

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica e dell'Automazione



TESI DI LAUREA

**Esperienze di utilizzo del software Gazebo nella modellazione di un
case study legato all'ingegneria dell'automazione**

**Experiences of using the software Gazebo in modeling a case study
related to automation engineering**

Relatore

Prof. Domenico Ursino

Candidato

Federico Beni

ANNO ACCADEMICO 2022-2023

*Tutto ciò che fai
ha il suo valore.*

Sommario

Questa tesi esplora l'integrazione e l'applicazione dell'Intelligenza Artificiale (AI) e dell'automazione nell'ingegneria moderna, con un focus particolare sull'uso del software Gazebo, per la modellazione e simulazione di un robot automatizzato. In essa, vengono analizzate le evoluzioni e le applicazioni dell'AI nell'automazione, evidenziando i benefici e le sfide poste da questa tecnologia rivoluzionaria. Il lavoro si concentra sullo studio di Gazebo, esaminandone le caratteristiche tecniche, le capacità di simulazione e l'integrazione con i sensori. Attraverso esperienze pratiche con i sensori IMU, di contatto e LIDAR, vengono testate le funzionalità di Gazebo e le potenzialità dell'automazione avanzata. La tesi fornisce un'analisi dettagliata delle configurazioni dei sensori, degli esperimenti condotti e dei risultati ottenuti, fornendo un contributo significativo alla comprensione del ruolo dell'AI nell'evoluzione della robotica e dell'automazione. Infine, vengono discusse le implicazioni future di queste tecnologie, ponendo le basi per ulteriori ricerche e sviluppi nel campo dell'ingegneria dell'automazione.

Keyword: Intelligenza Artificiale, AI, Automazione, Robotica, Gazebo, Simulazione, Sensore IMU, Sensore di contatto, Sensore LIDAR.

| | |
|--|-----------|
| Introduzione | 1 |
| 1 Utilizzo dell'AI nell'automazione | 4 |
| 1.1 Introduzione all'automazione e all'Intelligenza Artificiale | 4 |
| 1.1.1 Panoramica sull'automazione nell'ingegneria | 4 |
| 1.1.2 L'emergere dell'Intelligenza Artificiale | 5 |
| 1.1.3 Benefici dell'AI nell'automazione | 5 |
| 1.2 Il ruolo dell'AI nell'automazione moderna | 5 |
| 1.2.1 Evoluzione dell'automazione: dalla programmazione rigida all'intelligenza adattiva | 6 |
| 1.2.2 Applicazioni concrete dell'AI nell'automazione | 6 |
| 1.2.3 Miglioramenti portati dall'AI nell'efficienza e nella sicurezza | 7 |
| 1.3 Tecnologie e metodi di implementazione dell'AI nell'automazione | 8 |
| 1.3.1 Tecnologie AI specifiche per l'automazione | 8 |
| 1.3.2 Integrazione efficiente dell'AI nei sistemi di automazione | 9 |
| 1.4 Successi e Prospettive dell'AI nell'Automazione | 9 |
| 1.4.1 Progressi recenti e innovazioni emergenti | 10 |
| 1.4.2 Prospettive future dell'AI nell'automazione | 10 |
| 2 Gazebo | 12 |
| 2.1 Introduzione a Gazebo | 12 |
| 2.1.1 Origine e sviluppo di Gazebo | 12 |
| 2.1.2 Caratteristiche uniche di Gazebo | 13 |
| 2.1.3 Integrazione con l'Intelligenza Artificiale | 14 |
| 2.2 Caratteristiche tecniche di Gazebo | 15 |
| 2.2.1 SDFFormat | 15 |
| 2.2.2 Architettura e design del software | 15 |
| 2.2.3 Motori di fisica e simulazione in Gazebo | 17 |
| 2.2.4 Cinematica e modellazione dei robot | 17 |
| 2.2.5 Networking e integrazione con altri strumenti in Gazebo | 20 |
| 2.3 Utilizzo di Gazebo nell'ingegneria dell'automazione | 21 |
| 2.3.1 Sfide e limitazioni nell'uso di Gazebo | 23 |
| 2.3.2 Innovazioni ed aggiornamenti recenti in Gazebo | 23 |

| | | |
|----------|--|-----------|
| 3 | Esperienza relativa al sensore IMU | 25 |
| 3.1 | Introduzione al sensore IMU | 25 |
| 3.1.1 | Caratteristiche tecniche e componenti del sensore IMU | 25 |
| 3.1.2 | Importanza del sensore IMU nell'automazione e nella robotica | 27 |
| 3.2 | Configurazione del sensore IMU in Gazebo | 27 |
| 3.3 | Esperimenti e risultati | 30 |
| 3.4 | Conclusioni | 30 |
| 4 | Esperienza relativa al sensore di contatto | 32 |
| 4.1 | Introduzione al sensore di contatto | 32 |
| 4.1.1 | Importanza del sensore di contatto nell'automazione e nella robotica | 32 |
| 4.1.2 | Impiego del sensore di contatto in Gazebo | 33 |
| 4.2 | Creazione dell'ostacolo per la simulazione | 34 |
| 4.3 | Configurazione del sensore di contatto | 35 |
| 4.4 | Implementazione del TriggeredPublisher plugin | 37 |
| 4.4.1 | Importanza del TriggeredPublisher plugin in contesti reali | 38 |
| 4.5 | Esperimenti e risultati | 39 |
| 4.6 | Conclusioni | 40 |
| 5 | Esperienza relativa al sensore LIDAR | 41 |
| 5.1 | Introduzione al sensore LIDAR | 41 |
| 5.1.1 | Importanza del sensore LIDAR nell'automazione e nella robotica | 43 |
| 5.2 | Configurazione del sensore LIDAR in Gazebo | 43 |
| 5.3 | Analisi della configurazione del sensore LIDAR | 46 |
| 5.3.1 | Parametri principali del sensore LIDAR | 46 |
| 5.3.2 | Importanza dei parametri nel contesto specifico | 46 |
| 5.4 | Implementazione del codice per l'elaborazione dei dati del LIDAR | 47 |
| 5.5 | Ottimizzazione all'avvio di Gazebo | 50 |
| 5.6 | Esperimenti e risultati | 51 |
| 5.7 | Conclusioni | 51 |
| | Conclusioni | 53 |
| | Bibliografia | 55 |
| | Ringraziamenti | 57 |

Elenco delle figure

| | | |
|-----|---|----|
| 2.1 | Linea temporale della storia e dello sviluppo di Gazebo | 13 |
| 2.2 | Esempio di un ambiente industriale, con annessa presenza di un braccio robotico, sviluppato con Gazebo | 14 |
| 2.3 | Interfaccia grafica di Gazebo all'avvio del mondo <code>shapes.sdf</code> | 16 |
| 2.4 | La figura illustra un giunto meccanico dove il frame J_C è solidale al collegamento figlio C , e il frame J_P è solidale al collegamento genitore P , con la posa X_{CJ_C} che definisce la configurazione spaziale di J_C rispetto a C | 20 |
| 2.5 | Modello multibody 3D di un robot mobile su ruote nell'ambiente Gazebo | 22 |
| 2.6 | Il mondo VRX Sand Island renderizzato in Gazebo con un grafico della velocità del vento nel tempo | 22 |
| 3.1 | Rappresentazione degli assi e degli angoli pitch, roll e yaw di cui si serve l'IMU per eseguire le misurazioni | 26 |
| 3.2 | Robot utilizzato nella simulazione | 28 |
| 3.3 | Visualizzazione dei dati relativi ad orientamento, accelerazione lineare e velocità angolare del robot in movimento riportati sul topic <code>/imu</code> | 29 |
| 4.1 | Robot industriale dotato di sensore di contatto impiegato in una catena di montaggio | 33 |
| 4.2 | Rappresentazione del robot e dell'ostacolo | 35 |
| 4.3 | Messaggio <code>data: true</code> sul topic <code>/wall/touched</code> quando il robot tocca il muro | 37 |
| 4.4 | Visualizzazione dei messaggi dal topic <code>/cmd_vel</code> | 38 |
| 5.1 | Raffigurazione grafica della scansione meccanica LIDAR | 42 |
| 5.2 | Raffigurazione grafica della tecnologia Flash LIDAR | 42 |
| 5.3 | Parte dei dati di range visualizzati sul topic <code>/lidar</code> | 45 |

La rivoluzione tecnologica che ha caratterizzato l'ultimo secolo ha portato alla nascita e allo sviluppo di concetti innovativi, come l'Intelligenza Artificiale (AI) e l'automazione, due forze motrici che stanno ridefinendo in modo radicale il nostro modo di vivere, lavorare e interagire con il mondo. Questi progressi tecnologici hanno aperto la strada a una trasformazione senza precedenti in molti aspetti della vita quotidiana, rendendo l'AI e l'automazione non solo elementi centrali di ricerca accademica, ma anche strumenti fondamentali nel progresso di settori diversificati. Settori come la manifattura, l'assistenza sanitaria, la logistica e l'agricoltura stanno sperimentando cambiamenti rivoluzionari grazie all'introduzione di queste tecnologie, che offrono nuove possibilità per l'efficienza, l'accuratezza e la personalizzazione. Nella manifattura, ad esempio, l'integrazione dell'IA ha portato alla creazione di impianti produttivi intelligenti, dove le macchine comunicano tra loro e si adattano in modo dinamico per ottimizzare i processi produttivi. Nel settore sanitario, l'AI sta trasformando la diagnostica e il trattamento dei pazienti, fornendo strumenti di analisi dei dati che possono prevedere malattie e personalizzare le cure. Nel campo della logistica, l'automazione e l'IA stanno rivoluzionando la gestione della supply chain, migliorando l'efficienza del trasporto merci e riducendo gli sprechi e gli errori. Questi esempi evidenziano come l'IA e l'automazione siano diventati pilastri fondamentali per il progresso e l'innovazione in una vasta gamma di settori industriali.

L'era dell'Industria 4.0, contraddistinta dall'integrazione di tecnologie digitali avanzate nei processi produttivi, ha segnato l'inizio di un'epoca in cui l'AI e l'automazione non sono solo strumenti per l'efficienza, ma fondamentali per una trasformazione radicale dei sistemi produttivi. Tale transizione sta preparando il terreno per l'avvento dell'Industria 5.0, che prevede un ulteriore salto qualitativo nell'ambito della produzione. L'Industria 5.0, infatti, enfatizza l'importanza di una produzione più personalizzata, sostenibile ed efficiente, ponendo al centro l'interazione tra uomo e macchina. Questa nuova era si distingue per il suo approccio olistico, dove l'AI e l'automazione non sono viste come sostituti dell'intelligenza umana, ma come complementi che potenziano le capacità umane, portando ad un livello di collaborazione mai visto prima.

Esempi contemporanei di questa sinergia uomo-macchina includono l'adozione di robot collaborativi, noti come cobot, che lavorano a fianco degli operai per compiti complessi, migliorando la sicurezza e l'efficienza. Allo stesso modo, l'IA sta giocando un ruolo cruciale nell'ottimizzazione delle catene di fornitura, dove algoritmi avanzati analizzano e gestiscono enormi quantità di dati per ottimizzare la logistica, ridurre i tempi di inattività e anticipare le esigenze del mercato. Questa evoluzione dell'IA e dell'automazione rappresenta non solo una rivoluzione tecnologica, ma anche un cambio paradigmatico nel modo in cui concepiamo

il lavoro, la produzione e l'interazione tra intelligenza umana e Intelligenza Artificiale.

In questo contesto di evoluzione tecnologica e di stretta collaborazione tra uomo e macchina, l'esperienza e il lavoro descritti in questa tesi mirano a esplorare le frontiere dell'automazione e dell'Intelligenza Artificiale attraverso l'uso innovativo del software Gazebo. Specializzandoci nella modellazione e simulazione di un robot automatizzato, abbiamo avuto l'opportunità di coniugare la teoria con la pratica in un ambito estremamente dinamico e in rapida evoluzione. Il robot, una struttura dotata di tre ruote, è stato realizzato per ospitare e sperimentare con tre tipi di sensori cruciali per l'automazione: sensore IMU, sensore di contatto e sensore LIDAR. Questa configurazione specifica ha aperto le porte ad un'analisi dettagliata delle capacità e delle potenzialità dell'automazione e della robotica avanzata, consentendoci di testare in modo efficace l'integrazione dell'Intelligenza Artificiale in un ambiente simulato ma estremamente realistico. L'impiego del sensore IMU è stato fondamentale per fornire dati precisi sulla posizione e l'orientamento del robot, elementi chiave per garantire una navigazione accurata e un controllo ottimale del movimento. Ciò ha portato ad una maggiore comprensione di come i movimenti e le reazioni del robot possano essere affinati per adattarsi a diversi ambienti ed a diversi compiti. Il sensore di contatto, integrato per percepire e reagire a stimoli fisici, ha permesso di esplorare nuove dimensioni dell'interazione robot-ambiente, mostrando come un robot possa rispondere in modo intelligente e adattivo a vari scenari, un aspetto fondamentale per applicazioni pratiche come la robotica collaborativa e l'automazione industriale. Il sensore LIDAR, infine, ha svolto un ruolo essenziale nella mappatura spaziale e nella percezione dell'ambiente circostante, consentendo al robot di navigare e operare in modo autonomo e sicuro, anche in ambienti complessi.

Questa combinazione di sensori non ha solo evidenziato la versatilità del robot in diversi contesti operativi, ma ha anche messo in luce l'importanza vitale di una configurazione precisa e di una calibrazione accurata per garantire operazioni automatizzate di successo. Oltre alla progettazione fisica e alla configurazione del robot, il nostro lavoro ha incluso l'analisi dei dati raccolti e dei risultati ottenuti dagli esperimenti svolti, sfruttando le capacità avanzate di Gazebo come piattaforma di simulazione. Tale approccio ha permesso di andare oltre i limiti dei test tradizionali in ambienti reali, minimizzando i rischi e ampliando le possibilità di sperimentazione in condizioni controllate e sicure.

Il valore di questa ricerca risiede nella sua capacità di fornire insight concreti sulle potenzialità e sulle sfide dell'automazione avanzata. Attraverso questo lavoro, si apre un nuovo cammino per il futuro dell'ingegneria robotica e dell'Intelligenza Artificiale, il quale indica nuove direzioni per lo sviluppo di soluzioni innovative e per la comprensione delle complesse dinamiche che intercorrono tra robot, ambienti e umani. Il nostro percorso, intrapreso con questa tesi, rappresenta un passo avanti verso un futuro in cui l'integrazione tra Intelligenza Artificiale, robotica e automazione gioca un ruolo chiave nell'evoluzione tecnologica e sociale.

La presente tesi è strutturata come di seguito specificato;

- Nel Capitolo 1, *Utilizzo dell'AI nell'automazione*, viene trattata l'importanza crescente dell'AI e dell'automazione nel contesto moderno, con un'analisi della loro evoluzione, delle applicazioni pratiche e dei benefici apportati in termini di efficienza e sicurezza. È presente, inoltre, un approfondimento sulle tecnologie specifiche di AI impiegate nell'automazione e sulle prospettive future del settore.
- Nel Capitolo 2, *Gazebo*, viene effettuato uno studio approfondito del software Gazebo, dalle sue origini alle caratteristiche uniche che lo rendono adatto all'integrazione con l'IA. Vengono, inoltre, analizzate le sue capacità tecniche, ovvero l'SDF format, l'architettura del software, i motori di fisica, la cinematica dei robot, il networking e l'integrazione con altri strumenti. Si ha, inoltre, una discussione sul suo utilizzo nell'ingegneria dell'automazione, affrontando le sfide, le limitazioni e le recenti innovazioni.

- Il Capitolo 3, *Esperienza relativa al sensore IMU*, tratta la descrizione del sensore IMU, la sua importanza nell'automazione e nella robotica, oltre che l'analisi delle sue caratteristiche tecniche e dei suoi componenti. Sono presenti dettagli sulla configurazione del sensore IMU in Gazebo, sugli esperimenti condotti e sui risultati ottenuti, seguiti da una sezione di conclusioni che riflettono sull'esperienza.
- Nel Capitolo 4, *Esperienza relativa al sensore di contatto*, è presente un approfondimento sul sensore in questione, sulla sua rilevanza nell'automazione e nella robotica, e sul suo impiego in Gazebo. Il capitolo procede, poi, con una presentazione della creazione dell'ostacolo per la simulazione, con la configurazione del sensore di contatto, e con l'implementazione del TriggeredPublisher plugin, per concludersi poi con l'analisi degli esperimenti condotti, dei risultati e delle conclusioni tratte.
- Nel Capitolo 5, *Esperienza relativa al sensore LIDAR*, viene preso in esame, appunto, il sensore LIDAR, enfatizzando la sua importanza e analizzando la configurazione specifica in Gazebo, per poi procedere con una discussione sull'analisi della sua configurazione, sull'importanza dei suoi parametri, sull'implementazione del codice per l'elaborazione dei dati, e sull'ottimizzazione all'avvio di Gazebo. Il capitolo termina, poi, con la presentazione degli esperimenti, dei risultati ottenuti e delle conclusioni.
- Nel Capitolo 6, *Conclusioni*, viene fornito un riassunto generale dell'intero lavoro di ricerca, e vengono proposte delle riflessioni sull'esperienza di studio e sperimentazione con Gazebo ed i diversi sensori. Successivamente è presente una discussione sulle implicazioni di queste esperienze per il futuro dell'ingegneria dell'automazione e sull'importanza dell'innovazione continua nel campo dell'AI e della robotica.

Utilizzo dell'AI nell'automazione

L'evoluzione e l'integrazione dell'Intelligenza Artificiale (AI) nell'ambito dell'automazione rappresentano un punto di svolta nel panorama dell'ingegneria moderna. Questa esplorazione inizia con una panoramica dell'automazione nell'ingegneria, mettendo in luce la sua continua evoluzione e l'integrazione in vari settori industriali. L'attenzione si sposta, poi, sull'emergere dell'AI, evidenziando come questa tecnologia stia rivoluzionando le pratiche tradizionali di automazione, portando benefici notevoli in termini di efficienza, precisione e innovazione.

L'analisi prosegue esaminando l'AI nell'automazione moderna, dove si evidenzia la transizione verso sistemi che utilizzano l'intelligenza adattiva. Viene sottolineata l'importanza delle applicazioni concrete dell'AI e dei miglioramenti che essa ha apportato in termini di efficienza e sicurezza.

Segue un approfondimento sulle tecnologie e i metodi di implementazione dell'AI nell'automazione. Qui, si esplorano le specifiche tecnologie AI e le strategie per una loro integrazione efficiente nei sistemi di automazione, evidenziando l'importanza di un approccio sinergico e innovativo.

Il discorso si conclude con una riflessione sui successi e le prospettive future dell'AI nell'automazione. I recenti progressi e le innovazioni emergenti vengono analizzati, insieme alla proiezione delle potenziali future applicazioni e sfide. Quest'area di studio non solo mostra come l'AI stia migliorando i processi esistenti, ma anche come stia aprendo la strada a nuove possibilità innovative, delineando un futuro in cui l'AI sarà ancor più integrata e centrale nell'evoluzione dell'automazione.

1.1 Introduzione all'automazione e all'Intelligenza Artificiale

1.1.1 Panoramica sull'automazione nell'ingegneria

L'ingegneria dell'automazione è un campo, discendente dall'ingegneria dell'informazione, che si occupa di modellazione, simulazione e controllo di sistemi servendosi dell'utilizzo di tecnologie informatiche. La conseguente autonomia di cui vengono dotate le macchine e i sistemi minimizza il bisogno di intervento umano, migliorando l'efficienza, la precisione e la sicurezza sul lavoro.

L'automazione si è rapidamente diffusa in vari ambiti, diventando un pilastro fondamentale in diverse industrie. Un esempio emblematico è il settore manifatturiero, rivoluzionato dall'automazione nei metodi di produzione. Qui, robot e sistemi automatizzati non solo svolgono operazioni ripetitive, ma affrontano anche mansioni ad alto rischio, aumentando la sicurezza e riducendo lo sforzo fisico umano. Un altro esempio da considerare si trova nel campo dell'energia, dove l'automazione gioca un ruolo cruciale nella gestione efficiente delle risorse, contribuendo ad ottimizzare l'uso e la distribuzione di energia elettrica. Con l'avanzamento tecnologico, l'ambito di applicazione dell'automazione si è esteso ulteriormente. Oggi comprende sistemi complessi e connessi, come quelli impiegati nella gestione

intelligente delle abitazioni (domotica) e nei veicoli a guida autonoma, evidenziando la sua versatilità e capacità di adattamento alle esigenze emergenti del nostro tempo.

1.1.2 L'emergere dell'Intelligenza Artificiale

L'Intelligenza Artificiale, ovvero la capacità di un computer di eseguire funzioni che, se svolte da esseri umani, richiederebbero intelligenza, rappresenta una svolta nell'ingegneria moderna. L'AI imita aspetti dell'intelligenza umana, come l'apprendimento, il ragionamento, il problem-solving e la percezione. Queste capacità permettono agli ingegneri di progettare sistemi che non solo eseguono compiti pre-programmati, ma che possono adattarsi e rispondere a situazioni inaspettate tramite esperienza ottenuta dalla realizzazione di vari tentativi. Applicazioni come il riconoscimento vocale, la visione artificiale e i sistemi di raccomandazione sono esempi pratici di come l'AI stia trasformando il mondo dell'ingegneria.

Esistono principalmente due tipi di AI:

- *Intelligenza Artificiale debole* (weak/specialist IA), la quale comprende sistemi in grado di simulare alcune funzionalità cognitive dell'uomo senza, tuttavia, raggiungere le capacità intellettuali tipiche umane; si intendono programmi di problem-solving in grado di replicare ed applicare la logica umana per la risoluzione di problemi e la presa di decisioni.
- *Intelligenza Artificiale forte* (strong/general IA), la quale comprende sistemi in grado di diventare sapienti o addirittura coscienti di sé; questo potere intellettuale e cognitivo ha, infatti, portato scienziati ed esperti a ritenere che un giorno le macchine avranno un'intelligenza propria, autonoma e superiore a quella umana.

1.1.3 Benefici dell'AI nell'automazione

L'integrazione dell'AI nell'automazione rappresenta un cambiamento fondamentale nella progettazione e implementazione di sistemi automatizzati. Tale sinergia consente la creazione di sistemi in grado di apprendere e correggersi tramite tentativi, di ottimizzare i processi in tempo reale e di prendere decisioni autonome basate su dati complessi. Possiamo prendere come esempio l'industria manifatturiera, dove i robot dotati di AI possono adattarsi a vari compiti senza necessità di riconfigurazione manuale, mentre nei sistemi di trasporto intelligenti l'AI contribuisce a migliorare la sicurezza e l'efficienza attraverso la gestione avanzata del traffico.

Questa integrazione non solo migliora l'efficienza e la flessibilità dei sistemi automatizzati ma apre anche la strada a nuove possibilità, come la manutenzione predittiva e l'automazione personalizzata.

1.2 Il ruolo dell'AI nell'automazione moderna

L'avvento dell'Intelligenza Artificiale (AI) ha segnato un'epoca di trasformazione radicale nel campo dell'automazione. Tradizionalmente, l'automazione si è affidata a sistemi programmabili rigidi, capaci di eseguire compiti ripetitivi con alta precisione ma limitati nella loro capacità di adattamento e reattività a situazioni variabili. Con l'integrazione dell'AI, questo panorama è cambiato drasticamente.

L'AI introduce un nuovo livello di intelligenza e flessibilità nei sistemi automatizzati. Attraverso tecnologie come l'apprendimento automatico (Machine Learning), l'elaborazione del linguaggio naturale (Natural Language Processing - NLP) e la visione computerizzata, l'AI permette ai sistemi di apprendere dall'esperienza, di adattarsi a nuove condizioni

e di prendere decisioni autonome basate su dati complessi e variabili. Questo ha aperto nuove frontiere nell'automazione, permettendo applicazioni che vanno ben oltre la semplice esecuzione di compiti predefiniti.

1.2.1 Evoluzione dell'automazione: dalla programmazione rigida all'intelligenza adattiva

L'innesto di sistemi che incorporano Intelligenza Artificiale, rispetto all'adozione di sistemi programmabili rigidi, rappresenta la rivoluzione dell'automazione, che, a sua volta, porta ad una delle trasformazioni più significative nel panorama tecnologico contemporaneo. Inizialmente, l'automazione si basava su sequenze fisse e predefinite, progettate per eseguire attività ripetitive con precisione meticolosa. Questi sistemi, pur essendo altamente affidabili ed efficienti nello svolgimento dei compiti per cui erano programmati, mostravano limitazioni sostanziali quando si trattava di gestire variabilità o situazioni impreviste. Con l'avvento dell'AI, la natura stessa dell'automazione ha iniziato a cambiare. L'introduzione di algoritmi di apprendimento automatico e sistemi basati su dati ha permesso ai sistemi automatizzati di apprendere dall'esperienza, adattandosi in modo dinamico alle nuove situazioni. Questo cambiamento ha trasformato i sistemi automatizzati da semplici esecutori di compiti pre-programmati a entità capaci di interagire con l'ambiente in modo intelligente, prevedendo problemi, ottimizzando le operazioni e rispondendo in modo proattivo ai cambiamenti.

L'impiego dell'AI ha aperto la strada a un'era di automazione "intelligente", nella quale i sistemi non sono più vincolati dai loro programmi originari. Ne abbiamo un esempio per quanto riguarda le linee di produzione industriali, dove l'automazione assistita dall'AI può ora rintracciare e rimediare ad errori in tempo reale, sviluppando, inoltre, adattamenti a variazioni del materiale o al processo di produzione. Inoltre, l'automazione basata sull'AI può analizzare enormi quantità di dati per ottimizzare le operazioni, ridurre gli sprechi e migliorare l'efficienza complessiva.

1.2.2 Applicazioni concrete dell'AI nell'automazione

L'approdo dell'AI nel mondo dell'automazione ha portato a dei cambiamenti significativi riguardo le tecniche e le metodologie di lavoro in diversi ambiti, segnando, così, lo sviluppo e la crescita in tali settori. Più specificatamente, i settori maggiormente influenzati sono i seguenti:

- *Automazione industriale.* Grazie all'AI la robotica industriale ha subito un incremento di flessibilità e adattabilità. I robot interessati a questo cambiamento, infatti, possono eseguire compiti considerati più complessi, come l'assemblaggio di componenti variabili, la verniciatura personalizzata ed il controllo della qualità. Un aspetto cruciale nell'ambito dell'automazione industriale è la manutenzione predittiva; infatti, i sistemi dotati di AI analizzano in modo costante i dati dei sensori prevenendo guasti, riducendo tempi di inattività e migliorando l'efficienza delle operazioni.
- *Automazione dei Processi Aziendali(RPA).* L'AI consente alle aziende di automatizzare processi che richiedono un certo grado di decisione ed adattabilità; i sistemi di AI possono ora, infatti, interpretare ed estrarre informazioni da diversi tipi di documenti, prendere decisioni basate su regole complesse ed interagire con i clienti tramite chatbot intelligenti.
- *Automobili e trasporti autonomi.* L'AI è un elemento fondamentale nello sviluppo di veicoli autonomi. Tramite l'utilizzo di combinazioni di sensori, telecamere e radar per mappare l'ambiente circostante, i veicoli dotati di AI riescono a prendere decisioni di

guida in tempo reale e prevedere il comportamento di altri veicoli e pedoni. Questa tecnologia non si limita alle automobili, ma si estende a droni, camion autonomi e, persino, navi. La sfida principale è garantire una guida sicura e affidabile in tutte le condizioni, un obiettivo che l'AI sta continuando a perfezionare attraverso algoritmi sempre più sofisticati e l'analisi di enormi quantità di dati raccolti da veicoli in tutto il mondo.

- *Assistenza sanitaria.* Anche nell'ambito sanitario l'AI sta avendo un impatto significativo, dalla diagnostica automatizzata, dove l'AI analizza immagini mediche per identificare segni di malattie, alla gestione dei dati dei pazienti e alla personalizzazione dei piani di trattamento.
- *Agricoltura e gestione ambientale.* L'AI sta trasformando anche l'agricoltura, con droni e robot automatizzati utilizzati per la mappatura dei terreni, il monitoraggio delle colture e la gestione sostenibile delle risorse. Un notevole aiuto fornito, inoltre, dall'AI è quello che si ha nella previsione del meteo e nella gestione delle risorse idriche, contribuendo a pratiche agricole più efficienti e sostenibili.

1.2.3 Miglioramenti portati dall'AI nell'efficienza e nella sicurezza

L'introduzione dell'Intelligenza Artificiale nei sistemi di automazione ha portato a miglioramenti significativi sia in termini di efficienza operativa che di sicurezza. Questi progressi sono percettibili soprattutto in settori in cui l'errore umano può avere conseguenze gravi, e dove la precisione e la velocità di elaborazione dei dati sono cruciali. Più specificatamente, i miglioramenti apportati dall'AI sono i seguenti:

- *Efficienza operativa.* L'efficienza operativa è uno degli ambiti che si giova dei miglioramenti dovuti all'utilizzo dell'IA. In primo luogo, attraverso l'automazione di compiti ripetitivi e ad alta intensità di manodopera, l'AI libera risorse umane e consente, quindi, il loro impiego in attività più strategiche e creative. In secondo luogo, l'AI ottimizza i processi decisionali. Ad esempio, nei sistemi di gestione della supply chain, l'AI può prevedere le fluttuazioni della domanda e ottimizzare la logistica e l'inventario, riducendo i costi e migliorando la soddisfazione del cliente. Un altro vantaggio consentito dall'AI, in ambito manifatturiero, è un controllo della qualità più accurato e in tempo reale, identificando difetti e inefficienze che potrebbero sfuggire al controllo umano.
- *Sicurezza.* L'AI contribuisce notevolmente alla sicurezza nei diversi ambiti. Un esempio cruciale riguarda i sistemi di sorveglianza, dove l'AI migliora la rilevazione di attività sospette o pericolose, analizzando in tempo reale grandi flussi di dati video e audio. Un altro ruolo importante svolto dall'AI in questo contesto è quello di prevenzione ed allerta ai lavoratori in caso di imminenti pericoli, ciò soprattutto in ambienti di lavoro pericolosi, come le piattaforme petrolifere o i siti di ricostruzione. L'AI sta migliorando, inoltre, la sicurezza nell'ambito dei trasporti, in particolare con lo sviluppo di veicoli autonomi che permettono di ridurre gli incidenti causati da errori umani.
- *Adattabilità e risposta alle emergenze.* L'AI ha la capacità di adattarsi rapidamente a situazioni impreviste, che è fondamentale per mantenere l'efficienza e la sicurezza in condizioni mutevoli. Ad esempio, in risposta a interruzioni della catena di fornitura o a emergenze sanitarie globali, i sistemi di AI possono rapidamente riprogrammare le operazioni logistiche o riadattare le linee di produzione, garantendo continuità e riducendo l'impatto negativo.

L'integrazione dell'AI nell'automazione non solo ha potenziato l'efficienza operativa attraverso l'ottimizzazione dei processi e la riduzione dei costi, ma ha anche migliorato significativamente la sicurezza, sia riducendo i rischi di incidenti che aumentando la capacità di risposta alle emergenze. Questi miglioramenti sono cruciali per lo sviluppo sostenibile e responsabile di numerosi settori industriali e commerciali.

1.3 Tecnologie e metodi di implementazione dell'AI nell'automazione

È ormai risaputo che l'integrazione dell'AI nel mondo dell'automazione ha dato il via ad una rivoluzione in diversi settori dell'ambito industriale. In questa sezione illustreremo determinate tecnologie AI e vedremo come queste vengono impiegate ed implementate in determinate situazioni, creando dei sistemi automatici intelligenti. La scelta delle tecnologie di AI appropriate e la loro integrazione nei processi di automazione non sono solo questioni di capacità tecnologica, ma richiedono, anche, una profonda comprensione dei bisogni operativi e delle possibili limitazioni dei sistemi di automazione. Verrà, quindi, discusso come le tecnologie AI, selezionate per la loro idoneità specifica, vengano integrate in modo da migliorare flessibilità, efficienza ed efficacia dei processi di automazione. Di vitale importanza, inoltre, è il mantenimento dell'equilibrio tra l'introduzione di innovazioni tecnologiche e la necessità di mantenere sistemi affidabili e robusti, tramite una pianificazione ed implementazione sensata e strategica.

1.3.1 Tecnologie AI specifiche per l'automazione

Le tecnologie fondamentali dell'AI che vengono impiegate nell'automazione moderna, sono essenziali per consentire l'esistenza, appunto, dell'automazione intelligente. Queste comprendono una serie di algoritmi e metodologie grazie alle quali la macchina può adattarsi ed imparare dall'esperienza. Inoltre il loro studio ed incremento rappresentano la base per la stesura dell'automazione intelligente verso nuovi confini. Più specificatamente, tali tecnologie sono le seguenti:

- *Machine Learning (apprendimento automatico)*. L'apprendimento automatico è uno dei concetti focali dell'AI nell'automazione; questo consente ai sistemi di imparare e migliorare la loro prestazione tramite l'analisi di dati, evitando, così, che la macchina venga programmata ogni volta per ogni singolo scopo. In automazione ciò comporta la definizione di macchine in grado di ottimizzare i loro algoritmi attraverso l'analisi di dati operativi, producendo un miglioramento dell'efficienza e della precisione della loro attività.
- *Elaborazione del Linguaggio Naturale (NLP)*. L'NLP consente ai sistemi automatizzati di comprendere e interpretare il linguaggio umano; ciò li rende più accessibili e in grado di interagire più naturalmente con gli utenti. Questo principio trova applicazione in ambienti di automazione dove l'interfaccia uomo-macchina è cruciale, come nei sistemi di assistenza clienti automatizzati o nei dispositivi di controllo vocale.
- *Visione computerizzata*. La visione computerizzata permette ai sistemi automatizzati di "vedere" e interpretare l'ambiente circostante; ciò è essenziale per applicazioni come la robotica mobile e i veicoli autonomi. Questa tecnologia consente alle macchine di identificare oggetti, navigare nello spazio e svolgere compiti complessi che richiedono una percezione visiva.

- *Reti neurali e Deep Learning (apprendimento profondo)*. Il deep learning si identifica come un ramo del machine learning ed è basato su reti neurali artificiali; esso ha un ruolo cruciale nell'automazione, soprattutto nei sistemi che richiedono di analizzare una grande quantità di dati. Deep learning e reti neurali sono fondamentali in applicazioni come il riconoscimento facciale, l'analisi medica automatizzata e l'analisi predittiva nella manutenzione degli impianti.

Le sopraelencate tecnologie, oltre che incrementare le capacità dell'automazione, rendono ora possibile lo svolgimento di compiti e azioni considerati prima troppo complessi o sottili per essere portati a termine da una macchina. La loro continua integrazione nel mondo dell'automazione sta fondando un mondo nel quale i sistemi controllati non solo si limitano a svolgere compiti, ma anche a collaborare attivamente ed intelligentemente con gli esseri umani.

1.3.2 Integrazione efficiente dell'AI nei sistemi di automazione

L'equilibrio che deve esistere tra l'innovazione tecnologica e la praticità operativa è una delle sfide più importanti e significative, dovuta alla coesione tra AI e automazione. Un ruolo fondamentale in questo ambito è ricoperto dalla standardizzazione delle interfacce, la quale rende più fluida la comunicazione tra l'AI e i sistemi che ne traggono beneficio. In questo modo è consentita una maggiore scalabilità e facilità di manutenzione che, a loro volta, permettono aggiornamenti o modifiche senza dover interrompere l'intero sistema.

Nell'integrazione dell'AI si deve una grande importanza, come già citato, all'adattamento e all'apprendimento, i quali consentono ai sistemi "intelligenti" di adattarsi e migliorare le proprie performance tramite l'analisi di dati in tempo reale e l'implementazione di algoritmi.

Oltre che all'adattamento e alla standardizzazione, prima della piena implementazione, è necessario sottoporre i sistemi AI integrati a rigorosi test e processi di validazione, per garantire una maggiore efficienza e sicurezza all'interno dell'ambiente di lavoro; questi test sono di grande importanza in quanto assicurano la compatibilità e l'affidabilità del sistema AI nell'ampio contesto dell'automazione industriale.

1.4 Successi e Prospettive dell'AI nell'Automazione

Come ormai ben sappiamo, la coesistenza dell'AI e dell'automazione ha rivoluzionato e rinnovato numerosi settori industriali, sia in ambito di efficienza che di innovazione. Sono diverse le applicazioni che derivano da questa nuova automazione intelligente, dalle catene di produzione avanzate alla gestione ottimizzata delle risorse; in questa sezione, analizzeremo alcuni dei case study più emblematici.

Iniziamo con la definizione di case study: un caso di studio è un rapporto completo dei risultati dei test teorici o dell'esame dei temi emergenti di un'azienda nel contesto della vita reale. Vediamo ora, alcuni tra i case study più significativi:

- *Automazione avanzata nella produzione di veicoli elettrici*. Un esempio notevole è rappresentato dalla produzione di veicoli elettrici di Tesla. L'uso dell'AI nella robotica e nei sistemi di controllo della qualità ha permesso a Tesla di ottimizzare la produzione, riducendo i tempi di assemblaggio e aumentando la precisione nella costruzione dei veicoli.
- *Sistemi di Logistica Intelligenti*. Amazon ha implementato sistemi di automazione alimentati da AI nei suoi centri di distribuzione. Questi sistemi utilizzano robot per il picking e il packing, integrati con algoritmi avanzati per la gestione degli inventari e la

pianificazione delle spedizioni, migliorando notevolmente l'efficienza e riducendo gli errori.

- *Automazione nella Sanità.* L'utilizzo dell'AI in ambito sanitario, come negli ospedali dove robot dotati di Intelligenza Artificiale assistono in interventi chirurgici complessi, evidenzia il progresso nell'automazione medica. Questi robot possono eseguire movimenti precisi, riducendo i rischi e migliorando gli esiti degli interventi.

Dal precedente elenco possiamo percepire come l'AI non solo è di interesse in ambito industriale, ma sta anche diventando parte della quotidianità e si sta integrando sempre di più con l'essere umano. Con la continua evoluzione della tecnologia, è sempre più forte la consapevolezza che l'AI sarà al centro di nuovi progetti, sistemi ed applicazioni sempre più sofisticate ed integrate, ponendosi al centro dello sviluppo dell'automazione.

1.4.1 Progressi recenti e innovazioni emergenti

L'integrazione dell'AI con l'ingegneria dell'automazione è in continua espansione, portando allo sviluppo di ambienti di produzione sempre più dinamici, connessi, ma anche intrinsecamente più complessi, con ulteriori interdipendenze, incertezze e la generazione di grandi volumi di dati. I recenti progressi nell'Intelligenza Artificiale hanno mostrato il potenziale di questa tecnologia anche nell'ambito industriale. In particolare, si tratta di un alleato importante nell'affrontare le sfide associate a questa trasformazione digitale, grazie alla possibilità di eseguire analisi predittive basate sui dati e sulla capacità di assistere il processo decisionale in ambienti altamente complessi.

- *Manutenzione predittiva intelligente con l'AI.* L'analisi di grandi quantità di dati ottenuti da sensori posti direttamente sul sistema, in grado di captare, ad esempio, vibrazioni o temperature, consente non solo di definire lo stato della macchina, ma anche di effettuare previsioni relative ai guasti imminenti. Ciò non solo riduce i tempi di inattività, ma migliora anche la sicurezza e la longevità delle apparecchiature.
- *Robotica Collaborativa.* Alcuni task risultano ancora troppo complessi per essere svolti interamente da robot, ma ciò non implica che tali compiti non possano essere svolti dall'uomo con l'ausilio di sistemi intelligenti, sgravandolo da compiti ripetitivi e logoranti. L'AI adattiva consente alle macchine di imparare dai movimenti e dagli stili di lavoro umani; l'inserimento in queste ultime di celle robotiche che includono il riconoscimento automatico di persone ed oggetti consente la creazione di barriere virtuali, fonti di una collaborazione tra uomo e robot in completa sicurezza.
- *Quality inspection ottimizzata grazie all'AI.* L'Intelligenza Artificiale può essere istruita per il riconoscimento dei prodotti o dei pezzi difettosi; ciò è reso possibile dall'ausilio di moderni algoritmi di Image Processing, che consentono, quindi, di individuare parti difettose anche durante la realizzazione della catena di montaggio, e non solo esaminando il prodotto finito. Questa abilità porta ai vantaggi relativi al risparmio sui costi delle lavorazioni e ad un'analisi semplificata nell'individuare i difetti nel processo di produzione.

1.4.2 Prospettive future dell'AI nell'automazione

Il futuro dello sviluppo dell'AI nell'automazione è fiorente e ricco di opportunità grazie alle prospettive di sviluppo delle nuove tecnologie automatiche dotate di intelligenza artificiale. Si delinea, infatti, un panorama in cui l'innovazione tecnologica contribuisce alla sostenibilità

e all' integrazione sociale, consentendo previsioni di notevole crescita che porteranno l'AI a spingere l'automazione oltre i limiti che noi conosciamo.

A proposito di sostenibilità, quella ambientale è uno degli aspetti più promettenti in cui viene impiegata l'AI, che può consentire di gestire in maniera più appropriata risorse naturali ed energetiche, diminuendo sprechi e rendendo più efficienti i processi industriali, riducendo, quindi, l'impatto ambientale. Possiamo prendere come esempio impianti solari ed eolici, dove impianti sofisticati di AI potrebbero essere impiegati per migliorare la gestione delle energie rinnovabili, massimizzando la produzione energetica e minimizzando gli sprechi.

Parallelamente un altro forte impatto che l'AI avrà in futuro sarà quello sulle smart city, grazie al miglioramento della gestione del traffico e all'ottimizzazione dei servizi pubblici, potenziando, così, la qualità della vita. Esempi pratici potrebbero essere il monitoraggio dei flussi di traffico, riducendo, così, anche l'inquinamento atmosferico ed acustico, e il miglioramento della raccolta e del riciclo dei rifiuti, rendendo le città, in generale, più vivibili e sostenibili.

Nonostante i benefici che emergono da questi prospetti futuri, l'integrazione dell'Intelligenza Artificiale nella società non è esente da sfide, in particolare quelle relative all'etica e alla regolamentazione; la gestione di tematiche come la privacy dei dati, la sicurezza informatica, i diritti dei lavoratori e l'uso responsabile dell'AI risulterà fondamentale per mantenere armonia e consentire a macchine intelligenti e umani di coesistere.

In conclusione, le prospettive future dell'AI nell'automazione offrono un panorama di opportunità straordinarie, aperte a nuove applicazioni e innovazioni. Tuttavia, il successo di questa evoluzione tecnologica dipenderà dalla nostra capacità di gestire con saggezza e responsabilità le sfide che essa comporta, assicurando che i benefici siano distribuiti in modo equo e che gli impatti siano gestiti in modo etico e sostenibile.

Questo capitolo fornisce una panoramica approfondita di Gazebo, un software di simulazione ampiamente utilizzato nell'ingegneria dell'automazione. Esso inizia con un'introduzione a Gazebo, delineando la sua origine e il suo sviluppo, mettendo in luce le sue caratteristiche uniche e l'integrazione con l'Intelligenza Artificiale.

Successivamente, il capitolo esplora in dettaglio le caratteristiche tecniche di Gazebo, includendo una discussione sul formato SDFormat, per la definizione di modelli e scenari, l'architettura e il design del software, i motori di fisica e di simulazione, la cinematica e la modellazione dei robot, nonché le capacità di networking e integrazione con altri strumenti.

Il focus si sposta, poi, sull'uso pratico di Gazebo nell'ingegneria dell'automazione. Vengono presentati dei case study, i quali illustrano applicazioni specifiche di Gazebo in progetti di automazione, evidenziando le sfide e le limitazioni nell'utilizzo del software in questo ambito.

Infine, il capitolo si conclude con un esame delle innovazioni e degli aggiornamenti recenti di Gazebo, sottolineando come questi sviluppi stiano plasmando il futuro dell'ingegneria dell'automazione e della robotica. Attraverso questo capitolo, il lettore acquisisce una comprensione completa del ruolo critico di Gazebo nella simulazione e nello sviluppo di sistemi automatizzati.

2.1 Introduzione a Gazebo

Nel contesto dell'ingegneria informatica e dell'automazione, Gazebo si pone come uno strumento fondamentale per la simulazione di ambienti e robot. Questo software offre agli ingegneri ed ai ricercatori una piattaforma versatile e potente per testare algoritmi, progettare robot e simulare complessi scenari di automazione in un ambiente virtuale dettagliato e realistico. Nello specifico, Gazebo è una raccolta di librerie software open source progettate per semplificare lo sviluppo di applicazioni ad alte prestazioni. Il pubblico principale del software sono sviluppatori di robot, progettisti ed educatori. Tuttavia, Gazebo è strutturato per adattarsi a molti casi d'uso differenti, infatti, ogni libreria al suo interno ha dipendenze minime; ciò consente alle librerie di essere utilizzate in compiti che vanno dalla risoluzione di trasformazioni matematiche, alla codifica video, fino alla simulazione e alla gestione dei processi.

2.1.1 Origine e sviluppo di Gazebo

Gazebo è stato un componente del Player Project, un'iniziativa nata all'inizio degli anni 2000 che mira a sviluppare software libero per la ricerca in robotica, dal 2004 al 2011. Nel 2011, Gazebo è diventato un progetto indipendente supportato da Willow Garage; quest'ultima era un'organizzazione di ricerca e sviluppo privata focalizzata sulla robotica e sull'Intelligenza

Artificiale. Nel 2012, la Open Source Robotics Foundation (OSRF) è diventata il custode del progetto Gazebo. L'OSRF ha cambiato il suo nome in Open Robotics nel 2018.

L'ultima e più importante versione di Gazebo Classic è la Versione 11, una release di lungo termine con fine vita prevista per gennaio 2025. Tutte le versioni di Gazebo precedentemente rilasciate hanno ricevuto anche loro supporto a lungo termine, con aggiornamenti minori rilasciati per Gazebo 9 e 10 insieme al rilascio di Gazebo 11.0.0. Con il rilascio di Gazebo 11 a gennaio 2020, Open Robotics ha spostato il suo focus sullo sviluppo di Ignition, una raccolta di librerie software utili per semplificare lo sviluppo di applicazioni complesse, con un pubblico target di sviluppatori di robot, progettisti ed educatori. La prima versione di Ignition è stata rilasciata nel febbraio 2019. Nell'aprile 2022, a seguito di una disputa sui marchi, il nome Ignition è stato ritirato e si è tornati a Gazebo. Per evitare confusione tra la versione originale del simulatore e le sue iterazioni più recenti, il sito web di Gazebo distingue chiaramente la versione originaria del software, denominandola "Gazebo Classic", dalla versione attuale di Gazebo. Open Robotics ha menzionato la necessità di una significativa modernizzazione nel codice di Gazebo, insieme all'opportunità di passare da un'architettura monolitica a una raccolta di librerie poco accoppiate. Nella Figura 2.1 viene riportata la linea temporale della storia e dello sviluppo di Gazebo.

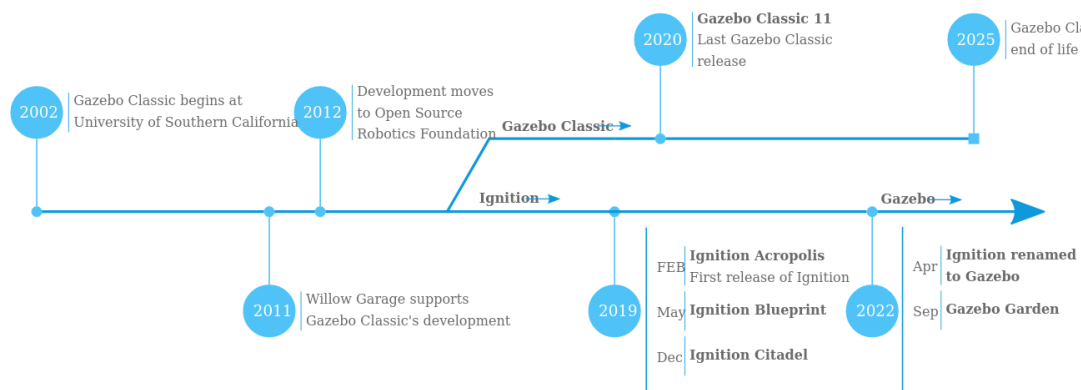


Figura 2.1: Linea temporale della storia e dello sviluppo di Gazebo

2.1.2 Caratteristiche uniche di Gazebo

Gazebo è uno dei software più adatti all'ingegneria dell'automazione e alla robotica grazie ad una serie di caratteristiche che integrano perfettamente il software ed i due settori appena citati.

Una delle caratteristiche principali di Gazebo è la sua capacità di simulare ambienti fisici con un alto grado di realismo. L'accurata modellazione della fisica, come la gravità, le collisioni e i materiali, oltre alla rappresentazione dettagliata di oggetti e superfici, consente la realizzazione di ambienti di simulazione notevolmente realistici. Questo livello di dettaglio permette agli ingegneri di testare i robot in scenari che simulano fedelmente il mondo reale, riducendo così il divario tra la simulazione e l'applicazione pratica.

Un'altra caratteristica degna di nota è il supporto di una vasta gamma di sensori e di robot, il che lo rende uno strumento estremamente versatile e completo. Gli utenti possono simulare una varietà di sensori, come lidar, telecamere e sensori di contatto, che sono fondamentali per la navigazione e l'interazione dei robot con l'ambiente. Questa flessibilità permette di testare diverse configurazioni di robot e sensori, e ciò è essenziale per la ricerca e lo sviluppo in ambito automazione.

Gazebo ha la capacità di integrarsi facilmente con altri software e framework di sviluppo robotico. Ciò include l'interfaccia con ROS (Robot Operating System), che consente una notevole espansione delle possibilità di simulazione e controllo. Tale caratteristica è fondamentale per testare algoritmi di Intelligenza Artificiale e per lo sviluppo di sistemi di controllo robotici complessi. Nella Figura 2.2 viene mostrato un esempio di ambiente industriale sviluppato con Gazebo.

Infine, Gazebo è dotato di una risorsa inestimabile, ovvero, il suo carattere open-source e la sua ampia comunità di utenti e sviluppatori. Questo non solo garantisce un continuo aggiornamento e miglioramento del software, ma fornisce anche una vasta gamma di risorse, come plugin, modelli e tutorial, che sono disponibili gratuitamente.

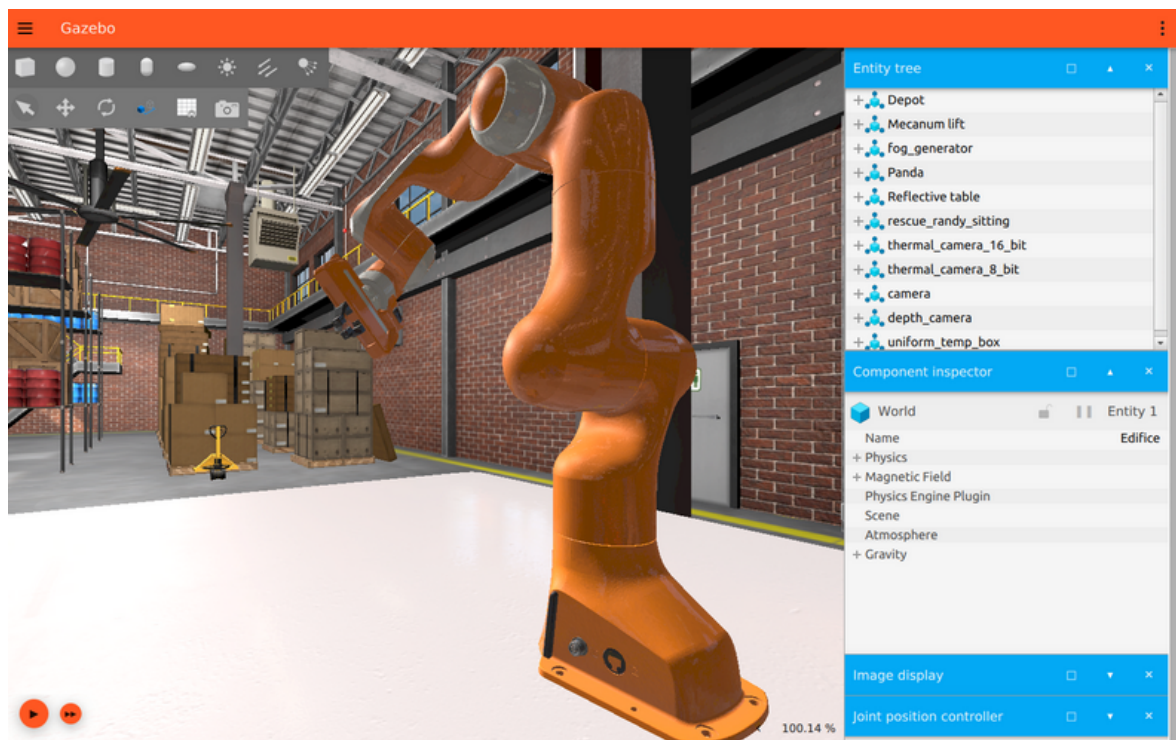


Figura 2.2: Esempio di un ambiente industriale, con annessa presenza di un braccio robotico, sviluppato con Gazebo

2.1.3 Integrazione con l'Intelligenza Artificiale

L'integrazione dell'Intelligenza Artificiale (AI) con Gazebo rappresenta un aspetto cruciale per l'avanzamento dell'ingegneria dell'automazione e della robotica. Questa sinergia tra simulazione e AI apre l'accesso a numerose possibilità per lo sviluppo e il testing di sistemi automatizzati intelligenti. Nello specifico, grazie a tale integrazione, è possibile fruire delle seguenti funzionalità:

- *Simulazione di comportamenti intelligenti.* Uno degli aspetti più significativi dell'integrazione tra Gazebo e l'AI è la capacità di simulare comportamenti intelligenti dei robot. Grazie alla simulazione di ambienti complessi, è possibile testare e affinare algoritmi AI per la navigazione autonoma, la manipolazione di oggetti e la risoluzione di problemi in contesti dinamici. Questa simulazione permette agli sviluppatori di valutare come i robot reagirebbero a situazioni impreviste, migliorando la loro affidabilità e la loro efficienza in scenari realistici.

- *Integrazione con framework di Machine Learning.* Gazebo si integra efficacemente con vari framework di apprendimento automatico e di deep learning, come, ad esempio, TensorFlow e PyTorch. Tale possibilità è essenziale per lo sviluppo di robot capaci di apprendere e adattarsi in base alle esperienze simulate. Gli algoritmi di apprendimento possono essere testati e ottimizzati in Gazebo prima di essere applicati in robot reali, riducendo i rischi ed i costi associati ai test in ambienti fisici.
- *Simulazioni per il Reinforcement Learning (apprendimento per rinforzo).* Il Reinforcement Learning è una branca del Machine Learning che coinvolge agenti software, i quali imparano a prendere decisioni ottimizzando le azioni tramite delle interazioni del tipo "trial-and-error" (eseguite per tentativi ed errori). Gazebo è particolarmente adatto per il rafforzamento dell'apprendimento; infatti, simulare scenari in cui i robot devono navigare attraverso ostacoli o compiere compiti specifici permette agli algoritmi di apprendimento per rinforzo di essere testati in modo sicuro e controllato.

2.2 Caratteristiche tecniche di Gazebo

2.2.1 SDFFormat

Iniziamo questa sezione introducendo SDFFormat (o, abbreviato, SDF), il linguaggio di definizione dei modelli di Gazebo. SDF (Simulation Description Format) è un formato XML che descrive oggetti e ambienti per simulatori di robot, visualizzazione e controllo. Originariamente sviluppato come parte del simulatore di robot Gazebo, SDFFormat è stato progettato pensando alle applicazioni robotiche scientifiche. Nel corso degli anni, esso è diventato un formato stabile, robusto ed estensibile, capace di descrivere tutti gli aspetti dei robot, oggetti statici e dinamici, illuminazione, terreno e, persino, la fisica.

2.2.2 Architettura e design del software

Per ciò che concerne tale aspetto Gazebo presenta le seguenti proprietà:

- *Struttura modulare di Gazebo.* Gazebo si distingue per la sua architettura modulare; con "modularità", si intende la progettazione di un qualsiasi sistema a partire da componenti separate, chiamate moduli, che sono connesse tra loro. Questa caratteristica consente agli utenti di personalizzare e ampliare le funzionalità secondo le specifiche esigenze del progetto. Tale struttura modulare è resa possibile grazie all'uso di plugin, che possono essere aggiunti o rimossi in base alle necessità. Questa flessibilità rende Gazebo uno strumento adatto a una vasta gamma di applicazioni nell'ingegneria dell'automazione, dalla progettazione di semplici robot a sistemi complessi multi-robot.
- *Interfaccia Utente Grafica (GUI).* La GUI di Gazebo è progettata per essere intuitiva e user-friendly, permettendo anche agli utenti meno esperti di navigare e manipolare l'ambiente di simulazione con facilità. Attraverso la GUI, gli utenti possono inserire, configurare e interagire con modelli di robot, oggetti, e scenari di simulazione. La GUI offre, inoltre, funzionalità avanzate come la visualizzazione in tempo reale di dati di simulazione, il controllo di telecamere e sensori, e le modifiche interattive dell'ambiente. Analizziamo ora, come appare graficamente la GUI ed alcune delle sue funzionalità basilari: nella Figura 2.3 possiamo osservare come appare l'interfaccia grafica relativa ad un mondo (chiamato shapes.sdf) messo a disposizione direttamente da Gazebo per eseguire tutorial ed esempi. Proprio per il suo scopo, il mondo appare semplicemente dotato di alcune figure geometriche, che l'utente può spostare e modificare. La barra

degli strumenti nella parte superiore dell'interfaccia grafica contiene due pulsanti: un pulsante del menù file sulla sinistra e un pulsante dei plugin sulla destra. Il pulsante del menù file consente di salvare il mondo in un file, salvare e caricare configurazioni dell'interfaccia e personalizzare le impostazioni dello stile dell'interfaccia. Una configurazione dell'interfaccia consiste nei plugin caricati e nelle impostazioni di stile in atto.

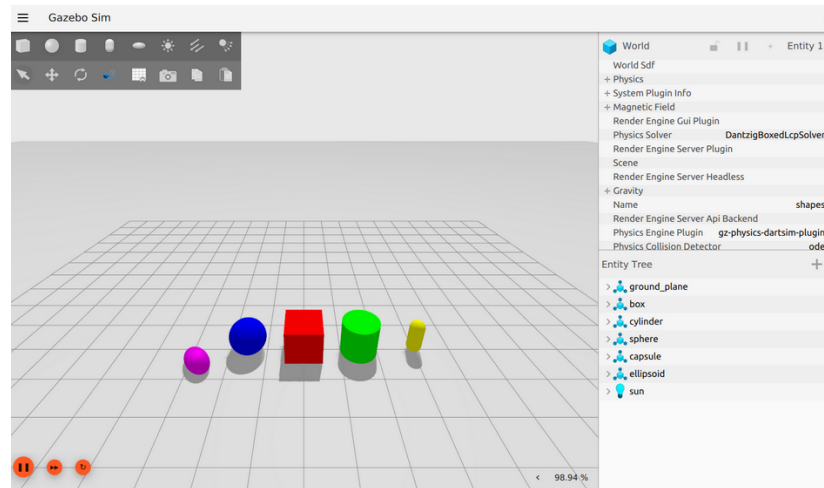


Figura 2.3: Interfaccia grafica di Gazebo all'avvio del mondo `shapes.sdf`

La barra degli strumenti in alto a sinistra, sospesa sotto il pulsante del menù file, contiene pulsanti per forme (sfera, scatola, cilindro, capsula ed ellissoide), così come controlli per l'illuminazione (punto, direzionale e spot). I pulsanti delle forme consentono di inserire direttamente nel mondo un modello di scatola, sfera, cilindro, capsula o ellissoide. La forma, dopo essere stata selezionata ed inserita nel mondo, si aggancerà automaticamente al piano del terreno, e le sue proprietà saranno visualizzate nella finestra del plugin Entity Tree (albero delle entità). I primi cinque pulsanti sulla seconda riga sono per il Controllo della Trasformazione.

Il pulsante dei plugin elenca tutti i plugin disponibili. È possibile scorrere tale lista per vederli e selezionare quello adeguato o necessario per la propria versione di Gazebo. Quando ne viene selezionato uno, la sua interfaccia apparirà nel pannello di destra, dove sono contenuti due plugin che si vedono a partire dall'avvio. In alto si trova l'Ispettore dei Componenti (component inspector), seguito dall'Albero delle Entità. Le entità presenti nell'Entity Tree possono essere espansive per rivelare i loro collegamenti, così come le corrispondenti visualizzazioni, collisioni, e giunture, se esistono.

- *Integrazione e compatibilità.* L'architettura di Gazebo è progettata per garantire l'integrazione con una varietà di altri strumenti e piattaforme, inclusi sistemi di controllo e pacchetti software per la robotica. Questa compatibilità è cruciale per permettere agli utenti di incorporare Gazebo in flussi di lavoro esistenti e di sfruttare altre risorse software per massimizzare l'efficacia delle simulazioni.
- *Estendibilità e Personalizzazione.* Gazebo permette agli utenti di sviluppare e integrare i propri plugin, offrendo un alto grado di personalizzazione. Questo aspetto è particolarmente importante in campi come l'ingegneria dell'automazione, dove ogni progetto può avere esigenze uniche. La capacità di personalizzare e estendere il software consente a Gazebo di adattarsi a una vasta gamma di scenari di simulazione e requisiti tecnici.

2.2.3 Motori di fisica e simulazione in Gazebo

L'integrazione, da parte di Gazebo, di diversi motori di fisica, consente agli utenti di poter scegliere quello più adatto alle proprie esigenze di simulazione. Tra i motori più noti supportati da Gazebo ci sono ODE (Open Dynamics Engine), Bullet, Simbody e DART. Ognuno di questi motori offre caratteristiche specifiche in termini di precisione, performance e tipi di simulazione che possono gestire. Per esempio, mentre ODE è ampiamente usato per la sua robustezza generale, Bullet è spesso preferito per simulazioni che richiedono interazioni fisiche complesse.

Come già preannunciato, la capacità di Gazebo di simulare una vasta gamma di sensori ed attuatori è un altro aspetto fondamentale. Questo include sensori come lidar, fotocamere, sensori di contatto e GPS, così come attuatori come motori elettrici e pistoni idraulici. La simulazione realistica di questi componenti è essenziale per validare algoritmi di controllo e per assicurare che i sistemi progettati funzionino come previsto nel mondo reale. Inoltre, Gazebo permette di testare i sensori e gli attuatori in diversi ambienti e condizioni, fornendo un feedback prezioso per l'ottimizzazione dei progetti.

La combinazione di diversi motori di fisica e la simulazione dettagliata di sensori e attuatori rendono Gazebo uno strumento estremamente potente per l'ingegneria dell'automazione e la robotica. Questa versatilità permette di simulare scenari complessi, dalla robotica mobile all'automazione industriale, facilitando lo sviluppo e il testing di sistemi automatizzati in un ambiente sicuro e controllato.

2.2.4 Cinematica e modellazione dei robot

In questa sezione analizzeremo come Gazebo gestisce la cinematica e la modellazione dei robot, elementi cruciali per qualsiasi simulazione realistica nel campo dell'ingegneria dell'automazione. La modellazione accurata dei robot in Gazebo si basa su diversi componenti fondamentali: modelli, link, giunti e posizioni. Ciascuno di questi elementi svolge un ruolo specifico nel definire il comportamento fisico e le capacità di movimento dei robot all'interno dell'ambiente di simulazione. Analizziamoli separatamente:

- *Modelli*. Il tag `<model>` funge da contenitore per insiemi di link e giunture. I modelli necessitano di un nome, che viene richiesto tramite l'attributo `name`. Ad esempio, un modello vuoto senza giunture o link viene definito nel seguente modo:

```
<model name="empty" />
```

Inoltre, un modello può stabilire una posizione che modifica l'orientamento del suo frame in relazione al suo elemento genitore, al momento dell'inserimento in un ambiente di simulazione.

```
<model name="model_with_pose">  
  <pose>{xyz_WM} {rpy_WM}</pose>  
</model>
```

- *Link*. Il tag `<link>` rappresenta un corpo rigido nominato che deve essere figlio di un `<model>`. Anche i link richiedono l'attributo `name`:

```
<model name="one_link">
  <link name="link"/>
</model>
```

I link fratelli appartenenti allo stesso modello devono avere nomi univoci, ma non uguali, altrimenti il modello risulta non valido. Ad esempio:

```
<model name="two_links">
  <link name="link1"/>
  <link name="link2"/>
</model>
```

Ogni link ha un frame fissato al corpo, la cui posa iniziale rispetto al frame del modello è specificata dal tag `<pose>` del link.

```
<model name="model_and_link_pose">
  <pose>{xyz_WM} {rpy_WM}</pose>
  <link name="link">
    <pose>{xyz_ML} {rpy_ML}</pose>
  </link>
</model>

<model name="equivalent_link_pose">
  <link name="link">
    <pose>{xyz_WL} {rpy_WL}</pose>
  </link>
</model>
```

- *Giunti*. Il tag `<joint>` rappresenta una relazione cinematica tra collegamenti di corpi rigidi che vincola i corrispettivi gradi di libertà. Deve essere figlio di un modello.

Ci sono diversi tipi di giunti che possono essere specificati nell'attributo `type`; essi sono elencati di seguito, insieme al numero di gradi di libertà rimanenti tra i due collegamenti:

- ball: 3 gradi di libertà rotazionali;
- continuous: 1 grado di libertà rotazionale senza limiti di giunto;
- fixed: 0 gradi di libertà;
- prismatic: 1 grado di libertà traslazionale;
- revolute: 1 grado di libertà rotazionale con limiti di giunto;
- screw: 1 grado di libertà rotazionale/traslazionale accoppiato;
- universal: 2 gradi di libertà rotazionali;

I collegamenti padre e figlio di un giunto sono specificati, rispettivamente, dai tag `<parent>` e `<child>`. Questi tag si riferiscono ad altri collegamenti all'interno del modello tramite i loro nomi. Ad esempio, il seguente modello contiene due collegamenti e un giunto fisso tra i collegamenti:


```

<model name="two_links_fixed">
  <link name="link1"/>
  <link name="link2"/>
  <joint name="joint" type="fixed">
    <parent>link1</parent>
    <child>link2</child>
  </joint>
</model>

```

Come discusso precedentemente, la specifica richiede che tutti gli elementi fratelli dello stesso tipo abbiano nomi univoci. Questo permette tecnicamente a un collegamento e a un giunto fratelli di avere lo stesso nome, come nell'esempio seguente, ma ciò non è consigliato, poiché, può portare a confusione e potrebbe essere proibito da una versione futura della specifica.

```

<model name="sibling_link_joint_namesake_not_recommended">
  <link name="base"/>
  <link name="attachment"/>
  <joint name="attachment" type="fixed">
    <parent>base</parent>
    <child>attachment</child>
  </joint>
</model>

```

Per i tipi di giunto che hanno uno o due gradi di libertà, le proprietà degli assi di rotazione/traslazione sono specificate dai tag `<axis>` e `<axis2>`. Entrambi contengono il tag `<xyz>` che specifica il vettore unitario lungo l'asse di movimento. Nelle versioni SDFFormat 1.4 e precedenti, questo vettore unitario è espresso nel frame del modello del collegamento genitore. Nelle versioni SDFFormat 1.5 e 1.6, di default, il vettore unitario è espresso nel frame del giunto che contiene il tag `<axis>`. Tuttavia, il tag `<use_parent_model_frame>` può essere impostato su true per specificare il vettore unitario nel frame del modello del collegamento genitore. Questo viene utilizzato per replicare il comportamento del tag `<xyz>` nella Versione 1.4 e precedenti. Da notare che il tag `<use_parent_model_frame>` potrebbe essere rimosso in una versione futura di SDFFormat. Vediamo un esempio:

```

<model name="joint_axis">
  <link name="A"/>
  <link name="B"/>
  <link name="C"/>
  <joint name="J1" type="revolute">
    <pose>0 0 0 1.57 0 0</pose>
    <parent>A</parent>
    <child>B</child>
    <axis>
      <xyz>0 0 1</xyz>
    </axis>
  </joint>

```

```

<joint name="J2" type="revolute">
  <parent>B</parent>
  <child>C</child>
  <axis>
    <xyz>0 0 1</xyz>
    <use_parent_model_frame>true</use_parent_model_frame>
  </axis>
</joint>
</model>

```

Per un giunto con frame del collegamento genitore P e frame del collegamento figlio C , il suo tag `<pose>` specifica la posa X_{CJ_C} di un frame del giunto J_C rigidamente attaccato al collegamento figlio. Allo stesso modo, un frame J_P è rigidamente attaccato al corpo genitore. Questi frame sono illustrati nella Figura 2.4.

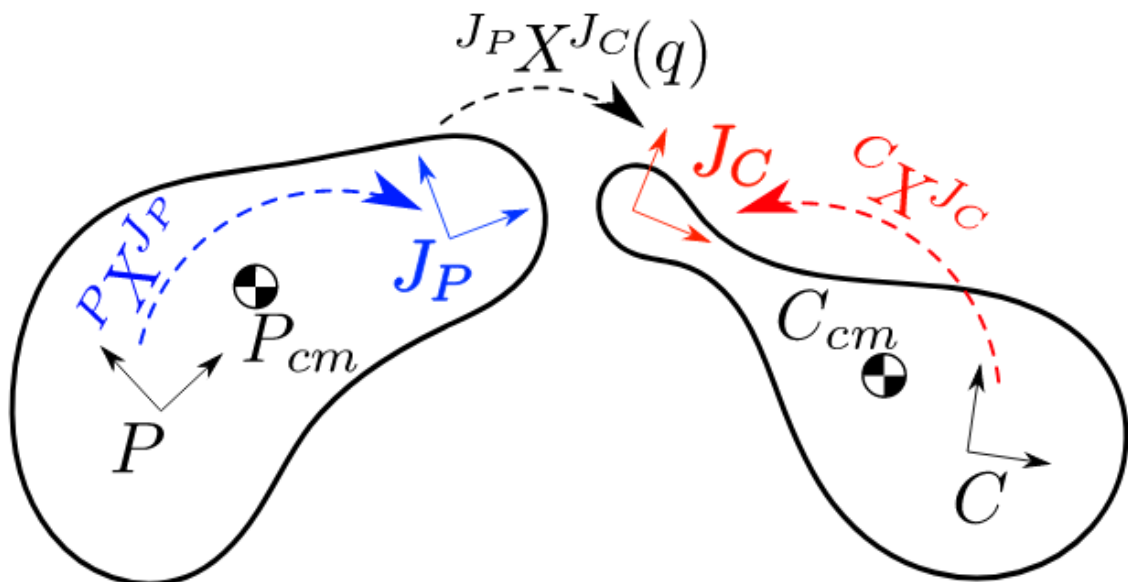


Figura 2.4: La figura illustra un giunto meccanico dove il frame J_C è solidale al collegamento figlio C , e il frame J_P è solidale al collegamento genitore P , con la posa X_{CJ_C} che definisce la configurazione spaziale di J_C rispetto a C .

La posa del frame J_P in P non è specificata esplicitamente, ma è dedotta dalle pose di P e C date nel frame del modello alla "configurazione zero" del giunto. La maggior parte dei tipi di giunti specifica come i frame J_P e J_C possano muoversi l'uno rispetto all'altro (con l'eccezione notevole del giunto fisso). Una configurazione zero è definita per ogni tipo di giunto (ad esempio, angolo zero per un giunto revolute) in modo tale che i frame J_P e J_C coincidano nella configurazione zero. Sebbene un tag `<pose>` non sia necessario per un giunto fisso, viene utilizzato regolarmente in altri tipi di giunti.

2.2.5 Networking e integrazione con altri strumenti in Gazebo

In questa sezione, analizzeremo l'integrazione di Gazebo con reti di computer e altri strumenti software, evidenziando la sua versatilità e la capacità di funzionare in sistemi complessi di automazione e robotica.

- *Capacità di networking di Gazebo.* Una caratteristica distintiva di Gazebo è la sua abilità nel supportare la simulazione distribuita attraverso il networking. Ciò permette agli utenti di eseguire simulazioni che si estendono su più macchine o sistemi in rete, facilitando la collaborazione e la condivisione delle risorse. Per esempio, in un contesto di ricerca e sviluppo, diversi team possono lavorare contemporaneamente su aspetti differenti della stessa simulazione, sia che si tratti di testare algoritmi di controllo, valutare la risposta dei sensori, o ottimizzare la progettazione meccanica dei robot.
- *Integrazione con strumenti e piattaforme esterni.* Gazebo, come già preannunciato in precedenza, non opera isolatamente ma è progettato per interagire efficacemente con una varietà di strumenti e piattaforme esterni. Tale capacità si estende a vari aspetti, dal controllo dei robot fino alla raccolta e analisi dei dati di simulazione. Uno degli esempi più notevoli è l'integrazione di Gazebo con ROS (Robot Operating System), una piattaforma ampiamente utilizzata per lo sviluppo di software robotico. Questa sinergia permette di simulare robot e ambienti in Gazebo, utilizzando, allo stesso tempo, gli strumenti di controllo e di analisi dati forniti da ROS.
- *API e personalizzazione.* Gazebo offre un'API (Interfaccia di Programmazione Applicativa) ben documentata. Infatti, dando la possibilità agli sviluppatori di creare plugin personalizzati o di integrare il software con sistemi esistenti, si ha il vantaggio di adattare Gazebo per soddisfare le esigenze specifiche di un progetto, che si tratti di aggiungere nuovi tipi di sensori, implementare algoritmi di controllo personalizzati, o collegare il simulatore con altri sistemi software per analisi più approfondite.

In conclusione, la capacità di Gazebo di collegarsi e interagire con vari strumenti e piattaforme lo rende uno strumento estremamente flessibile e potente nell'ambito dell'ingegneria dell'automazione e della robotica. La sua compatibilità con il networking e l'integrazione con sistemi esterni apre a un'ampia gamma di applicazioni, a partire dalla formazione e ricerca accademica fino allo sviluppo di soluzioni industriali avanzate.

2.3 Utilizzo di Gazebo nell'ingegneria dell'automazione

Nell'ambito dell'ingegneria dell'automazione, l'utilizzo di Gazebo in combinazione con ROS ha permesso di esplorare nuove frontiere nella simulazione robotica. Questi strumenti offrono ambienti virtuali estremamente dettagliati e realistici, essenziali per il testing e lo sviluppo di sistemi robotici innovativi. I seguenti casi di studio illustrano l'ampia gamma di applicazioni e i benefici derivanti dall'uso di Gazebo.

- *Unmanned Ground Vehicle Modelling in Gazebo/ROS-Based Environments.* Nell'ambito della modellazione di veicoli terrestri senza equipaggio (Unmanned Ground Vehicle-UGV), l'uso di Gazebo si è rivelato cruciale. Questo studio ha impiegato Gazebo/ROS per simulare ambienti complessi e variabili, consentendo ai ricercatori di testare e perfezionare sistemi di navigazione autonomi in condizioni controllate, ma anche realistiche. La simulazione ha consentito una sperimentazione sicura e dettagliata di strategie di movimento e di interazione con l'ambiente, elementi fondamentali nell'automazione e nell'IA applicata ai sistemi mobili. L'impiego di Gazebo in questo contesto ha permesso di affrontare sfide legate alla percezione sensoriale e al decision-making autonomo, due pilastri fondamentali dell'automazione moderna. (Figura 2.5).
- *ROS/Gazebo-based Simulation of Quadcopter Aircrafts.* La simulazione di quadricotteri in Gazebo/ROS ha aperto nuove frontiere nella progettazione e nel testing di questi

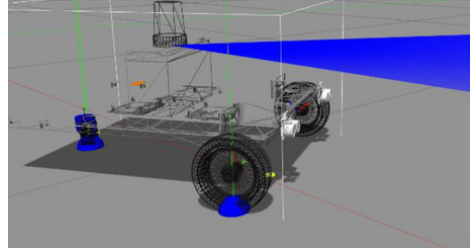


Figura 2.5: Modello multibody 3D di un robot mobile su ruote nell'ambiente Gazebo

velivoli. La sfida principale era creare un modello di simulazione accurato che potesse replicare fedelmente la complessa dinamica di volo dei quadricotteri. Attraverso la simulazione, i ricercatori hanno potuto testare algoritmi avanzati per il controllo del volo e la stabilità, evitando, così, danni ai costosi prototipi fisici. In questo contesto, Gazebo ha fornito un ambiente virtuale estremamente versatile, che ha permesso di esplorare l'integrazione di sistemi di IA per il volo autonomo e la reattività in scenari dinamici.

- *ROS/Gazebo based simulation of co-operative UAVs.* Un UAV (Unmanned Aerial Vehicle), comunemente noto come drone, è un tipo di aereo senza equipaggio umano a bordo. La simulazione di UAV cooperativi in Gazebo ha permesso di studiare e sviluppare algoritmi per la collaborazione e la comunicazione tra più droni. Questa ricerca è fondamentale per applicazioni come il monitoraggio ambientale e le operazioni di soccorso, dove il coordinamento è essenziale. Utilizzando Gazebo, i ricercatori hanno creato scenari complessi dove diversi UAV dovevano interagire e collaborare per raggiungere un obiettivo comune. Ciò ha richiesto lo sviluppo di modelli di IA avanzati per la gestione della cooperazione, la navigazione e la divisione dei compiti, sottolineando l'importanza di Gazebo nell'ambito dell'AI applicata alla robotica cooperativa.
- *Toward Maritime Robotic Simulation in Gazebo.* Infine, il caso studio sulla simulazione robotica marittima in Gazebo ha aperto nuove prospettive nell'esplorazione e nelle operazioni subacquee. Questo studio ha affrontato la sfida di simulare le complesse interazioni tra robot e ambiente marino, come le correnti e le interazioni con elementi sottomarini. L'utilizzo di Gazebo ha permesso di testare e ottimizzare i robot per queste condizioni uniche, integrando algoritmi di IA per la navigazione e il riconoscimento degli ambienti subacquei. Questa ricerca evidenzia il potenziale di Gazebo nell'automazione e nell'IA in ambienti marittimi, un campo ancora relativamente inesplorato ma di grande interesse per il futuro della robotica. (Figura 2.6).



Figura 2.6: Il mondo VRX Sand Island renderizzato in Gazebo con un grafico della velocità del vento nel tempo

Ognuno di questi studi dimostra come l'uso di Gazebo e ROS possa portare a significativi avanzamenti nella robotica, consentendo lo sviluppo di soluzioni più sofisticate e adatte a una vasta gamma di applicazioni. Attraverso la simulazione, è possibile sperimentare e

ottimizzare sistemi robotici in ambienti controllati, riducendo tempi e costi e aumentando la sicurezza e l'efficacia delle soluzioni proposte.

2.3.1 Sfide e limitazioni nell'uso di Gazebo

Nell'esplorazione delle potenzialità di Gazebo come strumento di simulazione nell'ingegneria dell'automazione, è essenziale considerare anche le sfide e le limitazioni che accompagnano il suo utilizzo. Questa sezione mira a esaminare tali aspetti, fondamentali per un'analisi equilibrata e completa.

- *Complessità nella Modellazione Realistica.* Una delle sfide principali nell'uso di Gazebo riguarda la creazione di modelli e ambienti di simulazione estremamente realistici. La riproduzione fedele di ambienti reali richiede una notevole attenzione ai dettagli e competenze avanzate in modellazione 3D, oltre a una comprensione approfondita della fisica dei materiali e delle dinamiche ambientali. Questo può rappresentare un ostacolo significativo, specialmente in progetti con risorse limitate o con requisiti di simulazione particolarmente sofisticati.
- *Interfaccia Utente e Curva di Apprendimento.* L'interfaccia utente di Gazebo, sebbene funzionale, può risultare poco intuitiva per i nuovi utenti, con una curva di apprendimento piuttosto ripida. Questo aspetto può rallentare l'avanzamento dei progetti, in particolare per coloro che sono alle prime armi con la simulazione robotica o con il software stesso. Una formazione adeguata e risorse di supporto adeguate sono cruciali per superare queste difficoltà iniziali.
- *Integrazione con Altri Sistemi e Tecnologie.* Sebbene Gazebo sia ben integrato con ROS, l'interoperabilità con altri sistemi software e hardware può, talvolta, presentare delle sfide. La compatibilità con diversi formati di dati, protocolli di comunicazione o hardware specifici richiede spesso soluzioni personalizzate, che possono richiedere competenze tecniche specialistiche e un lavoro di sviluppo aggiuntivo.

In conclusione, pur essendo uno strumento potente e versatile, Gazebo presenta delle sfide che richiedono attenzione. La consapevolezza di queste limitazioni è fondamentale per gli sviluppatori e i ricercatori che si avvicinano a Gazebo, permettendo loro di pianificare in modo adeguato e di cercare soluzioni alternative quando necessario. Tale approccio equilibrato è essenziale per sfruttare al meglio le capacità di Gazebo nel campo dell'automazione e dell'Intelligenza Artificiale.

2.3.2 Innovazioni ed aggiornamenti recenti in Gazebo

Gli ultimi anni hanno visto un notevole sviluppo in Gazebo. Gli aggiornamenti si sono concentrati su diversi fronti, migliorando significativamente sia l'usabilità che le capacità tecniche del software.

Innanzitutto, l'interfaccia utente di Gazebo ha subito trasformazioni che ne hanno aumentato l'intuitività e l'accessibilità. Questi miglioramenti includono una visualizzazione 3D più avanzata, strumenti di modellazione semplificati e una documentazione più esaustiva. Queste modifiche rendono l'esperienza utente molto più agevole, specialmente per i principianti del software.

Dal punto di vista tecnico, l'integrazione con ROS è stata notevolmente rafforzata, consentendo una maggiore interoperabilità con diverse versioni di ROS e altri sistemi esterni. Ciò ha reso Gazebo uno strumento più versatile e flessibile, capace di adattarsi a una varietà più ampia di ambienti e applicazioni di simulazione. Importanti sono stati anche i progressi nella

gestione delle prestazioni, in particolare per quanto riguarda la scalabilità e l'efficienza delle simulazioni complesse. Questo aspetto è fondamentale per affrontare scenari che coinvolgono un gran numero di entità o ambienti altamente dettagliati.

Le funzionalità di simulazione stesse sono state ampliate, con l'aggiunta di nuove capacità, come la simulazione migliorata di fenomeni fisici complessi, tra cui fluidi e materiali deformabili. Tali innovazioni aprono nuove possibilità in settori come la robotica marittima e la simulazione di interazioni robot-ambiente in contesti dinamici.

Infine, il ruolo della comunità open source è stato fondamentale per guidare lo sviluppo continuo di Gazebo. Il contributo attivo degli utenti e degli sviluppatori ha portato a un flusso costante di idee, miglioramenti e correzioni di bug, consolidando la posizione di Gazebo come strumento chiave nel panorama della simulazione robotica.

In sintesi, le recenti innovazioni in Gazebo hanno notevolmente ampliato le sue capacità e applicazioni, rendendolo uno strumento ancora più prezioso per la ricerca e lo sviluppo in automazione e Intelligenza Artificiale. Questi sviluppi non solo migliorano la qualità delle simulazioni ma offrono anche nuove opportunità per esplorare e sviluppare tecnologie robotiche avanzate.

Esperienza relativa al sensore IMU

Questo capitolo esplora il ruolo e l'implementazione del sensore IMU (Unità di Misura Inerziale) in un contesto di simulazione robotica utilizzando Gazebo. Si inizia con una panoramica delle caratteristiche tecniche e dei componenti del sensore IMU, sottolineando la sua importanza fondamentale nell'automazione e nella robotica per la precisione nel monitoraggio del movimento e dell'orientamento.

La sezione successiva descrive in dettaglio la configurazione del sensore IMU nel modello di un robot a tre ruote in Gazebo. Viene illustrato dettagliatamente il processo di implementazione del sensore IMU, prestando attenzione alla spiegazione del codice impiegato per implementare il plugin del sensore IMU e per integrare quest'ultimo nel modello del robot in questione.

Successivamente, il capitolo si focalizza su una serie di osservazioni pratiche, dove il movimento del robot e i cambiamenti nei valori del sensore IMU vengono analizzati per valutare il comportamento e le prestazioni del sensore. Questa sezione fornisce una comprensione diretta di come i dati dell'IMU rispondono a vari stimoli in un ambiente simulato.

Il capitolo si conclude con un riepilogo dell'esperienza di utilizzo del sensore IMU in Gazebo, riflettendo su come quest'ultima si inserisca nel contesto più ampio della robotica e dell'ingegneria dell'automazione. Questa analisi fornisce una comprensione preziosa dell'importanza degli IMU nei sistemi robotici e sottolinea il valore della simulazione nella verifica e nello sviluppo di tecnologie robotiche.

3.1 Introduzione al sensore IMU

L'Unità di Misura Inerziale (IMU-Inertial Measurement Unit) rappresenta una tecnologia fondamentale nell'ambito dell'ingegneria dell'automazione e della robotica. Questo dispositivo è essenziale per fornire dati precisi sulla posizione, sull'orientamento, sulla velocità e sull'accelerazione di un oggetto mobile, come un robot o un veicolo automatizzato. Un sensore IMU tipicamente combina accelerometri e giroscopi per tracciare sia il movimento lineare che quello rotazionale lungo i tre assi X, Y e Z (Figura 3.1). Questa combinazione di dati permette al sensore in questione di fornire una rappresentazione completa del movimento e dell'orientamento di un oggetto in uno spazio tridimensionale. Inoltre, moderni sensori IMU possono includere anche magnetometri per migliorare la precisione nella determinazione dell'orientamento rispetto al campo magnetico terrestre.

3.1.1 Caratteristiche tecniche e componenti del sensore IMU

Il sensore IMU è un componente tecnologico sofisticato, fondamentale nell'ambito della robotica e dell'automazione. Esso combina più dispositivi di misurazione per fornire dati precisi sulla posizione, l'orientamento e il movimento di un oggetto. Le sue caratteristiche tecniche chiave comprendono:

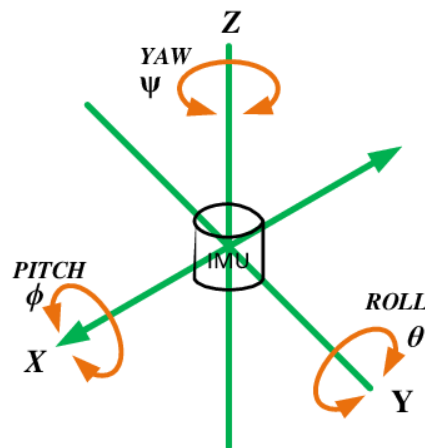


Figura 3.1: Rappresentazione degli assi e degli angoli pitch, roll e yaw di cui si serve l'IMU per eseguire le misurazioni

- *Componenti integrati:*
 - *Accelerometri:* misurano l'accelerazione lineare lungo gli assi X, Y e Z. Essenziali per determinare la velocità e la direzione del movimento lineare.
 - *Giroscopi:* forniscono dati sulla velocità angolare, ossia sulla rapidità di rotazione attorno agli assi X, Y e Z. Questi sono cruciali per comprendere l'orientamento e la rotazione dell'oggetto.
 - *Magnetometri (opzionali):* in alcuni modelli di IMU sono presenti anche magnetometri, che misurano l'orientamento rispetto al campo magnetico terrestre, migliorando ulteriormente la precisione dell'orientamento.
- *Precisione e affidabilità:* gli IMU sono progettati per essere estremamente precisi, con errori minimi nelle misurazioni. Tuttavia, possono essere soggetti a deriva nel tempo, soprattutto nei giroscopi, i quali richiedono tecniche di calibrazione e correzione.
- *Resistenza alle interferenze:* a differenza di altri sensori che possono essere influenzati da fattori ambientali, come la luce o gli ostacoli, gli IMU forniscono dati affidabili anche in condizioni avverse o in ambienti chiusi, rendendoli ideali per applicazioni come la navigazione interna.
- *Interfaccia e integrazione:* gli IMU sono progettati per essere facilmente integrati con altri sistemi, grazie a interfacce standardizzate. Ciò permette di collegarli con una varietà di dispositivi e piattaforme, inclusi sistemi di controllo e software di simulazione come Gazebo.
- *Dimensioni e consumo energetico:* i moderni IMU sono spesso compatti e leggeri, con un basso consumo energetico, rendendoli adatti per l'uso in dispositivi portatili e veicoli autonomi, dove lo spazio e l'energia sono fattori critici.

In conclusione, le caratteristiche tecniche degli IMU li rendono strumenti indispensabili nell'ingegneria dell'automazione e della robotica. La loro capacità di fornire dati accurati e affidabili sull'orientamento e il movimento in una varietà di condizioni li rende versatili e preziosi per un'ampia gamma di applicazioni.

3.1.2 Importanza del sensore IMU nell'automazione e nella robotica

Nel contesto dell'automazione e della robotica, l'importanza del sensore IMU è molteplice. In primo luogo, esso offre ai sistemi automatizzati la capacità di comprendere e reagire al proprio ambiente fisico, il che è cruciale per le applicazioni che richiedono un alto grado di precisione e affidabilità, come i veicoli autonomi, i droni e i sistemi di navigazione interna. Un ulteriore aspetto dell'IMU è la capacità di fornire dati in tempo reale, caratteristica fondamentale per il controllo dinamico di robot e macchinari; ciò permette ad essi di adattarsi rapidamente a variazioni ambientali o a modifiche nel loro compito operativo.

In questo capitolo, dopo aver delineato le basi teoriche e pratiche del sensore IMU, esploreremo la sua configurazione e integrazione nel software di simulazione Gazebo. Ciò includerà una discussione dettagliata su come il sensore IMU sia stato configurato e utilizzato nel contesto di un robot a tre ruote durante un esercizio pratico. Attraverso questo esempio, evidenzieremo le specifiche tecniche del sensore, il processo di raccolta e analisi dei dati, e, infine, le sfide e le opportunità che emergono dal suo utilizzo in scenari di simulazione.

3.2 Configurazione del sensore IMU in Gazebo

Il processo di configurazione del sensore IMU in Gazebo è fondamentale per simulare in modo accurato l'orientamento e il movimento del robot impiegato per la simulazione. Nel seguito di questa sezione analizzeremo, quindi, passo per passo, la configurazione del sensore IMU.

Per iniziare, è necessario l'utilizzo di un mondo dove è stato definito il modello relativo al robot e dove sono stati realizzati i processi per consentire ad esso di muoversi, nell'interfaccia grafica, tramite tastiera. Il mondo in questione è nominato `moving_robot.sdf`; per comodità, e distinzione tra i tutorial eseguiti, il mondo che considereremo noi per la configurazione del sensore IMU (e, successivamente, degli altri sensori) è il mondo `sensor_tutorial.sdf`, dove, all'interno, sono presenti le righe di codice relative al mondo `moving_robot.sdf`. In questo modo abbiamo già il nostro robot, che possiamo vedere nella Figura 3.2, definito e abile di muoversi.

Per la definizione del sensore IMU all'interno di questo mondo, si inserisce il seguente blocco di codice sotto il tag `<world>`:

```
<plugin filename="gz-sim-imu-system"
  name="gz::sim::systems::Imu">
</plugin>
```

Questo codice definisce il plugin del sensore IMU da utilizzare nel nostro mondo virtuale. Il passaggio successivo consiste nell'aggiungere effettivamente il sensore IMU al robot, inserendo il seguente codice:

```
<sensor name="imu_sensor" type="imu">
  <always_on>1</always_on>
  <update_rate>1</update_rate>
  <visualize>true</visualize>
  <topic>imu</topic>
</sensor>
```

Il sensore è generalmente aggiunto a uno dei collegamenti (link) del modello; nel nostro caso, è stato inserito sotto il link del telaio (`chassis_link`).

Analizziamo ora i tag utilizzati:

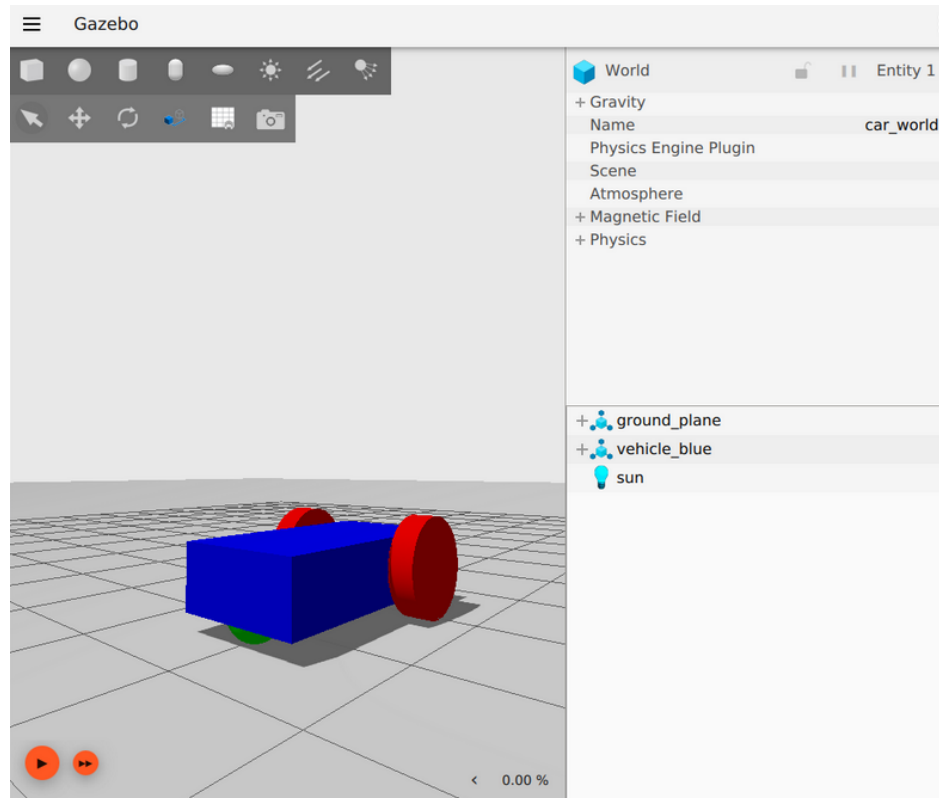


Figura 3.2: Robot utilizzato nella simulazione

- `<always_on>`: determina se il sensore è continuamente attivo o meno. Un valore di "1" assicura che il sensore aggiorni costantemente i suoi dati, il che è fondamentale per applicazioni che richiedono il monitoraggio in tempo reale. Tuttavia, questa scelta può influenzare il consumo di risorse in sistemi reali.
- `<update_rate>`: definisce la frequenza, in Hertz, con cui i dati del sensore vengono aggiornati. Un valore più alto fornisce dati più aggiornati ma richiede maggiori risorse di calcolo, influenzando le prestazioni in applicazioni critiche.
- `<visualize>`: questo tag, impostato su "true", abilita la visualizzazione del sensore nell'interfaccia grafica di Gazebo. È particolarmente utile durante la fase di sviluppo e test, permettendo agli utenti di vedere direttamente l'effetto delle loro configurazioni.
- `<topic>`: definisce il nome del topic su cui i dati del sensore sono pubblicati. Questa funzionalità è cruciale per l'integrazione del sensore con altri sistemi, permettendo una facile acquisizione e analisi dei dati.

È importante notare che non tutti i tag sono supportati per tutti i sensori.

Per leggere i dati dal sensore IMU si esegue il mondo virtuale in un terminale e si preme il pulsante di play:

```
gz sim sensor_tutorial.sdf
```

In un altro terminale, si esegue il seguente comando:

```
gz topic -e -t /imu
```

Quest'ultimo comando ascolta i messaggi inviati sul topic `/imu` (Figura 3.3). Nei dati IMU ricevuti, l'orientamento (`orientation`) viene rappresentato in quaternioni, con i componenti X, Y, Z (componenti vettoriali) che definiscono la direzione dell'asse di rotazione, mentre la componente W (valore scalare) rappresenta l'angolo di rotazione attorno a quell'asse. D'altro canto, la velocità angolare (`angular_velocity`) e l'accelerazione lineare (`linear_acceleration`) sono rappresentate lungo i tre assi cartesiani X, Y e Z. Qui, X, Y e Z indicano movimenti o rotazioni lungo ciascuno di questi assi cartesiani. Questa differenza tra l'utilizzo delle coordinate X, Y e Z riflette la natura unica dei quaternioni per descrivere l'orientamento in uno spazio tridimensionale, a differenza della rappresentazione più tradizionale delle accelerazioni e delle velocità angolari. Per osservare il cambiamento dei valori del sensore, si muove il robot in avanti usando il tasto freccia su della tastiera.

```
beniubunto@beniubunto-HP-Pavilion-Laptop-15-cw1xxx:~/SensorTutorial$ sudo gz topic -e -t /imu
[sudo] password di beniubunto:
header {
  stamp {
    nsec: 1000000
  }
  data {
    key: "frame_id"
    value: "vehicle_blue::chassis::imu_sensor"
  }
  data {
    key: "seq"
    value: "0"
  }
}
entity_name: "vehicle_blue::chassis::imu_sensor"
orientation {
  x: -2.2979951752539915e-13
  y: 3.1892792117841435e-06
  z: 7.3468558603757517e-07
  w: 0.99999999999946444
}
orientation_covariance {
  data: 0
  data: 0
  data: 0
  data: 0
  data: 0
  data: 0
  data: 0
  data: 0
}
angular_velocity {
  x: -4.5959903504506179e-10
  y: 0.0063785584234885761
  z: 0.0014693711720567887
}
angular_velocity_covariance {
  data: 0
  data: 0
  data: 0
  data: 0
  data: 0
  data: 0
  data: 0
  data: 0
}
linear_acceleration {
  x: 1.5228332687576427
  y: 0.73467884237701464
  z: 6.7069440501306943
}
```

Figura 3.3: Visualizzazione dei dati relativi ad orientamento, accelerazione lineare e velocità angolare del robot in movimento riportati sul topic `/imu`

Nel corso del tutorial un dettaglio interessante, che emerge durante l'analisi dei dati raccolti dal sensore IMU, è la presenza di valori di covarianza per l'accelerazione lineare, la velocità angolare e l'orientamento, come possiamo vedere nella Figura 3.3. La covarianza in questo contesto rappresenta una misura statistica che descriverebbe normalmente il grado di variazione congiunta tra due variabili casuali. Tuttavia, nei dati raccolti durante la simulazione, i valori di covarianza sono costantemente nulli e non mostrano alcuna variazione, indipendentemente dai movimenti del robot.

Questo fatto può essere spiegato considerando che, in molti sistemi di simulazione e in alcuni tipi di sensori IMU reali, la covarianza è spesso impostata su valori predefiniti o nulli, specialmente in scenari dove la precisione e la variabilità dei sensori non sono simulate con alta fedeltà. Nel caso del tutorial seguito, la costante assenza di variazione nei valori di covarianza suggerisce che questo parametro non sia stato configurato per essere simulato attivamente in Gazebo. Ciò è coerente con l'obiettivo primario della simulazione, che si concentra più sul funzionamento di base del sensore IMU che non su una simulazione dettagliata di tutte le sue potenziali caratteristiche statistiche.

È importante notare che, in applicazioni reali e in simulazioni più avanzate, la covarianza può svolgere un ruolo significativo, specialmente in algoritmi di fusione dei sensori e nella stima dello stato. Tuttavia, per i fini di questo esperimento specifico e del tutorial seguito, l'inclusione di valori di covarianza costantemente nulli può essere considerata un dettaglio secondario che non influisce sull'obiettivo principale dell'esercizio, che è quello di comprendere il funzionamento e l'uso del sensore IMU in Gazebo.

3.3 Esperimenti e risultati

In questa sezione ci concentriamo sull'osservazione dei cambiamenti nei valori forniti dal sensore IMU quando il robot è in movimento. Sebbene non siano stati condotti esperimenti formali, l'analisi dei dati IMU durante il movimento del robot fornisce intuizioni significative sul funzionamento e le prestazioni del sensore. In particolare, consideriamo i seguenti aspetti:

- *Osservazione del movimento del robot:* durante il tutorial, abbiamo potuto osservare la variazione dei valori dell'IMU in risposta al movimento del robot. Quando il robot rimane fermo, i valori relativi all'accelerazione lineare e alla velocità angolare mostrano una minima variazione, rispecchiando la mancanza di movimento. Al contrario, una volta che il robot inizia a muoversi, ad esempio quando viene usato il tasto freccia su della tastiera per farlo avanzare, i valori dell'IMU mostrano variazioni significative.
- *Analisi dei dati dell'IMU:* l'accelerazione lineare e la velocità angolare rispondono direttamente ai comandi di movimento, fornendo una conferma visiva che il sensore IMU sta operando come previsto. Ciò è evidenziato dai dati raccolti che mostrano un incremento dell'accelerazione lineare e modifiche nella velocità angolare quando il robot cambia posizione o orientamento.
- *Riflessioni sulle prestazioni del sensore IMU:* le osservazioni fatte suggeriscono che il sensore IMU funziona correttamente, rilevando in maniera affidabile i cambiamenti di movimento del robot. Questo corrobora l'importanza dell'IMU in applicazioni di robotica, dove il monitoraggio preciso del movimento e dell'orientamento è cruciale per il controllo e la navigazione.

In conclusione, l'analisi dei cambiamenti nei dati dell'IMU durante il movimento del robot ha fornito una valida dimostrazione del corretto funzionamento del sensore. Questi risultati confermano l'efficacia dell'IMU nell'identificare e registrare dinamiche di movimento, evidenziando il suo valore in scenari di automazione e robotica.

3.4 Conclusioni

Concludendo il nostro studio sull'impiego del sensore IMU nel software di simulazione Gazebo, abbiamo ottenuto importanti intuizioni riguardo al suo ruolo nell'ingegneria dell'automazione e della robotica. Attraverso il tutorial esaminato è emersa l'importanza critica

dell'IMU nel fornire dati accurati sull'orientamento, l'accelerazione e la velocità angolare di un robot, una precisione indispensabile per lo sviluppo e la verifica di algoritmi di controllo e di navigazione in ambienti simulati.

La simulazione ha evidenziato come un sensore, che può sembrare semplice, giochi un ruolo fondamentale nel definire la capacità di un robot di interagire con e rispondere al suo ambiente. Questa consapevolezza è vitale per la progettazione di sistemi robotici che siano non solo efficaci ma anche reattivi alle condizioni dinamiche. Inoltre, l'efficacia dell'IMU dimostrata in Gazebo apre la strada a indagini più approfondite e a sviluppi futuri in ambito robotico.

In questo capitolo, abbiamo posto le basi per una migliore comprensione e utilizzo del sensore IMU in scenari di simulazione, sottolineando la sua rilevanza tecnica e la sua applicabilità pratica. Questa comprensione approfondita dell'IMU, e della sua implementazione in un ambiente di simulazione come Gazebo, è fondamentale per avanzare verso la realizzazione di sistemi robotici più sofisticati e affidabili, dimostrando l'importanza del sensore in questione non solo dal punto di vista tecnico, ma anche per le sue potenziali applicazioni nel mondo reale.

Esperienza relativa al sensore di contatto

Il seguente capitolo della tesi esplora in dettaglio l'impiego e l'integrazione del sensore di contatto in un ambiente di simulazione Gazebo, un tema centrale nell'ambito dell'ingegneria dell'automazione e della robotica. Questa sezione inizia con una panoramica sull'importanza dei sensori di contatto, evidenziando il loro ruolo cruciale sia in contesti teorici che pratici, e prosegue analizzando specificamente il loro utilizzo in Gazebo.

Vengono, poi, descritti in modo approfondito i passaggi per la creazione di un ostacolo virtuale in Gazebo, che serve come punto di interazione per il sensore di contatto. Ciò include dettagli sulla modellazione, dimensione, forma e colore dell'ostacolo, fornendo una comprensione chiara di come tali elementi vengano configurati in un ambiente di simulazione.

Successivamente, si esamina la configurazione tecnica del sensore di contatto, dettagliando il processo di aggiunta e di programmazione del sensore nel modello del muro. Viene anche esplorata l'implementazione del plugin TriggeredPublisher, un componente chiave per la reattività del sistema, evidenziandone l'importanza in scenari reali.

Il capitolo prosegue con una sezione dedicata agli esperimenti e ai risultati ottenuti, dove viene validata la funzionalità del sistema attraverso test pratici, monitorando la risposta del sensore di contatto e del robot all'interno della simulazione.

Per terminare, vengono espresse riflessioni personali sull'esperienza di lavoro con il sensore di contatto in Gazebo, insieme a considerazioni sulle potenziali applicazioni future di questa tecnologia nell'ingegneria dell'automazione. Questa sezione sottolinea come l'integrazione di sensori avanzati possa portare a sviluppi significativi in termini di sicurezza, efficienza e autonomia nel campo della robotica.

4.1 Introduzione al sensore di contatto

Il sensore di contatto rappresenta uno strumento fondamentale nell'ambito dell'ingegneria dell'automazione, giocando un ruolo critico in numerose applicazioni, dalla robotica industriale alla progettazione di sistemi autonomi. Tale sensore, come suggerisce il nome, rileva la presenza fisica o il contatto con un oggetto o una superficie. La loro semplicità concettuale nasconde un'ampia gamma di impieghi, permettendo ai sistemi automatizzati di interagire in modo più sicuro e intelligente con l'ambiente circostante.

4.1.1 Importanza del sensore di contatto nell'automazione e nella robotica

La semplicità concettuale del sensore di contatto nasconde un'ampia gamma di impieghi, permettendo ai sistemi automatizzati di interagire in modo più sicuro e intelligente con l'ambiente circostante.

Nella robotica industriale, ad esempio, i sensori di contatto sono spesso utilizzati in bracci robotici per assicurare che gli oggetti vengano afferrati con la giusta pressione, evitando così

di danneggiarli (Figura 4.1). Un esempio concreto è rappresentato dai sistemi di assemblaggio automatico, dove la precisione è fondamentale: i sensori di contatto aiutano i robot a determinare se un componente è stato posizionato correttamente o se è necessario un ulteriore aggiustamento.

Un altro esempio significativo è dato dai robot di esplorazione spaziale, come i rover marziani, dove i sensori di contatto sono impiegati per navigare in ambienti sconosciuti e potenzialmente pericolosi. Questi sensori aiutano i rover a evitare rocce grandi o altri ostacoli che potrebbero compromettere la loro missione.

Nel settore automobilistico, i sensori di contatto trovano impiego nei sistemi di parcheggio assistito. Questi sensori permettono alle auto di rilevare la presenza di ostacoli durante le manovre di parcheggio, riducendo il rischio di collisioni e facilitando il posizionamento del veicolo.

In tutti questi esempi, i sensori di contatto forniscono un feedback vitale che permette ai sistemi automatizzati di operare in modo più sicuro ed efficiente. La loro semplice funzione di rilevamento del contatto si traduce in un'enorme versatilità, rendendoli componenti indispensabili nell'automazione moderna e nella robotica avanzata.

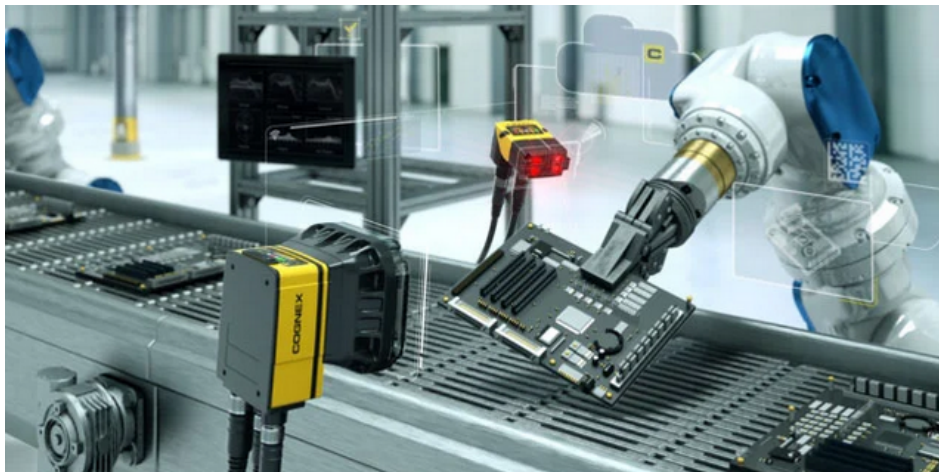


Figura 4.1: Robot industriale dotato di sensore di contatto impiegato in una catena di montaggio

4.1.2 Impiego del sensore di contatto in Gazebo

Nel contesto del presente case study, l'uso del sensore di contatto è stato esplorato attraverso la piattaforma di simulazione Gazebo. Nella simulazione effettuata, il sensore di contatto è stato implementato nel modello di un muro; tale applicazione del sensore permette di rilevare il contatto fisico tra il muro e qualsiasi oggetto in movimento, come il nostro robot a tre ruote. Quando il robot entra in contatto con il muro, il sensore rileva questo impatto, permettendo di monitorare le interazioni tra il robot e l'ambiente circostante. Questo esempio illustra come i sensori di contatto possano essere utilizzati non solo per proteggere i robot da eventuali collisioni, ma anche per studiare e migliorare la loro capacità di navigazione in ambienti complessi.

L'implementazione di questo sensore in un ambiente simulato come Gazebo offre, inoltre, l'opportunità di sperimentare con sistemi di controllo e algoritmi di reazione e di comprendere meglio come tali sensori possano essere impiegati in applicazioni reali. Ad esempio, in un contesto industriale, i sensori di contatto possono essere utilizzati per prevenire collisioni tra robot e oggetti, migliorando la sicurezza e l'efficienza operativa.

Nel nostro caso, il sensore di contatto è stato integrato in un robot a tre ruote, permettendo a quest'ultimo di percepire e reagire al contatto con un ostacolo predeterminato. Questo

semplice esperimento serve come modello di base per esplorare concetti più complessi di interazione fisica tra robot e ambiente, aprendo la strada a future innovazioni nell'ambito dell'ingegneria dell'automazione.

4.2 Creazione dell'ostacolo per la simulazione

Nella simulazione Gazebo, la creazione di un ostacolo rappresenta un passaggio fondamentale per testare e dimostrare l'efficacia dei sensori di contatto. Questo processo inizia con la definizione di un modello semplice, in questo caso un muro, che fungerà da ostacolo nel mondo virtuale. Di seguito vengono illustrate le righe di codice per la realizzazione dell'ostacolo, inserite poi nel mondo `sensor_tutorial.sdf`; successivamente, vediamo una descrizione dettagliata dei tag utilizzati.

```
<model name='wall'>
  <static>true</static>
  <pose>5 0 0 0 0 0</pose>
  <link name='box'>
    <visual name='visual'>
      <geometry>
        <box>
          <size>0.5 10.0 2.0</size>
        </box>
      </geometry>
      <material>
        <ambient>0.0 0.0 1.0 1</ambient>
        <diffuse>0.0 0.0 1.0 1</diffuse>
        <specular>0.0 0.0 1.0 1</specular>
      </material>
    </visual>
    <collision name='collision'>
      <geometry>
        <box>
          <size>0.5 10.0 2.0</size>
        </box>
      </geometry>
    </collision>
  </link>
</model>
```

Il primo passo è definire le specifiche di base del muro nel linguaggio XML utilizzato da Gazebo. Il muro viene modellato come un unico collegamento (`link`) con forma di scatola (`box`). Questo è realizzato mediante il tag `<model>` con il nome `'wall'`, seguito dalla specifica `<static>true</static>`, che indica che l'ostacolo è fisso e non soggetto a forze fisiche come la gravità o le collisioni.

Successivamente, viene stabilita la posizione dell'ostacolo nel mondo virtuale usando il tag `<pose>`, che, in questo caso, è impostato a `5 0 0 0 0 0`, posizionando, così, il muro a una distanza di 5 metri lungo l'asse X dalla posizione originale.

Il modello del muro è definito con dimensioni specifiche all'interno del tag `<box>`, dove il tag `<size>` è impostato su `0.5 10.0 2.0`. Tale assetto crea un muro di mezzo metro di spessore, dieci metri di lunghezza e due metri di altezza, dimensioni adeguate per un test efficace nel contesto della simulazione.

Per aggiungere un aspetto visuale al muro, viene utilizzato il tag `<visual>`, all'interno del quale vengono definiti la geometria e il materiale. La scelta di un colore blu vivido, realizzato attraverso la definizione dei valori di `<ambient>`, `<diffuse>`, e `<specular>` tutti impostati a `0.0 0.0 1.0 1`, rende l'ostacolo facilmente distinguibile nel contesto della simulazione.

Parallelamente alla componente visiva, è fondamentale definire la componente fisica del muro. Ciò viene realizzato tramite il tag `<collision>`, che utilizza la stessa geometria del tag `<visual>`, assicurando che la simulazione fisica del contatto tra il robot e il muro sia accurata e affidabile.

Con questi passaggi, il muro è pienamente configurato e pronto per essere inserito nel mondo di simulazione Gazebo. Una volta avviato il mondo, l'ostacolo appare nella simulazione (Figura 4.2), offrendo una piattaforma realistica per testare l'interazione del robot con sensori di contatto.

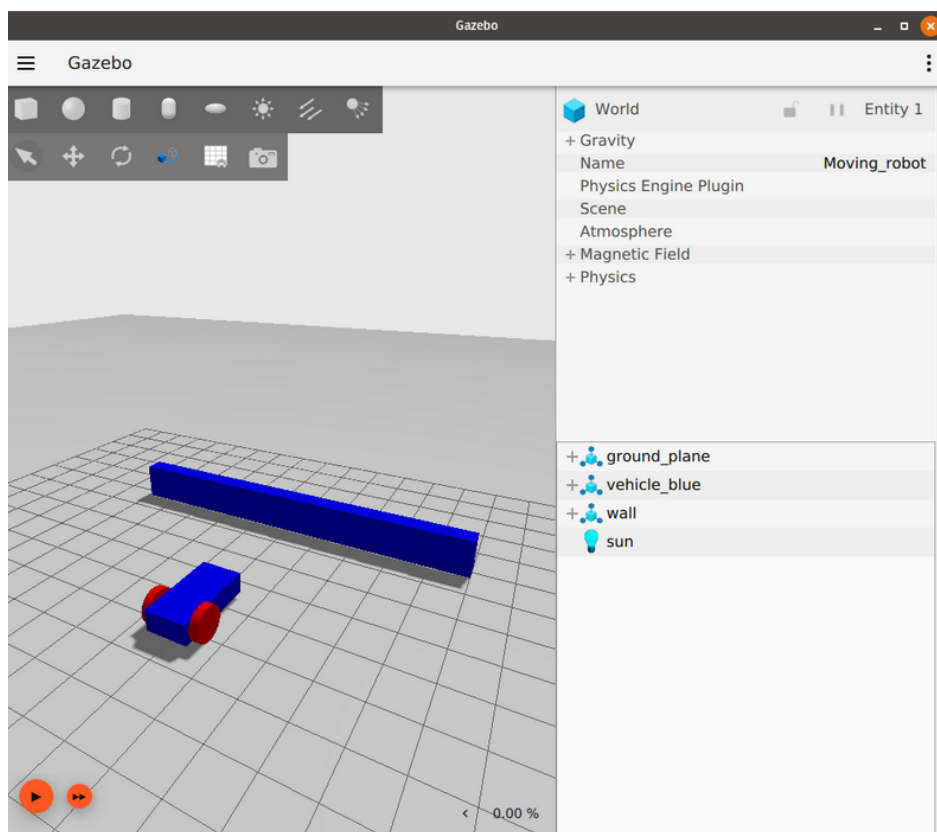


Figura 4.2: Rappresentazione del robot e dell'ostacolo

Questo processo di costruzione di un modello semplice, ma funzionale, in Gazebo, non solo fornisce una base per testare il sensore di contatto, ma serve anche come esempio pratico di come oggetti e ambienti possano essere creati e manipolati in simulazioni di robotica avanzata.

4.3 Configurazione del sensore di contatto

L'aggiunta del sensore di contatto al modello del muro nel simulatore Gazebo è un processo composto da diversi passaggi di configurazione; questi vengono elencati e spiegati di seguito.

Il primo passo consiste nell’inserire il codice per il sistema di contatto del sensore nel modello del muro. Ciò è realizzato attraverso l’aggiunta del seguente plugin:

```
<plugin filename="gz-sim-contact-system"
      name="gz::sim::systems::Contact">
</plugin>
```

Questo codice definisce un sistema di contatto che permette al sensore di rilevare le collisioni. Il plugin `gz-sim-contact-system` è responsabile della gestione delle interazioni fisiche del sensore nel mondo di simulazione.

Successivamente, il sensore di contatto viene effettivamente aggiunto al link `box` del modello del muro, come segue:

```
<sensor name='sensor_contact' type='contact'>
  <contact>
    <collision>collision</collision>
  </contact>
</sensor>
```

In queste righe di codice, il sensore è definito con un nome (`sensor_contact`) e un tipo (`contact`). All’interno dell’elemento `<contact>` si specifica il nome della collisione (in questo caso, `collision`) che si riferisce alla parte fisica del muro con cui il sensore interagirà.

Per ampliare la funzionalità del sensore di contatto, viene, poi, aggiunto il `TouchPlugin` al modello del muro. Quest’ultimo ha il compito di inviare un messaggio quando il muro viene toccato dal robot:

```
<plugin filename="gz-sim-touchplugin-system"
      name="gz::sim::systems::TouchPlugin">
  <target>vehicle_blue</target>
  <namespace>wall</namespace>
  <time>0.001</time>
  <enabled>>true</enabled>
</plugin>
```

Il `TouchPlugin` è configurato per reagire quando il `target` (in questo caso, il robot denominato `vehicle_blue`) entra in contatto con il muro. Il namespace `wall` viene utilizzato per specificare il topic su cui il messaggio verrà inviato. Quando il robot colpisce il muro, un messaggio viene pubblicato sul topic `/wall/touched`.

Per testare la configurazione, viene eseguito il comando `gz sim sensor_tutorial.sdf` in un terminale, per avviare la simulazione con il sensore configurato. In un altro terminale, si ascolta il topic `/wall/touched` usando il comando `gz topic -e -t /wall/touched`, dove:

- `-e` rappresenta un’opzione del comando `topic` che sta per `echo`; è utilizzata per stampare i messaggi ricevuti su un determinato topic.
- `-t` rappresenta il "topic target" e specifica il topic che il comando `gz topic -e` deve ascoltare.

Durante la simulazione, guidando il robot verso il muro, si può osservare che, al contatto, viene visualizzato un messaggio `data: true` sul terminale che ascolta il topic, indicando l’attivazione del sensore di contatto. Possiamo osservare il tutto nella Figura 4.3.

```
beniubunto@beniubunto-HP-Pavilion-Laptop-15-cw1xxx:~/SensorTutorial$ sudo gz top
ic -e -t /wall/touched
[sudo] password di beniubunto:
data: true
█
```

Figura 4.3: Messaggio `data: true` sul topic `/wall/touched` quando il robot tocca il muro

Questo processo dimostra non solo come configurare un sensore di contatto in Gazebo, ma anche come esso possa essere utilizzato per monitorare le interazioni fisiche in un ambiente di simulazione, fornendo dati cruciali per la progettazione e il test di comportamenti robotici.

4.4 Implementazione del TriggeredPublisher plugin

La fase finale della configurazione del nostro scenario di simulazione in Gazebo include l'implementazione del plugin `TriggeredPublisher`, un componente fondamentale per controllare il comportamento del robot in risposta agli input del sensore di contatto. Questo plugin è cruciale per dimostrare un'applicazione pratica dei sensori di contatto in termini di sicurezza e automazione.

La funzione principale del `TriggeredPublisher` è quella di fermare il robot quando entra in contatto con l'ostacolo (il muro). Ciò viene eseguito mediante la seguente configurazione del plugin:

```
<plugin filename="gz-sim-triggered-publisher-system"
  name="gz::sim::systems::TriggeredPublisher">
  <input type="gz.msgs.Boolean" topic="/wall/touched">
    <match>data: true</match>
  </input>
  <output type="gz.msgs.Twist" topic="/cmd_vel">
    linear: {x: 0.0}, angular: {z: 0.0}
  </output>
</plugin>
```

In questa configurazione, il plugin ascolta il topic `/wall/touched`; quando riceve un messaggio con `data: true` (che indica che il muro è stato toccato dal robot), genera un comando di output che ferma il robot, inviando un messaggio sul topic `/cmd_vel` con valori zero sia per il movimento lineare (`linear: x: 0.0`) che per quello angolare (`angular: z: 0.0`).

Il topic `/cmd_vel` in Gazebo è un elemento chiave per la comunicazione e il controllo dei movimenti dei robot all'interno della simulazione. Il topic in questione è utilizzato per inviare comandi di movimento al modello del robot, permettendo di controllare la sua velocità lineare e angolare. Nel contesto del tutorial e della simulazione, `/cmd_vel` funge da canale

attraverso il quale vengono trasmessi i messaggi che dirigono il comportamento dinamico del robot. Per esempio, per far muovere il robot nella simulazione, evitando quindi l'utilizzo della tastiera, si utilizza il seguente comando in un terminale:

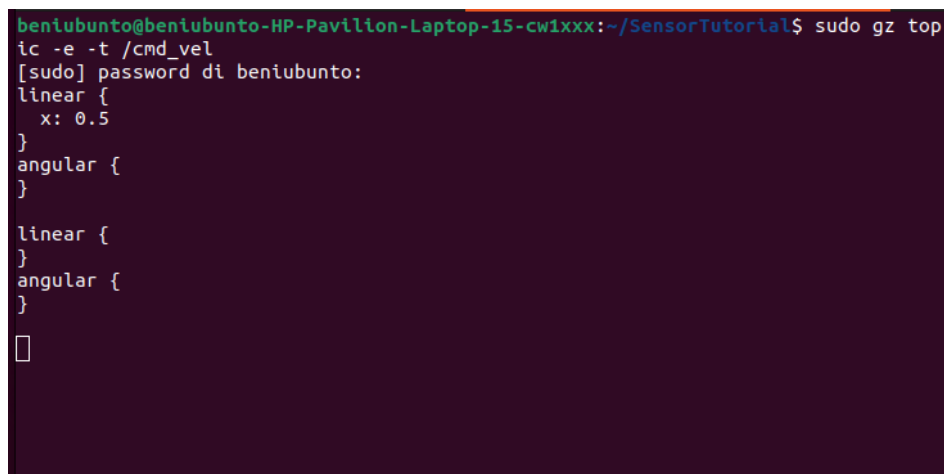
```
gz topic -t "/cmd_vel" -m gz.msgs.Twist -p "linear: {x: 0.5},  
angular: {z: 0.05}"
```

Questo comando specifica che si desidera pubblicare un messaggio sul topic `/cmd_vel`. Il tipo di messaggio, specificato dopo l'opzione `-m`, è `gz.msgs.Twist`, che include componenti sia lineari che angolari. L'opzione `-p` permette di definire il contenuto effettivo del messaggio: in questo caso, una velocità lineare (`linear`) di 0.5 lungo l'asse `x` e una velocità angolare (`angular`) di 0.05 attorno all'asse `z`.

A questo punto, configurato il `TriggeredPublisher` plugin e compreso l'impiego del topic `/cmd_vel`, possiamo avviare il mondo tramite il solito comando `gz sim sensor_tutorial.sdf` in un terminale, mentre, in un altro terminale, tramite il comando `gz topic -e -t /cmd_vel`, possiamo visualizzare il messaggio che verrà inviato al robot una volta che esso sarà entrato in contatto con il muro.

Nella Figura 4.4 vediamo comparire due messaggi sul terminale:

- `linear {x: 0.5} angular { }`: messaggio relativo al comando "freccia su" da tastiera, utilizzato per far sì che il robot, una volta avviata la simulazione, si diriga verso l'ostacolo. L'assenza di un valore all'interno delle parentesi graffe indica che tale valore è nullo.
- `linear { } angular { }`: messaggio inviato al robot quando quest'ultimo tocca il muro; i valori all'interno delle parentesi graffe sono entrambi nulli, ciò indica al robot di fermarsi.



```
beniubunto@beniubunto-HP-Pavilion-Laptop-15-cw1xxx:~/SensorTutorial$ sudo gz top  
ic -e -t /cmd_vel  
[sudo] password di beniubunto:  
linear {  
  x: 0.5  
}  
angular {  
}  
  
linear {  
}  
angular {  
}  
□
```

Figura 4.4: Visualizzazione dei messaggi dal topic `/cmd_vel`

4.4.1 Importanza del TriggeredPublisher plugin in contesti reali

L'uso del `TriggeredPublisher` plugin evidenzia come i sensori di contatto, combinati con sistemi di controllo intelligenti, possano migliorare notevolmente la sicurezza nell'ambiente di automazione. Nel contesto reale, questo tipo di configurazione può essere utilizzato per prevenire danni ai robot o ad altri oggetti nel loro ambiente operativo. Per esempio, in un

contesto industriale, un robot che si ferma automaticamente al contatto con un ostacolo può evitare incidenti o danni a oggetti sensibili.

Inoltre, l'implementazione del `TriggeredPublisher` in questo scenario di simulazione Gazebo è un esempio eccellente di come i principi di automazione possano essere applicati per realizzare sistemi robotici più autonomi e responsivi. Ciò non solo aumenta l'efficienza operativa, ma apre anche la strada a nuove possibilità di interazione tra robot e ambiente, in cui i robot sono in grado di prendere decisioni in tempo reale basate sulle informazioni sensoriali ricevute.

In conclusione, l'aggiunta del `TriggeredPublisher` al nostro modello di simulazione Gazebo non solo dimostra un'interazione efficace tra sensori e sistemi di controllo, ma fornisce anche una base solida per ulteriori esplorazioni nel campo dell'automazione robotica, con un particolare focus sulla sicurezza e sulla reattività dei robot in ambienti complessi.

4.5 Esperimenti e risultati

La fase sperimentale della simulazione in Gazebo ha avuto come obiettivo primario la verifica del corretto funzionamento del sensore di contatto e dei plugin correlati. L'esperimento è stato strutturato per testare la reattività del sistema alla rilevazione di un contatto e la conseguente trasmissione dei messaggi di controllo. L'attenzione si è focalizzata, in particolare, sui messaggi generati sul topic `/wall/touched` e sulle modifiche di stato trasmesse attraverso il topic `/cmd_vel`.

Dopo aver configurato il sensore di contatto sul modello del muro e dopo aver implementato i plugin necessari, abbiamo proceduto con l'avvio della simulazione. Il robot è stato guidato manualmente verso il muro, con conseguente attivazione del sensore di contatto al momento dell'impatto. Il risultato atteso era la generazione di un messaggio `data: true` sul topic `/wall/touched`, un segnale chiaro che il sensore di contatto aveva correttamente percepito l'interazione fisica con l'ostacolo.

Il verificarsi di questo evento è stato un indicatore chiave del successo dell'esperimento. La visualizzazione del messaggio in questione ha confermato che il sistema di sensori e il processo di comunicazione interna operano come previsto. Ciò ha dimostrato che il modello del muro era adeguatamente configurato e che il sensore di contatto era in grado di rilevare efficacemente l'interazione con il robot.

Parallelamente, abbiamo monitorato il topic `/cmd_vel` per osservare le modifiche nei comandi di movimento inviati ad esso in seguito al contatto. Come previsto, al momento dell'impatto il sistema ha generato i comandi per ridurre a zero sia l'accelerazione lineare ($x: 0.0$) che la velocità angolare ($z: 0.0$). Tale osservazione ha confermato il funzionamento corretto del plugin `TriggeredPublisher`, dimostrando la sua capacità di rispondere a eventi sensoriali con azioni appropriate. Il robot, ricevendo questi comandi, ha immediatamente arrestato il suo movimento, un risultato fondamentale per la validazione della sicurezza e dell'efficacia del sistema di controllo automatizzato.

In conclusione, gli esperimenti condotti hanno fornito risultati positivi, dimostrando che la configurazione del sensore di contatto, insieme ai plugin utilizzati, funzionava come previsto. Questi test hanno validato non solo la configurazione tecnica, ma anche l'efficacia del sensore di contatto come strumento di sicurezza nell'ambito della robotica automatizzata. Le implicazioni di questi risultati sono significative, poiché indicano la fattibilità di implementare sistemi di controllo reattivi e automatizzati in scenari più complessi e realistici.

4.6 Conclusioni

L'esperienza lavorativa con il sensore di contatto in Gazebo si è rivelata estremamente formativa, offrendo intuizioni preziose riguardo le sfide e le opportunità nell'ambito dell'ingegneria dell'automazione. Lavorare con questa tecnologia in un ambiente di simulazione ha fornito una comprensione più profonda di come i sensori e i sistemi di controllo possano essere integrati per creare soluzioni robotiche più sicure e intelligenti.

Uno degli aspetti più significativi che sono stati osservati è la capacità dei sensori di contatto di aggiungere un livello di percezione critico ai robot, dando loro l'opportunità di interagire con l'ambiente circostante in modo più consapevole e reattivo. Ciò non solo aumenta la sicurezza operativa, evitando potenziali collisioni o danni, ma apre anche la strada a nuove possibilità nell'automazione, come l'esplorazione autonoma e la gestione di scenari imprevisti.

Le potenziali applicazioni di questa tecnologia nell'ingegneria dell'automazione sono vaste e variegata. Un chiaro esempio è quello del settore manifatturiero, dove i sensori di contatto potrebbero migliorare l'efficienza e la precisione dei robot industriali, permettendo loro di adattarsi dinamicamente a cambiamenti nell'ambiente di lavoro. Nell'automazione domestica, questi sensori potrebbero rendere i robot domestici più sicuri e funzionali, evitando ostacoli e interagendo più efficacemente con gli utenti e gli oggetti domestici.

In conclusione, l'esperienza con il sensore di contatto in Gazebo ha non solo rafforzato la nostra comprensione tecnica, ma ha anche stimolato la nostra consapevolezza su come queste tecnologie possano essere applicate per risolvere problemi reali nell'ingegneria dell'automazione. È evidente che l'integrazione di sensori avanzati, come il sensore di contatto, in sistemi robotici rappresenta un passo avanti fondamentale verso la realizzazione di un futuro in cui i robot siamo in grado di operare con maggiore intelligenza, sicurezza e autonomia.

Esperienza relativa al sensore LIDAR

Il Capitolo 5 della tesi esplora l'implementazione e l'uso del sensore LIDAR (Light Detection and Ranging) in un ambiente simulato di Gazebo, evidenziando la sua rilevanza nell'ingegneria dell'automazione. Il sensore LIDAR, noto per la sua precisione nella rilevazione di ostacoli e distanze, è stato integrato in un modello di robot a tre ruote in Gazebo, fornendo uno scenario dettagliato e realistico per testarne le capacità.

Inizialmente, il capitolo presenta un'introduzione al sensore LIDAR, enfatizzando la sua importanza e le sue applicazioni nell'automazione, nonché la motivazione dietro la scelta di questo sensore rispetto ad altri. Segue una descrizione approfondita della configurazione del sensore nel modello di robot, con particolare attenzione ai dettagli tecnici del processo, come il frame di posizionamento e le specifiche tecniche del sensore stesso.

Un'analisi dettagliata dei parametri di configurazione del LIDAR, come il rate di aggiornamento, l'angolo di raggio, il numero di campioni, e i range di distanza, mette in luce la loro importanza nel contesto specifico dell'applicazione. Questa sezione è essenziale per comprendere come la calibrazione accurata di questi parametri influenzi la performance del sensore nel rilevare e navigare tra gli ostacoli.

Il capitolo prosegue esaminando l'implementazione del codice in C++ per l'elaborazione dei dati del sensore LIDAR, con un focus sulle funzioni chiave, come la callback e il metodo di pubblicazione dei comandi di movimento. Tale sezione fornisce una comprensione tecnica della logica di programmazione dietro il controllo del movimento del robot.

L'ottimizzazione del processo di avvio in Gazebo è stata un altro punto focale, con la creazione di un file di lancio unificato per semplificare l'esecuzione simultanea del mondo sensoriale e del nodo LIDAR.

Gli esperimenti condotti e i risultati ottenuti hanno fornito una panoramica efficace di come il robot, dotato di sensore LIDAR, evita gli ostacoli e interagisce con l'ambiente circostante. La parte in questione è cruciale per dimostrare l'efficacia del sensore nella navigazione autonoma.

In conclusione, il capitolo riflette sull'integrazione di questa esperienza con il resto della tesi e sulle potenziali applicazioni future del sensore LIDAR, sottolineando come le conoscenze acquisite possano contribuire allo sviluppo di sistemi robotici più avanzati e autonomi. Questa analisi completa fornisce un quadro chiaro e approfondito dell'uso del sensore LIDAR in un contesto simulato, aprendo la strada a future innovazioni nel campo dell'ingegneria dell'automazione.

5.1 Introduzione al sensore LIDAR

Il sensore LIDAR, acronimo di "Light Detection and Ranging" (rilevamento e misurazione della luce), si impone come una delle innovazioni più significative nel panorama dell'ingegneria dell'automazione. Questa tecnologia, sfruttando avanzate tecniche di misurazione basate sulla luce, è capace di determinare con precisione la distanza tra il sensore stesso e vari oggetti presenti nell'ambiente.

Il principio di funzionamento del sensore in questione si basa sull'emissione di impulsi laser che, riflettendosi sugli oggetti e tornando indietro, permettono di calcolare la distanza in

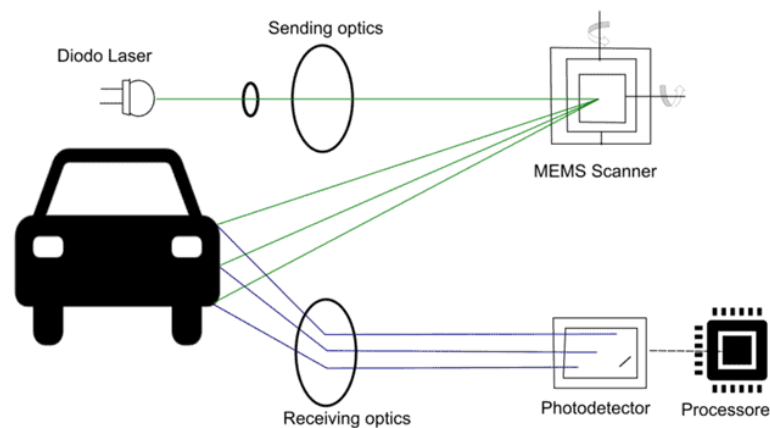


Figura 5.1: Raffigurazione grafica della scansione meccanica LIDAR

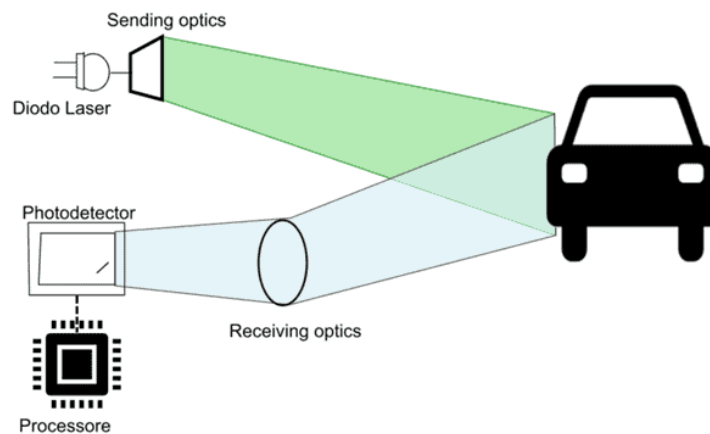


Figura 5.2: Raffigurazione grafica della tecnologia Flash LIDAR

base al tempo impiegato dal segnale per compiere il tragitto di andata e ritorno. I sistemi LIDAR sono costituiti da componenti piuttosto sofisticati e, a seconda del loro tipo, contengono: sorgenti laser e diodi laser, elementi ottici (lenti, specchi e diffusori), elementi per orientare il fascio emesso nello spazio, fotorivelatori e unità di elaborazione del segnale. Tali sistemi sono, generalmente, realizzati secondo due principali metodologie, ovvero quella a scansione meccanica (scanner) e quella a immagine fissa (flash). Più specificatamente:

- *La scansione meccanica:* consiste nella rotazione fisica del laser e del ricevitore a 360°, per ottenere una visione molto ampia. Essa può avvenire ruotando o muovendo fisicamente sugli assi cartesiani l'emettitore e il rilevatore di luce, oppure utilizzando dei microspecchi comandabili realizzati in tecnologia MEMS (Micro Electro-Mechanical Systems), come mostrato in Figura 5.1. Il campo visivo di un sensore LiDAR a scansione viene determinato dai gradi di libertà e dagli eventuali vincoli di movimento del sistema meccanico o MEMS di scansione.
- *Il Flash LIDAR:* non prevede alcuna parte o componente in movimento e assomiglia di più, come concetto, ad una fotografia istantanea. Il campo visivo del sensore LIDAR Flash viene determinato sostanzialmente dalle caratteristiche dell'ottica che emette e riceve il fascio di luce (ciò è visibile nella Figura 5.2).

La lunghezza d'onda del raggio laser emesso e la potenza con cui avviene l'emissione, insieme alla sensibilità del rilevatore di luce, sono alcuni dei fattori che determinano la portata visiva del LiDAR e la sua capacità di "individuare" i dettagli.

Le precedenti metodologie consentono di generare mappe dettagliate e tridimensionali dell'area circostante, rendendo il LIDAR estremamente efficace nella ricostruzione di ambienti complessi. La tecnologia del sensore in questione si distingue per la sua elevata risoluzione e precisione, capaci di fornire dati accurati anche in condizioni di visibilità limitata, come in ambienti polverosi o in condizioni di scarsa illuminazione. Inoltre, la capacità di effettuare misurazioni rapide e in tempo reale rende il LIDAR particolarmente adatto a scenari dinamici e in rapido cambiamento.

5.1.1 Importanza del sensore LIDAR nell'automazione e nella robotica

Nell'ambito dell'ingegneria dell'automazione, il LIDAR gioca un ruolo cruciale nella realizzazione di sistemi robotici avanzati. La sua capacità di rilevare ostacoli e determinare distanze con estrema precisione permette ai robot di navigare in modo sicuro ed efficiente in ambienti complessi. Questo aspetto è fondamentale, poichè garantisce la sicurezza operativa di tali sistemi e perchè ottimizza le loro prestazioni in compiti specifici, come il trasporto di materiali, la sorveglianza o l'esplorazione di aree inaccessibili.

L'utilizzo del sensore LIDAR, quindi, non si limita a una mera funzione di rilevamento; esso abilita i robot a comprendere e interagire con l'ambiente in modi sempre più sofisticati. Attraverso l'analisi dei dati raccolti dal LIDAR, è possibile implementare algoritmi complessi per il controllo del movimento, la pianificazione del percorso e la risposta a stimoli ambientali, trasformando i robot da semplici esecutori di compiti predefiniti a entità capaci di prendere decisioni autonome basate su informazioni real-time.

In questo contesto, il nostro studio sull'integrazione del sensore LIDAR in un modello robotico su Gazebo assume un'importanza significativa. Esso non solo dimostra l'applicabilità pratica di questa tecnologia in un ambiente controllato, ma fornisce anche intuizioni preziose su come ottimizzare l'uso del LIDAR in scenari di automazione più ampi e complessi.

5.2 Configurazione del sensore LIDAR in Gazebo

Nel nostro studio, l'impiego del sensore LIDAR in Gazebo è stato fondamentale per evitare che il robot entrasse in contatto con le pareti, un evento, realisticamente parlando, potenzialmente dannoso. Questo approccio ha rappresentato un'alternativa significativa all'uso del sensore di contatto, fornendo una soluzione più sofisticata e proattiva per la rilevazione di ostacoli.

La configurazione del LIDAR è iniziata con la creazione di un frame dedicato, denominato "lidar_frame", fissato direttamente al telaio del robot. Il frame è stato inserito all'interno del tag `<model>` del veicolo, specificatamente sotto `vehicle_blue`, per garantire il corretto allineamento e la stabilità del sensore. La sua posizione, definita nel tag `<pose>`, è stata impostata per ottimizzare la portata e l'angolo di rilevazione del sensore.

```
<frame name="lidar_frame" attached_to='chassis'>
  <pose>0.8 0 0.5 0 0 0</pose>
</frame>
```

Proseguendo con l'implementazione, è stato integrato un plugin essenziale sotto il tag `<world>`, necessario per attivare e gestire il sensore LIDAR all'interno della simulazione. Il plugin in questione, identificato come `gz-sim-sensors-system`, ha incorporato il motore

di rendering `ogre2`, permettendo, così, una rappresentazione visuale del sensore all'interno dell'ambiente simulato.

```
<plugin
  filename="gz-sim-sensors-system"
  name="gz::sim::systems::Sensors">
  <render_engine>ogre2</render_engine>
</plugin>
```

La configurazione del sensore LIDAR, sotto il collegamento del telaio (`chassis`), è stata realizzata attraverso un preciso blocco di codice XML. Questo include la definizione del nome e del tipo del sensore (`gpu_lidar`), la sua posizione relativa al frame del LIDAR, il topic per la pubblicazione dei dati, e il tasso di aggiornamento del sensore.

```
<sensor name='gpu_lidar' type='gpu_lidar'>"
  <pose relative_to='lidar_frame'>0 0 0 0 0 0</pose>
  <topic>lidar</topic>
  <update_rate>10</update_rate>
  <ray>
    <scan>
      <horizontal>
        <samples>640</samples>
        <resolution>1</resolution>
        <min_angle>-1.396263</min_angle>
        <max_angle>1.396263</max_angle>
      </horizontal>
      <vertical>
        <samples>1</samples>
        <resolution>0.01</resolution>
        <min_angle>0</min_angle>
        <max_angle>0</max_angle>
      </vertical>
    </scan>
    <range>
      <min>0.08</min>
      <max>10.0</max>
      <resolution>0.01</resolution>
    </range>
  </ray>
  <always_on>1</always_on>
  <visualize>>true</visualize>
</sensor>
```

Una volta completata la configurazione, la simulazione è stata avviata attraverso il comando `gz sim sensor_tutorial.sdf`. L'attivazione del sensore e il suo funzionamento sono stati monitorati attraverso il messaggio pubblicato sul topic `/lidar`, che fornisce dati dettagliati sulle distanze rilevate dal sensore LIDAR. Il "LIDAR message" comprende i seguenti attributi:

```
message LaserScan
{
```

```

Header header                = 1;

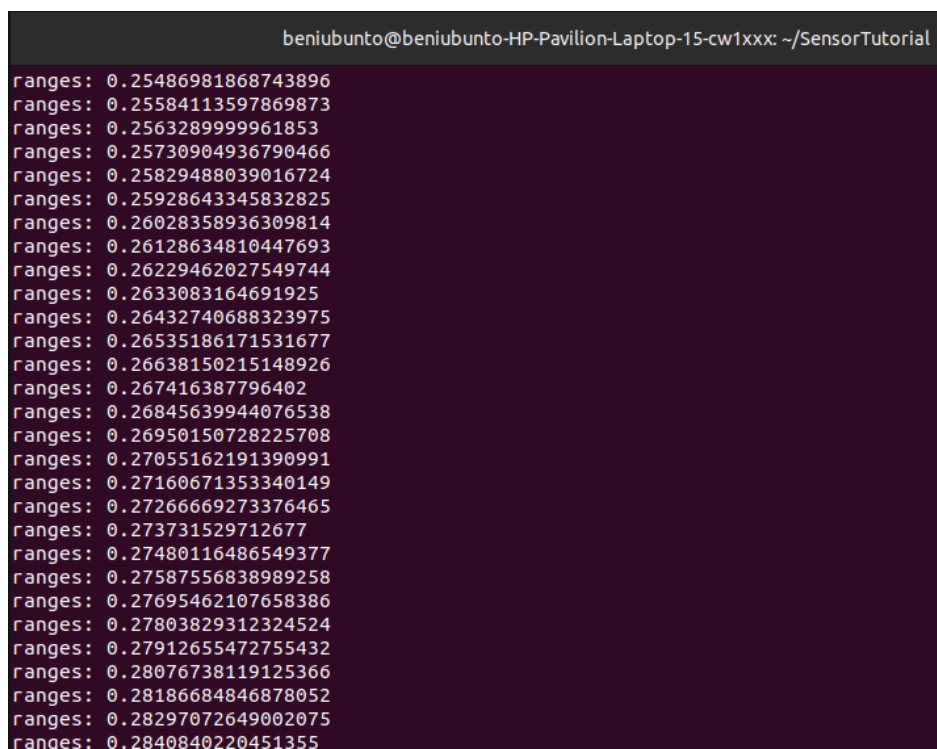
string frame                  = 2;
Pose world_pose              = 3;
double angle_min             = 4;
double angle_max             = 5;
double angle_step            = 6;
double range_min             = 7;
double range_max             = 8;
uint32 count                 = 9;
double vertical_angle_min    = 10;
double vertical_angle_max    = 11;
double vertical_angle_step   = 12;
uint32 vertical_count        = 13;

repeated double ranges       = 14;
repeated double intensities  = 15;
}

```

Questi dati comprendono una serie di attributi cruciali, come l'angolo minimo e massimo di scansione, il conteggio dei raggi, e l'intensità dei segnali rilevati.

L'output del messaggio del LIDAR, specificatamente i dati di range (Figura 5.3), è stato visualizzato utilizzando il comando `gz topic -e -t /lidar`, fornendo informazioni vitali per la navigazione autonoma e per evitare ostacoli da parte del robot.



```

beniubunto@beniubunto-HP-Pavilion-Laptop-15-cw1xxx: ~/SensorTutorial
ranges: 0.25486981868743896
ranges: 0.25584113597869873
ranges: 0.2563289999961853
ranges: 0.25730904936790466
ranges: 0.25829488039016724
ranges: 0.25928643345832825
ranges: 0.26028358936309814
ranges: 0.26128634810447693
ranges: 0.26229462027549744
ranges: 0.2633083164691925
ranges: 0.26432740688323975
ranges: 0.26535186171531677
ranges: 0.26638150215148926
ranges: 0.267416387796402
ranges: 0.26845639944076538
ranges: 0.26950150728225708
ranges: 0.27055162191390991
ranges: 0.27160671353340149
ranges: 0.27266669273376465
ranges: 0.273731529712677
ranges: 0.27480116486549377
ranges: 0.27587556838989258
ranges: 0.27695462107658386
ranges: 0.27803829312324524
ranges: 0.27912655472755432
ranges: 0.28076738119125366
ranges: 0.28186684846878052
ranges: 0.28297072649002075
ranges: 0.2840840220451355

```

Figura 5.3: Parte dei dati di range visualizzati sul topic `/lidar`

In questa fase, il sensore LIDAR ha dimostrato la sua efficacia nella rilevazione di ostacoli e nella fornitura di dati essenziali per la navigazione sicura e indipendente del robot, evidenziando il suo ruolo cruciale nella robotica avanzata e nell'ingegneria dell'automazione.

5.3 Analisi della configurazione del sensore LIDAR

In questa sezione, ci concentreremo sull'analisi dei parametri chiave del sensore LIDAR configurato nel nostro modello robotico su Gazebo. La comprensione di questi parametri è cruciale per ottimizzare l'uso del sensore nelle applicazioni specifiche.

5.3.1 Parametri principali del sensore LIDAR

I parametri in questione che analizzeremo sono i seguenti:

- *Rate di aggiornamento* (<update_rate>): il rate di aggiornamento, espresso in hertz (Hz), determina la frequenza con cui il sensore LIDAR raccoglie i dati. Nel nostro caso, il sensore è stato impostato con un rate di aggiornamento di 10 Hz, ovvero raccoglie i dati ogni 0.1 secondi. Tale frequenza influisce direttamente sulla fluidità e sulla tempestività della percezione ambientale del robot, e ciò la rende un fattore cruciale in ambienti dinamici dove le condizioni cambiano rapidamente.
- *Angolo di raggio* (<min_angle>|, <max_angle>): gli angoli minimi e massimi, definiti in radianti, determinano la portata orizzontale e verticale del raggio LIDAR. Nel nostro modello, gli angoli orizzontali vanno da -1.396263 a 1.396263 radianti, permettendo una copertura quasi completa davanti al robot. Questa ampiezza dell'angolo di raggio è fondamentale per garantire che il robot possa rilevare ostacoli da diverse direzioni e angolazioni.
- *Numero di campioni* (<samples>): tale parametro indica il numero di raggi laser simulati che il LIDAR emette in un ciclo completo di scansione. Un numero maggiore di campioni aumenta la risoluzione dei dati rilevati, permettendo una percezione più dettagliata dell'ambiente. Nel nostro caso, abbiamo utilizzato 640 campioni per la scansione orizzontale, così da garantire una rappresentazione dettagliata degli ostacoli circostanti.
- *Range minimo e massimo* (<range>, <min>, <max>): questi parametri definiscono la distanza minima e massima a cui il LIDAR può rilevare oggetti. Nel nostro modello, il range va da 0.08 a 10.0 metri, un range particolarmente importante per adattare il sensore alle dimensioni e alle caratteristiche specifiche dell'ambiente di navigazione del robot.

5.3.2 Importanza dei parametri nel contesto specifico

La configurazione dei parametri del sensore LIDAR deve essere attentamente calibrata in base alle esigenze specifiche dell'applicazione. Ad esempio, in un ambiente dove gli ostacoli possono apparire e cambiare rapidamente, un alto rate di aggiornamento è essenziale per una reattività adeguata. Allo stesso modo, in ambienti con spazi ristretti o con ostacoli posizionati ad altezze diverse, l'angolo di raggio e il numero di campioni acquisiscono una notevole importanza per garantire una rilevazione accurata.

Nel contesto della nostra simulazione, il rate di aggiornamento selezionato garantisce che il robot riceva dati in tempo reale per identificare efficacemente la presenza di potenziali ostacoli, come il muro presente nell'ambiente. Inoltre, la scelta di un angolo di raggio ampio permette al sensore di catturare un'area più estesa, aumentando, così, le possibilità di rilevare ostacoli da diverse angolazioni.

Il numero elevato di campioni scelto garantisce una ricostruzione dettagliata dell'ambiente circostante. Questo dettaglio è fondamentale per il riconoscimento preciso di oggetti di

varie forme e dimensioni, permettendo al robot di navigare con maggiore sicurezza. Infine, il range minimo e massimo di rilevamento sono stati impostati per adattarsi alle dimensioni dell'ambiente simulato, assicurando che il robot possa percepire ostacoli sia molto vicini sia ad una distanza ragionevole.

In conclusione, la configurazione ottimale dei parametri del sensore LIDAR è dettata da un equilibrio delicato che deve considerare le specifiche dell'ambiente di applicazione e le esigenze del sistema robotico. La nostra analisi in Gazebo ha dimostrato come una calibrazione accurata di questi parametri possa migliorare significativamente le capacità di navigazione e di interazione del robot con il suo ambiente, un aspetto chiave per il successo in applicazioni reali di ingegneria dell'automazione.

5.4 Implementazione del codice per l'elaborazione dei dati del LIDAR

In questa sezione viene esaminato il processo di implementazione del codice utilizzato per l'elaborazione dei dati raccolti dal sensore LIDAR, analizzando in particolare sul nodo C++ che funge da fulcro per il controllo del movimento del robot.

- *Descrizione del nodo in C++.* Nella nostra implementazione, utilizziamo un nodo C++ per interpretare i dati del sensore LIDAR e per controllare il movimento del robot. Il nodo in questione, chiamato `lidar_node`, è essenziale nel processo di elaborazione dei dati e agisce come un intermediario tra il sensore LIDAR e i sistemi di movimento del robot.

Il nodo si iscrive al topic `/lidar`, che è la fonte dei dati del sensore LIDAR, e pubblica comandi di movimento al topic `/cmd_vel`. Per fare ciò, abbiamo dichiarato un nodo `gz::transport::Node` e abbiamo definito il tipo di messaggio `gz::msgs::Twist`, che rappresenta i comandi di velocità lineare e angolare per il robot.

```
gz::transport::Node node;
std::string topic_pub = "/cmd_vel";
gz::msgs::Twist data;
auto pub = node.Advertise<gz::msgs::Twist>(topic_pub);
```

- *Analisi delle funzioni chiave*
 - *Funzione di callback* `cb`. La funzione di callback `cb` rappresenta un elemento cardine nella gestione dei dati provenienti dal sensore LIDAR. Il suo compito è quello di interpretare queste informazioni per influenzare in modo intelligente il comportamento del robot nell'ambiente circostante.

```
void cb(const gz::msgs::LaserScan &_msg)
{
    bool allMore = true;
    for (int i = 0; i < _msg.ranges_size(); i++)
    {
        if (_msg.ranges(i) < 1.0)
        {
            allMore = false;
            break;
        }
    }
}
```

```

    }
    if (allMore) //if all bigger than one
    {
        data.mutable_linear()->set_x(0.5);
        data.mutable_angular()->set_z(0.0);
    }
    else
    {
        data.mutable_linear()->set_x(0.0);
        data.mutable_angular()->set_z(0.5);
    }
    pub.Publish(data);
}

```

All'interno della funzione `cb`, la prima operazione è l'analisi della dimensione dell'array dei raggi, ottenuta tramite `_msg.ranges_size()`. Questo step è di vitale importanza poiché ci fornisce il numero di misurazioni di distanza che il sensore LIDAR ha raccolto. Ogni elemento nell'array corrisponde a una misurazione distinta, rappresentando la distanza tra il robot e gli oggetti circostanti in una specifica direzione.

Successivamente, la funzione entra in un ciclo `for` che scorre attraverso ciascuna di queste misurazioni. Per ogni raggio, eseguiamo un controllo critico: `if (_msg.ranges(i) < 1.0)`. Questa condizione verifica se la distanza rilevata in qualsiasi direzione è inferiore a 1.0 metro. Se questa condizione è soddisfatta per qualsiasi raggio, vuol dire che il robot è troppo vicino a un ostacolo in quella specifica direzione. In tal caso, impostiamo la variabile `allMore` su `false` e interrompiamo il ciclo. Questa logica è fondamentale per evitare collisioni, poiché permette al robot di reagire tempestivamente alla presenza di ostacoli nelle sue immediate vicinanze.

Una volta completata l'iterazione su tutti i raggi, la funzione procede alla decisione sul movimento del robot. Se `allMore` rimane `true`, vuol dire che tutti i raggi sono maggiori di 1 metro, indicando che non ci sono ostacoli nelle immediate vicinanze. In questo caso, il robot riceve il comando di proseguire in avanti, implementato attraverso `data.mutable_linear()->set_x(0.5);`, che imposta una velocità lineare positiva.

D'altra parte, se anche un solo raggio è inferiore a 1 metro, `allMore` diventa `false`, e il robot deve adottare una strategia per evitare l'ostacolo. In questa situazione, impostiamo `data.mutable_linear()->set_x(0.0);` e `data.mutable_angular()->set_z(0.5);`, che fa ruotare il robot. In questo modo esso può eseguire una manovra che è cruciale per evitare impatti.

Infine, indipendentemente dall'esito della verifica, il comando di movimento viene pubblicato con `pub.Publish(data);`. Tale pubblicazione è il passo finale che trasmette effettivamente il comando al sistema di movimento del robot, permettendo ad esso di reagire in base all'analisi dei dati del sensore LIDAR.

- *Metodo di pubblicazione dei comandi di movimento.* Nel metodo `main`, configuriamo e avviamo il nodo. Qui, ci iscriviamo al topic `/lidar` utilizzando `node.Subscribe(topic, cb)`. Questa iscrizione consente al nostro nodo di ricevere continuamente i dati del LIDAR e di processarli attraverso la funzione di callback `cb`.

```

int main(int argc, char **argv)
{

```

```

std::string topic = "/lidar";
// Subscribe to a topic by registering a callback.
if (!node.Subscribe(topic, cb))
{
    std::cerr << "Error subscribing to topic [" << topic << "]"
    return -1;
}

// Zzzzzz.
gz::transport::waitForShutdown();

return 0;
}

```

Una volta configurato il nodo, utilizziamo `gz::transport::waitForShutdown()`; per mantenere il nodo in esecuzione fino a quando non viene chiuso manualmente o fino a quando il programma non termina.

- *Costruzione del nodo.* Per costruire il nodo, si utilizza un processo standard con CMake, uno strumento open-source di automazione del processo di build del software. Come prima cosa è stato necessario eseguire il download del file di configurazione `CMakeLists.txt`, che troviamo di seguito:

```

cmake_minimum_required(VERSION 3.10.2 FATAL_ERROR)

project(avoid_wall)

# Find the Gazebo_Transport library
find_package(gz-transport12 QUIET REQUIRED
OPTIONAL_COMPONENTS log)
set(GZ_TRANSPORT_VER ${gz-transport12_VERSION_MAJOR})

include_directories(${CMAKE_BINARY_DIR})

if (EXISTS "${CMAKE_SOURCE_DIR}/lidar_node.cc")
    add_executable(lidar_node lidar_node.cc)
    target_link_libraries(lidar_node gz-transport$
    {GZ_TRANSPORT_VER}::core)
endif()

```

Come step successivo si crea una directory di costruzione, che abbiamo chiamato con il nome "build", all'interno dello stesso folder dove è presente il file `lidar_node`, e si eseguono i seguenti comandi per compilare il codice:

```

cmake ..

make lidar_node

```

- *Esecuzione del nodo.* Per avviare il nodo, si esegue `./build/lidar_node` da un terminale; ciò avvia il processo di iscrizione ai topic e quello di elaborazione dei dati del LIDAR.

Per verificare, ora, il funzionamento del nodo, e quindi del sensore LIDAR, in un altro terminale si avvia il mondo `sensor_tutorial.sdf`, tramite il solito comando `gz sim sensor_tutorial.sdf`. Attraverso l'interpretazione intelligente dei dati del sensore LIDAR e la generazione di comandi di movimento adeguati, siamo in grado di far navigare il robot in modo sicuro nell'ambiente, evitando ostacoli e adattandosi dinamicamente alle variazioni delle condizioni circostanti.

5.5 Ottimizzazione all'avvio di Gazebo

Un aspetto cruciale nella gestione di simulazioni complesse, come quella del nostro robot equipaggiato con un sensore LIDAR in Gazebo, è l'ottimizzazione del processo di avvio. Nel nostro caso, abbiamo semplificato tale processo creando un unico file di lancio (launch file) che consente di eseguire simultaneamente sia il mondo sensoriale che il nodo LIDAR, evitando, così, di eseguire più comandi in terminali diversi. Questo approccio riduce la complessità operativa e migliora l'efficienza dello sviluppo e del testing.

- *Creazione del file di lancio.* Il file di lancio, che consiste in un documento XML, è stato progettato per eseguire comandi specifici sotto il tag `<executable>`. La struttura del file in questione è la seguente:

```
<?xml version='1.0'?>
<gz version='1.0'>
  <executable name='sensor-world'>
    <command>gz sim sensor_tutorial.sdf</command>
  </executable>

  <executable name='lidar_node'>
    <command>./build/lidar_node</command>
  </executable>
</gz>
```

Il primo comando, `gz sim sensor_tutorial.sdf`, avvia l'ambiente di simulazione, mentre il secondo, `./build/lidar_node`, esegue il nodo LIDAR. Salvando questo file come `sensor_launch.gzlaunch`, siamo stati in grado di lanciare entrambe le componenti con un singolo comando, ovvero `gz launch sensor_launch.gzlaunch`.

- *Vantaggi dell'ottimizzazione.* Tale processo di ottimizzazione presenta diversi vantaggi. Innanzitutto, semplifica il processo di avvio, riducendo la necessità di aprire e gestire più terminali contemporaneamente. Inoltre, consente di mantenere un ambiente di lavoro più pulito e organizzato, facilitando la gestione di compiti multipli e riducendo la possibilità di errori umani durante il processo di avvio.

Un altro vantaggio significativo è l'efficienza temporale: con un singolo comando, entrambi i componenti della simulazione sono prontamente attivi e sincronizzati, permettendo di concentrarsi sull'analisi e l'interpretazione dei risultati piuttosto che sulla configurazione iniziale.

- *Avvio e funzionamento della simulazione.* Una volta lanciato il file `sensor_launch.gzlaunch`, la simulazione è stata avviata premendo il pulsante "play" nell'interfaccia grafica

di Gazebo. Il risultato è stato immediatamente visibile: il robot ha iniziato a muoversi nell'ambiente, utilizzando il sensore LIDAR per rilevare e evitare il muro fisso. Questa semplicità e fluidità nell'avvio della simulazione hanno notevolmente facilitato la fase di testing, permettendo di concentrarci sull'analisi delle performance del robot e del sensore LIDAR in situazioni simulate che riflettono scenari reali di ingegneria dell'automazione.

5.6 Esperimenti e risultati

Questo paragrafo si dedica all'esplorazione degli esperimenti condotti e dei risultati ottenuti attraverso la simulazione in Gazebo, focalizzandosi sul comportamento del robot dotato di sensore LIDAR.

Durante la simulazione, abbiamo osservato attentamente il comportamento del robot nell'ambiente virtuale. Un aspetto fondamentale che si è evidenziato è stato il modo in cui il robot, guidato dal sensore LIDAR, ha navigato attraverso lo spazio evitando il muro fisso. Grazie alla configurazione accurata del sensore, il robot ha dimostrato un'efficace capacità di percepire la presenza dell'ostacolo e di modificare di conseguenza il proprio percorso.

Il sensore in questione ha fornito dati essenziali sulla distanza tra il robot e il muro, permettendo al sistema di calcolare in tempo reale la traiettoria ottimale per evitare collisioni. Interessante è stato notare come il robot, avvicinandosi al muro, rallentasse e poi cambiasse direzione, una reazione che evidenzia l'efficacia del sistema di rilevamento e della logica di controllo implementata.

La fluidità con cui il robot ha reagito all'ostacolo nel suo percorso dimostra non solo l'efficienza del sensore LIDAR, ma anche la robustezza dell'algoritmo di navigazione impiegato. Tale comportamento è particolarmente rilevante in scenari di automazione reale, dove la capacità di navigare in modo sicuro e autonomo è fondamentale.

In sintesi, gli esperimenti condotti e i risultati ottenuti hanno fornito preziose intuizioni sulle potenzialità del sensore LIDAR in applicazioni di robotica e automazione. La simulazione ha sottolineato come una configurazione e calibrazione attente del sensore possano migliorare notevolmente la capacità del robot di interagire con il suo ambiente, un passo cruciale verso sistemi robotici più avanzati e autonomi.

5.7 Conclusioni

Riflettiamo, ora, sui punti salienti appresi dall'esperienza con il sensore LIDAR in Gazebo, integrando queste osservazioni nel contesto più ampio della tesi e contemplando le potenziali applicazioni future.

L'esperienza con il sensore LIDAR in Gazebo ha rivelato aspetti fondamentali nell'uso di questa tecnologia avanzata in scenari di robotica e automazione. Tra i punti chiave appresi, emergono l'importanza di una configurazione accurata del sensore, la necessità di un'integrazione precisa nel modello robotico, e l'efficacia del LIDAR nella navigazione e nell'evitare ostacoli. Abbiamo osservato come parametri quali il rate di aggiornamento, l'angolo di scansione, e il range operativo influenzino direttamente la capacità del robot di percepire e reagire all'ambiente circostante.

Lo studio effettuato sul sensore LIDAR si inserisce in modo coeso nel contesto più ampio della tesi, che esplora l'uso dell'Intelligenza Artificiale e dei sensori nell'ingegneria dell'automazione. Le informazioni e le competenze acquisite attraverso questa simulazione specifica forniscono una base solida per comprendere come diversi tipi di sensori e tecnologie AI possano essere integrati per sviluppare sistemi robotici più efficaci e autonomi.

Guardando al futuro, l'esperienza acquisita con il sensore LIDAR in Gazebo apre numerose possibilità in termini di ricerca e applicazioni pratiche. La capacità di questo sensore di fornire dati dettagliati e precisi sull'ambiente circostante lo rende una scelta eccellente per applicazioni in settori come l'esplorazione spaziale, la sorveglianza, la logistica e, naturalmente, la guida autonoma. Inoltre, la comprensione acquisita può essere utilizzata per sviluppare algoritmi di navigazione e di interazione ambiente-robot ancora più sofisticati, contribuendo alla realizzazione di robot sempre più autonomi e intelligenti.

In conclusione, il lavoro svolto con il sensore LIDAR in Gazebo non solo arricchisce il nostro studio con conoscenze tecniche preziose, ma apre anche la strada a interessanti sviluppi futuri nel campo dell'ingegneria dell'automazione e della robotica. Questa esperienza dimostra l'enorme potenziale delle tecnologie avanzate di percezione sensoriale, sottolineando il loro ruolo critico nell'evoluzione dei sistemi robotici autonomi.

Nel corso di questa tesi è stata esplorata l'applicazione del software Gazebo nella modellazione di un case study specifico nell'ambito dell'ingegneria dell'automazione, offrendo una prospettiva unica sull'integrazione dell'Intelligenza Artificiale e della simulazione avanzata in ambienti complessi. Il nostro lavoro ha messo in luce come Gazebo, con le sue funzionalità avanzate, sia un alleato indispensabile nella creazione di scenari di simulazione realistici e dinamici, essenziali per affrontare le sfide dell'automazione moderna.

Nell'affrontare la configurazione e l'implementazione del sensore IMU, abbiamo potuto sperimentare in prima persona la complessità e l'importanza di una calibrazione accurata e di una configurazione attenta. Le simulazioni svolte con il sensore in questione hanno permesso di comprendere come le variazioni, anche minime, nelle impostazioni possano influenzare in maniera significativa l'accuratezza e l'affidabilità dei risultati. Questo ha sottolineato l'importanza di una comprensione approfondita dei parametri del sensore e della loro influenza sul comportamento del sistema simulato. Analogamente, l'esperienza con il sensore di contatto ha offerto un'occasione unica per esplorare l'interazione tra gli oggetti fisici simulati e l'ambiente virtuale creato in Gazebo. La creazione di ostacoli virtuali e la configurazione del sensore di contatto hanno fornito una visione pratica delle sfide legate alla percezione ambientale nei sistemi di automazione. La capacità di simulare interazioni complesse tra componenti robotici e ostacoli virtuali ha dimostrato l'efficacia di Gazebo nel riprodurre situazioni che potrebbero essere difficili o pericolose da replicare in un ambiente reale. Infine, l'esperienza con il sensore LIDAR ha rappresentato un punto di svolta nel nostro studio, permettendoci di approfondire la nostra comprensione dell'elaborazione dei dati sensoriali e dell'ottimizzazione dei processi di automazione. La configurazione di tale sensore in Gazebo e l'analisi dettagliata dei dati raccolti hanno rivelato come la precisione e la rapidità di risposta di questi sensori siano fondamentali in applicazioni di automazione, specialmente in scenari che richiedono una mappatura accurata e un rilevamento rapido degli ostacoli. In sintesi, il nostro viaggio attraverso la modellazione di questo case study ha non solo arricchito la nostra comprensione teorica dell'ingegneria dell'automazione, ma ha anche fornito una base solida di esperienza pratica. Questa esperienza ha rafforzato la nostra convinzione che strumenti di simulazione avanzati, come Gazebo, siano essenziali per lo sviluppo e il test di soluzioni innovative nel campo dell'automazione.

L'utilizzo del software in questione ha permesso di superare sfide significative nell'automazione, offrendo la possibilità di simulare con precisione scenari complessi e di valutare l'efficacia dei sensori e delle strategie di automazione in condizioni controllate. Ciò ha avuto un impatto notevole sulla comprensione e sul miglioramento dell'efficienza, della sicurezza e dell'affidabilità dei sistemi automatizzati. Inoltre, l'interazione con i sensori ha evidenziato

l'importanza di un'accurata configurazione e calibrazione per garantire risultati ottimali nella simulazione, aspetti che si rivelano cruciali anche in contesti reali.

Guardando al futuro, la continua evoluzione di Gazebo e l'impiego di tecnologie avanzate come l'Intelligenza Artificiale aprono nuove strade di ricerca e applicazione. In particolare, l'emergere di concetti come Industry 5.0 pone l'accento sull'interazione uomo-macchina e sulla personalizzazione, aspetti che potrebbero essere esplorati ulteriormente attraverso l'utilizzo di strumenti di simulazione avanzati come Gazebo. La prospettiva di integrare ulteriormente l'AI in Gazebo, rendendo le simulazioni ancora più intuitive e adattive, rappresenta un orizzonte entusiasmante per futuri sviluppi e ricerche.

In conclusione, questa tesi ha dimostrato che l'utilizzo di Gazebo nel contesto dell'ingegneria dell'automazione non solo migliora la comprensione e l'efficacia delle soluzioni di automazione, ma apre anche la strada a infinite possibilità di innovazione e di progresso tecnologico nel settore. La nostra esperienza personale con questo strumento è stata estremamente formativa, fornendoci un solido fondamento per la nostra futura carriera e per ulteriori ricerche nel settore dell'automazione avanzata.

- (2018), «ROS/Gazebo-Based Simulation of Quadcopter Aircrafts», .
- (2019), «Toward Maritime Robotic Simulation in Gazebo», .
- ABBAS, H. A. (2019), «Social Integration of Artificial Intelligence: Functions, Automation Allocation Logic and Human-Autonomy Trust», *Cognitive Computation*.
- ALEXANDRU STAN, M. O. (2022), «Applied learning of artificial intelligence techniques by using the Gazebo simulator and Turtlebot3 multi-robot system», Rap. tecn., Petroleum-Gas University of Ploiesti.
- BERNARDINI, A. (2023), «Introduzione all'intelligenza artificiale», *Fondazione Ugo Bordoni, Ricerca e Innovazione*.
- CALINON, S. (2021), «An Open-Source ROS-Gazebo Toolbox for Simulating Robots With Compliant Actuators», *frontiers in Robotics and AI*.
- CINZIA BERNARDESCHI, M. P. G. S. F. S., ADRIANO FAGIOLINI (2018), «ROS/Gazebo based simulation of co-operative UAVs», Rap. tecn., Department of Information Engineering, University of Pisa, Italy.
- FRACKIEWICZ, M. (2023), «Exploring the Next Generation of Robot Simulators with Gazebo Harmonic», .
- HISHAM O. KHOGALI, S. M. (2023), «The blended future of automation and AI: Examining some long term societal and ethical impact features», *Technology in Society*.
- MOORE, P. V. (2019), *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Human Body and Motion*, Vincent G. Duffy.
- NEGNEVITSKY, M. (2005), *Artificial Intelligence: A Guide to Intelligent Systems*, Addison-Wesley.
- NIELSEN, M. (2019), *Neural Networks and Deep Learning*.
- NORHAFIZAN AHMAD, R. G., RAJA ARIFFIN e KHAIRI, N. M. (2013), «Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications», *International Journal of Signal Processing Systems Vol. 1*.
- RAIMUNDA BUKARTAITE, D. H. (2023), «Automation, artificial intelligence and future skills needs: an Irish perspective», *European Journal of Training and Development*.

- STEPNEY, S. (2012), *Proceedings of the 2012 Workshop on Complex Systems Modelling and Simulation*, Luniver Press.
- WANG, J. (2022), «How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion», Rap. tecn., ngenuity Labs Research Institute, Queen's University.
- ZAMBONI, C. (2023), «AI: applicazioni concrete dell'intelligenza artificiale in ogni settore», .
- ZANDRA B. RIVERA, D. G., MARCO C. DE SIMONE (2019), «Unmanned Ground Vehicle Modelling in Gazebo/ROS-Based Environments», *machines*.

Siti web consultati

- IBM – www.ibm.it
- Sandvik – www.home.sandvik
- Sick, Sensor Intelligence – www.sick.
- Gazebo – <https://gazebo.org>
- GNU Operating System – www.gnu.org
- ROS – <https://www.ros.org/>
- TensorFlow – www.tensorflow.org
- PyTorch – <https://pytorch.org>
- MathWorks – <https://it.mathworks.com>
- Ipertech, Configuratore di prodotto – <https://configuratore.net>
- SDFFormat – <http://sdformat.org>
- The Robot Report – www.therobotreport.com
- HBK – www.microstrain.com
- StudySmarter – www.studysmarter.co.uk
- CONSYSTEM – www.consystem.it
- Manufacturing Tomorrow – www.manufacturingtomorrow.com
- Wikipedia – www.wikipedia.org

Ringraziamenti

Terminati questi tre anni di studio, e compiuto questo passo, ci tengo molto a ringraziare diverse persone.

Sarò per sempre grato, innanzitutto, ai miei genitori, Mauro e Valeria, che mi hanno consentito di svolgere i miei studi e mi hanno sempre sostenuto in tutto ciò che ho fatto e nelle scelte che ho preso, sapendomi anche dare preziosi consigli, così come mio fratello Francesco, e tutto il resto della mia famiglia.

Un'altra persona a cui dedico un enorme grazie è Alessandro, mio compagno di studi dal primo giorno e divenuto ben presto grande amico. Abbiamo trascorso tre anni a frequentare lezioni e studiare, aiutandoci e sostenendoci reciprocamente, condendo il tutto con innumerevoli cavolate e scherzi, i quali hanno reso tutto questo tempo trascorso insieme molto più piacevole e gradevole. Vorrei ringraziare, inoltre, il mio collega, nonché amico, Gianluca, che ho conosciuto soltanto quest'ultimo anno. Sono molto contento di aver condiviso con lui progetti, informazioni, ma anche battute e risate, riuscendo, anche in questo caso, ad alleggerire le situazioni o i problemi che ci siamo trovati davanti. Un'altra persona in particolare a cui voglio dire grazie è il mio caro amico Giacomo, che mi ha sempre aiutato nel momento del bisogno, nonostante anch'egli avesse innumerevoli impegni. Con il suo bagaglio di conoscenze e la sua pazienza, è riuscito a rendermi più semplici situazioni che, a me personalmente, apparivano, invece, complesse.

Altre persone che vorrei ringraziare sono i miei amici, quelli di sempre, che conosco da una vita. Ognuno con i suoi impegni, i suoi studi o i suoi lavori, le proprie storie e novità da raccontare, le birre bevute insieme, accompagnate dalle chiacchiere fatte. È sempre stato un piacere parlare con loro, confrontarci, sostenerci ed, in alcuni casi, aiutarci, svagarsi insieme qualche fine settimana e riprendere poi le proprie vite, universitarie o lavorative, sempre pronti poi a raccontarci cosa accadesse di nuovo ad ognuno.

Vorrei, inoltre, ringraziare il professor Ursino, per la possibilità che mi ha dato di realizzare la tesi in sua compagnia, per la sua immediata prontezza, puntualità, gentilezza e cordialità. Un grazie va, anche, a tutti i ragazzi e tutte le persone che ho conosciuto in questi tre anni, con i quali ho passato tempo in università, a lezione ed a studiare, condividendo appunti, nozioni, consigli, ma anche storie personali, risate e tanto altro.

Grazie a tutti, per essermi stati vicini e per avermi aiutato.