


Figura 2.11.: Schema delle connessioni elettriche e relativa legenda.

Capitolo 3.

Architettura del sistema

3.1. Overview dell'architettura

L'architettura del sistema è stata progettata come un insieme integrato di componenti hardware e software in grado di operare in modo coordinato. Tale configurazione consente al veicolo di acquisire dati, interpretarli e generare azioni di controllo coerenti con le condizioni ambientali, garantendo autonomia decisionale e precisione operativa.

La complessità della competizione richiede una struttura capace di funzionare in tempo reale, assicurando continuità nell'acquisizione dei dati, robustezza nelle elaborazioni e stabilità nell'attuazione dei comandi. I componenti descritti nel capitolo precedente si organizzano in tre blocchi funzionali:

- sensori, dedicati alla misura dello stato dinamico del veicolo e delle caratteristiche dell'ambiente circostante;
- unità di calcolo e controllo, responsabili dell'elaborazione dei dati e della generazione dei comandi;
- attuatori, incaricati dell'esecuzione fisica delle azioni pianificate.

Il sottosistema sensoriale fornisce le informazioni necessarie per la percezione e la localizzazione mediante il sistema UWB, IMU, gli encoder, la telecamera e il LiDAR. Le unità di calcolo costituiscono il livello decisionale del sistema: la Jetson Orin NX elabora i dati provenienti dai sensori e pianifica i movimenti, mentre la STM32 Nucleo-G491RE gestisce il controllo in tempo reale e la trasmissione dei segnali di comando agli attuatori. Il gruppo degli attuatori, composto dal motore di trazione e dal servosterzo, realizza infine i movimenti pianificati, chiudendo il ciclo di controllo.

3.2. Architettura modulare e pipeline di elaborazione

Il comportamento del veicolo è governato da una pipeline di elaborazione organizzata in moduli funzionali interconnessi. Questa struttura consente una gestione ordinata del flusso informativo, in cui i dati acquisiti vengono progressivamente trasformati in comandi di controllo. Le principali fasi del processo sono le seguenti:

Capitolo 3. Architettura del sistema

1. acquisizione sensoriale: raccolta e sincronizzazione dei dati provenienti dai sensori di bordo e dal sistema di localizzazione;
2. fusione e pre-elaborazione: combinazione delle misure eterogenee per la stima coerente dello stato dinamico del veicolo;
3. pianificazione e decisione: definizione delle azioni da eseguire in base alla percezione dell'ambiente e agli obiettivi di guida;
4. controllo e attuazione: generazione dei segnali di comando e loro applicazione ai dispositivi di trazione e sterzo.

Come illustrato in Figura 3.1, le funzioni di alto livello sono eseguite sulla Jetson Orin NX e comprendono la percezione, la localizzazione e la pianificazione del moto. Il microcontrollore STM32 Nucleo-G491RE si occupa invece del controllo in tempo reale, della gestione dei sensori a basso livello e dell'attuazione dei comandi. L'interfaccia di comunicazione tra l'unità di calcolo e il microcontrollore è basata su un protocollo seriale UART operante a 921600 baud. Al fine di assicurare la robustezza del flusso odometrico, l'integrità dei dati è garantita dall'implementazione di un meccanismo di controllo degli errori tramite checksum CRC-16 su ogni frame. Sul versante STM32, la trasmissione è asservita a controller DMA (Direct Memory Access): tale configurazione risulta determinante per minimizzare il jitter temporale e garantire il determinismo dei task di comunicazione, riducendo al contempo l'overhead computazionale della CPU. L'architettura risultante presenta un'elevata modularità e una chiara separazione funzionale, che consentono l'integrazione di nuovi algoritmi e la riconfigurazione del sistema senza comprometterne la stabilità. La localizzazione del veicolo rappresenta il nucleo logico del sistema, poiché fornisce le informazioni di riferimento necessarie per la pianificazione e il controllo della traiettoria.

L'intero framework di elaborazione è implementato secondo una logica a nodi distribuiti, in cui ogni modulo esegue una funzione specifica ed è connesso agli altri mediante interfacce standardizzate. La comunicazione tra moduli avviene attraverso scambi asincroni di messaggi, che consentono l'esecuzione parallela dei processi e un uso efficiente delle risorse computazionali. Questa configurazione riduce i tempi di latenza e permette di aggiornare i dati sensoriali e gli stati di controllo con frequenza costante, caratteristica essenziale per la gestione di un sistema dinamico. La modularità della pipeline favorisce inoltre la manutenibilità del codice e l'adattabilità a scenari differenti, consentendo di sostituire o migliorare singoli componenti senza dover modificare l'intero sistema.

Dal punto di vista software, la gestione delle comunicazioni e la sincronizzazione dei flussi dati sono affidate a un middleware robotico basato su ROS 2, che garantisce astrazione delle interfacce e scalabilità dell'architettura. Ogni nodo del sistema pubblica e sottoscrive messaggi su specifici topic, permettendo il passaggio dei dati sensoriali e dei comandi di controllo in modo trasparente.

Capitolo 3. Architettura del sistema

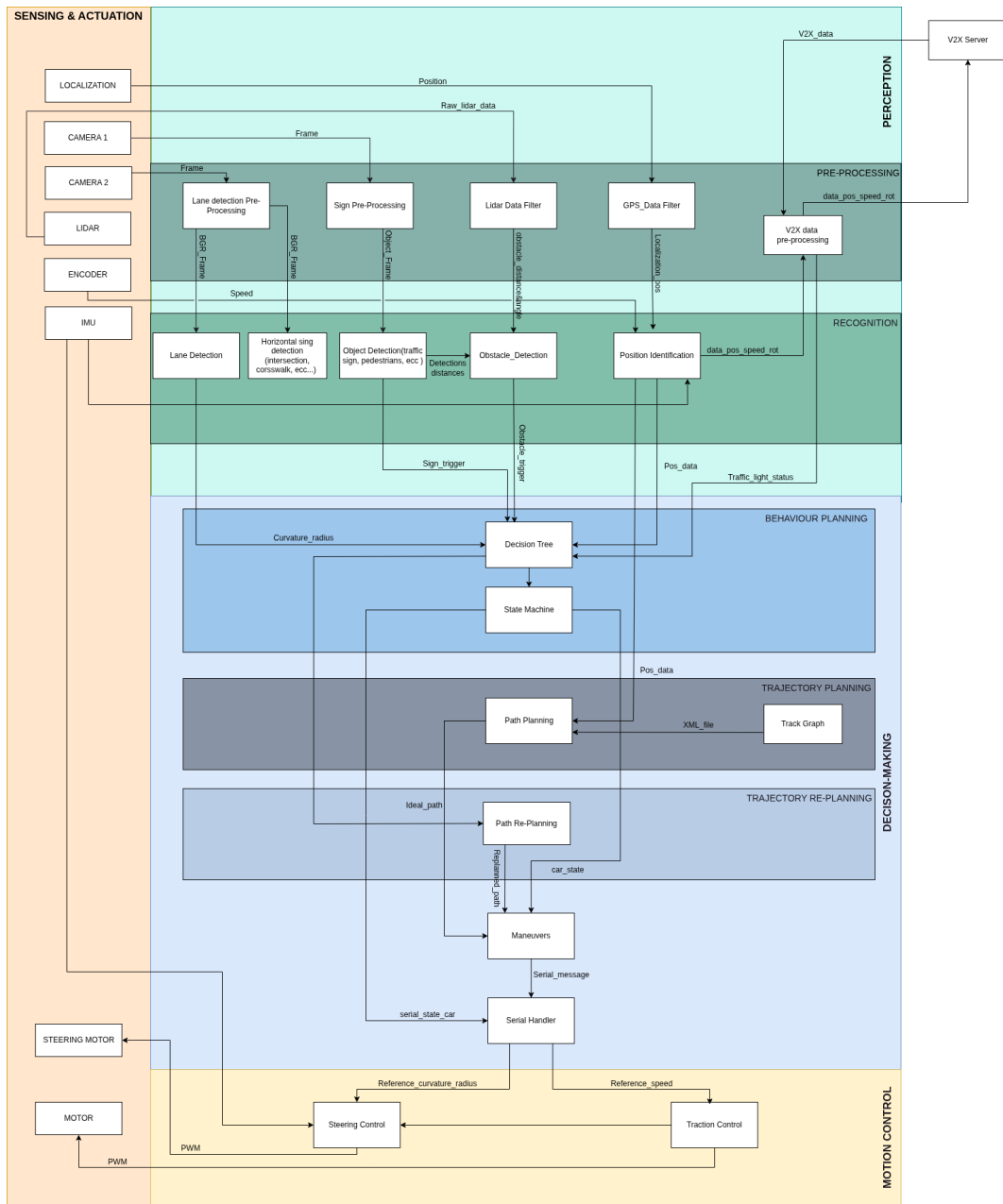


Figura 3.1.: Schema dell'architettura modulare del sistema.

In questa sezione viene presentata l'architettura generale del sistema per la guida autonoma. L'architettura ROS 2 specificamente dedicata alla localizzazione, che costituisce l'obiettivo principale della tesi, verrà descritta in dettaglio nei capitoli successivi, con l'analisi del flusso informativo, delle interazioni tra nodi e delle strategie di fusione e stima adottate.

Parte III.

Fondamenti teorici e metodologie di localizzazione autonoma

Capitolo 4.

Localizzazione locale

La localizzazione rappresenta un elemento fondamentale per il funzionamento dei sistemi autonomi mobili, poiché consente al veicolo di stimare in modo continuo la propria posa, ossia posizione e orientamento, rispetto all'ambiente circostante. Essa costituisce la base informativa per la pianificazione del percorso, l'evitamento degli ostacoli e l'esecuzione delle manovre di guida. In assenza di una stima accurata e robusta della posa, anche algoritmi di controllo e decisione avanzati risulterebbero inefficaci, poiché il sistema perderebbe la capacità di mantenere la coerenza spaziale tra percezione sensoriale e rappresentazione del mondo esterno.

Nel contesto della navigazione autonoma, la localizzazione si colloca all'interno del ciclo funzionale composto da percezione, cognizione e controllo del moto: la percezione acquisisce le informazioni dai sensori, la cognizione elabora le strategie operative e il controllo del moto traduce tali strategie in comandi per gli attuatori. La localizzazione fornisce la stima dello stato corrente del sistema, fungendo da interfaccia tra percezione e controllo e garantendo la coerenza temporale e spaziale delle informazioni.

Si distinguono due principali approcci alla localizzazione:

- localizzazione locale, che stima la variazione di posa rispetto a una configurazione iniziale, basandosi su misure interne al veicolo;
- localizzazione globale, che determina la posa rispetto a un sistema di riferimento assoluto o a una mappa predefinita dell'ambiente.

Nel seguito l'attenzione è rivolta alla localizzazione locale, che costituisce la base di ogni architettura di navigazione autonoma. Essa si fonda sull'impiego di sensori propriocettivi, quali encoder e IMU, per stimare il moto del veicolo mediante l'integrazione temporale delle grandezze cinematiche, come velocità e accelerazioni. Questo approccio consente di ricostruire in tempo reale la traiettoria percorsa, ma risulta soggetto a errori cumulativi e a una deriva progressiva della posa nel tempo. Le misure sensoriali risultano affette da rumore stocastico e bias sistematici dovuti a limitazioni fisiche, condizioni ambientali e imperfezioni meccaniche. Vibrazioni, riflessioni e variazioni termiche possono compromettere la precisione dei sensori inerziali o degli encoder, mentre fenomeni come lo slittamento o i disallineamenti meccanici introducono rumore attuativo nella stima del moto. Questi effetti determinano una

deriva odometrica che si manifesta attraverso errori di distanza, rotazione e deriva laterale, con una propagazione non lineare che produce deviazioni crescenti rispetto alla traiettoria reale.

L'incertezza nella localizzazione deriva quindi sia dalla limitata capacità osservativa dei sensori, sia dalla variabilità introdotta dagli attuatori. Per mitigare tali effetti si impiegano tecniche di fusione sensoriale, che combinano in modo ottimale informazioni provenienti da sensori eterogenei secondo un approccio bayesiano di predizione e aggiornamento. In tale formalismo, la localizzazione può essere modellata come un problema di stima di stato di un sistema dinamico non lineare:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k), \quad \mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k)$$

dove $\mathbf{x}_k$ rappresenta lo stato (posa), $\mathbf{u}_k$ gli ingressi di controllo, $\mathbf{y}_k$ le uscite misurate, e $\mathbf{w}_k, \mathbf{v}_k$ i rumori di processo e di osservazione.

L'obiettivo della fusione sensoriale è stimare lo stato $\mathbf{x}_k$ in modo coerente e a varianza minima, compensando le limitazioni individuali dei sensori e migliorando l'accuratezza complessiva del sistema. Essa consente inoltre di aumentare la robustezza del processo di localizzazione e di agevolare la diagnosi precoce di guasti o anomalie sensoriali.

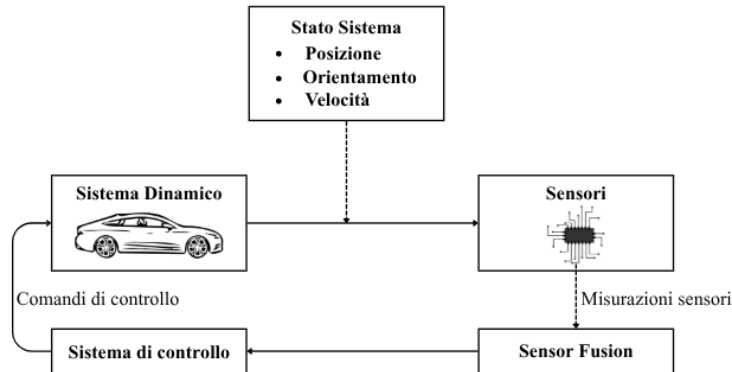


Figura 4.1.: Schema funzionale del processo di localizzazione locale.

La Figura 4.1 illustra la struttura generale del ciclo di localizzazione odometrica, evidenziando il flusso informativo tra sensori, stimatore di stato e controllore del moto. Il modulo di fusione sensoriale costituisce il nucleo del processo, garantendo una stima accurata e stabile della posa del veicolo. Nei paragrafi successivi verranno analizzati i modelli dinamici e le principali strategie di fusione basate sul filtro di Kalman e sulle sue estensioni non lineari, che rappresentano i metodi di riferimento per la localizzazione odometrica nei sistemi autonomi.

4.1. Filtri gaussiani

Come introdotto nella sezione precedente, la fusione sensoriale si basa su metodologie probabilistiche che consentono di combinare misure provenienti da sensori eterogenei,

tenendo conto dell'incertezza associata sia al processo di misura sia al modello dinamico del sistema. Tra le tecniche più diffuse per la stima dello stato si annoverano i filtri gaussiani, che forniscono un approccio ottimale o approssimato alla fusione delle informazioni, assumendo che l'incertezza segua una distribuzione di probabilità gaussiana. Essi costituiscono il fondamento teorico dei filtri di Kalman e delle loro estensioni non lineari, ampiamente impiegati nelle moderne strategie di localizzazione.

4.1.1. Filtro di Kalman (KF)

Il filtro di Kalman è uno strumento statistico per la stima dello stato di sistemi dinamici lineari affetti da rumore. È largamente impiegato in ambiti quali la robotica, la navigazione e il controllo, dove le misure dirette dello stato non risultano sempre affidabili o completamente disponibili. Il principio di base consiste nel combinare in modo ottimale due fonti di informazione: la predizione fornita dal modello dinamico e la misura effettiva, entrambe affette da incertezza.

Il filtro è progettato per sistemi dinamici lineari discreti nel tempo, descritti dal modello di stato

$$\mathbf{x}_k = A_k \mathbf{x}_{k-1} + B_k \mathbf{u}_{k-1} + \mathbf{w}_k \quad (4.1)$$

$$\mathbf{y}_k = H_k \mathbf{x}_k + \mathbf{v}_k \quad (4.2)$$

dove:

- $\mathbf{x}_k$ è il vettore di stato al tempo k ;
- A_k è la matrice di transizione di stato, che descrive l'evoluzione dinamica del sistema;
- B_k è la matrice di ingresso associata al vettore di controllo $\mathbf{u}_{k-1}$;
- $\mathbf{y}_k$ è il vettore di misura osservato;
- $\mathbf{w}_k$ e $\mathbf{v}_k$ rappresentano rispettivamente i rumori di processo e di misura, modellati come variabili gaussiane a media nulla:

$$\mathbf{w}_k \sim \mathcal{N}(0, Q_k), \quad \mathbf{v}_k \sim \mathcal{N}(0, R_k)$$

dove Q_k e R_k sono le corrispondenti matrici di covarianza.

Il filtro di Kalman opera in modo ricorsivo e si articola in due fasi fondamentali:

- fase di predizione, nella quale si stima lo stato futuro e la relativa incertezza;
- fase di aggiornamento, nella quale la stima viene corretta sulla base della misura disponibile.

Fase di predizione

In questa fase il filtro utilizza il modello dinamico per stimare lo stato al tempo k prima della nuova osservazione. Si ottengono le stime a priori dello stato e della covarianza dell'errore:

$$\hat{\mathbf{x}}_{k|k-1} = A_k \hat{\mathbf{x}}_{k-1|k-1} + B_k \mathbf{u}_{k-1} \quad (4.3)$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k \quad (4.4)$$

La prima equazione fornisce la predizione dello stato a partire dalla stima precedente e dal comando di controllo, mentre la seconda aggiorna la covarianza dell'errore, includendo l'incertezza introdotta dal rumore di processo $\mathbf{w}_k$.

Fase di aggiornamento

Quando una nuova misura $\mathbf{y}_k$ diventa disponibile, il filtro corregge la stima a priori sulla base dell'informazione osservata. Il passo centrale è il calcolo del guadagno di Kalman K_k , che determina il peso relativo della predizione e della misura:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (4.5)$$

Il guadagno di Kalman agisce come coefficiente di fusione adattivo: un valore elevato di R_k induce maggiore fiducia nel modello, mentre una misura più precisa comporta una correzione più incisiva.

A partire da K_k , la stima dello stato e la covarianza vengono aggiornate secondo:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{y}_k - H_k \hat{\mathbf{x}}_{k|k-1}) \quad (4.6)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (4.7)$$

Forma algoritmica

Le due fasi appena descritte possono essere sintetizzate nella forma ricorsiva mostrata nell'Algoritmo 1, che riassume i passaggi di predizione e aggiornamento.

Algorithm 1 Filtro di Kalman

Require: $\hat{\mathbf{x}}_{k-1|k-1}$, $P_{k-1|k-1}$, $\mathbf{u}_{k-1}$, $\mathbf{y}_k$

Ensure: $\hat{\mathbf{x}}_{k|k}$, $P_{k|k}$

▷ Fase di predizione

1: $\hat{\mathbf{x}}_{k|k-1} = A_k \hat{\mathbf{x}}_{k-1|k-1} + B_k \mathbf{u}_{k-1}$

2: $P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k$

▷ Fase di aggiornamento

3: $K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$

4: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{y}_k - H_k \hat{\mathbf{x}}_{k|k-1})$

5: $P_{k|k} = (I - K_k H_k) P_{k|k-1}$

6: **return** $\hat{\mathbf{x}}_{k|k}$, $P_{k|k}$

La Figura 4.2 illustra in modo qualitativo il funzionamento del filtro di Kalman, evidenziando l'alternanza tra le fasi di predizione e aggiornamento. La distribuzione della credenza evolve nel tempo secondo il modello di transizione, mentre le misure acquisite introducono correzioni che riducono progressivamente l'incertezza associata alla stima dello stato.

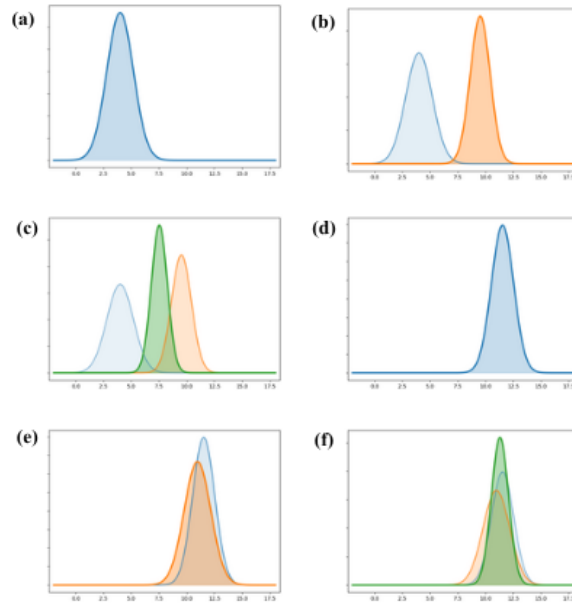


Figura 4.2.: Schema qualitativo del funzionamento del filtro di Kalman.

Le sottosezioni rappresentano le principali fasi del processo ricorsivo:

- (a) distribuzione iniziale della credenza;
- (b) acquisizione di una nuova misura;
- (c) aggiornamento della stima;
- (d) predizione successiva;

- (e) nuova osservazione;
- (f) stima finale risultante dalla fusione tra informazione predetta e misurata.

Il filtro di Kalman fornisce una stima ottima in termini di errore quadratico medio, a condizione che il sistema sia lineare e che i rumori di processo e di misura siano gaussiani. Nella pratica, tuttavia, molti sistemi fisici presentano dinamiche non lineari, rendendo necessarie estensioni del metodo classico, quali il filtro di Kalman esteso (EKF) e il filtro di Kalman unscented (UKF).

4.1.2. Filtro di Kalman esteso (EKF)

Il filtro di Kalman esteso (Extended Kalman Filter, EKF) rappresenta una delle estensioni più diffuse del filtro di Kalman, utilizzata per la stima dello stato in sistemi caratterizzati da modelli di transizione e osservazione non lineari. Il principio di base consiste nella linearizzazione locale delle funzioni che descrivono la dinamica e la misura del sistema mediante lo sviluppo di Taylor di primo ordine attorno alla stima corrente dello stato. Questa approssimazione consente di mantenere una rappresentazione gaussiana del processo di stima, garantendo un compromesso efficace tra accuratezza e efficienza computazionale.

Modello del sistema

Un sistema dinamico non lineare può essere descritto come:

$$\mathbf{x}_k = g(\mathbf{u}_{k-1}, \mathbf{x}_{k-1}) + \boldsymbol{\varepsilon}_k, \quad (4.8)$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \boldsymbol{\delta}_k, \quad (4.9)$$

dove:

- $\mathbf{x}_k$ è il vettore di stato;
- $\mathbf{u}_{k-1}$ l'ingresso di controllo;
- $\mathbf{y}_k$ il vettore di misura;
- $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, Q_k)$ e $\boldsymbol{\delta}_k \sim \mathcal{N}(0, R_k)$ rappresentano rumori gaussiani a media nulla e covarianza nota.

Linearizzazione di primo ordine

Le funzioni non lineari $g(\cdot)$ e $h(\cdot)$ vengono linearizzate attorno ai valori stimati dello stato tramite sviluppo di Taylor di primo ordine:

$$g(\mathbf{u}_{k-1}, \mathbf{x}_{k-1}) \approx g(\mathbf{u}_{k-1}, \hat{\mathbf{x}}_{k-1|k-1}) + G_k (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}), \quad (4.10)$$

$$h(\mathbf{x}_k) \approx h(\hat{\mathbf{x}}_{k|k-1}) + H_k (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}), \quad (4.11)$$

dove le matrici Jacobiane sono definite come:

$$G_k = \left. \frac{\partial g(\mathbf{u}_{k-1}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}}, \quad H_k = \left. \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}.$$

Le matrici G_k e H_k forniscono un'approssimazione lineare locale delle funzioni di transizione e osservazione.

Equazioni operative

Le equazioni di predizione e aggiornamento dell'EKF assumono la forma:

$$\hat{\mathbf{x}}_{k|k-1} = g(\mathbf{u}_{k-1}, \hat{\mathbf{x}}_{k-1|k-1}), \quad (4.12)$$

$$P_{k|k-1} = G_k P_{k-1|k-1} G_k^T + Q_k, \quad (4.13)$$

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}, \quad (4.14)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})), \quad (4.15)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \quad (4.16)$$

Le matrici Jacobiane vengono ricalcolate a ogni iterazione per adeguare la linearizzazione alla regione locale della stima corrente.

Forma algoritmica

Algorithm 2 Filtro di Kalman esteso

Require: $\hat{\mathbf{x}}_{k-1|k-1}$, $P_{k-1|k-1}$, $\mathbf{u}_{k-1}$, $\mathbf{y}_k$

Ensure: $\hat{\mathbf{x}}_{k|k}$, $P_{k|k}$

- 1: $G_k \leftarrow \left. \frac{\partial g(\mathbf{u}_{k-1}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}}$ ▷ Linearizzazione del modello di transizione
 - ▷ Fase di predizione
 - 2: $\hat{\mathbf{x}}_{k|k-1} = g(\mathbf{u}_{k-1}, \hat{\mathbf{x}}_{k-1|k-1})$
 - 3: $P_{k|k-1} = G_k P_{k-1|k-1} G_k^T + Q_k$
 - ▷ Linearizzazione del modello di osservazione
 - 4: $H_k \leftarrow \left. \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}}$
 - ▷ Fase di aggiornamento
 - 5: $S_k = H_k P_{k|k-1} H_k^T + R_k$
 - 6: $K_k = P_{k|k-1} H_k^T S_k^{-1}$
 - 7: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k (\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1}))$
 - 8: $P_{k|k} = (I - K_k H_k) P_{k|k-1}$
 - 9: **return** $\hat{\mathbf{x}}_{k|k}$, $P_{k|k}$
-

L'algoritmo dell'EKF conserva la struttura del filtro di Kalman lineare, articolandosi nelle fasi di predizione e aggiornamento. Durante la predizione, la stima dello stato è

calcolata mediante la funzione non lineare $g(\cdot)$, mentre la covarianza viene aggiornata utilizzando il Jacobiano G_k . Nella fase di aggiornamento, la predizione della misura è ottenuta attraverso $h(\cdot)$, e il Jacobiano H_k fornisce la linearizzazione locale del modello di osservazione. Queste operazioni estendono la formulazione classica del filtro di Kalman a sistemi non lineari, preservandone la logica computazionale.

Tabella 4.1.: Confronto tra filtro di Kalman e filtro di Kalman esteso.

Operazione	Filtro di Kalman	EKF
Predizione dello stato	$A_k \hat{\mathbf{x}}_{k-1 k-1} + B_k \mathbf{u}_{k-1}$	$g(\mathbf{u}_{k-1}, \hat{\mathbf{x}}_{k-1 k-1})$
Predizione della misura	$H_k \hat{\mathbf{x}}_{k k-1}$	$h(\hat{\mathbf{x}}_{k k-1})$
Derivate richieste	Nessuna	Jacobiane G_k, H_k
Tipo di modello	Lineare	Non lineare (linearizzato)
Accuratezza	Ottima se lineare	Buona per deboli non linearità
Complessità computazionale	Bassa	Media

L'affidabilità dell'EKF dipende dalla qualità dell'approssimazione lineare nella regione locale della stima. In presenza di forti non linearità o di elevata incertezza, l'errore sulla stima di media e covarianza può aumentare sensibilmente. In tali condizioni risultano più adatte tecniche alternative, quali l'unscented Kalman filter (UKF) o metodi basati su rappresentazioni multi-ipotesi mediante miscele gaussiane.

4.1.3. Filtro di Kalman unscented (UKF)

Il filtro di Kalman unscented (Unscented Kalman Filter, UKF) costituisce un'evoluzione del filtro di Kalman esteso, concepita per migliorare la stima dello stato in presenza di modelli fortemente non lineari. Mentre l'EKF si basa sulla linearizzazione mediante sviluppo di Taylor di primo ordine, l'UKF utilizza la trasformazione unscented (Unscented Transform, UT), che consente di approssimare in modo più accurato la propagazione di una distribuzione gaussiana attraverso funzioni non lineari, senza ricorrere a derivate analitiche.

Principio della trasformazione unscented

La trasformazione unscented si fonda sull'assunto che sia più efficace approssimare la distribuzione statistica di una variabile casuale piuttosto che linearizzare direttamente la funzione non lineare. A tale scopo, l'UKF genera un insieme deterministico di punti campione, detti sigma points, scelti in modo da riprodurre esattamente la media e la covarianza della distribuzione gaussiana associata allo stato stimato. Questi punti vengono propagati attraverso le funzioni di transizione e di osservazione, e dai risultati della propagazione si ricostruiscono una nuova media e una nuova covarianza.

Capitolo 4. Localizzazione locale

Per uno stato di dimensione n , vengono generati $2n + 1$ sigma points $\mathbf{X}^{[i]}$ secondo:

$$\mathbf{X}^{[0]} = \boldsymbol{\mu}, \quad \mathbf{X}^{[i]} = \boldsymbol{\mu} + \sqrt{(n + \lambda)\boldsymbol{\Sigma}_i}, \quad \mathbf{X}^{[i+n]} = \boldsymbol{\mu} - \sqrt{(n + \lambda)\boldsymbol{\Sigma}_i},$$

dove $\lambda = \alpha^2(n + \kappa) - n$ è un parametro di scala che controlla la dispersione dei punti attorno alla media, e α, κ sono coefficienti di taratura. A ciascun punto sono associati pesi per il calcolo della media ($w_m^{[i]}$) e della covarianza ($w_c^{[i]}$):

$$w_m^{[0]} = \frac{\lambda}{n + \lambda}, \quad w_c^{[0]} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta), \quad w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n + \lambda)}.$$

Il parametro β permette di introdurre informazioni a priori sulla distribuzione; per una distribuzione perfettamente gaussiana, il valore ottimale è $\beta = 2$.

I sigma points vengono propagati attraverso la funzione di transizione di stato $g(\cdot)$:

$$\mathbf{Y}^{[i]} = g(\mathbf{X}^{[i]}),$$

e la nuova media e covarianza vengono calcolate come combinazioni pesate dei punti trasformati:

$$\bar{\boldsymbol{\mu}} = \sum_{i=0}^{2n} w_m^{[i]} \mathbf{Y}^{[i]}, \quad \bar{\boldsymbol{\Sigma}} = \sum_{i=0}^{2n} w_c^{[i]} (\mathbf{Y}^{[i]} - \bar{\boldsymbol{\mu}})(\mathbf{Y}^{[i]} - \bar{\boldsymbol{\mu}})^T.$$

Questo metodo garantisce un'accuratezza fino al secondo ordine nello sviluppo di Taylor, migliorando sensibilmente la precisione rispetto al filtro di Kalman esteso.

Algoritmo UKF

L'algoritmo dell'UKF mantiene la stessa struttura del filtro di Kalman, sostituendo la fase di linearizzazione con la trasformazione unscented. A ogni iterazione, i sigma points vengono generati a partire dalla distribuzione dello stato corrente e propagati attraverso i modelli dinamico e di misura.

Algorithm 3 Filtro di Kalman unscented

Require: $\boldsymbol{\mu}_{k-1}$, $\boldsymbol{\Sigma}_{k-1}$, $\mathbf{u}_{k-1}$, $\mathbf{y}_k$

Ensure: $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$

- 1: $\lambda = \alpha^2(n + \kappa) - n$
 - 2: $\gamma = \sqrt{n + \lambda}$
 - 3: Calcola $\{\mathbf{X}_{k-1}^{[i]}\}_{i=0}^{2n}$ da $\boldsymbol{\mu}_{k-1}$ e $\boldsymbol{\Sigma}_{k-1}$ ▷ Generazione dei sigma points
 - 4: Propaga $\mathbf{X}_{k|k-1}^{[i]} = g(\mathbf{u}_{k-1}, \mathbf{X}_{k-1}^{[i]})$
 - 5: $\bar{\boldsymbol{\mu}}_k = \sum w_m^{[i]} \mathbf{X}_{k|k-1}^{[i]}$
 - 6: $\bar{\boldsymbol{\Sigma}}_k = \sum w_c^{[i]} (\mathbf{X}_{k|k-1}^{[i]} - \bar{\boldsymbol{\mu}}_k)(\mathbf{X}_{k|k-1}^{[i]} - \bar{\boldsymbol{\mu}}_k)^T + Q_k$ ▷ Predizione dello stato
 - 7: Propaga $\mathbf{Z}_k^{[i]} = h(\mathbf{X}_{k|k-1}^{[i]})$
 - 8: $\hat{\mathbf{z}}_k = \sum w_m^{[i]} \mathbf{Z}_k^{[i]}$
 - 9: $S_k = \sum w_c^{[i]} (\mathbf{Z}_k^{[i]} - \hat{\mathbf{z}}_k)(\mathbf{Z}_k^{[i]} - \hat{\mathbf{z}}_k)^T + R_k$
 - 10: $\Sigma_{xz} = \sum w_c^{[i]} (\mathbf{X}_{k|k-1}^{[i]} - \bar{\boldsymbol{\mu}}_k)(\mathbf{Z}_k^{[i]} - \hat{\mathbf{z}}_k)^T$
 - 11: $K_k = \Sigma_{xz} S_k^{-1}$
 - 12: $\boldsymbol{\mu}_k = \bar{\boldsymbol{\mu}}_k + K_k(\mathbf{y}_k - \hat{\mathbf{z}}_k)$ ▷ Aggiornamento della misura
 - 13: $\boldsymbol{\Sigma}_k = \bar{\boldsymbol{\Sigma}}_k - K_k S_k K_k^T$
 - 14: **return** $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$
-

L'UKF mantiene la stessa complessità computazionale dell'EKF, ma fornisce stime più accurate in presenza di forti non linearità. Il filtro non richiede la derivazione di Jacobiani, risultando più stabile e applicabile anche a sistemi con modelli complessi o discontinui. Per sistemi lineari, i risultati coincidono con quelli del filtro di Kalman classico, mentre per sistemi non lineari l'UKF produce stime più coerenti della media e della covarianza.

Grazie alla maggiore robustezza e flessibilità, l'UKF trova ampio impiego nelle applicazioni di fusione sensoriale, in cui è necessario integrare misure provenienti da sensori eterogenei. Nella sezione successiva verrà presentato il modello cinematico del veicolo, che costituisce la base per la definizione delle equazioni di stato e di osservazione utilizzate nella fase di predizione del filtro, consentendo di stimare posizione e assetto del veicolo all'interno del processo di fusione sensoriale.

4.2. Modello cinematico del veicolo

Definite le metodologie di fusione basate su filtri gaussiani, si introduce il modello cinematico che descrive l'evoluzione dello stato del veicolo nel tempo, elemento fondamentale per l'applicazione pratica dei filtri EKF e UKF. Il modello adottato è di tipo planare e assume condizioni di aderenza pura, ovvero assenza di slittamento laterale tra pneumatico e superficie di contatto. Gli effetti dovuti all'attrito e allo

slittamento vengono trascurati, poiché di difficile caratterizzazione analitica e di entità ridotta nei veicoli in scala, per i quali le basse velocità e le masse contenute minimizzano l'influenza di tali fenomeni.

In tali condizioni, il veicolo è rappresentato come un corpo rigido in moto planare, caratterizzato da due gradi di libertà principali: traslazione e rotazione attorno a un asse verticale.

4.2.1. Sistemi di riferimento

La descrizione formale del moto richiede l'introduzione di più sistemi di riferimento, necessari per correlare le grandezze misurate dai sensori con la posizione e l'orientamento del veicolo.

- Sistema di riferimento inerziale: solidale con il suolo, definito da un sistema cartesiano destrorso (x, y, z) , dove z è ortogonale al piano di moto e orientato in senso opposto alla gravità. Esso costituisce il riferimento globale per la localizzazione del veicolo.
- Sistema di riferimento veicolo: solidale al telaio, con origine generalmente posta al centro dell'asse posteriore. Gli assi (x_v, y_v, z_v) seguono la convenzione ISO 8855: x_v orientato in avanti, y_v verso sinistra e z_v verso l'alto.

La Figura 4.3 illustra la relazione geometrica tra i due sistemi di riferimento, mostrando come il moto del veicolo possa essere interpretato come una trasformazione rigida tra riferimento inerziale e riferimento solidale.

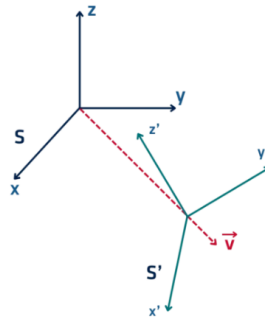


Figura 4.3.: Relazione tra il sistema di riferimento inerziale $S(x, y, z)$ e quello solidale al veicolo $S'(x', y', z')$. Il vettore $\vec{v}$ rappresenta la velocità del veicolo rispetto al sistema inerziale.

In questo contesto, la posizione (x, y) e l'orientamento ψ del veicolo sono definiti nel sistema inerziale, mentre le grandezze cinematiche, quali la velocità longitudinale v e l'angolo di sterzo δ , sono espresse nel sistema veicolo. Il passaggio tra i riferimenti è pertanto necessario per formulare le equazioni del modello cinematico in modo coerente con le misure sensoriali.

4.2.2. Modello a bicicletta

Il comportamento cinematico del veicolo è rappresentato dal modello a bicicletta, una semplificazione del modello Ackermann che riduce le quattro ruote a due ruote equivalenti, una anteriore e una posteriore. Tale formulazione mantiene la coerenza geometrica della sterzata e consente di descrivere il moto con un numero ridotto di variabili, garantendo al contempo una buona accuratezza per manovre a bassa velocità.

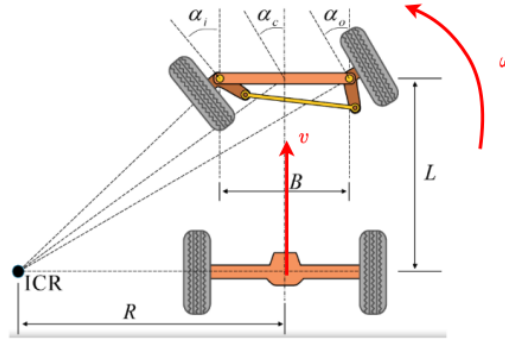


Figura 4.4.: Rappresentazione schematica del modello cinematico Ackermann.

Durante una manovra di sterzata, tutte le ruote descrivono archi di circonferenza con centro comune nel punto di istantanea rotazione (Instantaneous Center of Rotation, ICR). L'angolo di sterzo δ della ruota anteriore definisce la curvatura della traiettoria, mentre il raggio di curvatura R risulta legato all'interasse L dalla relazione:

$$R = \frac{L}{\tan(\delta)}. \quad (4.17)$$

4.2.3. Equazioni cinematiche

Assumendo moto planare e rotolamento puro, la dinamica del punto di riferimento posto al centro dell'asse posteriore è descritta dal sistema:

$$\dot{x} = v \cos(\psi), \quad (4.18)$$

$$\dot{y} = v \sin(\psi), \quad (4.19)$$

$$\dot{\psi} = \frac{v}{L} \tan(\delta), \quad (4.20)$$

dove:

- (x, y) rappresentano le coordinate del veicolo nel sistema di riferimento inerziale;
- ψ indica l'angolo di imbardata;
- v è la velocità longitudinale espressa nel sistema veicolo;
- δ è l'angolo di sterzo anteriore;

- L è l'interasse.

La relazione

$$\dot{\psi} = \frac{v}{R} = \frac{v}{L} \tan(\delta) \quad (4.21)$$

definisce la velocità angolare di imbardata, grandezza fondamentale per la stima dell'orientamento e per il controllo della traiettoria.

Il modello cinematico a bicicletta costituisce la base per la definizione delle equazioni di stato e di osservazione impiegate nei filtri di stima, come EKF e UKF. In tali algoritmi, le variabili di ingresso (v, δ) rappresentano le grandezze note o misurate, mentre lo stato del sistema è composto da (x, y, ψ) . Il modello fornisce la funzione di transizione necessaria alla fase di predizione, consentendo di aggiornare la stima della posizione e dell'orientamento del veicolo sulla base dei dati sensoriali.

Grazie alla semplicità e alla coerenza geometrica, questa formulazione risulta ampiamente adottata nei sistemi di fusione sensoriale e nei modelli di navigazione per veicoli autonomi, in particolare nelle fasi di localizzazione e controllo della traiettoria.

4.3. Modellazione dei sensori

La modellazione dei sensori costituisce un aspetto essenziale nei sistemi di localizzazione e navigazione autonoma, poiché descrive formalmente la relazione tra lo stato cinematico del veicolo e le grandezze fisiche effettivamente misurate. Nel contesto della robotica mobile, una rappresentazione accurata dei sensori è indispensabile per garantire coerenza e affidabilità nelle stime di posizione, velocità e orientamento anche in condizioni operative non ideali.

Le osservazioni sensoriali rappresentano la principale fonte informativa del sistema, ma risultano affette da rumore, bias e incertezze strumentali che ne degradano la precisione. Per migliorare la robustezza delle stime, i dati provenienti da sensori eterogenei vengono combinati mediante tecniche di fusione sensoriale, che integrano informazioni ridondanti e complementari in un quadro probabilistico coerente. In tale ambito, i filtri bayesiani consentono l'aggiornamento in tempo reale della stima dello stato combinando il modello dinamico con le misure osservate, pesate in funzione della loro incertezza.

La dinamica del veicolo è descritta da un modello di transizione di stato generale:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \quad (4.22)$$

dove $\mathbf{x}_k$ è il vettore di stato al tempo k , $\mathbf{u}_k$ il vettore di ingresso e $\mathbf{w}_k$ il rumore di processo, modellato come variabile gaussiana a media nulla e covarianza Q_k .

I sensori comunemente impiegati comprendono:

- IMU con accelerometri e giroscopi, per le accelerazioni lineari e le velocità angolari;
- encoder ruota, per la stima delle velocità e degli spostamenti relativi;

- sistema di posizionamento assoluto UWB, per la determinazione delle coordinate globali.

Le misure fornite da ciascun sensore sono descritte dal modello osservazionale:

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (4.23)$$

dove $h(\cdot)$ è la funzione di osservazione e $\mathbf{v}_k$ il rumore di misura, modellato come variabile gaussiana a media nulla e covarianza R_k .

Gli errori sensoriali si distinguono in componenti deterministiche e stocastiche. Le prime includono bias costanti, errori di fattore di scala, non linearità e dipendenze ambientali, mitigabili mediante calibrazione. Le seconde comprendono rumore bianco, deriva del bias e instabilità di scala, che richiedono una gestione dinamica nel processo di stima. La modellazione probabilistica consente di rappresentare tali incertezze e di valutarne in tempo reale l'impatto sulle misure.

4.3.1. Modellazione matematica dell'IMU BNO055

Il sensore inerziale Bosch BNO055 integra giroscopio, accelerometro e magnetometro triassiali. A differenza delle IMU convenzionali, incorpora un microcontrollore che esegue fusioni interne e compensazioni di errori deterministici tramite calibrazioni automatiche, riducendo bias statici, errori di scala e disallineamenti.

In modalità NDOF (Nine Degrees of Freedom) fornisce direttamente quaternioni di orientamento o angoli di Eulero (ϕ, θ, ψ) . Ai fini della stima si impiega un modello a misura elementare che mette in relazione le grandezze reali con le misure acquisite:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g, \quad \mathbf{a}_m = \mathbf{a} + \mathbf{b}_a + \mathbf{n}_a, \quad (4.24)$$

dove $\mathbf{b}_g$ e $\mathbf{b}_a$ sono bias a lenta variazione e $\mathbf{n}_g$, $\mathbf{n}_a$ sono rumori gaussiani bianchi con covarianze R_g e R_a . L'evoluzione dei bias è modellata come:

$$\mathbf{b}_{g,k+1} = \mathbf{b}_{g,k} + \mathbf{w}_{g,k}, \quad \mathbf{b}_{a,k+1} = \mathbf{b}_{a,k} + \mathbf{w}_{a,k}, \quad (4.25)$$

con $\mathbf{w}_{g,k}$ e $\mathbf{w}_{a,k}$ gaussiane a media nulla e covarianze Q_g e Q_a . La distribuzione congiunta dei rumori è

$$p(\mathbf{n}_g, \mathbf{n}_a, \mathbf{w}_{g,k}, \mathbf{w}_{a,k}) = \mathcal{N}(\mathbf{0}, \Sigma), \quad \Sigma = \text{diag}(R_g, R_a, Q_g, Q_a). \quad (4.26)$$

Tali relazioni rappresentano la base teorica del modello d'errore inerziale, con il bias (4.25) come stato aumentabile per una stima in linea. Nel BNO055, però, il bias è compensato intrinsecamente: il microcontrollore integrato esegue una calibrazione interna all'inizializzazione e applica un filtro di Kalman (firmware di sensor fusion Bosch) che attenua il rumore di misura, restituendo grandezze già corrette. Il bias del giroscopio non rientra quindi tra le variabili di stato $[x, y, \psi, v]$ del filtro implementato: misurato in condizioni statiche è risultato trascurabile (Capitolo

7), per cui il giroscopio è trattato come misura non polarizzata della velocità di imbardata con varianza costante R_g . Il modello a passo casuale resta riferimento teorico per una stima in linea su orizzonti più lunghi, qualora la deriva divenisse significativa.

4.3.2. Modellazione matematica dell'encoder ruota

L'encoder incrementale rappresenta la principale sorgente di informazione odometrica per la stima della velocità longitudinale del veicolo. Installato sull'albero motore, esso fornisce un segnale proporzionale alla velocità di rotazione, da cui si ricava la velocità lineare tramite le relazioni cinematiche del sistema di trasmissione.

Indicando con n il numero di impulsi rilevati nell'intervallo Δt e con N_m il numero di impulsi per rivoluzione, la velocità angolare del motore risulta

$$\omega_m = \frac{2\pi n}{N_m \Delta t}. \quad (4.27)$$

Il rapporto totale di riduzione G , ottenuto come prodotto tra rapporto di trasmissione e rapporto del differenziale, consente di determinare la velocità angolare della ruota:

$$\omega_w = \frac{\omega_m}{G}. \quad (4.28)$$

Con raggio dinamico R della ruota, la velocità lineare del veicolo è data da

$$v = R \omega_w = \frac{2\pi R}{60 G} \text{RPM}_m, \quad (4.29)$$

dove $\text{RPM}_m = 60 \omega_m / (2\pi)$ rappresenta la velocità di rotazione del motore in giri al minuto. Un coefficiente di calibrazione k_{slip} tiene conto di slittamenti e tolleranze meccaniche:

$$v = k_{\text{slip}} \frac{2\pi R}{60 G} \text{RPM}_m. \quad (4.30)$$

Nel modello cinematico a bicicletta, illustrato in Sezione 4.2.2, la velocità longitudinale v_x è approssimata dalla velocità calcolata dall'encoder, $v_x \approx v$. La misura fornita dal sensore è modellata come osservazione affetta da bias statico e rumore gaussiano additivo:

$$v_{x,m} = v_x + b_e + n_e, \quad n_e \sim \mathcal{N}(0, R_e). \quad (4.31)$$

Tale formulazione definisce il modello di osservazione dell'encoder impiegato nei filtri di fusione gaussiani non lineari, consentendo l'integrazione coerente dell'informazione odometrica con le altre sorgenti sensoriali e garantendo consistenza statistica nella stima dello stato del veicolo.

4.3.3. Modellazione matematica del sensore UWB DWM1001

Il sistema UWB DWM1001 stima la distanza tra un nodo mobile e antenne fisse di posizione nota tramite misura del tempo di volo del segnale radio. Indicate (x, y) le

coordinate del nodo mobile e (x_i, y_i) quelle della i -esima ancora, la misura è descritta da:

$$y_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} + b_i + n_i, \quad (4.32)$$

dove b_i è il bias dovuto a condizioni non line-of-sight o a errori di sincronizzazione, e n_i è rumore gaussiano con varianza $\sigma_{r,i}^2$.

La funzione di osservazione è

$$h_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2}, \quad (4.33)$$

con jacobiano locale

$$H_i = \begin{bmatrix} \frac{x - x_i}{\rho_i} & \frac{y - y_i}{\rho_i} \end{bmatrix}, \quad \rho_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \quad (4.34)$$

Il modello di errore è

$$p(n_i) = \mathcal{N}(0, \sigma_{r,i}^2), \quad (4.35)$$

con varianza dipendente dal rumore del ricevitore e dalle condizioni di propagazione. In presenza di multipath o ostacoli, la varianza effettiva aumenta e il bias b_i tende ad assumere valori positivi, con sovrastima della distanza reale.

4.3.4. Validazione statistica delle misure

Le misure dei sensori di posizionamento assoluto, in particolare UWB, possono essere affette da misure anomale dovuti a multipath, perdite di sincronizzazione o condizioni non line-of-sight. Per limitarne l'impatto nella stima, si impiega un test statistico del chi-quadrato (χ^2).

Sia ε_k l'innovazione al passo k :

$$\varepsilon_k = \mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1}),$$

dove $\hat{\mathbf{x}}_{k|k-1}$ è la stima predetta e $h(\cdot)$ la funzione di osservazione. Sotto ipotesi gaussiane con covarianza $S_k = H P_{k|k-1} H^T + R_k$, la statistica

$$\Lambda_k = \varepsilon_k^T S_k^{-1} \varepsilon_k$$

segue una distribuzione χ^2 con m gradi di libertà, pari alla dimensione della misura. La misura è considerata compatibile se

$$\Lambda_k < \chi_{\alpha, m}^2,$$

dove $\chi_{\alpha, m}^2$ è il valore soglia per livello di confidenza α , tipicamente 0.95. In caso contrario, l'osservazione è scartata e non viene usata nell'aggiornamento del filtro, migliorando coerenza e stabilità del processo di fusione.

Tabella 4.2.: Valori soglia del test χ^2 per diversi numeri di misure.

Gradi di libertà	Dimensione misura	Soglia per $\alpha = 0.95$
1	singola misura scalare	3.84
2	coordinate (x, y)	5.99
3	posizione tridimensionale (x, y, z)	7.81

4.4. Filtri bayesiani per la stima dello stato

La stima dello stato del veicolo è implementata con un approccio bayesiano ricorsivo basato su filtri di tipo Kalman non lineari. Si adottano due schemi equivalenti per struttura ma differenti nel trattamento della non linearità: EKF, che linearizza il modello, e UKF, che utilizza la trasformazione unscented per propagare media e covarianza. In entrambi i casi la catena operativa si compone di predizione (propagazione tramite modello cinematico) e aggiornamento (correzione con misure di posizionamento assoluto), secondo le equazioni introdotte nella sezione sui filtri gaussiani.

4.4.1. Stato, ingressi e modello di transizione

Il veicolo è descritto da un modello planare con geometria Ackermann; il vettore di stato è

$$\mathbf{x} = \begin{bmatrix} x & y & \psi & v \end{bmatrix}^T, \quad (4.36)$$

dove x, y sono le coordinate nel riferimento inerziale, ψ è l'angolo di imbardata e v la velocità longitudinale nel sistema veicolo, mantenuta come componente di stato per robustezza rispetto a variazioni lente della velocità.

Gli ingressi disponibili sono la velocità angolare di imbardata $\dot{\psi}$ dall'IMU e la velocità v dagli encoder. Con periodo di campionamento Δt , la funzione di transizione discreta è

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + \Delta t v_k \cos \psi_k \\ y_k + \Delta t v_k \sin \psi_k \\ \psi_k + \Delta t \dot{\psi}_k \\ v_k \end{bmatrix} + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, Q_k). \quad (4.37)$$

4.4.2. EKF: jacobiano del modello

La propagazione della covarianza avviene mediante il jacobiano della transizione rispetto allo stato:

$$F_k = \begin{bmatrix} 1 & 0 & -\Delta t v_k \sin \psi_k & \Delta t \cos \psi_k \\ 0 & 1 & \Delta t v_k \cos \psi_k & \Delta t \sin \psi_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.38)$$

La covarianza predetta è $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$. La scelta di Q_k riflette l'incertezza degli ingressi inerziali e odometrici; una parametrizzazione pratica diagonale è

$$Q_k = \text{diag}(0, 0, \Delta t^2 \sigma_\psi^2, \sigma_v^2), \quad (4.39)$$

dove σ_ψ^2 deriva dal giroscopio e σ_v^2 è il rumore di processo associato alla velocità longitudinale.

4.4.3. UKF: propagazione unscented

Nel caso UKF, la non linearità della transizione è gestita tramite punti sigma generati da $(\hat{\mathbf{x}}_{k-1|k-1}, P_{k-1|k-1})$ e propagati con la stessa funzione di transizione. Media e covarianza predette si ottengono per combinazione pesata dei punti trasformati, con l'aggiunta di Q_k . Non sono richiesti jacobiani; valgono i medesimi ingressi e la stessa definizione di Q_k . È raccomandata la normalizzazione dell'angolo ψ dopo ogni predizione e aggiornamento, riportandolo in $(-\pi, \pi]$.

4.4.4. Modello di osservazione e aggiornamento

Le misure assolute fornite da UWB sono posizioni nel riferimento globale. Con la stessa nomenclatura adottata in precedenza, si pone

$$\mathbf{y}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, R_k), \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (4.40)$$

con $R_k = \text{diag}(\sigma_{p,x}^2, \sigma_{p,y}^2)$. L'aggiornamento segue le equazioni standard dei filtri gaussiani; nell'UKF si effettua la trasformazione dei punti sigma nello spazio osservazioni, con calcolo della covarianza innovativa e del guadagno.

4.4.5. Accorgimenti numerici e strategie di filtraggio

Per garantire stabilità numerica, coerenza statistica e affidabilità delle stime, sono adottati i seguenti accorgimenti operativi:

Capitolo 4. Localizzazione locale

- Normalizzazione angolare: l'angolo di imbardata ψ è riportato in $(-\pi, \pi]$ dopo ogni operazione di predizione e aggiornamento, evitando discontinuità dovute al wrapping.
- Sincronizzazione temporale: i dati di IMU ed encoder sono interpolati su un passo costante $\Delta t = 0.01$ s (100 Hz), garantendo uniformità nella propagazione dello stato e nella generazione dei punti sigma per l'UKF. Le misure assolute (UWB) sono acquisite a frequenza inferiore (10 Hz) e utilizzate nella fase di aggiornamento.
- Adattamento numerico del modello: pur mantenendo un passo di predizione fisso (100 Hz), il filtro può modificare internamente la discretizzazione del modello in presenza di dinamiche rapide, al fine di preservare la stabilità dell'integrazione.
- Rilevamento e rimozione delle misure anomale: un test di gating basato sulla distanza del chi-quadrato consente di identificare e scartare misure anomale, prevenendo la perdita di consistenza del filtro.

L'adozione congiunta di tali accorgimenti consente di mantenere stabilità numerica e coerenza statistica anche in presenza di rumore non ideale o misure corrotte. L'impiego del test del chi-quadrato incrementa la robustezza della stima rispetto a misure anomale dovuti a riflessioni multipath o perdite di sincronizzazione UWB. L'approccio illustrato in Figura 4.5 garantisce una stima continua, stabile e coerente della posa del veicolo, combinando in tempo reale la predizione ad alta frequenza basata su sensori relativi (IMU, encoder) con la correzione a bassa frequenza fornita dai sensori di posizionamento assoluto (UWB).

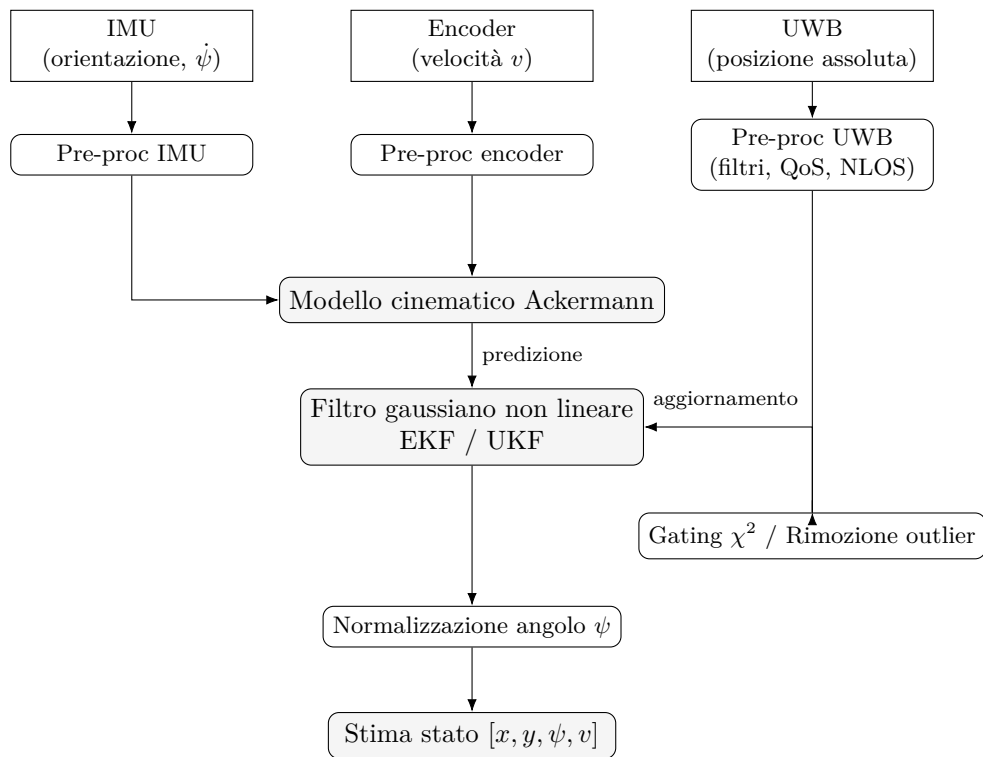


Figura 4.5.: Schema del sistema di fusione sensoriale.

Capitolo 5.

Localizzazione globale

5.1. Localizzazione globale

Nel capitolo precedente è stata implementata la localizzazione locale, ottenuta mediante tecniche di fusione sensoriale tra dati provenienti da sensori inerziali, encoder e sistema UWB, con l'obiettivo di stimare in modo continuo e relativo la posa del veicolo nel tempo. Tale approccio consente di ottenere una stima locale accurata dello stato, ma risulta soggetto all'accumulo progressivo dell'errore e alla deriva temporale.

In questo capitolo viene introdotto il livello superiore della localizzazione, definito come localizzazione globale, in cui le stime odometriche vengono integrate con informazioni ambientali derivate da sensori esteroceettivi e da mappe di riferimento. Questo passaggio costituisce il collegamento teorico tra la stima basata esclusivamente su sensori e quella supportata da mappe, consentendo di ridurre l'incertezza accumulata e di ottenere una rappresentazione coerente e assoluta della posa del veicolo nello spazio.

In tale contesto, il sottosistema di percezione assume un ruolo centrale. Attraverso l'elaborazione dei dati sensoriali, in particolare delle misurazioni fornite dal LiDAR, il sistema è in grado di estrarre le caratteristiche geometriche dell'ambiente circostante, utilizzandole come riferimenti metrici per la correzione e la stabilizzazione della stima odometrica.

La presente sezione fornisce la base teorica per l'implementazione sperimentale della localizzazione globale del veicolo, realizzata mediante l'integrazione del pacchetto Adaptive Monte Carlo Localization (AMCL) con mappe generate attraverso algoritmi di Simultaneous Localization and Mapping (SLAM). Tale approccio consente di estendere la capacità percettiva del sistema, migliorando la robustezza e la precisione della localizzazione assoluta in ambienti complessi o parzialmente conosciuti.

Il capitolo approfondisce i principi teorici e metodologici alla base della localizzazione globale. Nella prima parte viene introdotto il concetto di percezione robotica, con riferimento ai criteri di acquisizione ed elaborazione dei dati per la rappresentazione spaziale dell'ambiente. Successivamente, viene analizzato il processo di mappatura, con riferimento ai metodi grid-based e alle tecniche di costruzione di mappe di occupazione, comunemente impiegate nei moderni sistemi di SLAM. Infine, viene

descritto un modello probabilistico di localizzazione basato su filtri a particelle, che consente di stimare in modo coerente e robusto la posa del veicolo all'interno di mappe note o parzialmente osservabili.

5.2. Percezione e rappresentazione spaziale

La percezione robotica costituisce il processo mediante il quale un sistema autonomo acquisisce, elabora e interpreta i dati provenienti dai sensori di bordo, al fine di costruire una rappresentazione coerente dell'ambiente operativo. Essa rappresenta l'interfaccia tra il mondo fisico e lo stato interno del veicolo, traducendo segnali grezzi in informazione spaziale strutturata, utile ai processi di localizzazione, mappatura e controllo.

La percezione svolge una funzione di collegamento tra la fusione sensoriale locale e la localizzazione basata su mappa. Attraverso l'elaborazione dei dati, le misure acquisite dai sensori vengono trasformate in entità geometriche e probabilistiche che descrivono l'ambiente circostante in una forma compatibile con i moduli di pianificazione e navigazione.

Ogni sensore fornisce una descrizione parziale e soggetta a incertezza. Il legame tra misura, posa e ambiente è descritto dalla relazione probabilistica

$$p(z_t \mid x_t, m),$$

dove z_t indica la misura acquisita al tempo t , x_t rappresenta la posa del veicolo e m la mappa dell'ambiente. Tale relazione definisce la probabilità di osservare un determinato valore di misura dato lo stato del veicolo e la configurazione dello spazio circostante.

La localizzazione locale, trattata nel capitolo precedente, fornisce una stima relativa della posa del veicolo nel tempo, modellata dalla distribuzione

$$p(x_t \mid x_{t-1}, u_t),$$

che costituisce la prior del modello di stato. La localizzazione globale integra invece questa informazione con la componente percettiva rappresentata dalla verosimiglianza (likelihood field) $p(z_t \mid x_t, m)$, ancorando la stima alla mappa di riferimento.

La combinazione tra prior e verosimiglianza consente di ottenere la distribuzione a posteriori dello stato:

$$p(x_t \mid z_{1:t}, u_{1:t}, m) \propto p(z_t \mid x_t, m) p(x_t \mid x_{t-1}, u_t),$$

che costituisce il principio fondamentale dei metodi SLAM e la base concettuale per gli algoritmi AMCL.

5.2.1. Mappatura a griglia di occupazione

La mappatura a griglia di occupazione rappresenta un metodo probabilistico per la costruzione di mappe metriche a partire da misure sensoriali. Lo spazio bidimensionale viene suddiviso in un insieme discreto di celle $\{m_i\}$, ciascuna caratterizzata dalla probabilità di essere occupata o libera. Il problema consiste nello stimare, per ogni cella, la probabilità condizionata alle osservazioni e alle pose note del veicolo:

$$p(m_i \mid z_{1:t}, x_{1:t}).$$

Assumendo indipendenza condizionata tra le celle, la distribuzione congiunta può essere approssimata come prodotto dei contributi individuali:

$$p(m \mid z_{1:t}, x_{1:t}) \approx \prod_i p(m_i \mid z_{1:t}, x_{1:t}),$$

consentendo l'aggiornamento indipendente di ciascuna cella tramite un filtro bayesiano binario.

Nel caso di traiettorie note, il processo di mappatura può essere descritto come un modello grafico statico in cui la mappa m influenza tutte le osservazioni z_t , mentre le pose x_t evolvono nel tempo secondo il modello cinematico del veicolo. Ogni misura dipende unicamente dallo stato corrente e dalla struttura dell'ambiente, come rappresentato in Figura 5.1.

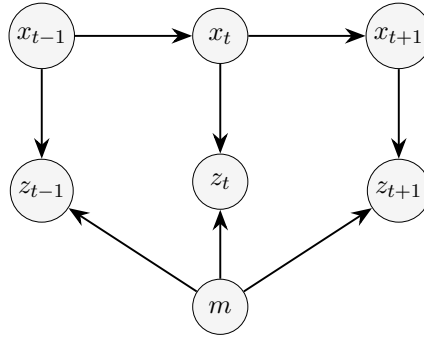


Figura 5.1.: Modello grafico per la mappatura con pose note.

Per evitare instabilità numeriche e semplificare l'aggiornamento, la probabilità di occupazione viene espressa nella forma logaritmica:

$$l_{t,i} = \log \frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})}.$$

In tale formulazione, l'aggiornamento bayesiano si riduce a una somma additiva.

La conversione inversa è data da:

$$p(m_i \mid z_{1:t}, x_{1:t}) = \frac{1}{1 + \exp(-l_{t,i})}.$$

Valori positivi di $l_{t,i}$ indicano evidenza di occupazione, mentre valori negativi rappresentano celle libere.

A ogni nuova acquisizione sensoriale, la mappa viene aggiornata iterativamente. Per ciascuna cella m_i , si verifica se essa rientra nel campo visivo del sensore; in tal caso, il valore log-odds viene corretto in base al contributo calcolato tramite il modello sensore inverso. Quest'ultimo determina la variazione di credenza in funzione della geometria sensore-cella e della misura rilevata. Il procedimento complessivo è riportato di seguito.

Algorithm 4 Aggiornamento della mappa di occupazione grid-based

```

1: Ingresso:  $\{l_{t-1,i}\}$  (log-odds), posa  $x_t$ , misura  $z_t$ 
2: for ogni cella  $m_i$  do
3:   if  $m_i$  rientra nel campo visivo then
4:     calcola le coordinate  $(x_i, y_i)$  della cella
5:     determina: la distanza  $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$  e l'angolo relativo
        $\phi = \text{atan2}(y_i - y, x_i - x) - \psi$ 
6:     individua il fascio  $k$  con direzione  $\theta_k$  più prossima a  $\phi$ 
7:     if  $|\phi - \theta_k| > \beta/2$  oppure  $r > \min(z_{\max}, z_t^k + \alpha/2)$  then
8:        $l_{\text{agg}} = l_0$  (cella non osservata)
9:     else if  $z_t^k < z_{\max}$  e  $|r - z_t^k| < \alpha/2$  then
10:       $l_{\text{agg}} = l_{\text{occ}}$  (cella occupata)
11:    else if  $r < z_t^k$  then
12:       $l_{\text{agg}} = l_{\text{free}}$  (cella libera)
13:    else
14:       $l_{\text{agg}} = l_0$ 
15:    end if
16:    aggiorna  $l_{t,i} = l_{t-1,i} + l_{\text{agg}} - l_0$ 
17:  else
18:     $l_{t,i} = l_{t-1,i}$ 
19:  end if
20: end for
21: Uscita: nuova mappa  $\{l_{t,i}\}$ 

```

I parametri α e β definiscono rispettivamente la tolleranza radiale e l'ampiezza angolare del fascio. I valori l_{occ} e l_{free} determinano l'intensità dell'aggiornamento per celle occupate o libere, mentre $l_0 = \log \frac{p(m_i)}{1-p(m_i)}$ rappresenta il valore iniziale di credenza, solitamente nullo per indicare incertezza uniforme.

L'integrazione nel tempo delle osservazioni provenienti da pose diverse consente di

ottenere una stima progressivamente più accurata della distribuzione di occupazione. L'approccio a griglia fornisce una rappresentazione metrica coerente e statisticamente consistente dell'ambiente, costituendo la base per sistemi di navigazione e pianificazione autonoma.

Quando la posa del veicolo non è nota con certezza, la mappatura deve essere eseguita congiuntamente alla stima della traiettoria. Tale formulazione conduce al problema dello SLAM, trattato nelle sezioni successive.

5.3. Simultaneous Localization and Mapping (SLAM)

Le tecniche di fusione sensoriale descritte nei capitoli precedenti consentono di stimare la posa del veicolo combinando modelli cinematici e misure propriocettive. Tali stime risultano accurate in termini relativi, ma soggette a deriva cumulativa nel tempo. Per ridurre tale effetto, si introducono sensori esteroceettivi in grado di fornire riferimenti assoluti e di ancorare la stima alla struttura dell'ambiente.

Il problema di SLAM consiste nella stima congiunta della traiettoria del veicolo e della mappa dell'ambiente a partire da osservazioni sensoriali e comandi di controllo, superando i limiti dei metodi puramente odometrici. A differenza della localizzazione, che presuppone una mappa nota, o della mappatura, che assume pose note, nel SLAM entrambe le variabili sono inizialmente ignote e vengono determinate simultaneamente tramite inferenza probabilistica.

Formalmente, l'obiettivo è calcolare la distribuzione a posteriori:

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t})$$

dove:

- $x_{1:t} = \{x_1, x_2, \dots, x_t\}$ è la sequenza delle pose del veicolo fino all'istante t ;
- m rappresenta la mappa dell'ambiente;
- $z_{1:t}$ raccoglie le osservazioni sensoriali acquisite;
- $u_{1:t}$ indica i comandi di controllo applicati.

Questa distribuzione esprime congiuntamente l'incertezza sulla traiettoria e sulla struttura dell'ambiente. La sua stima esatta risulta tuttavia impraticabile, a causa della dipendenza temporale delle osservazioni e delle correlazioni tra le variabili di stato. Di conseguenza, si adottano approssimazioni probabilistiche che permettono di stimare la distribuzione in modo incrementale o globale, bilanciando accuratezza e complessità computazionale.

I metodi di tipo filtrante, come l'EKF-SLAM e il FastSLAM, aggiornano ricorsivamente la stima dello stato e della mappa mediante un ciclo predittivo-correttivo. Questi algoritmi operano in tempo reale, ma la necessità di propagare le covarianze

tra tutte le variabili comporta una complessità quadratica rispetto al numero di punti di riferimento (landmark). Inoltre, la linearizzazione dei modelli introduce approssimazioni che possono compromettere la consistenza statistica delle stime.

Per ovviare a tali limiti, sono stati sviluppati approcci globali basati su ottimizzazione, che rappresentano il problema come un insieme di vincoli geometrici tra le pose del veicolo e le osservazioni ambientali. In questo quadro, ogni misura definisce una relazione spaziale tra variabili di stato, e la soluzione corrisponde alla configurazione di pose e landmark che minimizza l'errore complessivo sui vincoli.

Questa impostazione porta naturalmente alla formulazione a grafo del problema SLAM, nella quale le variabili di stato e le relazioni osservabili vengono organizzate in una struttura sparsa di vincoli. La soluzione è ottenuta attraverso tecniche di ottimizzazione non lineare, che permettono di correggere retrospettivamente la traiettoria e di mantenere coerenza geometrica anche in presenza di chiusure di ciclo.

5.3.1. GraphSLAM

GraphSLAM fornisce una rappresentazione globale e compatta del problema SLAM, in cui la sequenza delle pose e le misure osservate vengono organizzate in un grafo di vincoli. In tale modello:

- i nodi corrispondono alle pose del veicolo e alle feature ambientali;
- gli archi esprimono i vincoli probabilistici derivanti dai modelli di moto e dalle osservazioni sensoriali.

Il problema viene quindi formulato come un'ottimizzazione non lineare che mira a determinare la configurazione di stato più coerente con le misure acquisite. La struttura sparsa del grafo, dovuta al fatto che ciascun vincolo coinvolge un numero limitato di variabili, consente l'impiego di tecniche numeriche efficienti basate su decomposizioni sparse o strategie incrementalì.

A differenza dei metodi filtranti, GraphSLAM non aggiorna progressivamente una singola stima, ma accumula vincoli locali risolvendoli globalmente. Questo approccio consente di correggere retrospettivamente gli errori di traiettoria e di generare mappe globalmente consistenti, mantenendo al tempo stesso efficienza e stabilità numerica in scenari di ampia scala.

La formulazione a grafo rappresenta quindi un paradigma scalabile e informativamente coerente per la risoluzione del problema SLAM, oggi ampiamente adottato nei sistemi di navigazione autonoma.

Costruzione del grafo

La costruzione del grafo rappresenta la fase iniziale dell'algoritmo GraphSLAM, in cui le informazioni di moto e di misura vengono convertite in una struttura di vincoli tra variabili di stato. A partire dai dati acquisiti $z_{1:t}$ e dai comandi di controllo $u_{1:t}$,

il sistema genera un grafo informativo che descrive la relazione tra le pose del veicolo e gli elementi rilevati dell'ambiente.

In tale rappresentazione, i nodi del grafo corrispondono alle pose x_t del veicolo e, a seconda della formulazione adottata, ai landmark m_j osservati. Gli archi modellano i vincoli probabilistici derivanti dai modelli di moto e di osservazione, ciascuno associato a una misura dell'incertezza tramite la relativa matrice di covarianza. I vincoli di moto collegano pose consecutive, mentre i vincoli di misura connettono la posa corrente alle caratteristiche ambientali rilevate.

Un vincolo di osservazione definisce una relazione tra la posa x_t e il landmark m_j misurato al tempo t :

$$(z_{t,i} - h(x_t, m_j))^T Q_t^{-1} (z_{t,i} - h(x_t, m_j)),$$

dove $h(\cdot)$ è il modello di misura e Q_t la covarianza del rumore sensoriale. Analogamente, un vincolo di moto collega due pose successive:

$$(x_t - g(u_t, x_{t-1}))^T R_t^{-1} (x_t - g(u_t, x_{t-1})),$$

con $g(\cdot)$ modello di moto e R_t covarianza del rumore cinematico. In entrambi i casi, il termine quadratico rappresenta la deviazione tra la misura effettiva e quella prevista dal modello, pesata in funzione dell'incertezza associata.

Le informazioni contenute nei vincoli vengono aggregate in una rappresentazione matriciale compatta mediante la matrice di informazione Ω e il vettore di informazione ξ :

$$\Omega = \sum_k H_k^T Q_k^{-1} H_k, \quad \xi = \sum_k H_k^T Q_k^{-1} e_k,$$

dove H_k è la jacobiana del vincolo e e_k l'errore residuo associato. Ogni osservazione o comando di controllo produce un contributo locale a Ω e ξ , che vengono aggiornati secondo un principio additivo dell'informazione.

Dopo l'integrazione di tutte le misure e dei comandi $z_{1:t}$ e $u_{1:t}$, si ottiene un sistema lineare sparso che descrive la struttura globale del problema. La formulazione complessiva del costo può essere espressa come:

$$\begin{aligned} J_{\text{GraphSLAM}} = & x_0^T \Omega_0 x_0 + \sum_t (x_t - g(u_t, x_{t-1}))^T R_t^{-1} (x_t - g(u_t, x_{t-1})) \\ & + \sum_t \sum_i (z_{t,i} - h(x_t, m_{c_{t,i}}))^T Q_t^{-1} (z_{t,i} - h(x_t, m_{c_{t,i}})), \end{aligned} \quad (5.1)$$

dove il termine $x_0^T \Omega_0 x_0$ costituisce il vincolo di ancoraggio, necessario a fissare la posa iniziale come riferimento del sistema di coordinate.

La matrice di informazione risultante presenta una struttura fortemente sparsa: i termini non nulli si concentrano nei blocchi corrispondenti alle coppie di pose connesse da vincoli di moto e alle relazioni tra pose e landmark osservati. Non sono invece presenti termini diretti di correlazione tra landmark, poiché non vengono

acquisite misure che ne leghino esplicitamente le posizioni relative. Questa proprietà di sparsenza costituisce un aspetto essenziale dell'approccio GraphSLAM, poiché consente l'utilizzo di algoritmi di ottimizzazione numerica efficienti e la scalabilità del metodo in ambienti di grandi dimensioni.

Inferenza e fattorizzazione dell'informazione

Dopo la costruzione del grafo dei vincoli, la fase di inferenza mira a determinare la configurazione di stato che massimizza la coerenza globale tra modelli di moto, osservazioni e variabili di stato. Nel formalismo informativo di GraphSLAM, la conoscenza del sistema è rappresentata dalla matrice di informazione Ω e dal vettore di informazione ξ , dai quali si ricava la stima ottima dello stato risolvendo il sistema lineare:

$$\mu = \Omega^{-1}\xi.$$

Tale relazione fornisce la media della distribuzione a posteriori, ma l'inversione diretta della matrice Ω comporta un costo computazionale elevato nei sistemi di grandi dimensioni. Per questo motivo si adottano tecniche di fattorizzazione e riduzione, volte a migliorare l'efficienza numerica preservando l'equivalenza statistica del sistema.

La complessità dell'inferenza dipende dalla topologia del grafo e dal grado di connettività dei vincoli. In configurazioni quasi lineari, prive di chiusure di ciclo, la matrice Ω può essere ordinata in forma bandata e risolta con complessità proporzionale al numero di variabili. In presenza di cicli o osservazioni ripetute, la matrice tende invece a densificarsi, rendendo inefficiente l'inversione diretta. In tali casi si impiegano metodi iterativi di ottimizzazione, come il gradiente coniugato o Gauss-Newton, che consentono di calcolare il prodotto $\Omega^{-1}\xi$ senza eseguire un'inversione esplicita.

Un ulteriore miglioramento dell'efficienza si ottiene tramite la riduzione progressiva delle variabili associate ai landmark. L'obiettivo è eliminare i landmark dal sistema trasferendo le loro informazioni alle pose che li hanno osservati, ottenendo un grafo equivalente ma più compatto. Formalmente, se un landmark m_j è stato rilevato nelle pose appartenenti all'insieme $\tau(j) = \{x_t \mid m_j \text{ osservato in } t\}$, la sua eliminazione introduce nuovi vincoli equivalenti tra tutte le coppie di pose $x_t, x_{t'} \in \tau(j)$. In questo modo, l'informazione originariamente associata al landmark viene propagata tra le pose, riducendo il numero di variabili mantenendo invariata la consistenza statistica. Il processo di riduzione produce una matrice di informazione compatta $\tilde{\Omega}$ e un vettore $\tilde{\xi}$, definiti unicamente sulle variabili di posa. Il sistema ridotto conserva le relazioni cicliche del grafo originale, ma risulta più trattabile mediante tecniche di ottimizzazione sparse. La configurazione delle pose si ottiene risolvendo:

$$\tilde{\mu} = \tilde{\Omega}^{-1}\tilde{\xi}.$$

Successivamente, le posizioni dei landmark vengono determinate risolvendo sotto-

sistemi locali (Ω_j, ξ_j) , che coinvolgono ciascun landmark e le sole pose osservanti. Queste operazioni, di dimensione ridotta, permettono una ricostruzione efficiente e coerente delle coordinate dei landmark.

Questa strategia consente di gestire in modo efficiente problemi di larga scala, garantendo coerenza globale tra mappa e traiettoria e preservando la struttura sparsa del sistema, caratteristica fondamentale per la scalabilità del metodo GraphSLAM.

Algoritmo GraphSLAM

L'algoritmo GraphSLAM realizza la stima congiunta della traiettoria del veicolo e della mappa dell'ambiente rappresentando il problema SLAM come un grafo di vincoli probabilistici tra pose e landmark. Ogni comando di controllo e ogni osservazione generano relazioni locali tra le variabili di stato, aggiornando la matrice di informazione Ω e il vettore ξ , che descrivono in forma compatta la distribuzione a posteriori del sistema. Le principali fasi operative di inizializzazione, linearizzazione, riduzione, risoluzione e associazione dei dati traducono a livello algoritmico i processi di costruzione del grafo e inferenza informativa illustrati nelle sezioni precedenti.

Algorithm 5 GraphSLAM($u_{1:t}, z_{1:t}$)

```

1: Inizializza tutte le corrispondenze  $c_{1:t}$  con un valore univoco
2:  $\mu_{0:t} = \text{INIZIALIZZAZIONE}(u_{1:t})$ 
3:  $\Omega, \xi = \text{LINEARIZZAZIONE}(u_{1:t}, z_{1:t}, c_{1:t}, \mu_{0:t})$ 
4:  $\tilde{\Omega}, \tilde{\xi} = \text{RIDUZIONE}(\Omega, \xi)$ 
5:  $\mu, \Sigma_{0:t} = \text{RISOLUZIONE}(\tilde{\Omega}, \tilde{\xi})$ 
6: repeat
7:   for ogni coppia di landmark non ancora accoppiati  $(m_j, m_k)$  do
8:      $\pi_{j=k} = \text{TESTCORRISPONDENZA}(\Omega, \xi, \mu, \Sigma_{0:t}, j, k)$ 
9:     if  $\pi_{j=k} > \chi$  then
10:      per ogni  $c_i = k$ , imposta  $c_i = j$ 
11:       $\Omega, \xi = \text{LINEARIZZAZIONE}(u_{1:t}, z_{1:t}, c_{1:t}, \mu_{0:t})$ 
12:       $\tilde{\Omega}, \tilde{\xi} = \text{RIDUZIONE}(\Omega, \xi)$ 
13:       $\mu, \Sigma_{0:t} = \text{RISOLUZIONE}(\tilde{\Omega}, \tilde{\xi})$ 
14:    end if
15:  end for
16: until nessuna nuova coppia  $(m_j, m_k)$  soddisfa  $\pi_{j=k} > \chi$ 
17: return  $\mu$ 

```

Inizializzazione

Si genera una prima stima della traiettoria concatenando le trasformazioni cinematiche associate ai comandi di controllo $u_{1:t}$. La sequenza di pose $\mu_{0:t}$ costituisce il punto di partenza per la linearizzazione e l'ottimizzazione successiva.

Linearizzazione e aggiornamento informativo

I modelli di transizione $p(x_t | u_t, x_{t-1})$ e di misura $p(z_t | x_t, m)$ vengono linearizzati mediante espansione di Taylor attorno alle stime correnti. Ciascun vincolo di moto o di misura contribuisce all'aggiornamento locale di Ω e ξ , mantenendo la struttura sparsa del sistema e consentendo una rappresentazione matriciale efficiente delle dipendenze tra variabili.

Riduzione e sistema compatto

Le variabili dei landmark sono progressivamente eliminate applicando il complemento di Schur, trasferendo la loro informazione alle pose che li hanno osservati. Il risultato è un sistema ridotto $(\tilde{\Omega}, \tilde{\xi})$ definito unicamente sulle pose, statisticamente equivalente a quello completo ma più trattabile dal punto di vista numerico.

Risoluzione del sistema ridotto

Il sistema sparso $\tilde{\Omega}\tilde{\mu} = \tilde{\xi}$ viene risolto tramite fattorizzazioni sparse (ad esempio Cholesky o QR) o metodi iterativi preconditionati. La soluzione $\tilde{\mu}$ rappresenta la configurazione di pose che minimizza la funzione di costo del problema GraphSLAM, equivalente alla massimizzazione della distribuzione a posteriori $p(x_{0:t}, m | z_{1:t}, u_{1:t})$. Le posizioni dei landmark vengono poi determinate risolvendo sotto-sistemi locali ristretti alle pose osservanti.

Associazione dei dati

Nel caso in cui le corrispondenze tra osservazioni e landmark non siano note, si esegue una procedura iterativa di associazione basata su misure di coerenza statistica. Per ogni coppia di landmark (m_j, m_k) si calcola la probabilità di equivalenza

$$\pi_{j=k} = p(m_j \equiv m_k | \Omega, \xi, \mu, \Sigma_{0:t}),$$

stimata tramite la distanza di Mahalanobis. Le coppie con $\pi_{j=k} > \chi$ vengono fuse e il sistema informativo viene aggiornato. Il processo si ripete fino al raggiungimento di stabilità, con l'impiego di strategie di gating spaziale e aggiornamenti in batch per limitare il costo computazionale.

Convergenza e stabilità numerica

L'algoritmo itera le fasi di linearizzazione, riduzione e risoluzione fino alla stabilizzazione del funzionale di costo. La convergenza è garantita da una stima iniziale coerente e da accorgimenti numerici quali riordinamento delle variabili (AMD, COLAMD), preconditionamento e criteri di arresto basati sulla variazione relativa del costo. Queste tecniche assicurano stabilità, scalabilità e coerenza globale della soluzione finale.

Efficienza e implementazioni pratiche

Per ridurre la complessità computazionale, le implementazioni operative adottano strategie di sottomappatura: le osservazioni vengono aggregate in mappe locali, ottimizzate indipendentemente e successivamente fuse in modo gerarchico. Nel caso di sensori LiDAR, si assume mutua esclusione tra le misure di una stessa scansione per semplificare l'associazione dei dati. Versioni avanzate includono inoltre la gestione di corrispondenze errate e l'integrazione dell'informazione negativa (assenza di rilevamento atteso), migliorando la consistenza statistica complessiva.

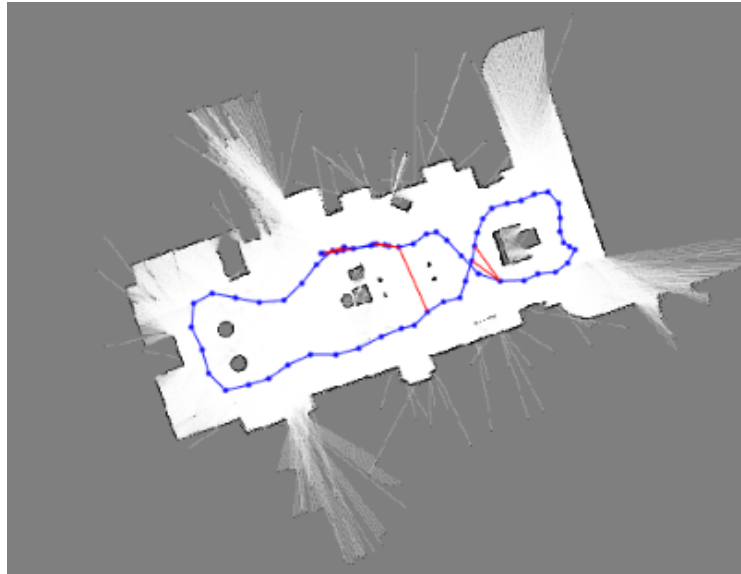


Figura 5.2.: Esempio di risultato del processo di GraphSLAM.

Completata la fase di ottimizzazione, la mappa risultante viene consolidata come riferimento globale per la localizzazione. In questa fase, il sistema non aggiorna più la mappa ma utilizza le osservazioni correnti per correggere in tempo reale la stima odometrica, ancorando la posa del veicolo alla struttura ambientale. La stima probabilistica della posa viene quindi affidata all'algoritmo AMCL, che integra dati sensoriali e odometrici in un modello basato su campioni, garantendo robustezza e adattività nella localizzazione globale. La sezione seguente ne illustra i principi e le principali fasi operative.

5.4. Adaptive Monte Carlo Localization (AMCL)

L'AMCL rappresenta una formulazione avanzata della localizzazione probabilistica basata su campioni, utilizzata per stimare la posa di un veicolo autonomo all'interno di una mappa nota. L'AMCL assume come riferimento la mappa generata nella fase di SLAM e ne sfrutta la coerenza metrica per realizzare una localizzazione assoluta in tempo reale, compensando la deriva odometrica. L'algoritmo deriva dal paradigma

della Markov Localization, di cui costituisce un'estensione efficiente e adattiva, capace di mantenere un'elevata accuratezza con un numero variabile di ipotesi di stato.

Il principio generale si fonda sulla rappresentazione della conoscenza del sistema come distribuzione di probabilità della posa, aggiornata in modo ricorsivo sulla base dei modelli di movimento e delle osservazioni sensoriali. Tale distribuzione viene approssimata mediante un insieme discreto di particelle, che ne consente la propagazione e l'aggiornamento anche in presenza di modelli non lineari o di distribuzioni multimodali.

5.4.1. Formalismo probabilistico

Nel contesto della localizzazione markoviana, la stima della posa del veicolo al tempo t è descritta dalla distribuzione a posteriori:

$$p(x_t \mid z_{1:t}, u_{1:t}, m)$$

dove x_t rappresenta la posa, $z_{1:t}$ l'insieme delle osservazioni fino all'istante t , $u_{1:t}$ la sequenza dei comandi di controllo e m la mappa di occupazione.

L'aggiornamento segue la formulazione ricorsiva del filtro di Bayes:

$$p(x_t \mid z_{1:t}, u_{1:t}, m) = \eta p(z_t \mid x_t, m) \int p(x_t \mid u_t, x_{t-1}) p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1}, m) dx_{t-1}$$

dove $p(x_t \mid u_t, x_{t-1})$ è il modello di transizione del moto, $p(z_t \mid x_t, m)$ il modello sensoriale, e η un fattore di normalizzazione. La Markov Localization tradizionale calcola questa distribuzione su una griglia regolare, con costo computazionale crescente al crescere della risoluzione. L'AMCL sostituisce tale rappresentazione densa con un approccio stocastico a campioni, che concentra le risorse computazionali nelle regioni a maggiore probabilità.

5.4.2. Filtro a campioni

La distribuzione a posteriori viene approssimata da un insieme di N particelle $\{x_t^{[i]}, w_t^{[i]}\}_{i=1}^N$, ciascuna caratterizzata da una posa ipotetica $x_t^{[i]}$ e da un peso $w_t^{[i]}$ proporzionale alla probabilità relativa di quella configurazione. L'algoritmo procede iterativamente secondo tre fasi principali.

Predizione

$$x_t^{[i]} \sim p(x_t \mid u_t, x_{t-1}^{[i]})$$

La predizione propaga la distribuzione nel tempo secondo il modello di moto, introducendo l'incertezza odometrica.

Correzione

$$w_t^{[i]} = p(z_t \mid x_t^{[i]}, m)$$

I pesi vengono aggiornati in funzione della coerenza tra le osservazioni sensoriali e la mappa, attribuendo maggior peso alle particelle compatibili con le misure acquisite.

Rinormalizzazione e campionamento I pesi normalizzati generano un nuovo insieme di particelle proporzionale alla distribuzione stimata. Le ipotesi più probabili vengono replicate, mentre quelle meno coerenti vengono scartate. In tal modo la distribuzione tende progressivamente a concentrarsi attorno alla posa reale del veicolo.

5.4.3. Adattività del numero di particelle

Una criticità dei filtri Monte Carlo riguarda la scelta del numero di particelle, che influenza il compromesso tra accuratezza e costo computazionale. L'AMCL impiega un meccanismo di adattività basato sulla divergenza di Kullback–Leibler (KLD-sampling), che regola dinamicamente il numero di campioni richiesto per mantenere l'errore di approssimazione entro una soglia prefissata ϵ con livello di confidenza $1 - \delta$:

$$N_t \geq \frac{1}{2\epsilon} \left(k - 1 + \ln \frac{1}{\delta} \right)$$

dove k rappresenta il numero di celle occupate nello spazio di stato. Durante le fasi di incertezza, come l'inizializzazione o la perdita temporanea della localizzazione, N_t aumenta automaticamente; al contrario, in condizioni stabili, il numero di campioni si riduce senza perdita di accuratezza.

5.4.4. Modelli probabilistici

Il modello di moto $p(x_t | u_t, x_{t-1})$ descrive la transizione di stato indotta dai comandi odometrici. Nel caso di veicolo con geometria Ackermann:

$$\begin{aligned} x_t &= x_{t-1} + \Delta s \cos(\psi_{t-1} + \frac{\Delta\psi}{2}) \\ y_t &= y_{t-1} + \Delta s \sin(\psi_{t-1} + \frac{\Delta\psi}{2}) \\ \psi_t &= \psi_{t-1} + \Delta\psi \end{aligned}$$

dove Δs rappresenta lo spostamento lineare e $\Delta\psi$ la variazione di orientamento dovuta alla sterzata.

Il modello di osservazione $p(z_t | x_t, m)$ quantifica la probabilità di ottenere una misura z_t in una posa x_t rispetto alla mappa. Nei sistemi basati su sensori LiDAR è adottato il modello di verosimiglianza (likelihood field), in cui ciascun raggio è confrontato con la distanza d_k tra il punto di impatto previsto e la cella occupata più vicina:

$$p(z_t | x_t, m) = \eta \sum_k \exp\left(-\frac{d_k^2}{2\sigma_{\text{hit}}^2}\right) + w_{\text{rand}},$$

dove σ_{hit} controlla la sensibilità alla deviazione spaziale e w_{rand} rappresenta la componente di rumore casuale.

Localizzazione in ambienti dinamici

In ambienti reali la presenza di oggetti mobili genera misure incoerenti con la mappa statica. Per attenuare tali effetti, i moderni filtri Monte Carlo applicano una procedura di esclusione dei raggi anomali, nota come beam skipping. I raggi per i quali la distanza osservata risulta discordante rispetto alla previsione vengono ignorati nel calcolo della verosimiglianza, riducendo l'impatto delle letture corrotte e mantenendo la stabilità della stima anche in contesti dinamici.

5.4.5. Algoritmo

Algorithm 6 Adaptive Monte Carlo Localization

Require: mappa m , modelli $p(x_t | u_t, x_{t-1})$ e $p(z_t | x_t, m)$, soglie ϵ, δ

Ensure: stima della posa $\hat{x}_t$

- 1: inizializza $\{x_0^{[i]}, w_0^{[i]}\}_{i=1}^{N_0}$ secondo la modalità operativa
 - 2: **for** ogni istante $t \dots$ **do**
 - 3: predizione: $x_t^{[i]} \sim p(x_t | u_t, x_{t-1}^{[i]})$
 - 4: correzione: $w_t^{[i]} = p(z_t | x_t^{[i]}, m)$
 - 5: normalizzazione: $w_t^{[i]} \leftarrow w_t^{[i]} / \sum_j w_t^{[j]}$
 - 6: calcolo adattivo: $N_t = \frac{1}{2\epsilon}(k - 1 + \ln \frac{1}{\delta})$
 - 7: resampling sistematico di N_t particelle proporzionale ai pesi $w_t^{[i]}$
 - 8: stima della posa: $\hat{x}_t = \sum_i w_t^{[i]} x_t^{[i]}$
 - 9: **end for**
-

Durante la navigazione, la combinazione tra la predizione odometrica e la correzione basata sulle osservazioni LiDAR consente al filtro di ridurre progressivamente l'incertezza sulla posa, mantenendo la distribuzione dei campioni coerente con la posizione reale del veicolo. L'aggiornamento sequenziale dei pesi e il ri-campionamento adattivo favoriscono la concentrazione della densità di probabilità nelle regioni a maggiore coerenza con la mappa, garantendo una convergenza stabile anche in presenza di rumore sensoriale o in ambienti con simmetrie geometriche.

L'algoritmo AMCL rappresenta il punto di chiusura del processo di localizzazione globale, integrando in modo coerente le informazioni odometriche e sensoriali con la mappa generata dal modulo SLAM. Attraverso questa integrazione, la posa stimata viene costantemente ancorata alla rappresentazione metrica dell'ambiente, compensando la deriva accumulata nelle fasi di esplorazione e migliorando la consistenza geometrica della traiettoria.

Il carattere adattivo del metodo, che regola dinamicamente il numero di particelle in funzione dell'incertezza stimata, consente di bilanciare precisione e efficienza computazionale. Nel complesso, l'AMCL realizza la transizione dal processo di mappatura probabilistica alla localizzazione assoluta, completando il ciclo di percezione del veicolo autonomo.

5.5. Pipeline globale

La catena di elaborazione si articola nei seguenti passaggi:

1. Fusione sensoriale: fusione di misure inerziali e odometriche per ottenere una stima relativa continua della posa e delle incertezze associate.
2. SLAM: costruzione e ottimizzazione della mappa di occupazione e della traiettoria globale mediante vincoli di moto e di misura.
3. AMCL: localizzazione assoluta su mappa nota tramite filtro a campioni, con predizione cinematica, aggiornamento rispetto alla mappa, beam skipping e resampling adattivo.

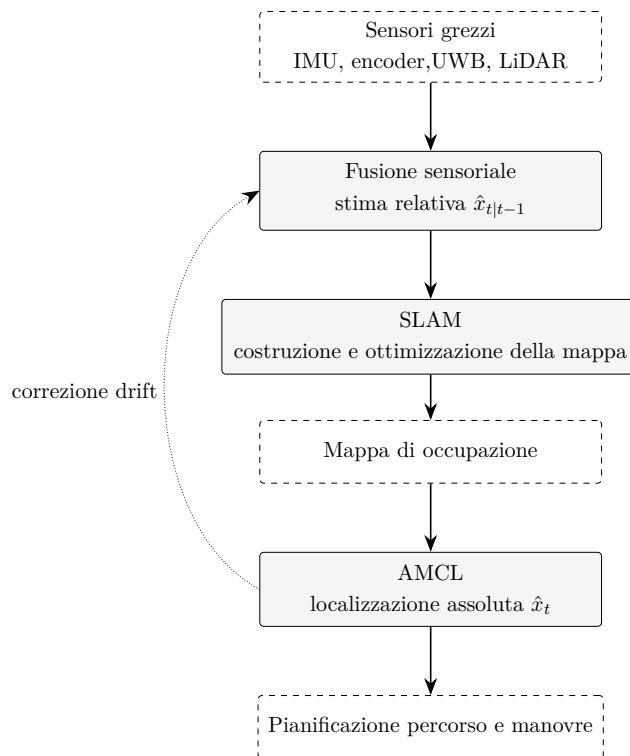


Figura 5.3.: Pipeline globale di localizzazione.

La pipeline sintetizza il flusso completo del sistema di localizzazione: la fusione sensoriale fornisce la base odometrica, il modulo SLAM costruisce la mappa di occupazione e la traiettoria globale, e l'AMCL utilizza tale mappa per stimare la posa assoluta del veicolo in tempo reale. L'insieme dei moduli descritti costituisce il livello percettivo dell'architettura di navigazione, responsabile della costruzione e del mantenimento della conoscenza spaziale necessaria alla pianificazione del moto autonomo.

Parte IV.

Implementazione

Capitolo 6.

Architettura ROS 2 Localizzazione

Il sistema di localizzazione si articola in una catena funzionale che collega il mondo reale all'attuazione. Le informazioni acquisite dai sensori vengono trasformate da misure fisiche in stime di stato utilizzate per il controllo del veicolo e per l'interazione con l'ambiente. Il flusso informativo si sviluppa attraverso tre livelli principali: percezione, elaborazione e controllo.

Nel livello di percezione, i sensori raccolgono dati relativi all'ambiente e al moto del veicolo. I dispositivi di visione forniscono immagini e profili spaziali, i sensori di distanza misurano le posizioni rispetto agli ostacoli, mentre le unità inerziali e gli encoder rilevano accelerazioni e velocità angolari. Le misure ottenute, affette da rumore e asincronie temporali, vengono filtrate e sincronizzate.

Nel livello di elaborazione, i dati sensoriali vengono combinati per ottenere stime coerenti della posa e dell'orientamento. Questa fase comprende la fusione delle informazioni provenienti da sensori diversi, la compensazione degli errori di misura e l'aggiornamento continuo dello stato stimato nel tempo. Il risultato è una rappresentazione consistente della traiettoria del veicolo nel proprio sistema di riferimento.

Nel livello di controllo e attuazione, le stime di stato vengono utilizzate per generare i comandi di movimento. Il sistema opera in retroazione: i comandi influenzano la dinamica del veicolo, la quale modifica a sua volta le misure sensoriali, chiudendo il ciclo di aggiornamento in tempo reale.

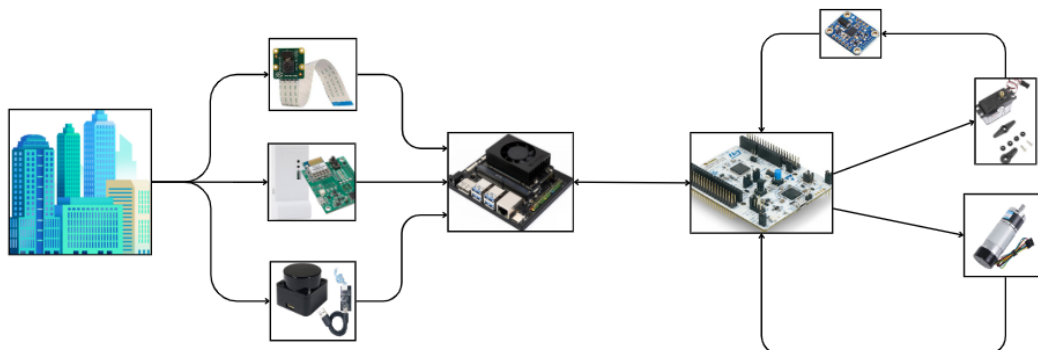


Figura 6.1.: Flusso dei dati dal mondo reale all'attuazione.

Nei capitoli successivi vengono discusse le implementazioni pratiche di questo flusso,

con riferimento ai meccanismi di comunicazione, sincronizzazione e gestione dei dati tra i dispositivi sensoriali, l'unità di calcolo principale e il microcontrollore. La piattaforma di calcolo NVIDIA Jetson Orin NX elabora le misure acquisite, esegue la fusione dei dati e calcola le stime di stato. Il microcontrollore STM32 gestisce l'acquisizione a bassa latenza dei segnali e la distribuzione dei comandi verso gli attuatori. La comunicazione tra le due unità è biunivoca: lo STM32 fornisce alla Jetson i dati derivanti da odometria, encoder e sensori inerziali, mentre la Jetson trasmette al microcontrollore la posa stimata e i comandi di controllo, successivamente convertiti in segnali di attuazione. Questo scambio continuo garantisce coerenza temporale e spaziale tra misure e comandi, assicurando la stabilità del comportamento dinamico del sistema.

6.1. Middleware ROS 2

ROS 2 (Robot Operating System 2) costituisce il middleware adottato per lo sviluppo e l'integrazione dei moduli funzionali del veicolo. Fornisce una piattaforma distribuita per la comunicazione tra processi eterogenei, garantendo interoperabilità, modularità e scalabilità. L'architettura si basa su nodi indipendenti che comunicano attraverso lo scambio di messaggi, utilizzando il protocollo DDS (Data Distribution Service) come livello di trasporto. Questo approccio consente l'esecuzione concorrente dei moduli e l'instradamento automatico dei dati tra produttori e consumatori senza vincoli di dipendenza diretta.

Il middleware offre diversi meccanismi di comunicazione che ne aumentano la flessibilità in contesti distribuiti:

- I topic implementano la comunicazione asincrona publish/subscribe, adatta alla trasmissione continua di dati ad alta frequenza tra nodi produttori e consumatori.
- I service consentono comunicazioni sincrone punto-punto basate su richiesta e risposta, utilizzate per configurazioni dinamiche e operazioni di inizializzazione.
- Le action estendono il concetto di servizio per la gestione di processi a durata variabile, permettendo il monitoraggio e l'interruzione delle operazioni in corso.
- Il sistema di trasformazioni tf2 mantiene la coerenza spaziale e temporale tra i frame di riferimento, consentendo la conversione automatica di pose e coordinate.
- Il parameter server gestisce in modo centralizzato i parametri di configurazione dei nodi, garantendo riproducibilità e tracciabilità del sistema.

L'adozione di ROS 2 rispetto alla versione precedente ROS 1 è motivata dal supporto nativo al real-time, dalla migliore gestione dei thread multipli e dalla compatibilità

con piattaforme embedded basate su architettura ARM come Jetson Orin NX. La separazione dei nodi e la comunicazione basata su DDS permettono di distribuire il carico computazionale tra core differenti, mantenendo latenze ridotte anche in presenza di flussi dati ad alta frequenza. Il middleware integra strumenti di diagnostica, logging e visualizzazione (RViz2, rqt, rosbag2) che semplificano il monitoraggio dei processi e l'analisi sperimentale.

Nel presente progetto ROS 2 costituisce il livello di coordinamento tra i moduli di acquisizione sensoriale, fusione dei dati e localizzazione. Ogni componente è implementato come nodo indipendente con interfacce standardizzate per la pubblicazione e la sottoscrizione dei messaggi. La sincronizzazione temporale dei pacchetti assicura coerenza tra i frame di riferimento e consente la parallelizzazione delle operazioni di predizione, correzione e aggiornamento della posa. L'architettura risultante è efficiente, scalabile e predisposta all'integrazione di moduli aggiuntivi.

Nei paragrafi successivi viene descritta la struttura software sviluppata, illustrando la pipeline di comunicazione, l'organizzazione dei nodi, le frequenze di aggiornamento dei sensori e la configurazione dei parametri per la stima di stato e la localizzazione.

6.1.1. Distribuzione utilizzata

Esistono diverse distribuzioni del middleware ROS 2, ciascuna caratterizzata da differenti livelli di supporto, stabilità e compatibilità con le versioni dei sistemi operativi. Nel contesto applicativo considerato è stata adottata la distribuzione Humble Hawksbill, scelta per la maturità e per il supporto a lungo termine garantito dalla comunità di sviluppo.

La distribuzione Humble è progettata per operare in modo ottimale su sistemi basati su Ubuntu 22.04, configurazione pienamente compatibile con la piattaforma hardware Jetson Orin NX utilizzata nel progetto. Tale versione assicura una gestione stabile delle dipendenze, un'integrazione semplificata con i principali pacchetti per la localizzazione e la navigazione e prestazioni adeguate per applicazioni robotiche in tempo quasi reale. Le informazioni tecniche relative alla distribuzione sono tratte dalla documentazione ufficiale [24].

6.1.2. Architettura ROS 2 Localizzazione

Come illustrato nella Figura 3.1 del Capitolo 3, l'architettura generale del veicolo adotta un paradigma modulare in cui ciascun livello funzionale, comprendente percezione, pianificazione, controllo e attuazione, è costituito da insiemi di nodi ROS 2 interconnessi e comunicanti attraverso topic, servizi e trasformazioni tf2. La sezione dedicata alla localizzazione appartiene al livello di percezione ed è responsabile della stima della posa del veicolo rispetto alla mappa globale. Tale componente è implementata nel workspace ROS 2 mediante un insieme di package indipendenti e scalabili, ognuno dei quali esegue un compito specifico all'interno della pipeline di

localizzazione. La Tabella 6.1 riassume la struttura dei package coinvolti e le relative interfacce di comunicazione.

Tabella 6.1.: Moduli software dedicati alla localizzazione.

Dettagli tecnici (package, nodo, topic, servizi)	Descrizione
Package: Lidar Driver Nodo: lidar_publisher_node Publisher: /scan (sensor_msgs/LaserScan)	Pubblica i dati del sensore LiDAR utilizzati per la costruzione della mappa e per la localizzazione probabilistica.
Package: Modello URDF Nodo: robot_state_publisher Subscriber: /joint_states (sensor_msgs/JointState) Publisher: /tf_static (tf2_msgs/TFMessage)	Gestisce la descrizione cinematica e geometrica del veicolo (URDF/Xacro) pubblicando le trasformazioni statiche tra i frame sensoriali e strutturali.
Package: Serial Communication Nodo: serial_handler_node Publisher: /imu (sensor_msgs/Imu) Publisher: /encoder_twist (geometry_msgs/TwistStamped) Publisher: /uwb (geometry_msgs/PoseStamped)	Gestisce la comunicazione seriale con il microcontrollore STM32 acquisendo e pubblicando in tempo reale i dati provenienti da IMU, encoder e sistema di posizionamento UWB.
Package: Sensor Fusion Nodo: sensor_fusion_node Subscriber: /imu, /encoder_twist, /uwb Publisher: /odom (nav_msgs/Odometry) Publisher: /tf (tf2_msgs/TFMessage)	Implementa la fusione sensoriale mediante filtro gaussiano non lineare producendo una stima odometrica continua e coerente nel frame odometrico.
Package: SLAM Mapping Nodo: slam_node Subscriber: /scan (sensor_msgs/LaserScan), /odom (nav_msgs/Odometry) Publisher: /map (nav_msgs/OccupancyGrid) Publisher: /map_metadata (nav_msgs/MapMetaData) Publisher: /tf (tf2_msgs/TFMessage) Service: /serialize_map, /deserialize_map, /pause_new_measurements	Costruisce e aggiorna in tempo reale la mappa di occupazione bidimensionale calcolando la trasformazione tra i frame odom e map.
Package: Adaptive Monte Carlo Localization Nodo: amcl_node Subscriber: /scan (sensor_msgs/LaserScan), /odom (nav_msgs/Odometry), /map (nav_msgs/OccupancyGrid) Publisher: /amcl_pose (geometry_msgs/PoseWithCovarianceStamped) Publisher: /particle_cloud (geometry_msgs/PoseArray) Publisher: /tf (tf2_msgs/TFMessage) Service: /global_localization	Stima la posa globale del veicolo nel frame map mediante algoritmo AMCL basato su filtraggio a particelle.

Il flusso informativo all'interno della pipeline di localizzazione segue una sequenza logica e temporale definita. Il nodo `serial_handler_node` acquisisce e pubblica in tempo reale i dati sensoriali grezzi provenienti da IMU, encoder e sistema UWB, che costituiscono le misure primarie di ingresso. Queste informazioni vengono successivamente fuse dal nodo `sensor_fusion_node` mediante un filtro gaussiano non lineare, il quale produce una stima odometrica filtrata e continua del moto del veicolo nel frame odometrico `odom`.

In parallelo, il nodo `lidar_publisher_node` fornisce il flusso di dati LiDAR necessario ai processi di mappatura e localizzazione. Il nodo `slam_node` utilizza tali misure, in combinazione con l'odometria filtrata, per costruire la mappa di occupazione dell'ambiente e per stimare la trasformazione `map→odom`. Su questa base, il nodo `amcl_node` applica un approccio probabilistico di tipo Monte Carlo per stimare la posa globale del veicolo nel frame `map`, pubblicando la correzione `map→odom` tramite il sistema `tf2`; la trasformazione `odom→base_link` rimane di competenza dell'odometria. I nodi `slam_node` e `amcl_node` operano in fasi distinte, rispettivamente mappatura e localizzazione, e non sono mai attivi contemporaneamente, evitando conflitti sull'albero `tf`.

Il nodo `robot_state_publisher` garantisce la coerenza geometrica e cinematica del modello pubblicando le trasformazioni statiche tra i frame sensoriali e strutturali del veicolo. L'intera architettura opera in tempo reale con frequenze di aggiornamento comprese tra 1 e 100 Hz, mantenendo sincronizzazione temporale e coerenza spaziale tra i diversi frame di riferimento. La distribuzione dei processi su nodi indipendenti consente una parallelizzazione efficiente del carico computazionale, assicurando modularità, scalabilità e integrazione diretta con i moduli di pianificazione e controllo descritti nei capitoli successivi.

Per approfondire la comprensione delle trasformazioni introdotte in questa architettura, la sezione seguente analizza la descrizione del modello URDF, che definisce la struttura cinematica e i frame di riferimento del veicolo, fornendo la base geometrica necessaria alla coerenza spaziale del sistema di localizzazione.

6.2. Modello URDF

Nel contesto della robotica mobile, la definizione accurata del modello meccanico è essenziale per la coerenza funzionale del sistema di localizzazione. Una descrizione geometrica e cinematica corretta consente di mantenere l'allineamento spaziale tra i sensori, i frame di riferimento e le trasformazioni impiegate nei moduli di stima e fusione dei dati. In assenza di un modello coerente, anche algoritmi avanzati di localizzazione risulterebbero affetti da errori sistematici dovuti a disallineamenti geometrici, con conseguente perdita di precisione nella stima della posa.

ROS 2 adotta il formato URDF (Unified Robot Description Format), linguaggio basato su XML che rappresenta in modo gerarchico la struttura cinematica del robot, le proprietà fisiche e i riferimenti spaziali dei componenti. Il modello definisce l'insieme

dei link e dei joint del veicolo, specificandone masse, inerzie e vincoli di movimento. Questa rappresentazione fornisce al middleware le informazioni necessarie per generare l'albero delle trasformazioni (tf2) e mantenere la corrispondenza tra la configurazione fisica e la rappresentazione logica del sistema. Nel contesto della localizzazione, tale meccanismo consente la propagazione automatica delle trasformazioni tra i frame del robot, eliminando la gestione esplicita delle matrici di rotazione e garantendo la coerenza spaziale tra le misure acquisite e le stime calcolate.

L'integrazione con strumenti come `robot_state_publisher` e `RViz` permette di verificare in ambiente virtuale la correttezza delle trasformazioni e l'allineamento dei sensori rispetto al frame principale. L'URDF assume pertanto una funzione strutturale all'interno dell'architettura di localizzazione, assicurando consistenza spaziale e tracciabilità delle pose nei processi di fusione, odometria e mappatura.

6.2.1. Modello URDF/Xacro del veicolo

Il modello del veicolo in scala 1:10 è implementato in formato Xacro, estensione parametrica dell'URDF che consente la definizione modulare dei componenti e la riusabilità dei parametri comuni. Questa soluzione progettuale garantisce coerenza tra modello fisico e virtuale e facilita l'adattamento a differenti configurazioni hardware, preservando riproducibilità e scalabilità.

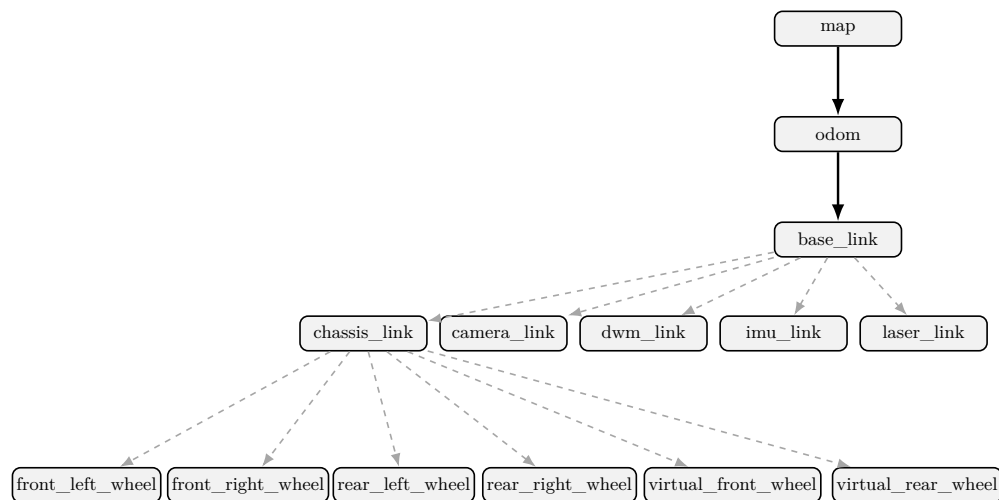


Figura 6.2.: Albero dei frame `tf` con le principali trasformazioni dinamiche e statiche.

La struttura del modello è organizzata gerarchicamente attorno al link `base_link`, nodo di riferimento a cui sono connessi i componenti meccanici e sensoriali tramite giunti fissi o mobili. Il link `chassis_link` rappresenta il corpo principale del veicolo, mentre le ruote anteriori e posteriori sono modellate con giunti di tipo `continuous` e `revolute`. Sono inoltre definiti giunti virtuali che riproducono la cinematica di tipo `car-like`, semplificando la gestione dei vincoli geometrici. Il tag `mimic` consente di sincronizzare il moto delle ruote simmetriche, riducendo la complessità del modello.

Ogni link è associato alle proprietà inerziali derivate dalle misure reali di massa e dimensione, incluse le matrici d'inerzia necessarie a garantire un comportamento dinamico realistico nelle simulazioni. L'assegnazione di materiali e colori differenti facilita la verifica visiva dell'allineamento geometrico in RViz.

Il modello include i principali sensori del veicolo, ossia unità IMU, sensore LiDAR e telecamera RGB, ciascuno montato su un link dedicato tramite giunto fisso. La definizione esplicita delle trasformazioni tra i sensori e il frame `base_link` consente di associare univocamente ogni misura alla posa del veicolo, condizione necessaria per la coerenza dei dati nei processi di fusione e localizzazione.

Il modello è realizzato in Xacro a partire dalle misure dirette del veicolo in scala 1:10, con le dimensioni definite come parametri simbolici riutilizzati nei link (Tabella 6.2); gli stessi valori fondano il modello cinematico del Capitolo 4. Il frame `base_link` è posto a livello del suolo e il telaio sollevato quanto basta perché le ruote tocchino il piano.

Tabella 6.2.: Parametri geometrici e di massa del modello URDF/Xacro.

Grandezza	Valore
Massa del telaio	4,0 kg
Lunghezza del telaio	0,43 m
Larghezza (carreggiata)	0,20 m
Altezza del telaio	0,030 m
Interasse	0,26 m
Raggio delle ruote	0,033 m
Larghezza delle ruote	0,027 m
Massa di ciascuna ruota	0,341 kg
Escursione angolare di sterzo	$\pm 0,4$ rad ($\approx \pm 23^\circ$)

A ogni link sono associate le proprietà inerziali: masse misurate e relativi tensori per telaio e ruote, tensori completi dei termini extra-diagonali esportati dai modelli CAD per IMU e LiDAR, tensori diagonali per telecamera e tag UWB. La cinematica car-like è resa con due ruote virtuali equivalenti del modello a bicicletta, in trazione posteriore e sterzo anteriore; le quattro ruote reali ne seguono il moto tramite il tag `mimic`.

La collocazione dei sensori richiede attenzione, poiché ne determina la correttezza delle misure. Ciascun sensore è fissato a `base_link` con un giunto fisso che riproduce la reale posizione di montaggio (Tabella 6.3); le trasformazioni sono propagate da `tf2`, che riferisce ogni misura grezza al frame del veicolo prima della fusione. Un errore negli offset si tradurrebbe in un bias sistematico nella posa: per questo sono stati misurati con cura sul mezzo reale.

Tabella 6.3.: Trasformazioni statiche dei sensori rispetto a base_link (giunti fissi).

Frame sensore	x [m]	y [m]	z [m]	Rotazione
imu_link	0,00	0,00	0,070	nessuna
laser_link	0,17	0,05	0,10	nessuna
camera_link	0,00	0,00	0,15	nessuna
dwm_link (UWB)	0,00	0,00	0,10	nessuna

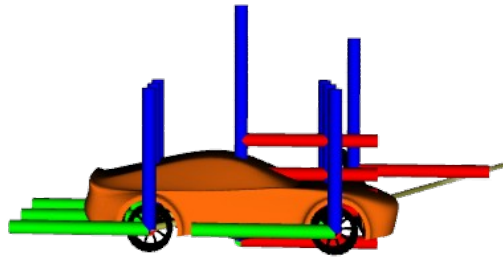


Figura 6.3.: Modello URDF del veicolo visualizzato in RViz.

L'URDF costituisce la base geometrica del sistema di localizzazione in ROS 2, poiché abilita la propagazione automatica delle trasformazioni tf tra i frame del robot. La definizione dei riferimenti spaziali nel modello permette al middleware di calcolare le trasformazioni tra i sistemi di coordinate in modo nativo, senza la necessità di gestire manualmente le matrici di rotazione. Questo meccanismo garantisce la coerenza tra le misure dei sensori e le stime derivate dagli algoritmi di localizzazione, mantenendo l'allineamento tra mondo reale e rappresentazione interna del veicolo. L'utilizzo del formato Xacro assicura inoltre flessibilità nella modifica e nel riuso della descrizione, favorendo la manutenibilità dell'architettura e la stabilità del flusso informativo nelle successive fasi di percezione e controllo. La definizione strutturata dei frame e la gestione automatica delle trasformazioni tf rappresentano infine la base per l'implementazione del modulo di localizzazione odometrica descritto nel capitolo seguente.

6.3. Architettura di comunicazione ROS 2 per la localizzazione

La comunicazione tra i moduli che compongono il sistema di localizzazione è implementata mediante il middleware ROS 2, il quale fornisce meccanismi standardizzati di scambio dati basati su topic, servizi e trasformazioni. Tale infrastruttura consente la trasmissione asincrona e continua dei flussi informativi tra i nodi dedicati

all'acquisizione sensoriale, alla fusione dei dati e alla stima della posa. L'architettura comunicativa è progettata per mantenere coerenza temporale e spaziale tra le misure provenienti dai sensori e le stime di stato derivate dagli algoritmi di localizzazione. La struttura risulta inoltre modulare e scalabile, condizione che permette l'integrazione di ulteriori sensori o moduli di elaborazione senza modificare le interfacce esistenti. La Figura 6.4 illustra la pipeline di comunicazione del sistema, evidenziando i nodi principali e le rispettive connessioni attraverso i topic e le trasformazioni. Ogni nodo è rappresentato come un'entità funzionale autonoma che pubblica o sottoscrive messaggi, contribuendo a un flusso dati distribuito ma coerente. I collegamenti continui indicano lo scambio di messaggi tra i nodi, mentre le connessioni tratteggiate rappresentano le trasformazioni statiche e dinamiche gestite dal sistema **tf2**. La distinzione tra i flussi consente di visualizzare la catena completa che collega i dispositivi di acquisizione (IMU, encoder, UWB, LiDAR) ai moduli di elaborazione (filtro di Kalman non lineare, SLAM, AMCL) e alla propagazione delle trasformazioni di riferimento.

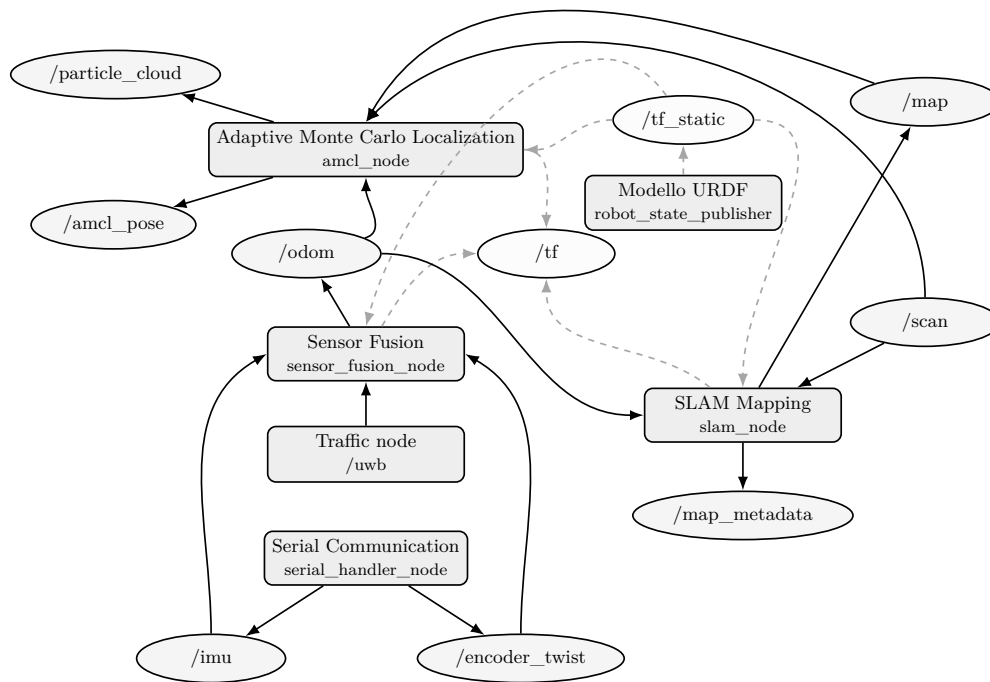


Figura 6.4.: Pipeline di comunicazione tra i nodi del sistema ROS 2 di localizzazione.

6.3.1. Sorgenti dati e interfacce di comunicazione

I dati impiegati nei processi di localizzazione provengono da tre tipologie di sensori: unità inerziali con encoder, sistema di posizionamento UWB e sensore LiDAR. Ogni dispositivo utilizza un canale di comunicazione dedicato verso la piattaforma di elaborazione, al fine di mantenere separazione funzionale, ridurre la latenza di trasfe-

rimento e garantire la sincronizzazione temporale delle misure. Questa configurazione consente di integrare sorgenti eterogenee di dati preservando la coerenza tra i flussi informativi e la stabilità del processo di fusione sensoriale.

IMU ed encoder

L'acquisizione dei dati inerziali e odometrici è gestita dal microcontrollore STM32, che comunica con la piattaforma di elaborazione NVIDIA Jetson Orin NX attraverso un collegamento seriale configurato a una velocità di trasmissione pari a 921 600 baud. Il canale seriale opera in modalità non bloccante con tempo di attesa di 0,1 secondi, condizione che consente una lettura continua dei pacchetti e riduce il rischio di accumulo nella coda di ricezione. Il nodo dedicato alla gestione della comunicazione, denominato serial handler node, esegue ciclicamente la lettura delle stringhe ricevute, la decodifica in formato UTF-8 e la suddivisione dei dati in campi separati da delimitatori. I valori numerici validati vengono convertiti in messaggi ROS 2 e pubblicati con cadenza regolare di 10 millisecondi, corrispondente a una frequenza di aggiornamento di 100 Hz.

Le informazioni vengono trasmesse su due flussi principali:

- topic /imu, che contiene le misure di velocità angolare e orientamento riferite al frame imu_link;
- topic /encoder_twist, che riporta la velocità lineare espressa nel frame base_link.

La gestione del flusso seriale è realizzata mediante un processo indipendente rispetto al ciclo principale di esecuzione, in modo da minimizzare la latenza e preservare la continuità del campionamento. Ogni messaggio è marcato temporalmente al momento della pubblicazione, garantendo la coerenza tra i dati provenienti dai diversi sensori e la corretta sincronizzazione temporale con le altre sorgenti del sistema di localizzazione. Questo schema di comunicazione consente di ottenere un flusso stabile e affidabile, idoneo all'elaborazione in tempo quasi reale da parte dei moduli di fusione e stima della posa.

6.3.2. Sistema di posizionamento UWB e comunicazione V2X

Il sistema di posizionamento basato su tecnologia Ultra Wide Band è integrato con un'infrastruttura di comunicazione V2X (Vehicle-to-Everything), che fornisce in tempo reale le informazioni di localizzazione del veicolo. I dati di posizione vengono trasmessi da un server dedicato tramite protocollo TCP e ricevuti dalla piattaforma di elaborazione installata a bordo. Il nodo responsabile della gestione di questa comunicazione, denominato traffic node, riceve i messaggi contenenti le coordinate spaziali del veicolo e li converte in un formato compatibile con ROS 2.

Le coordinate ricevute sono espresse nel frame dwm_link e comprendono le componenti planari x e y . Ogni messaggio è associato a un timestamp generato dal

clock ROS 2, in modo da mantenere la coerenza temporale con i dati provenienti dagli altri sensori del sistema di localizzazione. Le informazioni vengono pubblicate sul topic `/uwb` con frequenza di 10 Hz, fornendo un flusso continuo di dati di posa bidimensionale utilizzati dal nodo di fusione.

Il collegamento TCP impiegato garantisce un trasferimento affidabile dei pacchetti e consente di mantenere aggiornato lo stato del veicolo in tempo quasi reale. Questa architettura di comunicazione assicura l'integrazione dei dati UWB con le altre sorgenti di misura, migliorando la consistenza spaziale delle stime di localizzazione ottenute dai moduli di fusione e mappatura.

Sensore LiDAR

Il sensore LiDAR è connesso direttamente alla piattaforma di elaborazione Jetson Orin NX mediante interfaccia USB ad alta velocità. Il nodo dedicato all'acquisizione gestisce in modo continuo il flusso dei dati grezzi provenienti dal dispositivo, eseguendo la decodifica dei pacchetti e la loro conversione in messaggi compatibili con ROS 2. Le misure vengono pubblicate sul topic `/scan` con frequenza di 10 Hz, garantendo un aggiornamento costante delle informazioni di distanza.

Ogni messaggio contiene un profilo polare di misure angolari e radiali, che rappresenta la distribuzione spaziale degli ostacoli nell'ambiente circostante. Tali dati costituiscono l'ingresso principale per i moduli di mappatura e localizzazione probabilistica, nei quali vengono utilizzati per stimare la trasformazione tra i frame di riferimento e per aggiornare la rappresentazione della mappa di occupazione. La frequenza di 10 Hz bilancia risoluzione spaziale, latenza e carico computazionale nella pipeline.

6.3.3. Fusione e propagazione delle stime

I dati provenienti dai sensori rappresentano gli ingressi del nodo di fusione, che implementa un filtro gaussiano non lineare per la stima dello stato odometrico nel frame di riferimento `odom`. Il nodo combina le misure di velocità lineare, velocità angolare e posizione planare per ottenere una stima coerente della traiettoria del veicolo, riducendo l'influenza del rumore e delle asincronie temporali. In uscita vengono generati i seguenti flussi:

- topic `/odom`, contenente la stima odometrica del moto del veicolo;
- topic `/tf`, contenente la trasformazione dinamica tra i frame `odom` e `base_link`.

Il nodo `slam node` utilizza le informazioni pubblicate sui topic `/scan` e `/odom` per costruire e aggiornare in tempo reale la mappa di occupazione bidimensionale dell'ambiente. Il processo produce la trasformazione tra i frame `map` e `odom`, pubblicata attraverso il sistema di trasformazioni `tf`, e i topic `/map` e `/map_metadata`, che contengono rispettivamente la rappresentazione della mappa e i relativi parametri descrittivi.

La localizzazione globale è gestita dal nodo `amcl node`, che impiega i topic `/scan`, `/odom` e `/map` come sorgenti di ingresso. L'algoritmo implementato, basato su filtraggio a particelle, calcola la posa assoluta del veicolo nel frame `map`, stimando sia la posizione sia l'incertezza associata. Le uscite principali del nodo sono:

- topic `/amcl_pose`, che riporta la stima della posa con la matrice di covarianza associata;
- topic `/particle_cloud`, che contiene la distribuzione delle particelle utilizzate per l'aggiornamento probabilistico.

La combinazione tra i moduli di fusione, mappatura e localizzazione consente di mantenere una stima coerente della posizione del veicolo all'interno della mappa globale. Le trasformazioni generate dai nodi vengono propagate attraverso il sistema `tf`, garantendo l'allineamento geometrico e temporale tra i diversi frame di riferimento e assicurando la consistenza dell'intero processo di localizzazione.

6.3.4. Gestione delle trasformazioni e sincronizzazione temporale

Il nodo `robot state publisher` pubblica sul topic `/tf_static` le trasformazioni statiche definite nel modello URDF, che descrivono la configurazione geometrica e cinematica del veicolo. Tali trasformazioni definiscono le relazioni fisse tra i frame sensoriali e strutturali, garantendo la coerenza spaziale necessaria alle operazioni di localizzazione e mappatura.

Ogni messaggio del sistema è marcato temporalmente rispetto al clock della piattaforma Jetson, consentendo l'allineamento temporale tra i flussi provenienti dai sensori. La sincronizzazione tra topic a frequenze diverse non si basa su un allineamento rigido dei messaggi, ma sul meccanismo di interpolazione temporale offerto da `tf2`. Ogni trasformazione è conservata in un buffer con la propria marca temporale e, quando un nodo richiede la posa relativa tra due frame a un dato istante, `tf2` la ricostruisce interpolando tra i campioni adiacenti. Misure acquisite con cadenze differenti, come le scansioni LiDAR a 10 Hz e l'odometria, possono così essere riferite a un istante comune senza richiedere che i rispettivi messaggi arrivino simultaneamente. La tolleranza temporale ammessa nella ricerca delle trasformazioni è fissata a 1 s, valore del parametro `transform_tolerance` dell'AMCL, sufficiente ad assorbire i ritardi di trasporto e il jitter di rete mantenendo valide le trasformazioni per un breve intervallo oltre il loro timestamp. Concretamente, la scansione LiDAR in ingresso all'AMCL è gestita da un filtro di messaggi basato su `tf2`, che trattiene ogni scansione finché la trasformazione corrispondente al suo timestamp non diventa disponibile entro la tolleranza: le scansioni il cui istante ricade nel buffer vengono allineate per interpolazione, mentre quelle troppo vecchie o per le quali la trasformazione non arriva in tempo vengono scartate, evitando l'impiego di dati incoerenti. Nel nodo di fusione, come descritto nella Sezione 7.2, l'allineamento è invece garantito dal calcolo dell'intervallo Δt direttamente dai marcatori temporali dei messaggi. Le politiche

di qualità del servizio (QoS) sono configurate in base alle caratteristiche dei flussi informativi:

- flussi ad alta frequenza, come `/imu` e `/encoder_twist`, impostati in modalità `best effort`;
- flussi di stima, come `/odom` e `/amcl_pose`, impostati in modalità `reliable`;
- trasformazioni dinamiche su `/tf`, configurate in modalità `reliable` e `volatile`;
- trasformazioni statiche su `/tf_static`, configurate con `durability transient local`.

La struttura modulare dei nodi e la separazione dei canali di comunicazione consentono l'esecuzione parallela dei processi e una distribuzione efficiente del carico computazionale. Le comunicazioni asincrone riducono le dipendenze tra i moduli e favoriscono la scalabilità del sistema, assicurando nel contempo coerenza spaziale e temporale tra le misure sensoriali e le stime di stato.

Il capitolo successivo presenta l'implementazione della fusione sensoriale su Jetson Orin NX: caratterizzazione del rumore (analisi statistiche), identificazione delle matrici di covarianza, e applicazione di filtri gaussiani non lineare (EKF e UKF) per l'ottimizzazione della stima di stato.

Capitolo 7.

Implementazione del sistema di fusione sensoriale

La fusione sensoriale rappresenta il componente centrale del sistema di localizzazione presentato nel Capitolo 6, poiché consente di combinare misure eterogenee provenienti dai sensori di bordo all'interno di un unico stimatore di stato. Tale processo assicura coerenza spaziale e temporale tra le informazioni e fornisce al veicolo una stima continua della propria posa rispetto all'ambiente di riferimento.

Il funzionamento del modulo si basa sui principi probabilistici descritti nel Capitolo 4, in cui sono stati analizzati i filtri bayesiani e le varianti del filtro di Kalman per modelli dinamici non lineari. Sulla base di tali modelli, il presente capitolo illustra l'implementazione pratica delle tecniche di fusione nel contesto dell'architettura ROS 2 di localizzazione.

In apertura viene condotta l'analisi statistica dei sensori impiegati nel sistema, con particolare riferimento all'encoder, all'IMU e al modulo di localizzazione UWB. Per ciascun sensore vengono determinati i parametri di rumore e le corrispondenti covarianze, necessarie alla definizione delle matrici di osservazione e di processo del filtro.

Segue la descrizione dell'implementazione software dei filtri di stima, nelle versioni estesa (EKF) e unscented (UKF), comprendente la struttura dei cicli di predizione e aggiornamento, i criteri di inizializzazione e le strategie di gestione numerica adottate per garantire stabilità e convergenza.

La parte conclusiva riporta la validazione sperimentale del sistema attraverso il confronto tra le stime ottenute dai diversi approcci e le misure di riferimento. Le prestazioni vengono valutate mediante indici quantitativi di errore, tra cui radice quadratica media (RMSE) e errore medio assoluto (MAE), al fine di verificare la precisione e la robustezza della localizzazione ottenuta con la fusione.

7.1. Analisi dei sensori e determinazione delle covarianze

7.1.1. Encoder

L'encoder incrementale fornisce la misura primaria della velocità longitudinale del veicolo. La conversione degli impulsi in segnali digitali consente la stima della velocità angolare dell'albero motore, successivamente trasformata in velocità lineare mediante il rapporto totale di riduzione G e il raggio dinamico R della ruota. Il coefficiente k_{slip} , determinato sperimentalmente, compensa le variazioni dovute a slittamento o differenze tra raggio nominale ed effettivo.

La misura encoder è modellata come osservazione rumorosa della velocità longitudinale:

$$v_{x,m} = v_x + b_e + n_e,$$

dove:

- b_e rappresenta il bias di calibrazione;
- n_e un rumore gaussiano a media nulla con varianza R_e .

Il bias evolve secondo un processo a passo casuale di varianza Q_e , la cui stima è ottenuta da analisi statistica dei dati.

Implementazione del filtro di Kalman per l'encoder

Per incrementare la qualità del segnale misurato, sul microcontrollore STM32 è stato implementato un filtro di Kalman lineare discreto. Il filtro opera in tempo reale sulla misura della velocità angolare del motore, attenuando il rumore bianco gaussiano dovuto a quantizzazione, vibrazioni meccaniche e interferenze elettriche. Il modello dinamico discreto impiegato è di tipo lineare:

$$x_{k+1} = A_d x_k + B_d u_k + w_k, \quad y_k = C x_k + v_k,$$

dove:

- $x_k = [\theta_m, \omega_m]^T$ rappresenta lo stato;
- w_k e v_k sono rispettivamente i rumori di processo e di misura, gaussiani a media nulla e covarianze Q e R .

L'uscita filtrata $\hat{\omega}_m$ presenta un andamento regolare e coerente con la dinamica del sistema, riducendo le componenti ad alta frequenza senza introdurre ritardi apprezzabili. La velocità lineare del veicolo è calcolata come

$$\hat{v} = k_{\text{slip}} \frac{R}{G} \hat{\omega}_m.$$

La Figura 7.1 mostra il segnale grezzo acquisito dall'encoder, caratterizzato da rumore bianco di ampiezza variabile che giustifica l'utilizzo di una stima ottimale per la ricostruzione della velocità reale.

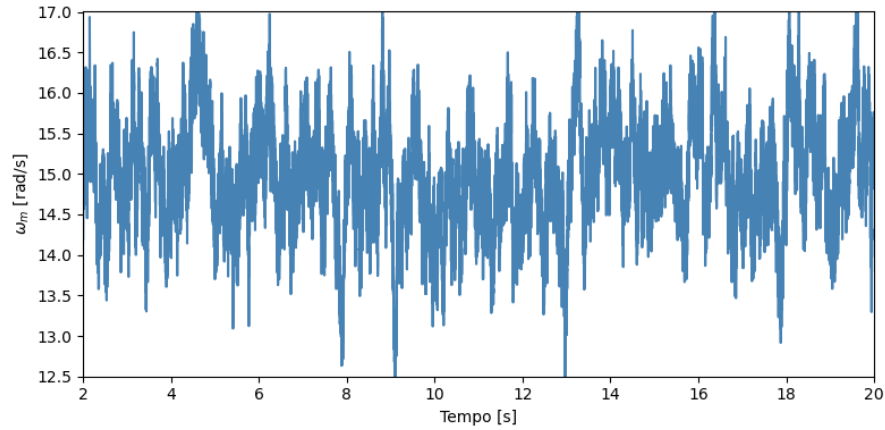


Figura 7.1.: Segnale grezzo dell'encoder velocità angolare ω_m .

Analisi dei risultati

La valutazione delle prestazioni del filtro è stata effettuata confrontando la velocità angolare misurata e quella stimata. La Figura 7.3 riporta l'errore residuo tra i due segnali, mentre la Figura 7.2 mostra il comportamento del filtro in una finestra temporale ristretta.

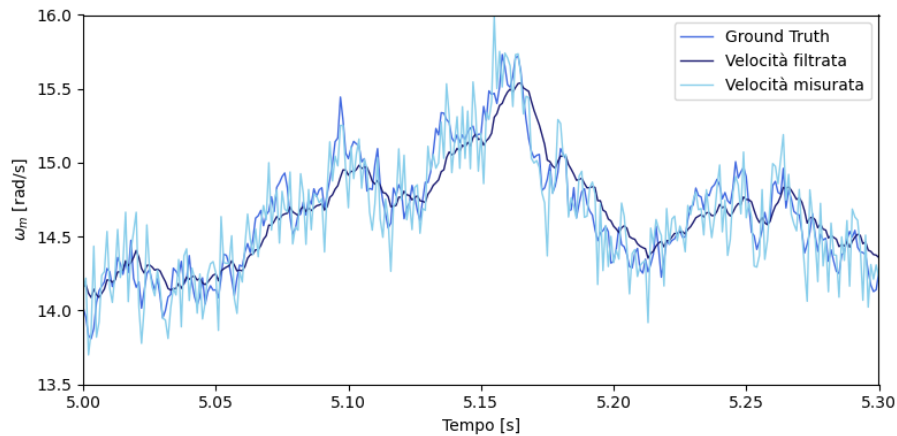


Figura 7.2.: Confronto tra misura grezza e stima filtrata della velocità angolare ω_m .

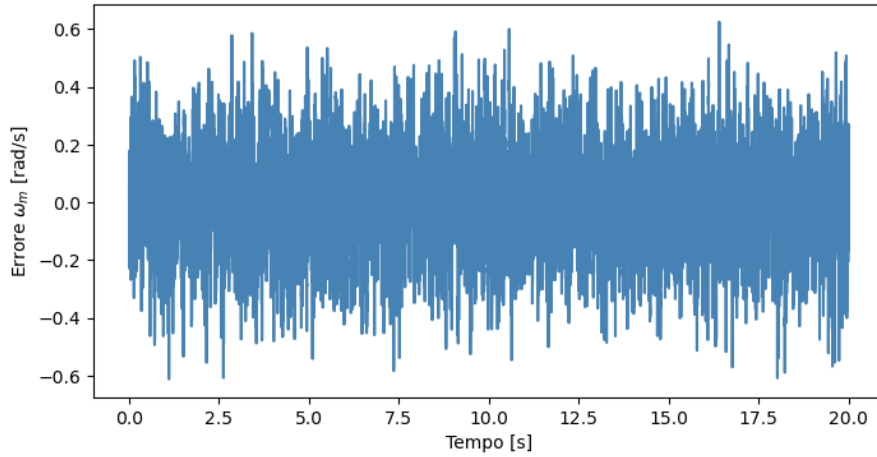


Figura 7.3.: Errore residuo tra misura e stima filtrata della velocità angolare ω_m .

Le metriche di prestazione calcolate sui dati acquisiti sono riportate in Tabella 7.1. L'analisi evidenzia una riduzione della varianza del rumore e una maggiore stabilità temporale del segnale stimato. La configurazione adottata delle covarianze Q e R risulta coerente con la dinamica del sistema e garantisce un comportamento stabile del filtro, in accordo con l'ipotesi di rumore gaussiano assunta nei modelli gaussiani. Il segnale filtrato costituisce pertanto un ingresso affidabile per la fase di fusione, migliorando la coerenza della stima di stato e riducendo la deriva tipica dell'odometria pura.

Tabella 7.1.: Metriche di valutazione del filtro applicato ai dati dell'encoder.

Metrica	Valore
Errore quadratico medio (RMSE)	0.1699 rad/s
Errore medio assoluto (MAE)	0.1352 rad/s

Covarianza associata alla misura di velocità lineare

Dato il filtro di Kalman implementato su STM32 con matrice delle covarianze $P_{k|k}$ e stato $x_k = [\theta_m, \omega_m]^\top$, la varianza della velocità lineare filtrata $\hat{v}_k = k_{\text{slip}} \frac{R}{G} \hat{\omega}_{m,k}$ è

$$\sigma_v^2(k) = \left(k_{\text{slip}} \frac{R}{G}\right)^2 e_2^\top P_{k|k} e_2, \quad e_2 = [0 \ 1]^\top.$$

Qualora l'incertezza dei parametri non sia stimata dallo stato della fusione, si adotta la propagazione

$$\sigma_v^2 \approx \left(k_{\text{slip}} \frac{R}{G}\right)^2 \sigma_\omega^2 + \left(\frac{R}{G} \omega\right)^2 \sigma_k^2 + \left(k_{\text{slip}} \frac{\omega}{G}\right)^2 \sigma_R^2 + \left(k_{\text{slip}} \frac{R}{G^2} \omega\right)^2 \sigma_G^2,$$

e si impiega σ_v^2 come covarianza di misura $R_{\text{eff}}(k)$ nel modello di misura dell'EKF e UKF di fusione.

7.1.2. IMU BNO055

La stima della covarianza di misura del giroscopio è stata effettuata mediante un'analisi statistica dei dati acquisiti in condizioni statiche, con il veicolo fermo su un piano orizzontale. L'obiettivo è la determinazione del bias e della varianza del rumore della velocità angolare attorno all'asse verticale, parametri necessari alla definizione della matrice di osservazione nel modello di fusione.

La prova è stata condotta con il sensore in assenza di sollecitazioni meccaniche esterne, in modo da isolare il rumore intrinseco dalle componenti dovute a vibrazioni o movimento del telaio. La Figura 7.4 mostra il segnale grezzo della velocità angolare $\dot{\psi}_m$, mentre le Figure 7.5 e 7.6 riportano rispettivamente la deviazione standard su finestra mobile e la distribuzione dei residui, approssimabile con una densità gaussiana a media nulla.

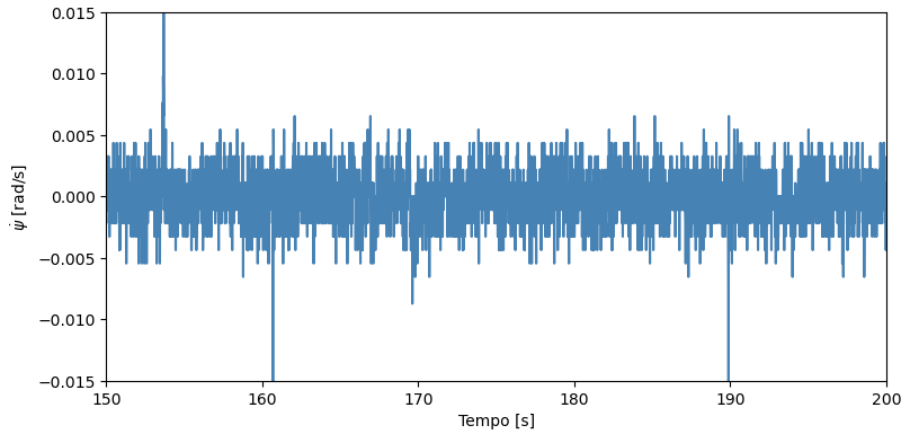


Figura 7.4.: Segnale grezzo della velocità angolare $\dot{\psi}_m$.

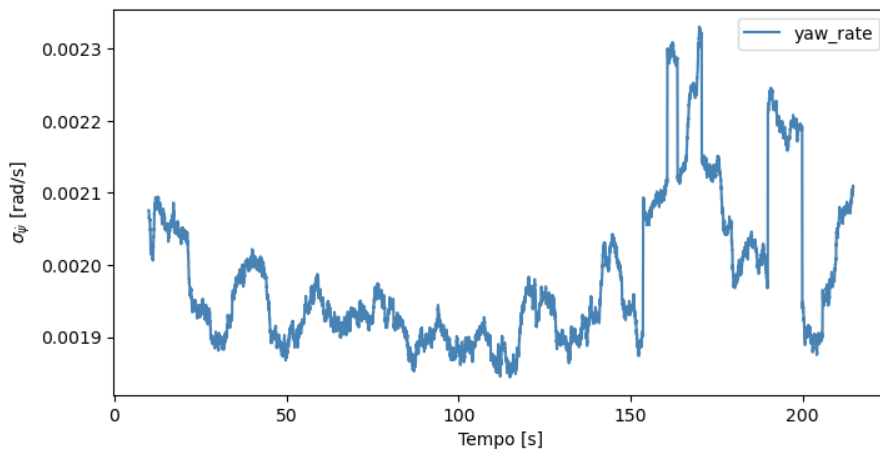


Figura 7.5.: Deviazione standard della velocità angolare $\dot{\psi}_m$.

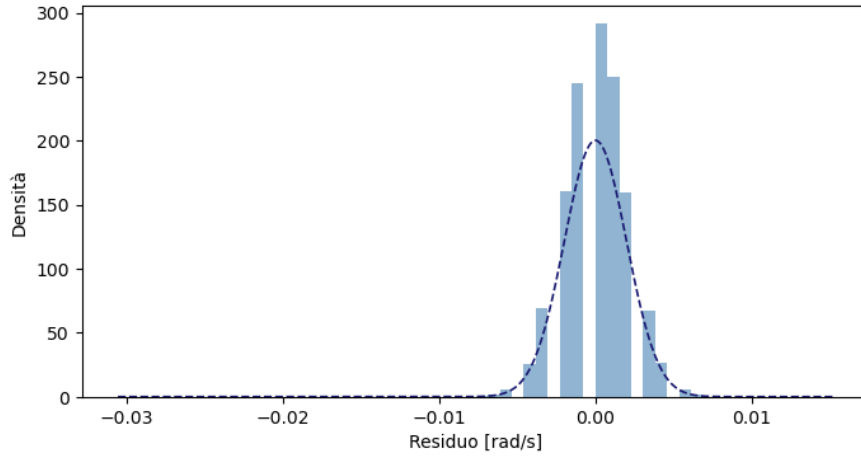


Figura 7.6.: Distribuzione dei residui della velocità angolare $\dot{\psi}_m$ con densità gaussiana stimata.

La Tabella 7.2 riassume i risultati dell'analisi statistica condotta in condizioni statiche. Il valore medio risulta prossimo allo zero, confermando la corretta calibrazione e l'assenza di drift significativo.

Tabella 7.2.: Risultati dell'analisi statistica del giroscopio.

Condizione	Media residui [rad/s]	Bias [rad/s]	Deviazione std [rad/s]	Varianza [(rad/s) ²]
Veicolo fermo	0	-9.65×10^{-7}	0.001992	4×10^{-6}

Il rumore del giroscopio lungo l'asse di imbardata può essere modellato come processo gaussiano a media nulla e varianza costante:

$$R_g = 4 \times 10^{-6} \text{ (rad/s)}^2.$$

Tale valore è adottato come covarianza di misura nel nodo di fusione sensoriale. Il bias medio stimato, pari a

$$b_g = -9.65 \times 10^{-7} \text{ rad/s},$$

risulta trascurabile e viene compensato in fase di inizializzazione del filtro.

Confronto con le specifiche del sensore

La varianza ottenuta sperimentalmente risulta superiore rispetto ai valori nominali riportati nel datasheet del sensore BNO055, che indicano un rumore tipico pari a circa $3 \times 10^{-6} \text{ (rad/s)}^2$ con banda di 47 Hz. La discrepanza è attribuibile alla maggiore banda di campionamento adottata in acquisizione (100 Hz) e alle vibrazioni residue

del telaio e dei componenti elettrici. La stima sperimentale rappresenta pertanto una valutazione conservativa, utile a descrivere la rumorosità effettiva del sistema nel contesto veicolare.

Si conclude che il rumore giroscopico può essere considerato stazionario e modellato con varianza costante. La varianza stimata $R_g = 4 \times 10^{-6} \text{ (rad/s)}^2$ è utilizzata come covarianza di misura del giroscopio all'interno del nodo di fusione.

7.1.3. Modulo di localizzazione UWB DWM1001

Il modulo DWM1001 consente la stima della posizione tridimensionale del nodo mobile mediante misure di tempo di volo (Time of Flight) tra tag e ancore fisse. Il sistema utilizza la tecnica di multilaterazione a partire dalle distanze misurate, fornendo in uscita le coordinate cartesiane (x_m, y_m, z_m) in un riferimento locale definito dalla disposizione delle ancore.

Analisi statistica in condizioni statiche

L'analisi statistica è stata condotta posizionando il tag in una posizione nota, con coordinate teoriche

$$x_{\text{true}} = 0.63 \text{ m}, \quad y_{\text{true}} = 1.73 \text{ m}, \quad z_{\text{true}} = 0.15 \text{ m}.$$

Sono stati acquisiti campioni di posizione mantenendo il tag fisso e in condizioni operative controllate. La prova è stata eseguita prevalentemente in configurazione line of sight (LOS), con visibilità diretta delle ancore e assenza di riflessioni significative; in alcune porzioni del tracciato è stato tuttavia introdotto un assetto parzialmente non line of sight (NLOS) mediante schermature localizzate, al fine di caratterizzare la variazione statistica del rumore in presenza di ostacoli.

Le Figure 7.7–7.9 riportano l'andamento temporale delle coordinate, le distribuzioni dei campioni e la dispersione spaziale nel piano orizzontale. I segnali risultano stazionari e affetti da rumore a media nulla, approssimabile con una distribuzione gaussiana.

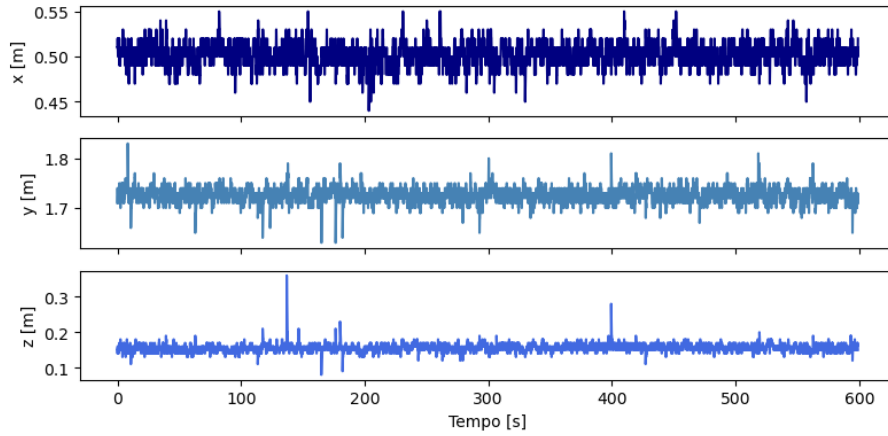


Figura 7.7.: Andamento temporale delle coordinate misurate dal modulo UWB.

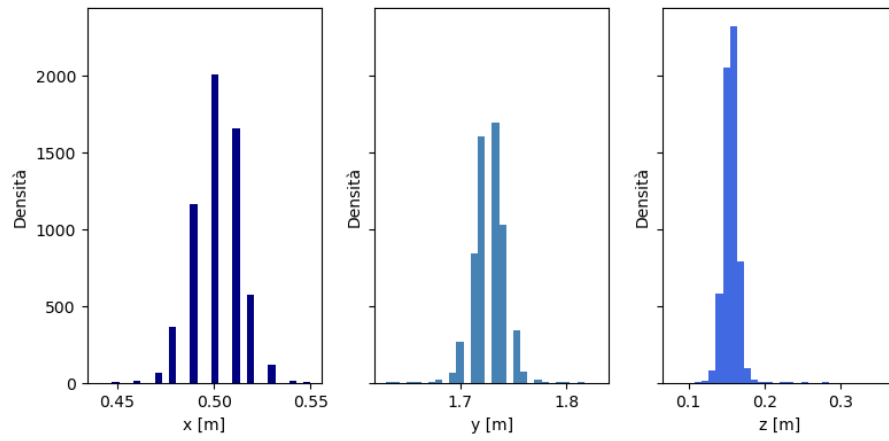


Figura 7.8.: Distribuzione dei campioni UWB lungo le due coordinate spaziali considerate.

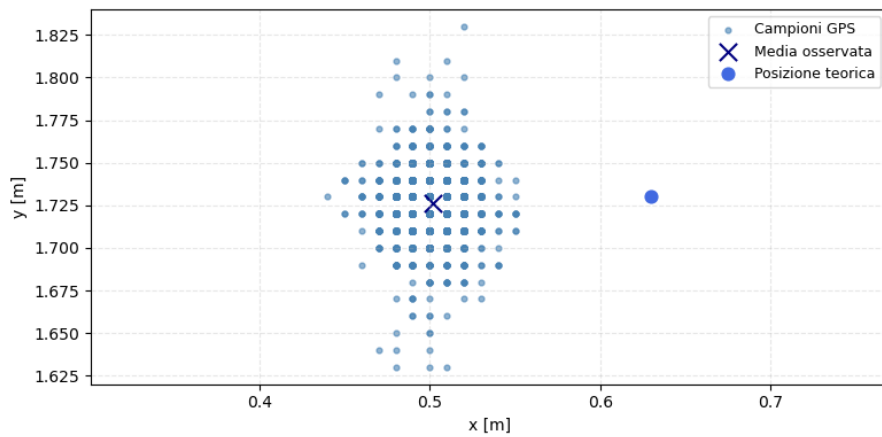


Figura 7.9.: Dispersione spaziale dei campioni UWB nel piano xy rispetto alla posizione teorica.

Risultati statistici e modellazione del rumore

I risultati dell'analisi statistica condotta in condizioni LOS sono riportati in Tabella 7.3. Le medie delle coordinate risultano prossime ai valori teorici di riferimento, con scarti inferiori a 15 cm, mentre le deviazioni standard sono comprese tra 1.1 cm e 1.5 cm. Tali valori sono coerenti con il comportamento atteso del sistema in ambiente controllato e costituiscono la base per la stima della varianza di misura nelle componenti orizzontali.

Tabella 7.3.: Statistiche delle coordinate UWB in condizioni statiche LOS.

Asse	Media [m]	Deviazione std [m]	Varianza [m ²]
x	0.501826	0.012159	0.000148
y	1.725901	0.014807	0.000219

Assumendo isotropia del rumore nel piano orizzontale, la covarianza di misura è stata valutata come media delle varianze sulle due componenti:

$$R_{\text{LOS}} = \frac{1}{2} (0.000148 + 0.000219) = 1.84 \times 10^{-4} \text{ m}^2.$$

Nelle porzioni del tracciato in cui il segnale è stato parzialmente degradato da ostruzioni o riflessioni (condizione NLOS), l'analisi ha evidenziato un incremento della deviazione standard fino a circa 0.3 m, corrispondente a una varianza di $9 \times 10^{-2} \text{ m}^2$. Questo comportamento è coerente con le specifiche del datasheet del modulo DWM1001, che riporta un'accuratezza tipica inferiore a 10 cm in condizioni LOS e una precisione ridotta fino a 30–40 cm in scenari NLOS, principalmente a causa degli effetti di multipath e attenuazione del segnale.

Considerando che il sistema di localizzazione è destinato a operare in ambienti misti, caratterizzati da alternanza tra zone con visibilità diretta e altre parzialmente schermate, è stata adottata una covarianza di misura equivalente riferita al piano orizzontale, rappresentativa di un compromesso tra le due condizioni:

$$R_{\text{UWB,eff}} = 0.5 R_{\text{LOS}} + 0.5 R_{\text{NLOS}} = 0.5 (1.84 \times 10^{-4}) + 0.5 (9 \times 10^{-2}) \approx 4.5 \times 10^{-2} \text{ m}^2.$$

Si osserva che $R_{\text{NLOS}} \approx 9 \times 10^{-2} \text{ m}^2$ è circa 400 volte superiore a $R_{\text{LOS}} \approx 1.84 \times 10^{-4} \text{ m}^2$; il valore efficace risulta pertanto dominato dal termine NLOS (deviazione standard $\approx 21 \text{ cm}$). Il peso uguale non rispecchia l'effettiva frazione di campioni NLOS lungo il percorso: è una scelta deliberatamente conservativa. Sovrastimare la covarianza è infatti meno rischioso che sottostimarla, poiché un valore troppo alto riduce il guadagno di Kalman e rallenta la correzione, mentre uno troppo basso dà credito a misure corrotte e può far divergere il filtro. Il gating chi-quadro del Capitolo 4 scarta inoltre a monte le letture più anomale, per cui a $R_{\text{UWB,eff}}$ resta da

descrivere la sola dispersione residua delle misure rimaste.

La condizione di visibilità non è del resto indeterminabile: esistono metodi di riconoscimento in tempo reale, basati sulla varianza dei range, sulla risposta all'impulso del canale o su classificatori appresi dai dati [1, 2], con relativi schemi di covarianza adattiva. Non sono stati adottati perché il modulo DWM1001 restituisce la sola posizione, senza esporre risposta di canale o distanze per singola ancora.

7.1.4. Sintesi delle covarianze di misura

La Tabella 7.4 riassume le covarianze di misura stimate per i sensori impiegati nel sistema di localizzazione. I valori derivano dalle analisi statistiche in condizioni statiche e sono utilizzati come parametri di rumore nelle matrici di osservazione dell'EKF e dell'UKF.

La covarianza dell'encoder R_e è stata ottenuta a partire dalla matrice delle covarianze $P_{k|k}$ del filtro di Kalman implementato sul microcontrollore STM32. Per l'unità IMU è stata impiegata la varianza del giroscopio attorno all'asse di imbardata, mentre per il modulo UWB è stato adottato un valore equivalente che rappresenta la media pesata tra i regimi LOS e NLOS.

Tabella 7.4.: Covarianze di misura dei sensori impiegate nel filtro di fusione.

Sensore	Quantità osservata	Covarianza di misura
Encoder	Velocità lineare v_x	$R_e = 0.0102 \text{ (m/s)}^2$
IMU (giroscopio)	Velocità angolare $\dot{\psi}$	$R_g = 4 \times 10^{-6} \text{ (rad/s)}^2$
UWB (DWM1001)	Posizione (x, y, z)	$R_{\text{UWB,eff}} = 4.5 \times 10^{-2} \text{ m}^2$

Le covarianze elencate rappresentano le incertezze di misura adottate nello stimatore di stato. La loro determinazione sperimentale garantisce coerenza tra modello e sensori reali, condizione necessaria per la stabilità e la convergenza numerica del filtro di fusione.

7.2. Implementazione software del filtro di fusione

L'implementazione dei filtri di Kalman, nelle versioni estesa (EKF) e unscented (UKF), segue le formulazioni riportate nel Capitolo 4, in cui sono stati definiti i modelli dinamici, le funzioni di osservazione e le matrici di covarianza. La struttura di comunicazione tra i nodi ROS 2, i topic impiegati e i flussi di dati è descritta nel Capitolo 6, unitamente all'architettura del sistema di localizzazione.

Il nodo di fusione sottoscrive `/encoder_twist` e `/imu` a una frequenza di 100 Hz, e il topic `/uwb` a 10 Hz. La stima odometrica risultante viene pubblicata sul topic `/odom` a 10 Hz, in coerenza con il rate del modulo UWB. Le covarianze di misura utilizzate

corrispondono ai valori riportati in Tabella 7.4, determinati sperimentalmente nella sezione precedente.

Entrambe le implementazioni includono un controllo di coerenza statistica basato sulla distanza di Mahalanobis e sul test χ^2 con livello di confidenza del 95%, introdotto nel Capitolo 4. Il gating consente di rigettare le misure UWB incoerenti con la predizione dello stato, migliorando la stabilità numerica e la robustezza del filtro in condizioni parzialmente NLOS.

Il filtro è stato dapprima prototipato in Python, sfruttando le librerie numeriche di `rclpy` e `NumPy` per una rapida validazione dei modelli, e successivamente riscritto in C++ per il funzionamento a bordo, dove la maggiore efficienza è necessaria a sostenere la frequenza di predizione in tempo reale sulla Jetson. In entrambe le versioni l'organizzazione del codice è la stessa e mantiene una netta separazione tra la logica di stima e l'interfaccia ROS. Una classe dedicata incapsula lo stato $[x, y, \psi, v]$ e la relativa matrice di covarianza, ed espone i due metodi fondamentali del filtro, la predizione e l'aggiornamento, insieme a una funzione di normalizzazione dell'angolo. Attorno a essa un nodo ROS gestisce le sottoscrizioni ai sensori, il timer di pubblicazione e la conversione tra i messaggi e il vettore di stato.

La sincronizzazione tra flussi a frequenze diverse è gestita in modo asincrono e guidato dagli eventi, senza ricorrere a un allineamento rigido dei messaggi. Ogni misura viene elaborata nel momento in cui arriva: i dati propriocettivi di encoder e IMU innescano la fase di predizione, mentre le posizioni assolute UWB, disponibili a frequenza inferiore, attivano la fase di aggiornamento. L'intervallo di integrazione Δt non è imposto a priori, ma ricavato dalla differenza tra i marcatori temporali di due messaggi consecutivi, così che la predizione resti corretta anche in presenza di jitter o di piccole variazioni nei tempi di arrivo. La pubblicazione dell'odometria avviene invece a cadenza fissa, regolata da un timer a 10 Hz indipendente dai callback dei sensori.

Sul piano numerico due accorgimenti garantiscono la stabilità della stima. L'angolo di imbardata viene riportato nell'intervallo $(-\pi, \pi]$ dopo ogni predizione, per evitare le discontinuità dovute all'avvolgimento; l'orientamento fornito dall'IMU come quaternion è convertito in angolo di Eulero in ingresso e nuovamente in quaternion per la pubblicazione. L'aggiornamento della covarianza è realizzato nella forma di Joseph, $(I - KH)P(I - KH)^T + K RK^T$, che rispetto alla forma semplificata preserva la simmetria e la semidefinizione positiva della matrice anche in presenza di errori di arrotondamento, requisito importante per l'esecuzione prolungata su aritmetica a precisione finita.

7.3. Validazione sperimentale del sistema di localizzazione

La validazione sperimentale ha consentito di verificare le prestazioni del sistema di localizzazione odometrica in condizioni operative analoghe a quelle della competizione.

L'analisi ha considerato stabilità numerica, coerenza spazio-temporale e consistenza statistica delle stime fornite dai filtri di fusione.

Le prove sono state articolate in due fasi successive a complessità crescente. La prima, condotta presso il laboratorio di Automazione dell'Ateneo UNIVPM, ha avuto funzione di collaudo e messa a punto del sistema, con l'obiettivo di validare l'integrazione sensoriale e la propagazione delle covarianze in ambiente controllato. Il circuito utilizzato riproduceva in scala ridotta le principali condizioni della competizione, consentendo di testare il comportamento del veicolo in scenari di mobilità urbana diversificati.

La seconda fase, svolta presso il Bosch Engineering Center di Cluj-Napoca, sede della competizione internazionale, ha previsto il test del sistema nella configurazione definitiva. Il tracciato, di dimensioni maggiori e con maggiore complessità dinamica, ha consentito di valutare la robustezza e la coerenza spaziale delle stime odometriche in presenza di disturbi ambientali e degrado parziale del segnale UWB.

Le sezioni seguenti riportano le evidenze sperimentali relative ai due scenari, comprendenti traiettorie stimate, distribuzioni di errore e metriche di accuratezza (RMSE, MAE ed errore massimo).

La traiettoria di riferimento impiegata come ground truth non deriva da alcun sensore di bordo, e in particolare non dall'UWB: utilizzare per il confronto lo stesso sistema oggetto di valutazione introdurrebbe una dipendenza circolare e priverebbe le metriche di significato. Il riferimento è stato invece ricavato dalla geometria nota del tracciato. La pista della competizione è definita metricamente da una rete di corsie i cui punti centrali, detti nodi, hanno coordinate note nella mappa. Durante le prove il veicolo è stato condotto mantenendo il centro corsia, percorrendo quindi la sequenza di nodi corrispondente al tragitto seguito; la traiettoria di riferimento è stata poi costruita interpolando tali nodi con una spline cubica, così da ottenere una curva continua e derivabile, coerente con la definizione metrica della pista.

L'incertezza di questo riferimento è limitata e quantificabile, poiché dipende da due soli contributi. Il primo è lo scostamento del veicolo dal centro corsia durante il mantenimento di traiettoria, contenuto entro pochi centimetri dal ridotto margine laterale della carreggiata in scala; il secondo è l'errore di interpolazione della spline tra nodi adiacenti, anch'esso dell'ordine del centimetro data la densità dei nodi e la regolarità del percorso. Nel complesso l'incertezza del ground truth si mantiene inferiore all'errore di localizzazione misurato, ed è quindi adeguata a validare un'accuratezza dell'ordine del decimetro: le metriche riportate nel seguito conservano pertanto piena significatività.

7.3.1. Validazione nel circuito dell'Ateneo UNIVPM

Il circuito di Figura 7.10 riproduce un contesto urbano con elementi caratteristici della mobilità stradale, tra cui gallerie, rotonde, incroci e tratti rettilinei di diversa lunghezza.

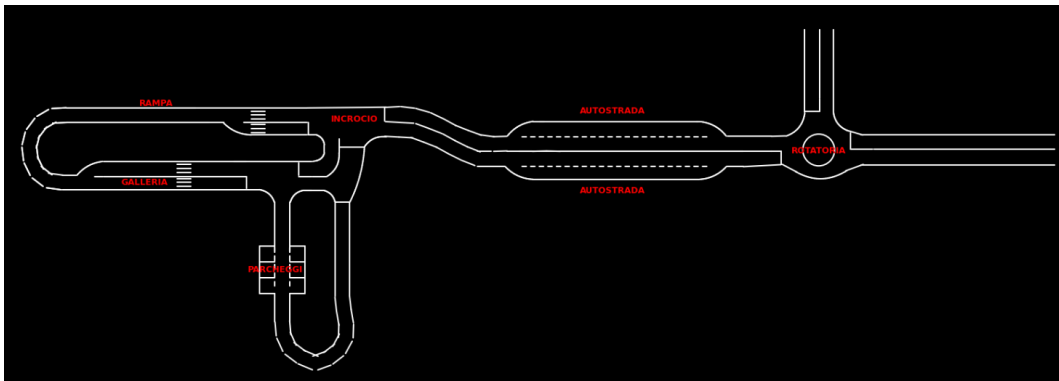


Figura 7.10.: Schema planimetrico del circuito realizzato presso l'Ateneo UNIVPM.

La varietà planimetrica ha permesso di analizzare la risposta del sistema in manovre di accelerazione, frenata e variazioni di sterzo, verificando la stabilità numerica dei filtri gaussiani non lineari e la coerenza delle covarianze di misura determinate in fase di calibrazione.

La Figura 7.11 mostra il confronto tra la traiettoria di riferimento (ground truth) e le traiettorie stimate dai due filtri di fusione. Entrambi gli stimatori mantengono una buona sovrapposizione con la traiettoria reale, con scostamenti limitati nelle sezioni a curvatura elevata. L'EKF risulta più stabile nella stima dell'orientamento, mentre l'UKF mostra lievi oscillazioni imputabili alla maggiore sensibilità alla scelta dei parametri di processo.

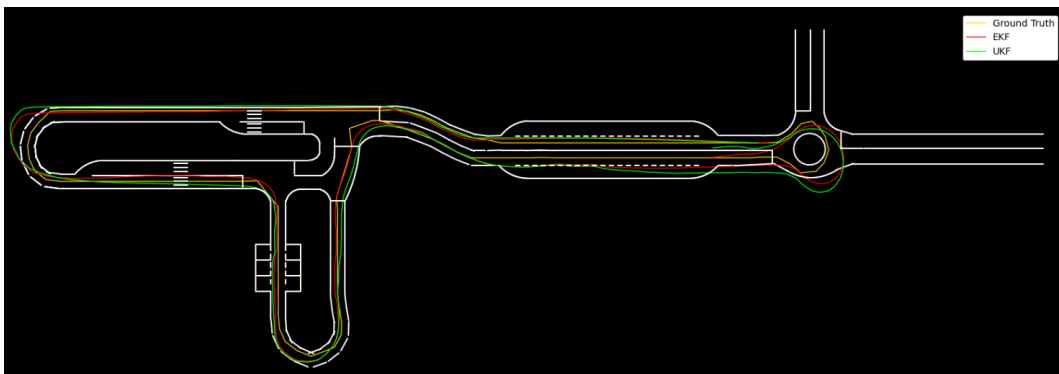


Figura 7.11.: Confronto tra la traiettoria di riferimento e le stime EKF e UKF.

L'andamento temporale dell'errore di localizzazione è riportato in Figura 7.12. L'EKF presenta un errore medio inferiore a 0.2 m lungo la maggior parte del tracciato, con incrementi fino a 0.5 m in corrispondenza delle curve più strette, dove l'errore odometrico tende ad accumularsi. L'UKF mostra un profilo più irregolare, con variazioni localizzate associate alle fasi di sterzo rapido.

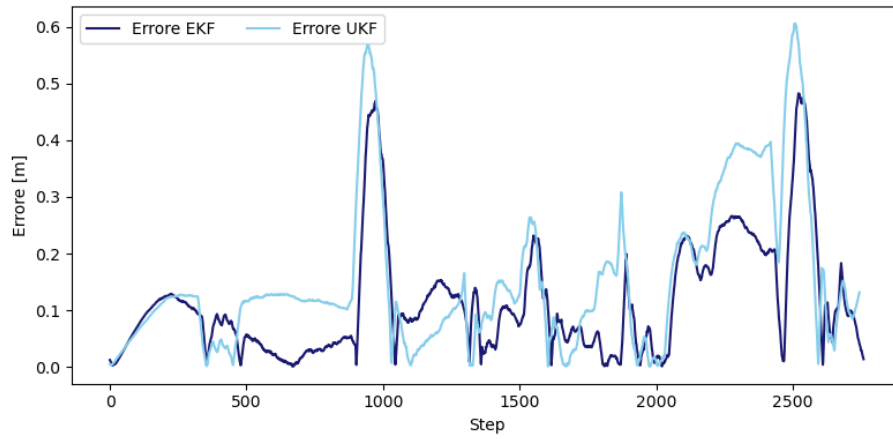


Figura 7.12.: Andamento temporale dell'errore di localizzazione per EKF e UKF.

La distribuzione cumulativa dell'errore (Figura 7.13) evidenzia che circa l'80% delle stime EKF ricade entro 0.2 m, mentre l'UKF raggiunge valori simili con una dispersione più ampia. Ciò conferma una maggiore robustezza dell'approssimazione lineare rispetto alla propagazione unscented nel contesto cinematico considerato.

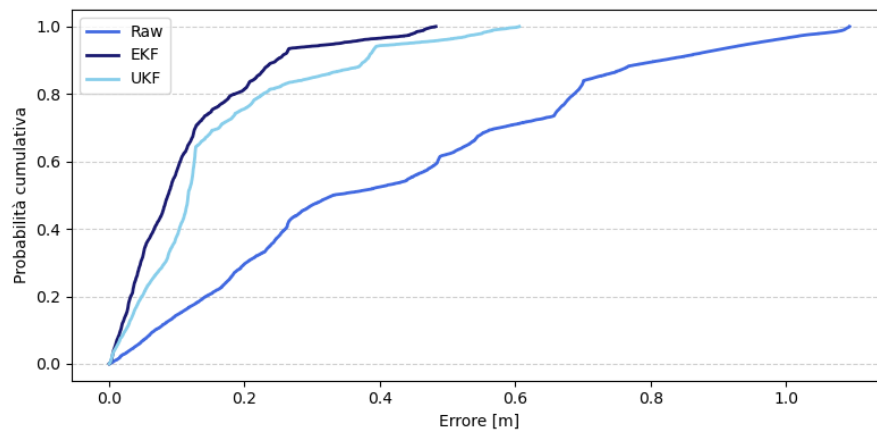


Figura 7.13.: Distribuzione cumulativa dell'errore per odometria grezza, EKF e UKF.

La distribuzione di frequenza in Figura 7.14 mostra la riduzione della varianza ottenuta mediante fusione sensoriale rispetto all'odometria grezza. La concentrazione dei valori entro 0.2 m conferma la corretta calibrazione dei parametri di misura e l'efficacia della modellazione probabilistica adottata.

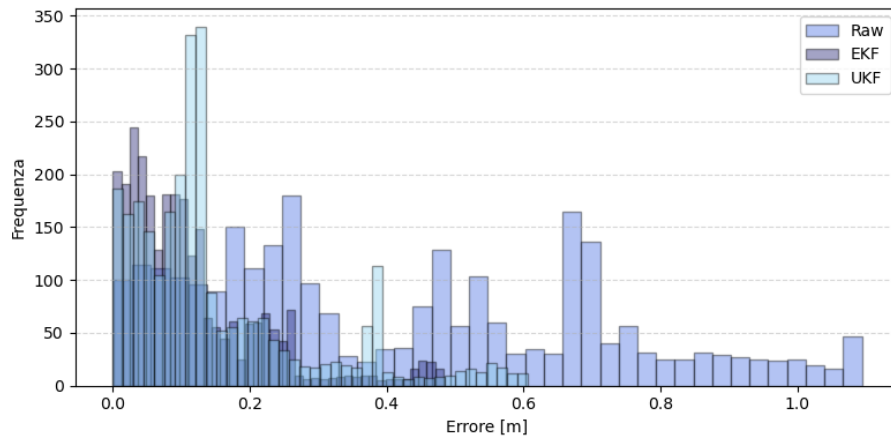


Figura 7.14.: Distribuzione di frequenza dell'errore di localizzazione per odometria grezza, EKF e UKF.

Le metriche sintetiche sono riportate in Tabella 7.5. L'EKF ha conseguito un RMSE pari a 0.153 m e un MAE di 0.114 m, risultando più preciso e stabile rispetto all'UKF, che presenta valori di 0.204 m e 0.154 m. Entrambi gli stimatori migliorano sensibilmente le prestazioni rispetto all'odometria grezza, che evidenzia un errore medio superiore a 0.4 m.

Tabella 7.5.: Metriche di accuratezza della localizzazione nel circuito UNIVPM.

Traiettoria	RMSE [m]	MAE [m]	Errore max [m]
Odometria grezza	0.5039	0.4144	1.0939
EKF	0.1535	0.1143	0.4829
UKF	0.2039	0.1542	0.6061

7.3.2. Validazione presso il Bosch Engineering Center

La seconda fase di validazione è stata condotta sul tracciato di Figura 7.15, di estensione e complessità superiori rispetto al circuito universitario. Il percorso include marcature discontinue, variazioni di illuminazione e sezioni con copertura parziale del segnale UWB, condizioni che riproducono fedelmente il contesto operativo della competizione.

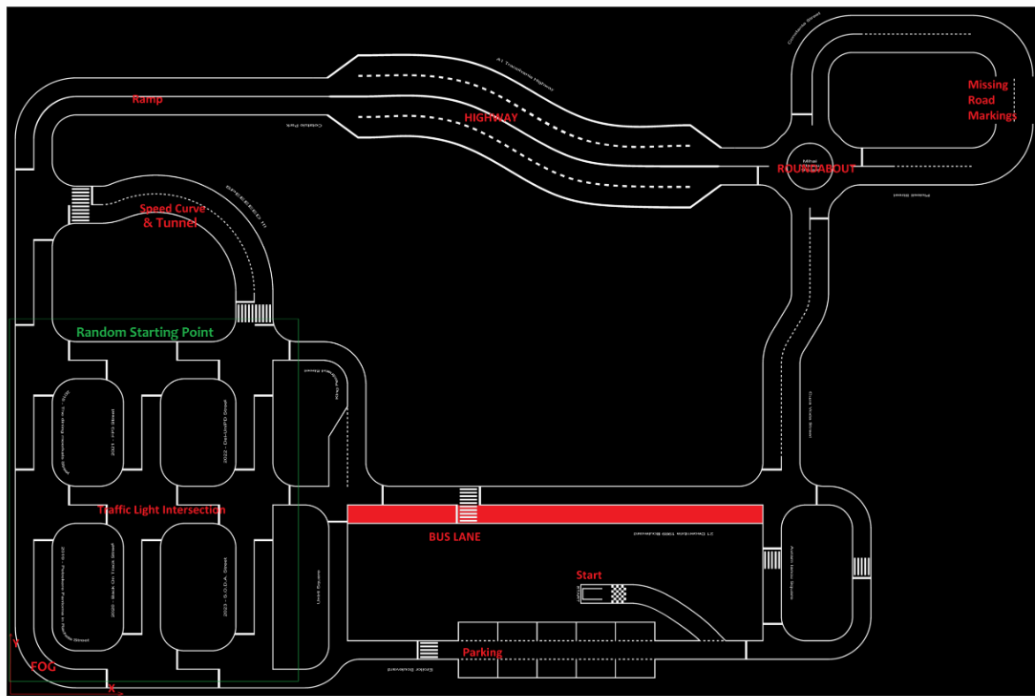


Figura 7.15.: Schema planimetrico del tracciato presso il Bosch Engineering Center di Cluj-Napoca.

La Figura 7.16 riporta il confronto tra la traiettoria di riferimento e le stime dei due filtri. L'EKF mostra una traiettoria regolare e ben allineata al percorso reale, mentre l'UKF evidenzia deviazioni localizzate in corrispondenza di curve a raggio ridotto e cambi di pendenza, dove la propagazione delle incertezze risente maggiormente della non linearità del modello.

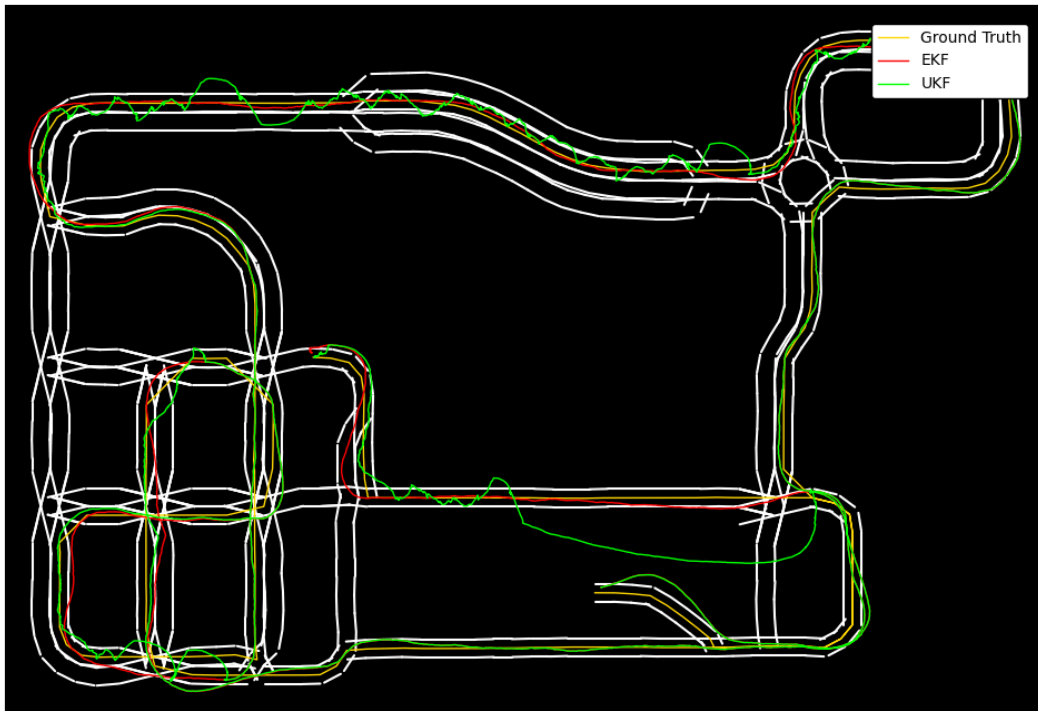


Figura 7.16.: Confronto tra la traiettoria di riferimento e le stime EKF e UKF.

L'andamento temporale dell'errore, riportato in Figura 7.17, conferma l'efficacia del filtro EKF, che mantiene valori medi inferiori a 0.2 m per la maggior parte del tracciato. L'UKF presenta oscillazioni fino a 1.3 m nelle zone di maggiore complessità dinamica, evidenziando una minore stabilità numerica.

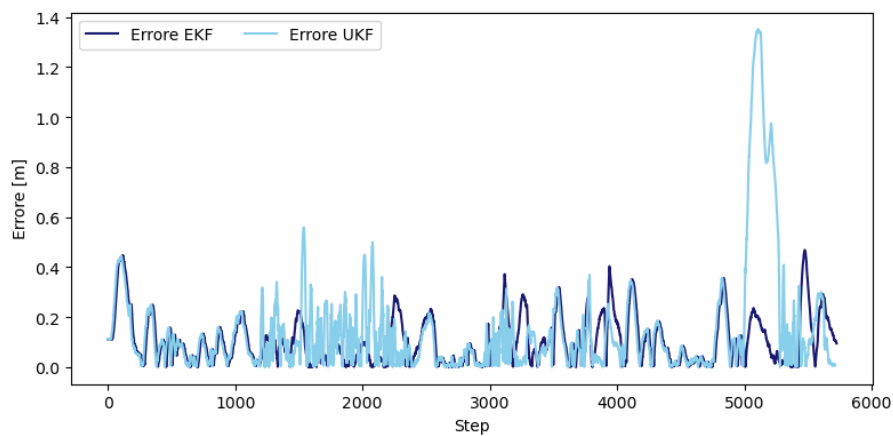


Figura 7.17.: Andamento temporale dell'errore di localizzazione per EKF e UKF.

La distribuzione cumulativa dell'errore (Figura 7.18) mostra che oltre il 90% delle stime EKF rientra entro 0.3 m, mentre l'UKF presenta una coda più estesa oltre 1 m, dovuta alla propagazione delle incertezze in condizioni non stazionarie.

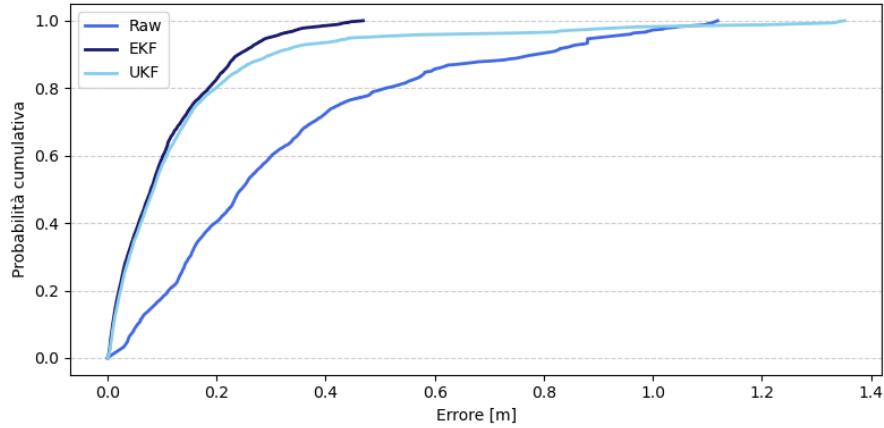


Figura 7.18.: Distribuzione cumulativa dell'errore per odometria grezza, EKF e UKF.

La distribuzione di frequenza (Figura 7.19) conferma la riduzione della varianza per entrambi i filtri rispetto ai dati grezzi. L'EKF concentra la maggior parte delle stime entro 0.2 m, mentre l'UKF mostra una dispersione più elevata dovuta all'effetto dei sigma-point su rumori non gaussiani.

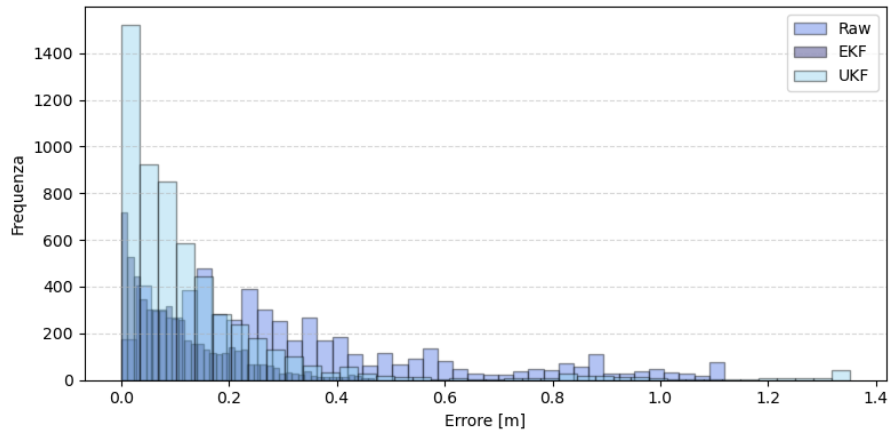


Figura 7.19.: Distribuzione di frequenza dell'errore per odometria grezza, EKF e UKF.

Le metriche sintetiche, riportate in Tabella 7.6, confermano le prestazioni superiori dell'EKF rispetto all'UKF. Entrambi i filtri riducono l'errore medio di oltre il 60% rispetto all'odometria grezza, validando l'efficacia del modello di fusione sensoriale.

Tabella 7.6.: Metriche di accuratezza della localizzazione nel tracciato Bosch Engineering Center.

Traiettoria	RMSE [m]	MAE [m]	Errore max [m]
EKF	0.1430	0.1271	0.4687
UKF	0.2557	0.1753	1.3514
Odometria grezza	0.4192	0.3242	1.1187

7.3.3. Analisi di consistenza del filtro

RMSE e MAE quantificano l'accuratezza della stima rispetto alla traiettoria di riferimento, ma non dicono se l'incertezza dichiarata dal filtro sia realistica. Un filtro di Kalman è consistente quando l'errore di stima è statisticamente compatibile con la covarianza che il filtro associa alla stima stessa: né troppo ottimista, sottostimando l'incertezza, né troppo prudente. La verifica si fonda su due indicatori normalizzati, l'errore quadratico normalizzato sullo stato (NEES) e quello sull'innovazione (NIS). Il NEES confronta l'errore di stima con la covarianza di stato,

$$\epsilon_k = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^\top P_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k),$$

dove $\mathbf{x}_k$ è lo stato reale, $\hat{\mathbf{x}}_k$ la stima e P_k la covarianza associata. Il NIS è invece definito sulla sola innovazione di misura,

$$\nu_k = (\mathbf{z}_k - \hat{\mathbf{z}}_k)^\top S_k^{-1} (\mathbf{z}_k - \hat{\mathbf{z}}_k),$$

con S_k covarianza dell'innovazione. Sotto le ipotesi di linearità e rumore gaussiano entrambi seguono una distribuzione chi-quadro con gradi di libertà pari alla dimensione del rispettivo vettore, e valore atteso uguale a tale dimensione: $\mathbb{E}[\epsilon_k] = n = 4$ per lo stato $[x, y, \psi, v]$ e $\mathbb{E}[\nu_k] = m = 2$ per la misura di posizione UWB. Il NIS non richiede la conoscenza dello stato reale e può quindi essere monitorato in linea durante il normale funzionamento.

Nel sistema sviluppato il NIS è già parte integrante del filtro: il gating del Capitolo 4, basato sulla distanza di Mahalanobis e sul test chi-quadro al 95%, calcola a ogni aggiornamento UWB la quantità ν_k e la confronta con la soglia teorica $\chi_{2,0.95}^2 = 5,99$ (Tabella 4.2), scartando le osservazioni incoerenti con la predizione. La consistenza è stata verificata a posteriori rieseguendo l'EKF sul recording finale, costituito da $N \approx 2600$ aggiornamenti UWB su circa 280 s, con la covarianza di misura adottata $\sigma_{\text{UWB}} \approx 0,21$ m. I risultati sono riassunti in Tabella 7.7.

A livello di innovazione il filtro risulta consistente: il 89,6% degli aggiornamenti cade entro il gate $\chi_{2,0.95}^2 = 5,99$ e il NIS medio vale 2,8, dell'ordine del valore atteso 2. Il NIS mediano, pari a 0,6, è nettamente inferiore al valore atteso perché la covarianza

efficace $R_{\text{UWB,eff}}$ è volutamente conservativa e pesata sul regime NLOS: per le misure in visibilità diretta, prevalenti, le innovazioni restano ben al di sotto della soglia, mentre il restante 10,4 % di aggiornamenti rigettati coincide con le letture NLOS più degradate. La coerenza tra il NIS misurato e il modello di rumore conferma quindi la scelta della covarianza di misura.

Il NEES è stato valutato confrontando la stima con la traiettoria di riferimento, normalizzando l'errore con la covarianza complessiva, somma di quella del filtro e dell'incertezza propria del riferimento, dello stesso ordine del rumore UWB. Sul recording finale il NEES medio in posizione vale 1,8, prossimo al valore atteso 2 e interno all'intervallo di consistenza: l'errore di stima è quindi statisticamente compatibile con la covarianza dichiarata dal filtro. Insieme al NIS, questo conferma la consistenza complessiva dell'EKF e ne supporta la scelta come filtro di riferimento per la navigazione.

Tabella 7.7.: Indicatori di consistenza dell'EKF sul recording finale ($N \approx 2600$ aggiornamenti UWB, ~ 280 s).

Indicatore	Valore	Riferimento
NIS medio	2,8	atteso 2
NIS mediano	0,6	atteso 2
Innovazioni entro il gate $\chi^2_{2,0.95} = 5,99$	89,6 %	95 %
Aggiornamenti rigettati (NLOS)	10,4 %	atteso ~ 5 %
NEES posizione medio	1,8	atteso 2

7.3.4. Analisi complessiva

I risultati sperimentali mostrano una chiara coerenza tra i modelli cinematici implementati e le prestazioni osservate nei due scenari di test. L'EKF (Extended Kalman Filter) ha evidenziato una stabilità numerica superiore e un errore quadratico medio (RMSE) contenuto entro i 0.15 m.

Tale prestazione, superiore a quella riscontrata con l'UKF (Unscented Kalman Filter), è riconducibile all'elevata frequenza di campionamento adottata, pari a 100 Hz. Riducendo l'ampiezza dell'intervallo temporale Δt , il sistema opera infatti in un regime di quasi-linearità dove l'approssimazione del primo ordine tramite Jacobiani risulta estremamente accurata. Al contrario, l'UKF ha manifestato una variabilità superiore, causata dalla spiccata sensibilità dei sigma-points verso il rumore non gaussiano e gli outlier tipici dei segnali UWB in ambienti indoor, rendendo il tuning dei parametri di scaling (α, β, κ) eccessivamente critico in contesti dinamici.

La fusione sensoriale tra encoder, IMU e UWB ha ridotto significativamente la varianza dell'errore rispetto all'odometria grezza, confermando la validità dell'approccio probabilistico. Tuttavia, la natura ricorsiva della stima odometrica comporta un accumulo progressivo dell'errore e una deriva temporale della posa (drift). Si rende

pertanto necessaria l'integrazione con un modulo di localizzazione globale basato su LiDAR, in grado di fornire riferimenti assoluti e compensare la deriva mediante aggiornamenti periodici della mappa.

Sulla base di queste evidenze, l'EKF è stato selezionato come algoritmo di riferimento per la navigazione, in virtù della sua robustezza computazionale e stabilità su hardware embedded. Entrambe le architetture rimangono comunque disponibili nel sistema, garantendo una piattaforma flessibile e modulare. L'integrazione di tale stima all'interno dello stack di navigazione tramite SLAM e AMCL è approfondita nel Capitolo 8.

Capitolo 8.

Implementazione del sistema di localizzazione globale

8.1. Implementazione dello SLAM

La mappatura dell'ambiente è stata realizzata mediante il pacchetto SLAM Toolbox del framework ROS 2 Navigation2, documentato in [25]. Tale pacchetto implementa un approccio di mappatura bidimensionale basato su grafi di pose, in cui ogni nodo rappresenta una stima della posizione del veicolo e gli archi corrispondono a vincoli derivati dalla correlazione tra scansioni laser successive. L'algoritmo consente di costruire in modo incrementale una mappa di occupazione dell'ambiente e di eseguire la chiusura automatica dei loop quando il veicolo ritorna in aree già esplorate.

SLAM Toolbox è stato scelto per la capacità di operare in modalità asincrona, consentendo la costruzione della mappa in tempo reale durante la navigazione, e per la compatibilità con l'architettura Navigation2 già impiegata per la pianificazione e la localizzazione. Inoltre, il pacchetto include un risolutore di ottimizzazione configurabile (basato su Ceres Solver) che consente una gestione robusta dei vincoli anche in presenza di rumore odometrico, condizione tipica dei veicoli in scala ridotta. Il nodo di mappatura riceve in ingresso le scansioni bidimensionali fornite dal sensore LiDAR LD19 con frequenza di 10 Hz e la stima odometrica pubblicata sul topic odom, generata dal filtro di fusione sensoriale descritto nel Capitolo 7. La trasformazione tra i sistemi di riferimento map, odom e base_link è gestita dal sistema tf2, con periodo di aggiornamento pari a 0.02 s, garantendo coerenza temporale tra le misure.

8.1.1. Parametrizzazione e tuning

La configurazione del nodo SLAM Toolbox è stata oggetto di taratura sperimentale per ottimizzare il compromesso tra accuratezza della mappa e carico computazionale. I parametri principali e le motivazioni delle scelte adottate sono riportati nella Tabella 8.1.

Tabella 8.1.: Parametri principali di configurazione del nodo SLAM Toolbox.

Parametro	Motivazione
<code>solver_plugin = CeresSolver</code>	Utilizza il risolutore non lineare Ceres con schema Levenberg–Marquardt, garantendo stabilità numerica e convergenza in presenza di rumore odometrico.
<code>resolution = 0.03 m</code>	Imposta la risoluzione della griglia di occupazione; il valore rappresenta un compromesso tra dettaglio e carico computazionale.
<code>minimum_travel_distance = 0.3 m</code> , <code>minimum_travel_heading = 0.2 rad</code>	Limitano la creazione di nuove pose per piccoli spostamenti o rotazioni, riducendo la densità del grafo in condizioni di moto lento.
<code>use_scan_matching = true</code>	Attiva l'allineamento delle scansioni consecutive, fondamentale in ambienti con pochi landmark.
<code>loop_search_maximum_distance = 3.0 m</code>	Definisce la distanza massima per la ricerca di loop closure, evitando associazioni spurie in ambienti simmetrici.
<code>do_loop_closing = true</code>	Abilita la chiusura automatica dei loop mediante correlazione globale delle scansioni.
<code>loop_match_minimum_response_fine = 0.45</code>	Impone una soglia minima di correlazione per validare la chiusura del loop, prevenendo errori di sovrapposizione.
<code>loop_match_maximum_variance_coarse = 3.0</code>	Limita la varianza ammissibile nella fase di ricerca grossolana del loop closure, aumentando la robustezza.
<code>correlation_search_space_resolution = 0.01 m</code>	Controlla la granularità della ricerca locale di correlazione; consente una maggiore precisione di allineamento tra scansioni.
<code>distance_variance_penalty = 0.5</code> , <code>angle_variance_penalty = 1.0</code>	Introducono penalità sui vincoli di distanza e orientamento, stabilizzando la stima in presenza di rumore angolare.
<code>fine_search_angle_offset = 0.00349 rad</code> , <code>coarse_search_angle_offset = 0.349 rad</code>	Definiscono gli offset angolari per la ricerca fine e grossolana di correlazione, determinando la risoluzione del matching rotazionale.
<code>minimum_distance_penalty = 0.5</code> , <code>minimum_angle_penalty = 0.9</code>	Regolano la sensibilità del matcher nei confronti di piccole variazioni di posizione e orientamento.

I valori riportati in Tabella 8.1 sono il risultato di una campagna di prove sperimentali e non di una semplice impostazione manuale. La taratura è stata condotta in laboratorio su un banco di prova ripetibile, che riproduceva in scala le condizioni di percorso e permetteva di confrontare le diverse configurazioni a parità di tracciato. Partendo dai valori consigliati dalla documentazione del pacchetto, si è proceduto con un'analisi di sensibilità sistematica, agendo su un parametro per volta e lasciando invariati gli altri, in modo da attribuirne con certezza l'effetto. Ogni impostazione è stata giudicata su grandezze misurabili: la coerenza geometrica della mappa e l'allineamento delle chiusure di loop, l'errore rispetto alla traiettoria di riferimento e il carico di calcolo sostenibile in tempo reale dalla Jetson. I valori finali sono quelli che, in modo ripetibile, hanno garantito il miglior compromesso tra accuratezza e costo computazionale, e sono stati poi confermati sul campo nei due circuiti di prova. Durante le prove preliminari si è osservato che risoluzioni inferiori a 0.03 m incrementavano il tempo medio di elaborazione senza apportare un miglioramento proporzionale dell'accuratezza. Allo stesso modo, valori di penalità inferiori producevano instabilità nella ricostruzione della mappa, mentre valori troppo elevati riducevano la capacità di adattamento nelle curve strette.

Il nodo è stato configurato in modalità online asincrona, con costruzione incrementale della mappa durante la navigazione del veicolo. La mappa generata è rappresentata come griglia di occupazione, nella quale a ciascuna cella è associata la probabilità di essere occupata secondo il modello introdotto nel Capitolo 5.

8.1.2. Risultati operativi

La costruzione e la verifica della mappa sono state articolate in due ambienti con caratteristiche geometriche differenti, corrispondenti a due fasi successive del processo di sviluppo. Nel circuito interno dell'Ateneo UNIVPM l'algoritmo è stato utilizzato per la taratura e la stabilizzazione del sistema di mappatura, mentre presso il Bosch Engineering Center di Cluj-Napoca è stato impiegato in configurazione definitiva, con adattamento parametrico a un contesto operativo più esteso e complesso.

Mappatura nel circuito UNIVPM

Il circuito universitario, costituito da un corridoio principale lungo circa 15 m e largo 3 m con intersezione a T nella parte centrale, è stato impiegato per verificare la coerenza spaziale della mappa e la stabilità numerica del grafo di pose in un ambiente controllato, simmetrico e privo di landmark univoci. La Figura 8.1 mostra la griglia di occupazione generata da SLAM Toolbox con risoluzione di 0.03 m.

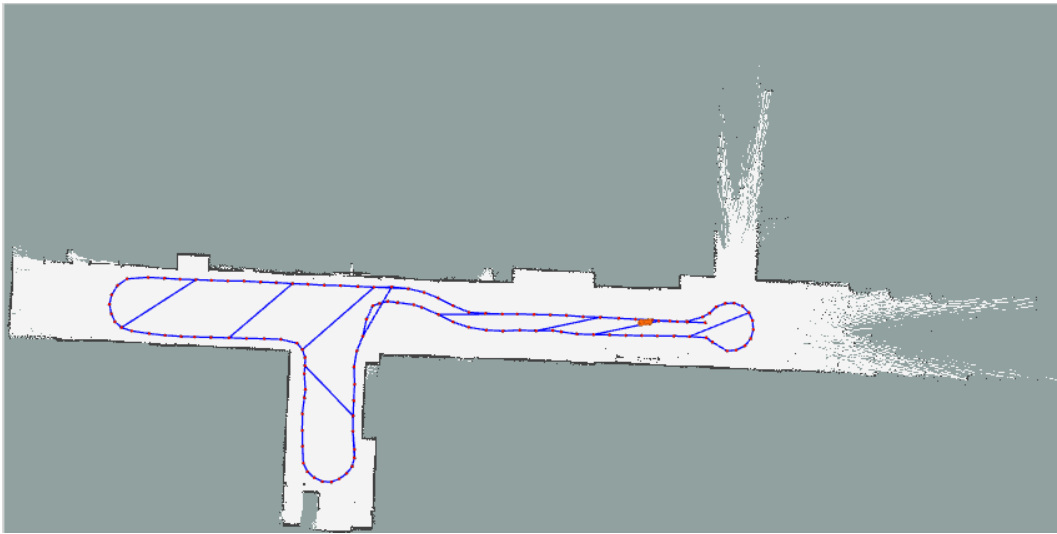


Figura 8.1.: Mappa di occupazione generata nel circuito dell'Ateneo con il grafo delle pose e i vincoli probabilistici.

Le celle bianche rappresentano superfici libere, le nere ostacoli rilevati dal sensore LiDAR e le grigie regioni non esplorate. Le linee blu indicano il grafo delle pose e i vincoli probabilistici derivati dalla correlazione tra scansioni. La mappa presenta elevata coerenza geometrica fra le sezioni percorse, assenza di disallineamenti nelle chiusure di loop e distribuzione regolare delle pose. Le pareti e le intersezioni risultano ben definite, evidenziando la corretta convergenza dell'ottimizzazione e la stabilità della taratura per ambienti caratterizzati da simmetria e geometrie ripetitive.

Mappatura presso il Bosch Engineering Center

L'algoritmo, già validato nel circuito UNIVPM, è stato successivamente impiegato presso il Bosch Engineering Center. L'area, di dimensioni pari a circa 20×14 m, presentava maggiore eterogeneità geometrica, con pareti, arredi e ostacoli di diversa natura. In tale contesto operativo è stato necessario adattare alcuni parametri del pacchetto SLAM Toolbox per mantenere la stabilità in tempo reale e contenere il carico computazionale, preservando la consistenza globale della mappa.

In particolare:

- `minimum_travel_distance` è stato incrementato a 0.5 m per generare nuove pose solo dopo spostamenti significativi;
- `correlation_search_space_resolution` è stato portato a 0.02 m per ridurre la complessità computazionale in uno spazio più ampio;
- `loop_search_maximum_distance` è stato aumentato a 5.0 m e `scan_buffer_size` a 20, estendendo la finestra di scansioni per la chiusura dei loop;
- la soglia `loop_match_minimum_response_fine` è stata ridotta a 0.40 per incrementare la sensibilità alle chiusure in presenza di variabilità geometrica;

- `use_scan_matching` è stato disattivato, evitando vincoli locali ridondanti in un ambiente ricco di landmark distintivi.

La mappa risultante, illustrata in Figura 8.2, mostra una rappresentazione coerente delle strutture principali. Le superfici riflettenti lungo il perimetro in vetro hanno generato parziali duplicazioni delle linee di bordo, fenomeno tuttavia non sufficiente a compromettere la consistenza del grafo di pose, che ha mantenuto stabilità per l'intera sessione di acquisizione.



Figura 8.2.: Mappa di occupazione generata presso il Bosch Engineering Center, con evidenza delle pose e dei vincoli probabilistici.

L'analisi complessiva mostra che l'algoritmo, dopo la taratura eseguita nel circuito

UNIVPM, ha mantenuto stabilità e coerenza anche in un ambiente di scala e complessità superiori. La pipeline SLAM Toolbox ha garantito una rappresentazione spaziale accurata e robusta, idonea alla successiva fase di localizzazione basata su AMCL.

8.2. Implementazione dell'AMCL

La localizzazione del veicolo autonomo è stata implementata mediante il pacchetto AMCL del framework ROS 2 Navigation2 [26]. Tale componente adotta un filtro di particelle a densità adattiva per stimare la posa bidimensionale del robot all'interno di una mappa di occupazione nota, aggiornando in modo iterativo un insieme di ipotesi di stato (particelle) pesate in funzione della probabilità di osservare una determinata scansione laser rispetto alla mappa fornita. L'insieme delle particelle costituisce quindi una rappresentazione discreta della distribuzione di probabilità della posa, la cui evoluzione riflette la coerenza tra odometria e osservazioni sensoriali. Come descritto nel Capitolo 5, l'AMCL integra due modelli fondamentali: il modello di movimento, derivato dall'odometria, e il modello di osservazione, basato sulle misure di distanza fornite dal sensore LiDAR. Il primo aggiorna la stima della posa in base alle variazioni di velocità e orientamento del veicolo, mentre il secondo corregge tale stima confrontando le scansioni acquisite con la mappa di riferimento. Il nodo di localizzazione riceve in ingresso la mappa generata tramite SLAM Toolbox e il topic *odom* prodotto dal sistema di fusione sensoriale descritto nel Capitolo 7, garantendo la continuità informativa tra i moduli di mappatura, navigazione e controllo.

Il pacchetto è stato selezionato per la sua integrazione nativa con l'architettura Navigation2 e per la capacità di adattare dinamicamente la densità di particelle in funzione dell'incertezza stimata. Tale caratteristica consente di bilanciare il compromesso tra accuratezza e complessità computazionale: durante la localizzazione stabile, il numero di particelle si riduce progressivamente, mentre in presenza di ambiguità o divergenze aumenta automaticamente, assicurando robustezza numerica e tempi di convergenza contenuti anche in scenari caratterizzati da rumore o disturbi ambientali.

8.2.1. Configurazione e tuning dei parametri

La configurazione del nodo AMCL è stata definita sulla base della mappa ottenuta con SLAM Toolbox e delle caratteristiche cinematiche del veicolo. Il processo di taratura ha mirato a ottimizzare la precisione di localizzazione senza compromettere la frequenza di aggiornamento del filtro. In particolare, l'attenzione è stata rivolta alla definizione dei parametri relativi al modello odometrico, al modello di misura e alla gestione adattiva delle particelle, con l'obiettivo di assicurare consistenza statistica e stabilità temporale della stima.

La Tabella 8.2 riporta i principali parametri di configurazione, accompagnati dalle relative motivazioni tecniche.

Anche per l'AMCL i parametri sono stati ricavati con lo stesso metodo sperimentale impiegato per lo SLAM. A partire dalla configurazione di riferimento del pacchetto, ciascun parametro è stato esplorato singolarmente attraverso prove ripetute, misurando di volta in volta l'errore rispetto alla traiettoria di riferimento e la frequenza di aggiornamento del filtro. Le considerazioni esposte di seguito, relative al numero di particelle, al modello di misura e al gating dei fasci, riportano le evidenze raccolte in questa analisi e motivano i valori adottati.

Tabella 8.2.: Parametri principali di configurazione del nodo AMCL.

Parametro	Motivazione
alpha1 = 0.2, alpha2 = 0.2, alpha3 = 0.2, alpha4 = 0.2	Coefficienti del modello odometrico che rappresentano la deviazione standard proporzionale alla rotazione e alla traslazione; valori moderati coerenti con la bassa velocità operativa del veicolo e la sua stabilità meccanica.
min_particles = 500, max_particles = 2000	Definiscono l'intervallo di variabilità del numero di particelle; assicurano efficienza computazionale nelle fasi stazionarie e capacità di recupero in caso di perdita di tracking.
robot_model_type = nav2_amcl::DifferentialMotionModel	Specifica il modello cinematico differenziale, adottato come approssimazione per compatibilità con i plugin standard di Navigation2. Il veicolo ha cinematica Ackermann; l'approssimazione è giustificata dalle ridotte velocità operative e dall'elevata frequenza di correzione LiDAR, che compensa il bias cinematico introdotto nella fase di predizione.
laser_model_type = likelihood_field	Impiega un modello probabilistico basato sul campo di distanze precomputato nella mappa, migliorando la robustezza in presenza di superfici riflettenti.
sigma_hit = 0.2, z_hit = 0.5	Parametri del modello di misura che definiscono la deviazione standard e il peso associato alle osservazioni coerenti con la mappa.
laser_likelihood_max_dist = 2.0 m	Limita la distanza massima considerata nel calcolo della probabilità, riducendo l'influenza di misure rumorose a lunga portata.
update_min_d = 0.3 m, update_min_a = 0.2 rad	Impongono una soglia minima di traslazione e rotazione per l'aggiornamento del filtro, evitando oscillazioni spurie.
pf_err = 0.05, pf_z = 0.99	Regolano la convergenza del filtro di particelle, mantenendo la sopravvivenza delle ipotesi con peso elevato.
do_beamskip = false (UNIVPM), true (Cluj)	Parametro dipendente dall'ambiente: disabilitato nel circuito UNIVPM statico, dove tutti i raggi contribuiscono all'aggiornamento; abilitato presso il Bosch Engineering Center per escludere automaticamente i raggi affetti da occlusioni, riflessioni e ostacoli temporanei.

Un numero eccessivo di particelle incrementa sensibilmente il carico computazionale, riducendo la frequenza di aggiornamento del filtro, mentre una densità troppo bassa può generare divergenze in presenza di rumore LiDAR o di slittamento delle ruote motrici. A seguito di diverse sessioni di calibrazione, è stato adottato un intervallo compreso tra 500 e 2000 particelle, che rappresenta il miglior compromesso tra accuratezza e stabilità.

Le calibrazioni hanno mostrato che l'aumento del numero massimo di particelle oltre 2000 non produce miglioramenti apprezzabili dell'errore medio di localizzazione, ma comporta un incremento significativo del tempo di aggiornamento del filtro. L'impiego del modello di misura `likelihood_field` ha consentito di mantenere una stima stabile anche in presenza di superfici riflettenti o parzialmente trasparenti, come le pareti vetrate del laboratorio Bosch. Tale modello, basato sulla correlazione tra le misure acquisite e un campo di distanze precomputato, riduce la sensibilità del sistema a misure incoerenti e consente una migliore gestione delle aree con informazione parziale.

Un ruolo determinante è stato assunto dal parametro `do_beamskip`, la cui abilitazione si è rivelata essenziale nelle fasi operative caratterizzate da elevata dinamicità ambientale, come quelle condotte presso il Bosch Engineering Center. L'attivazione di tale opzione permette di escludere automaticamente dal processo di aggiornamento i raggi LiDAR affetti da occlusioni, ostacoli temporanei o riflessioni, preservando la consistenza del filtro anche in presenza di disturbi locali. Questa configurazione ha garantito stabilità e continuità della localizzazione in condizioni reali di navigazione, caratterizzate da traffico di veicoli, assicurando la corretta convergenza del filtro di particelle rispetto alla mappa generata tramite SLAM Toolbox.

La scelta di un modello differenziale merita una precisazione, dal momento che il veicolo possiede una cinematica di tipo Ackermann. Il pacchetto AMCL di Navigation2 rende disponibili soltanto due modelli di moto, quello differenziale e quello omnidirezionale, e non prevede un modello specifico per la sterzata Ackermann [26]. Va però sottolineato che questi modelli non descrivono la dinamica del veicolo a partire dalle velocità delle ruote o dall'angolo di sterzo, ma operano direttamente sull'odometria: ricevono in ingresso soltanto lo spostamento relativo tra due pose successive, calcolato a monte dal modulo di fusione sensoriale. Poiché tale modulo, presentato nei Capitoli 4 e 7, si basa sul modello a bicicletta Ackermann [3], la geometria di sterzata del veicolo risulta già incorporata nella stima odometrica e non viene quindi persa durante la localizzazione.

Il modello odometrico descrive lo spostamento tra due pose come una rotazione iniziale, una traslazione e una rotazione finale. Questa scomposizione vale per qualunque moto rigido nel piano, indipendentemente dal meccanismo che lo ha generato, ed è quindi compatibile anche con una traiettoria prodotta da sterzata Ackermann. L'approssimazione introdotta non riguarda dunque la traiettoria stimata, ma solo il modo in cui viene distribuita l'incertezza di processo: i quattro coefficienti di rumore del modello ripartiscono la varianza tra le componenti di rotazione e di

traslazione secondo le sorgenti d'errore tipiche di un azionamento differenziale. In altre parole, l'aggettivo differenziale si riferisce alla sola modellazione del rumore e non a un vincolo geometrico imposto al moto del veicolo.

Il limite più noto di questa approssimazione, vale a dire la comparsa di incertezze angolari troppo elevate che spingono le particelle in configurazioni non raggiungibili dal veicolo, si manifesta soprattutto alle alte velocità, indicativamente oltre i 5 m/s, come riportato negli studi sulla localizzazione per la guida autonoma da competizione [4]. Alle velocità ridotte tipiche del contesto BFMC questo effetto è trascurabile. Per averne una stima quantitativa si può fare riferimento alla soglia minima di traslazione che innesca l'aggiornamento del filtro, pari a 0,3 m: con un coefficiente di rumore di 0,2, lo scostamento laterale consentito dal modello differenziale ma non ammesso dalla cinematica Ackermann resta dell'ordine di pochi centimetri a ciclo, e viene comunque corretto a ogni passo dal confronto tra scansione LiDAR e mappa, eseguito a 10 Hz.

Una conferma sperimentale di quanto detto si trova negli andamenti dell'errore nel tempo (Figure 8.4 e 8.7), che non mostrano alcuna deriva progressiva: l'errore oscilla attorno a un valore medio stabile e inferiore a 0,10 m, in linea con l'accuratezza dichiarata. Questo indica che la scelta del modello non introduce un errore sistematico apprezzabile. Resta comunque possibile, come sviluppo futuro orientato a velocità operative maggiori, realizzare un modello di moto Ackermann dedicato oppure ricorrere al modello omnidirezionale, di cui la cinematica Ackermann rappresenta un caso particolare.

8.2.2. Risultati sperimentali

La validazione del sistema di localizzazione mediante AMCL è stata condotta in due ambienti con caratteristiche geometriche e dinamiche differenti, al fine di valutare la robustezza dell'algoritmo in condizioni operative rappresentative del contesto applicativo. Entrambi gli scenari impiegano la mappa di occupazione generata tramite SLAM Toolbox e la stima odometrica derivata dal filtro di fusione sensoriale, assicurando coerenza metodologica tra le diverse fasi della pipeline di navigazione.

Circuito dell'Ateneo

La prima fase di validazione è stata svolta nel circuito interno dell'Ateneo, utilizzando la mappa statica costruita durante la fase di mappatura. L'ambiente, costituito da corridoi rettilinei e intersezioni a geometria simmetrica, rappresenta un caso di studio critico per i sistemi di localizzazione probabilistica, poiché la scarsità di landmark distintivi riduce la capacità del filtro di discriminare ipotesi alternative di posa. In tale configurazione, il sistema è stato impiegato per verificare la stabilità numerica del filtro di particelle e la coerenza spaziale della stima in condizioni controllate.

Come mostrato in Figura 8.3, la traiettoria stimata dall'AMCL risulta ampiamente sovrapposta alla traiettoria di riferimento (ground truth). Le deviazioni medie lungo

i tratti rettilinei sono inferiori a 10 cm, mentre lievi scostamenti sono osservabili in corrispondenza delle curve e delle intersezioni, dove l'informazione geometrica è meno discriminante.

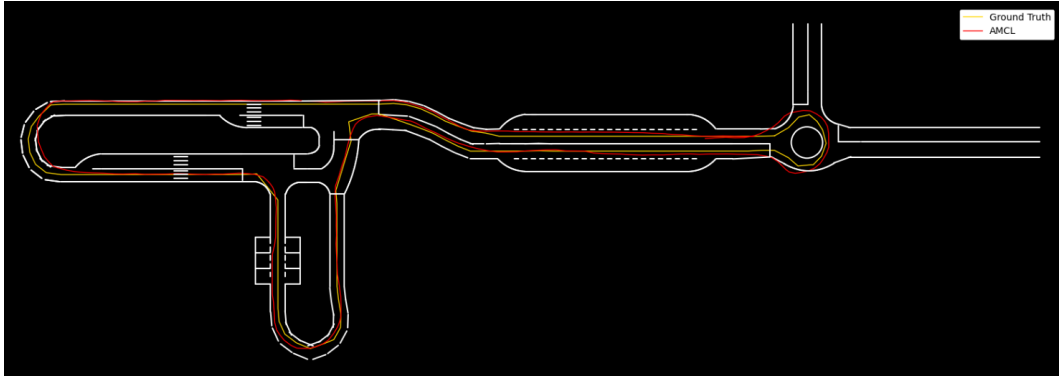


Figura 8.3.: Confronto tra traiettoria stimata da AMCL e traiettoria di riferimento.

L'andamento temporale dell'errore di localizzazione, riportato in Figura 8.4, evidenzia una variabilità compresa tra 0 e 0.30 m, con valore medio pari a 0.09 m.

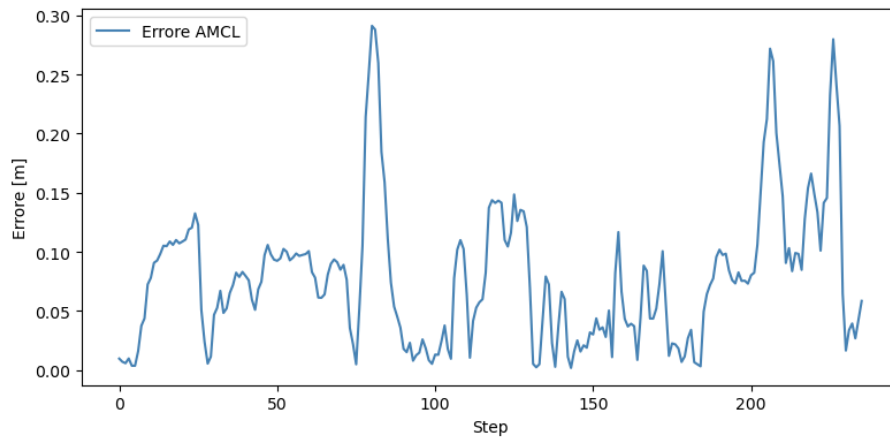


Figura 8.4.: Andamento temporale dell'errore di localizzazione dell'AMCL.

I picchi superiori a 0.25 m si verificano in prossimità di aree prive di discontinuità geometriche significative, dove la simmetria dell'ambiente riduce la capacità di convergenza del filtro.

La funzione di distribuzione cumulativa dell'errore (Figura 8.5) mostra che il 90% delle pose stimate presenta uno scostamento inferiore a 0.12 m rispetto alla ground truth. Questo risultato conferma la stabilità numerica del filtro e la coerenza spaziale della localizzazione, anche in presenza di configurazioni ambientali con basso contenuto informativo.

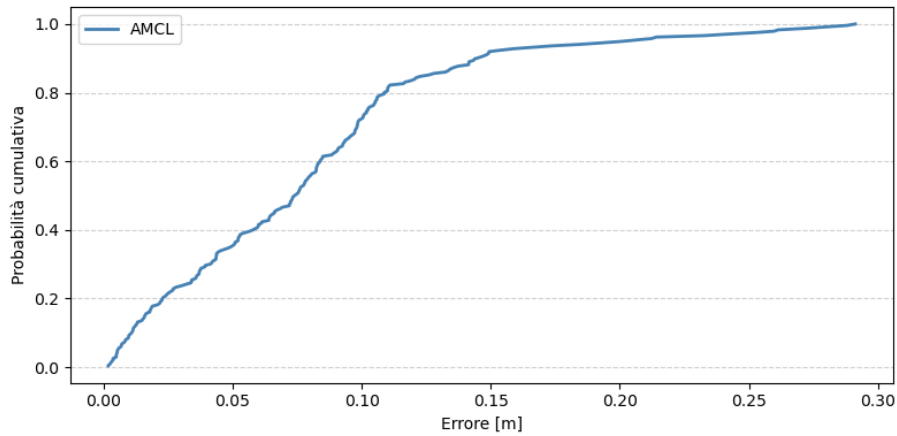


Figura 8.5.: Funzione di distribuzione cumulativa dell'errore di localizzazione dell'AMCL.

Bosch Engineering Center di Cluj-Napoca

L'algoritmo, già validato nel circuito universitario, è stato successivamente impiegato presso il Bosch Engineering Center, sede della competizione internazionale. L'area di prova, presentava maggiore eterogeneità geometrica, con percorsi multipli, rotatorie, zone di sosta e ostacoli disposti in modo irregolare. La presenza di superfici vetrate e di elementi mobili, quali altri veicoli e operatori, ha consentito di verificare il comportamento dell'AMCL in condizioni operative reali, caratterizzate da disturbi sensoriali e variabilità dinamica.

Il nodo di localizzazione è stato configurato con gli stessi parametri di base della fase universitaria, introducendo tuttavia alcune modifiche per adattare la risposta del filtro all'ambiente di Cluj. In particolare, il parametro `minimum_travel_distance` è stato incrementato a 0.5 m, al fine di ridurre la frequenza di aggiornamento e contenere il carico computazionale, mentre l'opzione `do_beamskip` è stata abilitata per filtrare automaticamente le misure incoerenti dovute a riflessioni o ostacoli temporanei. Questa configurazione ha consentito di preservare la consistenza della stima anche in presenza di disturbi locali e superfici parzialmente riflettenti.

La Figura 8.6 mostra la sovrapposizione tra la traiettoria stimata e la ground truth. Il tracciato evidenzia una buona corrispondenza spaziale, con deviazioni più marcate nelle aree perimetrali ad alta riflettività ottica, dove il segnale LiDAR risulta parzialmente degradato.

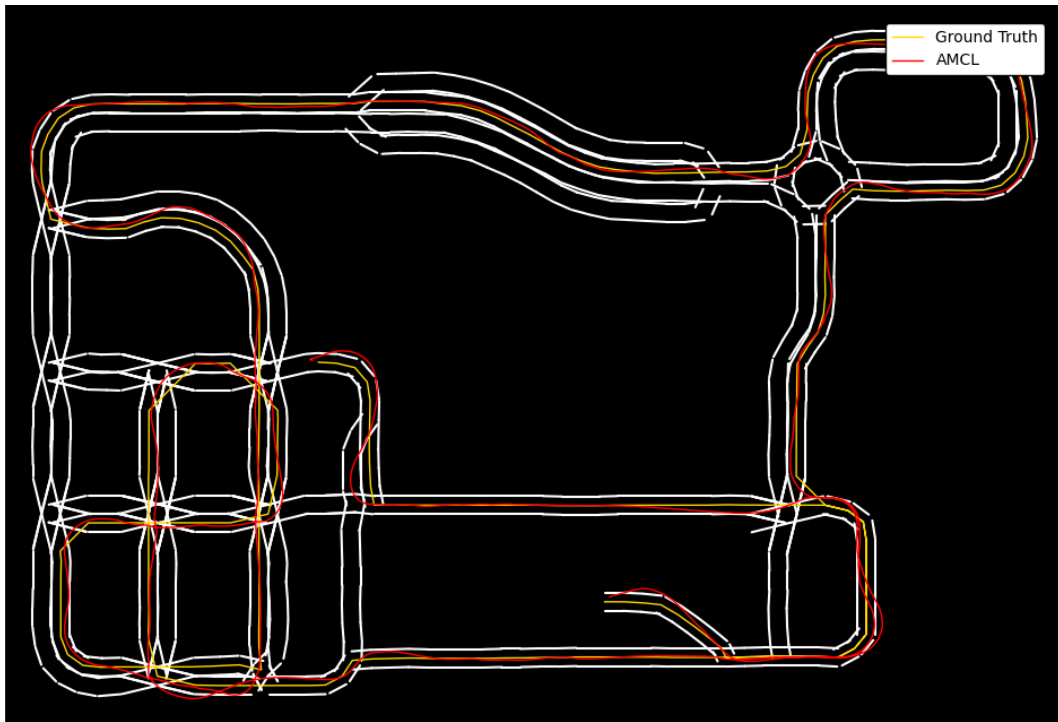


Figura 8.6.: Confronto tra traiettoria stimata da AMCL e traiettoria di riferimento.

L'errore di localizzazione, illustrato in Figura 8.7, risulta compreso tra 0 e 0.40 m, con valore medio pari a 0.11 m e deviazione standard di 0.06 m. I picchi oltre 0.30 m si concentrano nelle zone prossime alle pareti vetrate, dove le riflessioni parziali del fascio laser hanno generato misure spurie. L'attivazione del beam skipping ha evitato la propagazione di tali errori, preservando la stabilità della distribuzione di particelle e la coerenza temporale del filtro.

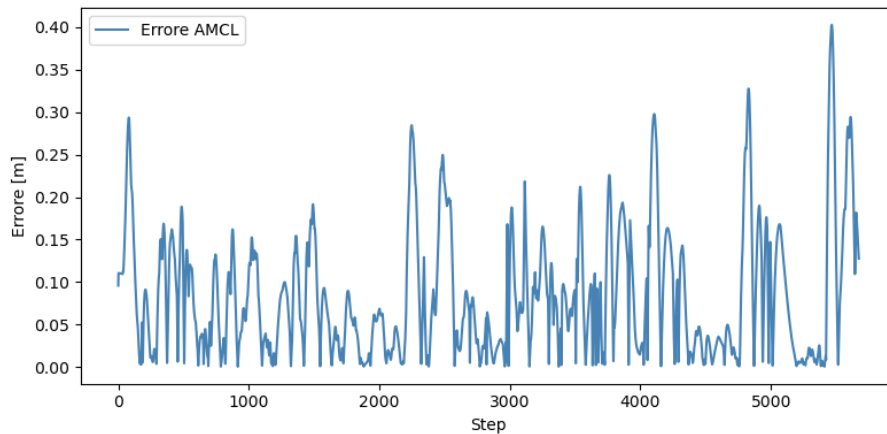


Figura 8.7.: Errore di localizzazione stimato dall'AMCL.

La funzione di distribuzione cumulativa dell'errore, riportata in Figura 8.8, indica che il 90% delle pose presenta uno scostamento inferiore a 0.18 m dalla traiettoria di riferimento. Tale valore, leggermente superiore rispetto a quello registrato nel circuito

dell'Ateneo, riflette la maggiore complessità geometrica e dinamica dell'ambiente, ma conferma la robustezza e l'affidabilità del filtro di particelle in condizioni operative realistiche.

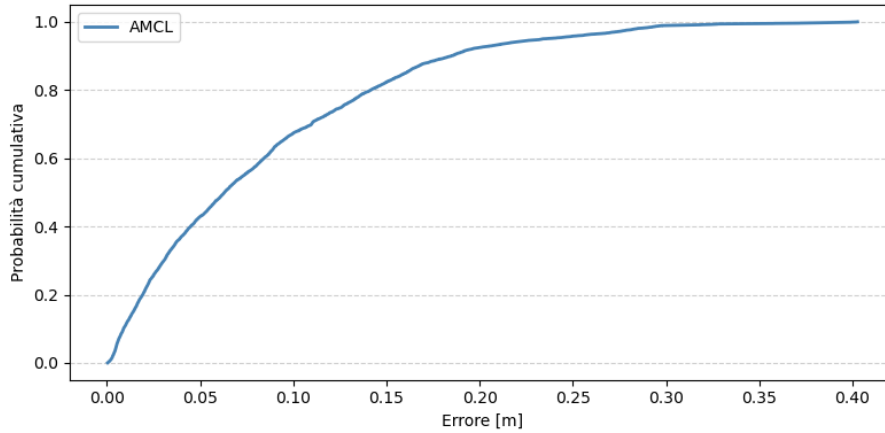


Figura 8.8.: Funzione di distribuzione cumulativa dell'errore di localizzazione dell'AMCL.

8.2.3. Comportamento in condizioni di guasto sensoriale

La robustezza dell'architettura deriva dalla ridondanza tra due sorgenti di correzione indipendenti, che intervengono in punti diversi della catena: l'UWB corregge la deriva all'interno della fusione odometrica, mentre il LiDAR, tramite l'AMCL, ancora la posa alla mappa. Poiché le due sorgenti vengono meno in circostanze differenti, il sistema può tollerare la perdita di una di esse senza interrompere la localizzazione. In caso di perdita completa del segnale UWB, ad esempio in un sottopasso o in una zona priva di copertura, la fase di aggiornamento assoluto si arresta e la fusione prosegue in sola predizione, integrando encoder e IMU. La stima resta accurata per brevi intervalli, ma l'errore cresce con la deriva odometrica: alle velocità operative del veicolo, attorno a 0,2 m/s, e con i livelli di deriva misurati, un'interruzione di pochi secondi introduce un errore aggiuntivo dell'ordine del decimetro, mentre assenze prolungate riportano l'accuratezza ai valori della sola odometria, con errori superiori al metro su percorsi estesi. Va però osservato che, finché la mappa offre riferimenti geometrici sufficienti, la correzione basata su LiDAR continua a operare attraverso l'AMCL e compensa in larga parte l'assenza dell'UWB.

In modo speculare, un'occlusione temporanea del LiDAR, tipicamente dovuta al passaggio di un pedone o di un altro veicolo, non compromette la localizzazione. L'AMCL scarta i raggi incoerenti tramite il beam skipping e, dato l'elevato numero di raggi per scansione, poche letture occluse non alterano in modo sensibile la distribuzione dei pesi. Durante un'occlusione totale ma breve il filtro prosegue sul solo modello di moto odometrico, mantenendo il tracking, per poi riconvergere non appena le scansioni tornano disponibili. Solo la perdita simultanea e prolungata di

UWB e LiDAR riporterebbe il sistema alla pura navigazione inerziale, condizione non riscontrata nelle prove.

8.2.4. Risultati complessivi e considerazioni finali

Al fine di valutare in modo sintetico le prestazioni del sistema di localizzazione, sono stati confrontati i risultati ottenuti con i diversi approcci analizzati nei capitoli precedenti: odometria grezza, filtro di Kalman esteso (EKF), filtro di Kalman unscented (UKF) e Adaptive Monte Carlo Localization (AMCL). Le metriche di accuratezza, espresse come errore quadratico medio (RMSE), errore medio assoluto (MAE) ed errore massimo di localizzazione, sono state calcolate rispetto alla traiettoria di riferimento (ground truth). Le tabelle seguenti riportano i valori ottenuti nei due ambienti sperimentali, riprendendo per completezza i risultati precedentemente discussi per EKF, UKF e odometria grezza, al fine di consentire un confronto diretto con l'AMCL.

Tabella 8.3.: Metriche di accuratezza della localizzazione nel circuito dell'Ateneo.

Metodo	RMSE [m]	MAE [m]	Errore max [m]
Odometria grezza	0.5039	0.4144	1.0939
EKF	0.1535	0.1143	0.4829
UKF	0.2039	0.1542	0.6061
AMCL	0.0979	0.0781	0.2912

Nel circuito universitario, l'AMCL ha fornito le migliori prestazioni complessive, con riduzione del RMSE del 35% rispetto all'EKF e superiore al 50% rispetto all'UKF. L'approccio probabilistico ha permesso di limitare la deriva odometrica e mantenere la consistenza della stima anche in presenza di geometrie simmetriche e basso contenuto informativo delle misure LiDAR. La rappresentazione della posa come distribuzione di particelle ha garantito una migliore gestione dell'incertezza rispetto ai filtri di Kalman, la cui linearizzazione introduce errori di approssimazione nei contesti altamente non lineari.

Tabella 8.4.: Metriche di accuratezza della localizzazione presso il Bosch Engineering Center di Cluj-Napoca.

Metodo	RMSE [m]	MAE [m]	Errore max [m]
Odometria grezza	0.4192	0.3242	1.1187
EKF	0.1430	0.1271	0.4687
UKF	0.2557	0.1753	1.3514
AMCL	0.1116	0.0836	0.29

Nel tracciato del Bosch Engineering Center di Cluj-Napoca, più esteso e dinamico, l'AMCL ha confermato le prestazioni osservate nel circuito dell'Ateneo, mantenendo un errore medio assoluto di 0.083 m e un errore massimo inferiore a 0.3 m.

L'analisi comparativa conferma la superiorità dell'approccio di localizzazione basato su filtro di particelle rispetto ai metodi di stima lineare.

Dal punto di vista architetturale, l'integrazione tra la fusione sensoriale odometrica e il modulo AMCL all'interno del framework ROS 2 Navigation2 ha dimostrato un'elevata compatibilità e un bilanciamento efficace tra accuratezza e carico computazionale. L'intera pipeline risulta implementabile su piattaforme embedded a risorse limitate, mantenendo prestazioni adeguate per applicazioni di navigazione autonoma in tempo reale.

Complessivamente, le evidenze sperimentali validano l'architettura proposta: la combinazione di tecniche di fusione sensoriale e di localizzazione probabilistica fornisce un sistema affidabile e scalabile, capace di operare in scenari caratterizzati da incertezza percettiva e variabilità ambientale. L'approccio implementato costituisce una base solida per ulteriori estensioni verso moduli di navigazione predittiva e pianificazione adattiva in contesti multi-sensore.

Parte V.

Conclusione

Capitolo 9.

Conclusione

Il lavoro ha affrontato la progettazione e lo sviluppo di un'architettura integrata per la mappatura e la localizzazione di un veicolo autonomo in scala 1:10, con l'obiettivo di ottenere una stima accurata della posa e una rappresentazione coerente dell'ambiente operativo. L'intero sistema è stato sviluppato all'interno della piattaforma ROS 2 e validato nel contesto della competizione Bosch Future Mobility Challenge, che ha fornito specifici vincoli progettuali e obiettivi funzionali.

L'approccio adottato ha previsto la definizione di una pipeline modulare costituita da un livello odometrico, un livello di localizzazione globale e un modulo di mappatura probabilistica. Nel primo, le misure provenienti da encoder, unità inerziale e UWB sono state fuse mediante un filtro bayesiano (EKF/UKF) calibrato sperimentalmente; nel secondo, la localizzazione è stata riferita a una mappa di occupazione ottenuta attraverso il pacchetto SLAM Toolbox, mentre il terzo modulo, basato su AMCL, ha consentito una stima probabilistica della posa adattiva rispetto all'ambiente. La separazione dei livelli ha reso possibile la parametrizzazione indipendente dei moduli e una più agevole attività di messa a punto.

Le prove sperimentali hanno confermato la correttezza del modello e la stabilità della pipeline di localizzazione. La fusione sensoriale ha ridotto significativamente la deriva odometrica, mentre la localizzazione globale basata su SLAM Toolbox e AMCL ha mantenuto un errore medio di posizione entro 10 cm in scenari differenti, garantendo continuità di stima anche in presenza di parziali occlusioni del sensore LiDAR. Il sistema ha dimostrato robustezza operativa e un carico computazionale compatibile con le risorse di bordo, confermando la validità dell'approccio integrato. Il principale contributo dell'elaborato consiste nella realizzazione di una pipeline coerente e scalabile per la localizzazione e la mappatura di veicoli autonomi, basata su componenti open-source e caratterizzata da un'elevata modularità. Il lavoro evidenzia inoltre la possibilità di applicare metodologie di fusione sensoriale e di localizzazione probabilistica anche in piattaforme di ridotte dimensioni, mantenendo un equilibrio tra accuratezza e complessità computazionale.

Alcune semplificazioni risultano tuttavia necessarie. Il modello cinematico Ackermann è stato approssimato con un modello differenziale per compatibilità con il nodo AMCL, mentre la mappa è stata generata unicamente a partire da dati LiDAR planari, limitando la capacità di riconoscimento in ambienti tridimensionali. Inoltre,

Capitolo 9. Conclusione

la calibrazione delle covarianze sensoriali è stata eseguita in ambiente controllato e potrebbe richiedere una ritaratura in scenari con differenti condizioni di rumore.

In prospettiva, si prevede di estendere l'architettura mediante l'integrazione di moduli di visual SLAM per la fusione di dati visivi e LiDAR, di adottare modelli cinematici più aderenti alla piattaforma reale e di introdurre procedure di calibrazione automatica delle covarianze per ridurre l'intervento manuale. Tali sviluppi potranno migliorare ulteriormente la robustezza del sistema e ampliarne la trasferibilità a contesti applicativi più complessi.

Nel complesso, i risultati ottenuti dimostrano il raggiungimento degli obiettivi definiti in apertura e forniscono una base sperimentalmente validata per l'evoluzione di sistemi di localizzazione e mappatura destinati a veicoli autonomi di piccola scala.

Bibliografia

- [1] Ismail Güvenç and Chia-Chin Chong. A survey on TOA based wireless localization and NLOS mitigation techniques. *IEEE Communications Surveys & Tutorials*, 11(3):107–124, 2009.
- [2] Stefano Marano, Wesley M. Gifford, Henk Wymeersch, and Moe Z. Win. NLOS identification and mitigation for localization based on UWB experimental data. *IEEE Journal on Selected Areas in Communications*, 28(7):1026–1035, 2010.
- [3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.
- [4] Tian Yi Lim, Edoardo Ghignone, Nicolas Baumann, and Michele Magno. Robustness evaluation of localization techniques for autonomous racing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024. arXiv:2401.07658.
- [5] Steve Macenski, Francisco Martín, Ruffin White, and Jonatan Ginés Clavero. Ros 2: Design, architecture, and use in the robot operating system. *IEEE Robotics and Automation Magazine*, 29(4):17–29, 2022.
- [6] Yan Wan, Haibo Zhang, and Chunyan Wang. Uwb-based indoor localization with imu sensor fusion using extended kalman filter. *IEEE Access*, 8:194646–194658, 2020.
- [7] Edwin R. Kraft. A quaternion-based unscented kalman filter for orientation tracking. In *Proceedings of the American Control Conference (ACC)*, pages 413–419, 2003.
- [8] Dietrich Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 343–349, 1999.
- [9] Michael Montemerlo and Sebastian Thrun. The graphslam algorithm with applications to large-scale mapping of urban structures. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 406–413, 2006.
- [10] Rainer Kümmerle, Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Robotics & Automation Magazine*, 21(4):23–39, 2014.

Bibliografia

- [11] L. D’Alfonso, F. Nardi, and P. Rizzo. Mobile robot localization via ekf and ukf: A comparison based on real data. *Robotics and Autonomous Systems*, 70:37–48, 2015.
- [12] S. Balasubramanian, A. Rajput, W. R. Hascaryo, and C. Rastogi. Comparison of dynamic and kinematic model driven extended kalman filters (ekf) for the localization of autonomous underwater vehicles. *arXiv preprint arXiv:2105.12309*, 2021.
- [13] Steve Macenski, Tom Moore, David V. Lu, Alexey Merzlyakov, and Michael Ferguson. From the desks of ros maintainers: A survey of modern and capable mobile robotics algorithms in ros 2. *arXiv preprint arXiv:2307.15236*, 2023. Analisi aggiornata di ROS 2, inclusi algoritmi SLAM, localizzazione e navigazione.
- [14] Yue Sheng, Hao Zhang, and Lei Liu. Integration of ros 2, lidar slam and amcl for autonomous navigation of scaled vehicles. In *IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1062–1067, 2023.

Sitografia

- [15] Reely. 1:10 electric car model "deathwatcher evo" 4wd arr – tc-04 onroad chassis. <https://asset.conrad.com/media10/add/160267/c1/-/gl/001406735ML02/manual-1406735-reely-tc-04-onroad-chassis-110-rc-model-car-electricroad-version-4wdarr.pdf>, 2025. Datasheet tecnico. Ultimo accesso: 6 novembre 2025.
- [16] Reely. Motore dc 36gp-555 – datasheet tecnico. <https://asset.conrad.com/media10/add/160267/c1/-/en/001365936DS01/datasheet-1365936-reely-dc-motor-36gp-555.pdf>, 2025. Datasheet tecnico. Ultimo accesso: 6 novembre 2025.
- [17] Reely. Standard servo rs-610wp-mg – servo analogico con ingranaggi in metallo. <https://asset.conrad.com/media10/add/160267/c1/-/en/001365926DS01/datasheet-1365926-reely-standard-servo-rs-610wp-mg-analogue-servo-gear-box-material-metal-connector-system-jr.pdf>, 2025. Datasheet tecnico. Ultimo accesso: 6 novembre 2025.
- [18] Bosch Sensortec. Bno055 intelligent 9-axis absolute orientation sensor. <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf>, 2025. Datasheet tecnico. Ultimo accesso: 6 novembre 2025.
- [19] STMicroelectronics. Nucleo-g491re development board – datasheet. <https://www.alldatasheet.com/datasheet-pdf/pdf/1644759/STMICROELECTRONICS/NUCLEO-G491RE.html>, 2025. Datasheet tecnico. Ultimo accesso: 6 novembre 2025.
- [20] NVIDIA Corporation. Jetson orin nano 8gb module – datasheet. https://www.mouser.com/pdfDocs/Jetson_Orin_Nano_Series_DS-11105-001_v11.pdf, 2023. Datasheet tecnico. Ultimo accesso: 6 novembre 2025.
- [21] Raspberry Pi Ltd. Camera module 3 – product brief / datasheet. <https://datasheets.raspberrypi.com/camera/camera-module-3-product-brief.pdf>, 2024. Datasheet tecnico. Ultimo accesso: 6 novembre 2025.
- [22] LDROBOT. Dtof lidar ld19 – datasheet. https://www.ldrobot.com/images/2023/05/23/LDROBOT_LD19_Datasheet_EN_v2.6_Q1JXIRVq.pdf, 2023. Datasheet tecnico. Ultimo accesso: 6 novembre 2025.

- [23] Qorvo / Decawave. Mdek1001 kit – user manual. <https://www.qorvo.com/products/d/da007995>, 2025. Manuale utente. Ultimo accesso: 6 novembre 2025.
- [24] Open Robotics. Ros 2 humble hawkbill – official documentation. <https://docs.ros.org/en/humble/>, 2025. Distribuzione ROS 2 Humble Hawksbill. Ultimo accesso: 6 novembre 2025.
- [25] Steve Macenski. slam_toolbox for ros 2. https://github.com/SteveMacenski/slam_toolbox, 2025. Repository e documentazione tecnica. Ultimo accesso: 6 novembre 2025.
- [26] ROS Navigation2. Amcl – adaptive monte carlo localization (ros 2 nav2). <https://navigation.ros.org/configuration/packages/configuring-amcl.html>, 2025. Documentazione ufficiale. Ultimo accesso: 6 novembre 2025.
- [27] Open Robotics. Ros 2 documentation. <https://docs.ros.org/en/>, 2025. Ultimo accesso: 6 novembre 2025.