



UNIVERSITÀ DI PISA
FACOLTÀ DI INGEGNERIA

**CORSO DI LAUREA MAGISTRALE IN INGEGNERIA ROBOTICA E
DELL' AUTOMAZIONE**

TESI DI LAUREA

**COMUNICAZIONE PER LA COOPERAZIONE E IL
CONTROLLO DI VEICOLI SUBACQUEI
AUTONOMI IN OPERAZIONI OFF-SHORE**

Relatori:

Prof. Andrea Caiti
Prof. Lorenzo Pollini
Ing. Davide Fenucci

Candidato:

Alberto Consensi

Anno Accademico 2015 – 2016

Alla mia famiglia

A Nunzia

*“Regrets, I’ve had a few
But then again, too few to mention
I did what I had to do
And saw it through without exemption
I planned each charted course,
Each careful step
Along the byway
And more, much more than this, I did it my way*

*Yes, there were times, I’m sure you knew
When I bit off more than I could chew
But through it all, when there was doubt
I ate it up and spit it out
I faced it all and I stood tall and did it my way”*

(F.Sinatra)

Sommario

Il lavoro descritto in questo documento ha avuto come scopo il raggiungimento degli obiettivi prefissati dall'Università di Pisa, in qualità di partner all'interno del progetto MARIS (*Marine Autonomous Robotics for InterventionS*). Il task principale è stato quello di sviluppare un sistema di comunicazione e localizzazione acustico subacqueo tra due AUV impiegati in operazioni di manipolazione e trasporto coordinato di oggetti in ambito off-shore. Pertanto, dopo aver realizzato uno specifico software in ambiente ROS (*Robotic Operating System*) per i dispositivi Evo Logics[®] utilizzati, il sistema è stato prima validato e successivamente testato in mare con il supporto del CSSN (*Centro Supporto e Sperimentazione Navale*) della Marina Militare Italiana. A seguito dell'analisi dei dati effettuata in post-processing, è stata svolta l'analisi delle performance del sistema e la caratterizzazione delle tempistiche di comunicazione.

Abstract

The work described in this paper was aimed to achieve the objectives set by the University of Pisa, as a partner within the MARIS project (*Marine Autonomous Robotics for Interventions*). The main task was to develop a communication and localization underwater acoustic system between two AUV which can perform underwater missions involving free-floating manipulation and transportation activities, within underwater scenarios which are deemed typical for the off-shore industry. Therefore, after realizing a specific software using ROS (*Robotic Operating System*) for Evo Logics[®] devices, the system was first validated and after tested in water, with the support of CSSN (*Centro Supporto e Sperimentazione Navale*) of Italian Navy. After data analysis done in post-processing, the analysis of system performance and the characterization of the communication timing were carried out.

Indice

1 Introduzione	6
1.1 Considerazioni iniziali	7
1.2 Il Progetto MARIS	10
1.3 Il ruolo dell'Università di Pisa	14
2 Descrizione Hardware e sviluppo Software	18
2.1 Il modem Evo Logics® S2CR 18/34 USBL	19
2.2 Analisi e selezione funzionalità	22
2.3 Il middleware ROS	29
2.4 Implementazione software	33
3 Pianificazione e realizzazione prove	39
3.1 Validazione del software	40
3.2 Analisi location e pianificazione prove	44
3.3 Allestimento e modalità di svolgimento prove	50
3.4 Sintesi dati acquisiti	54
4 Analisi dei dati e caratterizzazione del sistema	63
4.1 Analisi di precisione dei dati	64
4.2 Analisi di consistenza dei dati	77
4.3 Analisi delle prestazioni	88
4.4 Analisi delle tempistiche di comunicazione	98
4.5 Caratterizzazione RTT	113
Conclusioni	121
Bibliografia	122

1 Introduzione

La capacità che un insieme di agenti ha di cooperare dipende dalla possibilità di poter comunicare. In ambiente subacqueo, e in particolare in quello marino, questo concetto assume un'importanza ancora maggiore. La comunicazione di tipo elettromagnetico infatti, largamente utilizzata in ambito terrestre e aerospaziale, non può essere utilizzata, se non in caso di range operativi molto ridotti, a causa dell'elevata attenuazione del mezzo di trasmissione. La localizzazione e la navigazione subacquea soffrono dello stesso problema della comunicazione: l'impossibilità di utilizzare i sistemi di localizzazione globali (GPS) o locali (radio/laser) solitamente disponibili nell'ambito della robotica mobile aerea e terrestre. Anche per la localizzazione si deve far quindi ricorso a segnali acustici.

In questo contesto, dunque, la comunicazione di tipo acustico assume il ruolo di protagonista nonostante che anche questo tipo di canale soffra di alcune limitazioni.

L'efficienza di una rete di comunicazione acustica in ambito subacqueo non è infatti dipendente soltanto dalla tecnologia e dalle potenzialità dei modem acustici, ma anche dalle caratteristiche fisiche del canale di trasmissione.

Salinità, temperatura e profondità sono solo alcune tra le principali grandezze fisiche ambientali al variare delle quali si ha una alterazione del valore della velocità del suono nel mezzo, che è una delle principali grandezze caratterizzanti il canale.

In qualsiasi sistema di comunicazione acustica subacquea è insito quindi un legame imprescindibile tra le prestazioni di comunicazione e le condizioni oceanografiche locali.

1.1 Considerazioni Iniziali

Fino a poco tempo fa l'impiego degli AUV (*Autonomous Underwater Vehicles*) era limitato ad un numero molto ristretto di missioni, principalmente rientranti nell'ambito delle indagini oceanografiche, volte alla generazione di mappe dei fondali marini o alla raccolta di misure di parametri relativi alla colonna d'acqua nella quale operavano.

Tuttavia, nell'ultimo decennio, lo sviluppo tecnologico ha permesso di disporre di sistemi di alimentazione a batterie con una durata sempre maggiore e di processori con potenze di calcolo crescenti. Questi aspetti hanno permesso a questo tipo di veicoli di affermarsi sempre di più anche in scenari tipicamente riservati ai ROV (*Remotely Operated Vehicles*) o a sommergibili con equipaggio come ad esempio le attività industriali off-shore, le operazioni di soccorso, la difesa o la ricerca oceanografica [1].

Il vantaggio principale dell'utilizzo di ROV o di sommergibili era rappresentato dalla possibilità di disporre di potenti bracci robotici con i quali svolgere task di manipolazione e di avere un controllo visivo diretto sull'area d'interesse. Molteplici risultavano essere però anche gli svantaggi di questo tipo di approccio, come gli elevati costi associati all'utilizzo di grandi navi dotate di strutture in grado di gestire questo tipo di veicoli, il posizionamento dinamico dei sistemi, la gestione del cordone ombelicale nel caso dei ROV, o i rischi collegati alla presenza umana in ambiente ostile in caso di sommergibili con equipaggio [10]. Per tutte queste ragioni si è cominciato a pensare che gli I-AUV (*Intervention - Autonomous Underwater Vehicle*) potessero rappresentare l'evoluzione naturale dei ROV [2]. Senza bisogno di essere alimentati da remoto, escludendo quindi la gestione del cavo, del posizionamento e dell'uomo dall'anello di controllo, i costi di missione si riducono sensibilmente. Inoltre l'attività di gestione di questi veicoli, affidata ad imbarcazioni più piccole e quindi con costi di gestione minori, è limitata alle sole fasi di dislocazione e di recupero del veicolo, che può restare in missione anche per alcuni giorni.

Dato il recente esordio di questo tipo di approccio, sono ancora pochi gli esempi di missioni implementate con gli I-AUV. I primi riscontri si trovano a partire dalla metà degli anni novanta con l'impiego di AUV a supporto di ROV. Successivamente progetti come ALIVE [3] hanno dimostrato la capacità dei veicoli autonomi di compiere missioni che consistevano nell'approccio in hovering ad un target e nella manipolazione di valvole sullo stesso. Un altro passo importante è stato fatto nel 2009 con il progetto SAVIUM [4] il quale ha dimostrato che il veicolo era in grado di individuare e navigare verso un oggetto contrassegnato da marker artificiali le cui posizioni non erano propriamente note a priori.

Uno degli esempi più recenti di veicoli in grado di intervenire autonomamente è rappresentato dal progetto TRIDENT [1][2]. L'obiettivo del progetto è stato quello di dimostrare di poter dotare gli AUV di un'autonomia sempre maggiore anche in scenari di intervento a più veicoli. In questo caso erano coinvolti un I-AUV e un ASC (*Autonomous Surface Craft*) e la missione era divisa in due fasi.

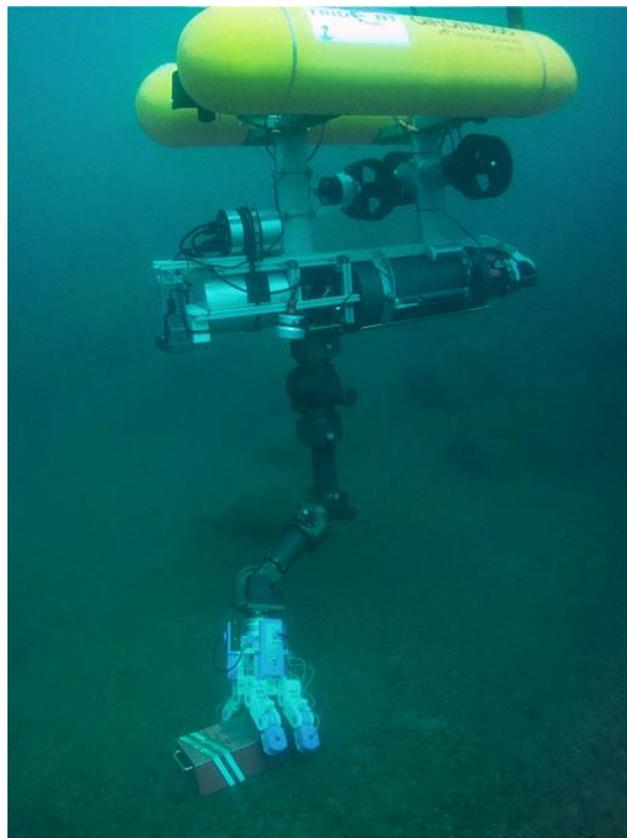


Figura 1: AUV con braccio robotico in fase di Grasping

Durante la prima fase l'AUV è responsabile dell'ispezione della zona di interesse attraverso l'utilizzo di sensori ottici ed acustici, mentre l'ASC naviga in modo coordinato per fornire fix di posizionamento attraverso moduli USBL (*Ultra Short Base Line*) [11]. Dopo aver elaborato i dati acquisiti e determinato l'oggetto da recuperare, in questo caso una scatola nera, il team entra nella seconda fase della missione dove l'AUV è responsabile, con il supporto dell'ASC, dell'individuazione dell'oggetto e del suo recupero attraverso l'utilizzo di un manipolatore.

Il principale fattore di successo del progetto è rappresentato dalla tipologia di AUV utilizzato, non solo per le sue dimensioni ridotte, che hanno semplificato l'infrastruttura necessaria per le operazioni di testing, ma anche per il design del suo particolare tri-scafo che lo rende intrinsecamente stabile e particolarmente adatto a questa tipologia di interventi.

Con TRIDENT si è aperta la strada ad una nuova metodologia multiuso per lo svolgimento di missioni subacquee nelle quali un team di robot eterogenei cooperanti con caratteristiche complementari collaborano per il raggiungimento di un task di manipolazione.

Sulla base di questi risultati è stato quindi logico pensare alla possibilità di realizzare un progetto che prevedesse la manipolazione e il trasporto di un target operati da due AUV omogenei cooperanti. Ecco come è nato il progetto MARIS [9].

1.2 Il Progetto MARIS



L'obiettivo principale del progetto MARIS (*Marine Autonomous Robotics for InterventionS*) è di studiare, sviluppare e integrare tecnologie e metodologie che consentano lo sviluppo di sistemi robotici subacquei per le attività di manipolazione e trasporto. Questi scenari sono tipici dell'industria off-shore, delle operazioni subacquee di ricerca e soccorso, nonché delle missioni scientifiche subacquee. Congiuntamente con questo obiettivo la rete proponente ha anche intenzione di dimostrare sperimentalmente, in forma proof-of-concept, le capacità operative realizzabili, integrando anche i risultati di ricerca all'interno di sistemi sperimentali reali. Sulla base delle conoscenze e delle esperienze proprie della rete proponente, delle strutture logistiche a disposizione, laboratori e attrezzature, nonché di disegni in stadio avanzato già disponibili per i sistemi sperimentali, il consorzio confida di realizzare, sviluppare e coordinare tutti gli aspetti tecnologici e metodologici necessari al raggiungimento dell'integrazione sperimentale finale a partire da sottosistemi e strumenti di design avanzati, sviluppati in parallelo messi a disposizione da alcune delle istituzioni proponenti [9].



Figura 2: Partner progetto MARIS

Partner di Progetto



Università di Genova

Coordinatore – Controllo coordinato e sensori tattili



CNR – ISSIA

Pianificazione missione e tecniche di controllo per sistemi multiagente



Università di Bologna

Design e sviluppo di mani robotiche, incluso l'aspetto sensoriale e del controllo



Università di Parma

Tecniche di visione per la pianificazione di riconoscimento e grasping



Università di Cassino e del Lazio meridionale

Coordinamento minore e localizzazione



Università di Pisa

Struttura della comunicazione acustica



Università del Salento

Guida e controllo

Al fine di raggiungere l'obiettivo finale del progetto, è necessario fissare alcuni obiettivi preliminari nell'ambito dello sviluppo integrato delle diverse tecnologie e metodologie.

In particolare MARIS mira a soddisfare i seguenti punti:

- ✓ La realizzazione di una guida e del controllo dei veicoli su base multisensoriale. Ciò avviene attraverso l'integrazione di sensori inerziali, di dispositivi di misurazione di velocità ad effetto Doppler (DVL), il supporto di dispositivi acustici esterni per la localizzazione (USBL), così come la realizzazione dello SLAM (*Simultaneous Localization And Mapping*) in tempo reale avviene attraverso l'osservazione del mondo marino.
- ✓ Realizzazione di mappe multimodali attraverso lo SLAM in tempo reale.
- ✓ Elaborazione di tecniche di stereovisione per il riconoscimento di oggetti specifici, della loro posizione e della stima della posa.
- ✓ Sviluppo di tecniche avanzate per la manipolazione e il grasping di oggetti da parte di ciascun veicolo operante in sospensione. Elaborazione di tecniche automatiche per il controllo reattivo basato su feedback di visione, forza e coppia e possibilmente sul rilevamento tattile del sistema.
- ✓ Sviluppo di metodi automatici per l'esecuzione delle operazioni di intervento coordinato da parte di gruppi cooperanti di sistemi di manipolazione in base al loro reciproco scambio di informazioni, alla loro reciproca visualizzazione e localizzazione.
- ✓ Sviluppo di tecniche di pianificazione missione ad alto livello, decomposizione e distribuzione automatica cooperativa dei compiti tra gli agenti, basate sullo scambio reciproco di informazioni.
- ✓ Sviluppo di tecniche avanzate di comunicazione acustica subacquea per le finalità espresse da entrambi i precedenti punti.

Secondo lo schema presentato il progetto MARIS mira a coordinare tutte le attività di sviluppo in parallelo volte al raggiungimento dei seguenti specifici obiettivi:

1. Destrezza e agilità per il controllo di galleggiamento dei sistemi di manipolazione, in modalità individuale o in funzione della squadra.
2. Sperimentazione del sistema autonomo “manipolatore unico galleggiante”
3. Sperimentazione del sistema autonomo “squadra di manipolatori galleggianti”

Il primo obiettivo si riferisce alle attività dedicate all’ulteriore sviluppo e all’integrazione sinergica delle tecnologie e delle metodologie precedentemente descritte. E’ finalizzato all’implementazione di capacità di destrezza e coordinamento da parte di singoli manipolatori o con l’aggiunta di funzionalità cooperanti quando si opera in gruppo. Il secondo ed il terzo obiettivo, invece, si riferiscono alla realizzazione dei prototipi e dei sistemi necessari al fine di dimostrare le capacità derivanti dalle tecniche precedentemente sviluppate sia nel caso di un singolo veicolo che nel caso di veicoli cooperanti.



Figura 3: Animazione manipolazione coordinata (sx), braccio robotico utilizzato (dx)

1.3 Il ruolo dell'Università di Pisa

L'attività metodologicamente centrale di UNIPI nel progetto MARIS è relativa al seguente task:

- ✓ Comunicazioni acustiche subacquee.

Da questa derivano le attività relative ai task di secondo livello che sono:

- ✓ Localizzazione
- ✓ Pianificazione con vincoli di comunicazione
- ✓ Sviluppo set-up del sistema di comunicazione

per i quali si implica la presenza di un sistema di comunicazione multi-veicolo.

L'attività dunque, nella sua forma più generale, comprende lo studio, la definizione e l'implementazione delle modalità di comunicazione più appropriate sia per i compiti di monitoraggio e coordinamento generale di missione, sia di coordinamento fine nelle specifiche operazioni di intervento e manipolazione.

Il profilo di missione del progetto MARIS è concettualmente suddiviso in tre fasi:

1) Fase di navigazione

In questa fase i due veicoli, una volta calati in acqua, in generale anche da posizioni diverse e distanti tra loro, navigano autonomamente fino al raggiungimento della posizione desiderata nella quale è collocato l'oggetto da afferrare.

2) Grasping

Rappresenta la fase finale di approccio, l'individuazione delle corrette posizioni di presa e il coordinamento reciproco per l'afferraggio dell'oggetto attraverso l'utilizzo dei bracci robotici.

3) Trasporto Coordinato

In questa fase i due veicoli, dopo aver afferrato l'oggetto, lo sollevano e provvedono al trasporto dello stesso manipolandolo in maniera coordinata.

In particolare si prevede che per il coordinamento generale di missione (raggiungimento della zona operativa, disposizione spaziale dei robot, approccio all'intervento) la comunicazione tra i veicoli sia di tipo acustico, in modo da realizzarsi anche a distanze moderatamente grandi (centinaia di metri).

Saranno pertanto utilizzati modem acustici commercialmente disponibili e verrà studiata la modalità di networking più opportuna in grado di ottimizzare specifici parametri di performance relativi alla qualità della comunicazione come il Round Trip Time e la Packet Loss.

Per quanto riguarda il coordinamento fine tra i veicoli in operazioni quali la manipolazione coordinata, si rende necessario un sistema di comunicazione punto-punto a distanza ravvicinata di qualche metro. Per il raggiungimento dell'obiettivo si investigheranno algoritmi di localizzazione che sfruttino la presenza sui veicoli di dispositivi acustici di tipo USBL, che consentono al veicolo che li ospita di determinare distanza e angolo di arrivo di ping acustici provenienti da altri veicoli.

Poiché il settore stesso delle reti acustiche subacquee si è fortemente sviluppato solo negli ultimi dieci anni, il trasferimento di una infrastruttura di rete nel dominio delle applicazioni, per esempio robotiche, non è ancora completamente avvenuto.

Anche a livello di ricerca sono relativamente pochi gli esempi di sistemi subacquei multi-robot dialoganti attraverso una rete di comunicazione acustica. Il problema principale al quale si deve far fronte in questi sistemi è il mantenimento di un livello accettabile di coordinamento del sistema multi-robot a fronte delle difficoltà di comunicazione. I sistemi studiati fino ad ora prevedono di mantenere gli agenti all'interno di un raggio di sicurezza, per garantire il mantenimento della connettività o comunque una "gentle degradation" delle prestazioni dell'intero sistema.

Questa soluzione, sebbene robusta, implica, in generale, vincoli eccessivamente stringenti sulla geometria e velocità degli agenti, limitando le performance del sistema. Inoltre, a causa delle limitazioni introdotte dal canale di trasmissione, l'informazione scambiata deve essere la minima necessaria per garantire il coordinamento [5].

Recentemente è stata studiata la possibilità di utilizzare la comunicazione ottica anche nell'ambito subacqueo per comunicazioni a breve distanza. In particolare è stata dimostrata in laboratorio la possibilità di trasmettere otticamente fino al Gb/s alla distanza di 2 m, usando laser verdi. Tuttavia il sistema non si presta, a causa delle dimensioni e delle difficoltà di puntamento, all'implementazione su robot subacquei.

Sistemi basati su emettitori LED sono stati studiati, e il loro utilizzo in sostituzione del cavo ombelicale per il controllo remoto di un robot è stato discusso.

I risultati sperimentali mostrano capacità di trasmissione fino a 30 m di distanza in acque trasparenti con capacità maggiori del Mb/s, e capacità di trasmissione in acque torbide (visibilità 3 m) fino a 9 m con capacità di 0.6 Mb/s. Sono state presentate anche soluzioni integrate con un modem acustico per trasmissione a grande distanza ed un modem ottico a LED per trasmissione ravvicinata a grande capacità.

Non è ancora stata presentata nessuna soluzione che preveda l'utilizzo di modem ottici in una rete di robot subacquei.

E' evidente che la necessità di mantenere contatto visivo tra i modem ottici rappresenta un ostacolo sistematico alla versatilità di utilizzo del sistema di comunicazione [11].

A questo punto la letteratura, per questa tipologia di modem, si è concentrata specificamente sul compito di “data mule”, cioè sul trasferimento di quantità ingenti di dati da robot a boa/stazione fissa o tra robot.

All'interno del progetto MARIS, ed in particolare all'interno del ruolo ricoperto dall'Università di Pisa nel progetto, si inserisce il lavoro di tesi qui riportato che ha avuto come scopo principale quello di implementare, testare e caratterizzare il sistema di comunicazione acustico attraverso le fasi qui di seguito elencate e descritte dettagliatamente nei capitoli successivi.

- ✓ Studio delle funzionalità del modem acustico, implementazione e sviluppo del software.
- ✓ Validazione del codice, pianificazione e realizzazione delle prove in mare.
- ✓ Analisi in post-processing dei dati raccolti e caratterizzazione del sistema.

2 Descrizione Hardware e Sviluppo Software

In questo capitolo sono descritte le varie fasi operative preliminari che è stato necessario svolgere per poter implementare e testare il sistema di comunicazione.

Nella prima parte del capitolo l'attenzione è focalizzata sull'hardware, quindi in particolare sui modem acustici che sono stati utilizzati per la realizzazione di questo lavoro.

Dopo una breve presentazione delle caratteristiche tecniche principali dei dispositivi, è descritta la metodologia utilizzata per la selezione delle funzionalità consone al raggiungimento del task previsto.

Nella seconda parte del capitolo viene invece descritta la realizzazione dello sviluppo del software e l'implementazione delle varie funzionalità attraverso l'utilizzo del middleware ROS (*Robot Operating System*).

2.1 Il modem Evo Logics® S2CR 18/34 USBL

Il modem USBL utilizzato per l'implementazione delle comunicazioni acustiche subacquee relative al progetto è un dispositivo universale per il tracking e la comunicazione in acque poco profonde riverberanti. E' caratterizzato da un beam pattern orizzontale omnidirezionale ed è in grado di fornire una velocità di trasferimento dati fino a 13.9 kbit/s ad un range massimo di 3500 m.



Figura 4: Modem USBL Evo Logics Utilizzato (sx), beam pattern del dispositivo

Nella tabella successiva sono riportate le principali caratteristiche tecniche del dispositivo.

Il protocollo di comunicazione implementato dal produttore sul dispositivo prende il nome di D-MAC ed ha come concetto chiave quello di usare differenti algoritmi per la trasmissione di grandi volumi di dati, che prendono il nome di *burst data*, o di piccoli volumi di dati chiamati *instant messages*.

Nel caso di utilizzo di burst data, l'efficienza dell'algoritmo di trasmissione è garantita dal fatto che vengono utilizzati i ritardi di propagazione per intrecciare i pacchetti di dati con i riconoscimenti di consegna.

I parametri caratterizzanti il canale vengono stimati inizialmente attraverso lo scambio di messaggi riservati di servizio tra i due device, al fine di mantenere il bitrate più elevato possibile. Gli stessi sono anche aggiornati dinamicamente durante la trasmissione.

I dati sono organizzati in treni di pacchetti che possono essere inviati in sequenza anche senza la ricezione dell'ack da parte del ricevente.

Profondità Operativa	200 m (Derlin)
Range Operativo	3500 m
Banda di Frequenza	18 – 34 kHz
Beam Pattern	Orizzontale Omnidirezionale
Accuratezza Slant Range	0.01 m
Risoluzione Bearing	0.1 degrees
SNR Nominale	10 dB
Connessione Acustica	Fino a 13.9 kbit/s
Bit Error Rate	Minore di 10^{-10}
Interfaccia	RS-232
Consumo	Stand-by 2.8 mW Listen 5-285 mW Receive 1.6 W Transmit 2.8 W @1000 m
Alimentazione	24 V-DC esterna
Dimensioni	Ø 110 x 170 mm Housing Ø 130 x 145 mm USBL sensor 315 mm Total length
Peso	5775 g Housing 730 g USBL sensor

Un algoritmo diverso è usato invece per la trasmissione degli Instant Messages, che hanno una lunghezza massima pari a 64 bytes e vengono trasmessi ad un bitrate fisso pari a 976 bps. Ci sono tre tipi di instant messages supportati dal dispositivo:

- ✓ Instant Messages (IM), sono messaggi che vengono inviati appena possibile e possono essere spediti con o senza la richiesta di ack da parte del ricevitore.

- ✓ Synchronous Instant Messages (IMS), sono messaggi il cui invio può essere schedulato ad un tempo predefinito.
- ✓ Piggyback Messages (PBM), sono messaggi che possono essere inviati solo se allegati ad altri dati, e pertanto l'invio non può essere schedulato.

Nonostante che il canale di comunicazione acustico sia da considerarsi tipicamente di natura half-duplex in termini di trasmissione e ricezione, il produttore afferma nel manuale che sul dispositivo è presente un protocollo in grado di garantire una comunicazione bidirezionale full-duplex tra i modem.

Nel modello di modem utilizzato è prevista anche la funzionalità di individuazione del posizionamento attraverso la tecnologia USBL. All'interno del manuale d'uso del dispositivo il produttore garantisce che la comunicazione e la localizzazione sono effettuate simultaneamente e che per ragioni di risparmio energetico non vi è un vero e proprio switch tra una modalità e l'altra. Al contrario, il calcolo dei parametri di posizionamento è effettuato durante la trasmissione tra i due device, utilizzando i segnali acustici.



Figura 5: Esempio di installazione del modem Evo Logics su AUV

2.2 Analisi e selezione funzionalità

Dopo aver illustrato in precedenza le caratteristiche tecniche del modem utilizzato, in questo paragrafo è riportata la descrizione di come è stata svolta la fase di individuazione delle funzionalità necessarie all'implementazione del sistema.

Per prima cosa è stato studiato a fondo il manuale d'uso fornito dal produttore per comprendere al meglio tutte le potenzialità del dispositivo. Dopodiché è stata individuata la modalità di trasmissione più adatta al task del progetto che è risultata essere quella degli IM.

Considerando che in questa fase l'obiettivo principale è stato quello di massimizzare la velocità di trasmissione tra i due device, la scelta fatta è supportata da due principali motivazioni.

La prima è che le informazioni che i due AUV, e quindi i due modem, dovranno scambiarsi durante la missione saranno limitate a quelle strettamente indispensabili per la guida e la manipolazione coordinata dell'oggetto, mentre le informazioni riguardanti il reciproco posizionamento sono fornite in simultanea dall'USBL. Pertanto la lunghezza massima del messaggio pari a 64 bytes non rappresenta una limitazione in questo senso.

La seconda motivazione è che gli IM vengono inviati appena possibile dal dispositivo, e questa caratteristica permette di massimizzare la velocità di trasferimento dei dati.

L'operatività del dispositivo è controllata attraverso uno specifico linguaggio di comando costituito da un insieme predefinito di *AT Commands*.

Gli *AT Commands* sono delle brevi stringhe di testo che indicano al dispositivo di eseguire una determinata azione. L'insieme di questi comandi contiene istruzioni per tutte le operazioni di comunicazione, visualizzazione, modifica e salvataggio del settaggio etc.

Il dispositivo prevede due principali modalità di funzionamento: *Data Mode* (default) e *Command Mode*.

Per eseguire istruzioni in *Command Mode* è necessario utilizzare la *Time Independent Escape Sequence* (TIES) che prevede di anteporre al comando desiderato i caratteri “+++”.

Pertanto la forma finale da utilizzare è la seguente: +++<*AT Command*>.

Tra le varie tipologie di comandi disponibili ci sono quelli di settaggio, attraverso i quali è possibile impostare tutta una serie di variabili che vanno a caratterizzare il funzionamento dei dispositivi.

I comandi che permettono di inserire il valore desiderato sono contraddistinti dal carattere ‘!’, mentre quelli che permettono di visualizzare il valore memorizzato hanno al loro interno il carattere ‘?’.

Di seguito sono riportati i principali comandi utilizzati per il settaggio preliminare.

- | | |
|-------|--|
| AT!L | Source Level
Definisce il sound pressure level (SPL) in trasmissione. E’ impostabile con valori da 0 a 3. Nel nostro caso è stato scelto di settare il valore minimo. |
| AT!G | Gain
E’ il valore del guadagno di amplificazione in ingresso. Può essere impostato high o low. E’ stato settato low in quanto valore consigliato dal produttore per test e prove a breve distanza. |
| AT!AL | Local Address
E’ l’indirizzo del dispositivo locale. I due modem utilizzati per il sistema di comunicazione sono stati inizializzati a valori 1 e 2. |

AT!RC	Retry Count	E' il numero di tentativi di connessioni acustiche, successive alla prima, che il sistema prova a stabilire all'inizio della comunicazione. Il suo valore è stato impostato a 0.
AT!RT	Retry Timeout	E' l'intervallo di tempo (in ms) che il dispositivo è disposto ad attendere per ricevere una risposta dal suo interlocutore. Il suo valore è stato variato durante le prove.
AT!CA	Sound Speed	E' il valore della velocità del suono. E' stato settato il valore di 1500 m/s.
AT!RI	IM Retry Timeout	E' il numero di tentativi di invio di un IM che il dispositivo è disposto ad effettuare in caso di insuccesso del primo. E' stato settato a 0.

Un'altra tipologia di comandi è quella relativa alla gestione degli IM. Di seguito è riportata una breve descrizione dei due comandi utilizzati.

AT*SENDIM, <length>, <destination address>, <flag>, <data> **Send Instant Messages**

Attraverso questo comando è possibile inviare un messaggio istantaneo (<data>), di lunghezza (<length>) prefissata e massima di 64 bytes, ad un device identificato da un indirizzo (<destination address>). Inoltre è possibile impostare il valore di un flag in modo tale da ricevere o meno l'ack da parte del ricevitore. Nel nostro caso la richiesta di ack è stata attivata. Una volta inviato l'IM, il dispositivo genera automaticamente un messaggio di notifica riguardo all'esito della trasmissione.

In caso di successo viene notificato il messaggio DELIVEREDIM, <destination address>, mentre in caso di insuccesso il messaggio FAILEDIM, <destination address>. Il messaggio relativo al fallimento della trasmissione viene notificato quando il trasmettitore non riceve l'ack da parte del ricevitore, quindi il messaggio potrebbe essere stato comunque consegnato.

RECVIM, <length>, <source address>, <destination address>, <flag>, <duration>, <rssi>, <integrity>, <velocity>, <data> **Receive Instant Messages**

Questo messaggio viene generato automaticamente dal ricevitore al momento della ricezione di un IM. Oltre ai campi già descritti precedentemente vengono anche forniti l'indirizzo del trasmettitore <source address>, la durata in μ s della trasmissione <duration>, il Receive Signal Strength Indicator in dB <RSSI>, un parametro del livello di integrità del messaggio <integrity> e la velocità relativa tra i due device in m/s <velocity>.

Per quanto riguarda l'invio e la ricezione di IM, attraverso il comando AT@ZX è possibile attivare anche quelle che il produttore chiama *extended notifications*.

Se quest'ultime sono attivate il dispositivo genera autonomamente messaggi di notifica aggiuntivi contenenti informazioni riguardanti le tempistiche di invio e di ricezione dei messaggi.

Le possibili notifiche sono:

RECVSTART

Questa notifica viene generata quando il dispositivo rileva un messaggio in ingresso ed è in attesa di demodularlo.

RECVFAILED, <velocity>, <rssi>, <integrity>

Notifica generata nel caso in cui in dispositivo fallisca la demodulazione del segnale in ingresso.

RECVEND, <timestamp>, <duration>, <rssi>, <integrity>

Se viene notificato questo messaggio la demodulazione del segnale in ingresso è avvenuta con successo.

SENDSTART, <destination address>, <type>, <duration>, <delay>

Questa notifica viene generata all'inizio di una trasmissione.

SENDEND, <destination address>, <type>, <timestamp>, <duration>

Notifica generata quando la trasmissione è completata.

In alcuni messaggi di notifica appena descritti appaiono alcuni nuovi campi non incontrati precedentemente. Il campo <timestamp> contiene il valore del clock di sistema, in μs , all'istante della generazione della notifica, il campo <delay> si riferisce al ritardo, in μs , introdotto dal dispositivo in trasmissione, mentre il campo <type> descrive il tipo di trasmissione avvenuta, nel nostro caso IM.

L'ultima tipologia di notifiche è quella relativa ai dati di posizionamento forniti dall'USBL. I possibili messaggi sono:

USBLLONG, <current time>, <measurement time>, <remote address>, <X>, <Y>, <Z>, <E>, <N>, <U>, <roll>, <pitch>, <yaw>, <propagation time>, <rssi>, <integrity>, <accuracy>

Questo messaggio viene generato automaticamente dal dispositivo in parallelo alla trasmissione degli IM. L'informazione di posizione del target è contenuta nei campi <X>, <Y> e <Z>, espressi in metri, riferiti al sistema di riferimento locale del dispositivo. I campi <E>, <N> e <U> contengono l'informazione di posizione compensata qualora siano disponibili segnali derivanti da sensori inerziali a bordo del veicolo. Considerazioni analoghe valgono per i campi <roll>, <pitch> e <yaw>. I campi <current time> e <measurement time> contengono valori del clock di sistema, in secondi, mentre il campo <accuracy> è riferito alla misura di posizione ed è espresso in radianti.

USBLANGLES, <current time>, <measurement time>, <remote address>, <lbearing>, <lelevation>, <bearing>, <elevation>, <roll>, <pitch>, <yaw>, <rssi>, <integrity>, <accuracy>

Anche questo messaggio viene generato come il precedente, quindi in parallelo alla trasmissione degli IM. In questo caso l'informazione di posizione del target è fornita attraverso i due campi <lbearing> e <lelevation>, espressi in radianti e riferiti al sistema di riferimento del dispositivo locale. Per tutti gli altri campi valgono le considerazioni fatte in precedenza. L'unica differenza è che in questo caso il campo

<accuracy> è espresso in metri, anziché in radianti come il caso precedente.

Quelle descritte in questo paragrafo sono tutte le principali funzionalità dei dispositivi che sono state utilizzate per l'implementazione del sistema di comunicazione.

Nei paragrafi successivi verrà descritta la fase di implementazione e sviluppo del software attraverso l'utilizzo del middleware ROS.

2.3 Il middleware ROS

ROS (*Robot Operating System*) è un meta-sistema operativo nato nel 2007 per lo sviluppo di applicazioni robotiche.

E' costituito da un insieme di pacchetti software e da librerie che mettono a disposizione dell'utente dei servizi analoghi a quelli di un sistema operativo inteso in senso classico come l'astrazione dall'hardware, il controllo dei dispositivi tramite l'utilizzo di driver, la comunicazione tra processi, la gestione delle applicazioni utente e altre funzioni di uso comune.

Inizialmente nato per un utilizzo in ambito accademico e scientifico, e su robot dedicati, ROS si è via via diffuso anche in ambiti prettamente industriali. Le ragioni di questo successo sono dovute oltre al fatto che è rilasciato sotto licenza BSD (il nome deriva dalla licenza originale che fu usata originariamente per distribuire il sistema operativo *Berkley Software Distribution*, una revisione libera di Unix sviluppata presso l'Università di Berkeley), anche nel fatto che questo insieme di strumenti risponde appieno alle attuali esigenze del mercato industriale della robotica, in cui c'è sempre più richiesta di processi di produzione ritagliati sulle caratteristiche specifiche del prodotto, e quindi verso dispositivi automatici sempre più flessibili, precisi e facili da programmare e gestire.

Sia gli strumenti per lo sviluppo che le librerie sono indipendenti dal linguaggio di programmazione utilizzato, come ad esempio C++ o Python, e la maggior parte di esse vengono rilasciate usando licenze di tipo open source.

Le principali librerie per processi ROS sono orientate verso i sistemi operativi UNIX soprattutto a causa della loro dipendenza da altri software open source.

Ubuntu Linux è elencato come sistema operativo ufficialmente supportato ed è quello che è stato utilizzato in questo lavoro. Altri sistemi operativi, tra cui Windows, sono supportati solo come sperimentali dalla comunità.

Come detto precedentemente, una delle potenzialità di ROS è quella di fornire, tra le varie funzionalità, quella di comunicazione tra processi.

Un insieme di processi all'interno di ROS può essere rappresentato da un grafo costituito da nodi, contenenti ciascuno uno o più processi, che possono inviare o ricevere messaggi da altri nodi.

Poiché l'architettura dell'intero sistema è di tipo distribuito, modulare, e basata sul paradigma *publisher/subscriber*, è possibile gestire la comunicazione tra applicazione software (nodi), che possono comunicare tra loro in maniera sincrona o asincrona.

La comunicazione sincrona è caratterizzata dalla condivisione di un canale sul quale il processo che invia il messaggio resta in attesa attiva fino a che qualcuno non lo riceve, e allo stesso modo il processo ricevente si ferma in attesa dell'arrivo del messaggio.

Nella comunicazione asincrona, invece, il processo che invia il messaggio lo fa e poi prosegue le proprie attività, così come il processo a valle si ferma solo se c'è l'effettiva disponibilità di un messaggio, altrimenti prosegue la sua normale esecuzione.

In ROS la comunicazione sincrona avviene attraverso l'utilizzo dei *services* mentre quella asincrona attraverso lo scambio di messaggio via apposite strutture chiamate *topics*.

Poniamoci nel caso rappresentato nella figura seguente dove ci sono due nodi in comunicazione tra loro.

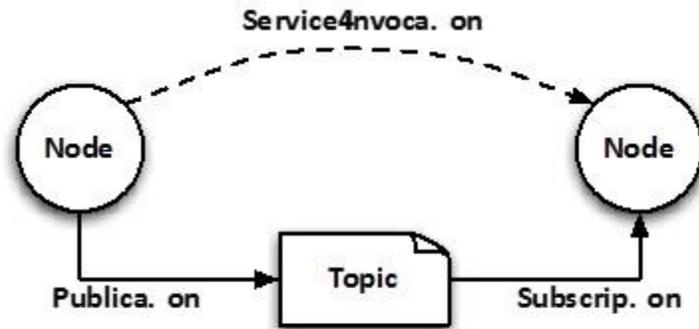


Figura 6: Schema comunicazione tra Nodi ROS

Nel caso di comunicazione asincrona si utilizza il paradigma *publisher/subscriber* rappresentato dalle frecce continue in basso.

Il nodo di sinistra quindi “pubblica” il proprio messaggio sul topic al quale il nodo di sinistra deve “isciversi” per poter accederne al contenuto. La comunicazione sui topics avviene attraverso l’invio di ROS *messages* tra i nodi.

Affinché il *publisher* e il *subscriber* possano comunicare, devono essere in grado di inviare e ricevere lo stesso tipo di messaggio. Questo significa che un *tipo* di topic è definito dal tipo di messaggio pubblicatovi.

Il paradigma *publisher/subscriber* è un modello di comunicazione molto flessibile, ma il trasferimento da molti nodi ad altri nodi a senso unico non è appropriato per le caratteristiche proprie di un sistema distribuito, caratterizzato dalla necessità di interazioni sincrone.

I *services* rappresentano infatti un altro modo attraverso il quale i nodi possono entrare in comunicazione inviando un messaggio di richiesta (*request*) e ricevendo un messaggio di risposta (*response*).

Il nucleo dell’intero sistema è rappresentato da un processo, chiamato *master*, che si occupa di gestire la comunicazione tra i vari nodi. ROS fornisce inoltre un’utilità per *data-logging* e *play-back*, detta *rosvag*, che si occupa della gestione del salvataggio dei dati che fluiscono nei vari *topics*.

Nel caso del sistema di comunicazione implementato per i due veicoli previsti nel progetto MARIS, la scelta è stata quella di implementare via software due nodi ROS, uno per ciascun modem, sui quali viene eseguito un unico processo.

I due nodi così generati comunicano tra loro in modo asincrono attraverso l'utilizzo dei *topics* così come descritto precedentemente.

Le motivazioni che hanno determinato che ROS fosse scelto per l'implementazione software del sistema sono principalmente le seguenti:

- ✓ Preesistenza di un software di riferimento sviluppato in ROS in occasione di progetti precedenti che già utilizzavamo i modem Evo Logics®.
- ✓ Possibilità di disporre di logging gratuiti attraverso l'utilizzo delle *rosvag* per la fase di analisi in post-processing dei dati.
- ✓ Utilizzo di ROS anche in altre applicazioni presenti sul veicolo implementate dagli altri partner di progetto.

Queste sono le principali ragioni che hanno portato alla scelta dell'utilizzo di ROS piuttosto che di altri middleware come MOOS (*Mission Oriented Operating Suite*) nato esplicitamente per applicazioni subacquee ma attualmente meno diffuso e sviluppato.

2.4 Implementazione software

In questo paragrafo sono descritte le principali sezioni del software implementato per il funzionamento dei modem acustici. Il linguaggio utilizzato è il C++, ed il codice è il medesimo per entrambi i nodi.

All'interno del file *usbl_node.cpp* è contenuto il “core” del codice relativo all'implementazione di tutte le funzionalità del nodo acustico.

Per prima cosa sono state importate tutte le librerie precedentemente realizzate e tutti i file necessari relativi sia ai comandi prettamente attinenti al dispositivo, sia a funzionalità di carattere generale.

Le librerie in questione sono denominate UWLib, mentre tra i più importanti file inclusi in questa fase ci sono quelli relativi alla definizione dei messaggi di notifica utilizzati in comunicazione come *<usbl/RecvdMsg.h>*, *<usbl/USBLLONG.h>* e *<usbl/USBLANGLES.h>*, *<usbl/Performance.h>*.

Dopo la definizione dei vari *namespace*, si passa alla creazione del nodo acustico vero e proprio definito come una classe USBL_ROS.

All'interno della sezione *private* della classe sono definite numerose variabili di servizio tra le quali *MY_ID* e *DEST_ID* contenenti gli indirizzi dei due modem, *MODE* invece è la variabile che indica la modalità di comunicazione utilizzata, *msg_length* è la lunghezza del messaggio in bytes, mentre *half_duplex* è un flag di attivazione/disattivazione della modalità omonima.

Dopodiché sono definiti i quattro *ROS publisher* corrispondenti ai quattro messaggi di notifica di cui sopra.

Sempre all'interno di questa sezione sono definite numerose funzioni.

La funzione *packet_loss* ritorna la percentuale di messaggi il cui invio fallisce rispetto a quelli consegnati implementando la seguente funzione:

$$packet_loss = \frac{failed}{failed+delivered} * 100$$

Questa funzione assumerà un ruolo molto importante in quanto rappresenterà uno dei principali parametri di performance del sistema di comunicazione.

Per l'implementazione delle funzioni di gestione della comunicazione acustica è stato necessario definire preventivamente di quali modalità di comunicazione fosse necessario disporre. Pertanto le modalità scelte sono le seguenti:

- ✓ TX-RX
- ✓ TRH-TRH
- ✓ TRF-TRF

La prima modalità prevede che uno dei due modem funzioni esclusivamente da trasmettitore (TX) e l'altro esclusivamente da ricevitore (RX) creando in questo modo un flusso monodirezionale di informazione.

Questa modalità va in direzione opposta al task principale, e cioè quello di instaurare una comunicazione bidirezionale tra i dispositivi, ma è stata concepita fin dall'inizio non con finalità operative, ma con l'unico scopo di avere un protocollo di trasmissione semplice sul quale effettuare i test.

La seconda modalità consiste invece in un rapporto di tipo half duplex tra i dispositivi. Questo tipo di protocollo prevede l'instaurarsi di una comunicazione monodirezionale a divisione di tempo nella quale ogni device opera alternativamente sia come trasmettitore che come ricevitore.

La terza ed ultima modalità di comunicazione è invece di tipo full duplex, ovvero entrambi i modem possono trasmettere e ricevere contemporaneamente fungendo allo stesso tempo sia da trasmettitore che da ricevitori.

E' stato scelto di implementare anche questo tipo di comunicazione, nonostante le caratteristiche intrinsecamente half duplex del canale acustico subacqueo, in quanto il produttore nel manuale afferma che i dispositivi possono funzionare anche in questa modalità.

Per l'implementazione di queste tre modalità di comunicazione è necessario realizzare numerose funzioni all'interno della sezione *private* della classe che costituisce il nodo acustico del dispositivo.

Per l'implementazione della modalità di comunicazione TX-RX è sufficiente inizializzare uno dei due device come trasmettitore, l'altro come ricevitore, previo controllo degli opportuni flag.

Questa inizializzazione avviene all'interno della funzione *start_transmission*.

Se il nodo viene inizializzato come RX il suo compito sarà quello di restare in costante ascolto dell'altro nodo, e di inviare un *ack* ogniqualvolta riceva un messaggio, mentre se viene inizializzato come TX provvederà ad inviare una serie di messaggi di lunghezza pari a *msg_length* ogniqualvolta sia possibile.

Tra un messaggio e l'altro viene effettuato il controllo di ricezione dell'*ack* inviato da parte del ricevitore. In caso di mancata ricezione dell'*ack* si procede comunque con l'invio del messaggio successivo.

Per l'implementazione della modalità TRH, per prima cosa è stata realizzata la funzione *half_duplex_manager* che quando invocata semplicemente modifica il valore del campo *mode* da TX a RX e viceversa.

Ciascun nodo, a seconda del valore della variabile *mode*, entra dunque in modalità TX attraverso la funzione *TX_t_work* o in modalità RX attraverso la funzione *RX_t_work*.

Queste due funzioni gestiscono il funzionamento di ogni device per il tempo prestabilito in modo del tutto analogo al caso TX - RX. L'unica differenza è che lo fanno per un tempo prestabilito il cui valore è contenuto della variabile *TRH_time*, essendo la modalità TRH – TRH a divisione di tempo.

Per l'implementazione della modalità TRF - TRF è sufficiente impostare entrambi i device in modalità TX. In questo modo si instaura un flusso continuo e bidirezionale di messaggi tra i device. Le notifiche dei messaggi ricevuti, invece, avvengono in automatico senza dover modificare il funzionamento del singolo modem dalla modalità TX a quella RX. Questo è possibile in quanto, da quello che si evince dal manuale, i canali di trasmissione e ricezione sul dispositivo sono separati.

Come ultima cosa, all'interno della sezione *private* della classe, sono presenti le definizioni delle funzioni di *callback*, cioè il corpo di quelle funzioni che all'arrivo di uno dei messaggi di notifica, aggiornano le variabili corrispondenti ai campi contenuti nei messaggi per poi pubblicarle attraverso i rispettivi *publisher*.

All'interno della sezione *public* della classe, invece, per prima cosa troviamo i costruttori del nodo acustico. Oltre a questi sono presenti altre due funzioni: la prima delle due è la funzione *init*, che inizializza tutti i parametri di settaggio con i valori inseriti dall'utente. I parametri inizializzati dalla funzione, con i rispettivi valori di default, sono:

- ✓ Numero porta seriale (/dev/ttyUSB0)
- ✓ Baud rate (19200)
- ✓ IM Retry Count (1)
- ✓ Retry Timeout (500)
- ✓ Message Length (40)
- ✓ Mode (TX)
- ✓ TRH time (1)

Sempre all'interno di questa funzione sono presenti le registrazioni delle *callback* di cui sopra e l'inizializzazione del modem attraverso una serie di chiamate ad opportune funzioni. E' in questa fase che vengono attivate sul modem anche le *extended notification*, descritte nel paragrafo precedente.

La seconda funzione presente nella sezione *public* della classe, è denominata *show_performance* e provvede, quando invocata, a pubblicare i valori di tutti i principali parametri di performance della comunicazione intercorrente.

I parametri pubblicati dalla funzione sono:

- ✓ Numero di messaggi ricevuti
- ✓ Numero di fix di posizione lineari ricevuti (USBLLONG)
- ✓ Numero di fix di posizione angolari ricevuti (USBLANGLE)
- ✓ Numero di messaggi falliti
- ✓ Valore della Packet Loss

Il file *usbl_node.cpp* termina con la funzione *main* nella quale sono presenti le chiamate alle funzioni di inizializzazione e di start del nodo acustico. Oltre a queste è presente anche la definizione e l'avvio di un thread indipendente che ad ogni intervallo di tempo pari a *pm_duration*, impostabile dall'utente, pubblica il valore dei parametri di performance relativi alla comunicazione.

La realizzazione di questo codice è stato il punto di partenza per l'utilizzo del sistema di comunicazione. A seguito di questa fase, in laboratorio, sono state svolte le prove tecniche di validazione del codice e, successivamente, i test veri e propri in mare.

Il personale coinvolto nello svolgimento di queste attività è stato sempre lo stesso, coadiuvato da altri soggetti per le prove pratiche in mare.

In accordo con quelli che sono gli obiettivi del progetto MARIS, è previsto che il sistema di comunicazione venga installato sui veicoli. In quella fase saranno coinvolti tutti i partner di progetto e il personale presente sarà eterogeneo dal punto di vista degli ambiti di sviluppo del progetto. Pertanto, fin dalla fase di realizzazione del software, il lavoro è stato svolto in modo tale che il codice risultasse “trasparente” all’utente che si trova ad utilizzare il sistema di comunicazione. E’ stato quindi realizzato un apposito file attraverso il quale l’utente può impostare tutti i parametri che caratterizzano la comunicazione semplicemente inserendo i valori desiderati negli appositi campi, senza rischiare di andare ad effettuare modifiche indesiderate direttamente all’interno del codice.

Di seguito è riportato il contenuto del file *maris.launch*. Utilizzando questo file è possibile variare il valore dei sette parametri principali della comunicazione.

```
<launch>
  <node pkg="rosbag" type="record" name="record" output="screen"
args="-o $(find usbl)/../../bagfiles/MARIS_LAB.bag -a" />

  <!-- USBL Modem -->
  <node name="USBL_1" pkg="usbl" type="usbl_node"
clear_params="true" output="screen" cwd="node">
    <param name="serial_port" value="/dev/ttyS0"/>
    <param name="baud_rate" value="19200"/>
    <param name="IM_retry_count" value="0"/>
    <param name="retry_timeout" value="1000"/>
    <param name="msg_length" value="40"/>
    <param name="pm_duration" value="120"/>
    <param name="mode" value="TRH"/>
    <param name="TRH_time" value="1"/>
  </node>
</launch>
```

3 Pianificazione e realizzazione prove

Nel capitolo precedente sono state descritte in dettaglio le fasi che hanno permesso di giungere alla realizzazione del software per il corretto funzionamento del sistema di comunicazione.

Il presente capitolo contiene invece la descrizione completa di tutte le operazioni di test, di configurazione del sistema e di classificazione dei dati che sono state svolte.

Per prima cosa dunque è riportata la descrizione della fase relativa alla validazione del codice effettuata attraverso lo svolgimento dei test in laboratorio presso il DII (*Dipartimento di Ingegneria dell'Informazione*) dell'Università di Pisa.

Successivamente si passa alla descrizione delle fasi di pianificazione e realizzazione delle prove svolte in mare presso il CSSN (*Centro di Supporto e Sperimentazione Navale*) della Marina Militare Italiana a La Spezia.

In questa sezione sono anche riportate tutte quelle modifiche effettuate sul campo nella giornata di prove.

Una volta svolti i test è stato quindi possibile disporre di una gran quantità di dati che si è reso necessario classificare attraverso l'elaborazione di apposite tabelle riassuntive, riportate nel capitolo, che sono state poi utilizzate come punto di partenza per lo svolgimento dell'analisi post-processing dei dati dalla quale deriva la fase di caratterizzazione del sistema di comunicazione descritta nel capitolo quattro.

3.1 Validazione del software

Una volta implementato il software è stato necessario passare alla sua validazione attraverso lo svolgimento di alcuni test. Per effettuare questo tipo di attività è stato sufficiente utilizzare i dispositivi in aria senza necessariamente immergerli in acqua.

Nel manuale, infatti, il produttore stesso descrive questa possibilità con l'unica raccomandazione di settare il Source Level al livello minimo possibile. Si ricorda che il Source Level definisce il Sound Pressure Level del dispositivo in trasmissione.

Un livello troppo elevato di questo parametro durante le prove in aria avrebbe potuto infatti danneggiare i trasduttori.

La configurazione utilizzata per lo svolgimento dei test è quella osservabile nelle immagini seguenti. Per prima cosa si è provveduto ad alimentare i due dispositivi con due batterie da 12 V ciascuna, collegate in serie. In parallelo ad esse è stato collegato anche un multimetro in modo tale da monitorare la tensione risultante e quindi la carica delle batterie. Uno dei due dispositivi è stato poi collegato, attraverso un cavo con interfaccia RS-232, ad un pc sul quale era caricato il software in oggetto.



Figura 7: Configurazione modem in aria utilizzata per la validazione del software



Figura 8: Batterie in serie per l'alimentazione dei modem

In questa fase iniziale la validazione del codice è stata effettuata utilizzando la modalità di comunicazione TX – RX in quanto quest'ultima rappresenta comunque una generalizzazione anche delle altre modalità in accordo con quanto descritto nel capitolo precedente.

La scelta di questo tipo di modalità ha permesso quindi di utilizzare un solo pc che è stato collegato al dispositivo facente funzione di trasmettitore. In questo modo non è stato possibile leggere dal ricevitore il messaggio ricevuto, in quanto non era presente un pc a valle, ma è stato possibile osservare, lato trasmettitore, la ricezione dell'ack inviato dal ricevitore oppure la generazione del messaggio di fallimento dell'invio del messaggio, generato dal trasmettitore stesso in assenza di ricezione dell'ack.

La generazione dell'ack da parte del ricevitore è infatti garantita dal fatto che il dispositivo sia alimentato.

Dal pc collegato al trasmettitore è stato inoltre possibile osservare la generazione della notifica USBLLONG che, in accordo con quanto previsto, viene generata per ogni messaggio consegnato.

Al termine di questo test è stato dunque possibile affermare che il codice implementato era corretto e funzionante in accordo con le specifiche prefissate.

A seguito del test appena descritto è stato deciso di effettuare un nuovo test immergendo i due dispositivi in acqua.

Per lo svolgimento di questa prova è stato utilizzato un recipiente della capacità di circa 85 litri all'interno del quale sono stati posizionati i due dispositivi così come è possibile osservare nell'immagine seguente.



Figura 9: Configurazione in acqua utilizzata per la validazione del software

In questo caso sono stati utilizzati due pc, collegati rispettivamente al trasmettitore e al ricevitore. Attraverso questa configurazione è stato possibile testare il funzionamento di tutte e tre le modalità di comunicazione implementate via software.

In particolare durante i test effettuati in modalità TX – RX è stato possibile verificare, lato ricevitore, la correttezza del messaggio ricevuto e constatare che il ricevitore stesso generava la notifica USBANGLES per ogni messaggio ricevuto.

Considerazioni analoghe, ovviamente simmetriche per i due dispositivi, sono state fatte a seguito dei test effettuati in modalità TRH – TRH, in quanto, come descritto in precedenza, questa modalità altro non è che una comunicazione monodirezionale a divisione di tempo e pertanto riconducibile alla precedente.

E' stato testato inoltre anche il funzionamento dei modem in modalità TRF – TRF ed anche in questo caso, grazie alla disponibilità di due pc, è stato possibile validare il codice sviluppato in quanto tutte le funzionalità previste funzionavano correttamente.

In merito ai test appena descritti è doveroso effettuare alcune precisazioni: l'attività di validazione del software è stata limitata al controllo della correttezza di esecuzione di tutte le funzionalità implementate con particolare attenzione ai settaggi dei vari parametri dei dispositivi e allo scambio di messaggi nelle varie modalità di comunicazione.

Non è stato ritenuto opportuno focalizzarsi sull'osservazione dei parametri di performance né sull'attendibilità delle notifiche riguardanti il posizionamento, in quanto la validazione della consistenza di questi valori non era oggetto di questa analisi.

Inoltre va tenuto presente che i test effettuati in aria sono stati svolti in condizioni operative ben diverse da quelle finali e da quelle tipiche per i trasduttori utilizzati, in grado di operare ad alcuni chilometri di distanza.

Riguardo alle prove effettuate in immersione c'è da tener conto del fatto che il volume d'acqua utilizzato era molto limitato e che le superfici lisce del contenitore hanno contribuito all'insorgenza di echi e riflessioni di vario genere che hanno inficiato senz'altro la qualità della comunicazione acustica.

Alla luce di queste considerazioni i risultati ottenuti sono stati ritenuti soddisfacenti e promettenti in previsione delle prove che sarebbero state effettuate di lì a poco in mare.

Al termine di questa fase è stato possibile iniziare la pianificazione delle prove presso il CSSN di La Spezia.

3.2 Analisi location e pianificazione prove

A seguito delle prove svolte in laboratorio inerenti alla validazione del software, è stata programmata una giornata di test in mare al fine di valutare le performance del sistema di comunicazione acustica in condizioni simili a quelle operative.

Grazie al coinvolgimento della Marina Militare Italiana nel progetto MARIS, la location scelta per lo svolgimento dei test è stata quella del CSSN di La Spezia.

Questa location era già stata utilizzata in passato anche dal personale del centro interuniversitario ISME (*Integrated System for the Marine Environment*) e pertanto era già nota la geometria degli spazi a disposizione e la tipologia delle banchine e dei fondali.

Data l'impossibilità di effettuare un sopralluogo, la logistica delle prove da effettuare nella giornata di test è stata pianificata sulla base delle esperienze precedenti in questa struttura e con l'ausilio di materiale fotografico satellitare.



Figura 10: Veduta aerea dell'area del CSSN

Gli edifici che ospitano il CSSN sono situati in prossimità del porto all'interno del golfo di La Spezia. L'immagine precedente mostra la dislocazione degli edifici circostanti e lo specchio d'acqua a disposizione.

Le caratteristiche di questa location si prestano molto bene alla fase di test dei dispositivi per numerose ragioni:

- ✓ E' possibile accedere alle banchine con gli automezzi per scaricare tutto il materiale necessario minimizzando il trasporto a mano
- ✓ E' possibile disporre di prese di corrente in prossimità di esse
- ✓ La profondità dell'acqua al di sotto delle banchine è sufficiente all'immersione dei dispositivi
- ✓ Il ridotto traffico marino permette di svolgere tutte le operazioni con disturbi minimi
- ✓ Si gode della piena collaborazione del personale militare interno alla struttura.

L'area scelta per lo svolgimento delle prove è quella evidenziata nella figura seguente.



Figura 11: Immagine tridimensionale dell'area di svolgimento delle prove

Una volta analizzata la location e definita l'area di svolgimento delle prove è stato possibile passare alla fase di pianificazione.

L'obiettivo delle prove era quello di individuare l'insieme dei settaggi più opportuno dei dispositivi per la massimizzazione della velocità di comunicazione tra i due device, unitamente all'ottimizzazione delle performance di localizzazione con USBL.

E' stata pertanto redatta una tabella, da utilizzare come vademecum per la giornata di prove, nella quale sono stati riportati i parametri di settaggio principali e più indicativi, almeno inizialmente, al variare dei quali si presume che si abbiano cambiamenti significativi delle performance.

Range	5 m	10 m	> 20 m
Configuration	banchina - banchina	banchina - banchina	banchina - largo
Message length	10, 40, 64 bytes	10, 40, 64 bytes	10, 40, 64 bytes
Source Level	minimum	minimum	minimum
Gain	minimum	minimum	minimum
IM Retry Count	0	0	0
Retry Timeout	500 ms – n.a.	500 ms – n.a.	500 ms – n.a.
Sound Speed	1500 m/s	1500 m/s	1500 m/s
Comunication Mode	TX-RX , TRH-TRH, TRF-TRF	TX-RX , TRH-TRH, TRF-TRF	TX-RX , TRH-TRH, TRF-TRF
Positioning	Register	Register	Register
Extended Notification	Register	Register	Register

L'idea è stata quella di organizzare il test in tre grandi insiemi di prove, ciascuna effettuata ad una diversa distanza intercorrente tra i due device.

I valori proposti sono stati 5 m, 10 m e un terzo di almeno 20 m. Il valore minimo è stato scelto pari a 5 m in quanto questa distanza avrebbe permesso di analizzare il

comportamento “macroscopico” della comunicazione tra i device auspicabilmente senza l’insorgenza di fenomeni derivanti dalla elevata distanza tra i dispositivi.

Poiché questo valore è anche tipico del funzionamento dei modem subacquei ottici, ad oggi in rapida diffusione, sarebbe stato inoltre possibile effettuare un confronto tra le performance dei due sistemi di comunicazione.

Il valore di 10 m è stato scelto in quanto secondo le specifiche del progetto MARIS i due veicoli dovranno manipolare tubature che ragionevolmente potrebbero avere dimensioni di circa 10 m.

Inoltre questo sarebbe anche il range massimo desiderato dei modem ottici con in quali potrebbe essere interessante fare un confronto.

Il valore di 20 m è stato scelto come limite superiore di distanza intercorrente tra i due veicoli durante la manipolazione di un tubatura. E’ evidente che l’assunzione fatta per la scelta di questi tre valori è che i due veicoli operino in acqua per la sola manipolazione di tubature e non per altri task che potrebbero far pensare anche ad una distanza relativa operativa molto più grande.

Come già detto, l’aerea individuata è quella interna al CSSN. Secondo la pianificazione prevista, le prove da svolgere alla distanza di 5 m sarebbero state effettuate da due punti delle banchine nell’area evidenziata nella foto.

Questa configurazione ha permesso di effettuare tutti i settaggi iniziali a terra con la dovuta calma e di affrontare problemi di setup iniziale.

E’ evidente che questa configurazione ha risentito dell’effetto di riflessione dovuto alle banchine in cemento, ma in questa fase l’attività principale da svolgere non era quella dell’ottimizzazione delle performance, ma quella di verificare il corretto funzionamento del sistema.

Lo svolgimento delle prove oltre i 20 m è stato previsto con l’ausilio di un’imbarcazione, posizionando un dispositivo in banchina e l’altro a bordo per simulare in modo più accurato le reali condizioni operative.

Sono stati scelti tre possibili valori di lunghezza del messaggio: 10, 40 e 64 bytes.

La scelta di questi tre valori è stata effettuata sulla base delle prestazioni dei dispositivi e delle tre fasi di missione precedentemente descritte.

Il valore di 10 bytes è stato scelto in quanto durante la prima fase della missione, quella di navigazione, potrebbe essere necessario ridurre al minimo il payload del messaggio per massimizzare la frequenza di trasmissione e di conseguenza anche quella di aggiornamento del fix di posizione tramite USBL.

In questa fase infatti, il campo critico non è la trasmissione di informazioni tra i device ma la velocità con la quale i dispositivi sono capaci di stabilire una mutua localizzazione.

Anche se questo valore non rappresenta ovviamente la minima dimensione del payload, è verosimile per questa fase della missione.

Il valore di 40 bytes è stato scelto come valore verosimile per la fase di grasping, durante la quale la quantità di informazioni che i due dispositivi si scambiano è maggiore rispetto alla fase precedente.

Per la fase di trasporto coordinato, invece, un valore verosimile del payload è pari a 24 bytes in quanto i due dispositivi necessitano di comunicarsi a vicenda i valori delle tre velocità lineari e delle tre velocità angolari. Questo valore risulta essere compreso tra i due precedentemente scelti.

Il valore di 64 bytes è stato scelto in quanto rappresenta la lunghezza massima disponibile del messaggio e è sembrato interessante effettuare prove anche con questo valore nel caso in cui, per ragioni di protocolli di guida, si rendesse necessario massimizzare la trasmissione per ulteriori campi informativi.

Per quanto riguarda il settaggio del Source Level il valore impostato è stato quello minimo, più che sufficiente per le distanze previste. Ovviamente il valore sarebbe potuto

aumentare qualora lo si fosse reso necessario. Considerazioni analoghe valgono per il settaggio del Gain.

Il valore dell'IM Retry Count previsto è stato pari a zero in quanto l'intento è stato quello di prediligere una comunicazione veloce ad una estremamente affidabile ma più lenta.

Per quanto riguarda invece il valore del Retry Timeout è stato impostato al livello minimo consentito dalle configurazioni operative scelte di volta in volta.

E' evidente che il valore ottimale è stato ottenuto effettuando un trade-off tra la velocità di comunicazione, che implica un valore più piccolo possibile, e il valore minimo che deve essere in ogni caso superiore al round trip time.

In assenza di misurazioni specifiche effettuate in loco, la velocità del suono è stata settata al valore standard di riferimento pari a 1500 m/s.

Con la parola "register" sono state indicate in tabella tutte le grandezze da registrare al fine di poterle analizzare in post-processing per valutare la bontà delle scelte effettuate e le prestazioni del sistema.

Il programma delle prove pianificate in questo paragrafo è stato concepito per essere svolto nell'arco di un'intera giornata ed è stato realizzato in data 4 Novembre 2015.

3.3 Allestimento e modalità di svolgimento prove

In data 4 Novembre 2015, presso il CSSN di La Spezia si sono svolte le prove in mare con l'obiettivo di testare il funzionamento e valutare le performance del sistema di comunicazione ultrasonica tra i due modem USBL. In accordo con quanto pianificato, le prove si sono svolte nell'area prevista e descritta nel paragrafo precedente.

La configurazione utilizzata è quella riportata nell'immagine successiva.



Figura 12: Disposizione operativa adottata per lo svolgimento delle prove

Ognuno dei due dispositivi, una volta calato in acqua dalla banchina e mantenuto a distanza dalla stessa con l'ausilio di un mezzo marinaio, è stato alimentato con delle batterie e collegato ad un pc sul quale era caricato il software sviluppato in precedenza.

Ciascun device è stato settato in maniera da notificare automaticamente informazioni sulla posizione relativa dell'altro dispositivo ("USBLLONG") e sulla direzione di provenienza del messaggio ("USBLANGLES") ogniqualvolta il dispositivo stesso fosse

in grado di estrapolarle dalle informazioni di comunicazione (tempo di volo, differenza di fase tra i segnali ricevuti dai vari traduttori componenti l'USBL).

In questa prima configurazione, durante la mattinata, si sono svolte le prove alle distanze di 5 m e di 10 m intercorrenti tra i device.

Nell'immagine seguente è possibile osservare in primo piano il mezzo di supporto di uno dei due modem e sullo sfondo l'altro modem alimentato e collegato al pc.

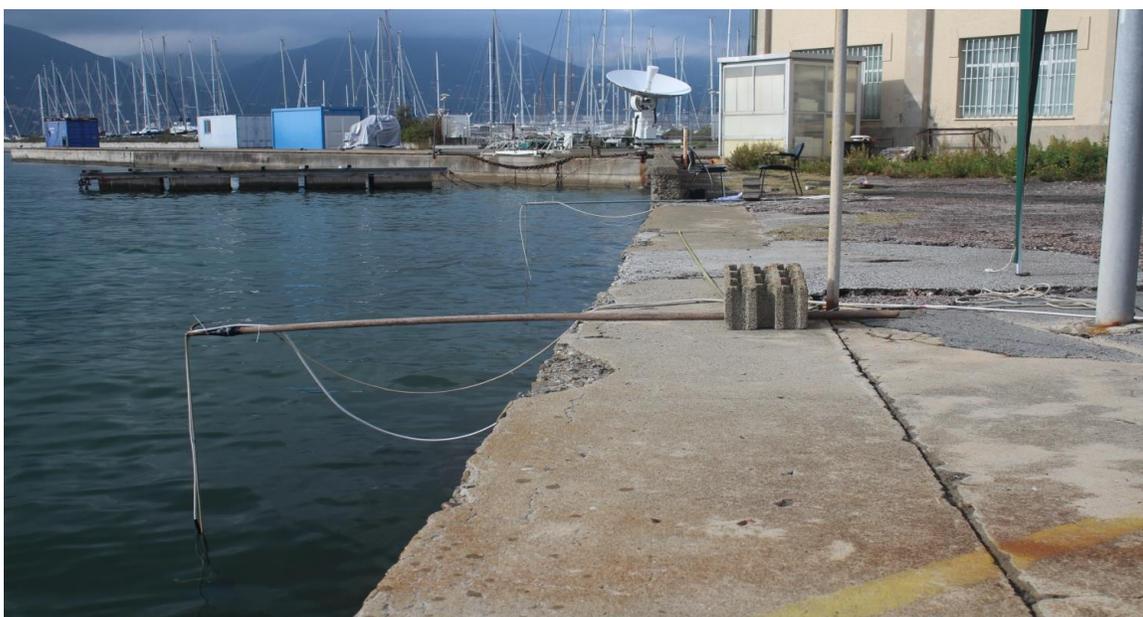


Figura 13: Mezzi marinai utilizzati per calare i modem in acqua

Nel pomeriggio, grazie all'ausilio del personale militare interno alla struttura, ci è stata messa a disposizione un'imbarcazione con relativo equipaggio per le prove a distanza maggiore.

Il natante, una volta imbarcato uno dei modem con annesse batterie e pc, ha stazionato nello specchio d'acqua antistante. In questo modo è stato possibile effettuare dei test ad una distanza compresa tra i 60 e i 90 m circa. La misura di questa distanza è stata effettuata attraverso un telemetro messo a disposizione dalla struttura e poi verificata dalle misure acquisite.

Nell'immagine seguente è possibile osservare in primo piano il supporto del modem calato dalla banchina e sullo sfondo l'imbarcazione ospitante il secondo modem.



Figura 14: In primo piano il mezzo marinaio con il modem calato in acqua, sullo sfondo l'imbarcazione con il secondo modem a bordo

Le prove di comunicazione sono state effettuate mantenendo di volta in volta fermo uno dei due USBL e variando i seguenti parametri con valori consistenti con lo scenario previsto dal progetto MARIS e corrispondenti a quelli pianificati:

- **Distanza tra i nodi:** il secondo USBL (Nodo 2) è stato spostato durante la mattinata dal punto 2 al punto 3 per effettuare le prove a 5 e a 10 m con l'altro modem posizionato nel punto 1. Nel pomeriggio lo svolgimento dei test è invece avvenuto con il posizionamento del primo nodo USBL (Nodo 1) a bordo dell'imbarcazione. In questo insieme di test la distanza intercorrente tra i modem è stata superiore a quella prevista per ragioni di sicurezza di operatività dell'imbarcazione.

- **Lunghezza del payload**

Impostata alternativamente a 10, 40 e 55 bytes, in accordo con quanto descritto nel paragrafo precedente. Per un problema relativo al software, poi risolto in laboratorio, non è stato possibile impostare la lunghezza massima del payload a valori superiori ai 55 bytes. Tuttavia le considerazioni fatte in precedenza restano valide.

- **Retry timeout**

Tempo massimo in cui un USBL attende la ricezione di un ack dall'altro modem prima di dichiarare fallita la consegna del messaggio. Tale parametro dipende dai due parametri precedenti ed è quindi stato oggetto di tuning.

- **Modalità di comunicazione**

Le configurazioni di comunicazione impostate e testate per i due dispositivi alle varie distanze sono state alternativamente:

- TX – RX: comunicazione monodirezionale con un dispositivo funzionante esclusivamente come trasmettitore e l'altro esclusivamente come ricevitore.
- TRH – TRH: comunicazione bidirezionale half-duplex a divisione di tempo.
- TRF – TRF comunicazione bidirezionale full-duplex.

- **Slot di tempo di trasmissione in modalità TRH**

La durata del periodo di trasmissione di ciascun nodo nella modalità di funzionamento a divisione di tempo (*TRH time*) dipende dal *Retry Timeout* ed è stata quindi anch'essa oggetto di tuning.

Tutti i dati registrati nella giornata di prove sono stati salvati su ciascun pc utilizzato e al rientro in laboratorio è stato effettuato un backup su hard disk esterni.

3.4 Sintesi dati acquisiti

Durante la giornata di prove sono stati registrati tutti i dati relativi alle performance di comunicazione di entrambi i dispositivi. In previsione dello svolgimento di un'analisi più approfondita di tutti i dati acquisiti, finalizzata alla caratterizzazione del sistema di comunicazione, di seguito è riportata una breve descrizione delle prove effettuate.

Nelle tabelle seguenti sono riportati in blu i parametri di settaggio dei dispositivi e in verde quelli di performance.

Poiché durante le prove l'acquisizione dei dati sui due pc è stata avviata e stoppata manualmente in modo non perfettamente sincronizzato, è possibile riscontrare nei dati una leggera discrepanza tra i valori a seconda che questi ultimi siano letti dall'uno o dall'altro pc.

Test 1 (Durata: 411 s, Range: 5 m)

ID nodo	1	2
Modalità di funzionamento	RX	TX
Lunghezza messaggio (B)	-	10
Retry timeout (ms)	-	500
Messaggi ricevuti	608	-
Messaggi consegnati	-	560
Messaggi falliti	-	35
Packet loss (%)	-	5,9
USBLONG	0	560
USBLANGLES	607	0

Note: TX genera “USBLONG” per ogni messaggio consegnato. RX genera “USBLANGLES” per ogni messaggio ricevuto.

Test 2 (Durata: 110 s, Range: 5 m)

ID nodo	1	2
Modalità di funzionamento	RX	TX
Lunghezza messaggio (B)	-	40
Retry timeout (ms)	-	500
Messaggi ricevuti	74	-
Messaggi consegnati	-	0
Messaggi falliti	-	127
Packet loss (%)	-	100
USBLONG	0	0
USBLANGLES	74	0

Note: Retry timeout troppo basso. TX non riceve l'ack, nonostante RX riceva il messaggio e generi "USBLANGLES".

Test 3 (Durata: 303 s, Range: 5 m)

ID nodo	1	2
Modalità di funzionamento	RX	TX
Lunghezza messaggio (B)	-	40
Retry timeout (ms)	-	750
Messaggi ricevuti	287	-
Messaggi consegnati	-	269
Messaggi falliti	-	21
Packet loss (%)	-	7,2
USBLONG	0	265
USBLANGLES	285	0

Note: Considerazioni analoghe al test 1

Test 4 (Durata: fail, Range: 5 m)

ID nodo	1	2
Modalità di funzionamento	RX	TX
Lunghezza messaggio (B)	-	64
Retry timeout (ms)	-	750
Messaggi ricevuti	n.d.	-
Messaggi consegnati	-	n.d.
Messaggi falliti	-	n.d.
Packet loss (%)	-	n.d.
USBLONG	n.d.	n.d.
USBLANGLES	n.d.	n.d.

Note: Fail del codice causato dalla lunghezza del messaggio.

Test 5 (Durata: 117 s, Range: 5 m)

ID nodo	1	2
Modalità di funzionamento	TRF	TRF
Lunghezza messaggio (B)	40	40
Retry timeout (ms)	750	750
Messaggi ricevuti	1	1
Messaggi consegnati	1	1
Messaggi falliti	105	111
Packet loss (%)	99,0	99,1
USBLONG	1	1
USBLANGLES	1	0

Note: Fallimento della modalità full-duplex

Test 6 (Durata: 94 s, Range: 5 m)

ID nodo	1	2
Modalità di funzionamento	TRF	TRF
Lunghezza messaggio (B)	10	10
Retry timeout (ms)	750	750
Messaggi ricevuti	0	0
Messaggi consegnati	0	0
Messaggi falliti	89	86
Packet loss (%)	100	100
USBLLONG	0	0
USBLANGLES	0	0

Note: Fallimento della modalità full-duplex

Test 7 (Durata: 465 s, Range: 5 m)

ID nodo	1	2
Modalità di funzionamento	TRH	TRH
Lunghezza messaggio (B)	40	40
Retry timeout (ms)	750	750
TRH time (s)	1	1
Messaggi ricevuti	210	204
Messaggi consegnati	196	198
Messaggi falliti	25	24
Packet loss (%)	11,3	10,8
USBLLONG	194	191
USBLANGLES	210	10

Note: TX genera “USBLLONG” per ogni messaggio consegnato ma genera pochi “USBLANGLES”. RX genera entrambi ad ogni messaggio.

Test 8 (Durata: 396 s, Range: 10 m)

ID nodo	1	2
Modalità di funzionamento	RX	TX
Lunghezza messaggio (B)	-	40
Retry timeout (ms)	-	750
Messaggi ricevuti	389	-
Messaggi consegnati	-	358
Messaggi falliti	-	23
Packet loss (%)	-	6,0
USBLLONG	0	358
USBLANGLES	389	0

Note: Considerazioni analoghe al test 1

Test 9 (Durata: 90 s, Range: 10 m)

ID nodo	1	2
Modalità di funzionamento	TRF	TRF
Lunghezza messaggio (B)	10	10
Retry timeout (ms)	750	750
Messaggi ricevuti	64	67
Messaggi consegnati	0	66
Messaggi falliti	67	5
Packet loss (%)	100	7,0
USBLLONG	0	65
USBLANGLES	64	67

Note: La comunicazione funziona solo in una direzione. Inoltre TX genera sia “USBLLONG” che “USBLANGLES”, mentre RX soltanto “USBLANGLES”.

Test 9bis (Durata: 72 s, Range: 10 m)

ID nodo	1	2
Modalità di funzionamento	TRF	TRF
Lunghezza messaggio (B)	10	10
Retry timeout (ms)	1000	750
Messaggi ricevuti	67	51
Messaggi consegnati	16	51
Messaggi falliti	69	3
Packet loss (%)	81,2	5,5
USBLLONG	16	51
USBLANGLES	67	51

Note: Considerazioni analoghe al test 9 nonostante il settaggio differenziato del TRH time sui due device.

Test 10 (Durata: 447 s, Range: 10 m)

ID nodo	1	2
Modalità di funzionamento	TRH	TRH
Lunghezza messaggio (B)	40	40
Retry timeout (ms)	750	750
TRH time (s)	1	1
Messaggi ricevuti	209	208
Messaggi consegnati	203	198
Messaggi falliti	15	14
Packet loss (%)	6,9	6,6
USBLLONG	209	197
USBLANGLES	203	11

Note: Considerazioni analoghe al test 7

Test 11 (Durata: 471 s, Range: 10 m)

ID nodo	1	2
Modalità di funzionamento	TRH	TRH
Lunghezza messaggio (B)	55	55
Retry timeout (ms)	1000	1000
TRH time (s)	1	1
Messaggi ricevuti	220	213
Messaggi consegnati	244	212
Messaggi falliti	27	13
Packet loss (%)	10,0	5,8
USBLONG	258	212
USBLANGLES	206	1

Note: Considerazioni analoghe al test 10

Test 12 (Durata: 229 s, Range: 10 m)

ID nodo	1	2
Modalità di funzionamento	TRH	TRH
Lunghezza messaggio (B)	55	55
Retry timeout (ms)	1000	1000
TRH time (s)	1,5	1,5
Messaggi ricevuti	67	64
Messaggi consegnati	67	63
Messaggi falliti	9	7
Packet loss (%)	11,8	10
USBLONG	67	63
USBLANGLES	67	1

Note: Considerazioni analoghe al test 10

Test 13 (Durata: 176 s, Range: ~ 69 m)

ID nodo	1	2
Modalità di funzionamento	RX	TX
Lunghezza messaggio (B)	-	40
Retry timeout (ms)	-	1000
Messaggi ricevuti	103	-
Messaggi consegnati	-	102
Messaggi falliti	-	42
Packet loss (%)	-	29,2
USBLLONG	0	102
USBLANGLES	103	0

Note: Performance degradata nella parte finale del test (scarroccio imbarcazione).

Test 14 (Durata: 427 s, Range: ~ 89 m)

ID nodo	1	2
Modalità di funzionamento	TRF	TRF
Lunghezza messaggio (B)	10	10
Retry timeout (ms)	1000	1000
Messaggi ricevuti	339	323
Messaggi consegnati	22	320
Messaggi falliti	348	43
Packet loss (%)	94,0	11,8
USBLLONG	22	319
USBLANGLES	338	301

Note: Considerazioni analoghe al test 9

Test 15 (Durata: 656 s, Range: ~ 65 m)

ID nodo	1	2
Modalità di funzionamento	TRH	TRH
Lunghezza messaggio (B)	40	40
Retry timeout (ms)	1000	1000
TRH time (s)	1	1
Messaggi ricevuti	314	314
Messaggi consegnati	310	309
Messaggi falliti	10	9
Packet loss (%)	3,1	2,8
USBLLONG	310	309
USBLANGLES	314	271

Note: Miglior performance della giornata

I test svolti hanno permesso di verificare il corretto funzionamento del sistema di comunicazione anche in condizioni operative simili a quelle reali.

Le performance ottenute sono da ritenersi soddisfacenti specialmente in riferimento alla Packet Loss che risulta inferiore rispetto a precedenti esperienze nell'utilizzo di questo tipo di dispositivi.

Contrariamente a quanto dichiarato dal produttore, data la natura half – duplex del canale di trasmissione, la modalità TRF – TRF, intrinsecamente full – duplex, presenta delle prestazioni notevolmente scarse e non adatte a garantire i livelli di performance richiesti. Pertanto, l'analisi dei dati in post – processing, oggetto del capitolo successivo, è stata effettuata limitatamente alle modalità di comunicazione TX – RX e TRH – TRH.

4 Analisi dei dati e caratterizzazione del sistema

Nel precedente capitolo sono state descritte in dettaglio le modalità di svolgimento delle prove sperimentali effettuate in mare finalizzate all'acquisizione dei dati.

Grazie alla disponibilità di questa grande quantità di informazioni è stato possibile effettuare una analisi approfondita dei dati raccolti e giungere alla caratterizzazione del sistema.

L'analisi dei dati è stata svolta su più fronti con lo scopo principale di valutare le performance e l'affidabilità del sistema.

Sono state effettuate infatti analisi di precisione e consistenza dei dati, di prestazione del sistema, di degradazione delle prestazioni in funzione di vari parametri operativi oltre allo studio delle dipendenze dei parametri di performance dai valori di settaggio iniziale.

Successivamente sono state effettuate le caratterizzazioni delle tempistiche del protocollo di comunicazione, della velocità di comunicazione e del settaggio del sistema.

Tutte le attività descritte in questo capitolo sono state svolte sulla base di opportune specifiche tecniche preventivamente fissate. In particolare si fa riferimento a specifiche metriche per la misura delle distanze, ai bytes per la lunghezza del messaggio trasmesso e al Round Trip Time (RTT) espresso in microsecondi per la velocità di comunicazione tra i device. Per la definizione dettagliata di queste specifiche si rimanda ai rispettivi paragrafi.

4.1 Analisi di precisione dei dati

Nel precedente capitolo, dopo aver descritto le modalità di svolgimento delle prove di funzionamento effettuate, i principali parametri di performance sono stati riportati in opportune tabelle.

Dall'osservazione di quei parametri è emerso l'interesse di procedere con una analisi dei dati relativi ai test più rappresentativi, concentrando quindi l'attenzione principalmente sulle modalità di funzionamento TX - RX e TRH - TRH per le ragioni già esposte in precedenza.

I dati, convertiti in formato *Excel*, sono stati importati nel software *Matlab* attraverso opportune funzioni, e sono stati poi analizzati attraverso la creazione di grafici.

Tutte le tipologie di analisi descritte in questo capitolo sono state effettuate su tutti i test elencati nel capitolo precedente, ad eccezione di quelli per i quali si è riscontrato un fallimento della comunicazione. In questo capitolo, per ovvie ragioni, è stato scelto di riportare i grafici relativi alle analisi effettuate sui test più significativi e comunque riferiti a condizioni operative diverse in termini di distanza tra i device, lunghezza del messaggio trasmesso e modalità di comunicazione.

La tipologia di dati dalla quale ha avuto inizio la fase di analisi e caratterizzazione del sistema, è quella relativa alle misure metriche fornite dall'USBL. Avendo a disposizione una grande quantità di dati sperimentali, provenienti da prove effettuate in condizioni operative reali, è sembrato logico e interessante avere a disposizione una visualizzazione grafica dei dati acquisiti in modo tale da capire la distribuzione macroscopica delle misure acquisite ed individuare così eventuali anomalie. Pertanto le misure acquisite tramite USBL lungo gli assi X, Y e Z, sono state riportate in dei grafici, in funzione del tempo. Di seguito sono riportati i grafici relativi ai test 3, 10 e 13 i quali fanno riferimento a condizioni operative diverse tra loro come detto poc'anzi.

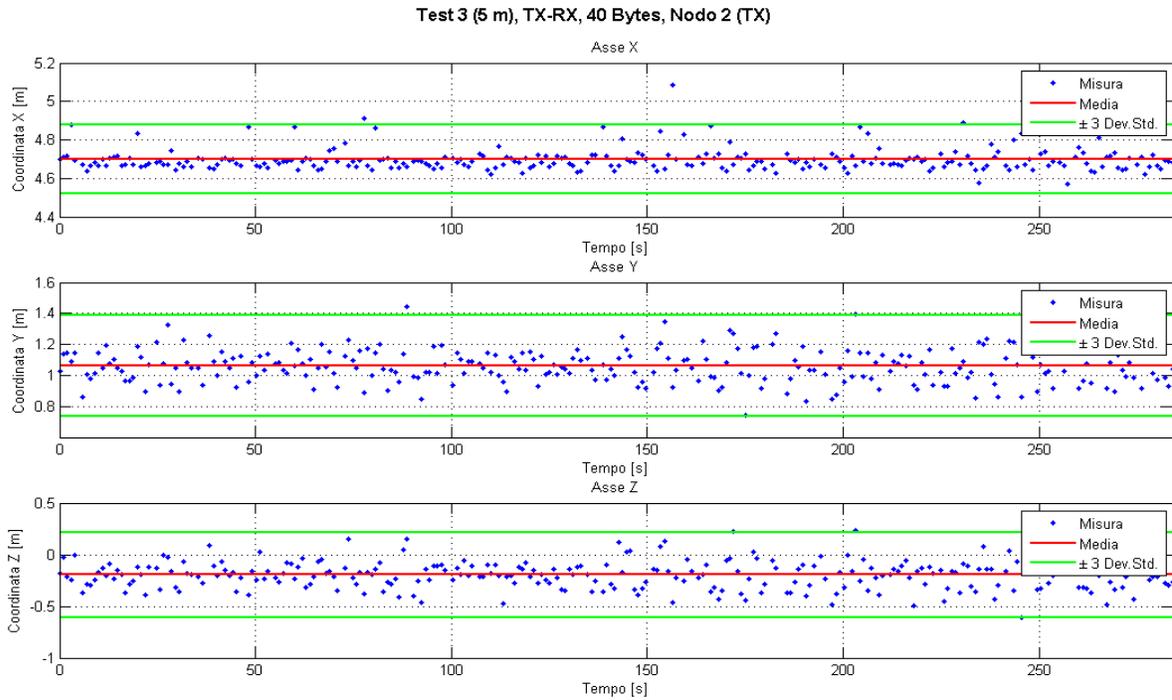


Figura 15: Distribuzione temporale delle misure di posizione del Test 3, lungo le tre coordinate

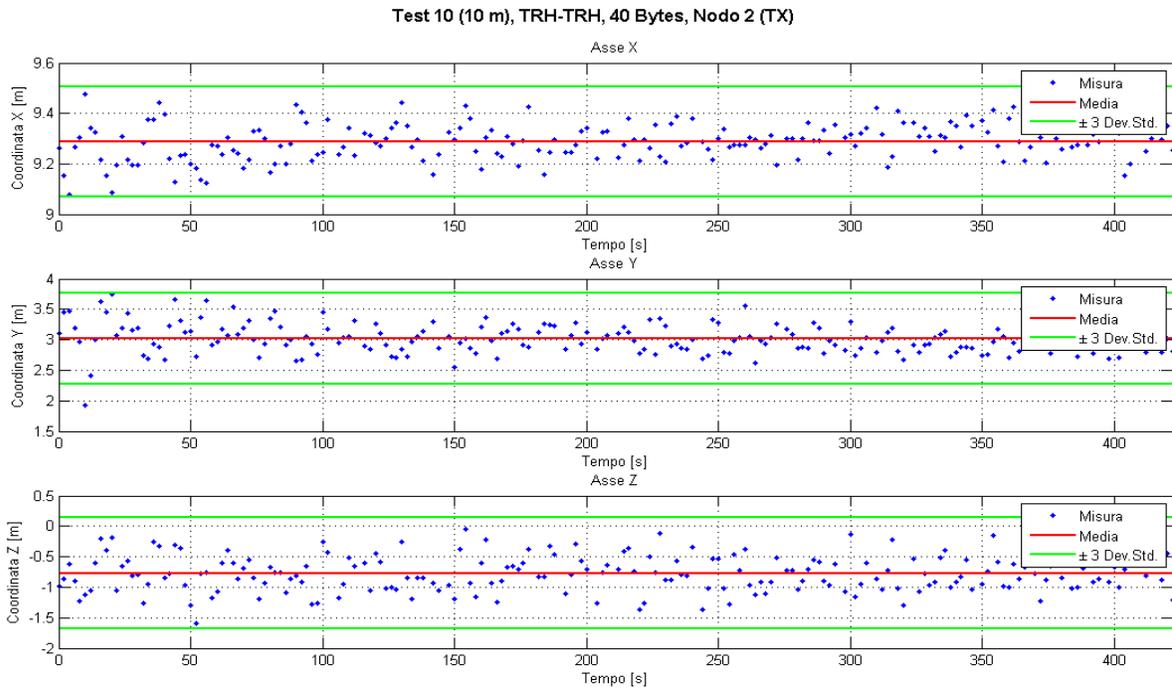


Figura 16: Distribuzione temporale delle misure di posizione del Test 10, lungo le tre coordinate

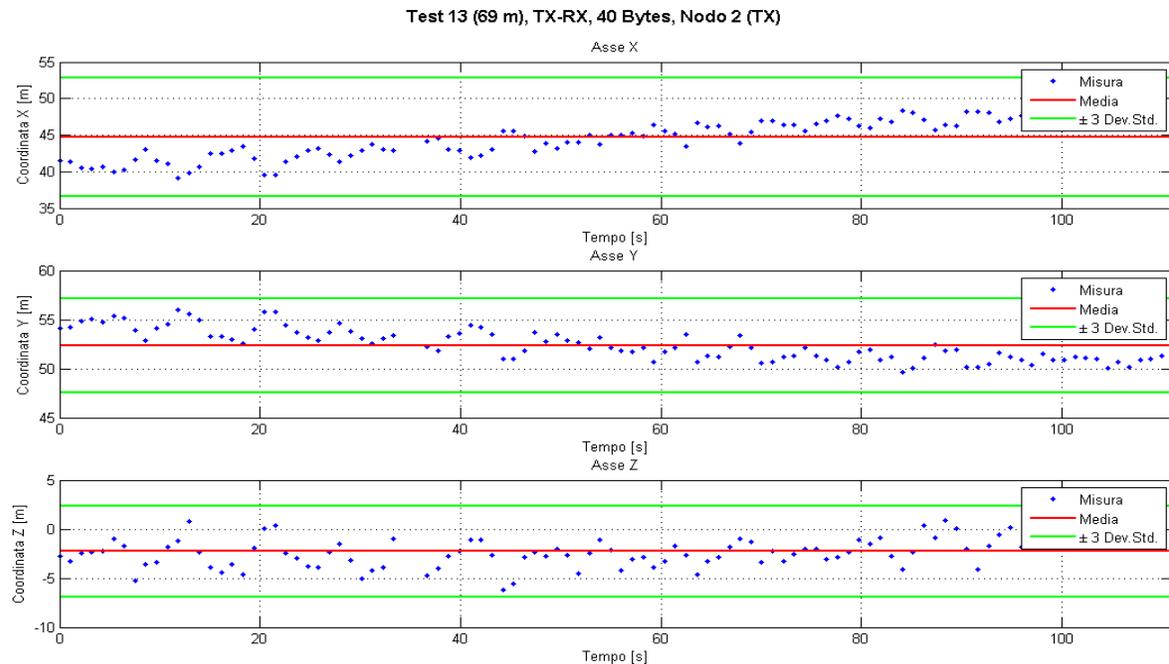


Figura 17: Distribuzione temporale delle misure di posizione del Test 13, lungo le tre coordinate

Dall'osservazione dei grafici precedenti è immediato notare che tutte le misure, su tutte le coordinate di ciascun test, sono ben distribuite e centrate intorno al valor medio rappresentato dalla linea rossa.

Il flusso dei dati risulta essere omogeneo sulla linea temporale e non si riscontrano interruzioni. Osservando meglio il terzo grafico, caratterizzato da una durata minore, nel quale dunque i punti in blu vengono visualizzati più distanziati tra loro, sembra inoltre che l'intervallo di tempo tra uno e l'altro sia costante, almeno visivamente, e questo è senz'altro un aspetto che sarà approfondito successivamente.

Sono inoltre molto rari, pari a poche unità a fronte di alcune migliaia di campioni, i valori che escono dall'intervallo a $\pm 3\sigma$ e che possono essere definiti outlier.

Pertanto, da una prima analisi ispettiva, il sistema risulta affidabile da un punto di vista della coerenza delle misure fornite e non affetto da comportamenti anomali o da malfunzionamenti di carattere generale.

Tenendo ancora sempre presente la natura sperimentale dei dati, è stato deciso di valutare la tipologia di funzione di densità di probabilità (o *pdf* dall'inglese *probability density function*) caratterizzante la distribuzione dei dati stessi all'interno del range di misura.

Per fare ciò è stata utilizzata la funzione Matlab *histfit*, in grado di plottare l'istogramma con le occorrenze risultanti in funzione dei valori di distanza misurati.

Questa funzione permette inoltre di effettuare un *fitting* della tipologia di *pdf* scegliendo tra un'ampia gamma di funzioni disponibili. Nel nostro caso la distribuzione normale, o Gaussiana, si è rivelata essere quella più rappresentativa.

Di seguito sono riportati i grafici, ottenuti secondo le modalità descritte, relativi alle misure lungo gli assi X, Y e Z fornite dall'USBL e riferiti agli stessi test utilizzati in precedenza e scelti per le motivazioni già descritte.

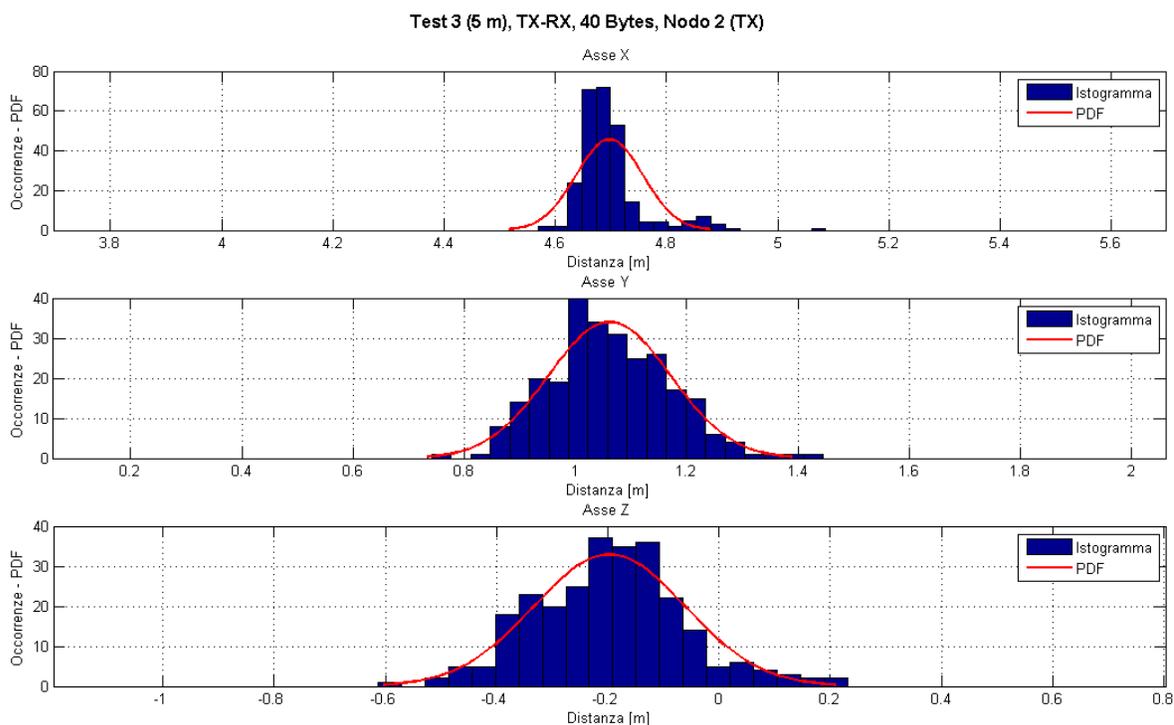


Figura 18: Fitting della funzione di densità di probabilità del Test 3, lungo le tre coordinate

Test 10 (10 m), TRH-TRH, 40 Bytes, Nodo 2

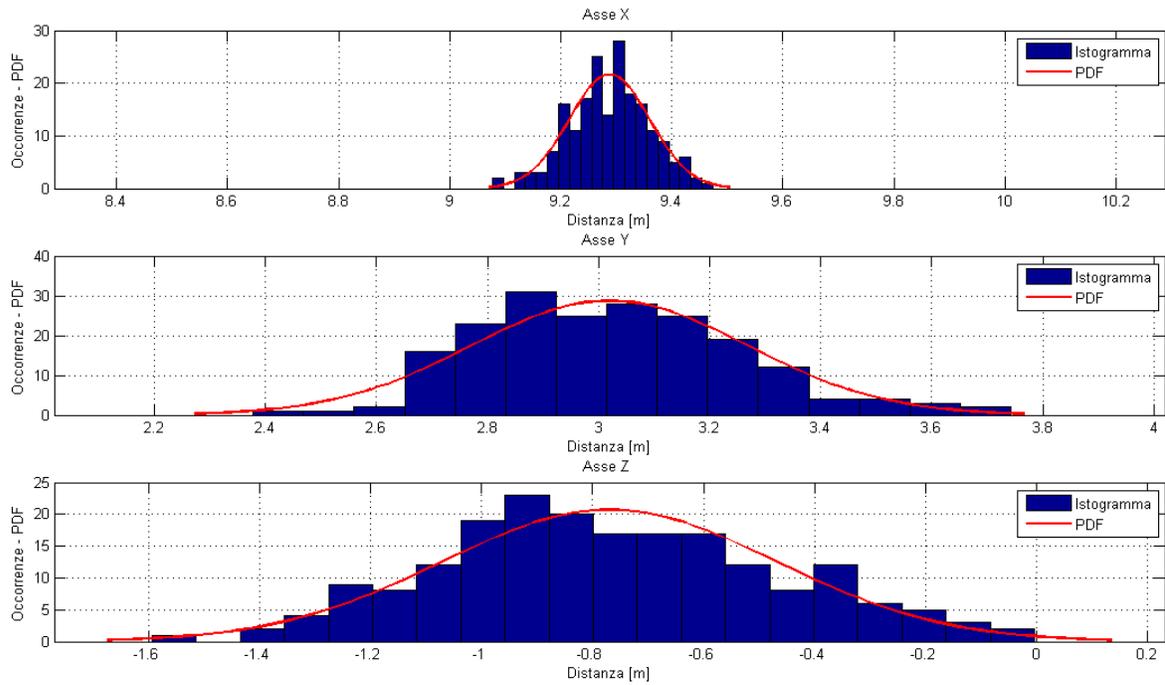


Figura 19: Fitting della funzione di densità di probabilità del Test 10, lungo le tre coordinate

Test 13 (69 m), TX-RX, 40 Bytes, Nodo 2 (TX)

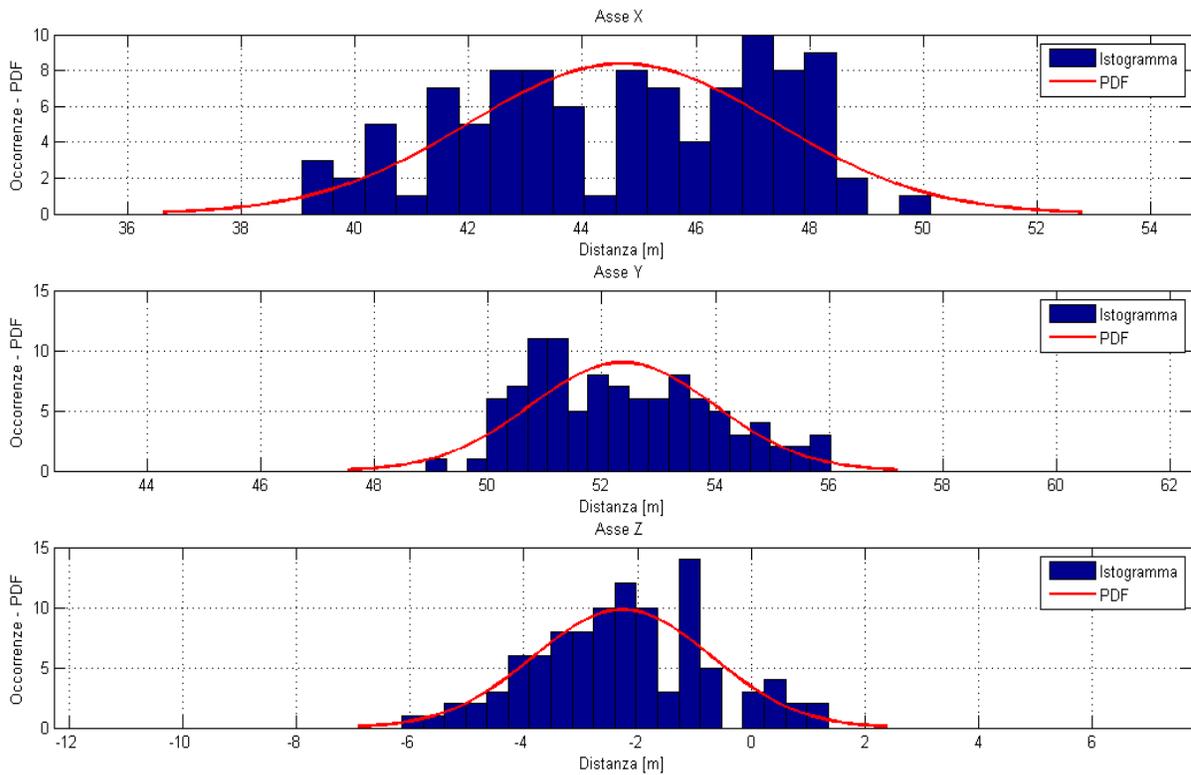


Figura 20: Fitting della funzione di densità di probabilità del Test 13, lungo le tre coordinate

Osservando i precedenti grafici relativi al test 3 e al test 10, è possibile affermare che la funzione di densità di probabilità di tipo gaussiano risulta essere in ogni caso la funzione che meglio approssima la distribuzione dei dati sull'intervallo di misure fornite dall'USBL.

Questa considerazione resta valida al variare della distanza intercorrente tra i device, della lunghezza del messaggio trasmesso e della tipologia di comunicazione utilizzata.

Pertanto il tipo di distribuzione delle misure fornite dai device non dipende dalle principali grandezze operative del sistema di comunicazione.

Per commentare e analizzare correttamente il terzo grafico, relativo al test 13, bisogna ricordare che si riferisce al caso in cui uno dei due modem era posizionato a terra e l'altro a bordo dell'imbarcazione.

Quest'ultima, durante l'acquisizione dei dati, stazionava a motori spenti di fronte alla banchina, ad una opportuna distanza di sicurezza, ma è stata inevitabilmente soggetta a scarroccio a causa delle correnti e del vento. Questo ha comportato una dispersione maggiore della distribuzione delle misure specialmente sull'asse lungo il quale la componente dello spostamento è stata maggiore.

Per questo motivo anche il fitting, effettuato con una funzione di densità di probabilità di tipo Gaussiano, non è così ben descrittivo dell'andamento dell'istogramma come per gli altri due grafici.

Tuttavia, in questa fase è stato deciso di mantenere questo tipo di assunzione in quanto non c'era, tra le principali altre curve con le quali effettuare il fitting dell'istogramma, una funzione che fornisse una corrispondenza più convincente.

Le considerazioni descritte per i primi due test possono dunque considerarsi estese anche al terzo, sulla base delle precisazioni fatte.

A seguito delle conclusioni effettuate è stato deciso di creare i grafici corrispondenti a quelli appena riportati nei quali però non fossero riportate le curve approssimate ottenute con il *fitting*, ma le gaussiane reali.

Pertanto per ognuno dei tre vettori di misure relativi ad ogni test, sono state calcolate la media μ e la deviazione standard σ . Per la generazione della curva gaussiana corrispondente è stata utilizzata la funzione *normpdf* di Matlab che implementa la seguente funzione:

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Di seguito sono riportati i grafici ottenuti secondo la procedura appena descritta. I dati a cui i grafici fanno riferimento sono presi dagli stessi test esaminati in precedenza.

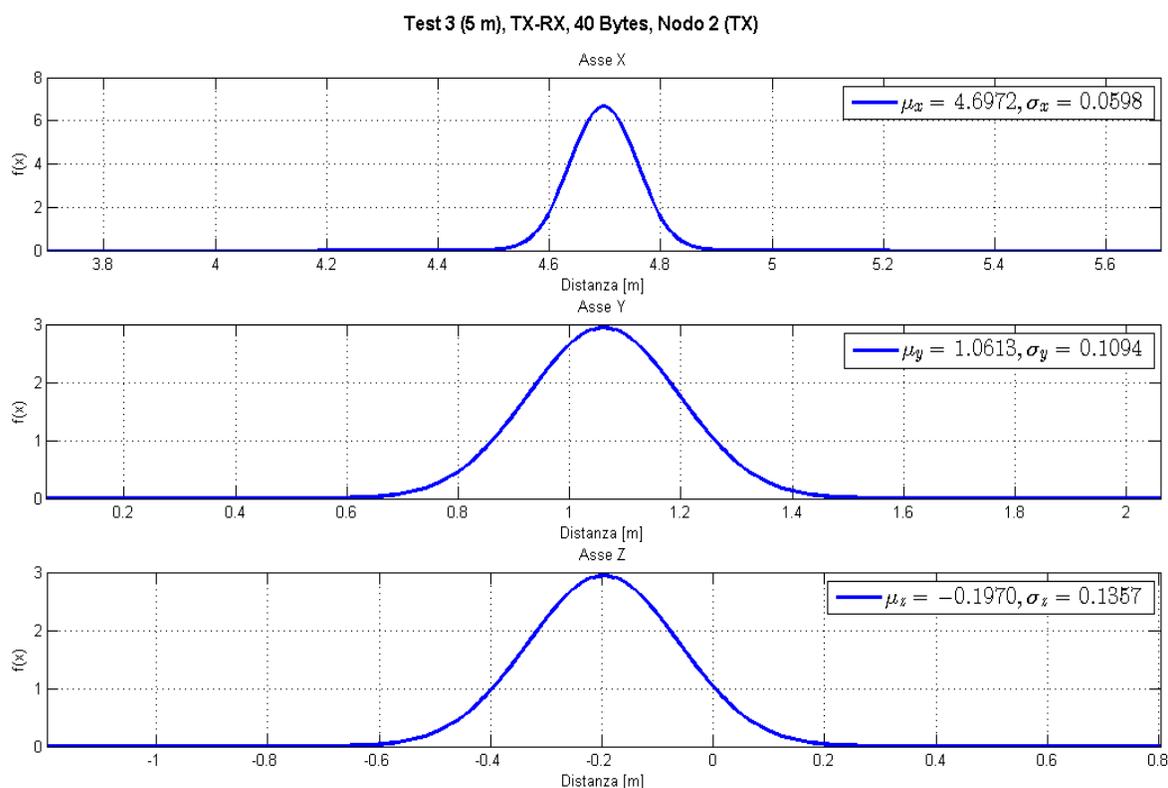


Figura 21: Funzione di densità di probabilità del Test 3, lungo le tre coordinate

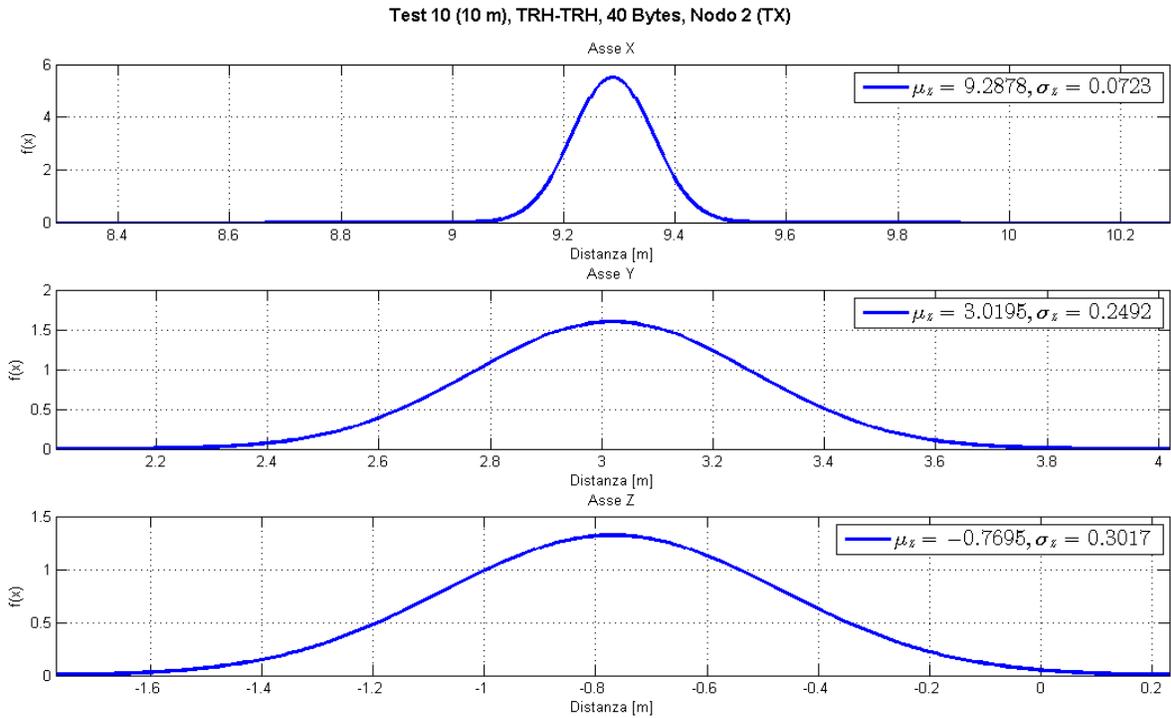


Figura 22: Funzione di densità di probabilità del Test 10, lungo le tre coordinate

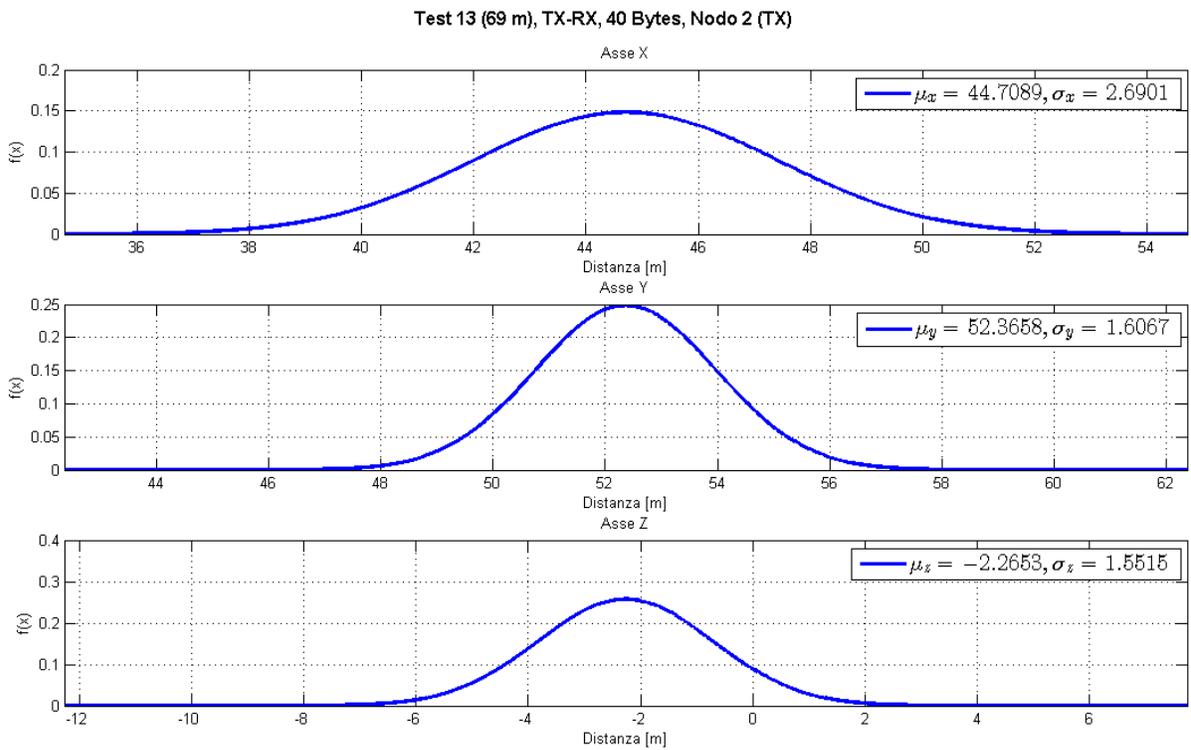


Figura 23: Funzione di densità di probabilità del Test 13, lungo le tre coordinate

Osservando dunque i primi due grafici è possibile notare come la deviazione standard delle misure lungo l'asse x risulti sempre di un ordine di grandezza inferiore rispetto a quelle sugli altri assi.

Per poter valutare al meglio questo tipo di fenomeno, sono stati riportati nella seguente tabella, i valori delle deviazioni standard delle misure lungo i tre assi, riferiti ad un numero maggiore di test effettuati con i dispositivi a terra.

#Test / σ	σ_x	σ_y	σ_z
Test 1	0.0412	0.1161	0.1361
Test 3	0.0598	0.1094	0.1357
Test 7	0.0711	0.1686	0.1817
Test 8	0.0813	0.2605	0.2856
Test 10	0.0723	0.2492	0.3017

Dall'osservazione della tabella emerge che il fenomeno osservato per i primi due grafici è riscontrabile anche negli altri test.

Pertanto è possibile affermare che il dispositivo fornisce misure meno disperse intorno al valor medio, e quindi più precise, lungo l'asse x piuttosto che lungo gli altri due assi.

Questo tipo di comportamento potrebbe essere sfruttato per ottimizzare l'orientamento dei device, una volta installati sui ROV, qualora ci fosse la convenienza di avere una precisione maggiore delle misure lungo una direzione preferenziale di moto.

I valori di deviazione standard riportati nella tabella precedente risultano essere soddisfacenti e pertanto le misure possono essere ritenute precise e caratterizzate da una dispersione limitata.

Per valutare in maniera globale la precisione della misura di posizione è stato calcolato, istante per istante, il modulo delle tre componenti ed è stato graficato su una scala temporale, dopodiché anche in questo caso si è proceduto alla creazione di una curva gaussiana secondo le modalità già utilizzate in precedenza.

I grafici ottenuti secondo quanto appena descritto sono riportati di seguito.

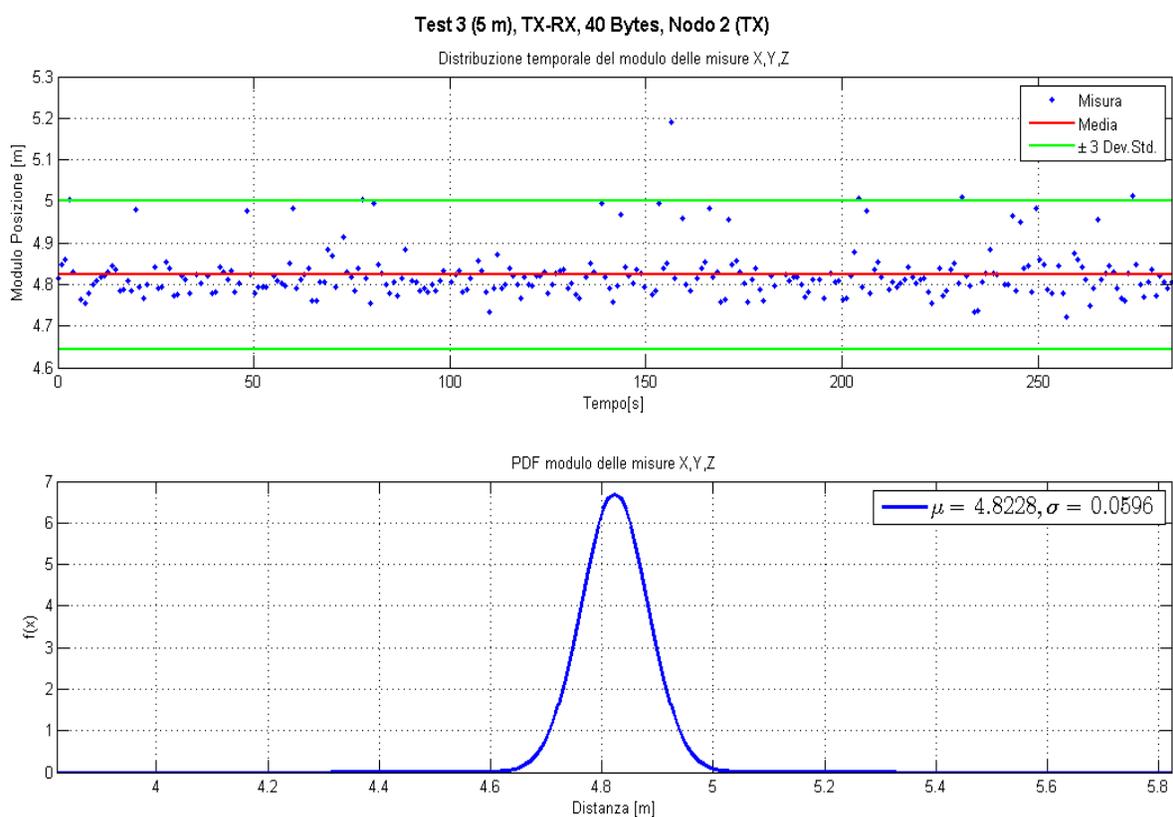


Figura 24: Distribuzione temporale e densità di probabilità del modulo delle misure del Test 3

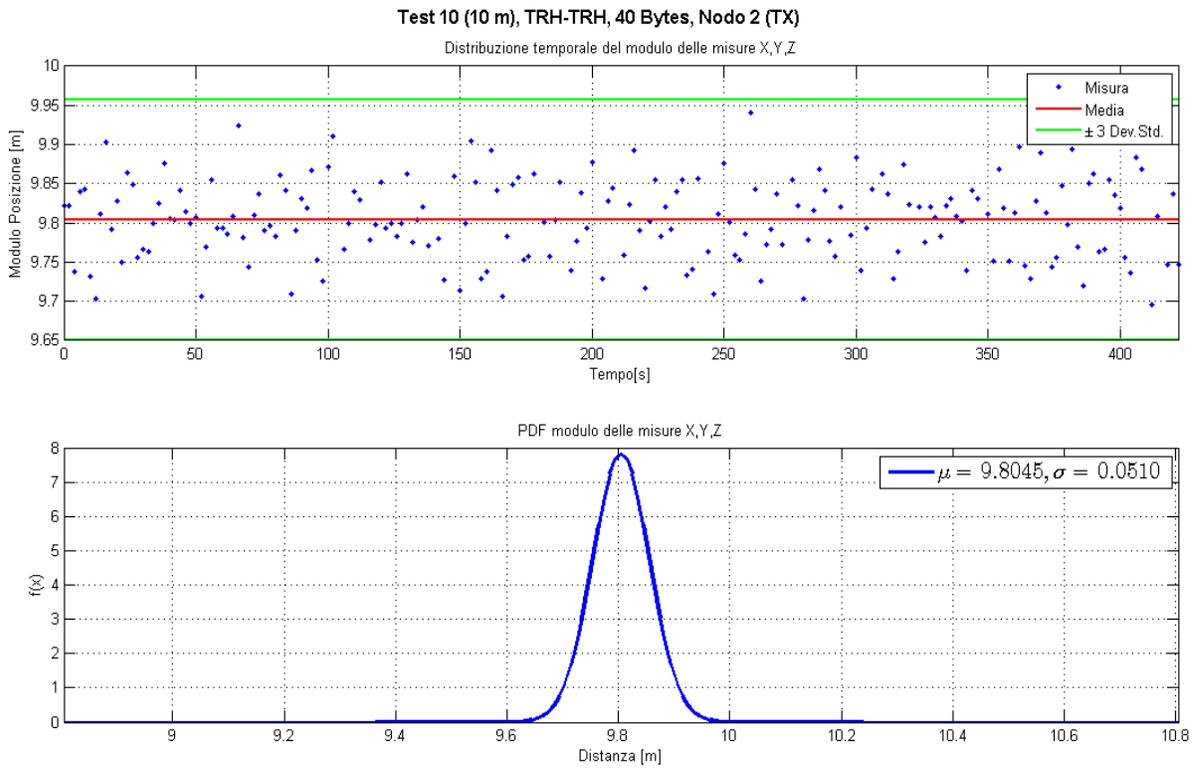


Figura 25: Distribuzione temporale e densità di probabilità del modulo delle misure del Test 10

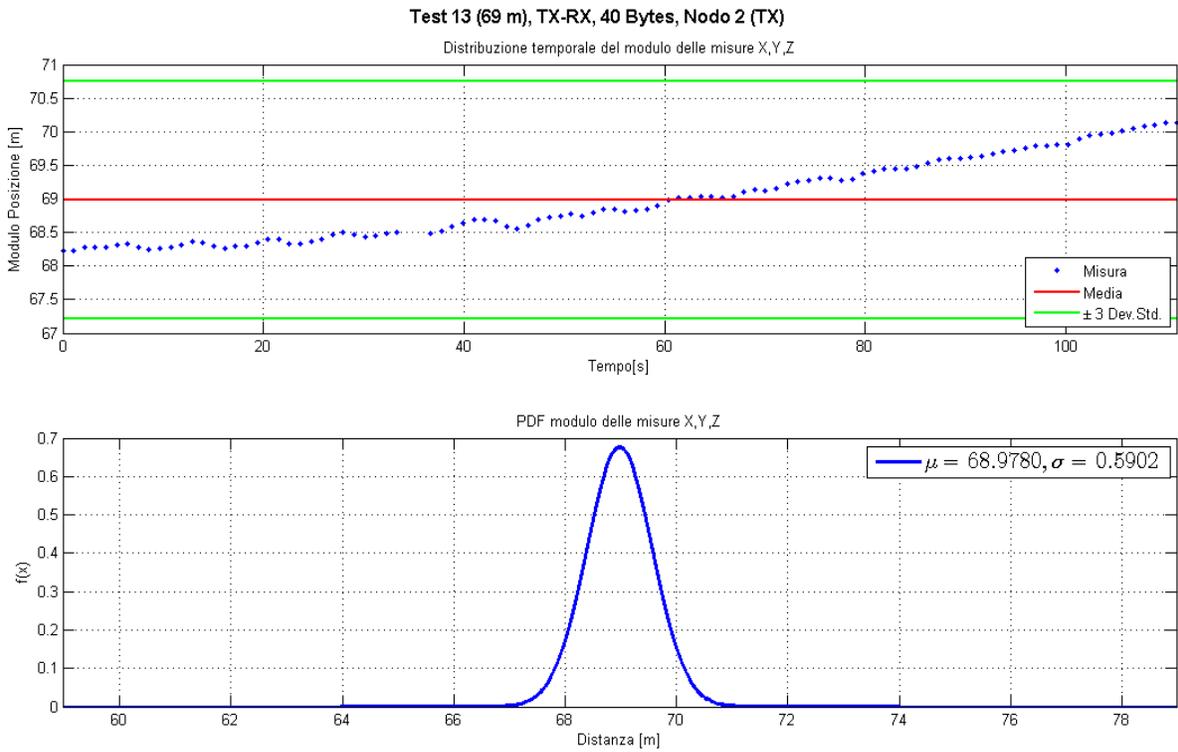


Figura 26: Distribuzione temporale e densità di probabilità del modulo delle misure del Test 13

Partiamo dall'osservazione dei primi due grafici temporali del modulo delle misure per dire che anche in questo caso il numero di outlier è contenuto in poche unità, addirittura pari a zero nel secondo.

Questo a conferma di quanto già detto in precedenza relativamente alla coerenza delle misure fornite dai device.

Passando all'osservazione delle gaussiane si vede che presentano una media molto vicina ai valori di riferimento misurati a terra e pertanto, in prima approssimazione, anche l'accuratezza della misura è da ritenersi soddisfacente, anche in virtù del fatto che i dispositivi erano sospesi con delle cime in acqua e pertanto erano liberi di muoversi.

Anche la precisione della misura è da ritenersi soddisfacente in quanto è presente una ridotta dispersione intorno al valor medio.

E' possibile osservare come i due valori di deviazione standard del modulo delle misure siano pienamente confrontabili tra i due test analizzati.

Pertanto si può affermare con soddisfazione come la precisione delle misure non sia influenzata, in prima approssimazione, né dalla distanza intercorrente dai device né dalla modalità di comunicazione utilizzata.

E' possibile inoltre osservare come il discostamento della media dei moduli delle misure dal valore "vero" rimanga invariato tra i due test.

A conferma di quanto appena osservato in riferimento alle σ dei moduli delle misure, di seguito è riportata una tabella contenente i valori riferiti anche agli altri test effettuati con i dispositivi a terra.

#Test / σ	σ
Test 1	0.0377
Test 3	0.0596
Test 7	0.0727
Test 8	0.0502
Test 10	0.0510

Per quanto riguarda invece il terzo test, dall'osservazione del grafico del modulo delle misure temporali si nota come la distanza tra i modem sia variata durante la prova a causa dello spostamento della barca.

Al termine di questa analisi si può concludere che la precisione delle misure di posizione fornite dal sistema è da ritenersi soddisfacente sia da un punto di vista prettamente metrico, sia in relazione all'assenza di una dipendenza diretta di quest'ultima dalla distanza intercorrente tra i device, dalla lunghezza del messaggio trasmesso e dalla modalità di comunicazione utilizzata.

4.2 Analisi di consistenza dei dati

A seguito dell'analisi di precisione è stato deciso di svolgere una analisi riguardante la consistenza dei dati.

Per validare questo tipo di proprietà sono stati utilizzati due approcci:

- ✓ Il confronto dei moduli delle misure dei due nodi
- ✓ La stima di una matrice di orientazione del modem

Per effettuare l'analisi di consistenza dei dati con il primo dei due metodi, l'attenzione è stata posta sulla modalità di comunicazione TRH – TRH.

In questa modalità, infatti, ciascun device funziona alternativamente con l'altro sia come ricevitore che come trasmettitore.

Come già sottolineato in precedenza, ad ogni messaggio trasmesso che viene rilevato dal ricevitore, il modem ricevente genera la notifica "USBLLONG", contenente il fix di posizione. Si viene così a creare una simmetria di funzionamento grazie alla quale ciascuno dei due dispositivi fornisce una serie di misure di posizione.

I test scelti per questo confronto sono i numeri 7, 10 e 15. Questi test, tutti riferiti, come già, detto a modalità di comunicazione a divisione di tempo, differiscono tra loro per la distanza intercorrente tra i device.

I dati del test 7 sono stati acquisiti con i modem ad una distanza di 5 m, il test 10 a 10 m ed il test 15 ad una distanza variabile poiché uno dei due modem era installato a bordo dell'imbarcazione.

Nei grafici successivi si è scelto di riportare le due serie temporali rappresentandole in colori diversi (blu e rosa), mentre in rosso è riportato il valor medio delle misure e in verde l'intervallo a $\pm 3\sigma$ come già fatto in precedenza.

Nel secondo grafico, per ciascun test, è riportato invece l'andamento dell'errore inteso come differenza tra le due serie temporali.

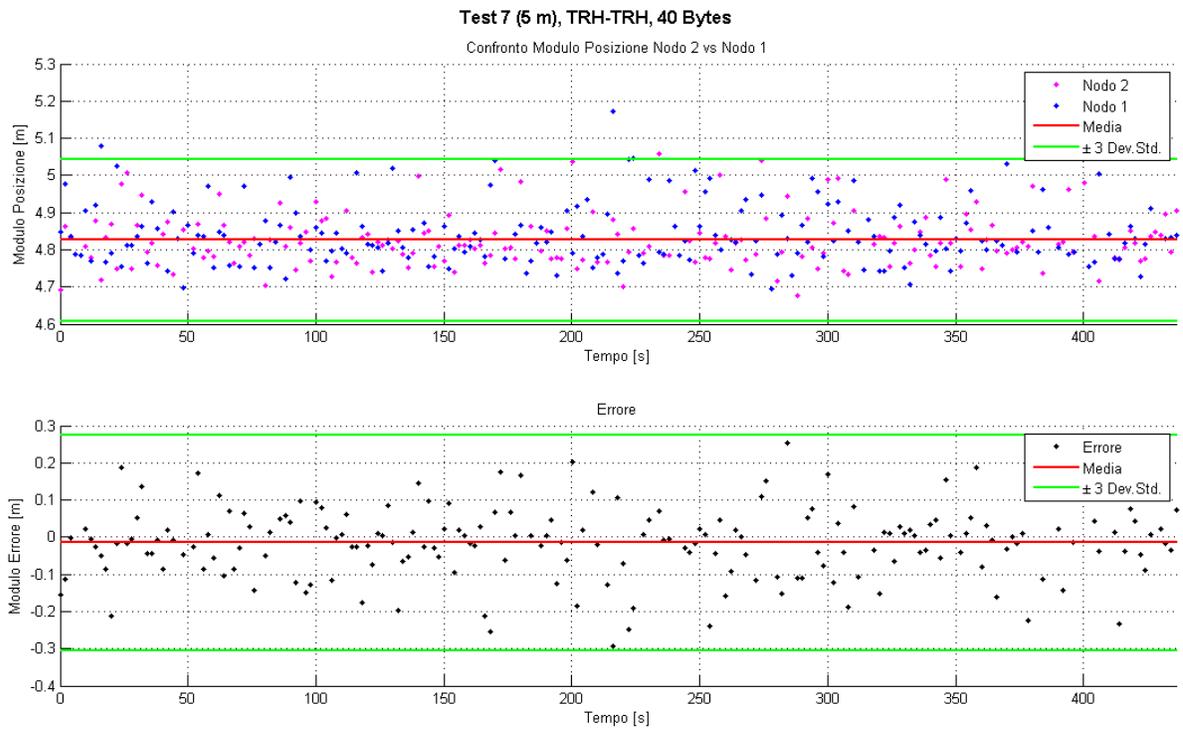


Figura 27: Confronto modulo misure di posizione dei due nodi ed errore, Test 7

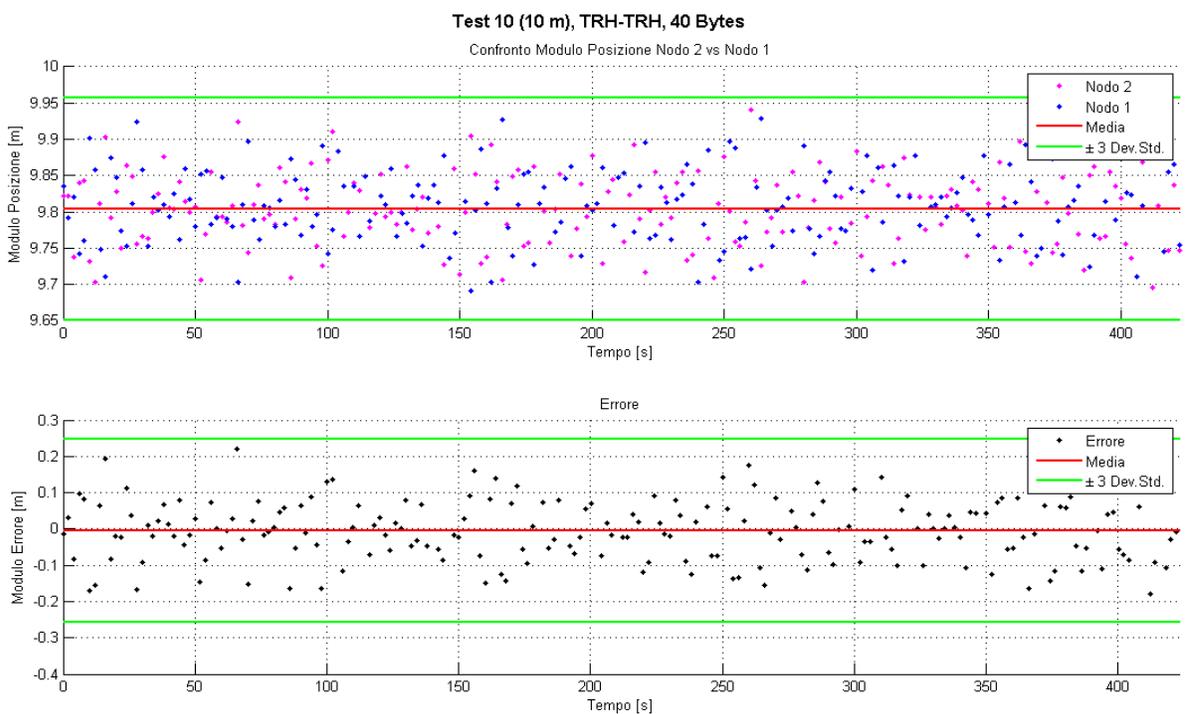


Figura 28: Confronto modulo misure di posizione dei due nodi ed errore, Test 10

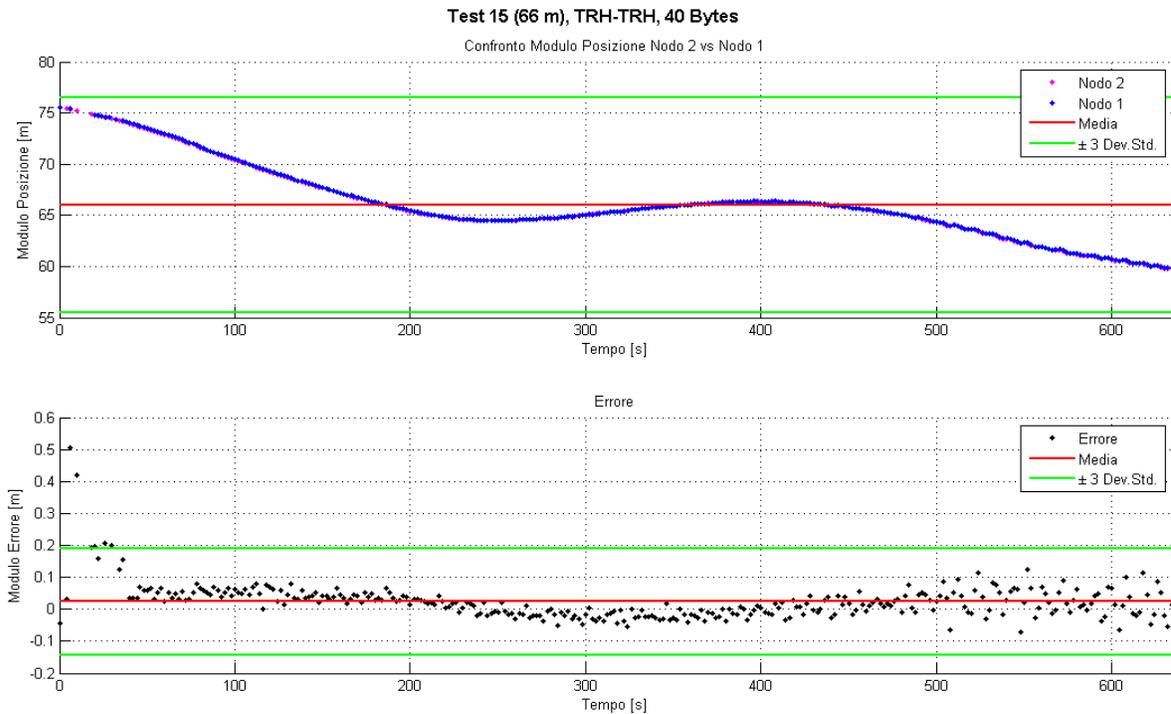


Figura 29: Confronto modulo misure di posizione dei due nodi ed errore, Test 15

Dall'osservazione dei primi due grafici si nota che le due serie sono pienamente confrontabili e che presentano valori di media e deviazione standard praticamente identici (nel grafico sono riportate soltanto le rette relative alla serie del nodo due per renderlo più leggibile).

Gli outlier anche in questo caso sono limitati a poche unità e tutti i valori sono distribuiti in modo omogeneo intorno al valor medio.

Ancor più significativo è il grafico dell'errore che non presenta outlier, ha una distribuzione omogenea intorno al valor medio che risulta essere praticamente pari a zero, e presenta una deviazione standard molto piccola.

Un commento a parte, merita come sempre, il test riferito al caso in cui uno dei due modem si trovava a bordo dell'imbarcazione.

A differenza del caso precedente, nel primo grafico è possibile apprezzare la variazione nel tempo del modulo delle misure delle distanze riportate dai due nodi.

La curva così descritta rappresenta lo scarroccio subito dall'imbarcazione la quale aveva i motori spenti durante il test.

L'aspetto più importante di questo grafico è rappresentato dalla perfetta corrispondenza delle misure rilevate dai due nodi.

Dal grafico dell'errore, analogamente al precedente, è possibile apprezzare come il valor medio sia molto prossimo allo zero e la deviazione standard anche in questo caso sia molto piccola.

Dai grafici elaborati si evince come le serie temporali dei moduli delle misure fornite dai due device siano consistenti e confrontabili tra loro con una approssimazione molto bassa.

I grafici precedenti esprimono dunque con evidenza la consistenza delle misure di distanza fornite dai dispositivi attraverso l'utilizzo del primo dei due metodi scelti.

Si ricorda che a garanzia del fatto che le due serie temporali riportate per ciascuno dei grafici precedenti siano effettivamente sincronizzate, e quindi ci sia corrispondenza tra i due campioni forniti da ciascuno dei device, i pc collegati ai modem sono stati preventivamente sincronizzati attraverso l'interfaccia ethernet e in riferimento ad un timer comune.

Il secondo metodo utilizzato per la validazione della consistenza dei dati disponibili è stato quello della stima di una matrice di orientazione planare del modem, da utilizzare per il tracciamento della posizione di un dispositivo rispetto all'altro.

Si precisa che l'intento di questa metodologia non è stato quello di risolvere un problema di tracciamento vero e proprio di una posizione o di una traiettoria, ma solo quello di utilizzare un feedback visivo e grafico della misura di posizione fornita, istante per istante da uno dei due dispositivi rispetto all'altro.

La configurazione iniziale sulla quale tutta la procedura si basa è illustrata nell'immagine seguente.

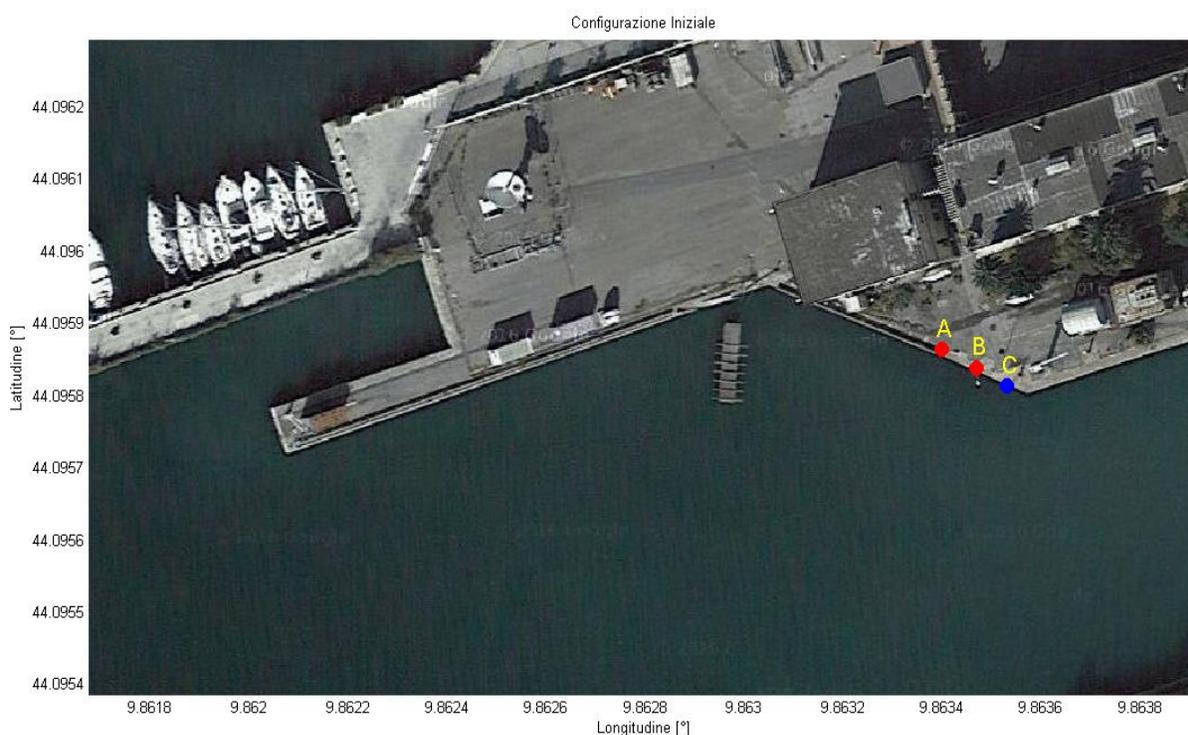


Figura 30: Disposizione modem all'interno dell'area prove

L'immagine satellitare mostra una panoramica dell'area di svolgimento delle prove effettuate. Con A, B e C sono stati indicati i punti nei quali sono stati posizionati i modem durante i test nell'arco della giornata.

Dei tre punti riportati sul grafico sono state acquisite sul posto anche le coordinate GPS (esclusa l'altezza) che sono riportate di seguito.

Punto/ GPS	Latitudine	Longitudine
A	44.095864°	9.863400°
B	44.095839°	9.863471°
C	44.095814°	9.863531°

Si ricorda che il fix di posizione fornito dal device è espresso in un sistema di riferimento solidale ad esso secondo quanto riportato sul manuale del prodotto.

Per la stima della matrice di orientazione del dispositivo sono stati scelti i dati relativi al test 8. Durante questo test il Nodo 2 è posizionato nel punto C e il Nodo 1 nel punto A. I dati che sono stati scelti sono quelli relativi alla localizzazione che il Nodo 2 compie rispetto al Nodo 1. Pertanto il sistema di riferimento in base al quale vengono fornite le misure dell'USBL è centrato nel punto C. Di seguito è riportato lo schema della terna solidale al modem.

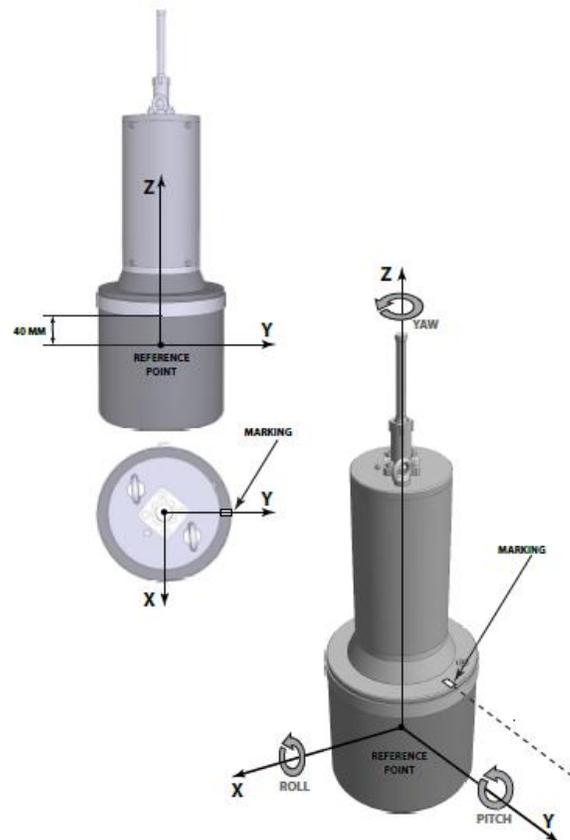


Figura 31: Sistema di riferimento solidale al modem

Come descritto in precedenza, il posizionamento dei moduli in acqua è avvenuto attraverso l'ausilio di un mezzo marinaio e di una cima; pertanto i modem sono stati semplicemente calati poco al di sotto del livello del mare.

Questa configurazione non permetteva né di conoscere né di controllare l'orientazione planare del sistema di riferimento del modulo che risultava perciò ignota.

Ciò di cui si disponeva erano le misure relative di un device rispetto all'altro riferite in C, la distanza tra i due modem e la loro posizione in coordinate GPS.

Supponendo dunque l'orientazione del device costante, è stato deciso di impostare un problema di minimizzazione in Matlab scegliendo come funzionale di costo una relazione che permettesse di effettuare la trasformazione delle coordinate GPS in coordinate locali. Data quindi la seguente relazione:

$$\overline{(P_{AC})}^N = R(\Xi) \begin{matrix} N \\ D \end{matrix} \overline{(P_{AC})}^D$$

Dove:

- $\overline{(P_{AC})}^N$ rappresenta il vettore della posizione del punto A rispetto al punto C, espresso nel sistema di riferimento NED (*North East Down*), centrato nel punto C.
- $\overline{(P_{AC})}^D$ rappresenta il vettore della posizione del punto A rispetto al punto C, espresso nel sistema di riferimento "Device" (vedi figura precedente), centrato nel punto C.
- $R(\Xi) \begin{matrix} N \\ D \end{matrix}$ rappresenta la matrice di trasformazione delle coordinate da "Device" a NED.
- $\Xi = \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}$, con ϕ, θ, ψ , gli angoli di roll, pitch e yaw.

Ciò di cui non si dispone è il vettore $\overline{(P_{AC})}^N$ ma quest'ultimo può essere calcolato effettuando una trasformazione da coordinate GPS (latitudine, longitudine, altezza) a coordinate *NED*, in quanto sono note le coordinate GPS dei punti A e C.

E' stato scelto di risolvere un problema di orientazione di tipo planare, pertanto delle tre coordinate caratterizzanti il vettore $\overline{(P_{AC})}^D$ sono stati posti a zero tutti i valori relativi all'asse *z*.

Per la creazione del vettore $\overline{(P_{AC})}^N$ invece, sono state prese le coordinate GPS del punto A e del punto B e sono state convertite in *NED* attraverso due conversioni successive. La prima conversione ha permesso di esprimere i valori in *ECEF* (*Earth-Centered, Earth-Fixed*) attraverso la funzione *lla2ecef*.

La seconda conversione, effettuata attraverso la funzione *ecef2ned*, ha permesso di disporre dei dati espressi nel sistema di riferimento *NED*.

A questo punto, affinché la relazione precedente avesse dimensioni coerenti, è stato necessario replicare le misure ottenute creando un vettore di lunghezza pari a $\overline{(P_{AC})}^D$.

Esprimendo dunque la precedente relazione nel seguente modo:

$$\bar{K} = 0 = R(\Xi)_D^N \overline{(P_{AC})}^D - \overline{(P_{AC})}^N$$

è stato scelto di rendere minima questa differenza attraverso la minimizzazione del parametro \bar{K} , funzione degli angoli di roll, pitch e yaw, su ogni misura disponibile.

Il funzionale di costo da minimizzare ha assunto dunque la seguente forma:

$$f = \sum_{i=0}^{\# \text{mis}} \| K_i \|$$

Per minimizzare il funzionale di costo è stata utilizzata la funzione *fminunc* di Matlab utilizzando come vettore delle condizioni iniziali il seguente:

$$x = \begin{pmatrix} \pi \\ 0 \\ 0 \end{pmatrix}$$

Questa scelta è stata fatta dopo alcuni tentativi, tra cui quello di partire da un vettore delle condizioni iniziali tutto nullo, in quanto il funzionale scelto non presenta un solo minimo assoluto, ma più di un minimo locale.

Pertanto è possibile che scelte diverse riguardo alle condizioni iniziali portino a risultati finali formalmente diversi.

Nel nostro caso la scelta di partire dal vettore x , piuttosto che da quello nullo, è motivata esclusivamente dal fatto di avere come risultato una matrice di rotazione planare con una convenzione sull'angolo coerente con quella scelta.

Al termine della procedura i valori degli angoli risultano:

$$\Xi = \begin{pmatrix} \pi \\ 0 \\ -0.7780 \end{pmatrix}$$

ai quali è associata la seguente matrice di rotazione:

$$R = \begin{pmatrix} 0.7123 & -0.7018 & 0 \\ -0.7018 & -0.7123 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

La presente matrice descrive una rotazione da “*Device*” a *NED* di un angolo di circa $-44,58^\circ$. Ricordiamo che questa matrice è stata assunta costante all’inizio della procedura di calcolo, supponendo quindi costante l’orientazione dei dispositivi immersi in acqua.

Per validare la bontà di questa assunzione è stato elaborato un grafico nel quale sono stati riportati i valori delle misure effettuate dai dispositivi al variare della distanza intercorrente tra gli stessi.

Sono stati scelti pertanto i test 3,8,13,14,15. I test 3 e 8, come già osservato in precedenza, si riferiscono a misure effettuate con entrambi i dispositivi a terra ad una distanza rispettivamente di 5 e di 10 metri, dove il dispositivo “target” era situato nel punto A.

I test 13,14 e 15 invece si riferiscono alle prove effettuate nel pomeriggio della giornata di test, quando uno dei due dispositivi si trovava a bordo dell’imbarcazione messa a disposizione.

Il risultato di questa visualizzazione è riportato nell’immagine seguente. Si ricorda che questo grafico non rappresenta un tracking vero e proprio dell’imbarcazione nello specchio d’acqua antistante alle banchine del CSSN, ma è stato creato per validare visivamente la bontà della stima della matrice di orientazione del device.

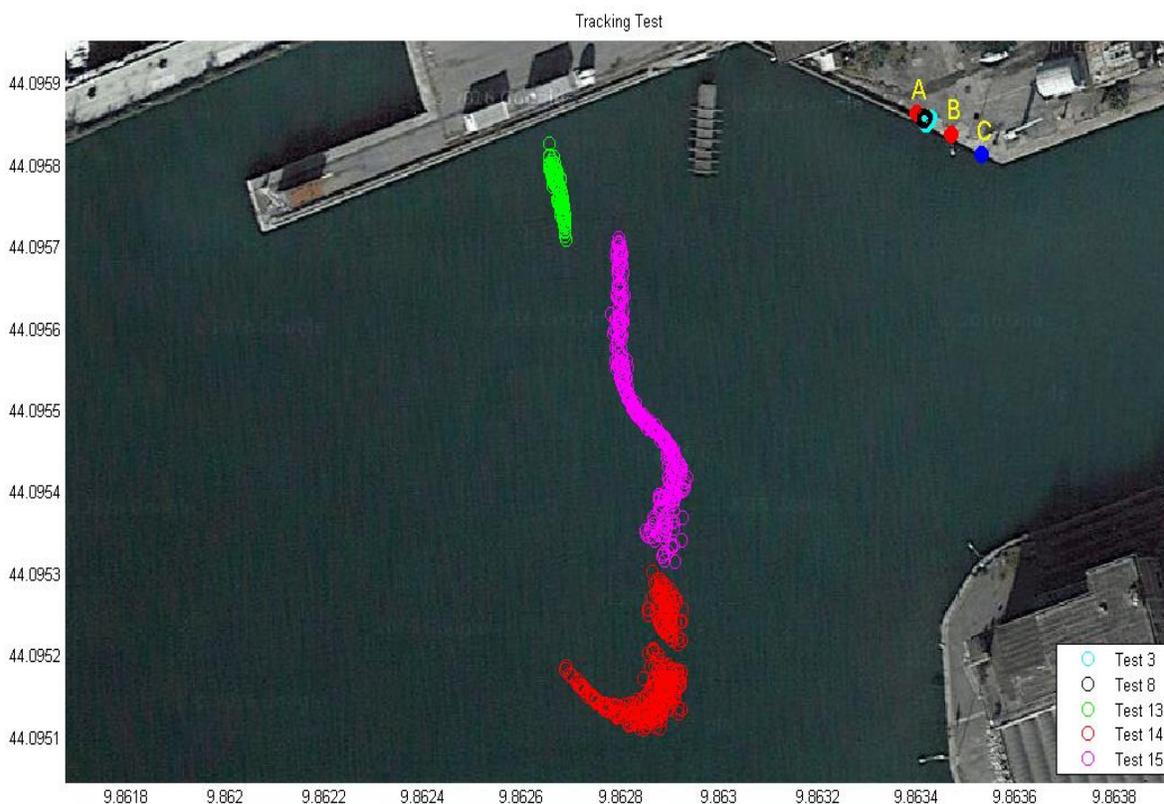


Figura 32: Validazione matrice di orientazione attraverso il tracking dei test 3, 8, 13, 14 e 15

Dall'analisi dell'immagine precedente si osserva che i marker di colore azzurro e nero relativi rispettivamente ai test 3 e 8, sono plottati correttamente in prossimità del punto A, con un certo errore in termini di distanza, ma con una orientazione pienamente soddisfacente, sulla congiungente dei punti A e B.

I marker di colore verde, rosso e magenta fanno invece riferimento rispettivamente ai test 13, 14 e 15, e la loro visualizzazione sulla mappa è coerente con gli spostamenti effettuati dall'imbarcazione.

La stessa, infatti, all'inizio del test 13, si trovava ad una certa distanza dalla banchina e stazionava in quel punto a motori spenti. L'assenza di propulsione ha fatto sì che la barca risentisse della corrente e subisse uno scarroccio verso la banchina.

Pertanto, il comandante dell'imbarcazione ha deciso di riposizionarsi più a largo.

Una volta raggiunta la nuova posizione i motori sono stati nuovamente spenti ed è partita l'acquisizione dei dati relativi al test 14.

La scia di marker rossi, relativa a questo test, risulta interrotta in un certo punto in quanto, durante il test, il device è stato sollevato dall'acqua e riposizionato sull'altro lato dell'imbarcazione.

E' comunque visibile anche in questo caso lo scarroccio subito dall'imbarcazione.

Considerazioni analoghe possono essere fatte per il test 15, contraddistinto dai marker di color magenta.

Pertanto, sulla base dell'osservazione del precedente grafico e del confronto dello stesso con gli spostamenti compiuti dall'imbarcazione, documentato con materiale fotografico, è possibile affermare che la procedura di stima della matrice di orientazione R ha avuto il successo sperato e che l'ipotesi iniziale di orientazione costante nel tempo del device immerso in acqua è pienamente attendibile.

4.3 Analisi delle prestazioni

In questo paragrafo saranno analizzate le prestazioni della comunicazione acustica implementata tra i dispositivi attraverso lo studio della dinamica dei principali parametri di performance disponibili per i modem Evo Logics utilizzati.

I parametri oggetto di questa analisi sono già stati citati nel capitolo due come campi informativi contenuti all'interno dei messaggi di notifica dei dispositivi, e sono:

- ✓ RSSI (*Received Signal Strength Indicator*)
- ✓ Integrity
- ✓ Accuracy

Per ciascuno di questi parametri è stata effettuata una analisi di dipendenza della dinamica del parametro stesso in funzione della distanza intercorrente tra i dispositivi.

Per raggiungere questo obiettivo, ed avere una caratterizzazione della dinamica del parametro più precisa possibile con i dati a disposizione, l'analisi è stata svolta sul maggior numero di test disponibili svolti a distanze diverse.

Per questo motivo sono stati selezionati i test 3, 8, 13, 14, 15. Si ricorda ancora una volta che mentre i primi due test si riferiscono a condizioni operative con i dispositivi a terra, per i restanti tre, i dati fanno riferimento a test effettuati con un dispositivo a bordo dell'imbarcazione.

E' stato già analizzato all'inizio di questo capitolo, quando sono stati commentati i grafici riguardanti la distribuzione delle misure dei test pomeridiani, quali fossero le limitazioni dell'assumere tali distribuzioni di tipo Gaussiano.

Pertanto, per ovviare a questo tipo di limitazione, e poter confrontare tutte le misure in maniera omogenea, è stato deciso di creare per i test 13, 14 e 15 dei cluster di dati, ovvero

dei sottoinsiemi di dati acquisiti ad istanti consecutivi, e quindi in intervalli di tempo per i quali la posizione dell'imbarcazione possa essere ritenuta costante.

Per la creazione di questi cluster la procedura seguita è stata la seguente: in riferimento alla tabella delle deviazioni standard dei moduli delle misure dei test effettuati con entrambi i device a terra, riportata nel paragrafo 4.1, è stato osservato che il valor medio di questo parametro, tra tutti i casi analizzati, era circa uguale a 0.06 m.

Poiché per la visualizzazione grafica dei dati è stata scelta una finestra di $\pm 3\sigma$ intorno al valor medio, all'interno della quale rientravano la quasi totalità dei campioni, è stato deciso di mantenere questo intervallo di dispersione anche per i cluster che si andavano a creare.

Pertanto la dispersione intorno al valor medio dei cluster è stata scelta a priori pari a $0.06 \times 6 \approx 0.4$ m, per quanto appena detto.

Successivamente, fissato l'intervallo di dispersione, è stato individuato, per ogni cluster, il valor medio intorno al quale, con quell'intervallo, si potesse disporre del maggior numero di campioni possibile per massimizzare la precisione del calcolo della media e della deviazione standard del cluster di dati.

Questa procedura ha portato alla scelta dei seguenti valori.

Cluster	Valor medio	#Campioni
13	68,4 m	37
14	97,4 m	27
15	66,2 m	44

Per i tre cluster così creati, è adesso corretto approssimare la distribuzione dei dati che li costituiscono con una curva di densità di probabilità di tipo Gaussiano.

Poiché a questo punto il tipo di funzione di densità di probabilità dei dati è la stessa per tutti e cinque i test, è coerente, per ciascuno dei tre parametri di performance, effettuare i confronti.

Il primo parametro esaminato, ovvero l'RSSI, è costituito da un valore intero negativo che indica il livello in dB del segnale ricevuto. A segnali più forti sono associati valori di RSSI più alti.

Nel nostro caso il produttore dichiara che livelli accettabili di RSSI sono compresi tra i -20 dB e i -85 dB.

Il valore di questo parametro è contenuto all'interno del payload della notifica RECVIM (*Receive Instant Messages*) introdotta nel capitolo due e generata automaticamente dal device al ricevimento di un messaggio istantaneo.

Poiché, per quanto detto in precedenza, il protocollo di comunicazione implementato, è basato sull'invio e sulla ricezione di questa tipologia di messaggi, l'analisi che segue è stata effettuata su una grande quantità di dati acquisiti durante le prove.

La prima operazione che è stata eseguita è stata appunto quella di confrontare tra loro le funzioni di densità di probabilità ottenute per i vari test.

Quindi, analogamente a quanto fatto in precedenza, per ognuno dei cinque test sono state calcolate la media e la deviazione standard dei vettori contenenti tutti i valori di RSSI acquisiti ad ogni notifica, e, con l'ausilio della funzione Matlab *normpdf* sono state plottate le varie curve Gaussianhe ottenute.

Il risultato è riportato nella figura seguente, nella quale è possibile osservare come all'aumentare della distanza intercorrente tra i dispositivi ci sia una degradazione della performance in termini di RSSI.

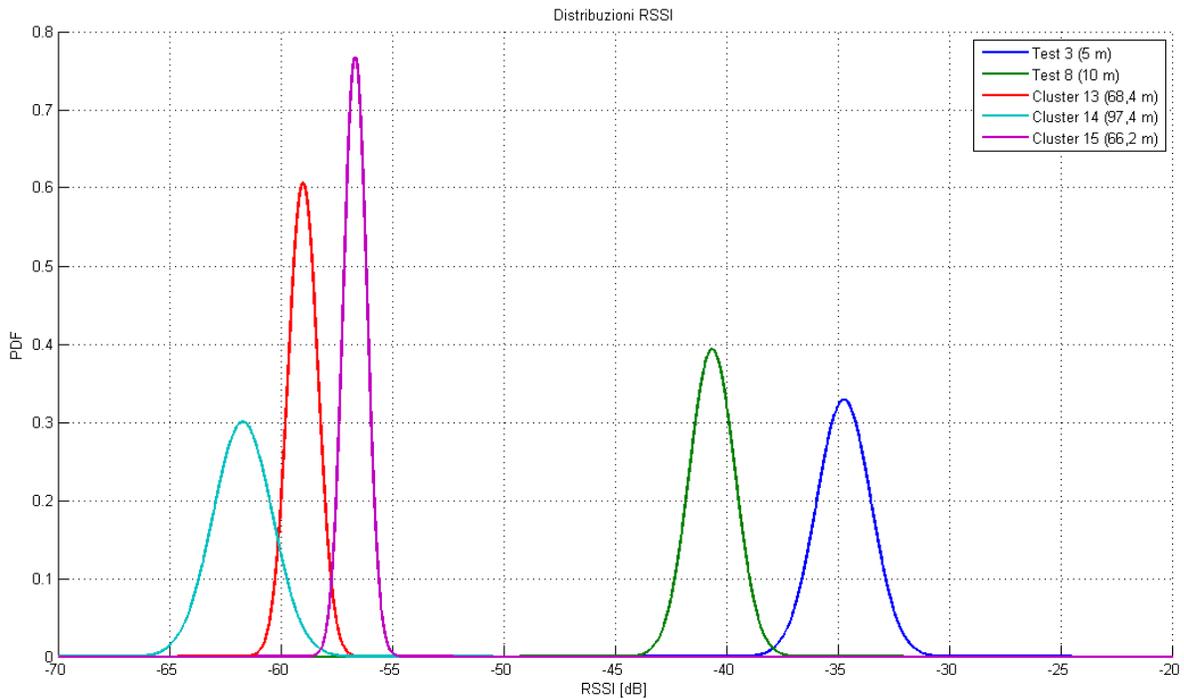


Figura 33: Funzioni di densità di probabilità del parametro RSSI per i test 3, 8 e per i cluster 13, 14 e 15

Questo tipo di comportamento del resto ci si poteva aspettare in quanto, durante lo svolgimento di tutti i test, il Source Level non è stato modificato.

Pertanto, essendo rimasto invariato il livello di potenza della sorgente, è evidente che ad una distanza maggiore il segnale arrivi più attenuato.

Tuttavia, anche per il test 14, che fornisce valori riferiti ad una distanza operativa di quasi cento metri, il valore dell’RSSI rimane al di sopra del limite minimo consigliato dal produttore.

E’ evidente quindi osservare che se anche per i test 13, 14 e 15 desiderassimo delle performance di RSSI paragonabili a quelle ottenute per i test 3 e 8, e quindi con valori compresi tra i -35 e i -40 dB, sarebbe sufficiente aumentare il valore del Source Level.

A questo punto è stato osservato che sarebbe stato utile capire quale fosse la relazione che potesse descrivere la dipendenza della degradazione della performance di questo parametro in funzione della distanza operativa intercorrente tra i due device.

Pertanto è stato elaborato un grafico nel quale sono stati riportati i valori delle medie dei vettori di RSSI dei test selezionati in funzione della distanza alla quale ciascun test è stato effettuato.

Successivamente, attraverso l'utilizzo del Matlab Toolbox *cftool*, è stata scelta la curva di fitting che meglio approssimava l'andamento dei valori al variare della distanza.

La curva scelta è di tipo *power*, ed è caratterizzata dalla seguente forma generale:

$$f(x) = a x^b \quad a = -25,89 \quad b = 0,1906$$

dove a e b sono valori reali costanti e x è la variabile.

La scelta di questa curva, tra tutte quelle disponibili, è avvenuta sulla base dell'errore di approssimazione commesso dall'operazione di fitting, utilizzando come parametro di performance l'RMSE (*Root Mean Square Error*), che in questo caso risulta pari a 0.91.

L'andamento risultante è riportato nella figura seguente.

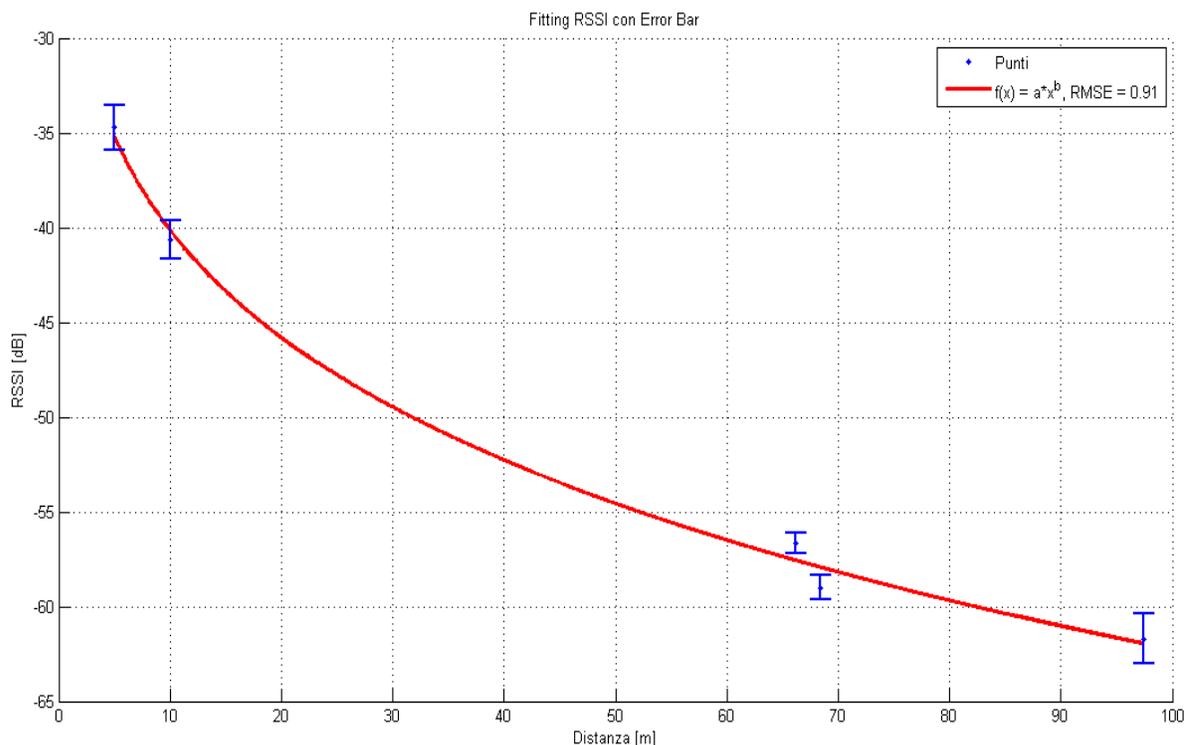


Figura 34: Curva di fitting con error bar per il parametro RSSI

I segmenti blu verticali sono gli “*error bar*” e rappresentano gli intervalli di dispersione della media intorno al suo valore. Questi segmenti quindi aiutano a comprendere e a visualizzare quale sia la bontà del fitting.

Con il grafico precedente è possibile dunque affermare di disporre di una caratterizzazione della dinamica del parametro RSSI in funzione della distanza operativa intercorrente tra i device.

Il secondo parametro di performance analizzato, l’*Integrity*, è costituito da un numero intero positivo e descrive il livello di distorsione del segnale acustico ricevuto.

A valori più elevati di Integrity sono associati livelli di distorsione più bassi. Secondo le specifiche riportate dal produttore, una connessione acustica è considerata debole quando presenta valori di Integrity inferiori a 100.

Il valore di questo parametro è contenuto all’interno del payload della notifica USBLLONG introdotta nel capitolo due e generata automaticamente dal device all’avvenuta consegna di un messaggio istantaneo.

Poiché, per quanto detto in precedenza, il protocollo di comunicazione implementato, è basato sull’invio e sulla ricezione di questa tipologia di messaggi, l’analisi che segue è stata effettuata su una grande quantità di dati acquisiti durante le prove.

Anche in questo caso la prima operazione che è stata eseguita è stata quella di confrontare tra loro le funzioni di densità di probabilità ottenute per i vari test analogamente a quanto fatto per l’RSSI.

Il risultato è riportato nella figura seguente.

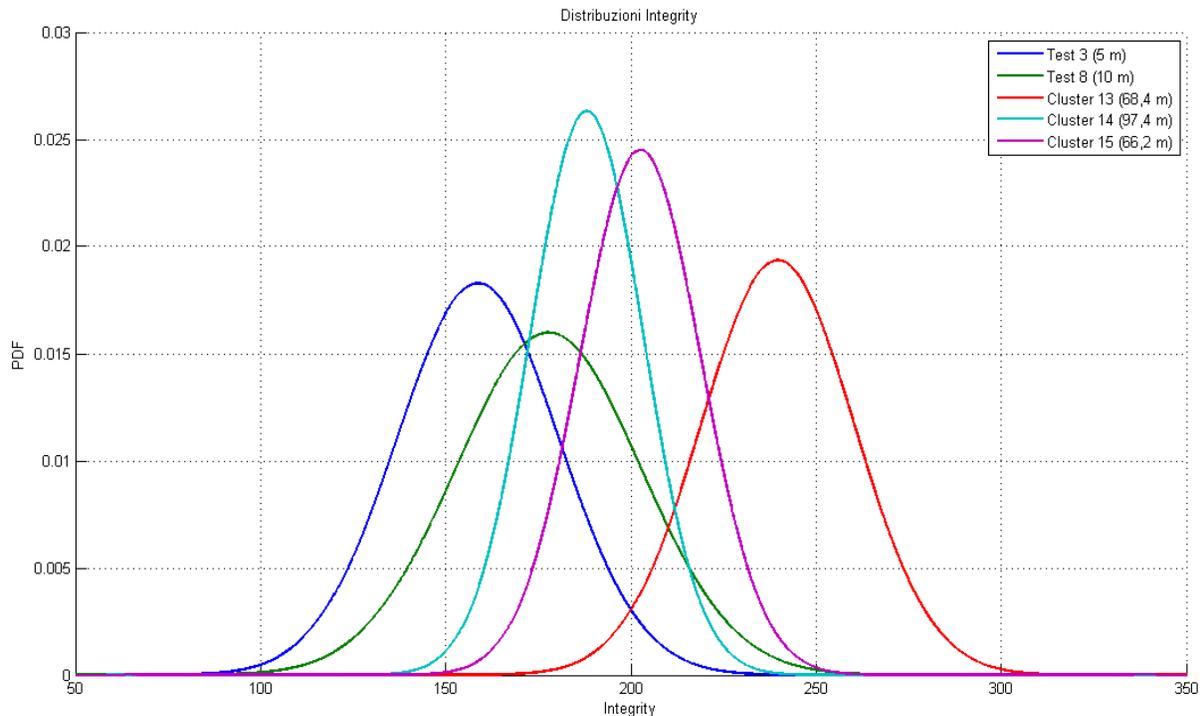


Figura 35: Funzioni di densità di probabilità del parametro Integrity per i test 3, 8 e per i cluster 13, 14 e 15

Dall'osservazione del grafico si evince che i risultati ottenuti non corrispondono a quelli sperati.

I valori più bassi di Integrity sono infatti associati ai test 3 e 8, svolti con i dispositivi ad una distanza rispettivamente di 5 e 10 metri, mentre i valori più elevati si riferiscono ai test svolti in barca ad una distanza maggiore.

Va ricordato che i test 3 e 8 sono stati svolti secondo le modalità descritte nel capitolo tre, ovvero in prossimità della banchine di cemento, in corrispondenza di un fondale di tipo roccioso profondo soltanto pochi metri, mentre i restanti test sono stati svolti a distanze molto più grandi, con uno dei modem posizionato in barca al di sotto della quale il fondale di tipo sabbioso raggiungeva profondità di oltre 12 metri.

Inoltre, i test a terra sono stati svolti la mattina mentre quelli in barca nel pomeriggio.

Le diverse condizioni di temperatura e salinità possono aver influito sul valore di questo parametro di performance della comunicazione acustica.

Queste, inoltre, sono solo alcune delle variabili che possono aver influito sulla distribuzione dei valori del parametro.

Per quanto detto, le funzioni di densità di probabilità, pur rimanendo ovviamente coerenti e corrette relativamente a ciascuno dei test, risultano essere non confrontabili tra loro.

Per questo parametro è stato scelto dunque di non elaborare il grafico di fitting in quanto sarebbe risultato non rappresentativo e poco attendibile.

Il terzo parametro di performance analizzato, l'*Accuracy*, è costituito da un numero espresso in radianti e descrive l'accuratezza del fix di posizione ricevuto.

Sono quindi preferibili livelli di *Accuracy* più bassi possibili.

Il valore di questo parametro è contenuto all'interno del payload della notifica USBLLONG introdotta nel capitolo due e generata automaticamente dal device all'avvenuta consegna di un messaggio istantaneo.

Poiché, per quanto detto in precedenza, il protocollo di comunicazione implementato, è basato sull'invio e sulla ricezione di questa tipologia di messaggi, l'analisi che segue è stata effettuata su una grande quantità di dati acquisiti durante le prove.

Anche in questo caso la prima operazione che è stata eseguita è stata quella di confrontare tra loro le funzioni di densità di probabilità ottenute per i vari test analogamente a quanto fatto precedentemente.

Il risultato è riportato nella figura seguente.

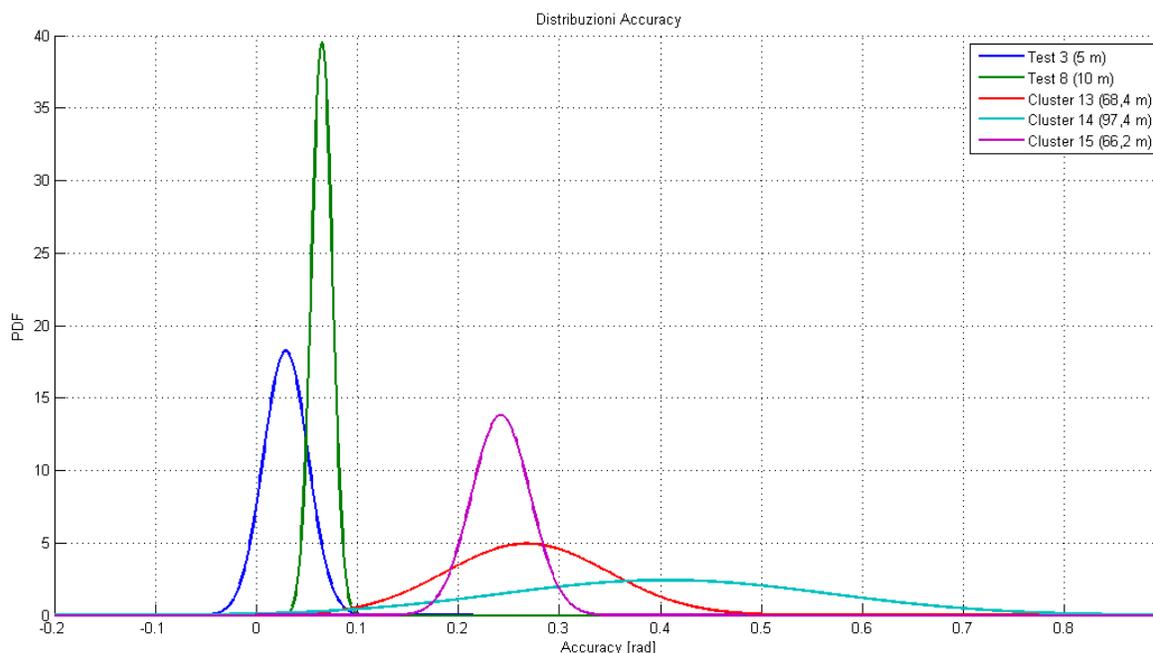


Figura 36: Funzioni di densità di probabilità del parametro Accuracy per i test 3, 8 e per i cluster 13, 14 e 15

Dall'analisi del grafico si nota come la forma delle funzioni di densità di probabilità sia molto diversa tra i vari test. La “campana” più stretta è quella riferita al test 8, mentre quella più larga è riferita al test 14. Anche per il test 13 siamo in presenza di una curva molto larga.

Per commentare correttamente questo grafico è necessario ricordare che l'Accuracy è un parametro di performance riferito all'accuratezza angolare della misura di posizione. Per questo motivo è espresso in radianti.

Osserviamo che le gaussiane più strette sono riferite ai test effettuati a terra mentre quelle più larghe a quelli effettuati in barca.

In particolare, se facciamo riferimento alla figura del tracking dei test, riportata nel paragrafo precedente, si osserva che il test 14 è l'unico che, a differenza degli altri, nell'andamento della distribuzione dei marker, presenta una curva piuttosto marcata.

E' possibile pertanto che per i test a bordo dell'imbarcazione, il movimento della barca così come la traiettoria da essa compiuta, abbiano influenzato la distribuzione dei valori di questo parametro.

In base alle considerazioni appena effettuate, anche per questo parametro, è stato scelto di non elaborare il grafico di fitting dell'andamento in funzione della distanza.

Una così marcata differenza di tipologia delle funzioni di densità di probabilità rende infatti il confronto tra di esse poco coerente e di conseguenza inadeguato al raggiungimento di una curva di caratterizzazione di questo parametro in funzione della distanza intercorrente tra i device.

4.4 Analisi delle tempistiche di comunicazione

Nella prima parte di questo capitolo l'attenzione è stata posta sui dati di posizionamento acquisiti durante le prove. Sono state svolte le analisi di coerenza e consistenza volte a valutarne la bontà. Successivamente è stata svolta l'analisi delle prestazioni del sistema, sulla base dei principali parametri di performance.

In questo paragrafo l'attenzione è posta sullo studio delle tempistiche di comunicazione tra i device.

Nella rete di comunicazione implementata, come noto, sono presenti due dispositivi in grado di comunicare tra loro attraverso lo scambio di Instant Messages (IM) di una lunghezza massima di 64 bytes, ad una velocità, dichiarata dal produttore, di 976 bps.

Per lo studio delle tempistiche di comunicazione è stata presa in esame la modalità TX – RX in quanto, come già osservato in precedenza, è quella che senza perdere di generalità, permette di estendere i risultati anche alla modalità TRH – TRH.

Pertanto, in questa configurazione, un dispositivo funziona esclusivamente come trasmettitore e l'altro esclusivamente come ricevitore.

Va ricordato che in fase di programmazione e di settaggio dei device è stato scelto di implementare la comunicazione attraverso gli IM con la richiesta di ack. Quindi, così come descritto nei capitoli precedenti, il trasmettitore, dopo aver inviato il messaggio, resta in attesa dell'ack da parte del ricevitore.

In assenza di malfunzionamenti, si assume che il messaggio venga sempre inviato con successo dal trasmettitore e che l'ack sia sempre generato e inviato con successo dal ricevitore.

In caso di insuccessi la gestione avviene secondo le modalità descritte nel capitolo due.

Attraverso l'analisi dei vari campi informativi contenuti nelle notifiche fornite dai dispositivi, è stato possibile ricostruire il diagramma temporale dell'unità temporale standard del sistema nella modalità scelta.

Il risultato, sintetizzato sotto forma di grafico nella pagina seguente, descrive dunque l'andamento temporale della comunicazione che intercorre per l'invio di un messaggio e la ricezione di un ack da parte del trasmettitore.

Va precisato che il diagramma non è in scala, ma che è stato realizzato in maniera proporzionale e generale comunque valida per ogni lunghezza di messaggio possibile e per ogni distanza operativa intercorrente tra i dispositivi.

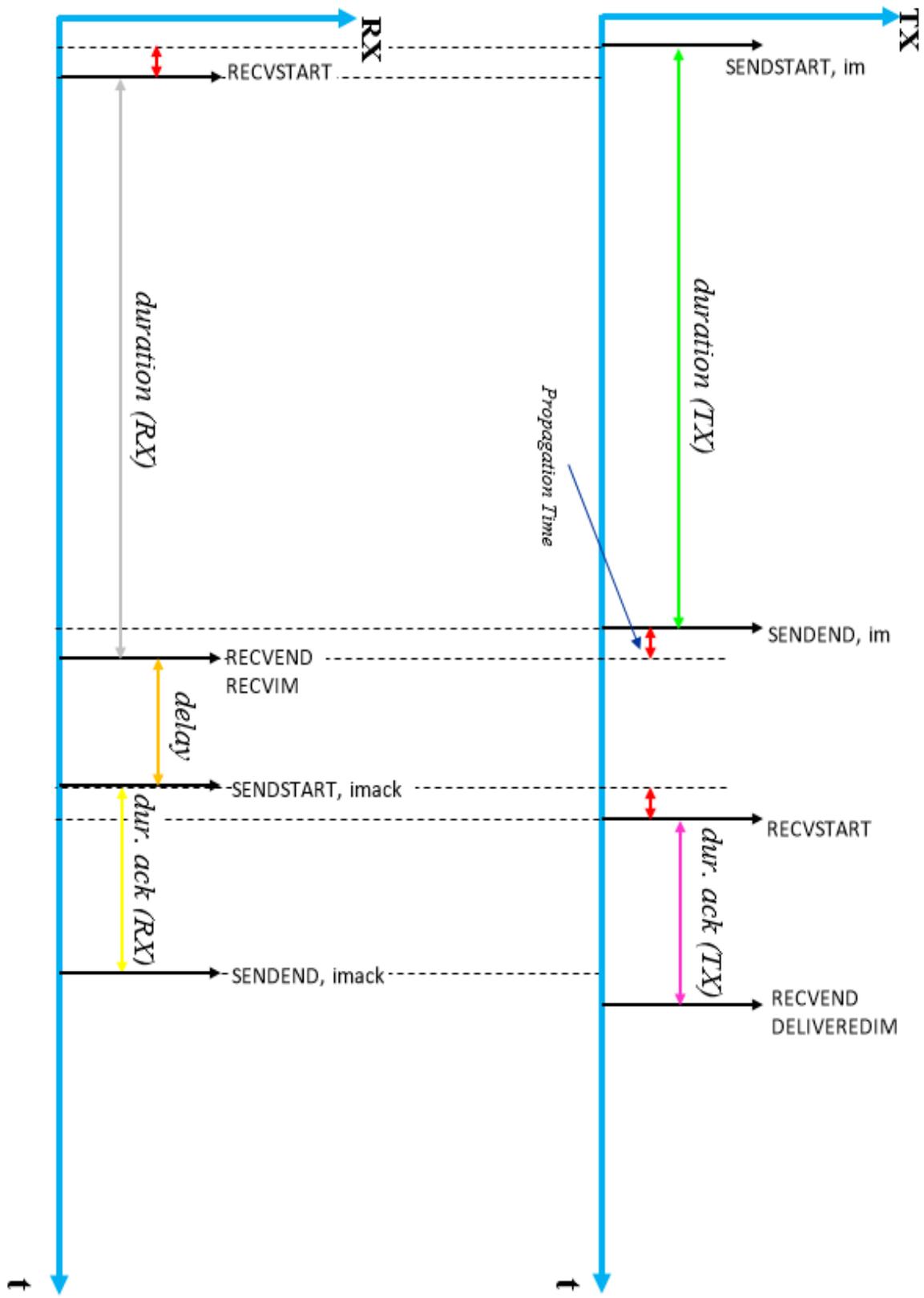
Nella parte alta del grafico è possibile osservare la linea temporale riferita a TX, mentre in basso quella riferita a RX.

Ad un dato istante di tempo il trasmettitore inizia ad inviare un IM e contemporaneamente viene generata la notifica *SENDSTART,im*, all'interno della quale viene notificato il valore del campo *duration*.

Questo valore, espresso in microsecondi, come tutti gli intervalli di tempo riportati nelle notifiche, è fisso, stabilito a priori dal trasmettitore e dipendente unicamente dalla lunghezza del messaggio inviato. Questo valore può essere interpretato come l'intervallo di tempo durante il quale il trasduttore è attivo in fase di trasmissione.

Allo scadere di questo intervallo di tempo viene generata dal trasmettitore la notifica *SENDEND,im*. Nello schema questo intervallo di tempo è evidenziato con il colore verde.

Dal punto di vista della linea temporale di RX, dopo un intervallo di tempo pari a *Propagation Time* dalla *SENDSTART, im* notificata da TX, viene rilevato in ingresso un IM e viene generata dal ricevitore la notifica *RECVSTART*.



Dopo un intervallo di tempo pari a *duration*, evidenziato nel grafico in grigio, vengono generate le notifiche *RECVEND* e *RECVIM*.

Dalla prima delle due viene estratto il valore proprio della *duration*, mentre dalla seconda quello del *Propagation Time*, evidenziato in rosso.

Poiché si sta facendo riferimento alla linea temporale di RX, la *duration* in esame può essere interpretata come l'intervallo di tempo in cui il traduttore del ricevitore rimane in ascolto ed è dipendente unicamente dalla lunghezza del messaggio inviato.

Il *Propagation Time*, invece, si riferisce all'intervallo di tempo necessario al segnale acustico per raggiungere il ricevitore ed è unicamente dipendente dalla distanza operativa intercorrente tra i device.

Rimanendo sullo schema temporale di RX, dopo un intervallo di tempo pari a *delay*, evidenziato in giallo nel grafico, il ricevitore genera la notifica *SENSTART,imack*.

Questa notifica viene generata in corrispondenza dell'istante di invio dell'ack al trasmettitore.

Tra i campi informativi contenuti al suo interno sono presenti proprio *delay* e *duration ack*.

Il valore di *delay* è costante, indipendente dalla lunghezza del messaggio e dalla distanza operativa. E' impostato dal produttore e non risulta modificabile.

Questo campo si riferisce all'intervallo di tempo intercorrente tra la fine della ricezione e l'inizio dell'invio dell'ack da parte di RX.

Per quanto riguarda invece *duration ack*, evidenziato in arancione, il suo valore, intrinsecamente dipendente dalla lunghezza del messaggio inviato, analogamente alle altre *duration*, risulta pressoché costante in tutti i test esaminati in quanto la lunghezza del messaggio di ack non è impostabile dall'utente, ma è fissata dal produttore.

Pertanto, per quanto già detto, il suo valore risulta costante e indipendente dalla lunghezza del messaggio inviato e dalla distanza operativa tra i dispositivi.

Tornando a considerare la linea temporale di TX, dopo un intervallo di tempo pari a *Propagation Time*, necessario all'ack per raggiungere il trasmettitore, quest'ultimo genera la notifica *RECVSTART* non appena viene rilevato un messaggio in ingresso.

Successivamente, dopo un intervallo di tempo pari a *duration ack*, evidenziato in rosa, il trasmettitore genera le notifiche *RECVEND* e *DELIVEREDIM*.

Dalla prima delle due viene estratto il valore di *duration ack*, mentre dalla seconda si evince l'avvenuta ricezione dell'ack e quindi la corretta ricezione da parte del ricevitore del messaggio inviato.

Terminata l'analisi e il commento del grafico relativo alle tempistiche di comunicazione tra i device, di seguito è riportata una tabella riassuntiva nella quale vengono riepilogate le dipendenze delle principali grandezze.

Grandezza	Dipendenza
<i>duration</i>	Lunghezza del messaggio
<i>Propagation Time</i>	Distanza Operativa
delay	Nessuna
<i>duration ack</i>	Nessuna

Dopo aver riassunto le dipendenze delle principali grandezze si passa adesso alla caratterizzazione di ciascuna di esse.

Per quanto riguarda la *duration* è stato possibile studiare con precisione la sua dipendenza in quanto, poiché il valore di questo parametro viene definito in modo automatico dal trasmettitore a seconda della lunghezza del messaggio, l'acquisizione dei valori è stata fatta in laboratorio senza bisogno del ricevitore.

I dati acquisiti sono stati riportati nella tabella seguente.

Lunghezza IM	Duration (μs)
5	198656
10	249856
15	299008
20	356352
25	413696
30	464896
35	524288
40	573440
45	647168
50	698368
55	749568
60	796672
64	849920

Nel diagramma temporale sono riportate le diciture *duration (TX)* e *duration (RX)* perché si è voluto differenziare volutamente il valore di questo parametro a seconda che venga notificato dal TX o dall'RX.

Va ricordato che, teoricamente, il valore di questo intervallo di tempo è da intendersi unico e costante indipendentemente dal dispositivo che lo notifica, ma sperimentalmente i valori notificati sono diversi.

In questa trattazione verranno analizzati entrambi e confrontati.

Nel grafico seguente sono stati disegnati in blu i punti ottenuti con i dati contenuti nella tabella precedente e quindi riferiti a $duration(TX)$.

Tali punti sono stati poi fittati, con l'ausilio del *cftool* di Matlab, con una retta, disegnata in rosso, di equazione e coefficienti:

$$f(x) = p_1x + p_2 \quad p_1 = 11,11 \quad p_2 = 136,7$$

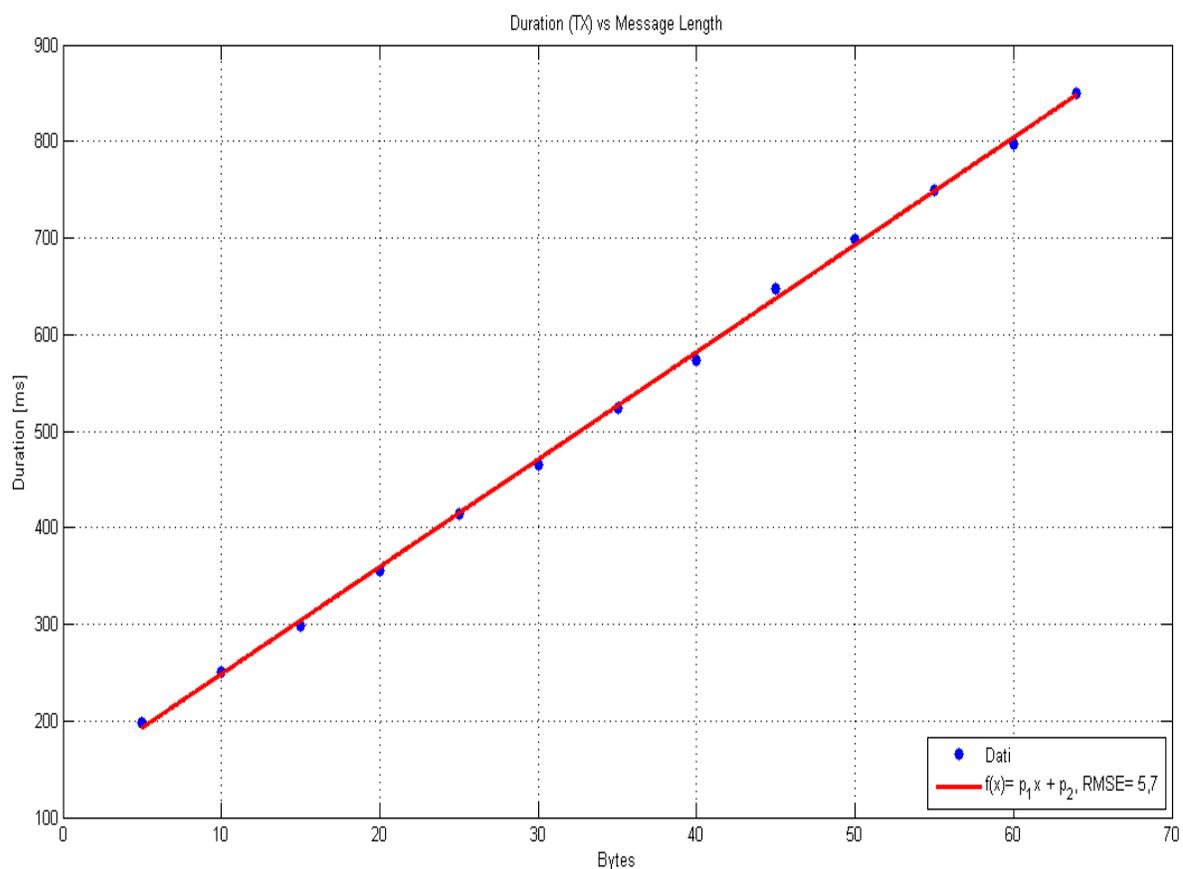


Figura 37: Curva di fitting per $duration(TX)$ in funzione della lunghezza del messaggio trasmesso

Poiché la curva di fitting è una retta, il coefficiente p_2 fornisce il valore dell'offset in corrispondenza del valore di zero bytes. L'offset è quindi pari a 136,7 ms.

In modo del tutto analogo è stata replicata la procedura anche per la $duration(RX)$.

In questo caso però è stato possibile utilizzare soltanto i valori acquisiti durante i test effettuati in mare, poiché per la generazione della notifica contenente questo campo informativo, il sistema di comunicazione doveva essere in funzione.

Di seguito è riportato il grafico ottenuto, completo di “error bar”.

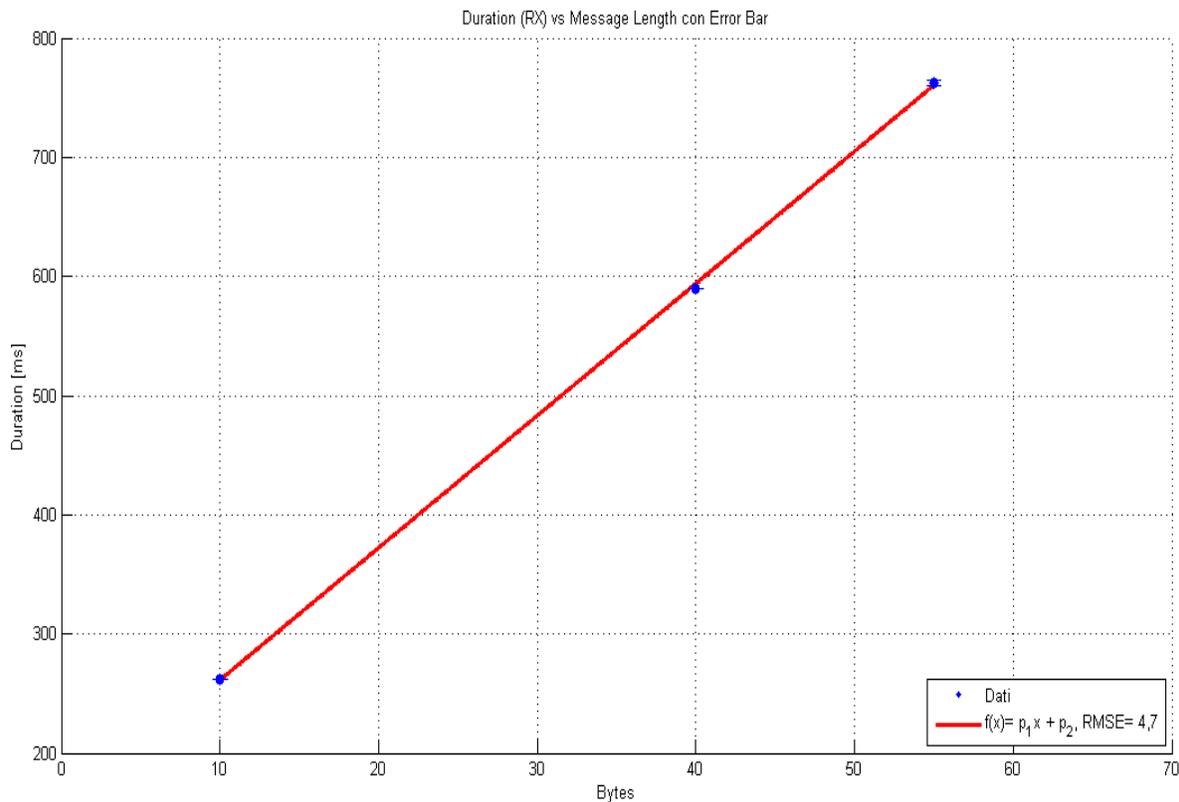


Figura 38: Curva di fitting con error bar per la $duration(RX)$ in funzione della lunghezza del messaggio

Anche in questo caso la curva di fitting utilizzata è una retta di equazione e coefficienti:

$$f(x) = p_1x + p_2 \quad p_1 = 11,09 \quad p_2 = 150,1$$

Analogamente a quanto detto per il caso precedente, è possibile osservare che l’offset in corrispondenza del valore di zero bytes è pari a 150,1 ms.

Il confronto tra i due grafici è riportato nel grafico seguente.

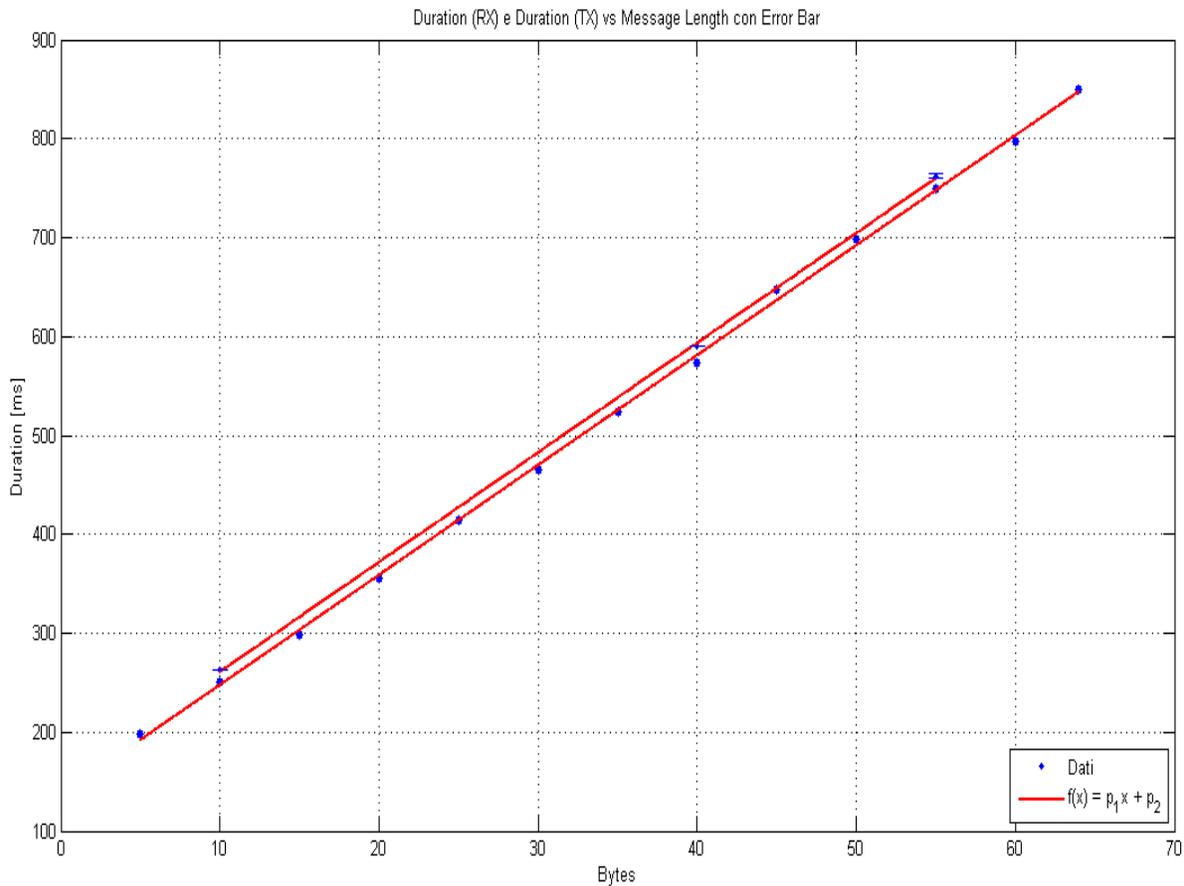


Figura 39: Confronto fitting duration(TX) e duration(RX) in funzione della lunghezza del messaggio

Come è possibile osservare dal grafico, le due rette risultano perfettamente parallele, come risulta anche dal confronto dei coefficienti p_1 .

La distanza tra le due altro non è che la differenza tra i due offset precedentemente riportati. Quindi:

$$\text{offset risultante} = 150,1 - 137,7 = 13,4 \text{ ms}$$

Per quanto riguarda invece il *delay* è stato osservato, come detto in precedenza, che il suo valore risulta essere sempre costante, non modificabile dall'utente, e indipendente da qualsiasi parametro.

Il suo valore risulta essere:

$$delay = 110000 \mu s$$

Considerazioni analoghe a quelle fatte in precedenza possono essere ripetute anche per la *duration ack*.

Per questo valore temporale è già stato osservato che non c'è dipendenza da nessun parametro di settaggio, in quanto la lunghezza del messaggio di ack è stabilita dal produttore e non risulta modificabile.

Come per la *duration*, anche in questo caso, nel diagramma temporale sono state utilizzate due terminologie diverse a seconda che il valore fosse notificato dal trasmettitore o dal ricevitore.

I valori notificati dai dispositivi sono:

$$duration\ ack(RX) = 133120 \mu s$$

$$duration\ ack(TX) = 147540 \mu s$$

$$offset\ risultante = 147540 - 133120 = 14,42\ ms$$

L'ultimo intervallo di tempo analizzato è il *Propagation Time*. Come già detto in precedenza, il suo valore è dipendente unicamente dalla distanza intercorrente tra i device. Questo intervallo di tempo, evidenziato in rosso nel diagramma temporale, ricorre più volte nel ciclo di comunicazione in quanto è interpretato come l'intervallo di tempo necessario al segnale acustico per coprire lo spazio tra il trasmettitore e il ricevitore.

Analogamente a quanto fatto in precedenza per altri parametri, come l'RSSI, la prima operazione che è stata eseguita è stata quella di confrontare tra loro le funzioni di densità di probabilità ottenute per i vari test.

Per questa analisi sono stati presi in considerazione soltanto i test 3 e 8, relativi alle prove effettuate rispettivamente a 5 e a 10 m, in quanto solo per questi test è stato possibile avere un riscontro metrico reale.

Quindi, per questi due test, sono state calcolate la media e la deviazione standard dei vettori contenenti tutti i valori di *Propagation Time* acquisiti ad ogni notifica, e, con l'ausilio della funzione Matlab *normpdf*, sono state plottate le due curve Gaussianhe ottenute.

Il risultato è riportato nella figura seguente, nella quale è possibile osservare come, all'aumentare della distanza intercorrente tra i dispositivi corrisponda anche un aumento del valore di *Propagation Time*.

Questo tipo di comportamento era auspicabile in quanto il *Propagation Time* è direttamente proporzionale alla distanza operativa tra i modem.

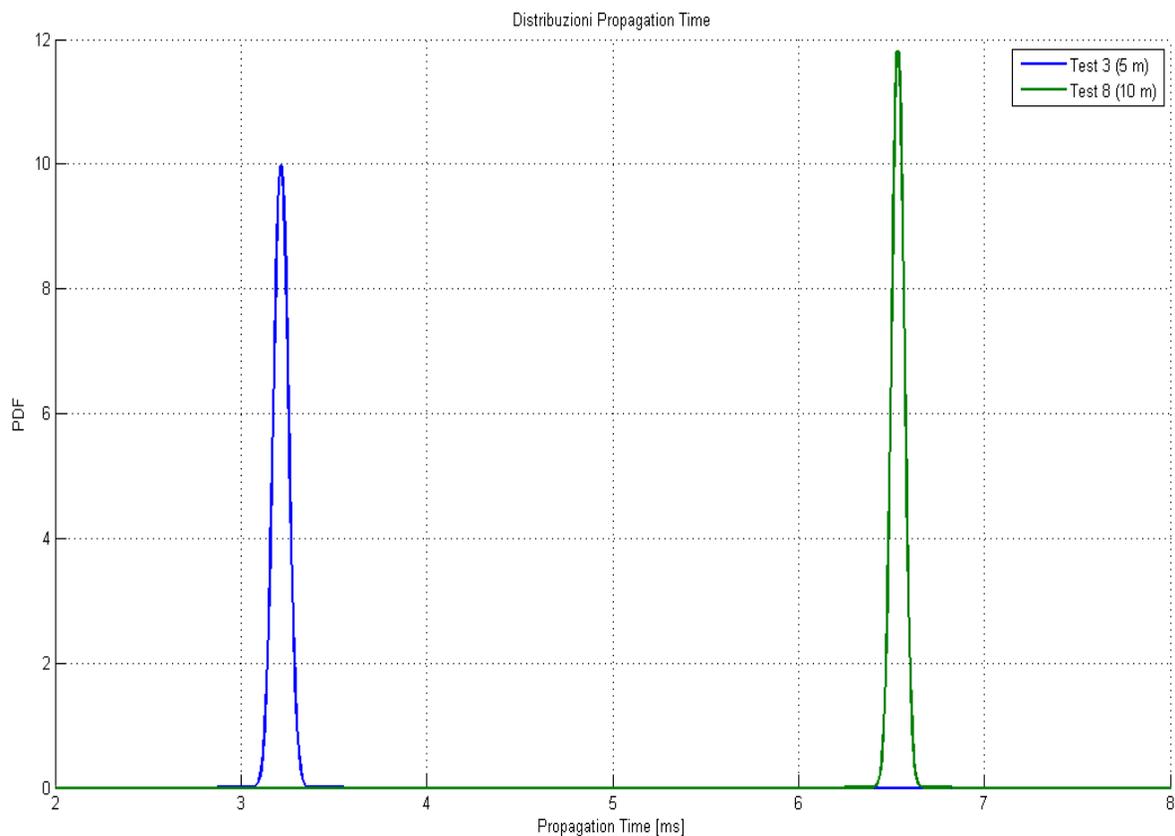


Figura 40: Funzioni di densità di probabilità del Propagation Time per i test 3 e 8

Coerentemente con le attese, al test 3 è associato un valor medio di *Propagation Time* più piccolo rispetto al test 8.

A questo punto si è osservato che sarebbe stato utile capire quale fosse la relazione che potesse descrivere la dipendenza di questo campo informativo dalla distanza operativa intercorrente tra i due dispositivi.

Pertanto è stato elaborato un grafico nel quale sono stati riportati i valori delle medie dei vettori di *Propagation Time* dei test selezionati, in funzione della distanza alla quale ciascun test è stato effettuato.

Successivamente, attraverso l'utilizzo del Matlab Toolbox *cftool*, è stata scelta la curva di fitting che meglio approssimava l'andamento dei valori al variare della distanza.

La curva selezionata è una retta caratterizzata dai seguenti parametri:

$$f(x) = p_1x + p_2 \quad p_1 = 0.66 \quad p_2 = -0.10$$

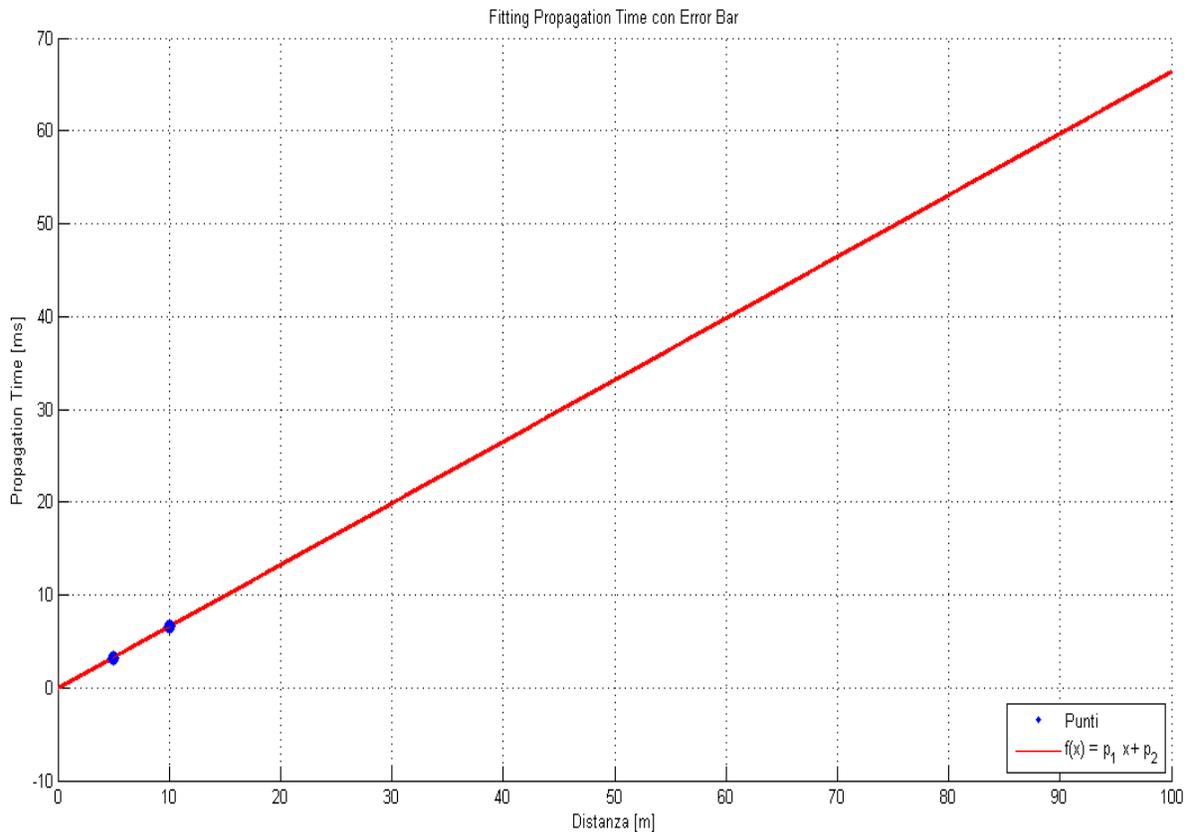


Figura 41: Curva di fitting con error bar per il Propagation Time in funzione della distanza

Nel grafico la curva di fitting ottenuta è stata prolungata analiticamente fino a coprire l'intervallo di distanza di interesse. L'offset in corrispondenza dell'origine degli assi è praticamente nullo in accordo con le aspettative.

L'acquisizione di tutti i dati è avvenuta assumendo come valore di setting del *Sound Speed* quello di default (1500 m/s).

Una volta fatta questa assunzione, è stato logico chiedersi quale fosse la bontà di questa scelta.

Il valore di velocità del suono c , impostato a 1500 m/s , è stato espresso in modo equivalente come 1.5 m/ms , dato che nel grafico, il *Propagation Time*, è espresso in millisecondi.

Poiché la retta di fitting esprime la dipendenza del tempo di propagazione dalla distanza intercorrente tra i device, il suo coefficiente angolare teorico risulta essere corrispondente a $1/c$.

Pertanto nel nostro caso la pendenza teorica di quella retta risulterebbe essere pari a:

$$\frac{1}{1.5} = 0,6667 \text{ [m/ms]}^{-1}$$

Si nota che il coefficiente p_1 della retta di fitting è molto prossimo a questo valore, essendo pari a 0,6639.

Pertanto, sulla base di questo confronto analitico, è possibile affermare che l'assunzione fatta sulla velocità del suono è pienamente attendibile.

Per avere un riscontro grafico di quanto già dimostrato in modo analitico, è stato elaborato un grafico nel quale sono state disegnate le due rette.

In blu è stata riportata la retta corrispondente all'andamento teorico del *Propagation Time*, passante correttamente per l'origine degli assi.

In rosso invece è riportata la retta ottenuta con i dati sperimentali, parallela alla precedente, ma traslata verso il basso, con un valore di offset espresso dal coefficiente p_2 .

Il grafico in questione è riportato alla pagina seguente.

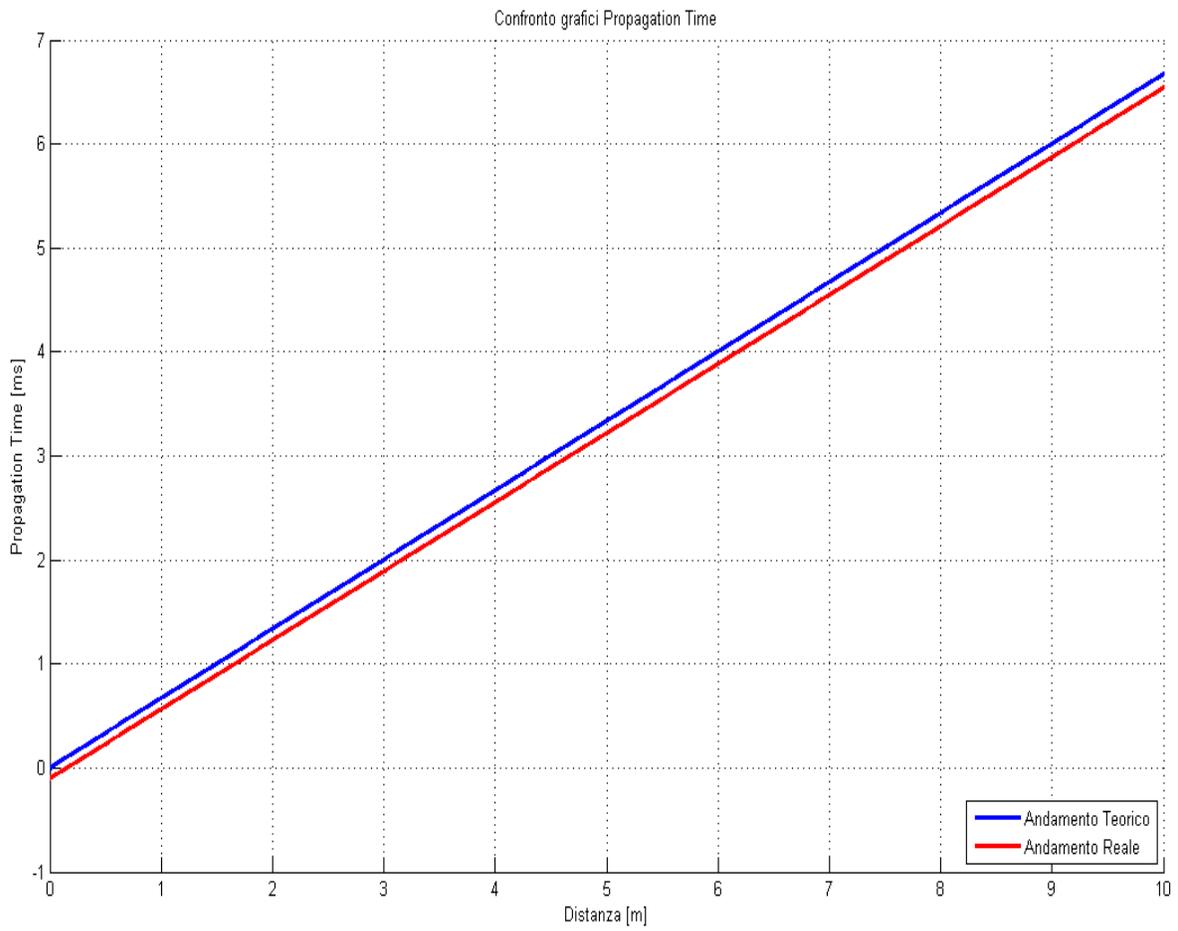


Figura 42: Confronto tra la pendenza teorica e quella reale della curva di fitting per il Propagation Time

4.5 Caratterizzazione RTT

Dopo aver analizzato nel paragrafo precedente tutte le componenti di tempo di un ciclo completo di comunicazione, in questo paragrafo sarà definito, analizzato e caratterizzato il *Round Trip Time* (RTT).

L'RTT è definito come l'intervallo di tempo necessario al trasmettitore per inviare un messaggio istantaneo e ricevere l'ack dal ricevitore.

Quindi, in riferimento al diagramma temporale proposto in precedenza, l'RTT corrisponde alla somma dei contributi di tempo compresi tra la *SENDSTART* e la *RECVEND* del trasmettitore.

La forma analitica assume dunque la seguente forma:

$$RTT = duration(RX) + delay + 2 Propagation Time + duration ack(TX)$$

Per quanto detto in precedenza, il valore di questo parametro è caratterizzato da due componenti considerate costanti, una dipendente dalla lunghezza del messaggio inviato, e l'altra dalla distanza operativa intercorrente tra i dispositivi.

Va ricordato che, in linea teorica, i valori di *duration* e *duration ack* dovrebbero essere indipendenti dal dispositivo che li notifica. Invece, come evidenziato nel paragrafo precedente, per ciascuno di essi c'è un offset tra il valore notificato dal trasmettitore e quello notificato dal ricevitore.

Per la definizione analitica dell'RTT è stato deciso di inserire i valori più grandi di queste due componenti in modo tale da avere una stima per eccesso dell'RTT.

Poiché, come già detto, l'RTT è caratterizzato da una doppia dipendenza, dal tempo e dalla distanza, è stato deciso di visualizzare questa dipendenza in due grafici separati.

Il primo grafico proposto è quello che descrive la dipendenza dell'RTT dalla lunghezza, in bytes, nel messaggio istantaneo trasmesso.

Per la generazione del grafico sono stati utilizzati i test effettuati in mare e descritti nel capitolo tre. Pertanto i valori di lunghezza, in corrispondenza dei quali è stato possibile disegnare i punti in blu, sono 10, 40 e 55 bytes, riferiti a test effettuati a una distanza di 5 metri.

Una volta calcolato il valore dell'RTT per questi valori, i punti così ottenuti sono stati fittati con una opportuna curva secondo la metodologia già descritta in precedenza.

I parametri che caratterizzano la funzione di fitting sono:

$$f(x) = p_1x + p_2 \quad p_1 = 11,09 \quad p_2 = 414,1$$

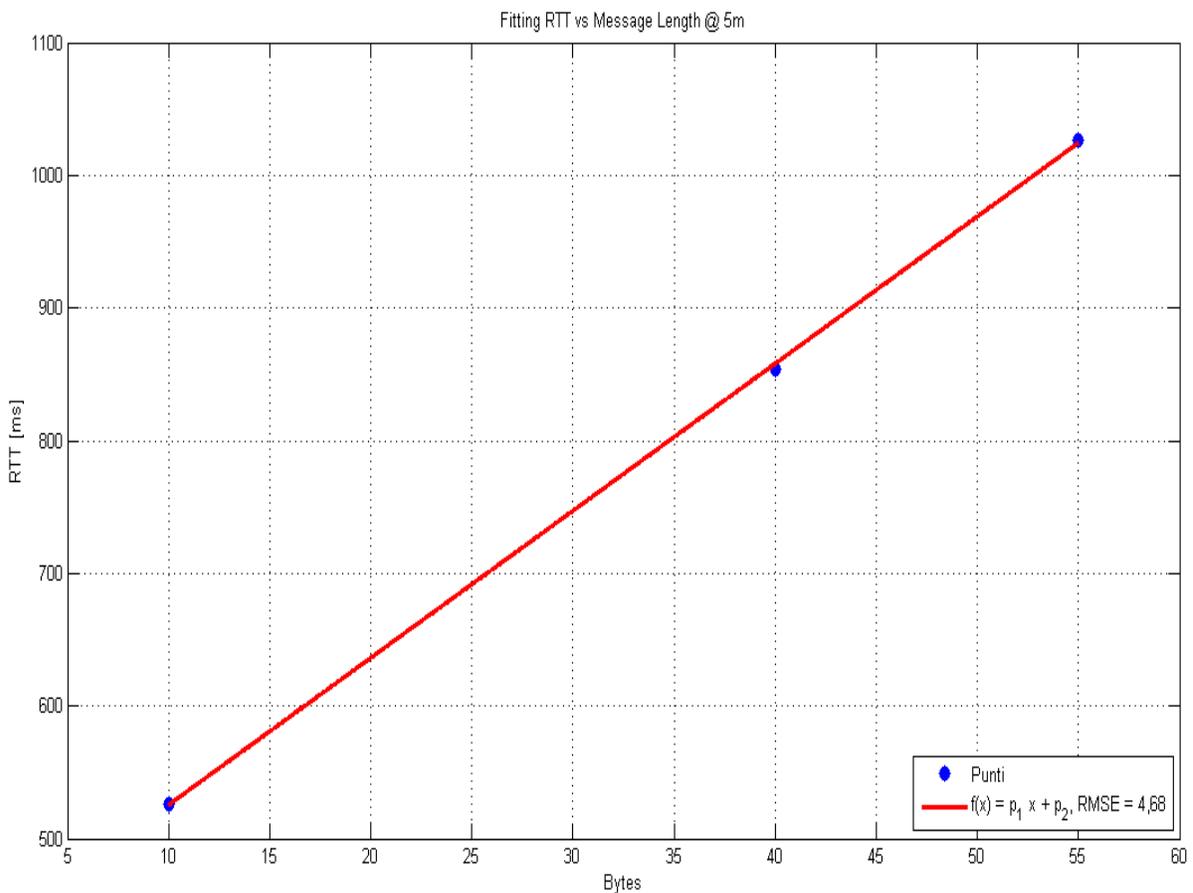


Figura 43: Curva di fitting per RTT in funzione della lunghezza del messaggio trasmesso

Il secondo grafico elaborato è quello che descrive la dipendenza dell'RTT dalla distanza operativa intercorrente tra i device.

Anche per la generazione di questo grafico sono stati utilizzati i dati relativi ai test effettuati in mare. I valori di distanza in corrispondenza dei quali è stato calcolato l'RTT sono 5, 10, 66.2 e 68.4 metri.

Per gli ultimi due valori sono stati utilizzati due dei cluster generati secondo le modalità descritte in precedenza.

Tutti i valori di RTT calcolati si riferiscono a prove effettuate con una lunghezza del messaggio inviato di 40 bytes.

Attraverso le medesime modalità del grafico precedente si giunge al fitting dei dati con la curva caratterizzata dai seguenti parametri:

$$f(x) = p_1x + p_2 \quad p_1 = 1,34 \quad p_2 = 847,2$$

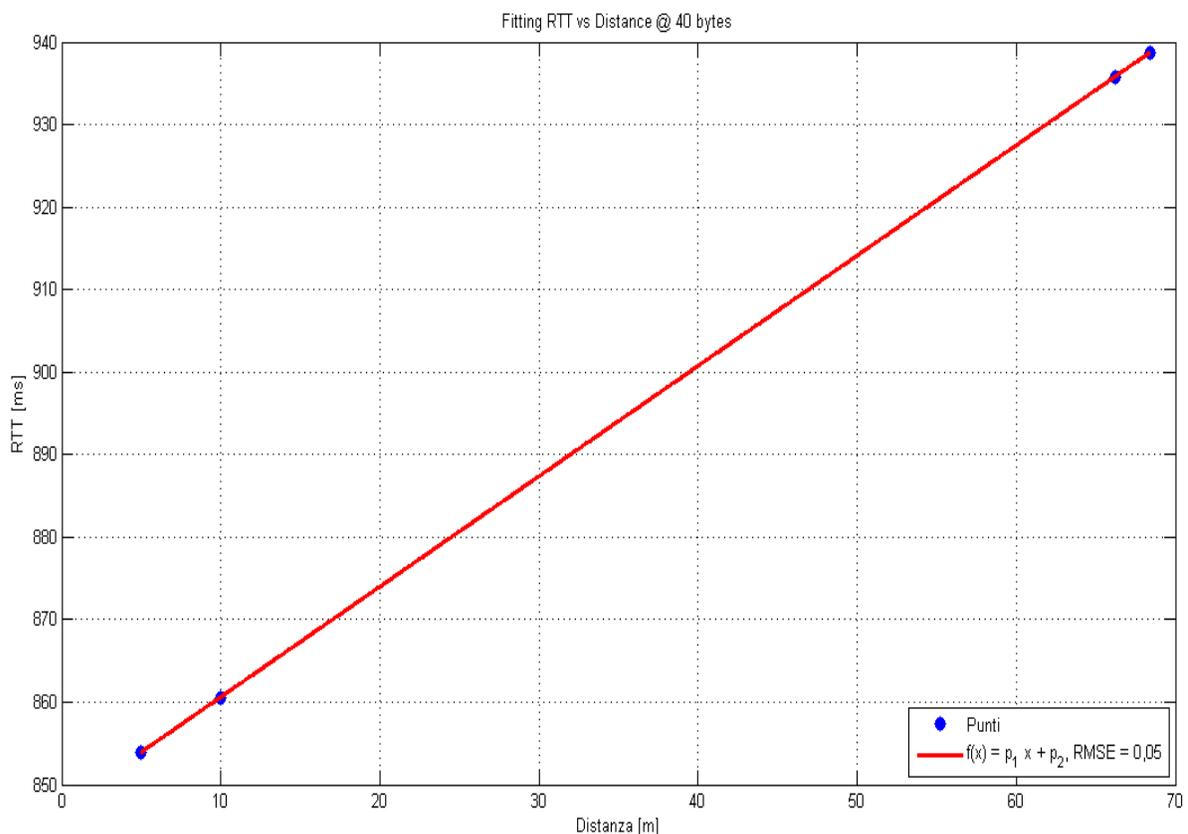


Figura 44: Curva di fitting per RTT in funzione della distanza

Dopo aver analizzato separatamente le due dipendenze dell'RTT dalla distanza e dalla lunghezza del messaggio, è sembrato logico procedere con la elaborazione di un grafico che fosse in grado di esprimere contemporaneamente questo rapporto di dipendenza.

E' stato così elaborato il grafico tridimensionale riportato di seguito.

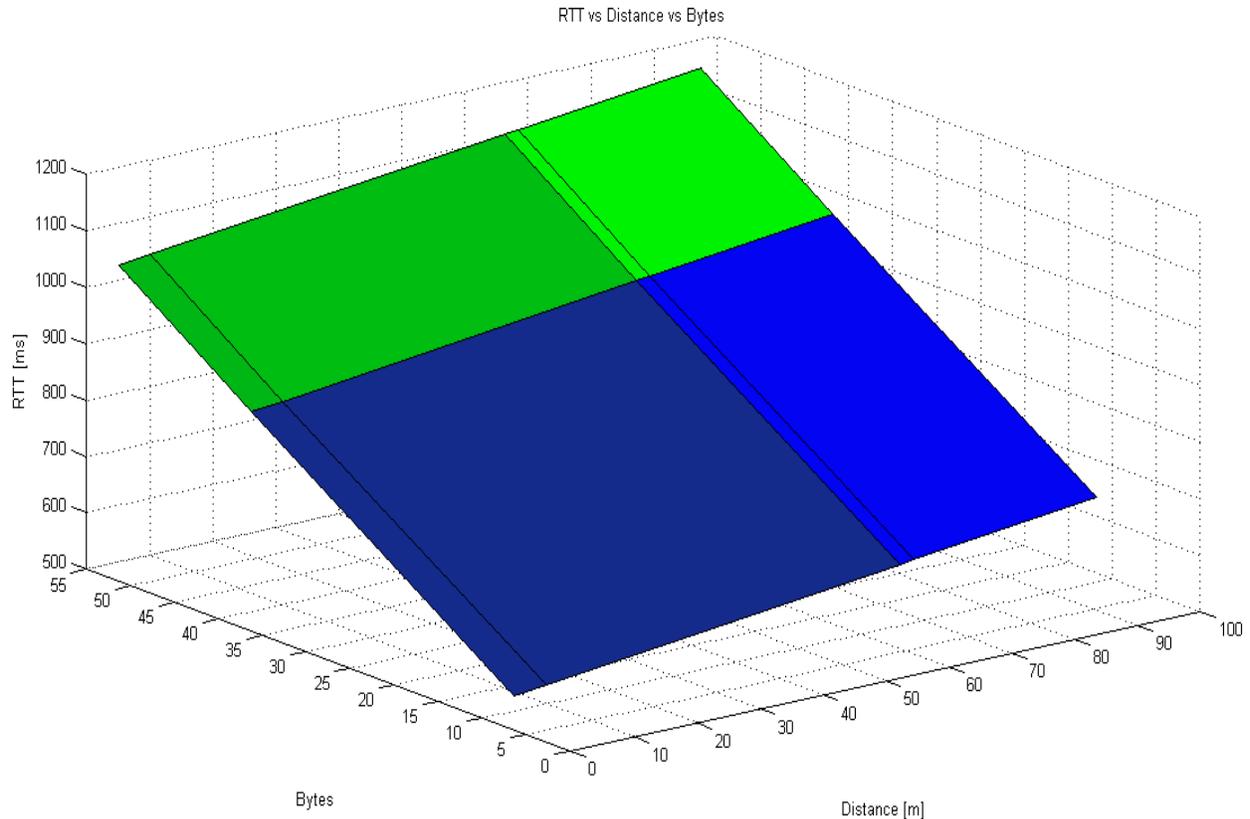


Figura 45: Grafico 3D descrivente la doppia dipendenza di RTT dalla distanza e dalla lunghezza del messaggio

Con la generazione di questo grafico l'utente e utilizzatore del sistema di comunicazione implementato, dispone di una "mappa" attraverso la quale può conoscere preventivamente l'RTT, e quindi la frequenza di trasmissione del sistema, una volta scelti la distanza a cui porre i dispositivi e la lunghezza del messaggio da inviare.

Allo stesso tempo il grafico può essere letto anche in senso opposto. Se si hanno dei vincoli di frequenza da rispettare, basterà calcolare il valore di RTT e trovare sul grafico 3D i corrispondenti valori di distanza e di bytes.

Con questo grafico può ritenersi completata la caratterizzazione delle tempistiche di comunicazione del sistema.

Va ricordato che il grafico è stato elaborato sulla base dei dati disponibili acquisiti durante la giornata di prove descritta nel capitolo tre.

Pertanto non rappresenta la gamma completa delle possibili configurazioni nelle quali il sistema può funzionare.

Il risultato ottenuto è da intendersi come un riferimento per comprendere quali siano le dipendenze delle prestazioni di comunicazioni del sistema.

Il metodo proposto può essere infatti utilizzato per estendere l'analisi ad una più ampia gamma di condizioni operative attraverso lo svolgimento di nuove campagne di acquisizione dati.

Fatta questa precisazione, sono riportati di seguito il valore massimo e quello minimo di RTT contenuti nel grafico precedente.

$$RTT_{min} = 526,22 \text{ ms}$$

$$RTT_{max} = 1150 \text{ ms}$$

Di conseguenza, definita la frequenza di trasmissione come:

$$f_T = \frac{1}{RTT}$$

è possibile calcolare i due valori di frequenza di trasmissione corrispondenti.

$$f_{T_{max}} = \frac{1}{RTT_{min}} = 1,90 \text{ Hz}$$

$$f_{T_{min}} = \frac{1}{RTT_{max}} = 0,87 \text{ Hz}$$

Ovviamente, qualora ci fosse l'interesse ad implementare una trasmissione ad una frequenza minore di quella minima calcolata, sarebbe sufficiente introdurre opportunamente dei ritardi nella comunicazione.

Ben più complesso sarebbe disporre di una frequenza massima più elevata in quanto, anche se quella indicata non è quella massima in senso assoluto, per farlo si renderebbe necessario trasmettere messaggi più corti o avvicinare molto i due dispositivi, andando ad operare magari in condizioni operative non compatibili con quelle del task da svolgere.

Facendo riferimento alle tre fasi della missione descritte nel paragrafo 1.3, è possibile adesso effettuare un confronto tra le condizioni operative presunte per ciascuna fase, e la reale prestazione del sistema in corrispondenza di ognuna di esse.

Fase di Navigazione

In questa fase i due veicoli navigano autonomamente fino al raggiungimento della posizione desiderata. All'inizio della missione è possibile che i due veicoli possano essere calati in acqua da posizioni diverse, o da imbarcazioni diverse, ed inizino la missione ad una distanza che verosimilmente può essere considerata intorno ai 100 m.

In questa fase ogni veicolo ha la necessità di conoscere la posizione dell'altro attraverso la ricezione di fix di posizione ottenuto attraverso il modulo USBL, e contenuto all'interno della notifica *USBLLONG*.

Pertanto, la trasmissione di un messaggio è funzionale solo alla ricezione del fix di posizione, e la lunghezza del messaggio trasmesso potrebbe, almeno

teoricamente, essere anche di 0 bytes. In queste condizioni operative, un valore di RTT stimato in base ai grafici di fitting presentati in precedenza, potrebbe essere il seguente:

$$RTT = 472,20 \text{ ms}$$

$$f_T = 2,12 \text{ Hz}$$

Questo valore di frequenza risulta essere superiore a quello massimo già indicato in quanto quello precedente era riferito ai dati ottenuti dai test reali, mentre questo è stato calcolato attraverso l'utilizzo delle funzioni di fitting.

Fase di Grasping

In questa fase i due veicoli si trovano ad una distanza proporzionale alla dimensione dell'oggetto da manipolare. Pertanto si può ragionevolmente assumere che la distanza operativa tra i due sia di circa 5 m. Questo anche in virtù del fatto che qualora la tubatura fosse più lunga, con ogni probabilità l'afferraggio della stessa da parte dei veicoli non avverrebbe alle estremità.

La fase di grasping può essere concettualmente divisa in due sottofasi:

- ✓ Approccio al target
- ✓ Post afferraggio

Durante la prima fase i veicoli si coordinano per afferrare la tubatura nei punti opportuni, prendendo a riferimento una terna comune fissata sull'oggetto.

Nella seconda fase gli end-effector sono posizionati sul target e i veicoli hanno la necessità di comunicarne la posizione rispetto alla terna comune.

In entrambe le fasi, la criticità è rappresentata dalla velocità di comunicazione tra i due piuttosto che dalla certezza della ricezione del messaggio. Pertanto la

trasmissione potrebbe essere implementata senza la richiesta di ack da parte del trasmettitore.

Per come è stato definito l'RTT, verrebbero azzerati i contributi dovuti al *delay*, e alla *duration ack*, e il *Propagation Time* verrebbe conteggiato una sola volta.

$$RTT_{grasp} = duration(RX) + Propagation Time$$

Per le ragioni descritte in precedenza, anche durante la fase di approccio, la lunghezza del messaggio trasmesso potrebbe essere, almeno teoricamente pari a 0 bytes. Pertanto, un valore di RTT stimato potrebbe essere il seguente:

$$RTT_{grasp_app} = 153,32 \text{ ms}$$

$$f_T = 6,52 \text{ Hz}$$

Per la fase Post Afferraggio, invece, la lunghezza del messaggio trasmesso potrebbe essere verosimilmente pari a 12 bytes, in quanto ogni veicolo comunica all'altro la posizione dell'end-effector rispetto alla terna comune. Pertanto, in queste condizioni operative un valore di RTT stimato potrebbe essere il seguente:

$$RTT_{grasp_post} = 286,40 \text{ ms}$$

$$f_T = 3,49 \text{ Hz}$$

Fase di Trasporto Coordinato

In questa fase, una volta afferrato l'oggetto, i due veicoli navigano ad una distanza di 5 metri, per le assunzioni fatte precedentemente, e necessitano di scambiarsi messaggi della lunghezza stimata di 24 bytes, contenenti velocità angolari e lineari, senza la necessità di ricevere l'ack. Pertanto, sulla base dei grafici precedenti, un valore di RTT plausibile potrebbe risultare il seguente:

$$RTT = 416,26 \text{ ms}$$

$$f_T = 2,40 \text{ Hz}$$

Per tutti i valori di tempo e frequenza riportati in quest'ultimo paragrafo, il tipo di comunicazione utilizzato, coerentemente con quanto osservato nei paragrafi precedenti, è di tipo half-duplex.

Nella tabella riportata di seguito sono stati riassunti i risultati presentati.

Fase della Missione	Distanza Stimata	Lunghezza IM Stimata	Ack	RTT Stimato	f_T Stimata
<i>Navigazione</i>	100 m	0 Bytes	Si	472,20 ms	2,12 Hz
<i>Grasping</i>	5 m	0 Bytes	No	153,32 ms	6,52 Hz
	5 m	12 Bytes	No	286,40 ms	3,49 Hz
Trasporto Coordinato	5 m	24 bytes	No	416,26 ms	2,40 Hz

Conclusioni

Sulla base del lavoro svolto è possibile affermare che gli obiettivi prefissati sono stati pienamente raggiunti.

Il sistema di comunicazione acustico è stato infatti implementato secondo le specifiche richieste sia dal punto di vista software che dal punto di vista operativo e le prestazioni ottenute sono da ritenersi pienamente soddisfacenti.

Oltre agli obiettivi dichiarati in fase di progetto è stata realizzata anche l'analisi delle performance del sistema al variare delle condizioni operative e la caratterizzazione delle tempistiche di comunicazione.

Pertanto, in riferimento al contesto del progetto MARIS, il sistema è pronto per essere testato in condizioni operative a bordo dei veicoli.

Sulla base delle prove che verranno svolte in condizioni operative reali, i possibili sviluppi futuri di questo sistema di comunicazione potrebbero mirare ad una migliore adattabilità delle sue caratteristiche alle necessità di missione e ad una caratterizzazione più approfondita di tutte le grandezze di interesse.

Bibliografia

- [1] Ribas, D.; Ridao, P.; Turetta, A.; Melchiorri, C.; Palli, G.; Fernandez, J.J.; Sanz, P.J., "I-AUV Mechatronics Integration for the TRIDENT FP7 Project," in *Mechatronics, IEEE/ASME Transactions on* , vol.20, no.5, pp.2583-2592, Oct. 2015doi: 10.1109/TMECH.2015.2395413
- [2] Sanz, P.J.; Ridao, P.; Oliver, G.; Casalino, G.; Petillot, Y.; Silvestre, C.; Melchiorri, C.; Turetta, A., "TRIDENT An European project targeted to increase the autonomy levels for underwater intervention missions," in *Oceans - San Diego, 2013* , vol., no., pp.1-10, 23-27 Sept. 2013
- [3] Evans, J.; Redmond, P.; Plakas, C.; Hamilton, K.; Lane, D., "Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation," in *OCEANS 2003. Proceedings* , vol.4, no., pp.2201-2210 Vol.4, 22-26 Sept. 2003 doi: 10.1109/OCEANS.2003.178243
- [4] G Marani, S. K. Choi, and J. Yuh, "Underwater autonomous manipulation for intervention missions AUVs," *Ocean Eng.*, vol. 36, no. 1, pp. 15–23, 2009.
- [5] Caiti, A.; Calabrò, V.; Fabbri, T.; Fenucci, D.; Munafo, A., "Underwater communication and distributed localization of AUV teams," in *OCEANS - Bergen, 2013 MTS/IEEE* , vol., no., pp.1-8, 10-14 June 2013 doi: 10.1109/OCEANS-Bergen.2013.6608166
- [6] Di Corato, F.; Fenucci, D.; Caiti, A.; Costanzi, R.; Monni, N.; Pugi, L.; Ridolfi, A.; Allotta, B., "Toward underwater acoustic-based simultaneous localization and mapping. Experimental results with the Typhoon AUV at CommsNet13 sea trial," in *Oceans - St. John's, 2014* , vol., no., pp.1-7, 14-19 Sept. 2014 doi: 10.1109/OCEANS.2014.7003092
- [7] Andrea Caiti, Francesco Di Corato, Davide Fenucci, Benedetto Allotta, Fabio Bartolini, Riccardo Costanzi, Jonathan Gelli, Niccolò Monni,"Acoustic communication and localization in AUV cooperative surveys". *Conference paper June 2014*.
- [8] Andrea Caiti. 2014. Cooperative Behaviours of AUV Teams and Networked Underwater Communication: How to Ask the Way and Not Go Astray. In *Proceedings of the International Conference on Underwater Networks & Systems (WUWNET '14)*. ACM, New York, NY, USA, , Article 50.

- [9] Casalino, G.; Caccia, M.; Caiti, A.; Antonelli, G.; Indiveri, G.; Melchiorri, C.; Caselli, S., "MARIS: A national project on marine robotics for interventions," in *Control and Automation (MED), 2014 22nd Mediterranean Conference of* , vol., no., pp.864-869, 16-19 June2014 doi: 10.1109/MED.2014.6961482.
- [10] Paull, L.; Saeedi, S.; Seto, M.; Li, H., "AUV Navigation and Localization: A Review," in *Oceanic Engineering, IEEE Journal of* , vol.39, no.1, pp.131-149, Jan. 2014 doi: 10.1109/JOE.2013.2278891
- [11] Ian F. Akyildiz, Dario Pompili, and Tommaso Melodia. 2006. State-of-the-art in protocol research for underwater acoustic sensor networks. In *Proceedings of the 1st ACM international workshop on Underwater networks (WUWNet '06)*. ACM, New York, NY, USA, 7-16. DOI=<http://dx.doi.org/10.1145/1161039.1161043>