



UNIVERSITA' DELLA CALABRIA

Dipartimento di Ingegneria Meccanica, Energetica e Gestionale (DIMEG)

Dottorato di Ricerca in

Ingegneria Civile ed Industriale - DICI

Con il contributo di (Ente finanziatore)

**Programma Operativo Nazionale Ricerca e Innovazione FSE-FESR 2014-2020
(CCI 2014IT16M2OP005)**


Fondo Sociale Europeo, Azione I.1 "Dottorati Innovativi con caratterizzazione Industriale"

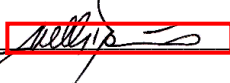
XXXIII CICLO

Studio delle tecnologie di localizzazione acustica subacquea e sviluppo di tecniche per la riduzione dell'errore di posizione.

Settore Scientifico Disciplinare: Disegno e Metodi dell'Ingegneria Industriale (ING-IND/15)

Coordinatore: Ch.mo Prof. Enrico Conte
Firma _____

Supervisore/Tutor: Ch.mo Prof. Fabio Bruno
Firma 

Dottorando: Dott. Rosario Aiello
Firma 

L'archeologia è una scienza profondamente umana. Attraverso l'oggetto l'archeologo deve risalire all'uomo che l'ha creato e da questo alla società in cui viveva. Chiunque può scavare oggetti, ma solo attraverso l'osservazione e l'interpretazione si può disepellire il passato.

[Leonard Woolley]



Dalla nascita l'uomo porta il peso della gravità sulle spalle.

Ma sotto la superficie dell'acqua siamo liberi.

[Jacques Yves Cousteau]

Sommario

L'attività è stata svolta nell'ambito del progetto di ricerca di dottorato "Sviluppo e sperimentazione di soluzioni innovative per la protezione e il monitoraggio degli ambienti archeologici sottomarini", finalizzata all'ideazione di nuovi metodi ingegneristici per la robotica subacquea, con applicazioni nel settore dell'archeologia subacquea. Si mira a dare supporto agli archeologi subacquei durante il complesso lavoro di ispezione, tutela, documentazione e monitoraggio dei siti archeologici sottomarini. Sono state incluse le tecnologie acustiche per la localizzazione e comunicazione dei dati, i veicoli autonomi di superficie e i veicoli subacquei a controllo remoto.

Il segnale del sistema GNSS si degrada già a pochi centimetri di profondità dal pelo libero dell'acqua, pertanto sono in fase di sviluppo tecniche acustiche a basso costo per aggirare il problema della geo-localizzazione anche nello scenario marino. Il principale focus del lavoro di tesi è stato quello di studiare e trovare delle soluzioni hardware e software per migliorare le performance dei sistemi di localizzazione acustica subacquea in termini di accuratezza del tracciamento del target subacqueo e della sua geo-localizzazione.

A causa della diversa natura della comunicazione sopra e sott'acqua, il flusso di dati scambiati tra questi ambienti richiede l'esistenza di una sorta di hub di comunicazione: si è definito un apparato modulare e configurabile che ha previsto l'utilizzo di un sistema di localizzazione acustico subacqueo SBL (*Short Base-Line*), un veicolo autonomo di superficie ASV (*Autonomous Surface Vehicle*) ed un tablet. È stato sviluppato un *package* in ambiente ROS (*Robot Operating System*) per garantire la comunicazione tra i vari componenti, ed una *webapp* di controllo il cui software ha permesso il monitoraggio dei dati scambiati tra i dispositivi, garantendo una stabile connettività tra di essi. Attraverso l'implementazione di una strategia di controllo automatica, il veicolo ASV rimane sempre nelle vicinanze del sub, quindi la comunicazione acustica con il tablet subacqueo risulta essere sempre robusta, ovviando al problema del degrado del segnale all'aumentare della distanza relativa. Si è affrontato, preliminarmente, il problema di come diverse sorgenti di errore possano andare a compromettere le performance della localizzazione subacquea. È stata effettuata l'analisi di propagazione degli errori a livello simulativo in diversi contesti statici e dinamici, ed in casi in cui la stima del dato di posizione venga effettuata con tecniche di localizzazione ibrida attraverso l'utilizzo di una unità inerziale montata nel ricevitore, o con tecniche di *Visual-Inertial Odometry* adoperando una camera subacquea. Ci si è concentrati, inoltre, ad indagare le performance di una stereo-camera commerciale in ambito subacqueo, svolgendo attività di progettazione e prototipazione; i risultati delle simulazioni sono stati valutati mediante prove sperimentali in piscina.

Si è pervenuti a dei risultati generali che sono stati molto soddisfacenti dal punto di vista del funzionamento hardware/software dell'intero sistema; inoltre, si sono ottenuti dei buoni miglioramenti sulla precisione del dato posizione con gli studi condotti in simulazione, in particolare adoperando strategie ibride con dati ottici.

Indice generale

Sommario	pag.	III
Introduzione	pag.	1
Capitolo 1 - Stato dell'Arte sulle tecniche di posizionamento subacqueo	pag.	8
1.1 I sistemi di localizzazione acustica subacquea	pag.	8
1.2 Dispositivi di supporto alla localizzazione subacquea	pag.	11
1.2.1 Piattaforme inerziali IMU	pag.	11
1.2.2 Global Navigation Satellite System	pag.	13
1.3 Metodologie impiegate per la localizzazione subacquea	pag.	16
1.3.1 Metodi di misura della distanza	pag.	16
1.3.2 Modello di propagazione del suono nell'acqua	pag.	17
1.3.3 Trilaterazione e triangolazione	pag.	18
1.3.4 Filtro di Kalman	pag.	20
1.3.4.1 Elementi e fasi del filtro di Kalman	pag.	20
1.3.4.2 Algoritmo e modelli	pag.	22
1.3.4.3 Problematiche di implementazione	pag.	26
1.3.4.4 Estensioni al filtro di Kalman	pag.	27
Capitolo 2 - Il sistema di localizzazione acustica subacquea del progetto BLUEMED	pag.	29
2.1 L'evoluzione dal progetto VISAS al progetto BLUEMED: contesto d'applicazione e tecniche operative	pag.	29
2.2 Strumentazione utilizzata nel progetto BLUEMED	pag.	30
2.2.1 Il sistema SBL	pag.	30
2.2.1.1 Trasmettitore (unità di superficie SBL)	pag.	31
2.2.1.2 Ricevitore	pag.	35
2.2.1.3 Principio di funzionamento	pag.	36
2.2.1.4 Interfacce di comunicazione	pag.	37
2.2.2 Descrizione del sistema ASV H2Omni-X	pag.	42
2.3 Integrazione del sistema SBL sull'ASV H2Omni-X	pag.	43
Capitolo 3 - Analisi di propagazione delle fonti di errore in un sistema SBL ..	pag.	48
3.1 Contesto	pag.	48
3.2 Sviluppo del modello	pag.	48
3.2.1 Principio di funzionamento del sistema SBL simulato	pag.	49
3.2.2 Sviluppo del modello Matlab/Simulink ed analisi delle fonti di errore	pag.	50
3.2.3 Considerazioni sull'errore introdotto dal solver di calcolo	pag.	53
3.3 Analisi di propagazione degli errori con ricevitore statico	pag.	53
3.3.1 Gestione del GNSS	pag.	54
3.3.1.1 Trasmettitore statico	pag.	54
3.3.1.2 Trasmettitore dinamico	pag.	57
3.4 Analisi di propagazione degli errori con ricevitore dinamico	pag.	59
3.4.1 Effetto della velocità del ricevitore sulla triangolazione in un modello		

semplice	pag.	59
3.4.2 Filtro di Kalman esteso (EKF)	pag.	61
3.4.3 Sensor-fusion del sistema acustico con la IMU del tablet e sensore di profondità	pag.	63
3.5 Sensor-fusion con dati ottici (visual-inertial odometry)	pag.	64
3.5.1 Metodi	pag.	64
3.5.2 Test sul campo	pag.	64
3.5.2.1 Protocollo dei test	pag.	65
3.5.2.2 Esecuzione dei test	pag.	66
3.5.2.3 Descrizione della ground-truth	pag.	67
3.5.3 Descrizione dei files di log	pag.	67
3.5.4 Descrizione del software di elaborazione dati	pag.	69
3.5.5 Risultati preliminari	pag.	69
3.5.6 Descrizione della strategia di hybrid tracking 2.0	pag.	72
Capitolo 4 – Integrazione del sistema SBL con l’ASV	pag.	73
4.1 Interfacciamento del sistema SBL su piattaforma ROS	pag.	73
4.1.1 ROS (Robot Operating System)	pag.	73
4.1.1.1 Generalità	pag.	73
4.1.1.2 Infrastruttura di comunicazione	pag.	74
4.1.1.3 Namespace e remapping	pag.	75
4.1.1.4 Feature specifiche per robot	pag.	75
4.1.1.5 Tools	pag.	75
4.1.2 Controllore ai potenziali virtuali	pag.	76
4.1.3 Simulazione in ambiente Matlab del controllore ai potenziali virtuali ..	pag.	78
4.1.4 Descrizione del software in ROS implementato sull’ASV H2Omni-X ..	pag.	80
4.1.5 Sviluppo del package di controllo	pag.	80
4.1.5.1 Launch file	pag.	81
4.1.5.2 to_sbl.py	pag.	81
4.1.5.3 Package per TCP	pag.	81
4.1.5.4 parser_tcp.py	pag.	81
4.1.5.5 to_vehicle.py	pag.	81
4.1.5.6 virtual_potential_controller.py	pag.	82
4.1.5.7 Servizi	pag.	82
4.2 Sviluppo della webapp di controllo	pag.	82
4.2.1 Descrizione della problematica	pag.	82
4.2.2 Presentazione della webapp di controllo	pag.	83
4.2.2.1 Front-end	pag.	84
4.2.2.2 Back-end	pag.	86
4.2.3 Gestione delle connessioni di rete	pag.	87
4.2.4 Infrastruttura di comunicazione della webapp	pag.	87
4.2.4.1 Porta 9001 per il trasmettitore acustico e protocollo TCP	pag.	87
4.2.4.2 Porta 9002 per il trasmettitore acustico e protocollo UDP	pag.	88
4.2.4.3 Porta 9003 per DEVSS (o boa) e protocollo UDP	pag.	89
4.2.4.4 Porta 9004 per H2Omni-X locale	pag.	89
4.2.4.5 Porta 9005 per il ricevitore acustico e protocollo TCP	pag.	89
4.2.4.6 Porta 9006 per il ricevitore acustico e protocollo UDP	pag.	90
4.2.5 Files della webapp	pag.	90
4.2.5.1 Veicolo H2Omni-X	pag.	90

4.2.5.2 <i>Trasmittitore acustico</i>	pag.	93
4.2.5.3 <i>Ricevitore acustico</i>	pag.	94
4.2.5.4 <i>DEVSS (o boa) per debug</i>	pag.	94
4.2.5.5 <i>Pc di bordo</i>	pag.	94
4.3 Test sperimentali e protocolli	pag.	94
Capitolo 5 - Il sistema commerciale USBL Seatrac della Blueprint: soluzioni adottate per il miglioramento delle sue performance	pag.	100
5.1 Il sistema USBL Seatrac e test preliminari	pag.	100
5.1.1 Descrizione dell'USBL Blueprint Seatrac	pag.	100
5.1.2 Test sperimentali per la localizzazione dei sub	pag.	101
5.1.3 Test sperimentali nel parco sommerso di Baia (NA)	pag.	104
5.1.4 Elaborazione dati del GNSS	pag.	105
5.1.5 Individuazione delle principali fonti di errore	pag.	105
5.2 Studio di un sistema per la riduzione delle oscillazioni dell'USBL	pag.	106
5.2.1 Analisi della problematica	pag.	106
5.2.2 Modello multi-body	pag.	107
5.2.3 Risultati analitici	pag.	110
5.2.4 Dall'analisi numerica alla creazione di un banco prova sperimentale	pag.	110
5.2.5 Analisi CFD	pag.	112
5.3 Filtro statistico a finestra mobile dinamica	pag.	113
5.3.1 Descrizione del setup	pag.	113
5.3.2 Descrizione dei parametri di input	pag.	113
5.3.3 Descrizione dell'algoritmo	pag.	114
5.3.4 Risultati in simulazione	pag.	116
5.4 Sensor-fusion con dati ottici (visual-inertial odometry)	pag.	117
5.4.1 Risultati preliminari	pag.	118
5.4.2 Strategie	pag.	120
5.4.2.1 <i>Strategia 1.0</i>	pag.	120
5.4.2.2 <i>Strategia 1.1</i>	pag.	121
Capitolo 6 - Ricostruzione 3D online in ambiente subacqueo utilizzando una camera di profondità a basso costo	pag.	122
6.1 Introduzione	pag.	122
6.2 Sviluppo del sistema e prototipazione	pag.	123
6.3 Analisi statica FEM sulla custodia impermeabile	pag.	125
6.4 Test sperimentali	pag.	127
6.5 Riepilogo	pag.	130
Conclusioni	pag.	131
Bibliografia e sitografia	pag.	133

Introduzione

Gli oceani ricoprono più del 70% della superficie terrestre, ma la maggior parte degli ambienti subacquei rimane ancora inesplorata. Per ragioni scientifiche ed economiche, la comunità accademica è interessata a studiare e capire meglio l'ambiente sottomarino per ragioni dovute all'esplorazione, al monitoraggio, alla prevenzione dei disastri, alla sorveglianza navale, alle operazioni offshore, all'archeologia ed alla biologia. L'Italia è conosciuta in tutto il mondo per il suo ricco patrimonio storico-artistico, che conta il maggior numero di siti dichiarati "patrimonio dell'umanità" dall'UNESCO. In particolare, data la ricchezza di aree costiere, il patrimonio archeologico subacqueo è vasto e rilevante tanto quanto quello terrestre. Tuttavia esso è ancora poco documentato e non adeguatamente valorizzato per via delle difficoltà intrinseche legate alla presenza dei beni in ambiente subacqueo. Il Mar Mediterraneo, infatti, possiede un vasto patrimonio marittimo il cui sfruttamento è reso difficile dai numerosi limiti imposti dall'ambiente sommerso. Le immersioni archeologiche risentono dell'impossibilità di fornire sott'acqua una spiegazione esauriente dei resti sommersi. Inoltre, ad aggravare ulteriormente la difficoltà delle immersioni, si aggiungono le condizioni di scarsa visibilità, dovute alla torbidità dell'acqua e alla colonizzazione biologica.

La tematica di ricerca affrontata nel presente lavoro di tesi si colloca in un contesto di studio e sviluppo di metodi e strumenti innovativi per applicazioni di monitoraggio, documentazione e valorizzazione nell'ambito dell'archeologia subacquea. In particolare il lavoro si focalizza sulla robotica subacquea, includendo le tecnologie acustiche per la localizzazione e comunicazione dei dati, i veicoli autonomi di superficie (ASV – Autonomous Surface Vehicle) ed i veicoli subacquei a controllo remoto (ROV – Remotely Operated underwater Vehicle). L'archeologia subacquea è un settore di intervento particolarmente bisognoso di tecnologie innovative poiché presenta delle difficoltà particolari derivanti dalla necessità di operare nell'ambiente marino. L'immersione è un'attività ad alto rischio [I.01] a causa dell'ambiente pericoloso, della dipendenza dall'attrezzatura tecnica per il supporto vitale, della complessità della navigazione subacquea e del monitoraggio limitato dalla superficie. I costi legati agli interventi di archeologia subacquea sono di gran lunga più elevati rispetto a quanto avviene sulla terraferma e, frequentemente, non è possibile utilizzare tecniche e strumenti tradizionali. In particolare, tutte quelle operazioni che richiedono l'intervento diretto degli archeologi sono limitate dai tempi di immersione consentiti dalle tabelle di decompressione che diminuiscono al crescere della profondità in cui ci si trova ad operare. Tali difficoltà operative hanno stimolato la ricerca e lo sviluppo di nuove tecnologie che puntano a rendere più efficiente e sicuro il lavoro degli archeologi subacquei.

Con il presente lavoro di tesi si mira quindi a sviluppare delle tecnologie che possano essere di supporto agli archeologi subacquei durante il complesso lavoro di ispezione, tutela, documentazione e monitoraggio dei siti archeologici sottomarini, mutuando, adattando ed integrando tecnologie innovative sviluppate per altri ambiti applicativi. Inoltre, tali tecnologie, potrebbero contribuire al miglioramento della fruizione turistica dei siti archeologici sottomarini con l'obiettivo di intercettare un numero sempre crescente di turisti subacquei, diversificando, quindi, l'offerta turistico-culturale delle aree costiere con un conseguente contributo alla destagionalizzazione dei flussi turistici, motivati dalla vastità di testimonianze storico-archeologiche presenti sui fondali marini (fig. I.1). Il tutto sarebbe permesso attraverso l'utilizzo di un innovativo tablet subacqueo, che sarà descritto nel capitolo 4 e che è stato oggetto di studio in questo lavoro di tesi: se l'ambiente è ben noto, i punti di interesse possono essere esattamente

localizzati sul display nelle mani dei fruitori; ogni utente conosce la sua posizione, e può osservare quella dei suoi compagni in un'esperienza di *augmented diving*.

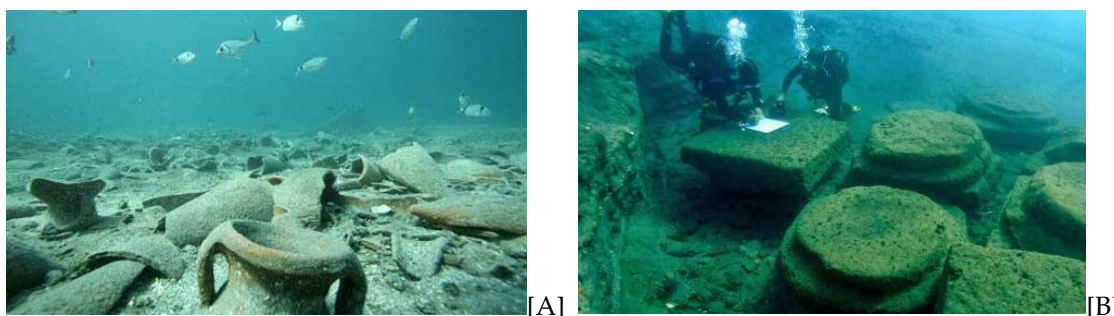


Figura I.1 – Esempio di patrimonio culturale sommerso dalle acque (a) e (b).

È noto che la tecnologia GNSS (Global Navigation Satellite System) permette di identificare la posizione di un oggetto sulla superficie terrestre. Il posizionamento, la navigazione e la comunicazione a terra sarebbero molto difficili e oggi inimmaginabili senza le reti di comunicazione satellitare. Un oggetto non può però essere geo-localizzato direttamente con il GNSS se si trova ad una certa profondità nell'acqua a causa dell'elevata attenuazione delle onde radio. È però possibile ricostruire la sua posizione rispetto ad una base posta sulla superficie dell'acqua, tramite sistemi inerziali o acustici. Dato che la base è geo-localizzabile, è possibile geo-localizzare anche l'oggetto in termini di latitudine e longitudine seppur si trovi ad una certa profondità nel mare (fig. I.2), adoperando strumenti matematici quali le matrici di rototraslazione. A tal proposito, lo si accenna brevemente ma verrà ampiamente descritto in seguito, ricopre un ruolo cruciale il fatto di dover avere una localizzazione molto precisa all'interno del sito archeologico sommerso, o più in generale all'interno dell'ambiente marino. Nel lavoro di tesi, in particolare nei capitoli 3 e 5, sono state studiate diverse soluzioni e strategie aventi l'obiettivo comune di voler migliorare il dato di posizionamento per consentire un'accurata geo-localizzazione subacquea.

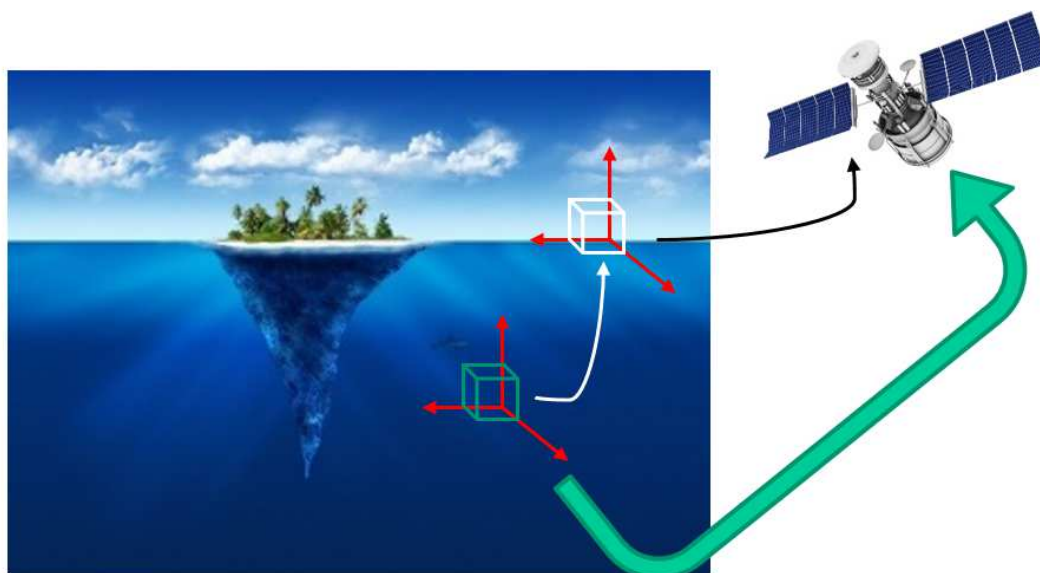


Figura I.2 – Schematizzazione logica per la geo-localizzazione subacquea.

In letteratura sono presenti alcuni lavori inerenti all'uso di tablet subacquei da parte dei sub, con o senza il supporto di veicoli robotici. In [I.02] è riportata un'applicazione dell'IoUT (Internet of

Underwater Things) che ha l'obiettivo di supportare il monitoraggio in tempo reale delle posizioni dei subacquei e le loro condizioni di salute, consentendo allo stesso tempo visite potenziate dei siti. Il sistema è composto da un modem acustico e da un tablet subacqueo sul quale è eseguita un'app dedicata; i sub possono comunicare sott'acqua, localizzarsi ed essere localizzati, navigare verso i punti di interesse e visualizzare contenuti in base alla loro posizione. Il progetto Lab4Dive [I.03] ha permesso di progettare, sviluppare e validare un sistema per supportare gli archeologi subacquei nel rilevamento e nella documentazione del patrimonio culturale sottomarino. Queste attività avrebbero notevoli margini di miglioramento se venissero introdotti nuovi dispositivi e strumenti tecnologici, aiutando sott'acqua archeologi per raccogliere dati dall'ambiente marino. I subacquei sono dotati di sistemi per la raccolta di dati portatili, a basso costo e di dimensioni relativamente ridotte (es. sensori ambientali, sistema di localizzazione acustica, telecamere e fotocamere, ecc.), così da poter migliorare significativamente sia la quantità di dati raccolti in una singola missione sia la gestione del tempo e delle risorse umane a disposizione. Il sistema sviluppato in Lab4Dive è basato su due moduli: il Modulo Cloud Server e il Modulo Subacqueo. Quest'ultimo supporta il subacqueo durante l'immersione fornendo un'interfaccia di navigazione che guida il subacqueo lungo i target selezionati, consente di acquisire e scambiare dati provenienti da diversi sensori ambientali e da una telecamera ad alta risoluzione. In [I.04] è descritta la progettazione e l'implementazione di un nuovo sistema per le operazioni subacquee basato su una efficiente rete acustica subacquea; il sistema è composto da nodi subacquei statici, e da sub che sono dotati di modem acustici e tablet subacquei. Il modulo di localizzazione è stato migliorato per aumentare la sicurezza dei subacquei le cui posizioni possono essere monitorate localmente e da remoto. Questi moduli interagiscono con un'innovativa app in esecuzione sul tablet consentendo così la stima ed il monitoraggio della posizione remota dei subacquei. Inoltre, i subacquei possono acquisire e condividere le immagini con altri operatori in tempo reale. L'interfaccia dell'applicazione è stata progettata secondo le esigenze dei sub, rendendola così perfettamente adattabile a diversi scenari di immersione. Il sistema è stato testato in diverse misurazioni in mare considerando diversi scenari applicativi. I risultati ottenuti confermano che il sistema proposto è idoneo per diverse esperienze subacquee, consentendo nuovi scenari applicativi ai fini della sicurezza dei subacquei. Il sistema PONTUS [I.05] è uno strumento di navigazione portatile, a basso costo, per scenari sottomarini, da utilizzare come dispositivo di localizzazione. PONTUS è costituito da un sistema integrato di posizionamento acustico USBL aiutato da un sistema di navigazione inerziale. Può essere montato su un veicolo robotico subacqueo o essere azionato da un subacqueo. Dispone inoltre di un'interfaccia utente grafica che fornisce informazioni sul tracciamento del target designato, oltre ad alcuni dettagli sulle proprietà fisiche all'interno di PONTUS. Il progetto MOLUX [I.06] ha sviluppato un sistema in grado di supportare gli operatori subacquei nella documentazione dei beni archeologici e naturalistici in ambiente marino integrando un tablet subacqueo con un veicolo autonomo di superficie e consentendo la comunicazione acustica fra questi due dispositivi. In [I.07] si indagano la fattibilità e le potenzialità delle tecnologie subacquee di Realtà Aumentata (UWAR) sviluppate nell'ambito del progetto iMARECulture per migliorare l'esperienza dei sub che visitano il Parco Archeologico Subacqueo di Baiae (Napoli). In particolare, l'articolo presenta due tecnologie UWAR che adottano tecniche di tracciamento ibrida per eseguire una visualizzazione aumentata delle condizioni reali e di un'ipotetica ricostruzione 3D dei resti archeologici come appariva in passato. Il primo integra un tracking basato su marker con sensori inerziali, mentre il secondo adotta un approccio "markerless" che integra la localizzazione acustica e l'odometria visivo-inerziale (visual-inertial odometry). Le sperimentazioni dimostrano che le tecnologie UWAR proposte potrebbero contribuire ad avere una migliore comprensione del sito sottomarino e dei suoi resti archeologici.

In [I.08] viene descritto l'utilizzo di un veicolo di superficie autonomo (ASV) come satellite privato che traccia i movimenti dei sub, aumentando così significativamente la sicurezza delle immersioni. Poiché il veicolo è sempre sopra il subacqueo, la comunicazione acustica con un tablet subacqueo è più efficiente e robusta, il che migliora la navigazione del sub e consente un monitoraggio affidabile dalla superficie. Questo articolo si concentra su una struttura di controllo per il rilevamento del sub, utilizzando uno stimatore del movimento per determinare la posizione del sub, anche nei casi in cui le misurazioni della posizione acustica non siano disponibili. In [I.09] l'Università di Zagabria ha sviluppato un veicolo di superficie autonomo portatile (ASV), supportato da un sistema per la localizzazione e la comunicazione dei dati. Il sistema è pensato per tre diverse applicazioni. La prima si basa sul supporto alle operazioni di immersione nelle quali il collegamento con il sub attraverso un tablet subacqueo garantisce un supporto al sub stesso come ad esempio l'essere inseguito, il monitoraggio, l'assistenza e maggiore sicurezza per l'immersione. La seconda prevede il supporto alle operazioni dei veicoli subacquei autonomi (AUV) consentendo la trasmissione in tempo reale dei dati provenienti dai sensori e la correzione dell'errore di navigazione "dead reckoning". La terza applicazione riguarda l'implementazione di reti di sensori wireless subacquei aiutati da sciami di ASV per l'inoltro, la localizzazione, la ricarica e la distribuzione dei dati in diverse condizioni ambientali. In [I.10] sono descritte le ultime prove di validazione del sistema CADDY (Cognitive Autonomous Diving Buddy). L'obiettivo principale del progetto è sostituire un compagno subacqueo umano con un veicolo autonomo subacqueo supportato da un veicolo di superficie autonomo per migliorare monitoraggio, assistenza e sicurezza della missione del sub. Il sistema è composto da tre componenti principali: un veicolo autonomo di superficie (ASV), un veicolo subacqueo autonomo (AUV) denominato BUDDY ed un subacqueo. CADDY mira a sperimentare la cooperazione tra un subacqueo e robot autonomi di accompagnamento (subacqueo e di superficie) che esibiscono un comportamento cognitivo adattandosi allo stato fisico dei subacquei ed alle loro azioni. Anche se alcuni problemi sono stati incontrati durante le prove, i dati raccolti e l'esperienza sono stati preziosi per identificare potenziali miglioramenti.

L'esperienza di dottorato innovativo a caratterizzazione industriale è stata supportata dall'Unical (DIMEG), dall'azienda 3DResearch, dove sono stati trascorsi 9 mesi e dal LABUST (Laboratory for Underwater Systems and Technologies) della Facoltà di Ingegneria elettrica ed informatica dell'Università di Zagabria (Croazia), presso il quale sono stati trascorsi 9 mesi.

I partner coinvolti nel percorso di dottorato hanno condotto diversi progetti di ricerca europei finalizzati allo sviluppo di nuove tecnologie per questo particolare settore di intervento legato all'archeologia subacquea, con soluzioni che mirano ad innovare lo stato dell'arte. In particolare 3DR ha partecipato ai progetti iMareCulture [I.11] nel quale ha realizzato un innovativo sistema di realtà virtuale e aumentata (VR e AR) subacquea descritto in precedenza, Lab4Dive [I.03] in cui ha sviluppato il prototipo di tablet subacqueo e MOLUX in cui ha svolto il ruolo di coordinatore. Il DIMEG dell'Unical è stato coinvolto nel progetto "Underwater Cultural Route in Classical Antiquity" [I.12] in cui si è puntato ad uno sviluppo di prodotti turistici legati all'archeologia subacquea e insieme a UniZg ha partecipato al progetto Bluemed finanziato dal programma Interreg-Med [I.13] che ha puntato a definire metodi e buone pratiche per la realizzazione di parchi archeologici sommersi attraverso l'uso delle nuove tecnologie. UniZg ha sviluppato diversi progetti europei fra cui CroMarX [I.14] con lo scopo di aumentare l'efficienza del monitoraggio subacqueo attraverso lo sviluppo di algoritmi di controllo cooperativo per veicoli autonomi di superficie e subacquei basati su comunicazione acustica. Nel progetto FP7-Caddy [I.15] il LABUST ha sviluppato l'ASV e ha svolto il ruolo di coordinatore. Nel progetto PON-Visas [I.16], che ha visto coinvolti la 3D Research e il DIMEG, si è sviluppato un tablet subacqueo con localizzazione LBL.

In tale contesto, il lavoro di tesi ha previsto un'integrazione sinergica dei risultati ottenuti dai sopracitati progetti al fine di contribuire alla definizione di una piattaforma che consentisse la cooperazione fra ROV, ASV e operatori subacquei che possono interagire e comunicare attraverso il tablet subacqueo. Questo approccio ha rappresentato un'innovazione nel settore di intervento specifico poiché non esistono soluzioni con caratteristiche paragonabili a quanto sviluppato, ed in grado di soddisfare le molteplici esigenze degli operatori subacquei che lavorano non solo nel campo dell'archeologia, ma anche dell'oceanografia.

L'attività di ricerca è iniziata con lo studio dello stato dell'arte dei vari progetti di ricerca internazionali inerenti alla tematica in questione, che hanno riguardato direttamente o indirettamente l'Unical e la 3DResearch, per poi passare ad approfondire più nello specifico il progetto Bluemed che è descritto nel capitolo 2.

Il principale focus del lavoro di tesi è stato quello di studiare delle soluzioni hardware e software per migliorare le performance dei sistemi di localizzazione subacquea, già esistenti o in via di sviluppo, in termini di accuratezza, il cui contesto è quindi quello dell'archeologia, ma le cui potenzialità sono estendibili all'esplorazione, al soccorso e a numerose altre applicazioni. Lo scenario su cui si è concentrata l'attività di ricerca del Dottorato è composto essenzialmente da un veicolo robotico di superficie (ASV) sul quale è stato integrato un trasmettitore acustico subacqueo, e da un target subacqueo che può essere un sub o un veicolo a controllo remoto (ROV), equipaggiato da un ricevitore acustico e da un tablet subacqueo (fig. I.3).

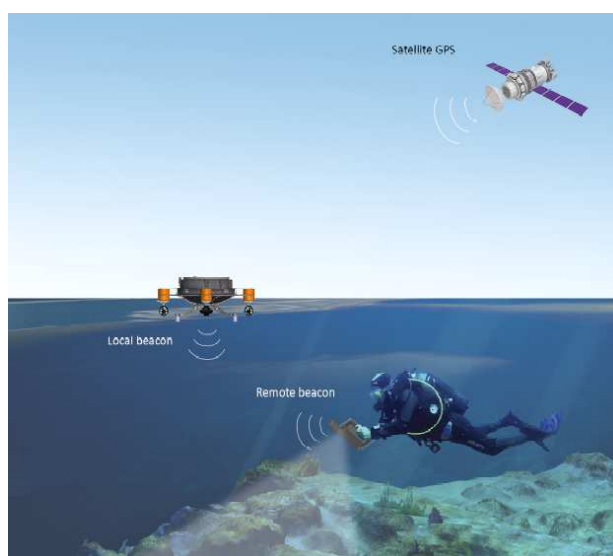


Figura I.3 – Ingegnerizzazione di un sistema di localizzazione acustico subacqueo.

Sul display del tablet è possibile visualizzare informazioni utili per l'immersione, in particolare la posizione su una mappa e la profondità. In sintesi, lo studio della configurazione appena descritta ha previsto una fase di studio di Stato dell'Arte, più fasi di studio in ambiente simulativo, e poi la fase di implementazione degli algoritmi sul sistema fisico, per poi passare alla fase dei test sperimentali. È importante sottolineare l'innovazione proposta dalla seguente configurazione che ha previsto un funzionamento del sistema acustico in "modalità inversa", ovvero è il trasduttore acustico in profondità che calcola la posizione e la trasferisce al tablet subacqueo. Inoltre, la possibilità del veicolo di spostarsi in modo intelligente, permette di inseguire il sub in uno spazio teoricamente infinito, mantenendo sempre una certa vicinanza ad esso; ciò permette di garantire sempre una certa robustezza nella trasmissione dei segnali acustici, evitando il degrado delle performance dovuto all'effetto acustico *multipath* all'aumentare della distanza relativa [I.17].

Il lavoro di tesi si articola in 6 capitoli. Nel **capitolo 1** viene presentata una panoramica dello Stato dell'Arte sui sistemi di localizzazione subacquea impiegati nel corso degli anni, seguendo la loro evoluzione e introducendo quelle che sono le generalità della trattazione; si è evidenziato l'importante ruolo svolto dai sistemi di localizzazione subacquea, entrando nel dettaglio in merito alla descrizione degli elementi della strumentazione che li caratterizzano. È stata inoltre fatta un'analisi delle metodologie impiegate in tali sistemi. Nel **capitolo 2** ci si concentra maggiormente sulla descrizione del progetto BLUEMED, elemento di connessione per tutte le attività condotte nel lavoro di tesi. Ci si concentra sui componenti e le metodologie impiegate nel sistema, focalizzandosi sul modo in cui questi interagiscono tra di loro per svolgere la funzione di localizzazione. È stato necessario approfondire tutti gli aspetti tecnici, hardware e software, per lo sviluppo della successiva fase implementativa. Nel **capitolo 3** è descritto lo sviluppo di un simulatore in ambiente Matlab/Simulink per l'emulazione del dato di posizionamento di un target subacqueo, affrontando il problema di come diverse sorgenti di errore possano andare a compromettere le performance sulla localizzazione. È stato creato un modello di simulazione capace di fornire informazioni sull'accuratezza del sistema di localizzazione subacquea. È fornita la descrizione di un sistema acustico SBL preso a riferimento per lo sviluppo del simulatore, utile per identificare le varie fonti di errore. È stata effettuata l'analisi di propagazione degli errori in un contesto in cui il ricevitore acustico sia sempre statico. È stata poi effettuata l'analisi di propagazione degli errori in un contesto in cui il ricevitore acustico sia dinamico, aggiungendo il caso in cui la stima della posizione venga effettuata insieme ad una unità inerziale montata nel ricevitore. Partendo da alcuni dati sperimentali del sistema SBL, sono state implementate delle strategie di sensor-fusion che vedono la combinazione del sistema acustico con una camera subacquea. Si è dunque proceduto a caratterizzare il modello secondo la realtà d'applicazione e sottolineare gli aspetti presentati nell'analisi. Il capitolo contiene le informazioni riguardo le simulazioni condotte, illustrando come queste sono state svolte e quali scenari volgono a simulare. Le simulazioni sono state validate tramite risultati descritti in letteratura in configurazioni coerenti con quella del caso di studio. Nel **capitolo 4** è stato analizzato uno scenario che ha previsto l'integrazione di un sistema acustico commerciale subacqueo di tipo SBL (Short Baseline) sviluppato da Applicon s.r.l. con un veicolo autonomo di superficie ASV e con un tablet subacqueo. Lo scopo del sistema è quello di permettere al veicolo di tracciare i movimenti di un sub tramite sensori acustici. È stata descritta l'architettura generale di partenza e gli interventi necessari. È stato descritto il software sviluppato in ambiente ROS (Robot Operating System) per l'integrazione software dei vari componenti. È stato descritto il software sviluppato per creare una webapp di controllo per il monitoraggio di tutti i parametri degli attori in gioco, e per garantire la connettività tra di essi. Sono stati riportati i risultati sperimentali provenienti da alcuni test effettuati in diversi momenti durante le fasi di sviluppo del sistema. È descritta la prova sperimentale svolta sul campo, utilizzando il prototipo del sistema di localizzazione con il ricevitore acustico, valutandone le capacità effettive in condizioni specifiche e confrontando i risultati con i dati simulati. Una delle parti dell'attività ha riguardato l'integrazione dei trasduttori e delle schede elettroniche dell'unità di superficie dell'SBL sull'ASV; successivamente è stata avviata l'attività di implementazione del software per interfacciare il sistema con il veicolo, e quella di un controllore che permettesse l'inseguimento del sub. Il **capitolo 5** ha riguardato lo sviluppo di soluzioni hardware e software per il miglioramento del dato di posizione di un sistema USBL commerciale. Sono descritti dei primi test sperimentali nei quali è stato testato il sistema USBL, mediante i quali è stato possibile testare le performance ed individuare le fonti di errore. È stato progettato e simulato un sistema di stabilizzazione meccanico per ridurre gli errori prodotti dalle oscillazioni della base acustica del sistema USBL. Lo stesso sistema USBL è stato posto sotto esame e sono stati apportati dei miglioramenti software sulla stima del dato di posizione mediante l'implementazione di un algoritmo basato sull'utilizzo

di un filtro statistico a finestra mobile. Come nel caso del sistema SBL nel capitolo 3, anche per il sistema USBL è stata prevista una integrazione con camera subacquea per testare come la fusione delle tecnologie possa portare a dei miglioramenti sulla qualità del dato di posizionamento. Incoraggiati dai risultati ottenuti, e tenendo conto dei benefici che potenzialmente può fornire la tecnologia ottica ai fini della localizzazione subacquea integrata insieme a quella dei sistemi acustici, ci si è concentrati, nel **capitolo 6**, ad indagare esclusivamente le performance di una stereo-camera commerciale in ambito subacqueo, ottenendo dati interessanti. Nelle **conclusioni**, infine, sono stati messe in risalto le criticità e i punti di forza del sistema, prevedendo i possibili sviluppi futuri e circoscrivendo i campi d'applicazione in relazione ai risultati del lavoro di analisi.

Capitolo 1

Stato dell'Arte sulle tecniche di posizionamento subacqueo.

In questo capitolo è descritto lo Stato dell'Arte nel quale si colloca il presente lavoro di tesi. Si è partiti da una premessa sul contesto operativo dei sistemi di localizzazione subacquea, storici ed attuali, di tipo commerciale o in fase di studio. Successivamente, è descritta la strumentazione hardware utilizzata per i sistemi di localizzazione acustica subacquea. In seguito, sono descritte le metodologie impiegate nei sistemi di localizzazione acustica subacquea, come ad esempio i metodi di misura della distanza, i modelli di propagazione del suono nell'acqua, la trilaterazione e la triangolazione, che sono stati in seguito applicati nel lavoro di tesi nel simulatore oppure in applicazioni reali. Infine, è presentata un'analisi del filtro di Kalman e del filtro di Kalman esteso, la cui approfondita conoscenza ed applicazione in differenti contesti è risultata essere indispensabile per apportare miglioramenti sui diversi casi studio affrontati in questo lavoro di tesi.

1.1 - I sistemi di localizzazione acustica subacquea

Per poter ispezionare un ambiente sconosciuto è necessario definire un sistema di localizzazione che ci permetta di conoscere la nostra posizione (o quella di un determinato obiettivo) con una certa accuratezza. Ogni ambiente ha caratteristiche peculiari che richiedono accorgimenti specifici e non sempre comuni ai molti, basti pensare alle differenze fra l'ambiente terrestre e quello spaziale. Un ambiente che è particolarmente ostico per l'essere umano è quello subacqueo a causa delle differenze rilevanti rispetto all'ambiente terrestre, non considerando solamente i limiti umani ma anche le tecnologie specifiche richieste.

I sistemi di localizzazione subacquea hanno fatto la loro comparsa sul mercato dai primi anni '60 e sono comunemente impiegati per varie applicazioni come l'esplorazione di siti impervi, operazioni di salvataggio, ricerca scientifica marina e operazioni archeologiche, di sicurezza e militari. Essi possono essere classificati, in funzione delle tecniche che impiegano, in tre tipologie: navigazione inerziale, navigazione geofisica e navigazione acustica. I sistemi di *navigazione inerziale* sono equipaggiati con giroscopi, bussole magnetiche, unità di misurazione inerziale a stato solido costituite da accelerometri e misuratori di velocità a effetto doppler. Il fine di un sistema di navigazione inerziale è quello di stimare posizione, velocità, accelerazioni e orientamento del veicolo senza la necessità di riferimenti esterni. In origine furono sviluppati per la guida dei razzi dal pioniere della missilistica americana Robert Goddard, tramite l'uso di sistemi giroscopici rudimentali. Un sistema a navigazione inerziale famoso fu il sistema Delco Carousel. I sistemi di *navigazione geofisica* utilizzano parametri geofisici confrontabili con una mappa nota elaborata precedentemente. Uno di questi parametri geofisici può essere il campo magnetico terrestre. I sistemi di *navigazione acustica* si basano sulla misura della distanza e della direzione dell'obiettivo che viene tracciato da un set di ricevitori (transponder/beacon) posti in punti di riferimento, chiamati "baseline station", con i quali tramite considerazioni geometriche si determina la posizione dell'obiettivo stesso. In funzione della distanza fra le baseline station questi vengono suddivisi in tre categorie [1.01], come si può vedere in fig. 1.1:

- Long Baseline (LBL);
- Short Baseline (SBL);
- Ultra Short Baseline (USBL).

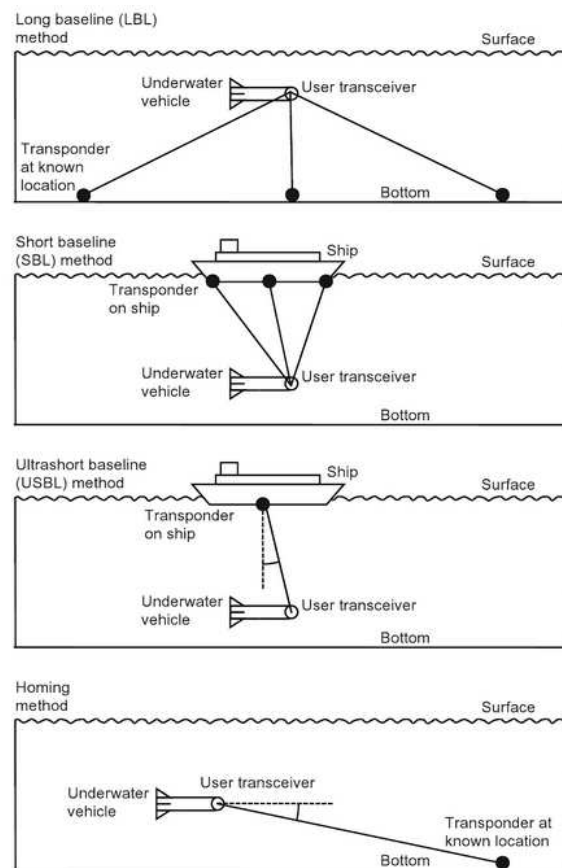


Figura 1.1 – Metodi di localizzazione acustica subacquea [1.02].

I sistemi LBL sono caratterizzati dal fatto che la distanza tra le baseline station è grande, ossia è paragonabile alla distanza che c'è fra l'oggetto da localizzare e i beacon (tipicamente dai 50m ai 2000m). I beacon vengono generalmente dispiegati sul fondale lungo il perimetro dell'area in cui vengono svolte le operazioni. Vi sono vari parametri che possono essere impiegati per definire la distanza fra l'emettitore e l'obiettivo da localizzare: l'indicatore di forza del segnale ricevuto, il tempo di arrivo, la differenza del tempo d'arrivo e l'angolo di arrivo. Nella pratica però i più utilizzati sono il tempo d'arrivo o la differenza del tempo d'arrivo [1.03]. Un segnale acustico viene inviato, generalmente dall'obiettivo, e i beacon rispondono una volta ricevuto definendo due istanti temporali sull'emettitore: quello di invio e quello di ricezione dell'eco. Quando il segnale svolge un doppio viaggio da trasmettitore a ricevitore, il metodo è denominato *two-way travel* (TWT). Alternativamente si ha il metodo *one-way travel* (OWT), più robusto nei confronti dei disturbi, in cui il segnale viene inviato una sola volta e tramite la sincronizzazione di clock atomici il ricevente può già sfruttare l'informazione sul tempo di invio per calcolare la distanza, senza ritrasmettere il segnale per far elaborare il calcolo al mittente primario. Il tempo di propagazione del segnale viene utilizzato per calcolare la distanza fra mittente e ricevente conoscendo il modello della propagazione del suono nel mezzo. Con il metodo OWT si può pensare di far inviare il segnale acustico alla baseline station e far elaborare l'informazione all'obiettivo. Se si dispone di un veicolo di dimensioni rilevanti, i beacon possono essere montati sul fondo (ad esempio sulla chiglia o sullo scafo) aumentando la versatilità del sistema. La

configurazione LBL nella maggioranza dei casi presenta un'accuratezza nella determinazione dell'oggetto da tracciare maggiore alle configurazioni SBL e USBL. Un'altra configurazione possibile del tipo LBL è quella con boe dotate di GPS (*GPS Intelligent Buoys*, GIB), come in fig. 1.2, che vengono ancorate al fondale o lasciate alla deriva. Il numero delle boe dipende dalla dimensione dell'area da ispezionare e dall'accuratezza desiderata. Il sistema GIB è stato brevettato dall'*ORCA Instrumentation* e dall'*ASCA*.

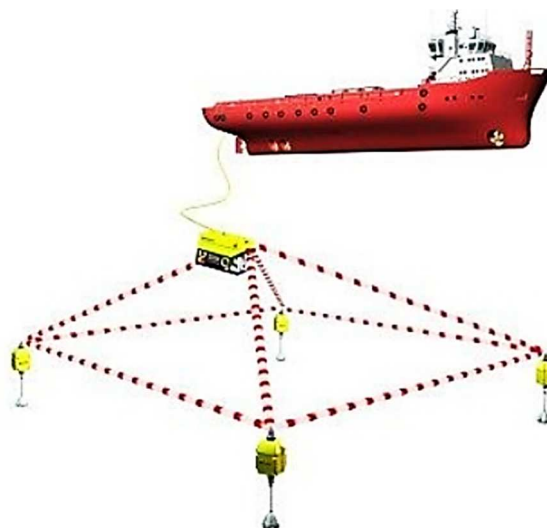
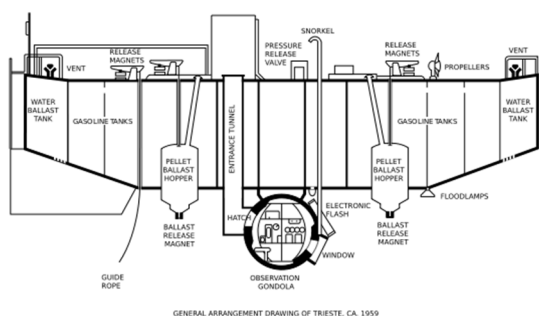
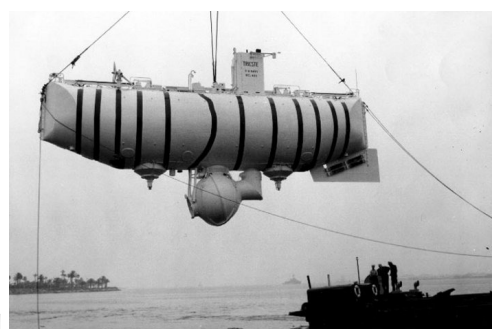


Figura 1.2 – Esempio di configurazione LBL con beacon fissi sul fondale e veicolo teleguidato localizzato sott'acqua [1.04].

I sistemi SBL invece, di dimensioni ridotte (dai 20m ai 50m), sono montati su piattaforme galleggianti come boe, barche o chiatte, di dimensioni inferiori a quelle usate per la configurazione LBL. Come i sistemi LBL questi impiegano una interrogazione circolare dei beacon per calcolare la distanza tramite trilaterazione o triangolazione dall'oggetto tracciato, spesso con l'ausilio di altri sensori come quello di profondità per migliorare la stima. L'accuratezza dei sistemi SBL cresce in modo proporzionale alla baseline fra i transponders, tuttavia la loro ridotta dimensione ne favorisce la versatilità e facilità d'impiego. I sistemi SBL risalgono storicamente al 1963, impiegati per la ricerca e ispezione del sottomarino nucleare scomparso *USS Thresher* dalla marina statunitense con il vascello *USNS Mizar*. Il suo sistema di localizzazione poteva vantare una precisione di circa 90m, che all'epoca era considerata rilevante per applicazioni simili. Alla ricerca dell'*USS Thresher* partecipò anche il batiscafo *Trieste*, in fig. 1.3, di produzione italiana, analogamente equipaggiato con un sistema SBL, famoso per essere stato uno tra i tre batiscafi capaci di raggiungere le profondità della fossa delle Marianne e tra i due ad averne fatto ritorno.



GENERAL ARRANGEMENT DRAWING OF TRIESTE, CA. 1959



[A]

[B]

Figura 1.3 – Principali caratteristiche del (a) batiscafo Trieste [1.05], (b) foto del batiscafo [1.06].

Sistemi SBL sono tutt'ora impiegati nel progetto SCINI in Antartide per tracciare il veicolo a controllo remoto sotto il ghiaccio (progetto SCINI), fig. 1.4. Gli USBL sono i sistemi più popolari di localizzazione subacquea grazie alla loro facilità d'utilizzo. I sistemi USBL comprendono due elementi: una trasmittente con una matrice di trasduttori che sono tipicamente disposti a meno di 10cm l'uno dall'altro, e un ricevitore. La trasmittente invia una pulsazione acustica e quando il beacon montato sull'oggetto da tracciare rileva questa pulsazione, replica con un'altra pulsazione. Il tempo che intercorre fra queste pulsazioni è impiegato per calcolare la distanza. Per misurare la direzione del trasponder dal trasmettitore, viene impiegato un metodo chiamato *phase-differencing* [1.07] con cui la matrice di trasduttori calcola l'angolo dal trasponder sottomarino. La posizione viene quindi ottenuta utilizzando la distanza e gli angoli fra trasmettitore e beacon. Chiaramente l'elevata flessibilità comporta un prezzo in termini di accuratezza.



Figura 1.4 – SCINI ROV del progetto SCINI in Antartide [1.08].

1.2 - Dispositivi di supporto alla localizzazione subacquea

In questa sezione sono descritti alcuni dei dispositivi che sono frequentemente impiegati nei moderni sistemi di localizzazione subacquea.

1.2.1 - Piattaforme inerziali IMU

Un componente essenziale dei sistemi di localizzazione è l'unità di misura inerziale (IMU). Gli elementi principali di una tipica unità di misura inerziale sono: accelerometro, giroscopio, processore, memoria per i parametri di calibrazione, sensore di temperatura, alimentazione. L'accelerometro e giroscopio da soli, senza gli altri elementi accessori, sono anche noti come *Inertial Sensor Assembly (ISA)*. Molte IMU sono dotate di tre accelerometri e tre giroscopi ad un grado di libertà, montati ortogonalmente agli assi di sensitività. In ogni caso, alcune IMU incorporano ulteriori sensori inerziali in una configurazione distorta per tutelarsi contro i guasti dei singoli sensori. Il processore dell'IMU svolge una conversione di unità sugli output dei sensori inerziali, provvedendo alla compensazione degli errori noti. La conversione consiste nel trasformare le grandezze elettriche in forze e variazioni angolari. Gli errori dei sensori inerziali sono spesso costanti e possono essere calibrati in laboratorio e memorizzati. Le IMU ad alte performance sono dotate di un sistema di controllo della temperatura per la compensazione degli errori associati ad essa essendo questi testati ad una temperatura fissata in fase di

caratterizzazione. Performance migliori si possono ottenere calibrando ciascun sensore individualmente e l'implementazione di un filtro di Kalman può aiutare ad ottenere facilmente i coefficienti di calibrazione dai dati misurati. Un ulteriore fonte d'errore per le IMU è quello legato all'effetto della dimensione (*size effect*). Nonostante l'accurato posizionamento del sensore, a causa delle sue dimensioni si può presentare un errore di qualche centimetro. Le vibrazioni introducono anch'esse un errore nelle misurazioni e alcune IMU sono dotate di un sistema di isolamento delle vibrazioni, che protegge i componenti anche dagli impatti. Questo sistema di isolamento deve essere progettato in modo tale da limitare la trasmissione di vibrazioni vicine alla frequenza di risonanza meccanica dei sensori o di aggiornamento computazionale dell'IMU. Ciascuna fonte d'errore sistematico ha quattro componenti: un contributo fisso, una variazione dipendente dalla temperatura, una variazione che cambia ad ogni utilizzo e una variazione all'interno di ogni utilizzo. Il contributo fisso viene corretto dal processore dell'IMU utilizzando i dati di calibrazione del laboratorio, e quando questo non viene corretto si nota uno sfasamento già nei primi minuti di utilizzo rispetto al valore vero. La variazione che cambia ad ogni utilizzo legata a ciascuna fonte d'errore è, per definizione, differente ad ogni utilizzo ma resta costante all'interno del singolo utilizzo. Questa non può essere corretta dal processore dell'IMU ma viene corretta in fase di calibrazione del sistema di guida inerziale o tramite algoritmi di integrazione ad ogni avvio. Infine, la variazione all'interno di ogni utilizzo, che provoca cambiamenti lenti durante l'esecuzione, teoricamente può essere corretta solo tramite processi di integrazione con altri sensori di navigazione ma nella pratica è difficile. Altri cambi repentini si osservano quando l'IMU subisce degli impatti [1.02]. Il bias è un errore costante presente in tutti gli accelerometri e giroscopi, fig. 1.5. Generalmente è indipendente da forze specifiche o variazioni angolari e rappresenta il termine dominante fra tutti gli errori di uno strumento inerziale.

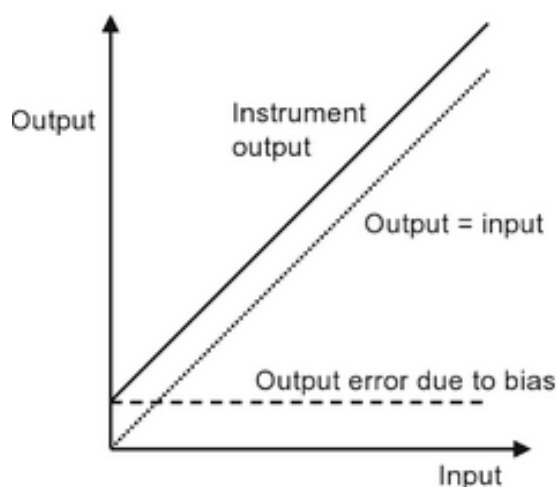
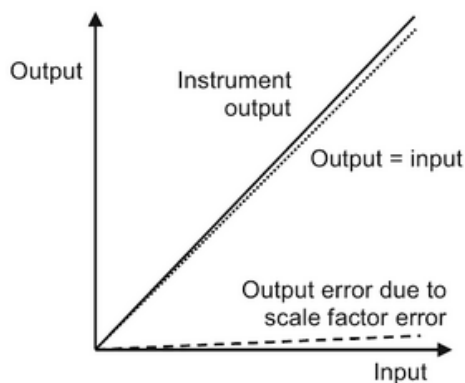


Figura 1.5 – Grafico Input-Output di un sensore soggetto a errore di bias [1.02].

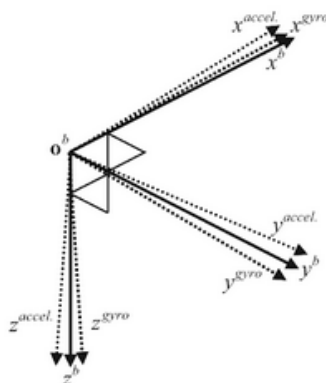
IMU Grade	Accelerometer Bias		Gyro Bias	
	mg	$m s^{-2}$	$^{\circ} hr^{-1}$	$rad s^{-1}$
Marine	0.01	10^{-4}	0.001	5×10^{-9}
Aviation	0.03–0.1	$3 \times 10^{-4} - 10^{-3}$	0.01	5×10^{-8}
Intermediate	0.1–1	$10^{-3} - 10^{-2}$	0.1	5×10^{-7}
Tactical	1–10	0.01–0.1	1–100	$5 \times 10^{-6} - 5 \times 10^{-4}$
Consumer	>3	>0.03	>100	$>5 \times 10^{-4}$

Figura 1.6 – Tipici bias di accelerometri e giroscopi per differenti gradi di IMU [1.02].

Tipici errori di bias di accelerometri e giroscopi sono rappresentati in fig. 1.6. L'errore di fattore di scala, fig. 1.7, è la divergenza del gradiente input-output dello strumento dall'unità secondo l'unità di conversione dell'IMU. L'errore sull'uscita dell'accelerometro dovuto al fattore di scala è proporzionale alla forza specifica lungo l'asse di sensitività mentre l'errore sull'uscita del giroscopio è proporzionale alla vera variazione angolare intorno all'asse di sensitività.



Scale factor error. (From: [19]. ©2002 QinetiQ Ltd. Reprinted with permission.)



Misalignment of accelerometer and gyro sensitive axes with respect to the body frame.

Figura 1.7 – Errore del fattore di scala e disallineamento dell'accelerometro e degli assi di sensitività del giroscopio rispetto al sistema di riferimento del veicolo [1.02].

Ulteriori fonti di rumore random introducono errori in qualsiasi sensore inerziale. Queste fonti di errore possono essere di natura elettrica (il rumore elettrico limita la risoluzione dei sensori inerziali, particolarmente per i MEMS dove il segnale è debole). Accelerometri a pendolo esibiscono rumore dovuto all'instabilità meccanica.

1.2.2 – Global Navigation Satellite System

Il sistema GNSS (Global Navigation Satellite System) è un sistema di geo-radiolocalizzazione e navigazione terrestre, marittima o aerea, che utilizza una rete di satelliti artificiali in orbita e pseudoliti. Un tale sistema fornisce un servizio di posizionamento geo-spaziale a copertura globale che consente a piccoli ed appositi ricevitori elettronici di determinare le loro coordinate geografiche (longitudine, latitudine ed altitudine) su un qualunque punto della superficie terrestre o dell'atmosfera con un errore di pochi metri elaborando segnali a radiofrequenza trasmessi in linea di vista da tali satelliti. Tra questi si annoverano il sistema statunitense NAVSTAR Global Positioning System (GPS), tuttora pienamente operativo. Il sistema russo GLONASS, pienamente operativo dal dicembre 2011. Il sistema europeo Galileo è entrato in servizio il 15 dicembre 2016. Tuttora in fase di implementazione, si prevede la sua piena

operatività nel 2019. La Cina vuole espandere e rendere globale il Sistema di posizionamento Beidou per ora regionale. L'India infine sta sviluppando IRNSS, un sistema GNSS di nuova generazione.

Il GPS fu sviluppato dal governo americano come sistema di navigazione militare. Questo offre due servizi di navigazione: lo *standard positioning service (SPS)* e il *precise positioning service (PPS)* per applicazioni militari. L'SPS è disponibile per tutti gli utenti con equipaggiamento idoneo mentre il servizio PPS è disponibile solo per gli utenti con autorizzazione dal governo americano, fra cui le forze militari americane e NATO. Gli utenti PPS hanno accesso a segnali criptati che non sono disponibili per gli utenti SPS. Negli anni '90 l'accuratezza dell'SPS è stato deliberatamente degradato sull'ordine dei 100m utilizzando una tecnica chiamata *disponibilità selettiva (Selective, Availability SA)*. Questo è stato voluto per negare una localizzazione precisa alle forze nemiche. Tuttavia, la SA è stata aggirata tramite l'utilizzo di *GPS differenziali (DGPS)*, mentre l'uso nemico del GPS può essere negato tramite tecniche di ostruzione locale (*jamming*). Nel tempo il GPS ha subito dei miglioramenti che ne hanno incrementato l'accuratezza sull'ordine dei 2.5m per l'SPS e di 1-1.5m per il PPS. Il GPS opera con una costellazione nominale di 24 satelliti (fino ad un massimo di 36). Tutti i satelliti operativi provvedono un servizio pieno e i satelliti di nuova generazione incorporano servizi di navigazione automatica (*Autonav*) che abilita le informazioni di navigazioni ad essere aggiornate indipendentemente dal segmento di controllo usando misurazioni di distanza intersatellitare. Vi sono 10 differenti segnali di navigazione GPS, trasmessi attraverso tre bande, conosciute come *Link 1 (L1)*, *Link 2 (L2)* e *Link 5 (L5)*. Le frequenze portanti sono 1575.42 MHz per L1, 1227.60 MHz per L2 e 1176.45 MHz per L5. Trasmettere segnali SPS su più di una frequenza migliora l'accuratezza e la stabilità. Dal lato europeo, è stato sviluppato il sistema di navigazione satellitare *GALILEO*, a partire dal 1999, da parte dell'*Unione Europe e dell'Agenzia Spaziale Europea (ESA)*. A differenza del GPS, il sistema GALILEO è impiegato solo per scopi civili. È gestito dal *GNSS Supervisory Authority (GSA)*. GALILEO offre quattro servizi di navigazione: un sistema aperto (OS), un servizio safety-of-life (SOL), un servizio commerciale (CS) e un servizio regolato pubblicamente (PRS). A questi si somma un servizio aggiuntivo di ricerca e salvataggio (SAR). GALILEO ha una costellazione nominale di 27 satelliti che orbitano a 29600 km con un periodo di 14 ore e trasmette 10 differenti segnali di navigazione attraverso tre frequenze di banda: E5, E6 e E1-L1-E2. I satelliti contengono un orologio atomico che calcola al millesimo di secondo il tempo che passa dalla richiesta effettuata dal ricevitore GPS alle risposte ottenute dai satelliti stessi. Con questa informazione ogni singolo satellite può determinare la distanza del dispositivo ricevitore e tramite triangolazione (con almeno tre satelliti) definirne la posizione spaziale. L'errore sulla localizzazione viene compensato trovando quel fattore con il quale si ha un unico punto di intersezione fra le sfere definite dalla distanza ricevitore-satellite. Con più di tre satelliti si va migliorare la stima. Nel caso di GPS differenziale si usa oltre alla stazione mobile (il satellite), una stazione di riferimento (terrestre) composta da un ricevitore GPS notevolmente accurato, fisso in un punto di coordinate note, che confronta la posizione determinata dal ricevitore con quella attesa e calcola le correzioni per annullare la differenza diffondendole alla stazione mobile che le applica aumentando l'accuratezza della misura fino a pochi centimetri [1.11]. Diversi sistemi di navigazione inerziale sono stati confrontati (fig.1.8, fig. 1.9, fig. 1.10):

	VN-110	VN-210	VN-310
NAVIGATION			
Heading (INS)	-	< 0.1 ° RMS	< 0.1 ° RMS
Heading (GPS-Compass) ¹	-	-	0.3 ° RMS
Pitch/Roll (Static)	< 0.05 ° RMS	< 0.05 ° RMS	< 0.05 ° RMS
Pitch/Roll (Dynamic)	-	< 0.03 ° RMS	< 0.03 ° RMS
Horizontal Position (w/ SBAS)	-	2.0 m RMS	2.0 m RMS
Vertical Position	-	5 m RMS	5 m RMS
Velocity	-	< 0.05 m/s	< 0.05 m/s
PHYSICAL			
Dimensions	56 x 56 x 23 mm	56 x 56 x 31 mm	56 x 56 x 31 mm
Weight	160 g	190 g	200 g
IMU			
	Accelerometer	Gyroscopes	Magnetometer
Range	±15 g	±490 °/s	±2.5 Gauss
In-Run Bias Stability	< 10 µg	< 1 °/hr	-
Noise Density	0.04 mg/√Hz	3.24 °/hr /√Hz	140 µGauss/√Hz
Bandwidth	240 Hz	240 Hz	200 Hz
GNSS (VN-210 and VN-310 Only)			
Receiver Type	72 Channels, L1, GNSS	Time-to-First-Fix (Cold/Warm Start)	26 s
Update Rate	5 Hz	Time-to-First-Fix (Hot Start)	< 1 s
Altitude Limit	50,000 m	Velocity Limit	500 m/s
ENVIRONMENT		INTERFACING	
Operating Temperature	-40°C to +85°C	IMU	Up to 800 Hz
Storage Temperature	-40°C to +85°C	INS Solution	Up to 400 Hz
IP Rating	IP 68 per IEC 60529	Primary Interface ²	RS-422 + 3 sync I/O pins
ELECTRICAL		Expansion Interface ²	RS-422
Power Consumption	< 2.5 watts	Serial Protocols	VectorNav binary and ASCII protocols, NMEA
Input Voltage	+12 to +36 VDC (MIL-STD-1275E)	Power & I/O Connectors	Circular Push-Pull Fischer UltiMate (10-pin x 2; Size 7)
		GNSS RF Connectors	SMA

Figura 1.8 - Specifiche per sistemi di navigazione inerziale della serie Tactical, NavtechGPS [1.12].

NAVIGATION		GNSS		COMMUNICATION			
Horizontal Position Accuracy	2.0 m	Model	u-blox M8P	Interface	RS232 (RS422 optional)		
Vertical Position Accuracy	3.0 m	Supported Navigation Systems	GPS L1 GLONASS L1 GALILEO E1 BeiDou L1	Speed	4800 to 2M baud		
Horizontal Position Accuracy (with RTK)	0.02 m	Supported SBAS Systems	WAAS EGNOS MSAS GAGAN QZSS	Protocol	AN Packet Protocol or NMEA		
Vertical Position Accuracy (with RTK)	0.03 m	Update Rate	10 Hz	Peripheral Interface	2x GPIO and Auxiliary RS232		
Horizontal Position Accuracy (Kinematic Post Processing)	0.01 m	Cold Start Sensitivity	-148 dBm	GPIO Level	5 V or RS232		
Vertical Position Accuracy (Kinematic Post Processing)	0.02 m	Tracking Sensitivity	-160 dBm	GPIO Functions	1PPS Odometer Stationary Pilot Tube NMEA input/output Navatel GNSS input Trimble GNSS input AN Packet Protocol input/output Packet Trigger Input Event Input Custom (Contact us)		
Velocity Accuracy	0.05 m/s	Hot Start First Fix	1 s				
Roll & Pitch Accuracy	0.1 °	Cold Start First Fix	26 s				
Heading Accuracy (Dynamic with GNSS)	0.2 °	Horizontal Position Accuracy (with SBAS)	2.5 m				
Heading Accuracy (Magnetic Only)	0.8 °	Horizontal Position Accuracy (with L1 RTK)	1 m				
Roll & Pitch Accuracy (Kinematic Post Processing)	0.04 °	Horizontal Position Accuracy (with L1 RTK)	0.02 m				
Heading Accuracy (Kinematic Post Processing)	0.08 °	Velocity Accuracy	0.05 m/s				
Heave Accuracy	5 % or 0.05 m (whichever is greater)	Timing Accuracy	30 ns				
Orientation Range	Unlimited	Acceleration Limit	4 g				
Hot Start Time	500 ms	SENSORS					
Internal Filter Rate	1000 Hz	SENSOR	ACCELEROMETERS	GYROSCOPES	MAGNETOMETERS	PRESSURE	
Output Data Rate	Up to 1000 Hz	Range (Dynamic)	2 g 4 g 16 g	250 °/s 500 °/s 2000 °/s	2 G 4 G 8 G	10 to 120 KPa	
Latency	0.4 ms	Bias Instability	20 µg	3 °/hr	-	10 Pa	
HARDWARE		Initial Bias	< 5 mg	< 0.2 °/s	-	< 100 Pa	
Operating Voltage	5 to 36 V	Initial Scaling Error	< 0.06 %	< 0.04 %	< 0.07 %	-	
Input Protection	±60 V	Scale Factor Stability	< 0.06 %	< 0.05 %	< 0.09 %	-	
Power Consumption (Typical)	0.5 W	Non-linearity	< 0.05 %	< 0.05 %	< 0.08 %	-	
Hot Start Battery Capacity	> 48 hrs	Cross-axis Alignment Error	< 0.05 °	< 0.05 °	< 0.05 °	-	
Hot Start Battery Charge Time	30 mins	Noise Density	100 µg/√Hz	0.004 °/s/√Hz	210 µG/√Hz	0.56 Pa/√Hz	
Hot Start Battery Endurance	> 10 years	Bandwidth	400 Hz	400 Hz	110 Hz	50 Hz	
Operating Temperature	-40 °C to 85 °C						
Environmental Protection	IP67 MIL-STD-810G						
MTBF	310,000 hrs						
Shock Limit	2000 g						
Dimensions(excluding tabs)	30x30x24 mm						
Dimensions(including tabs)	30x40.6x24 mm						
Weight	37 grams						

Figura 1.9 - Specifiche per sistema di navigazione inerziale Spatial, Advanced Navigation [1.13].

Sensor specification			
		Gyroscopes	Accelerometers
Standard full range		+/- 450 °/s	+/- 20 g
Initial bias error		0.2 °/s	5 mg
In-run bias stability		10 °/h	15 µg
Bandwidth (-3 dB)		415 Hz	375 Hz
Noise density		0.01 °/s/√Hz	60 µg/√Hz
g-sensitivity (calibrated)		0.003 °/s/g	N/A
Non-orthogonality		0.05 deg	0.05 deg
Non-linearity		0.01%	0.1%
		Magnetometer	Barometer
Standard full range		+/- 8 G	300-1100 hPa
Total RMS noise		0.5 mG	3.6 Pa
Non-linearity		0.2%	N/A
Resolution		0.25 mG	8cm (sea level, 15 °C)
GNSS receiver			
Receiver type	72ch with SBAS; GPS, QZSS, Galileo, GLONASS, BeiDou	Horizontal accuracy (CEP)	2.0 m 2.5 m w/o SBAS
Start-up time cold start	26 s	Vertical accuracy (CEP)	5 m
Tracking sensitivity	-167 dBm	Velocity accuracy (@30 m/s)	0.05 m/s

Figura 1.10 - Specifiche per sistema di navigazione inerziale serie MTi-G-710, XSens [1.14].

1.3 – Metodologie impiegate per la localizzazione subacquea

La tecnologia impiegata per la realizzazione del sistema risulta essere fine a sé stessa se non supportata da adeguate metodologie che permettano di comprendere, collegare e meglio utilizzare tutte le informazioni disponibili. Come accennato, è necessario tenere conto di numerosi modelli che descrivono il comportamento e i legami fra le varie grandezze misurate. Inoltre, bisogna discernere le metodologie impiegate e che meglio si congegnano con la strumentazione. Questo non risulta essere una conseguenza dell'equipaggiamento scelto ma una decisione mutuale dei due aspetti del progetto, quello teorico e quello pratico.

1.3.1 – Metodi di misura della distanza

Nell'ambito delle tecniche di localizzazione acustica descritti nella sezione 1.1, gli algoritmi impiegati possono essere suddivisi in due classi [1.03]: quelli basati sulla distanza e quelli indipendenti dalla distanza. I primi utilizzano strumenti addizionali per stimare la posizione dei trasduttori acustici sulla mappa attraverso la distanza assoluta punto-punto o le informazioni angolari. Si hanno quindi:

- **indicatore della forza del segnale ricevuto (RSSI):** è una tecnica che si basa sulla forza del segnale ricevuto per determinare la distanza fra due nodi. L'informazione sulla forza del segnale può essere raccolta per generare una curva che rappresenta la relazione fra la distanza e la forza del segnale ricevuto. È chiaro che la forza del segnale è dissipata con l'aumentare della distanza percorsa. Teoricamente la forza del segnale è inversamente proporzionale al quadrato della distanza.
- **tempo di arrivo (TOA):** misura la distanza fra due trasduttori che svolgono una misurazione del tempo. Il tempo che è richiesto dal segnale originato da un emettitore

per raggiungere il ricevitore può essere usato per stimare la distanza fra i due in funzione della velocità di propagazione. Questo metodo richiede che i due nodi sensori siano sincronizzati e che il pacchetto contenente l'informazione del tempo sia stampato esattamente all'istante di invio. Il sistema di localizzazione studiato impiega questa metodologia per il calcolo della distanza.

- differenza del tempo di arrivo (TDOA): utilizza la differenza del tempo di arrivo di segnali multipli inviati dal sensore nodo emettitore nello stesso istante temporale. A tal fine il sensore deve essere tale da poter inviare simultaneamente più segnali che propagano a velocità differenti. La distanza può essere misurata ricercando la differenza nel tempo di propagazione e la differenza nella velocità di propagazione dei due segnali.
- angolo di arrivo (AOA): piuttosto che calcolare la distanza tra due trasduttori, la tecnica AOA stima l'angolo al quale il segnale è ricevuto e utilizza le relazioni geometriche per calcolare la posizione del nodo sensore. Generalmente è utilizzato un metodo di triangolazione per calcolare la direzione dalla quale il nodo sensore riceve il segnale.

La tecnica TOA impiega la differenza di tempo che è più robusta in confronto alla tecnica dell'indicatore di forza del segnale. Questo rende la tecnica TOA capace di stime più accurate, tuttavia, è necessaria una perfetta sincronizzazione fra i trasduttori se si vuole utilizzarla nella forma one-way travel time. La tecnica TDOA non richiede nessuna sincronizzazione di mittente e ricevitore e può migliorare l'accuratezza della stima della distanza. È risultata una accuratezza di 2cm con una baseline fra i nodi di 3m [1.03] utilizzando segnali ad ultrasuoni. È necessario impiegare però strumenti più costosi e la sua accuratezza potrebbe non giustificare gli oneri. La tecnica AOA sperimentalmente mostra 5° di errore [1.03] e richiede anch'essa strumentazione aggiuntiva fra due nodi sensori e una distanza minima, rendendola quella meno preferibile.

1.3.2 - Modello di propagazione del suono nell'acqua

Uno dei passi fondamentali è quello di stimare la distanza fra i trasduttori tramite l'utilizzo dei modem acustici che inviano un segnale che si propaga nell'acqua. La trasmissione del suono nell'acqua è influenzata dalle proprietà del fluido. Si vuole ottenere la distanza percorsa conoscendo il tempo di volo ma questo legame non è diretto, bisogna quindi stabilire un modello che tenga conto dei vari fattori rilevanti. Due modelli semplificati sono forniti da *Mackenzie* [1.15] e *Coppens* [1.16], in funzione di temperatura, salinità e profondità.

$$\begin{aligned}
 \text{Equazione} \quad c(D, S, T) &= 1448.96 + 4.591 T - 5.304 * 10^{-2} T^2 + 2.374 * 10^{-4} T^3 + 1.340 \\
 \text{di} \quad & * (S - 35) + 1.630 * 10^{-2} D + 1.675 * 10^{-7} D^2 - 1.025 \\
 \text{Mackenzie} \quad & * 10^{-2} T * (S - 35) - 7.139 * 10^{-13} T * D^3 \quad (1.2)
 \end{aligned}$$

Con intervalli di validità: temperatura 2÷30 °C, salinità 25÷40‰, profondità 0÷8000 m.

$$\begin{aligned}
 \text{Equazione} \quad c(D, S, t) &= c(0, S, t) + (16.23 + 0.253t)D + (0.213 - 0.1t)D^2 \\
 \text{di} \quad & + [0.016 + 0.0002(S - 35)](S - 35)tD \\
 \text{Coppens} \quad c(0, S, t) &= 1449.05 + 45.7t - 5.21t^2 + 0.23t^3 + (1.333 - 0.126t \\
 & + 0.009t^2)(S - 35) \quad (1.3)
 \end{aligned}$$

Con intervalli di validità: temperatura 0÷35 °C, salinità 0÷45‰, profondità 0÷4000 m.

Dove:

- $t = T/10$ con T = temperatura in gradi Celsius;

- S = salinità ‰;
- D = profondità in km.

Lo standard UNESCO offre un algoritmo standard internazionale, dovuto a *Chen e Millero* [1.17] (1977), più complesso e in cui viene utilizzata la pressione piuttosto che la profondità:

Equazione UNESCO di Chen e Millero

$$c(S, T, P) = Cw(T, P) + A(T, P)S + B(T, P)S^{\frac{3}{2}} + D(T, P)S^2$$

$$Cw(T, P) = (C_{00} + C_{01}T + C_{02}T^2 + C_{03}T^3 + C_{04}T^4 + C_{05}T^5) + (C_{10} + C_{11}T + C_{12}T^2 + C_{13}T^3 + C_{14}T^4)P + (C_{20} + C_{21}T + C_{22}T^2 + C_{23}T^3 + C_{24}T^4)P^2 + (C_{30} + C_{31}T + C_{32}T^2)P^3$$

$$A(T, P) = (A_{00} + A_{01}T + A_{02}T^2 + A_{03}T^3 + A_{04}T^4) + (A_{10} + A_{11}T + A_{12}T^2 + A_{13}T^3 + A_{14}T^4)P + (A_{20} + A_{21}T + A_{22}T^2 + A_{23}T^3)P^2 + (A_{30} + A_{31}T + A_{32}T^2)P^3 \quad (1.4)$$

$$B(T, P) = (B_{00} + B_{01}T) + (B_{10} + B_{11}T)P$$

$$D(T, P) = D_{00} + D_{10}P$$

Con intervalli di validità: temperatura 0÷40 °C, salinità 0÷40‰, profondità 0÷1000 m.

Dove:

- t = T/10 con T = temperatura in gradi Celsius;
- S = salinità in Practical Salinity Units ‰;
- P = pressione in bar.

1.3.3 - Trilaterazione e triangolazione

Definita la distanza del tablet da ciascun beacon è necessario localizzarlo, proprio come avviene con i satelliti e i ricevitori GPS. La triangolazione è una tecnica che permette di calcolare distanze fra punti sfruttando le proprietà dei triangoli, fig. 1.11. La triangolazione geodetica è basata sulla determinazione, da una base di stazionamento, di tre valori fondamentali di un secondo punto del territorio: distanza in linea d'aria dalla stazione, angolo orizzontale, angolo zenitale. Sfrutta quindi la conoscenza degli angoli dei triangoli immaginari i cui vertici sono due punti noti e un punto incognito. Sfruttare la triangolazione consiste nell'applicare le formule trigonometriche.

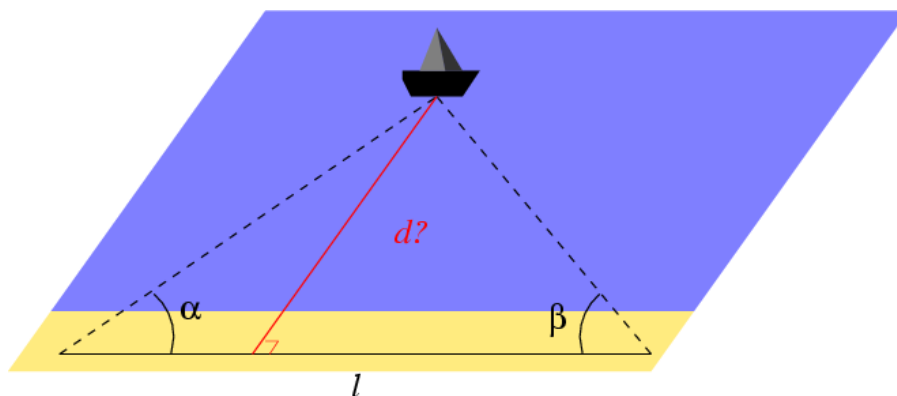


Figura 1.11 - Determinazione della posizione di un oggetto tramite triangolazione.

La trilaterazione, fig. 1.12, è un metodo per controllare estensioni, interruzioni e densificazioni che impiega strumenti elettronici di misurazione delle distanze per misurare le lunghezze dei lati di un triangolo piuttosto che angoli orizzontali, come nella triangolazione. Infatti, il principio fondamentale del metodo è quello dell'indeformabilità del triangolo. Date le misure dei lati ci sarà una sola configurazione che corrisponde alle misure date. Si conosce la posizione reciproca dei beacon e la distanza di questi dal tablet, la posizione del tablet è quindi univocamente determinata.

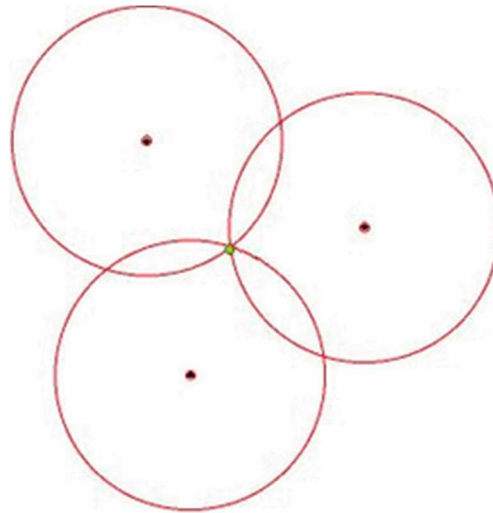


Figura 1.12 - Intersezione delle circonferenze di raggio pari alla distanza dei punti noti dall'oggetto da localizzare.

Nel caso spaziale al posto delle circonferenze si hanno delle sfere, come ad esempio per il sistema GPS (Fig. 1.13).

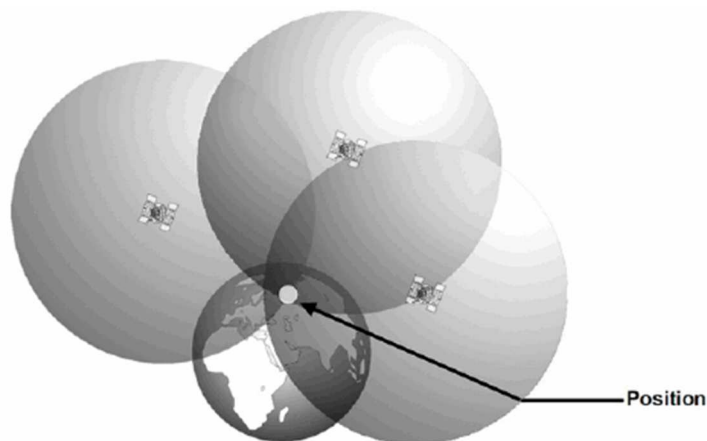


Figura 1.13 - Trilaterazione con satelliti e ricevitore terrestre.

La trilaterazione ha un rapporto costi-benefici vantaggioso e risulta, se ben eseguita, preferibile rispetto alla triangolazione perché altamente accurata e di esecuzione rapida. Al contrario, la triangolazione in alcune applicazioni presenta un numero di ridondanze maggiori rendendo più robusto il calcolo fornendo misure di verifica. Nella trilaterazione, essendo un metodo facile e affidabile, non sono richieste ridondanze e l'aggiunta di queste appesantirebbe il metodo riducendo il rapporto costi-benefici. Tenendo conto delle ridondanze geometriche, in uno schema di rilievo tramite trilaterazione e triangolazione, si ottiene la seguente tabella 1.1 [1.18]:

Tabella 1.1 – Caratteristiche in uno schema di rilievo tramite trilaterazione e triangolazione.

	Quadrilateral	Pentagon	Hexagon
Number of lines	6	10	15
Number of triangles	4	10	20
Number of triangles used in the computations	3	6	10
Number of check triangles	1	4	10
Number of geometrical conditions in trilateration	1	3	6
Number of geometrical conditions in triangulation	4	9	16

1.3.4 - Filtro di Kalman

Alla base di moltissimi sistemi di localizzazione subacquea c'è il filtro di Kalman [1.02] che è impiegato per la fusione dei dati per la localizzazione dell'obiettivo in modo quanto più accurato possibile e in tempo reale. Il filtro di Kalman è una tecnica di stima *Bayesiana* che utilizza le proprietà statistiche e deterministiche dei parametri del sistema e delle misure per ottenere uno stimatore ottimo (atto a minimizzare l'errore di stima). A tal fine, vengono fornite in input delle stime iniziali e ricorsivamente il filtro elabora e trasporta informazioni ad ogni iterazione, mantenendo un set di incertezze nelle sue stime e una misura delle correlazioni tra gli errori nelle stime dei diversi parametri, aggiornando le sue previsioni come media pesata dei valori precedenti e dei nuovi derivati dagli ultimi dati misurati. Per applicazioni in tempo reale, come la navigazione, l'approccio ricorsivo è più efficiente poiché solo le informazioni sulle nuove misurazioni necessitano di essere processate; quelle precedenti possono al contrario essere tralasciate.

1.3.4.1 - Elementi e fasi del filtro di Kalman

I cinque elementi chiave del filtro di Kalman sono:

- il vettore di stato e covarianza;
- il modello del sistema;
- il vettore delle misurazioni e covarianza;
- il modello di misurazione;
- l'algoritmo di implementazione.

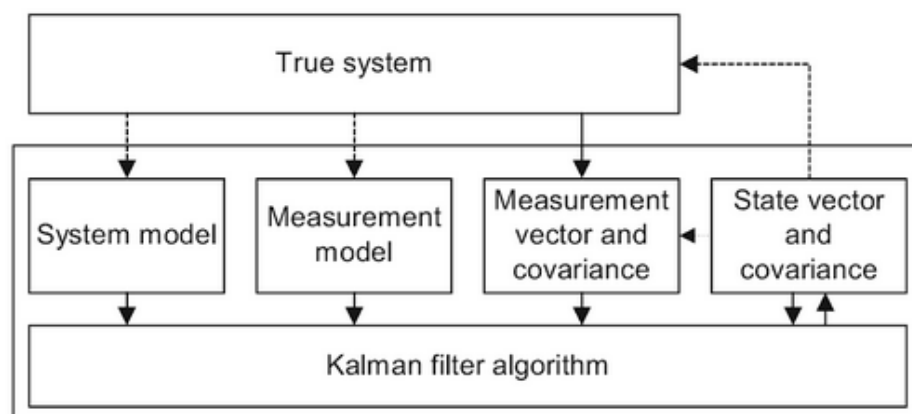


Figura 1.14 - Elementi del filtro di Kalman: la linea continua indica che il flusso di dati è sempre presente, la linea tratteggiata mostra invece che il flusso di dati è possibile in alcune applicazioni [1.02].

Il *vettore di stato* è costituito dai parametri descrittivi del sistema, conosciuti come stati, che il filtro di Kalman stima ad ogni iterazione. Associata al vettore di stato vi è la *matrice di covarianza degli errori*. Questa rappresenta le incertezze nelle previsioni di stato del filtro e il grado di correlazione fra gli errori nella stima. L'informazione della correlazione è importante dal momento che non vi è sempre sufficiente informazione proveniente dalle misurazioni per stimare indipendentemente gli stati del filtro di Kalman (mentre le correlazioni fra gli errori si possono sviluppare tra gli intervalli delle misurazioni). Il filtro di Kalman è un processo iterativo, quindi i valori iniziali del vettore di stato e della matrice di covarianza devono essere definiti dall'utente o determinati da un altro processo. Il *modello del sistema*, (o modello del processo) descrive come gli stati e la matrice di covarianza dell'errore del filtro di Kalman varino nel tempo. Il modello del sistema è deterministico per gli stati poiché è basato sulle proprietà note del sistema. L'incertezza dello stato dovrebbe essere incrementata nel tempo per tenere conto di cambiamenti non noti nel sistema che causano un discostarsi della stima dello stato del sistema in assenza di informazioni sulle nuove misurazioni. Questa variazione nei valori reali dello stato è nota come rumore del sistema, e le sue proprietà statistiche sono solitamente stimate a priori nella costruzione del filtro di Kalman. Il *vettore delle misurazioni* è composto da misurazioni simultanee delle proprietà del sistema, in funzione del vettore dello stato. Questa è l'informazione dalla quale derivano tutti gli stati previsti del sistema dopo l'inizializzazione. Associata al vettore delle misurazioni vi è la matrice di covarianza del rumore di misurazione, che descrive le statistiche del rumore di misurazione. Il *modello di misurazione* descrive come il vettore delle misurazioni varia in funzione del vettore di stato reale in assenza del rumore di misurazione. Come il modello del sistema, il modello delle misurazioni è deterministico ed è basato sulle conoscenze a priori del sistema. L'*algoritmo* del filtro di Kalman impiega il vettore delle misurazioni, il modello di misurazione e il modello del sistema per ottenere una stima ottima del vettore di stato. Vi sono numerosi step all'interno di un'iterazione, raggruppabili in due fasi: fase di *system-propagation* (previsione) e fase di *measurement-update* (correzione o aggiornamento). Lo scopo della fase di *system-propagation* (o *time-propagation*) è di predire lo stato del sistema e la matrice di covarianza degli errori, dal tempo di validità dell'ultima misurazione fino all'istante in cui vengono acquisite le nuove misurazioni, utilizzando le proprietà note del sistema. Nella fase di *measurement-update*, il vettore di stato stimato e l'errore di covarianza vengono aggiornati per incorporare le nuove informazioni sulle misurazioni. In fig. 1.16 viene illustrato qualitativamente in che modo il filtro di Kalman può determinare una previsione della posizione da misurazioni successive e incomplete. Nella prima iterazione (*Epoch 1*) si ha una stima della posizione con grande incertezza e la misurazione disponibile è una singola *linea di posizione* (LOP). Questa può essere derivante ad esempio da un misuratore di distanza a tempo di volo. La misurazione fornisce informazioni sulla posizione solo lungo la perpendicolare alla LOP: non si può quindi determinare in modo univoco la posizione da questa. Nella fase di *measurement-update* si aggiorna la stima sulla posizione in base alle informazioni ottenute dalla LOP, migliorandola, ma non ancora abbastanza.

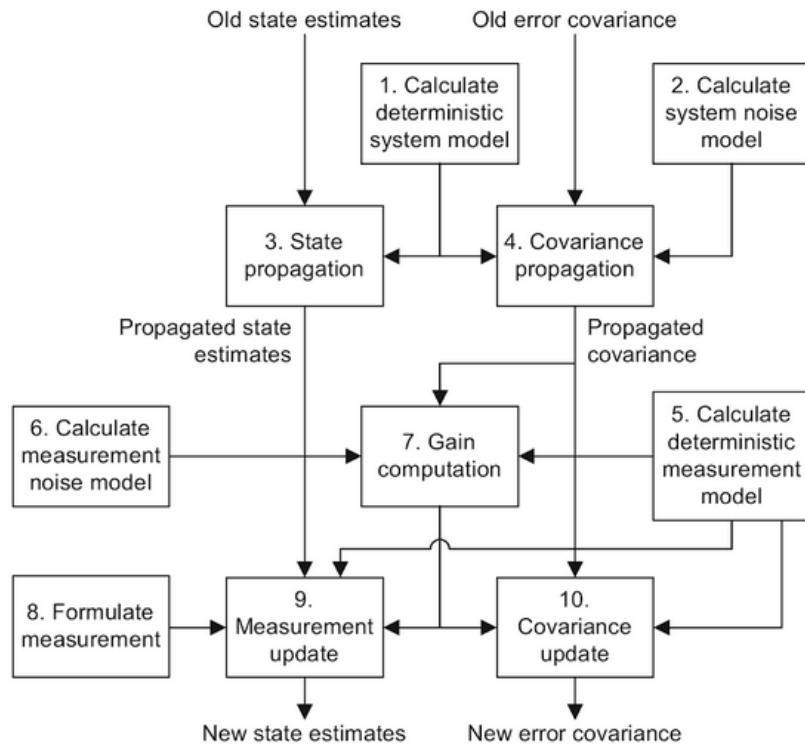


Figura 1.15 - Step dell'algoritmo del filtro di Kalman [1.02].

Nella seconda iterazione (*Epoch 2*), la previsione nella fase di system-propagation aumenta l'incertezza sulla posizione in tutte le direzioni tenendo conto della variazione della posizione dovuta ad uno spostamento. Anche in questo caso si ha un'informazione fornita da una LOP, però in direzione diversa. Questo permette di migliorare la stima dopo aver sfruttato tale informazione nella fase di correzione measurement-update.

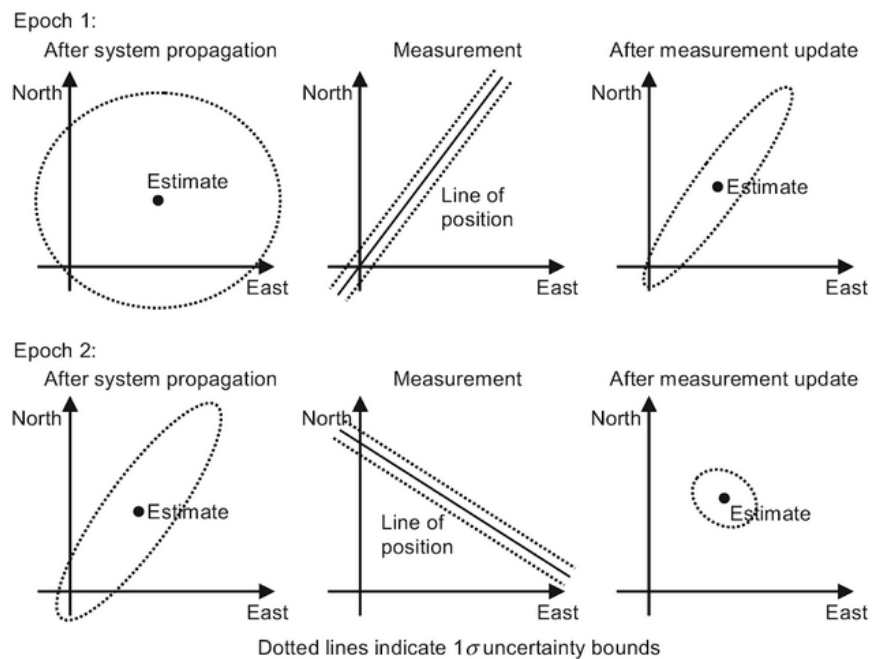


Figura 1.16 - Filtro di Kalman 2D: determinazione della posizione da due misurazioni incomplete [1.02].

1.3.4.2 - Algoritmo e modelli

Vengono ora specificati e definiti gli elementi utilizzati dall'algoritmo del filtro di Kalman. La classificazione secondo la variazione temporale degli errori modellati nel filtro è tale da specificare tre categorie, fig. 1.17:

- errori sistematici: assunti costanti, correlazione del 100% col tempo;
- rumori bianchi: non correlati col tempo;
- processi di Markov: che variano molto lentamente nel tempo.

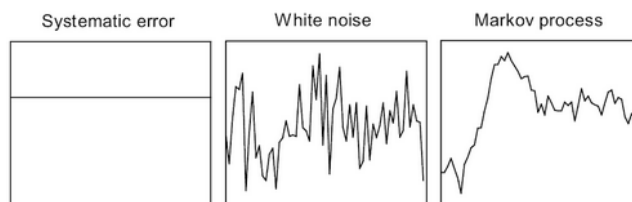


Figura 1.17 - Esempio di errore sistematico, rumore bianco, processo di Markov [1.02].

Per quanto riguarda il rumore bianco si ha che:

$$E(w_i(t_1) w_i(t_2)) = 0, t_1 \neq t_2 \quad (1.5)$$

I processi di Markov del primo ordine possono essere rappresentati come funzioni solo del loro precedente valore e rumore. Se queste proprietà sono note, allora può essere implementato in un filtro di Kalman. Un processo di Markov del primo ordine con una funzione di auto-correlazione con decadimento esponenziale, x_{mi} , può essere descritta come:

$$\frac{\partial x_{mi}}{\partial t} = -\frac{\partial mi}{\tau_{mi}} + w_i \quad (1.6)$$

Un'assunzione principale del filtro di Kalman è che gli errori del sistema devono essere modellati con una delle tre precedenti categorie. Possono anche essere combinazioni lineari o integrali di questi. Per esempio, il processo costituito da una camminata casuale è integrato con rumore bianco, mentre un errore su una accelerazione costante può portare a una crescita nel tempo dell'errore sulla velocità. Chiaramente, la modellazione tramite una di queste categorie introduce approssimazioni rispetto al caso di un problema di navigazione reale. Il set di parametri che devono essere stimati dal filtro di Kalman costituisce il vettore di stato indicato con x che è diverso dal vettore di parametri stimati dal filtro, indicato con \hat{x} . La stima di proprietà assolute del sistema (come la posizione, la velocità, ecc.) è detta implementazione *total-state*, mentre la stima degli errori in una misurazione fatta dal sistema come la posizione, la velocità di un sistema di navigazione integrata è detta implementazione *error-state*. In ogni caso, un vettore di stato può comprendere sia total-states che error-states. Il *vettore di stato residuo*, δx , è dato dalla differenza fra il vettore di stato reale e quello stimato dal filtro di Kalman.

$$\delta x = x - \hat{x} \quad (1.7)$$

In una implementazione del tipo error-state, il vettore di stato residuo rappresenta l'errore rimanente nel sistema dopo che le stime del filtro di Kalman sono state usate per correggerlo. La *matrice di covarianza dell'errore*, P , definisce l'aspettativa degli scarti quadratici del vettore di stato stimato dal vero valore del vettore di stato. Gli elementi sulla diagonale di P sono le varianze di ciascuno stato previsto, mentre le loro radici quadrata sono le incertezze.

$$P = E((\hat{x} - x)(\hat{x} - x)^T) = E(\delta x \delta x^T) \quad (1.8)$$

In un'implementazione error-state, tutte le stime del sistema sono inizializzate con valore nullo. I valori di inizializzazione della matrice di covarianza sono generalmente determinati in fase di progettazione del filtro di Kalman e indicano la confidenza nelle stime derivanti dai valori dello stato inizializzato. I valori dello stato inizializzato sono dunque definiti a priori, mentre i valori della matrice di covarianza indicano la confidenza in queste stime. Il vettore delle misurazioni, z , è un insieme di misurazioni delle proprietà del sistema, descritte dal vettore di stato. Questa comprende una funzione deterministica $h(x)$ e un rumore w_m . L'innovazione di misura, δz , è la differenza fra il vettore di misurazione reale e quello calcolato dal vettore dello stato stimato prima dell'aggiornamento delle misure.

$$\delta z^- = z - h(\hat{x}^-) \quad (1.9)$$

Mentre la differenza fra il vettore delle misurazioni reali e quello calcolato dal vettore di stato aggiornato:

$$\delta z^+ = z - h(\hat{x}^+) \quad (1.10)$$

Il filtro di Kalman standard assume che l'errore di misurazione forma una distribuzione gaussiana a media nulla, non correlata col tempo, e modella la deviazione standard con la matrice di covarianza del rumore di misurazione, R .

$$R = E(w_m w_m^T) \quad (1.11)$$

I termini sulla diagonale di R sono le varianze di ogni misurazione e i termini fuori la diagonale rappresentano la correlazione fra le differenti componenti del rumore di misurazione. Per molte applicazioni R è una matrice diagonale.

Algoritmo del filtro di Kalman

In figura 1.18 sono riportati gli step dell'algoritmo del filtro di Kalman.

- | |
|--|
| <ol style="list-style-type: none"> 1. Calculate the transition matrix, Φ_{k-1}. 2. Calculate the system noise covariance matrix, Q_{k-1}. 3. Propagate the state vector estimate from \hat{x}_{k-1}^+ and \hat{x}_{k-1}^-. 4. Propagate the error covariance matrix from P_{k-1}^+ to P_k^-. 5. Calculate the measurement matrix, H_k. 6. Calculate the measurement noise covariance matrix, R_k. 7. Calculate the Kalman gain matrix, K_k. 8. Formulate the measurement, z_k. 9. Update the state vector estimate from \hat{x}_k^- to \hat{x}_k^+. 10. Update the error covariance matrix from P_k^- to P_k^+. |
|--|

Figura 1.18 - Step dell'algoritmo del filtro di Kalman [1.02].

La matrice di transizione, Φ_{k-1} , definisce come il vettore dello stato cambi nel tempo in funzione della dinamica del sistema modellato dal filtro di Kalman. La matrice di transizione è sempre una funzione dell'intervallo di tempo τ_s tra le iterazioni del filtro di Kalman, e spesso è funzione di altri parametri; tuttavia, nella forma standard del filtro di Kalman essa non è funzione di nessuno degli stati. Se i parametri che caratterizzano questa matrice variano, questa deve essere ricalcolata

ad ogni iterazione. La *matrice di covarianza del rumore di sistema*, Q_{k-1} , definisce come l'incertezza delle previsioni degli stati incrementi col tempo a causa delle fonti di rumore nel sistema modellate nel filtro di Kalman (come ad esempio la dinamica non misurata e il rumore di strumentazione). È sempre una funzione dell'intervallo di tempo τ_s tra le iterazioni. È comune che questa matrice sia diagonale e costante, ma non è sempre così. La *matrice delle misurazioni*, H_k , definisce come il vettore delle misurazioni varia con il vettore dello stato. Nella navigazione, H_k è comunemente una funzione della cinematica definita dall'utente. La *matrice di covarianza del rumore di misurazione*, R_k , può essere assunta costante o modellata come una funzione della cinematica. La *matrice di guadagno di Kalman*, K_k , è usata per definire il peso delle informazioni sulle misurazioni aggiornando le previsioni dello stato. È funzione del rapporto fra l'incertezza della misurazione reale, z_k , e l'incertezza della misurazione predetta dagli stati stimati, $H_k \hat{x}^-$.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (1.12)$$

L'informazione sulla correlazione nei termini fuori dalla diagonale della matrice P_k accoppia il vettore delle misurazioni agli stati che non sono direttamente legati tramite la matrice H_k . Successivamente va formulato il vettore delle misurazioni z_k . In alcune applicazioni questo vettore ha le componenti già specificate nel sistema modellato dal filtro di Kalman. In altre applicazioni z_k deve essere calcolato come funzione di altri parametri del sistema.

Comportamento del filtro di Kalman

Le incertezze dello stato sono le radici della diagonale della matrice di covarianza P . Inizialmente, quando le incertezze dello stato sono grandi, il guadagno di Kalman sarà grande, pesando gli stati previsti verso i nuovi dati delle misurazioni. Le previsioni del filtro di Kalman cambieranno velocemente al convergere ai valori reali degli stati, così facendo l'incertezza dello stato diminuirà rapidamente. Tuttavia, assumendo una varianza del rumore di misurazione costante, R , si ha un calo del guadagno di Kalman, pesando la stima dello stato più verso i valori precedenti. Ne deriva che la velocità alla quale gli stati cambiano si riduce e di conseguenza la riduzione nell'incertezza degli stati rallenta.

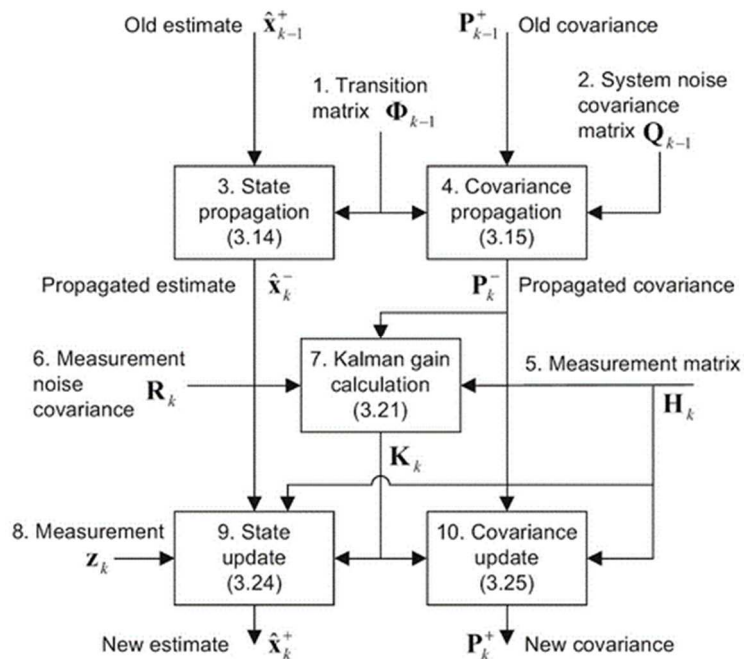


Figura 1.19 - Flusso dei dati nel filtro di Kalman [1.02].

La velocità alla quale la stima di uno stato converge dipende dall'*osservabilità dello stato*. L'osservabilità di molti parametri dipende dalla dinamica. Così come la relazione fra le misurazioni e gli altri stati, la velocità di convergenza dello stato di un filtro di Kalman dipende dalla frequenza di campionamento delle misurazioni, dall'intensità e dalle proprietà di correlazione del rumore di misura e dal livello del rumore del sistema. Questa è anche nota come *osservabilità stocastica*. Sistema e rumore di misurazione possono mascherare gli effetti di quegli stati che hanno solamente un basso impatto sulle misurazioni, rendendo tali stati effettivamente inosservabili. Il flusso dei dati nel filtro di Kalman è rappresentato in fig. 2.16 mentre l'incertezza dello stato del filtro di Kalman durante la convergenza è rappresentato in fig. 1.20.

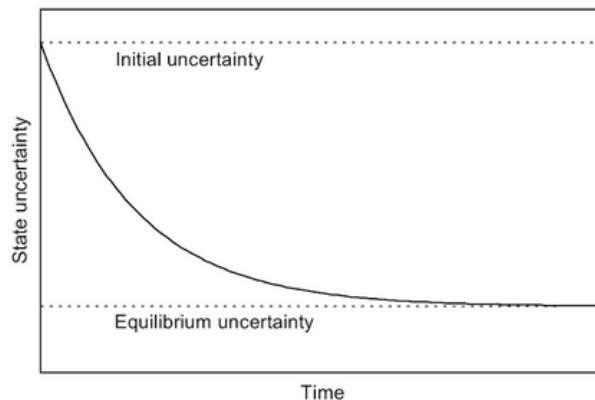


Figura 1.20 - Incertezza dello stato del filtro di Kalman durante la convergenza [1.02].

Modello del sistema

Per propagare il vettore dello stato stimato, \hat{x} , e la covarianza dell'errore, P , avanti nel tempo, è necessario conoscere come questi stati varino nel tempo. Questo è funzione del modello del sistema. Un'assunzione del filtro di Kalman è che la derivazione nel tempo di ciascuno stato è una funzione lineare degli altri stati e di fonti di rumore bianco.

$$\dot{\hat{x}}(t) = F(t) \hat{x}(t) + G(t)w_s(t) \tag{1.13}$$

Dove $x(t)$ è il vettore di stato reale al tempo t , $w_s(t)$ è il vettore del rumore di sistema, $F(t)$ è la matrice del sistema e $G(t)$ è la matrice di distribuzione del rumore di sistema. $F(t)$ e $G(t)$ sono delle funzioni sempre note e per determinare il modello del sistema, tali funzioni devono essere derivate dalle proprietà note del sistema.

Modello delle misure

In un filtro di Kalman standard, il vettore delle misurazioni, $z(t)$, è modellato come una funzione lineare del vettore di stato reale, $x(t)$, e dalle fonti di rumore bianco $w_m(t)$.

$$z(t) = H(t) x(t) + w_m(t) \tag{1.14}$$

$H(t)$ è la matrice delle misurazioni ed è determinata dalle proprietà note del sistema.

1.3.4.3 - Problematiche di implementazione

Messa a punto (tuning) e stabilità

Il tuning del filtro di Kalman si basa principalmente sulla definizione, da parte dell'utente, delle matrici: di covarianza del rumore di sistema Q_k , di covarianza del rumore delle misurazioni R_k , i

valori iniziali della matrice di covarianza degli errori P_0^+ . È importante scegliere opportunamente i valori da inserire in tali matrici. Se i valori selezionati sono troppo piccoli, l'errore effettivo nel filtro di Kalman sarà molto più grande rispetto all'incertezza dello stato ottenuta da P . In contrapposizione, se i valori scelti sono troppo grandi, l'incertezza riportata sarà troppo grande. In ogni caso, il parametro critico nel filtraggio di Kalman è il rapporto fra le matrici di covarianza dell'errore P_k e delle misurazioni R_k , poiché con esse si determina il guadagno di Kalman. Se P/R è sottostimata, il guadagno di Kalman sarà piccolo e gli stati predetti convergeranno con la loro controparte reale molto più lentamente del necessario. Viceversa, se P/R è sovrastimato, il guadagno di Kalman sarà troppo grande e questo può portare a previsioni instabili dovute al rumore delle misurazioni che avranno un'influenza maggiore sulle previsioni. Nella pratica conviene mettere a punto il filtro in modo da ottenere una incertezza dello stato di 1σ , o comunque sostanzialmente più grande della corrispondente deviazione standard dell'errore al fine di mantenere la stabilità.

Nel filtro di Kalman vi sono numerose fonti di approssimazione:

- gli stati d'errore più piccoli vengono spesso ignorati a causa dell'osservabilità del problema;
- il modello del sistema o delle misurazioni può avere delle approssimazioni al fine di avvicinarsi ai requisiti di linearità delle equazioni del filtro di Kalman;
- le proprietà stocastiche di stati che variano nel tempo lentamente sono spesso sovrasemplificate;
- tutte le fonti di rumore assunte dal filtro di Kalman sono di tipo bianco, mentre nella realtà queste esibiscono un comportamento correlato col tempo dovuto agli effetti di limitatezza della banda.

Un buon metodo di tuning è quello di fissare P_0^+ e Q_k e variare R_k con una procedura semplicemente di *trial and error* per individuare il più piccolo valore che fornisce stime dello stato stabili. Se questo non bastasse, si procede con la variazione anche di P_0^+ e Q_k .

Problemi numerici

Quando un filtro di Kalman viene implementato in un computer, la precisione è limitata dal numero di bits usati per memorizzare e processare ciascun parametro. Più a lungo il filtro di Kalman elabora e maggiore è la velocità delle iterazioni, maggiore dunque sarà la distorsione nella matrice P . Questa distorsione si manifesta come una rottura della simmetria rispetto alla diagonale e in alcuni casi può persino produrre termini negativi sulla diagonale che corrispondono a incertezze immaginarie. Grossi errori sulla matrice P distorcono il guadagno di Kalman K . Guadagni troppo piccoli producono previsioni di stato non pronte, guadagni troppo elevati producono previsioni di stato non stabili e oscillanti. P deve essere necessariamente definita positiva. La simmetria della matrice P può essere mantenuta forzatamente mediando la matrice con la sua trasposta dopo ogni step di aggiornamento delle misurazioni e propagazione del sistema.

1.3.4.4 - Estensioni al filtro di Kalman.

La tecnica del filtraggio di Kalman può essere estesa per gestire misurazioni o modelli di sistema non lineari, rumore correlato al tempo, sistemi sconosciuti o misure con distribuzioni non gaussiane. Queste caratteristiche sono quelle che si accostano maggiormente ai problemi reali che dovrebbero essere notevolmente approssimati per poter essere descritti da modelli lineari. In un filtro di Kalman standard, il modello della misurazione è assunto lineare (per esempio, il vettore della misurazione, z , è una funzione lineare del vettore dello stato, x). Non è sempre così nei

sistemi reali. In alcune applicazioni, come in molti problemi di calibrazione e allineamento dei sistemi di navigazione integrata, una approssimazione lineare del modello di misurazione è utile, sebbene questo può introdurre piccoli errori. Una versione non lineare del filtro di Kalman è l'EKF (*Extended Kalman Filter*). In un EKF, la matrice del sistema, F , e la matrice delle misure, H , può essere sostituita nella propagazione dello stato e nelle equazioni di aggiornamento da funzioni non lineari del vettore dello stato, rispettivamente $f(x)$ e $h(x)$.

$$\hat{x}(t) = f(x(t)) + G(t) w_s(t) \quad (1.15)$$

Nell'EKF, si assume che l'errore nella stima del vettore dello stato è molto più piccolo del vettore dello stato, abilitando un modello di sistema lineare da essere applicato al *residuo* del vettore dello stato.

$$\delta\dot{x}(t) = F(t) \delta x(t) + G(t) w_s(t) \quad (1.16)$$

Una conseguenza della linearizzazione di F e H è che la matrice di covarianza dell'errore, P , e il guadagno di Kalman K , sono funzioni delle previsioni dello stato. Questo può occasionalmente causare problemi di stabilità e l'EKF è più sensibile al tuning dell'inizializzazione della matrice P rispetto al filtro di Kalman standard. Un'alternativa che mantiene una covarianza dell'errore e del guadagno di Kalman indipendenti dalle previsioni dello stato, è il filtro di Kalman Linearizzato.

Capitolo 2

Il sistema di localizzazione acustica subacquea del progetto BLUEMED.

L'obiettivo di questo capitolo è quello di presentare e descrivere le caratteristiche del progetto BLUEMED, essenziale punto di partenza per lo sviluppo delle attività svolte durante il percorso di dottorato industriale. La conoscenza preliminare di tutti gli aspetti tecnici (meccanici, elettronici ed informatici) della strumentazione del sistema è stata di grande importanza per poter inizialmente procedere con l'integrazione meccanica ed elettrica del sistema SBL sul veicolo di superficie, descritta nel paragrafo 2.3; la conoscenza del protocollo di comunicazione tra di essi è stata indispensabile per poter successivamente gestire l'integrazione software tra i dispositivi e l'implementazione degli algoritmi di localizzazione, le quali descrizioni sono riportate nei capitoli successivi.

2.1 - L'evoluzione dal progetto VISAS al progetto BLUEMED: contesto d'applicazione e tecniche operative.

Nell'ambito del progetto BLUEMED [2.01] è stato sviluppato un servizio di *augmented diving* per la visita di musei subacquei partendo da un prototipo che era stato precedentemente realizzato nel progetto VISAS [I.16].

In VISAS, infatti, era stato sviluppato un sistema di posizionamento subacqueo a basso costo, costituito da un sistema di 4 beacon, come in fig. 2.1, che possono essere fissi o mobili ed equipaggiati con un modem acustico (tecnica LBL). Tale sistema è stato impiegato per stimare la posizione di un tablet di supporto ai sub che visitano siti archeologici subacquei.

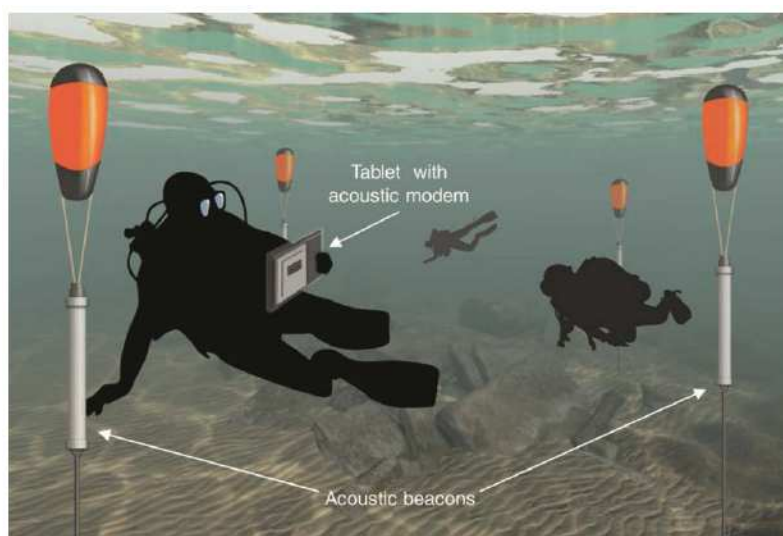


Figura 2.1 - Sistema di posizionamento VISAS [I.16].

L'ambiente marino introduce delle complicazioni che vanno ad inficiare sulla stima della localizzazione del tablet:

- la posizione dei beacon è soggetta a moto ondoso;
- l'installazione dei beacon avviene in un ambiente ostico che accentua gli errori di posizionamento;
- il mezzo è disomogeneo a causa delle correnti marine e le sue proprietà variano localmente, rendendo le approssimazioni del modello riguardante l'ambiente più grossolane;
- è possibile l'insorgenza di ostruzioni che bloccano la trasmissione dell'informazione.

Per migliorare l'accuratezza e la stabilità nel caso di perdita di segnale da uno o più beacon, l'algoritmo di fusione delle informazioni implementato nel tablet permette di sfruttare, come informazione aggiuntiva in modo vantaggioso, un sensore di profondità installato sul tablet stesso. Il sistema di localizzazione del tablet previsto nel progetto VISAS riceve ciclicamente le posizioni dei beacon e, tramite la loro sincronizzazione col tablet, riesce a determinare la propria posizione sfruttando la misurazione del tempo di volo e l'algoritmo di trilaterazione. Queste informazioni sono utili al software di navigazione per mostrare la posizione del tablet nella mappa 3D del sito archeologico.

Il progetto BLUEMED, a differenza di VISAS, coinvolge più di un tablet all'interno dello stesso sito archeologico ed inoltre si è voluto adottare una configurazione SBL, ovvero con i beacon installati a distanza ravvicinata su un veicolo di superficie capace di inseguire l'oggetto da localizzare. Questo permette il dispiegamento in modo molto più veloce e versatile rispetto al caso LBL, ma soprattutto non richiede installazioni sul fondale, preservando l'ambiente in virtù del fatto che le applicazioni maggiori sono proprio in siti di interesse archeologico.

Sia il sistema LBL usato in VISAS che l'SBL usato in BLUEMED sono stati sviluppati dalla Applicon s.r.l. una spin-off dell'Università della Calabria.

2.2 - Strumentazione utilizzata nel progetto BLUEMED

Gli elementi essenziali utilizzati nel sistema BLUEMED sono:

- il sistema SBL (trasmettitore e ricevitore acustici);
- il tablet subacqueo;
- il veicolo autonomo di superficie (*ASV - Autonomous Surface Vehicle*).

2.2.1 – Il sistema SBL

Il sistema di posizionamento SBL ha molti vantaggi rispetto all'LBL. Prima di tutto, come precedentemente descritto, può essere montato sul lato inferiore di una nave, su un piccolo veicolo di superficie o su un boa galleggiante. Infatti, può essere semplicemente dispiegato dalla superficie e non richiede ai subacquei di installare i transponder sul fondale come nel sistema LBL. Il principale svantaggio dei sistemi di posizionamento LBL (come nel caso di VISAS), infatti, è il tempo necessario per calibrare e installare i transponder.

Gli SBL installati vicino la superficie possono utilizzare un GPS per ottenere la posizione e un segnale di temporizzazione preciso, in modo che non sia necessario geo-referenziare con precisione ogni singolo transponder. Inoltre, qualsiasi numero di ricevitori può stimare la propria posizione nella stessa area se tutti i componenti del sistema SBL sono sincronizzati nel tempo e se sono utilizzati protocolli di tipo "one-way ranging". Si possono infatti stimare le distanze relative tra beacon e subacquei misurando il tempo di viaggio a due vie "two-way travel" (TWTT)

o il tempo di viaggio unidirezionale “one-way travel time” (OWTT) del segnale acustico. Anche se il vantaggio del primo metodo è che non è richiesto un orologio con precisione assoluta per la misurazione del tempo di viaggio, TWTT non lavora bene se aumenta il numero di tablet coinvolti nel processo di “ranging”. Invece, l'approccio OWTT si adatta perfettamente a qualsiasi numero di subacquei poiché tutti i tablet possono ricevere aggiornamenti sul ranging contemporaneamente. Infatti, tutti i tablet in grado di misurare il OWTT possono stimare le proprie distanze relative rispetto al mittente senza rispondere con un segnale acustico, riducendo così il sovraccarico della rete. Il principale svantaggio di questo approccio è che gli orologi dei nodi coinvolti devono essere sincronizzati per tutta la durata della missione.

Anche se potrebbe essere facile utilizzare l'ora GPS su beacon che consentono la sincronizzazione con ciascuno altro senza deriva, è molto più difficile sul tablet subacqueo. In effetti, lo standard reale degli orologi a tempo integrati nelle piattaforme embedded hanno derive e perdite di clock elevate e perdono la sincronizzazione dopo pochi minuti. Quindi, anche se gli orologi integrati sono sincronizzati fuori dall'acqua con il GPS, una volta in acqua la sincronizzazione si perde dopo poco tempo. Come rimedio, devono essere utilizzati orologi ad alta precisione per stimare l'OWTT, come Chip Scale Atomic Clocks (CSAC) [2.02] che hanno una deriva trascurabile su lunghi periodi di tempo, oppure altri tipi di orologi esterni altamente precisi. Dei buoni risultati utilizzando CSAC sono stati raggiunti in [2.03].

Il sistema di localizzazione SBL è quindi costituito da un'unità di superficie e alcuni ricevitori subacquei. La configurazione selezionata è di tipo invertito, in cui l'unità di superficie non conosce le posizioni del ricevitore ed è ogni ricevitore a calcolare la sua posizione rispetto all'unità di superficie, quindi rispetto al trasmettitore acustico. Infine, la posizione di almeno un ricevitore, insieme ad altri dati di stato, possono essere inviati periodicamente all'unità di superficie mediante la funzione “modem”.

Il sistema, come precedentemente detto, utilizza il metodo OWTT. In particolare, l'unità di superficie invia ciclicamente un messaggio acustico da ciascuno dei suoi trasduttori contenente i seguenti dati:

- ID del trasduttore che invia il messaggio;
- ora di invio del messaggio;
- posizione GPS con precisione centimetrica del trasduttore.

Questi messaggi vengono inviati in broadcast e vengono ricevuti quasi contemporaneamente da tutti i ricevitori nell'area di copertura. Il ricevitore confronta l'ora nel messaggio con il suo orologio interno, calcola la differenza e stima la distanza da ciascun trasduttore. Le distanze calcolate in questo modo vengono utilizzate per triangolare la posizione del ricevitore. Questo approccio richiede l'uso di orologi ad alta precisione integrati sia nell'unità di superficie SBL (trasmettitore) sia nei ricevitori. L'intero sistema è autonomo con il proprio pacco batteria e non richiede alimentazione elettrica esterna. Tuttavia, se necessario è anche possibile escludere la batteria e fornire l'alimentazione elettrica da una fonte esterna.

2.2.1.1 – Trasmettitore (unità di superficie SBL)

L'unità di superficie del sistema di posizionamento SBL ha quattro trasduttori acustici collegati alla scheda elettronica di controllo. L'unità di superficie è, inoltre, dotata di un GNSS Disciplined Oscillator ad alta risoluzione temporale, e di una piattaforma IMU GNSS/INS. L'unità di superficie è illustrata nella figura 2.2.

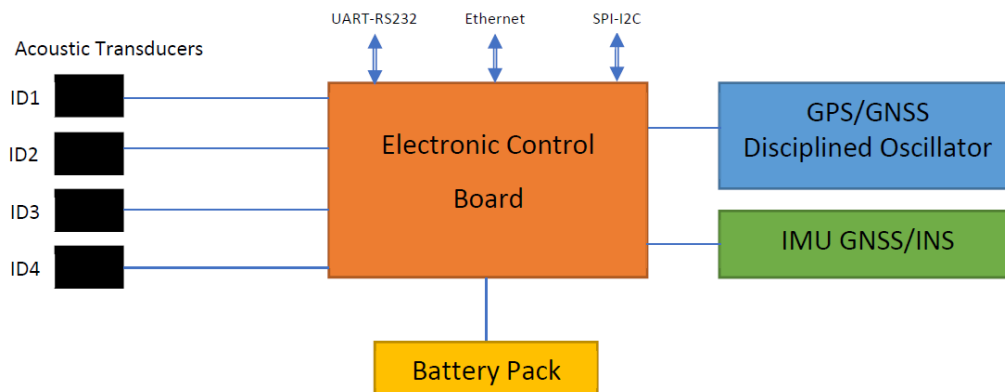


Figura 2.2 – Diagramma a blocchi del trasmettitore acustico SBL (unità di superficie).

I quattro trasduttori piezoelettrici, necessari per la comunicazione acustica con il ricevitore (o ricevitori), sono installati su un supporto personalizzato in una configurazione a croce distanziata di circa un metro, come mostrato nella seguente figura 2.3:

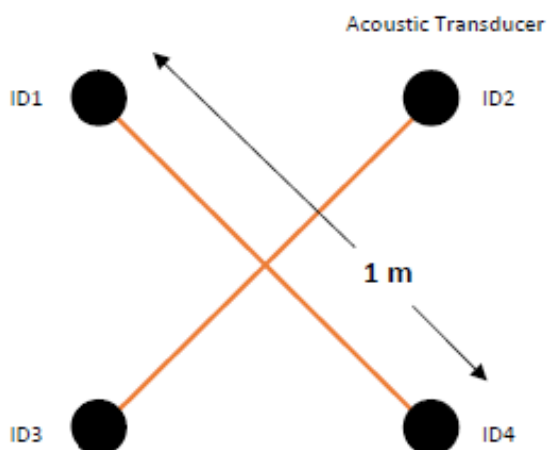


Figura 2.3 – Installazione dei trasduttori sull'unità di superficie.

Tale configurazione può essere facilmente installata sia su un veicolo di superficie autonomo che su un boa fissa. I test preliminari sono stati eseguiti in laboratorio utilizzando un supporto in alluminio fissato su una boa, come mostrato in figura 2.4.

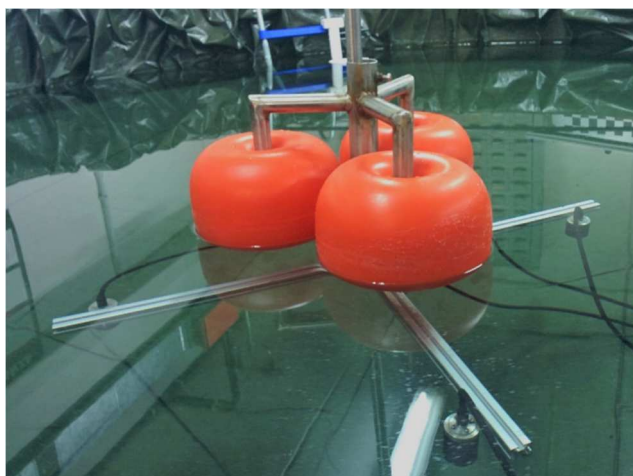


Figura 2.4 – Prototipo del supporto per trasduttori.

L'unità di superficie è dotata delle seguenti interfacce:

- 2 interfacce UART - RS-232;
- 1 interfaccia Ethernet;
- 1 interfaccia SPI;
- 1 interfaccia I2C.

Il sistema operativo dell'unità di superficie è Unix/Linux e garantisce una facile configurazione e personalizzazione. Le interfacce e i protocolli di comunicazione possono essere adattati ai requisiti dell'applicazione.

Il sistema di posizionamento utilizza il cosiddetto GPSDO (GPS Disciplined Oscillator). Questi orologi integrano un ricevitore GPS e un orologio con tecnologia OCXO (Oven Controlled XTAL Oscillators). La scheda di controllo SBL utilizza una comunicazione seriale con il GPSDO, per configurare e leggere tutti i parametri di controllo. In figura 2.5 è descritto lo schema a blocchi del dispositivo:

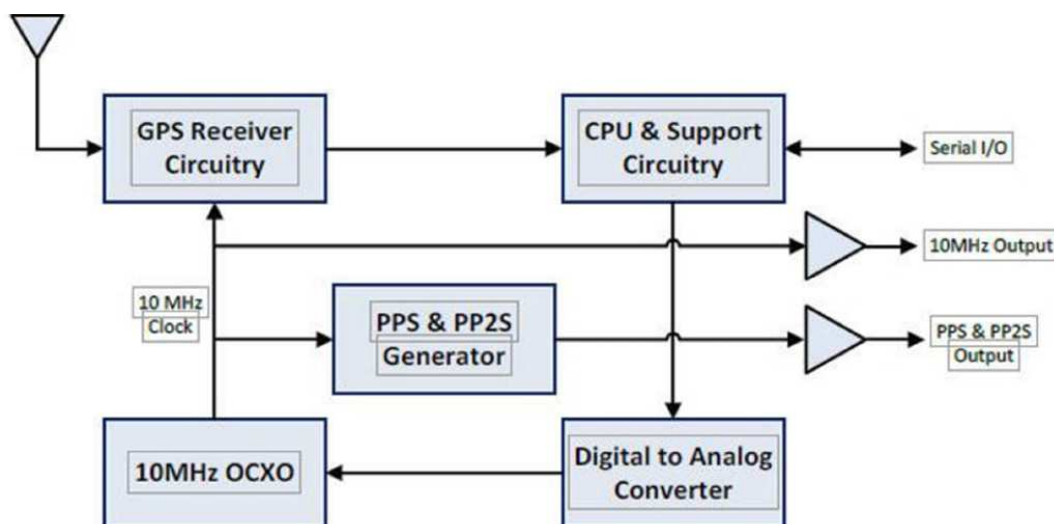


Figura 2.5 – Diagramma a blocchi del GPS/GNSS Disciplined Oscillator.

In un'unità OCXO, il cristallo e gli altri componenti sensibili alla temperatura sono installati in un case in cui un sensore di temperatura, un elemento riscaldante e un circuito di controllo forniscono un ambiente a temperatura stabilizzata, isolando così il cristallo dalle variazioni di temperatura esterna. Mantenendo il cristallo a una temperatura costante, le prestazioni dell'oscillatore e la sua precisione della frequenza sono migliorate.

Questa tecnologia permette di ottenere una precisione di sincronizzazione tra due differenti orologi fino a 50us/24h. I GPS Discipline Oscillator utilizzano il segnale orario ricevuto dai satelliti del GPS come fonte per il suo cristallo integrato, quindi tutti gli orologi che ricevono i segnali GPS sono sincronizzati con lo stesso riferimento. Nel sistema SBL, tutti gli orologi (unità di superficie e ricevitori) sono sincronizzati automaticamente prima dell'inizio di ogni missione, con il segnale ricevuto dall'antenna GPS integrata. Nessuna procedura di sincronizzazione o connessione cablata tra i componenti SBL sono necessari.

Dopo la sincronizzazione, la scheda di controllo elettronica utilizza il segnale di temporizzazione del GPSDO per inviare i pacchetti "range" ai ricevitori. In tabella 2.1 sono riportate le principali caratteristiche del dispositivo GPSDO integrato.

Tabella 2.1 – Principali caratteristiche del dispositivo GPSDO integrato.

Supply Voltage	3.7VDC	GNSS Reception Capability	GPS L1 C/A, GLONASS L1OF, QZSS L1 C/A, SBAS L1 C/A
Power Consumption	<0.45A	GNSS Reception	26 channels (GPS, GLONASS, QZSS, SBAS)
Operating Temperature	-40°C to +85°C	Sensitivity	GPS Tracking: -161 dBm Acquisition: -147 dBm GLONASS Tracking: -157 dBm Acquisition: -143 dBm
Outer Size	34mm x 27mm x 15.5mm	1PPS (Holdover) Output	1PPS Accuracy: $\pm 50\mu\text{s}/24\text{h}$
Protocol	eSIP (NMEA 0183 Standard Ver 4.10)	1PPS (Normal Status) Output	1PPS Accuracy: <math>< \pm 100\text{ns}</math> 1PPS Stability: <math>< 15\text{ns} (1\sigma)</math>

Il sistema di posizionamento satellitare GNSS (Global Navigation Satellite System) consente all'utente di calcolare le sue coordinate con precisione metrica ed elevata ripetibilità. Tuttavia, queste coordinate sono fortemente influenzate dalla qualità del segnale GNSS. Se si vuole raggiungere una posizione precisa e affidabile in qualsiasi ambiente, è necessario integrare il sistema GNSS con sensori inerziali INS (Inertial Navigation System). Questi sensori (generalmente chiamati IMU, acronimo di Inertial Measurement Unit) permettono di ottenere un posizionamento continuo ed anche robusto contro qualsiasi segnale GNSS perso a causa di ostacoli ambientali. Tuttavia, IMU accumula errori sistematici e accidentali dovuti alla deriva degli accelerometri e giroscopi. Questi errori generano incertezza di posizionamento che aumenta con il tempo trascorso senza segnale GNSS. L'uso integrato di questi due sensori consente di utilizzare la soluzione GNSS, quando disponibile, per ridurre gli errori a lungo termine dell'IMU e, allo stesso tempo, l'IMU è in grado di fornire una soluzione quando i dati GNSS non sono disponibili. Nella Figura 2.6 è riportato il diagramma a blocchi del dispositivo integrato.

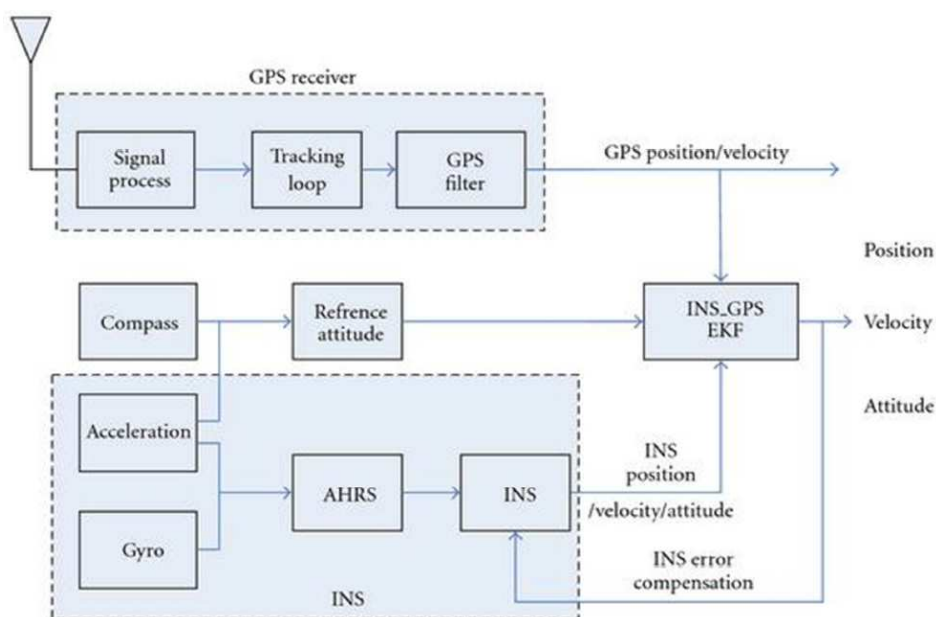


Figura 2.6 – Diagramma a blocchi del sistema GNSS/INS integrato.

L'IMU installata sull'unità di superficie SBL acquisisce informazioni sull'assetto della piattaforma e permette la correzione di eventuali errori dovuti ai movimenti dei trasduttori. Utilizzando questa informazione, la scheda elettronica di controllo conosce in tempo reale la posizione relativa di ciascun trasduttore. A questo punto l'unità di superficie può correggere qualsiasi movimento e inviare ciclicamente tramite ciascun trasduttore un messaggio acustico contenente la posizione, l'orario e l'ID del trasduttore.

Questo tipo di soluzione permette l'installazione del sistema anche su strutture galleggianti (es. boa fissa). Inoltre, se necessario, il sistema può ricevere tramite un'interfaccia Ethernet informazioni sull'assetto da un sensore esterno installato su un veicolo. Infatti, i messaggi inviati dall'IMU utilizzano due formati standard NMEA-0183: \$PRDID e \$GPGGA. Il primo invia i dati sull'orientamento dell'unità (roll, pitch e heading), il secondo la sua posizione GPS (latitudine e longitudine). Una descrizione più dettagliata di questi messaggi sarà fornita in seguito. Nella tabella 2.2 sono riportate le principali caratteristiche del dispositivo GNSS/INS IMU utilizzato.

Tabella 2.2 – Principali caratteristiche del dispositivo GNSS/INS IMU utilizzato.

Input voltage	4.5-34V or 3V3	Latency	< 2 ms
Power consumption	950 mW	Standard full range gyro	450°/s
Temperature	-40 to 85 °C	Standard full range acc	200m/s ²
Sampling frequency	10 kHz/channel (60 ks/s)	Bandwidth gyro	415 Hz
Interfaces	RS232/RS485/RS422/UART/USB	Bandwidth acc	415 Hz
Clock drift	10 ppm (1 ppm with GNSS)	In-run bias stability gyro	10°/h

2.2.1.2 - Ricevitore

Ogni ricevitore del sistema SBL è assemblato in una custodia impermeabile ed è dotato di:

- un ricetrasmittitore acustico con trasduttore piezoelettrico;
- un GPS/GNSS Disciplined Oscillator per la sincronizzazione dell'orario;
- un cavo waveguide subacqueo per la connessione Wi-Fi con il tablet.

In figura 2.7 è rappresentato lo schema a blocchi del ricevitore SBL:

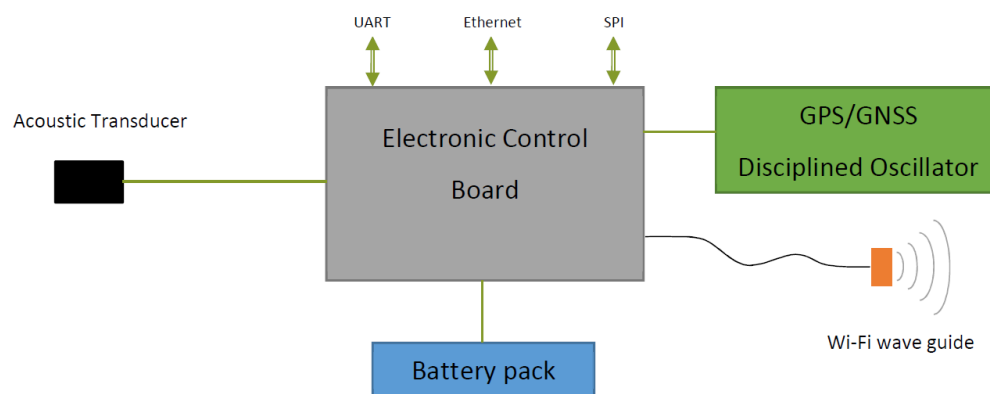


Figura 2.7 – Schema a blocchi del ricevitore.

Il trasduttore piezoelettrico consente lo scambio di messaggi acustici con l'unità di superficie. Il GPS/GNSS Disciplined Oscillator viene utilizzato in superficie prima di ogni immersione per sincronizzare l'orologio di sistema con il segnale orario del GNSS. Il segnale generato PPS (pulse

per second) dal GPSDO viene utilizzato dalla scheda elettronica di controllo per sincronizzare l'orologio di sistema con quello dell'unità di superficie. L'antenna GPS può essere fissa o rimovibile, quindi può essere rimossa prima dell'immersione.

La waveguide, invece, serve per interfacciare il ricevitore con un tablet/smartphone tramite la tecnologia Wi-Fi. La waveguide consente la trasmissione del segnale wireless in acqua utilizzando un cavo coassiale e un'antenna impermeabile che deve essere incollata sulla custodia impermeabile del tablet/smartphone. Questa soluzione consente di stabilire una connessione dati senza utilizzare un connettore impermeabile, evitando qualsiasi modifica alla custodia impermeabile.

Le schede elettroniche del ricevitore sono state progettate per essere assemblate su un supporto rotondo di plastica da inserire dentro un case cilindrico. In figura 2.8 è mostrata una sezione delle schede elettroniche nella custodia impermeabile.

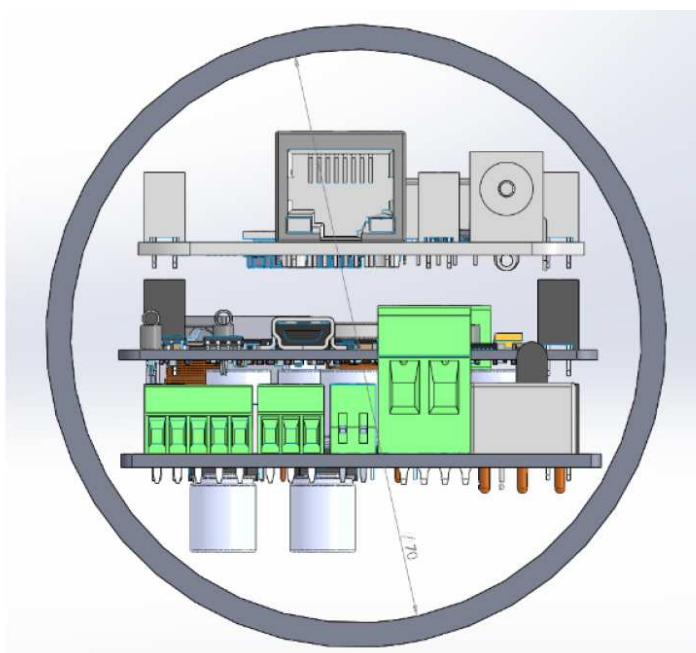


Figura 2.8 – Schede di controllo del ricevitore all'interno della custodia impermeabile.

2.2.1.3 - Principio di funzionamento

Sia l'unità di superficie che i ricevitori integrano la tecnologia Seamodem che consente, oltre alla capacità di determinare il posizionamento, di trasmettere dei dati tra i dispositivi. Le funzioni del modem consentono di inviare dati dal ricevitore all'unità di superficie. In particolare, qualsiasi ricevitore può essere configurato come "master" e può inviare la sua posizione geografica all'unità di superficie a intervalli regolari. Se l'unità di superficie è installata su un'unità mobile, può utilizzare questa informazione per seguire il ricevitore master. È possibile abilitare contemporaneamente più ricevitori come "master" ma, per evitare collisione, è necessario implementare una politica di scheduling per l'invio e la ricezione di messaggi riducendo la frequenza di aggiornamento della posizione. Inoltre, gli altri ricevitori potrebbero ascoltare il messaggio di posizione inviato dal master all'unità di superficie e, se necessario, inoltrare questa informazione ai tablet a cui sono collegati.

La figura 2.9 descrive la sequenza di comunicazione tra l'unità di superficie ed il ricevitore (o ricevitori). L'unità di superficie invia ciclicamente un messaggio da ciascun trasduttore, e tutti i ricevitori possono misurare la loro distanza dal trasduttore sorgente. Il ricevitore stima la sua posizione utilizzando le distanze tra il suo trasduttore e i quattro installati sull'unità di superficie.

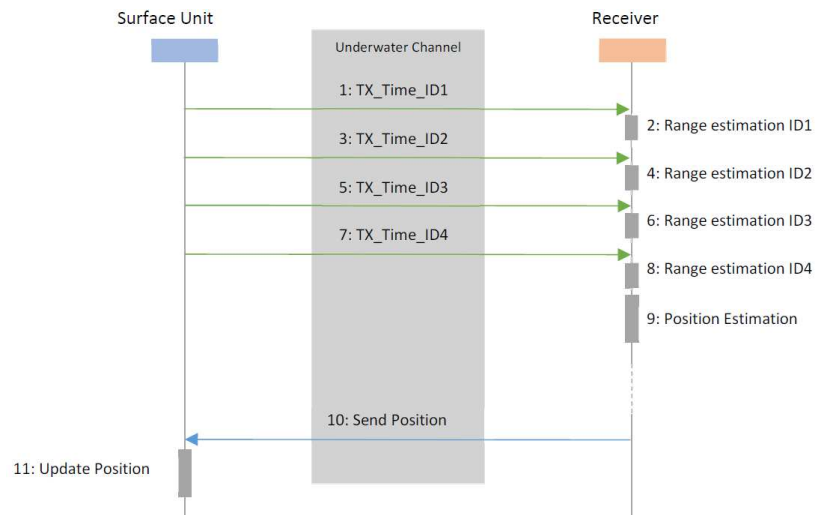


Figura 2.9 – Sequenza di comunicazione tra l'unità di superficie ed il ricevitore.

La sequenza di comunicazione prevede quindi le seguenti fasi:

1. l'unità di superficie invia il messaggio dal trasduttore 1;
2. il ricevitore stima la distanza dal trasduttore 1;
3. l'unità di superficie invia il messaggio dal trasduttore 2;
4. il ricevitore stima la distanza dal trasduttore 2;
5. l'unità di superficie invia il messaggio dal trasduttore 3;
6. il ricevitore stima la distanza dal trasduttore 3;
7. l'unità di superficie invia il messaggio dal trasduttore 4;
8. il ricevitore stima la distanza dal trasduttore 4;
9. il ricevitore trilatera le quattro distanze e stima la sua posizione;
10. il ricevitore master invia la posizione all'unità di superficie;
11. l'unità di superficie aggiorna la "master position".

La stima della posizione sul ricevitore viene effettuata utilizzando un filtro di Kalman che integra anche dati di profondità letti da un sensore di pressione aggiuntivo.

2.2.1.4 Interfacce di comunicazione

Protocollo di comunicazione tra l'unità di superficie SBL ed Host

L'unità di superficie SBL, come descritto in precedenza, fornisce un'interfaccia Ethernet cablata per connessione PC Host. Il PC Host può essere una scheda embedded installata su un veicolo o una docking station installata su una nave di supporto. L'interfaccia può essere facilmente convertita in Wi-Fi con a semplice adattatore Wi-Fi/Ethernet.

Il protocollo di comunicazione utilizza socket UDP e TCP per la trasmissione dei dati. I dati sono scambiati tra l'unità di superficie SBL verso il veicolo/PC Host e viceversa (figura 2.10).

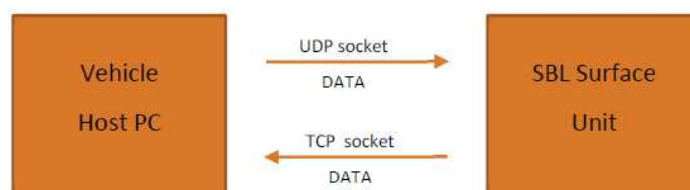


Figura 2.10 – Comunicazione tra veicolo/PC Host ed unità di superficie SBL.

L'unità di superficie SBL invia continuamente attraverso l'interfaccia ethernet integrata:

- la posizione assoluta (Latitudine e Longitudine) del ricevitore master;
- la profondità del ricevitore;
- l'ID di un messaggio predefinito ricevuto dal tablet master.

I dati inviati dall'unità di superficie al veicolo sono formattati secondo un pacchetto TCP definito nella tabella 2.3.

Tabella 2.3 – Pacchetto TCP inviato dall'unità di superficie al veicolo.

Field	Format	Unit	Description
Master ID	int8	-	The ID to identify the master tablet that data refer to
Timestamp	double	-	The packet timestamp
Latitude	double	°	Geographic coordinate of the master receiver
Longitude	double	°	Geographic coordinate of the master receiver
Depth	double	m	Depth of the receiver / tablet
Predefined message	int8	-	Default message ID

I dati inviati dal veicolo all'unità di superficie sono formattati secondo un pacchetto UDP definito nella tabella 2.4.

Tabella 2.4 – Pacchetto UDP inviato dal veicolo all'unità di superficie.

Field	Format	Unit	Description
Timestamp	double	-	The packet timestamp
Latitude	double	°	Geographic coordinate of vehicle
Longitude	double	°	Geographic coordinate of vehicle
Depth	double	m	Depth of the vehicle
Pitch angle	double	°	Vehicle pitch angle
Roll angle	double	°	Vehicle roll angle
Heading angle	double	°	Vehicle heading angle compared to the North

Il sistema può anche essere configurato per utilizzare pacchetti di dati adoperando lo standard NMEA-0183. Il comando CMD_REQ_SET_NMEA deve essere inviato all'unità di superficie per abilitare il protocollo di comunicazione NMEA. In questo caso, un messaggio modificato \$GPGGA con la posizione del ricevitore viene inviata dall'unità di superficie al veicolo tramite socket UDP. Il messaggio \$GPGGA è formattato secondo la seguente tabella 2.5.

Tabella 2.5 – Messaggio \$GPGGA secondo lo standard NMEA-0183.

Field	Description
0	Message ID [\$ GPGGA]
1	UTC time [hhmmss.ss]
2	Latitude [xxxx.xxxx]
3	Latitude direction [N = Nor d; S = South]
4	Longitude [xxxx.xxxx]
5	Longitude direction [E = East; W = West]
6	Quality of GPS position [0 = invalid; 1 = GPS fix; 2 = Diff. GPS fix]
7	Number of satellites in use
8	HDOP (Horizontal dilution of precision)
9	Altitude above/below mean sea level (MSL)
10	Units of Altitude [M = Meters]
11	Undulation - the relationship between the geoid and the WGS84 ellipsoid
12	Units of undulation [M = Meters]
13	Master ID (modified value respect standard NMEA-0813)
14	Message ID (modified value respect standard NMEA-0813)
15	Checksum [* xx]

Esempio di messaggio \$GPGGA:

\$ GPGGA, 172814.0,3723. 46587704, N, 12202.26957864, W, 2.6.1.2.18.893, M, -25.669, M, 2.0.0031 * 4F

In modalità NMEA, l'unità di superficie SBL è configurata per ricevere due diversi messaggi NMEA-0183 dall'interfaccia Ethernet: \$PRDID e \$GPGGA.

Il messaggio \$GPGGA ha lo stesso formato descritto nella sezione precedente. Al contrario, il messaggio \$PRDID è formattato come in tabella 2.6.

Tabella 2.6 – Messaggio \$PRDID secondo lo standard NMEA-0183.

Field	Description
0	Message ID [\$ PRDID]
1	Pitch angle [ppp.pp]
2	Roll angle [rrr.rr]
3	Heading angle compared to the North [hhh.hh]

Entrambi i messaggi possono essere utilizzati dall'unità di superficie quando viene installata su un veicolo dotato di piattaforma inerziale e sistema di posizionamento GPS. In questo caso, la piattaforma inerziale, integrata nell'unità di superficie, può essere rimossa, e la correzione

inerziale è calcolata utilizzando i dati ricevuti con il messaggio \$PRDID. Invece i dati ricevuti tramite il messaggio \$GPGGA vengono utilizzati per ottenere la posizione geografica assoluta del ricevitore/tablet subacqueo.

Protocollo di comunicazione tra ricevitore acustico e tablet

Il ricevitore è stato progettato per essere facilmente interfacciato con i dispositivi mobili più diffusi (tablet o smartphone) basati sia su iOS che su Android. Tuttavia, a causa del povero supporto di librerie cross-platform per la tecnologia Bluetooth, è stata utilizzata la comunicazione Wi-Fi standard per lo scambio di dati e comandi tra il tablet e ricevitore. In Figura 2.11 è riportato lo schema di connessione.

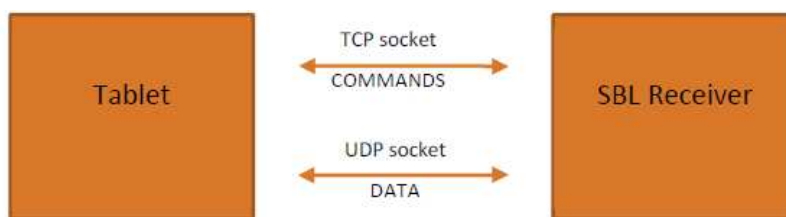


Figura 2.11 – Connessione TCP/UDP tra tablet e ricevitore.

Il protocollo di comunicazione utilizza un socket TCP per scambiare comandi tra ricevitore e tablet, ed un socket UDP per il trasferimento dei dati. La scelta di utilizzare il canale TCP per lo scambio di comandi garantisce l'arrivo con successo di ogni pacchetto alla destinazione, grazie alla trasmissione del protocollo intrinseco e le politiche di controllo. Attraverso il canale TCP il tablet è quindi in grado di:

- configurare i parametri principali del sistema;
- modificare i parametri utilizzati dall'algoritmo che stima il range;
- modificare i parametri del filtro di Kalman per la stima della posizione del ricevitore;
- avviare e interrompere il servizio di localizzazione acustica.

L'utilizzo del canale UDP, invece, aumenta la velocità di trasmissione dei dati di posizionamento verso il tablet. Attraverso il canale UDP il tablet è in grado di ricevere le seguenti informazioni:

- posizione relativa del subacqueo in tempo reale;
- temperatura dell'acqua;
- profondità del ricevitore;
- percentuale della batteria del ricevitore;
- livello di raggiungibilità dell'unità di superficie.

La scheda madre del sistema SBL gestisce l'invio e la ricezione di messaggi tramite un *network bridge interface* in grado di comunicare con il "Processor Core" che controlla il servizio di posizionamento acustico. La *network bridge interface* ha due thread di ascolto che usano due porte diverse a livello di applicazione per accettare nuove connessioni. Inoltre, per quanto riguarda il canale UDP, la network interface è in grado di supportare più connessioni, consentendo così a più dispositivi di ricevere contemporaneamente gli stessi dati di posizionamento. La multi-connessione sul canale UDP può essere utilizzata per eseguire il debug del sistema di posizionamento acustico durante il suo utilizzo. La Figura 2.12 mostra lo schema a blocchi dell'interfaccia di rete.

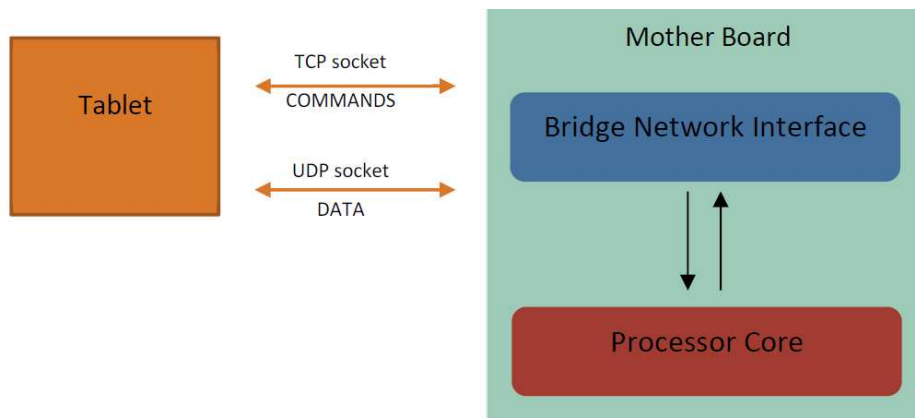


Figura 2.12 – Schema a blocchi della “network interface”.

Di seguito, in tabella 2.7, sono elencati e descritti i messaggi inviati dal tablet tramite il canale TCP per il controllo del servizio di localizzazione acustica.

Tabella 2.7 – Messaggi inviati dal tablet tramite canale TCP per il controllo del servizio di localizzazione acustica.

MESSAGE	DESCRIPTION
CMD_REQ_SET_INIT_P	Set the matrix P of the Kalman filter
CMD_REQ_SET_INIT_Q	Set the Q matrix of the Kalman filter
CMD_REQ_SET_INIT_R	Set the matrix R of the Kalman filter
CMD_REQ_SET_INIT_X	Set the initial position of the tablet within the site of interest
CMD_REQ_SET_INIT_U	Set the initial input values to the Kalman filter
CMD_REQ_SET_INIT_Z	Set the starting distance from the Beacons
CMD_REQ_SET_DT	Set the update time of the Kalman filter
CMD_REQ_SET_IDMODEM	Set the unique ID of the modem associated with the tablet
CMD_REQ_SET_GAINTX	Set the transmission power of the acoustic modem associated with the tablet
CMD_REQ_SET_FSKMOD	Set the transmission mode (number of FSK modulation tones) of the modem associated with the tablet
CMD_REQ_SET_VITERBI	Activate or deactivate the modem error correction code according to the Viterbi scheme
CMD_REQ_SET_GUARDPERIOD	Select the guard periods between the symbols during the acoustic data transmission
CMD_REQ_SET_ACKTIME	Set the retransmission time if a beacon is not reachable

Di seguito, in tabella 2.8, sono elencati e descritti i messaggi inviati dal ricevitore al tablet tramite il canale TCP.

Tabella 2.8 – Messaggi inviati dal ricevitore al tablet tramite il canale TCP.

MESSAGE	DESCRIPTION
CMD_INFO	Send GPSDO status [Fixing, Fixed]

I messaggi ricevuti sul socket UDP contengono dati di posizionamento e altre informazioni come profondità, temperatura dell'acqua e stato della batteria del ricevitore. Il ricevitore invia al tablet la sua posizione NED rispetto ad un punto di riferimento iniziale. Questo punto di riferimento è conosciuto sia dal tablet sia dall'unità di superficie, e viene utilizzato per convertire coordinate NED in posizioni geografiche assolute e viceversa.

Il messaggio (tabella 2.9) è così strutturato: **N#E#D#B1#B2#B3#B4#TEMP#BATTERY!**

Tabella 2.9 – Struttura del messaggio che il ricevitore invia al tablet.

GIVEN	DESCRIPTION
N, E, D	Relative NED position of receiver respect the initial reference point.
B1, B2, B3, B4	Distances between each transducer on the surface unit with respect to the receiver expressed in meters. (-1 in case of an unreachable transducer)
TEMP	Water temperature
BATTERY	Receiver battery level expressed as a percentage

2.2.2 – Descrizione del sistema ASV H2Omni-X

Il veicolo di superficie autonomo H2Omni-X (chiamato PlaDyPos [2.04] nella sua precedente versione) mostrato in figura 2.13, è un veicolo autonomo di superficie sovra-azionato autonomo capace di compiere movimenti omnidirezionali. Il veicolo è equipaggiato con quattro thrusters nella configurazione a "X" che consentono il movimento libero sul piano orizzontale per qualsiasi orientazione.



Figura 2.13 – Veicolo ASV H2Omni-X sviluppato dalla H2O Robotics di Zagabria.

Esso è dotato di un computer NGC (navigation, guidance and control). Dipendente dai requisiti di elaborazione, è riservato uno spazio extra per un secondo computer dedicato solitamente all'elaborazione dei dati dei sensori che vengono montati a bordo. Le dimensioni fisiche del veicolo sono riportate in figura 2.14. Esso pesa fino a 25 kg a seconda del carico utile e del tipo di batteria.

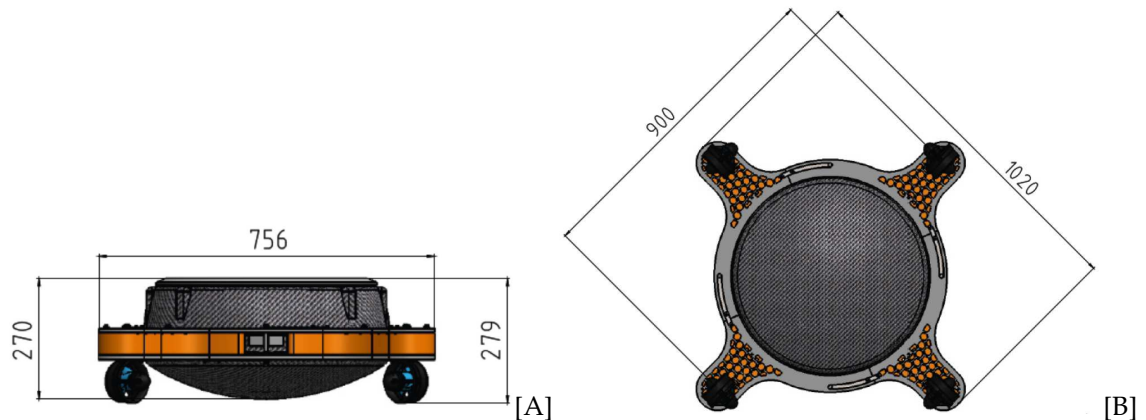


Figura 2.14 – Dimensioni fisiche del veicolo ASV H2Omni-X in una vista laterale (a) ed in una vista dall'alto (b).

Il dispositivo presenta le seguenti specifiche e caratteristiche hardware:

- autonomia da 3 a 6 ore a seconda della configurazione del pacco batteria;
- 1 m/s di velocità in qualsiasi direzione;
- scafo in fibra di carbonio contenente elettronica e batterie;
- inserti a galleggiamento variabili per varie opzioni di carico;
- 4 propulsori BlueRobotics T100 per ridondanza di tolleranza ai guasti;
- ampio portello marino per un facile accesso all'elettronica;
- sistema GPS ad alta precisione Ublox M8P;
- unità di misura inerziale MPU9250 MEMS;
- comunicazione wireless Ubiquiti 2,4 GHz;
- batteria 12V VRLA;
- diverse capacità di carico utile:
 - o USBL;
 - o DVL;
 - o modem acustico;
 - o multibeam imaging sonar;
 - o profilatore batimetrico;
 - o sonde biologiche e chimiche;
 - o altro payload personalizzato.

2.3 – Integrazione del sistema SBL sull'ASV H2Omni-X

L'idea principale alla base dell'integrazione del sistema SBL sull'ASV è aumentare la mobilità e la flessibilità. Quando trasportato dal veicolo, il sistema SBL può muoversi con il gruppo subacqueo e posizionare l'array di trasduttori vicino al gruppo per una migliore precisione di localizzazione. Poiché il sistema SBL è un carico utile personalizzabile, è stato creato un nuovo supporto per i trasduttori acustici. Inoltre, è stata resa necessaria l'integrazione elettronica e software per le schede di controllo del sistema SBL (figura 2.15).

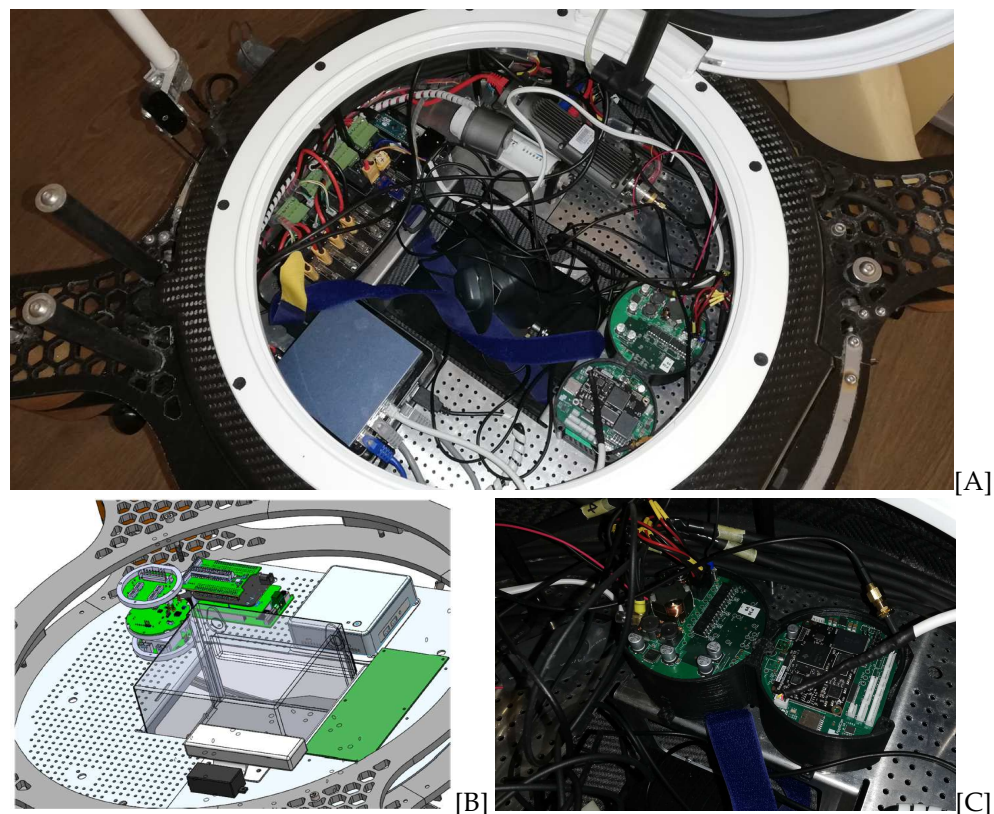


Figura 2.15 – Installazione dell'elettronica dell'unità SBL di superficie all'interno del veicolo; (a) elettronica generale all'interno dello scafo, (b) modello CAD, (c) elettronica del sistema acustico.

Tutta l'elettronica SBL è stata montata direttamente nel veicolo e l'alimentazione è fornita dalla batteria principale del veicolo. Le schede sono separate in due gruppi per adattarsi all'interno dello scafo del veicolo. Il montaggio è stato eseguito sul secondo rack dedicato al carico utile e l'orientamento finale e la distanza sono stati adattati nel modo più ottimale per questioni legate al cablaggio.

Per quanto riguarda la progettazione dei supporti relativi ai trasduttori acustici, essi sono stati disposti con una configurazione a croce; si è pensato di progettare un sistema con un meccanismo scorrevole per consentire la salvaguardia dei trasduttori in una posizione sicura quando si va a recuperare il veicolo. I trasduttori vengono spostati dietro i propulsori quando sono retratti e sotto quando sono completamente abbassati (figura 2.16). Questo montaggio offre una distanza di 0,53/0,74 m tra i trasduttori (figura 2.17). Questa soluzione fornisce un collegamento più stabile con la piattaforma, assicurando meno vibrazioni involontarie sul trasduttore causate dalle onde o durante il movimento del veicolo. Alcune modifiche, con minore sicurezza per i trasduttori, potrebbero riguardare in futuro una maggiorazione della linea di base (baseline) andando a sconfinare oltre le attuali dimensioni del veicolo di superficie.

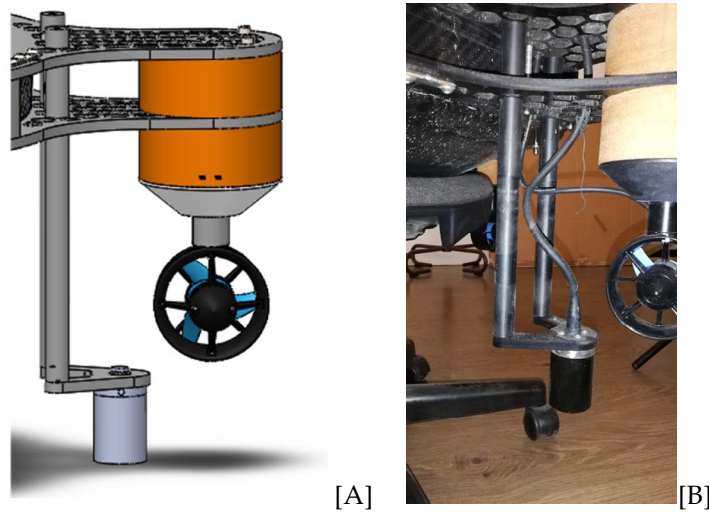


Figura 2.16 – Soluzione adottata per il posizionamento del trasduttore in configurazione estesa; (a) modello CAD, (b) prototipo reale.

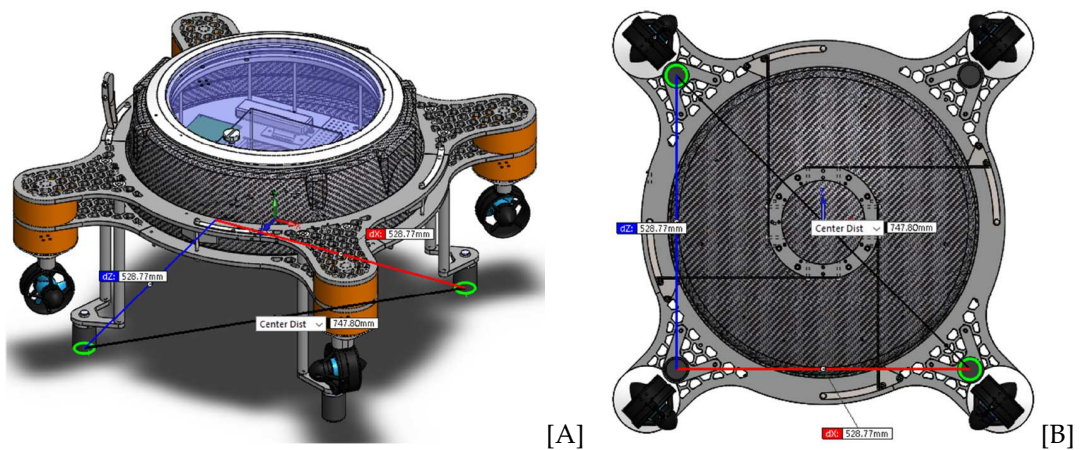


Figura 2.17 – Montaggio dei trasduttori sul veicolo con meccanismo scorrevole in vista generica (a) e vista dal basso (b).

Il montaggio dei trasduttori è avvenuto in ordine orario come mostrato in figura 2.18, in relazione alle coordinate NED del veicolo.

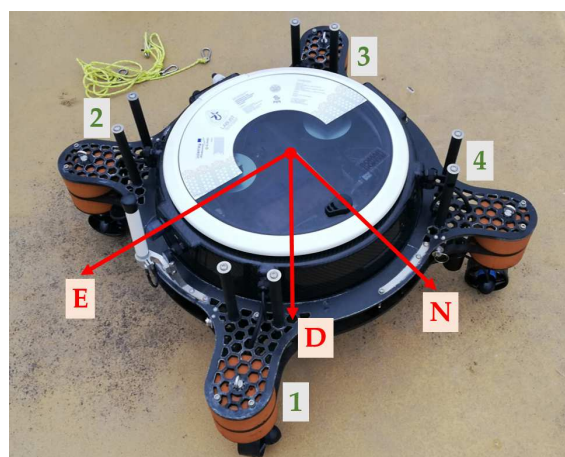


Figura 2.18 – Coordinate NED (rosso) del veicolo ed ordine in senso orario dei trasduttori (verde).

Osservando la figura 2.15(a), è possibile notare che il veicolo è stato dotato di uno switch Ethernet per garantire le connessioni tra l'unità SBL e la CPU del veicolo con il Gateway Wi-Fi. Lo schema di collegamento elettrico e dei dati è riportato in figura 2.19.

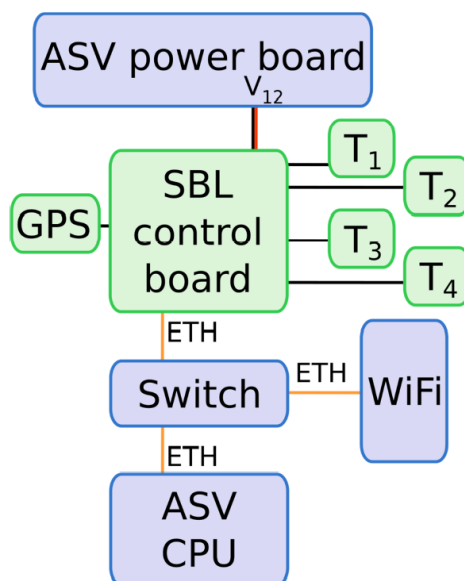


Figura 2.19 – Schema di collegamento elettrico e dei dati per l'integrazione dell'unità SBL sul veicolo.

Il sistema di controllo del SBL è stato collegato alla scheda elettrica principale del veicolo H2Omni-X con un connettore XT60 per la fornitura della potenza, in particolare ad una uscita a 12V che è possibile attivare e chiudere mediante il controllo di un relè. In questo modo l'unità di controllo SBL può essere spenta nel momento in cui il veicolo viene tolto dall'acqua. Le connessioni dei trasduttori sono state realizzate attraverso il passaggio dei cavi dall'esterno mediante dei connettori Bulgin.

L'unità di superficie SBL è collegata alla rete del veicolo con la sua interfaccia Ethernet. Utilizzando questa connessione di rete, l'unità di superficie SBL invia al veicolo, con protocollo UDP, i seguenti dati:

- la posizione relativa del ricevitore master rispetto al veicolo utilizzando coordinate locali NED;
- la profondità del ricevitore;
- un array di caratteri con messaggi predefiniti e configurabili utilizzati per l'invio di informazioni aggiuntive.

Il veicolo, invece, invia all'unità di superficie SBL due diversi messaggi NMEA-0183 con l'interfaccia Ethernet: \$PRDID e \$GPGGA. Il primo \$PRDID fornisce informazioni sull'orientamento del veicolo, mentre il secondo la sua posizione GPS. Il veicolo è dotato di un Ublox RTK GPS ed è stata sostituita l'unità di misura inerziale con una della serie XSENS; il computer NGC legge le informazioni utili da questi dispositivi e li inoltra all'unità di superficie SBL con i messaggi NMEA-0183. I messaggi \$PRDID e \$GPGGA sono utilizzati per inviare gli stessi dati dal sistema SBL integrato al computer Host. In questo modo è stato possibile testare tutte le combinazioni di integrazione del sensore, come ad esempio "Host localization" per SBL, "SBL localization" per Host, e "shared hybrid Host SBL localization". I principali scambi di dati per queste combinazioni sono mostrati nella figura 2.20, la quale definisce inoltre anche la porta UDP per l'utilizzo in termini di "message passing".

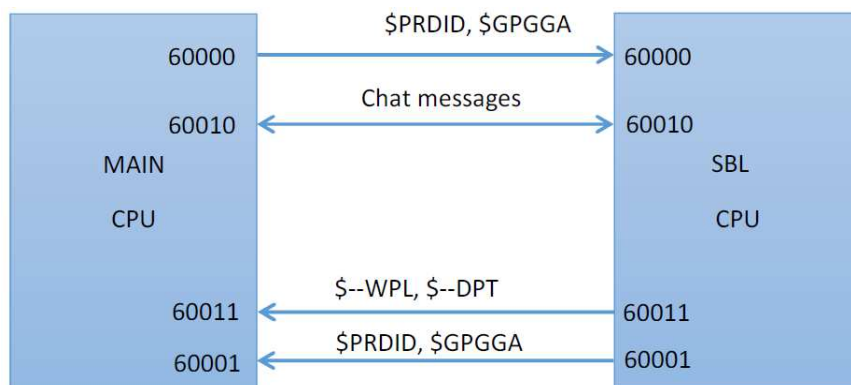


Figura 2.20 – Interfacce per lo scambio di dati.

Oltre ai messaggi per lo scambio dati IMU e GPS, può essere trasferita la posizione del subacqueo con un messaggio waypoint ****WPL** ed il relativo messaggio ****DPT** per specificare la profondità dell'acqua al trasduttore master. Questi messaggi sono definiti come riportato in tabella 2.10.

Tabella 2.10 – Definizione dei messaggi WPL e DPT scambiati dal sistema SBL verso la CPU del veicolo.

Field	Description
0	Message ID [\$--DPT]
1	Depth [xx.x]
2	Transducer offset [0.0]
Field	Description
0	Message ID [\$--WPL]
1	North in meters [xxx.x]
2	Static text "N"
3	East in meters [xxx.x]
4	Static text "E"
5	Transducer ID [name or number]

Capitolo 3

Analisi di propagazione delle fonti di errore in un sistema SBL.

In questo capitolo è descritto lo sviluppo di un simulatore in ambiente Matlab/Simulink per l'emulazione del dato di posizionamento di un target subacqueo, affrontando il problema di come diverse sorgenti di errore (ad esempio GPS, IMU, ranges, ecc.) possano andare a compromettere le performance. È stato creato un modello di simulazione capace di fornire informazioni sull'accuratezza del sistema di localizzazione subacqueo. È fornita la descrizione di un sistema acustico subacqueo SBL preso a riferimento per lo sviluppo del simulatore, utile per identificare le varie fonti di errore. È stata effettuata l'analisi di propagazione degli errori in un contesto in cui il ricevitore acustico sia sempre statico. È stata poi effettuata l'analisi di propagazione degli errori in un contesto in cui il ricevitore acustico sia dinamico, aggiungendo il caso in cui la stima della posizione venga effettuata insieme ad una unità inerziale montata nel ricevitore. Partendo da alcuni dati sperimentali del sistema SBL, sono state implementate delle strategie di sensor-fusion che vedono la combinazione del sistema acustico con una camera subacquea. In sintesi, quindi, sono state avviate delle indagini simulate e sperimentali per migliorare le performance di un sistema di localizzazione acustica, anche in combinazione con sensori inerziali ed ottici (tecniche di visual-inertial odometry). Il capitolo contiene le informazioni riguardanti le simulazioni condotte, illustrando come esse sono state svolte e quali scenari vanno a simulare. Le simulazioni sono state validate tramite risultati pervenuti in letteratura in configurazioni coerenti con quella del caso di studio.

3.1 - Contesto

In letteratura sono presenti diversi studi relativi all'analisi delle sorgenti d'errore per i sistemi di localizzazione acustica subacquea. In particolare, in [3.01] è stata trattata un'analisi che riguarda l'impatto di diverse fonti di errore in un sistema SBL sui risultati di localizzazione stimati da un filtro di Kalman Esteso (EKF); questo lavoro ha previsto lo sviluppo di un ambiente virtuale che ha consentito di effettuare un'analisi statistica dei dati simulati. In [3.02] gli stessi autori hanno approfondito le analisi condotte validando i risultati ottenuti in simulazione con dati registrati da test sperimentali. In entrambi gli articoli scientifici si è dimostrato che il GNSS (GPS), IMU, lunghezza della Baseline e Ranges sono i fattori che impattano di più sull'accuratezza della localizzazione.

3.2 Sviluppo del modello

In questo paragrafo è presentato lo sviluppo di un simulatore per sistema SBL in ambiente Matlab/Simulink. L'obiettivo dell'attività è stato quello di costruire un modello in grado di simulare il funzionamento del sistema acustico di localizzazione, composto essenzialmente da due parti:

- Trasmettitore (fig. 3.1), ovvero la base acustica posta in superficie, e composta da 4 trasduttori acustici (configurazione SBL);
- Ricevitore, ovvero il ricevitore acustico trasportato dal sub e composto da un singolo trasduttore acustico, chiamato a volte Seamodem.

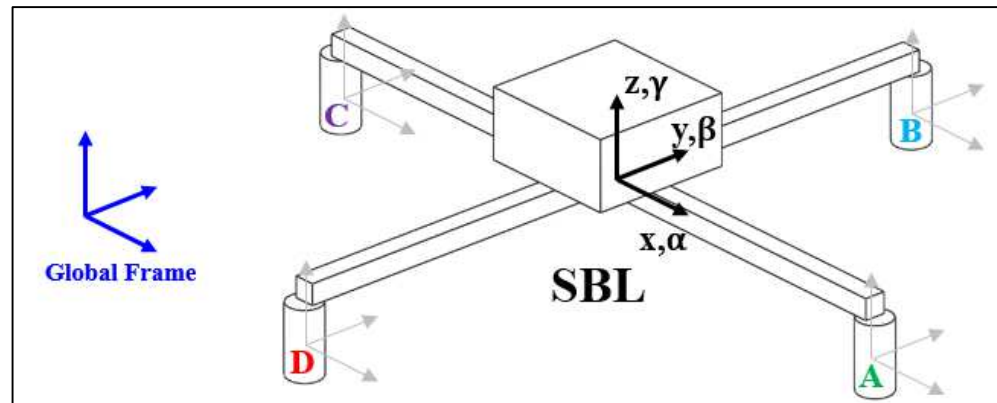


Figura 3.1 – Configurazione del trasmettitore acustico (sistema SBL).

In questo modo, è stato possibile valutare come le diverse sorgenti di errore vadano ad incidere sul risultato finale di localizzazione. È stato sviluppato un simulatore (modello multi-body) in Matlab/Simulink che permette di simulare la logica di funzionamento dell'hardware, in modo da poter individuare margini di miglioramento e valutare le migliori politiche di gestione dei 4 trasduttori che compongono l'unità del trasmettitore SBL. Sono state analizzate diverse problematiche, come ad esempio la compensazione software delle oscillazioni del sistema SBL e la possibile perdita di pacchetti di informazioni durante la trasmissione dei dati acustici (*ranging*). Ci si è posti come obiettivo quello di caratterizzare e simulare le possibili fonti di errore, e capire in che misura esse si propagano sullo Stato del sistema del ricevitore acustico, quindi sulla posizione e sulla velocità del ricevitore subacqueo.

L'unità SBL è equipaggiata con un sistema di navigazione inerziale che comprende GPS ed IMU e che fornisce posizione, velocità, accelerazione e orientamento in modo accurato e affidabile. Si è scelto di prendere come riferimento il sistema "Spatial" di "Advanced Navigation" [3.03], che è un sistema di navigazione inerziale in miniatura. Prevede GPS robusto e AHRS (Attitude Heading Reference System) che fornisce posizione, velocità, accelerazione e orientamento precisi nelle condizioni più difficili. Combina accelerometri calibrati a temperatura, giroscopi, magnetometri e un sensore di pressione con un ricevitore GNSS avanzato. Questi sono accoppiati in un sofisticato algoritmo di fusione.

3.2.1 – Principio di funzionamento del sistema SBL simulato

Il principio di funzionamento del sistema SBL preso a riferimento per la costruzione del modello, prevede le seguenti fasi:

- L'unità SBL è geolocalizzata in termini di X ed Y grazie al GPS rispetto al Global Frame;
- La posizione dei 4 sistemi di riferimento A,B,C,D è determinata per trasformazione rigida di sistemi di riferimento;
- Il sistema AHRS ci permette di conoscere le rotazioni roll, pitch e yaw del sistema di riferimento solidale alla croce;
- L'aggiornamento sugli spostamenti e sulle rotazioni dei sistemi di riferimento A,B,C,D è permesso grazie all'utilizzo di matrici di roto-traslazione.

Le matrici di traslazione, rotazione e roto-traslazione sono note in letteratura, pertanto per semplificare la trattazione si è scelto di non riportarle in questo testo. Si rimanda al riferimento [1.09] per la loro consultazione.

Presupponendo la sincronizzazione degli orologi dell'unità SBL e del ricevitore, i trasduttori A, B, C e D mandano a rotazione un messaggio acustico contenente il tempo di invio e l'informazione sulle coordinate X, Y e Z del sistema di riferimento solidale ad essi.

Il ricevitore riceverà tale pacchetto dopo un certo istante di tempo. È quindi possibile stimare il range (distanza) R. V è la velocità del suono nel mezzo e dipende da salinità dell'acqua, temperatura e profondità. Inoltre, $t^{**}-t^*$ rappresenta il tempo di volo. In figura 3.02 è riportato uno schema riassuntivo.

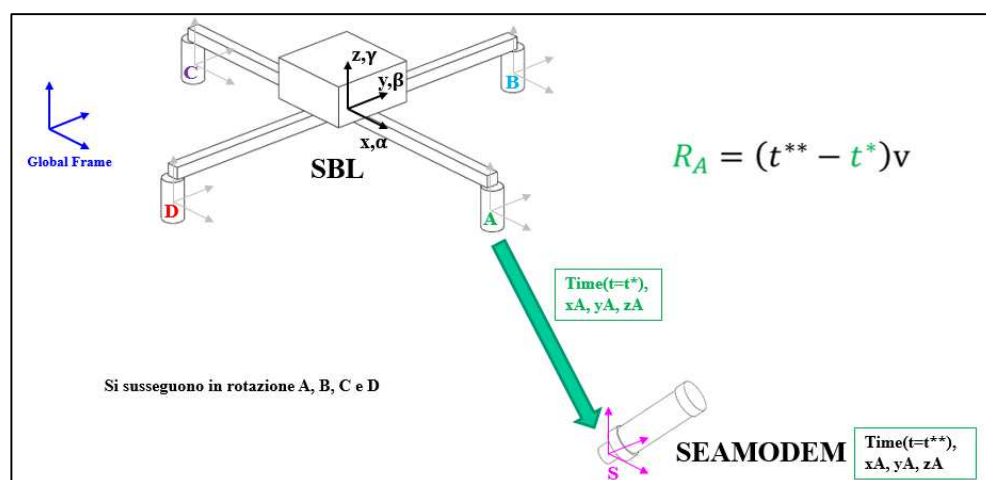


Figura 3.2 – Schema riassuntivo per il calcolo della distanza tra SBL e ricevitore acustico.

Le informazioni collezionate al termine della rotazione dei trasduttori sono i 4 range e 12 coordinate. Dal sistema di equazioni riportato in fig. 3.3 è possibile calcolare le coordinate X Y e Z del ricevitore, quindi esso è in grado di conoscere la sua posizione nel sistema di riferimento globale.

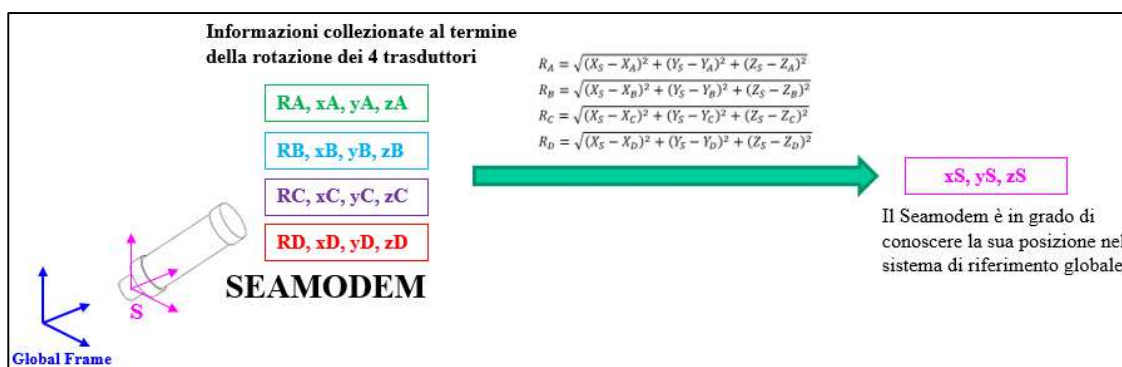


Figura 3.3 – Schema riassuntivo per il calcolo della posizione del ricevitore.

3.2.2 – Sviluppo del modello Matlab/Simulink ed analisi delle fonti di errore

È stato quindi programmato in Matlab/Simulink un simulatore che è stato nella maniera seguente. E' possibile dividere in 4 parti lo schema generale, ben distinguibili anche graficamente nella figura 3.4, senza addentrarci troppo nello specifico nei vari sotto-blocchi.

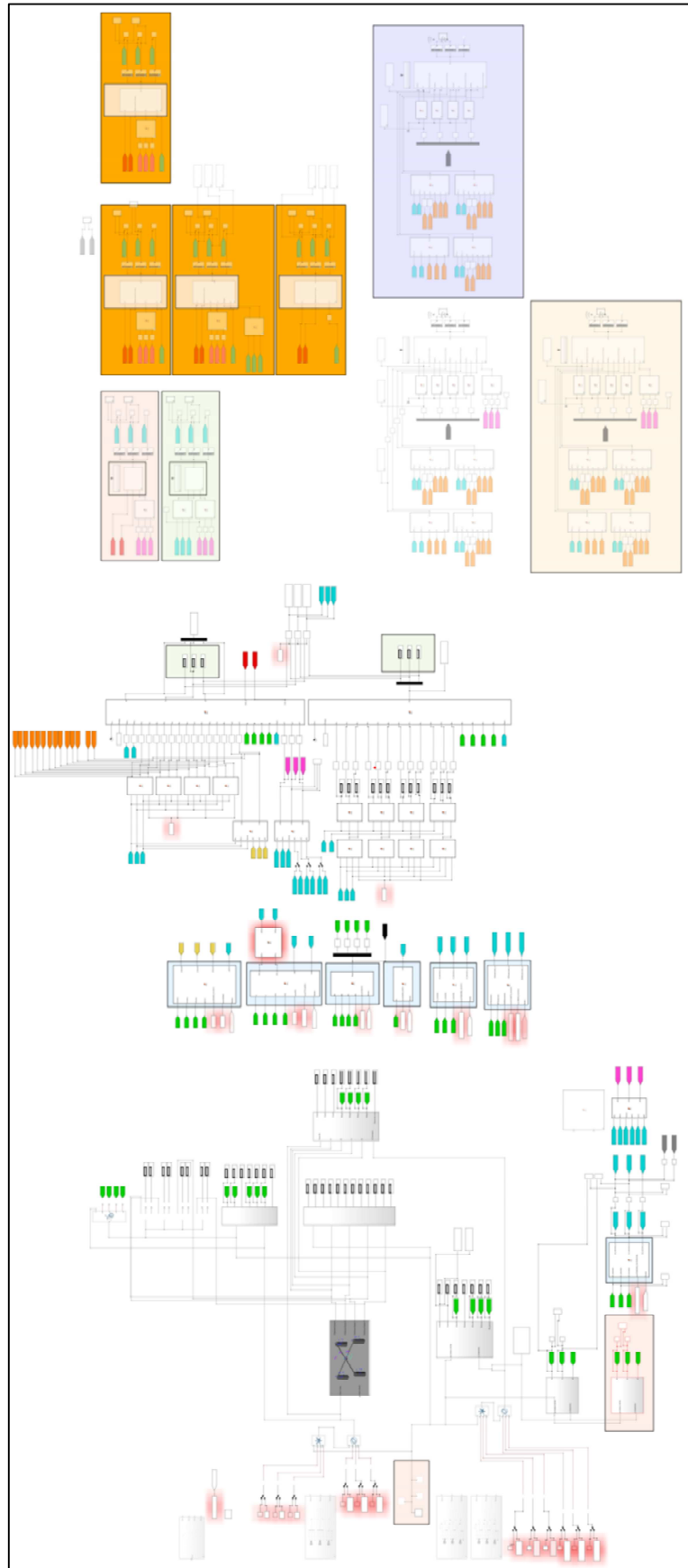


Figura 3.4 – Schema generale del modello sviluppato in Simulink.

Nei 4 macro-blocchi, procedendo dal basso verso l'alto troviamo quindi:

- parte relativa all'implementazione vera e propria del modello multi-body che ha previsto la costruzione del sistema SBL in grado di muoversi ed oscillare su un piano;
- parte relativa alla modellazione delle fonti di errore;
- parte relativa a strategie di gestione del GPS;
- parte relativa all'implementazione di diversi modelli del filtro di Kalman Estesio.

Il modello sviluppato ha previsto l'implementazione di diverse parti, mediante programmazione a blocchi e scripting, riassumili nel seguente elenco:

- modello multi-body (6 DOF per il trasmettitore e 6 DOF per il ricevitore);
- posizione statica, hovering, implementazione traiettorie personalizzabili per il trasmettitore;
- equipaggiamento con sensoristica (GPS, IMU, sensore di profondità);
- misura dei ranges;
- implementazione degli algoritmi triangolazione (6 strategie);
- 12 script per l'implementazione delle matrici di rotazione per l'aggiornamento dei sistemi di riferimento delle diverse parti;
- 2 script per l'implementazione di due strategie per la gestione del GPS;
- presenza di shift-register per l'accumulo e la memorizzazione dei dati per l'integrazione delle velocità e delle accelerazioni;
- trasformazione vettori dal sistema di riferimento locale al sistema di riferimento globale;
- implementazione degli Extended Kalman Filter (9 varianti).

Sugli errori è stata fatta l'ipotesi di distribuzione uniforme. I valori sono stati presi dai datasheet di una possibile strumentazione da utilizzare in fase di test, mentre quelli acustici (range) da risultati sperimentali di test precedentemente condotti con una strumentazione disponibile in laboratorio (tali valori prevedevano una deviazione standard pari a 0.3 metri). In figura 3.5 è riportato un elenco dei valori massimi e minimi implementati nel simulatore.

<p>Fonti di errore della strumentazione:</p> <p>SBL Errore GPS: ± 0.5 [m] X, Y Errore Pitch and Roll: ± 0.1 [deg] Alfa, Beta Errore Heading: ± 0.2 [deg] Gamma Errore Velocità: ± 0.05 [m/s] vX,vY</p> <p>SEAMODEM: Errore Sensore di Profondità: ± 0.2 [m] Z Errore Velocità: ± 1 [m/s] vX,vY,vZ Errore Pitch and Roll: ± 0.5 [deg] Errore Gamma: ± 1 [deg] Errore Accelerazione ± 0.02 [m/s²]</p> <p>RANGING ACUSTICO Errore ranging: ± 0.5 [m]</p>	<p>Tempi di campionamento nel simulatore: Simple Time Simulink: 0.001 [s] Simple Time AHRS velocity: 0.003 [s] Simple Time AHRS: 0.3 [s] Simple Time GPS: 1.2 [s] Simple Time Seamodem: 1.2 [s]</p> <p>Numero di campioni integrati: N= 401 ogni 1.2 secondi (frequenza= 334Hz)</p>
<p>Gli errori si vanno a propagare nel flusso di calcolo</p>	

Figura 3.5 – Valori massimi e minimi degli errori implementati nel simulatore.

Il simulatore prevede un flusso di segnali acquisiti con diversi tempi di campionamento a seconda che si parli della simulazione Simulink, del GPS, del ricevitore e del sistema AHRS, sia nelle matrici di rototraslazione sia nel caso in cui si venga effettuato un'integrazione di percorso, come sarà descritto più avanti. Nel simulatore è inoltre simulato lo sfasamento di arrivo dei ranges in 4 diversi istanti temporali. Gli errori si vanno a propagare nel flusso di calcolo, quindi è stato interessante capire come tutto ciò abbia ripercussioni sul calcolo della posizione del ricevitore.

3.2.3 - Considerazioni sull'errore introdotto dal solver di calcolo

Accade che, a causa della troppa variabilità dei ranges acustici, il solver non riesca a risolvere il sistema di equazioni della triangolazione, quindi approssimi la soluzione anche senza giungere a convergenza. Nello schema riportato (figura 3.6), per la coppia di trasduttori A e C ad esempio, posti sullo stesso asse, non si verifica la chiusura del triangolo se il range verde compreso di errore è maggiore della somma della baseline e del range viola compreso di errore. Questo errore non è trascurabile, ed usare la triangolazione in queste circostanze può non essere conveniente.

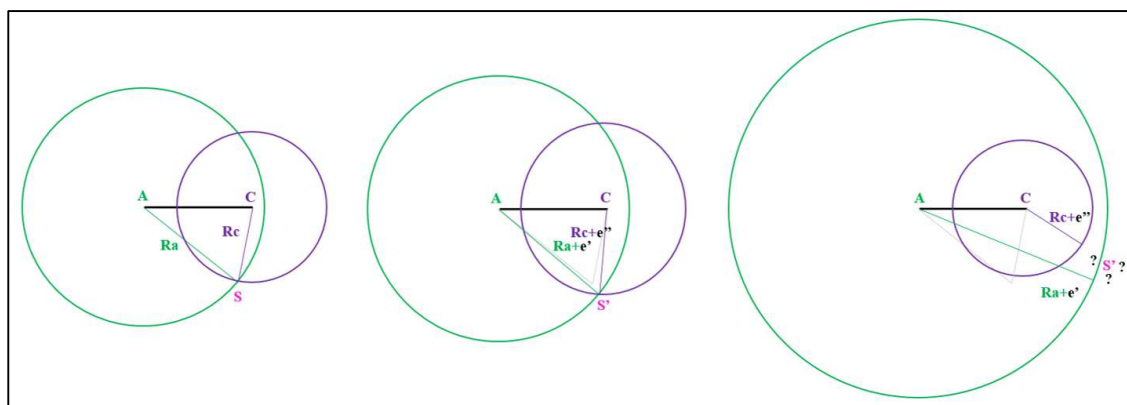


Figura 3.6 – Errore nella convergenza della triangolazione.

Una possibile fonte di errore è quindi rappresentata dall'approssimazione dell'algoritmo di calcolo. La triangolazione introduce un errore non trascurabile se i ranges acustici hanno troppa variabilità.

3.3 – Analisi di propagazione degli errori con ricevitore statico

In questa sezione è analizzato il caso in cui il trasmettitore sia statico oppure dinamico, mentre il ricevitore sia sempre statico.

Dal caso riportato in figura 3.7, che presuppone una base acustica (trasmettitore) dinamica, si evince come partendo da una configurazione iniziale, per praticità all'istante 0, i sistemi di riferimento B C e D subiscono dei cambiamenti perché l'acquisizione della loro posizione è sequenziale (non contemporanea); la struttura tende infatti a muoversi nell'arco di un breve intervallo temporale se consideriamo una situazione reale. È possibile determinare la posizione aggiornata dei sistemi di riferimento B, C e D mediante matrici di roto-traslazione. Si ipotizza che ogni 0.3 secondi avvenga l'invio di un segnale da parte di ciascun trasduttore, che verrà ricevuto dal ricevitore nei successivi 0.3 secondi.

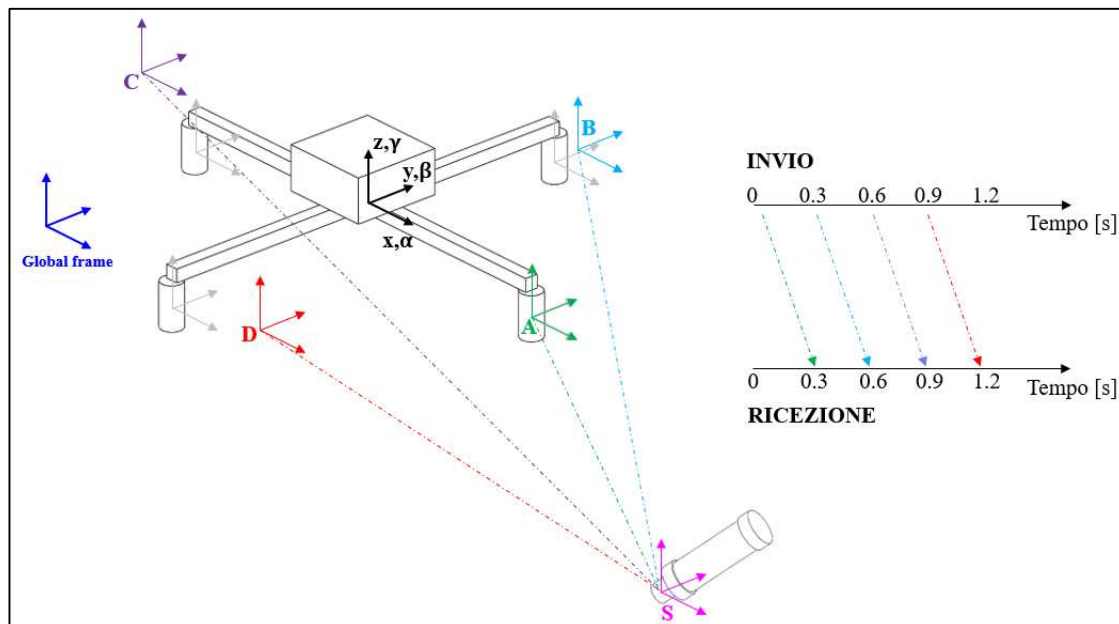


Figura 3.7 – Analisi di propagazione degli errori nel caso in cui il trasmettitore sia dinamico ed il ricevitore sia dinamico.

3.3.1 - Gestione del GNSS

È stato valutato l'impatto dell'errore del GNSS (Global Navigation Satellite System) montato sul sistema SBL, quindi sul trasmettitore, sia nel caso in cui quest'ultimo rimanga nella condizione di *hovering*, sia quando esso è in movimento.

3.3.1.1 - Trasmettitore statico

Si analizza prima il caso in cui il sistema SBL sia perfettamente fermo, ed ipotizziamo di considerare solo ed esclusivamente l'errore introdotto dal GNSS. Analizziamo due situazioni. In riferimento alla figura 3.8, nel primo caso ogni 0.3 secondi è campionato un valore del GNSS (riquadro centrale) che serve ad aggiornare di volta in volta le coordinate X e Y dei sistemi di riferimento A B C D dei trasduttori mediante una semplice traslazione lungo la baseline. Considerando che come ipotesi abbiamo assunto che la baseline sia statica, ed analizzando il secondo caso, ci rendiamo conto che basterebbe prendere solo un valore del GNSS all'istante 0 ed utilizzarlo per tutti gli altri 1.2 secondi totali che servono per la rotazione dei trasduttori. Nel primo caso (strategia S1), abbiamo una deformazione della baseline, ma i ranges calcolati dalla comunicazione acustica prendono come riferimento la posizione effettiva dei trasduttori, posti in croce perfetta. Per S1 si ha un tempo di aggiornamento del dato del GNSS ogni 0.3 [s]. Nel secondo caso (strategia S2), invece, si mantiene la forma della croce, ed abbiamo un offset lungo X e lungo Y. Per S2 si ha un tempo di aggiornamento del dato del GNSS ogni 1.2 [s].

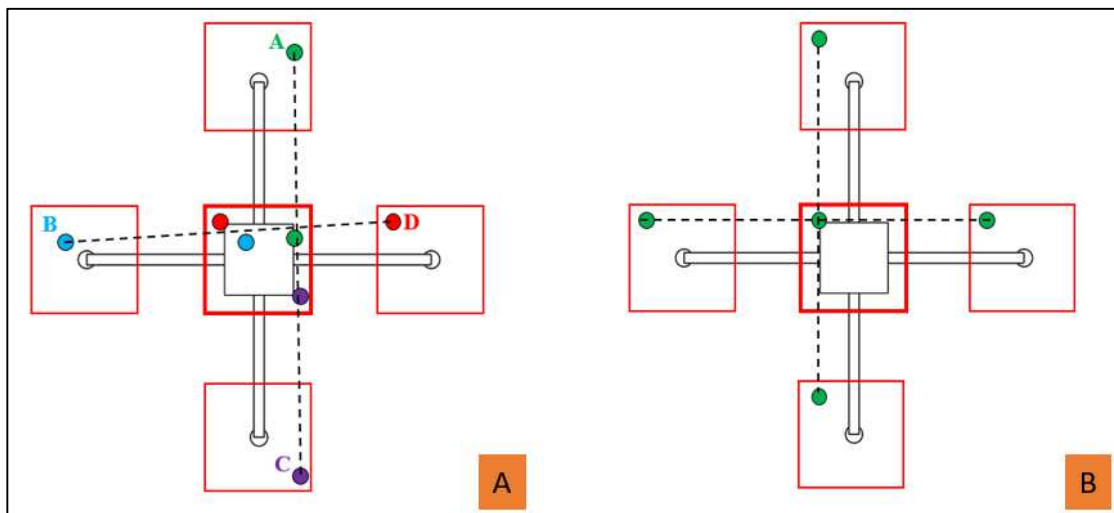


Figura 3.8 – Gestione del GNSS per trasmettitore statico: figura A in cui si verifica l’aggiornamento continuo per tutti i 4 trasduttori nell’arco di una sola rotazione (strategia S1), figura B in cui si utilizza un solo valore per rotazione dato dalla prima acquisizione (strategia S2).

In entrambe le configurazioni, si ribadisce che la baseline del sistema SBL sia statica. Si ipotizzi per il momento di considerare solo il GNSS come sorgente di errore. Analizzando diverse posizioni del ricevitore (12 punti rossi in figura 3.9) nello scenario acquatico, ed ipotizzando quindi che il trasmettitore sia statico nel punto (0,0), simulando un centinaio di punti per ogni posizione abbiamo una distribuzione incontrollata dell’errore che aumenta all’aumentare della distanza X,Y e si propaga più lentamente lungo Z.

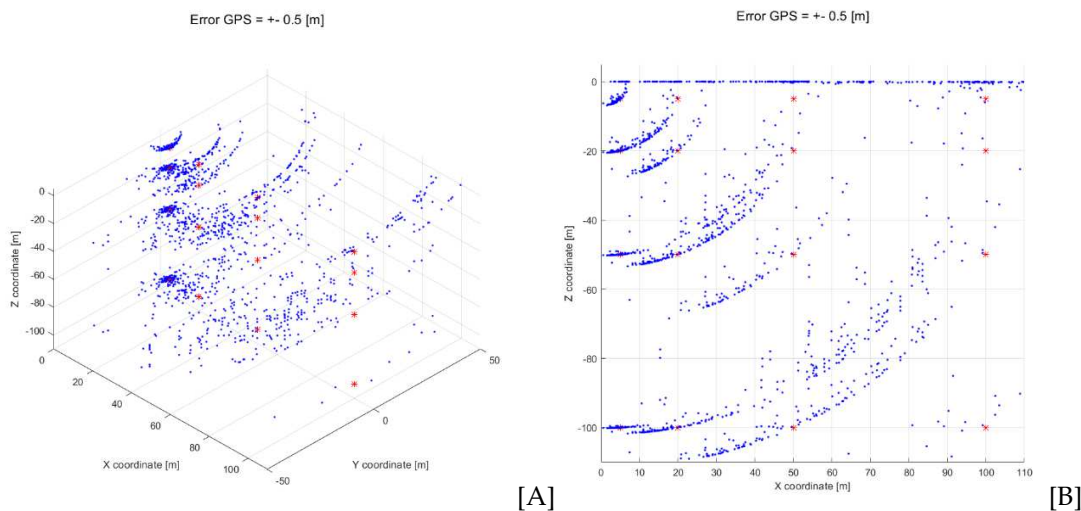


Figura 3.9 – Propagazione dell’errore del GNSS nel caso in cui il trasmettitore sia statico (strategia S1) ed il ricevitore statico; plot 3D X-Y-Z in [A], plot 2D X-Z in [B].

Risultati decisamente migliori si ottengono per la seconda strategia S2, in cui l’errore è sempre costante nell’intorno di ciascun punto in cui viene posizionato staticamente il ricevitore, e la distribuzione è proprio pari ad un quadrato di lato 1 metro perché l’errore introdotto dal GNSS ipotizzato in queste simulazioni è pari a $\pm 0,5$ [m]; si faccia riferimento alle figure 3.10, 3.11 e 3.12 per l’analisi dei risultati.

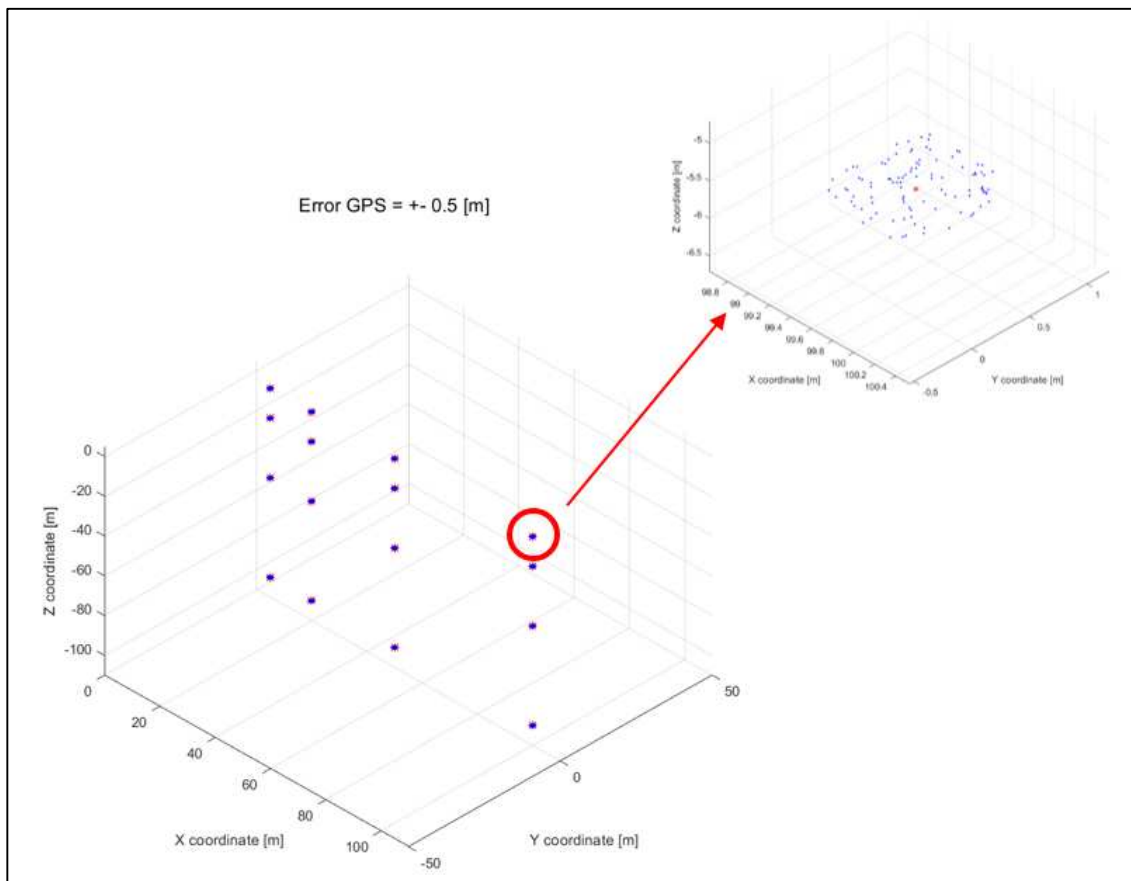


Figura 3.10 – Plot 3D X-Y-Z della propagazione dell'errore del GNSS nel caso in cui il trasmettitore sia statico (strategia S2) ed il ricevitore statico.

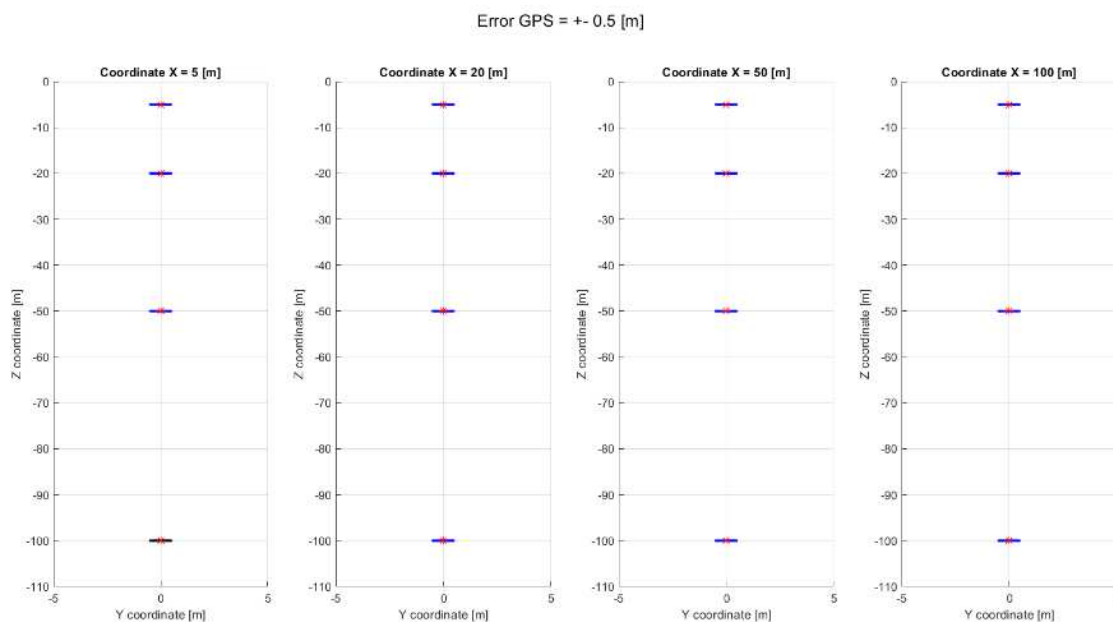


Figura 3.11 – Plot 2D Y-Z della propagazione dell'errore del GNSS nel caso in cui il trasmettitore sia statico (strategia S2) ed il ricevitore statico.

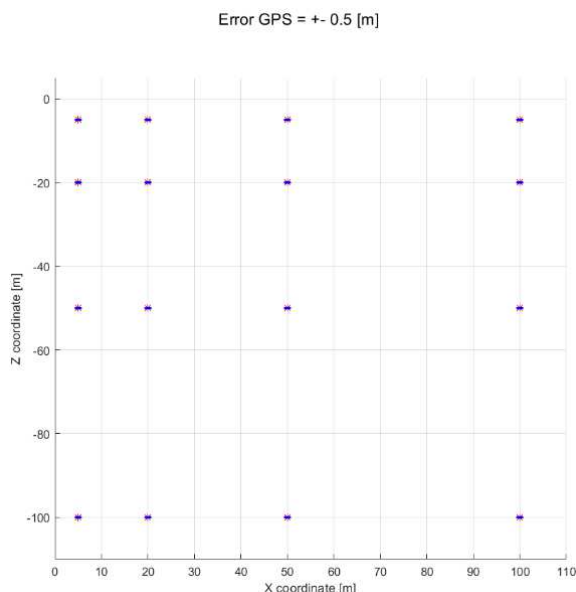


Figura 3.12 – Plot 2D X-Z della propagazione dell'errore del GNSS nel caso in cui il trasmettitore sia statico (strategia S2) ed il ricevitore statico.

3.3.1.2 - Trasmettitore dinamico

Si consideri adesso il caso in cui il trasmettitore acustico sia dinamico. Si immagina che esso sia montato sotto ad un veicolo di superficie. Ci si è posti come obiettivo quello di capire quale sia il modo più conveniente per aggiornare le posizioni dei sistemi di riferimento dei trasduttori A, B, C e D. Le strade individuate sono state essenzialmente due:

- fare l'aggiornamento del GNSS ed effettuare la traslazione lungo la baseline ogni 0.3 secondi (figura 3.13 A); per comodità indichiamo questa come strategia S3.
- campionare il valore del GNSS all'istante di inizio giro dei trasduttori (quindi da A) ed effettuare l'aggiornamento di B C e D mediante integrazione delle velocità (figura 3.13 B); in questo caso si introduce inevitabilmente un'ulteriore fonte di errore; per comodità indichiamo questa come strategia S4.

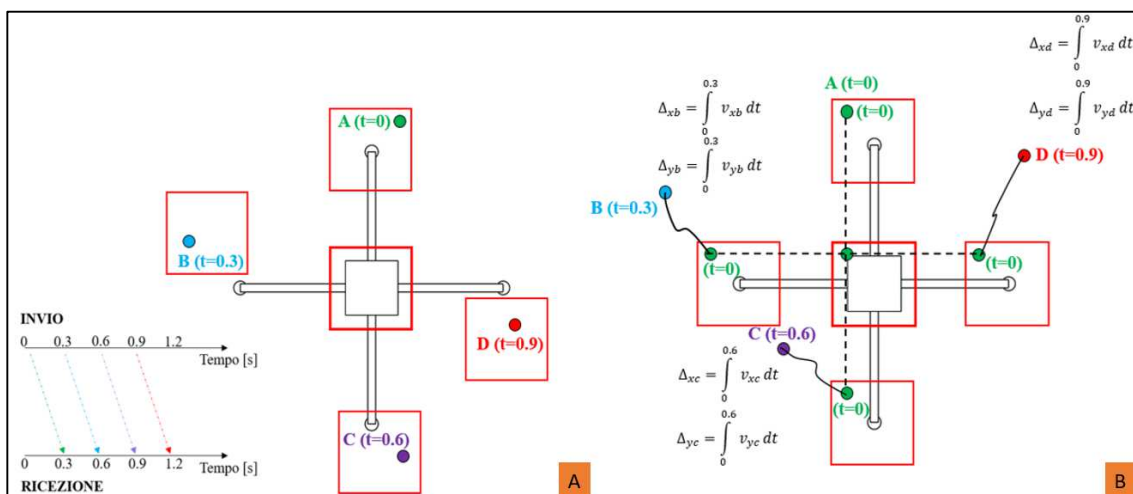


Figura 3.13 - Gestione del GNSS per trasmettitore dinamico. In [A] strategia S3 con aggiornamento multiplo del GNSS, in [B] aggiornamento posizioni mediante integrazione delle velocità (strategia S4).

Riassumendo quanto detto, da un lato si ha la strategia S3 che introduce per 4 volte l'errore del GNSS (una per ogni trasduttore), mentre dall'altro lato si ha la strategia S4 che introduce una sola volta l'errore del GNSS per il primo trasduttore A ma introduce errori causati dal calcolo delle posizioni dei sistemi di riferimento B, C e D dovuti alle integrazioni delle velocità.

Per praticità sono stati riportati solo i risultati ottenuti sulla prima riga di posizioni, e si è adoperato per queste simulazioni una strategia sulla triangolazione diversa che tiene conto di un sensor fusion con un sensore di profondità montato sul tablet subacqueo. Inoltre, per queste simulazioni si è tenuto conto di tutti gli errori nel modello. In figura 3.14 è stato riportato il risultato ottenuto utilizzando la strategia S3, ipotizzando un errore sui ranges pari a ± 0.5 [m] e sul sensore di profondità pari a ± 0.2 [m]. L'aggiornamento del GNSS ha previsto un tempo di 0.3 [s] mentre l'errore del GNSS in termini di posizionamento è pari a ± 0.5 [m].

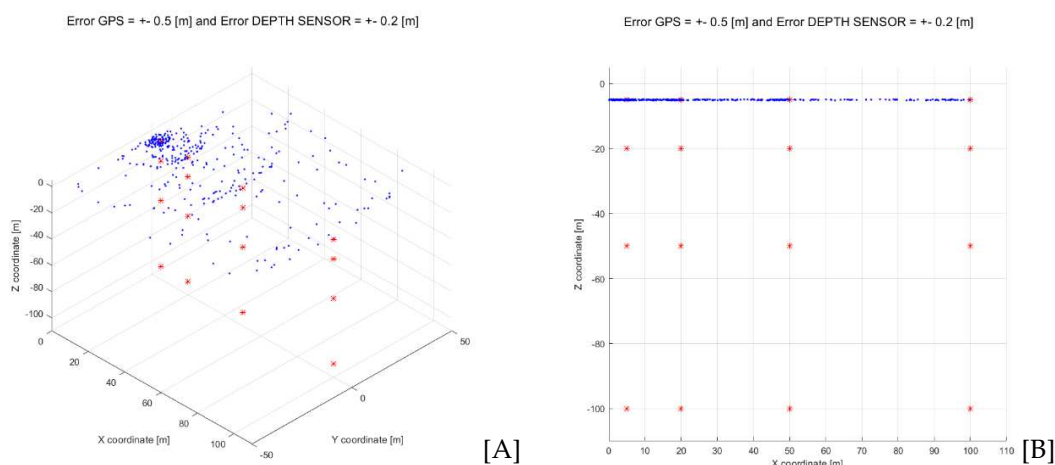


Figura 3.14 - Propagazione dell'errore del GNSS nel caso in cui il trasmettitore sia dinamico (strategia S3) ed il ricevitore statico. In [A] plot delle posizioni del ricevitore in X-Y-Z; in [B] plot delle posizioni del ricevitore in X-Z.

Per la strategia S4, si ottengono i risultati proposti in figura 3.15.

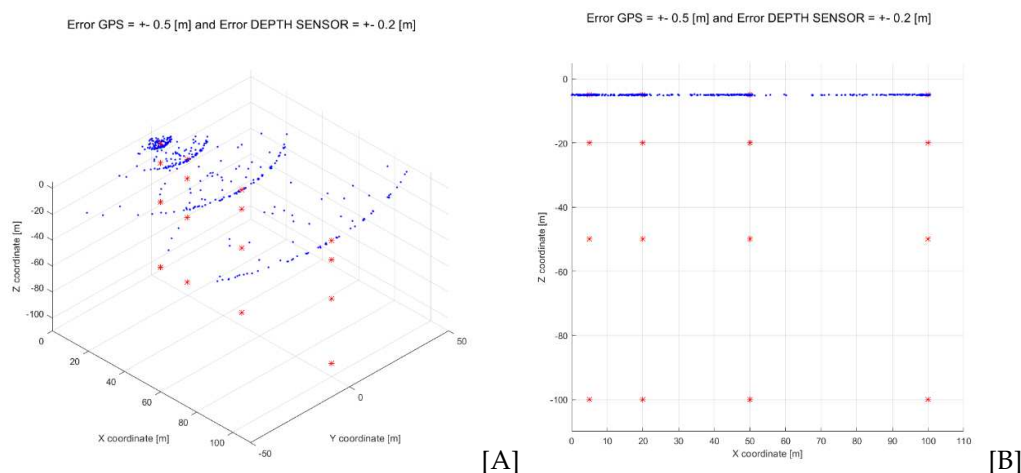


Figura 3.15 - Propagazione dell'errore del GNSS nel caso in cui il trasmettitore sia dinamico (strategia S4) ed il ricevitore statico. In [A] plot delle posizioni del ricevitore in X-Y-Z; in [B] plot delle posizioni del ricevitore in X-Z.

In quest'ultimo caso valgono le stesse considerazioni del caso precedente con l'unica differenza che il tempo di aggiornamento del GNSS è pari a 1.2 [s].

La differenza rispetto al caso precedente non è netta, questo perché nei risultati l'impatto dell'errore del range acustico gioca un ruolo non trascurabile. Per visualizzare meglio il beneficio della seconda strategia S4 rispetto alla S3 è stato abbassato l'errore del range acustico da ± 0.5 [m] a ± 0.1 [m]. In figura 3.16 è possibile osservare le due strategie S3 ed S4 messe a confronto.

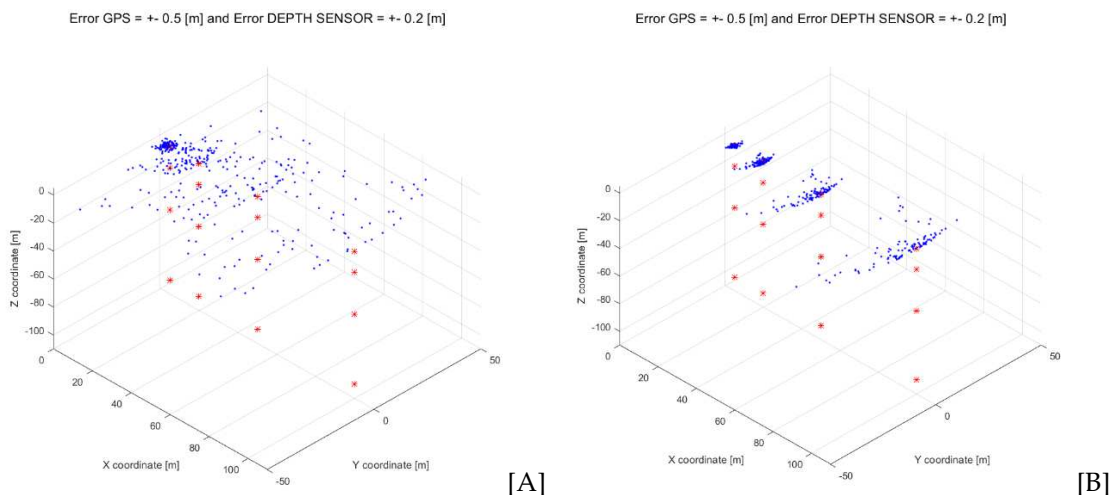


Figura 3.16 – Risultati della strategia S3 [A] e della strategia S4 [B] sulla stima della posizione del ricevitore acustico messi a confronto, nel caso in cui il trasmettitore sia dinamico, con errori sui ranges ridotti.

Integrando quindi nella logica di funzionamento una tecnica di tipo differenziale per la gestione del GNSS che memorizza ad intervalli di tempo il percorso del trasmettitore acustico, si sono ottenuti notevoli miglioramenti.

3.4 - Analisi di propagazione degli errori con ricevitore dinamico

Si consideri adesso un caso ancora più complesso, in cui il ricevitore acustico sia dinamico. In particolare, in questa sezione verrà effettuata una prima analisi in cui si analizzano gli effetti della triangolazione al variare della velocità del ricevitore acustico. Successivamente, sono presentati dei modelli di moto implementati in un filtro di Kalman esteso che calcola la posizione del ricevitore, ed è presentata una tecnica di sensor-fusion che prevede l'integrazione di una IMU all'interno del ricevitore. Sono state proposte alcune varianti di *sensor-fusion* con le velocità del tablet subacqueo ed il sensore di profondità.

3.4.1 - Effetto della velocità del ricevitore sulla triangolazione in un modello semplice

Si immagina che il ricevitore si muova lungo un percorso. Con la tecnica della triangolazione, si va a triangolare un punto equivalente "virtuale" che va ad unire le informazioni provenienti dai ranges che corrispondono ciclicamente a delle distanze di un punto che nel tempo tende a variare lungo un percorso. Questo introduce un'altra fonte di errore. Si faccia riferimento alla figura 3.17.

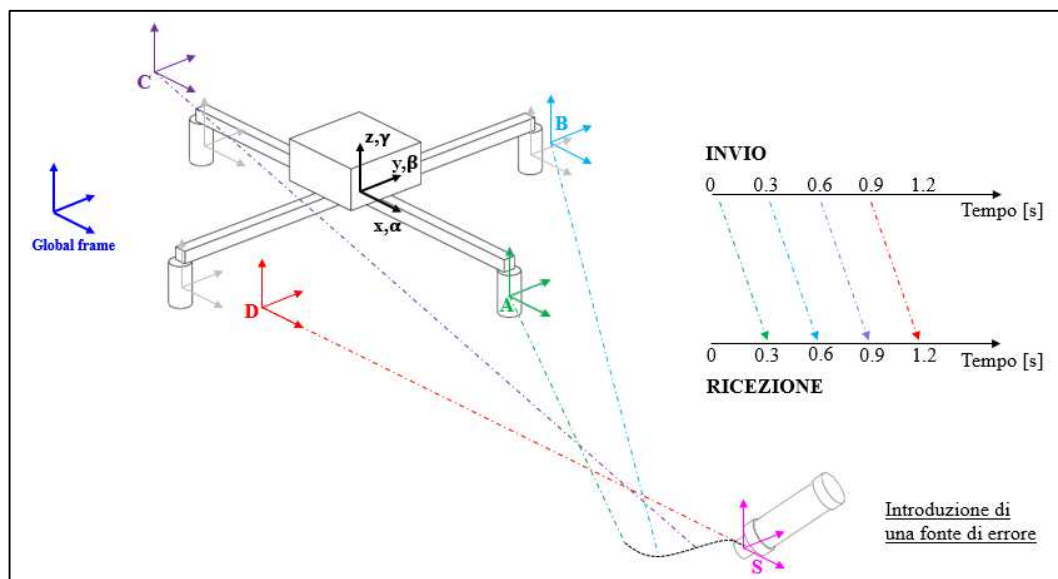


Figura 3.17 – Configurazione di funzionamento con ricevitore dinamico.

È stata avviata una simulazione con trasmettitore statico e ricevitore dinamico. Non sono state considerate fonti di errore nel modello, di nessuna tipologia. Utilizzando una strategia di triangolazione migliorata dalla presenza di un sensore di profondità), si è fatto muovere il ricevitore acustico: esso parte con velocità nulla, raggiunge un massimo di velocità circa pari a 0.65 [m/s] e finisce il tratto con velocità nulla, staziona nel vertice e riparte. La traiettoria utilizzata è un quadrato $37.5 \times 37.5 \text{ [m]}$. Si parte dal punto $(0,0)$ e si procede in senso anti-orario. L'obiettivo è stato quello di caratterizzare solo l'effetto dell'errore dovuto politica di gestione dei trasduttori in rotazione. Si è arrivati alla conclusione che l'errore aumenta all'aumentare della velocità, e come la base acustica (trasmettitore) è ruotata rispetto al ricevitore. In figura 3.18 sono riportati i risultati delle posizioni predette del ricevitore ed il percorso a forma quadrata.

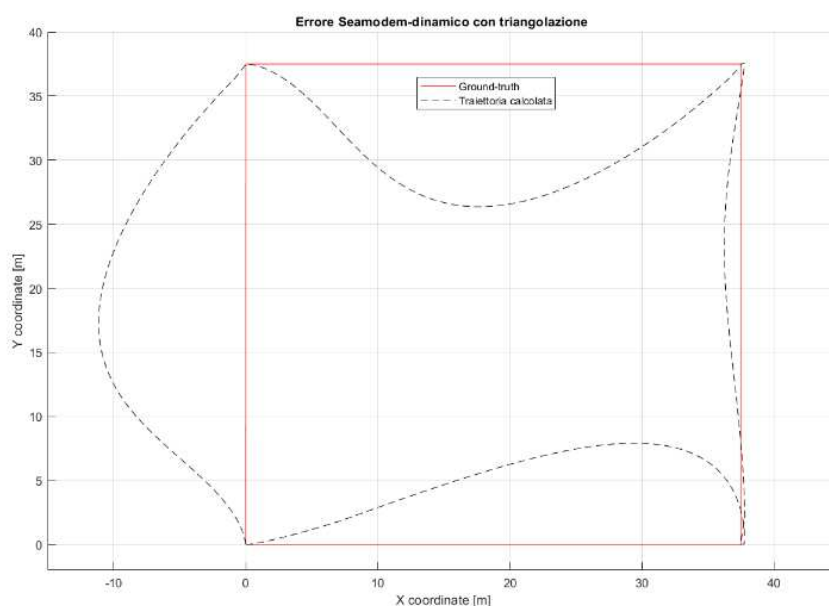


Figura 3.18 – Posizioni calcolate per il ricevitore dinamico che si muove a velocità variabile lungo un percorso quadrato, senza considerare fonti di errori riconducibili alla strumentazione.

3.4.2 - Filtro di Kalman Esteso (EKF)

Il filtro di Kalman è un efficiente filtro ricorsivo che valuta lo stato di un sistema dinamico a partire da una serie di misure soggette a rumore. È un ottimo filtro per rumori e disturbi agenti su sistemi gaussiani a media nulla. Inevitabilmente, nel caso studio in questione, si è dovuto ricorrere all'uso del Filtro di Kalman Esteso (EKF) data la non linearità del modello. In figura 3.19 è riportata la sequenza ricorsiva dell'EKF, nella quale si evincono le fasi di predizione e di correzione.

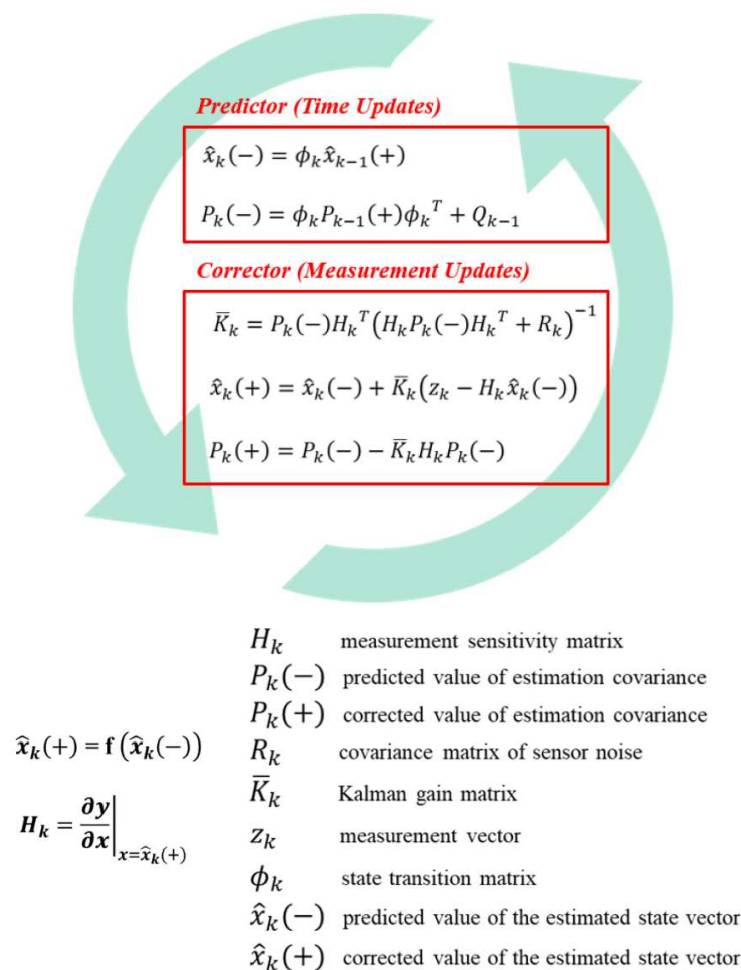


Figura 3.19 – Equazioni e termini del Filtro di Kalman Esteso (EKF).

In figura 3.20 sono riportate tutte le relazioni implementate nel filtro di Kalman esteso, a partire dalla costruzione delle equazioni alle differenze dello spazio di stato (composto dalle posizioni e dalle velocità del ricevitore) alla matrice Q di covarianza del rumore di processo, e le relazioni sulle misurazioni che sfruttano lo Jacobiano H per la loro implementazione, poiché non vi è un legame diretto con le variabili di stato. In particolare, sono presenti 4 ranges e le 3 velocità provenienti dal tablet (a seguito dell'integrazione delle accelerazioni) trasformate dal sistema di riferimento locale del tablet a quello globale. Si hanno inoltre le matrici di covarianza di rumore delle misurazioni: calibrando opportunamente le matrici di covarianza è possibile pesare l'impatto che le diverse misurazioni hanno sulla stima dello stato. Nelle equazioni di stato è implementato il modello cinematico, e questo anche viene tenuto in conto nella stima dello stato per ogni istante discreto.

SPAZIO DI STATO
Equazioni alle differenze

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ Z_{k+1} \\ \dot{X}_{k+1} \\ \dot{Y}_{k+1} \\ \dot{Z}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \\ Z_k \\ \dot{X}_k \\ \dot{Y}_k \\ \dot{Z}_k \end{bmatrix} + \begin{bmatrix} T_s/2 & 0 & 0 \\ 0 & T_s/2 & 0 \\ 0 & 0 & T_s/2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

$$\frac{X_{k+1} - X_k}{T_s} = \frac{\dot{X}_{k+1} + \dot{X}_k}{2}$$

$$\frac{Y_{k+1} - Y_k}{T_s} = \frac{\dot{Y}_{k+1} + \dot{Y}_k}{2}$$

$$\frac{Z_{k+1} - Z_k}{T_s} = \frac{\dot{Z}_{k+1} + \dot{Z}_k}{2}$$

$$Q = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

Osservazione dello STATO
Posizioni

$$Y_1 = R_A = \sqrt{(X_S - X_A)^2 + (Y_S - Y_A)^2 + (Z_S - Z_A)^2}$$

$$Y_2 = R_B = \sqrt{(X_S - X_B)^2 + (Y_S - Y_B)^2 + (Z_S - Z_B)^2}$$

$$Y_3 = R_C = \sqrt{(X_S - X_C)^2 + (Y_S - Y_C)^2 + (Z_S - Z_C)^2}$$

$$Y_4 = R_D = \sqrt{(X_S - X_D)^2 + (Y_S - Y_D)^2 + (Z_S - Z_D)^2}$$

$$H = \begin{bmatrix} \frac{X_S - X_A}{R_A} & \frac{Y_S - Y_A}{R_A} & \frac{Z_S - Z_A}{R_A} & 0 & 0 & 0 \\ \frac{X_S - X_B}{R_B} & \frac{Y_S - Y_B}{R_B} & \frac{Z_S - Z_B}{R_B} & 0 & 0 & 0 \\ \frac{X_S - X_C}{R_C} & \frac{Y_S - Y_C}{R_C} & \frac{Z_S - Z_C}{R_C} & 0 & 0 & 0 \\ \frac{X_S - X_D}{R_D} & \frac{Y_S - Y_D}{R_D} & \frac{Z_S - Z_D}{R_D} & 0 & 0 & 0 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}$$

Velocità

$$Y_1 = \dot{X}$$

$$Y_2 = \dot{Y}$$

$$Y_3 = \dot{Z}$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_2 = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

Figura 3.20 – Relazioni implementate nel filtro di Kalman esteso.

In figura 3.21 è riportato lo schema a blocchi del filtro di Kalman esteso implementato su Simulink. Le equazioni viste prima sono implementate nel blocco con contorno rosso sulla destra. Viene simulata la logica di arrivo sfasata dei 4 ranges, con tempo di campionamento ogni 1.2 secondi per ciascuno di essi ma ogni 0.3 il filtro riceve un valore secondo l'ordine A B C D, mentre per le velocità ogni 0.3 secondi. Nelle Matlab-function sono implementate le logiche di chiusura e apertura delle porte per l'arrivo dei dati nel filtro, ed è possibile simulare una percentuale di pacchetti persi nella comunicazione.

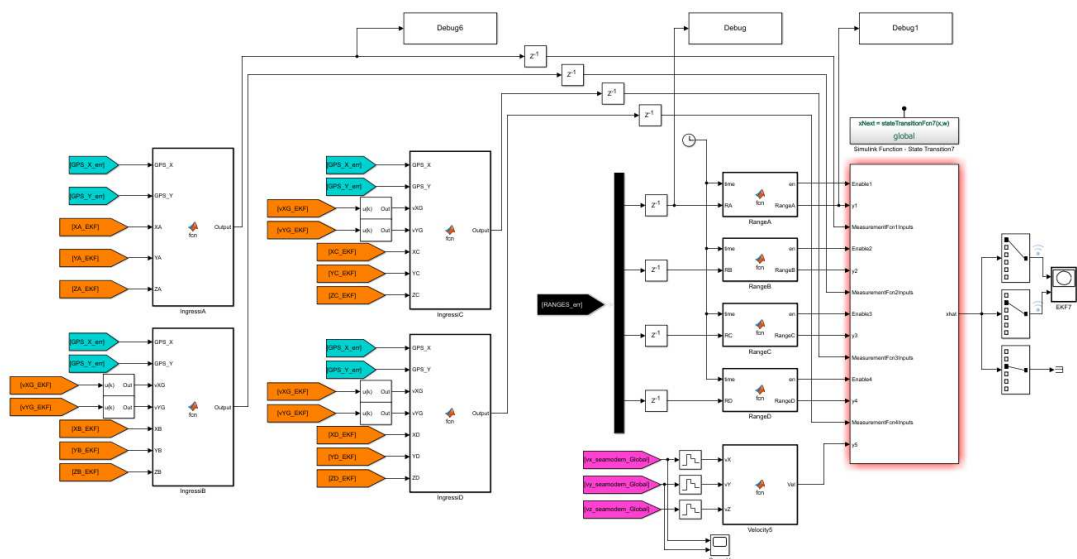


Figura 3.21 – Rappresentazione a blocchi del filtro di Kalman esteso implementato in Simulink.

3.4.3 – Sensor-fusion del sistema acustico con la IMU del tablet e sensore di profondità

È presentata qui di seguito un'analisi che ha previsto un sensor-fusion del sistema acustico con le velocità del tablet (IMU) ed un sensore di profondità. In figura 3.22 sono riportati alcuni risultati, che mettono a confronto in rosso una traiettoria quadrata ideale, in blu il percorso ottenuto dalla doppia integrazione delle accelerazioni nel tablet (si verifica l'effetto drift dovuto all'effetto di un leggero bias e di rumore) ed in nero la logica con EKF discusso in precedenza che prevede l'utilizzo dei ranges e delle velocità del tablet ottenute per integrazione delle accelerazioni.

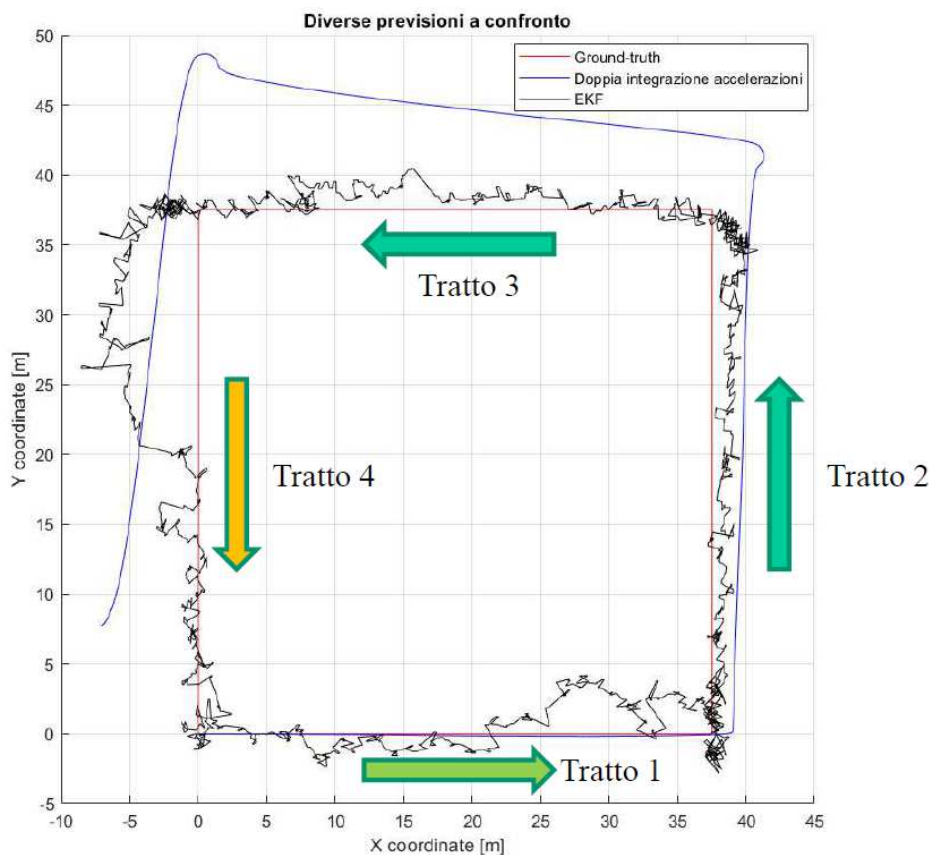


Figura 3.22 – Diverse previsioni a confronto per la stima della posizione del ricevitore (tablet) subacqueo. Linea rossa per il percorso, linea blu per la posizione stimata dalla doppia integrazione delle accelerazioni del tablet, linea nera per la posizione stimata con il EKF.

L'analisi dei risultati permette di trarre le seguenti conclusioni per la previsione dell'EKF:

- tratto 1: accettabile;
- tratto 2: ottimo;
- tratto 3: ottimo;
- tratto 4: discreto.

Nonostante l'incertezza tutto sommato accettabile per la maggior parte del percorso, la previsione riesce a seguire il quadrato e a chiudersi nel punto di inizio (0,0). Per questa simulazione condotta, si è accentuato di molto il rumore sulle accelerazioni del tablet e si è data più fiducia alle misurazioni dei ranges.

3.5 – Sensor-fusion con dati ottici (visual-inertial odometry)

Questa sezione descrive alcune metodologie che migliorano il tracciamento della posizione del subacqueo nell'ambiente sottomarino. L'idea è quella di utilizzare la tecnologia acustica in combinazione con quella ottica per migliorare la precisione seguendo l'approccio già sperimentato nel progetto iMARECulture [3.04]. La tecnologia ottica consente un tracciamento ottimale, anche sott'acqua, solo per brevi tratti. La tecnologia acustica, invece, garantisce una buona navigazione a livello globale ma non locale. Sono state sviluppate alcune strategie ibride per una navigazione più precisa. In data 05/07/2018 sono state avviate le prime prove sperimentali a Le Castella, in provincia di Crotone. La trattazione riguarda la descrizione e l'implementazione di strategie per la correzione dei dati acquisiti durante i test sperimentali condotti.

3.5.1 - Metodi

In questa sezione è presentata una descrizione della strumentazione utilizzata nell'architettura di localizzazione ibrida. L'architettura di localizzazione ibrida utilizzata per gli esperimenti ha previsto che il sub utilizzi un tablet subacqueo per consentire il funzionamento del framework ARKit, che mette a disposizione un algoritmo di natural feature tracking in grado di stimare gli spostamenti della camera del tablet nell'ambiente. Per il tracciamento della posizione tramite comunicazione acustica si è deciso di utilizzare, in diverse prove, la strumentazione SBL (sviluppata dall'azienda Applicon s.r.l.).

Un'App sviluppata da 3DResearch s.r.l. (figura 3.23) installata sul tablet, ha permesso al subacqueo di avere informazioni in tempo reale su profondità, orientamento, rotta, temperatura dell'acqua, intensità del segnale acustico e qualità del segnale ottico. Inoltre, l'App ha permesso di registrare un video tramite una registrazione dello schermo consentita dalla fotocamera del tablet, utile nella fase di post-elaborazione dei dati.

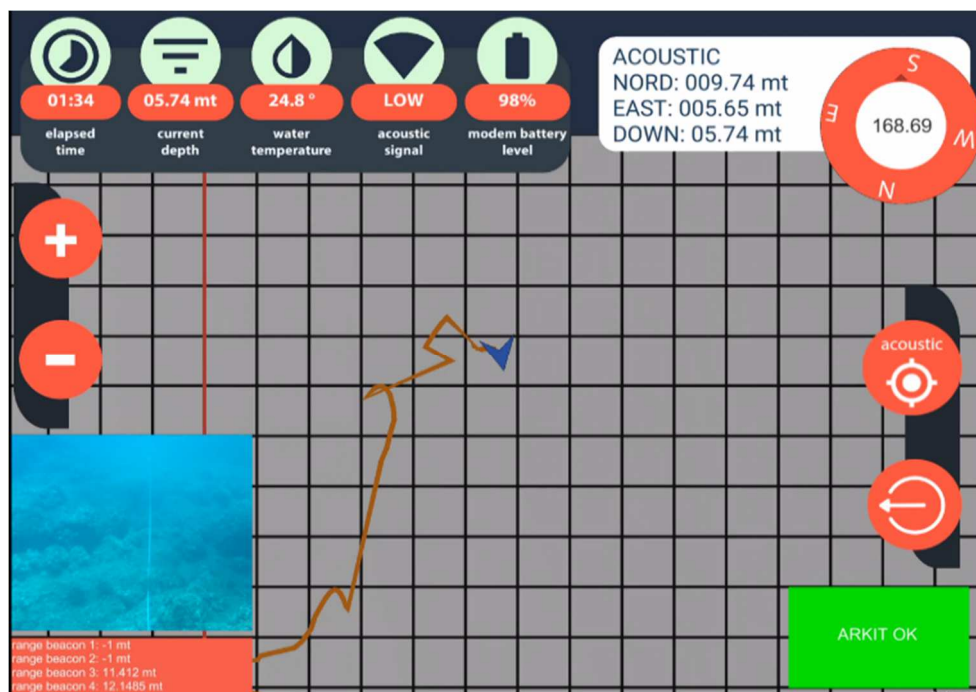


Figura 3.23 – Interfaccia dell'App sviluppata dall'azienda 3DResearch.

3.5.2 - Test sul campo

Dopo una fase di studio delle tecnologie appena descritte, è stato progettato un test subacqueo, realizzato nel sito di Le Castella, in provincia di Crotone. L'obiettivo di questo test era quello

misurare le prestazioni del sistema di posizionamento ibrido (acustico + ottico), paragonando i risultati ottenuti a quelli acquisiti utilizzando solo sistema di posizionamento acustico. Muovendosi all'interno del sito sottomarino, la posizione ottenuta è stata confrontata con una ground-truth (punti noti predefiniti del percorso).

3.5.2.1 - Protocollo dei test

L'attività sperimentale ha previsto le seguenti operazioni preliminari:

- Fase 1. Costruzione della ground-truth sott'acqua. È stato necessario definire un percorso noto per avere un riferimento per la valutazione delle prestazioni. Il percorso proposto era di forma quadrata (figura 3.24). Sono stati necessari quattro marker (A-B-C-D) per costruire i vertici e quattro stringhe graduate per costruire i lati. Il quadrato ha il lato BC rivolto a nord.

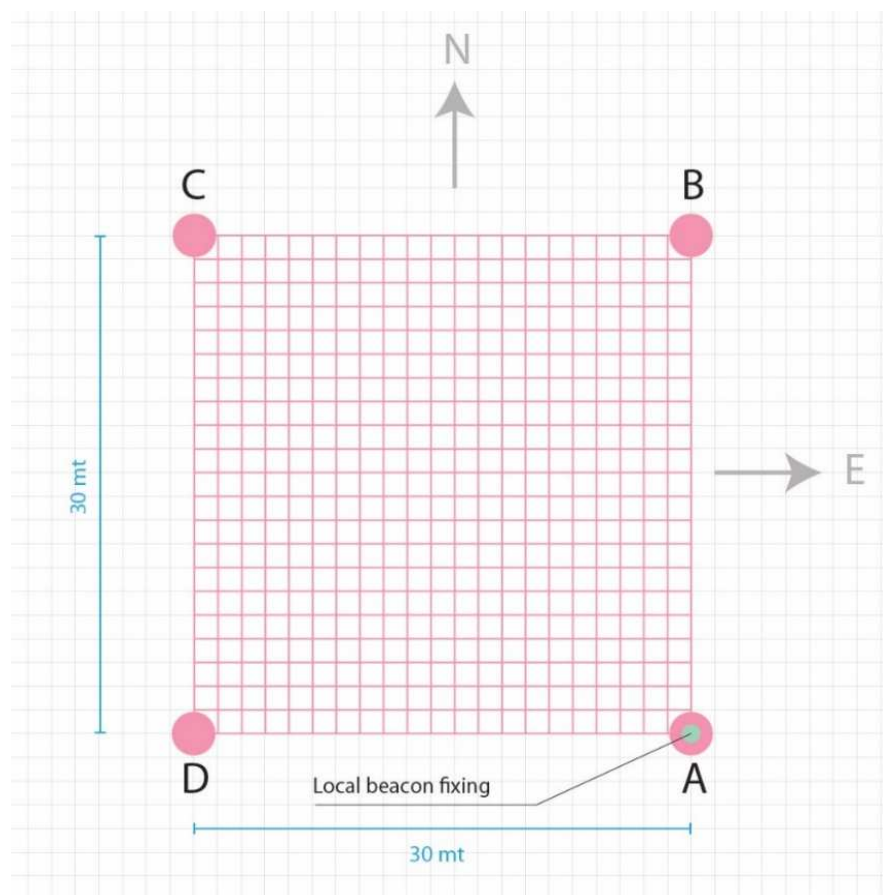


Figura 3.24 – Ground-truth teorica.

- Fase 2. Calibrazione sulla barca del sistema di localizzazione acustico.
- Fase 3. Calibrazione sulla barca del sistema ARKit. Il sistema di orientamento di ARKit è stato inizializzato per ogni sessione di immersione puntando il tablet verso nord lungo il lato AB.
- Fase 4. Messa in mare del sistema di localizzazione acustico. È stato previsto che la base del sistema di localizzazione acustico fosse posta in mare, posizionata nel vertice A. La configurazione è mostrata nella figura 3.25.

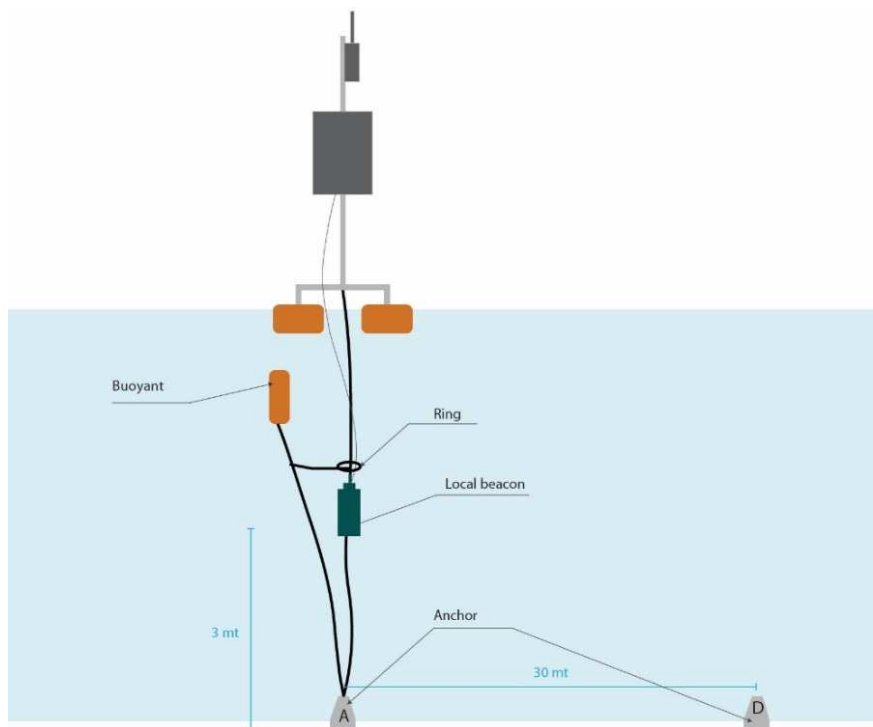


Figura 3.25 – Architettura della base del sistema di localizzazione acustico.

- Fase 5. Misurazione della profondità della base.
- Fase 6. Prove di funzionamento del sistema sulla superficie dell'acqua.
- Fase 7. Annotare la dimensione effettiva del quadrato e altri scostamenti rispetto a quanto indicato. Questa fase è stata importante per avere un riferimento preciso per confrontare i dati.
- Fase 8. Avvio della registrazione dello schermo sull'Ipad.

3.5.2.2 - Esecuzione dei test

Una volta avviata l'App sul punto A, e dopo aver atteso il “fixing” del segnale (riconoscimento del piano ARKit) è stato possibile avviare i test.

L'esperimento ha previsto che il subacqueo partisse dal vertice A del quadrato e percorresse 3 giri, fermandosi qualche secondo ogni volta aver raggiunto uno dei vertici. Durante tutto il percorso, il sistema ha raccolto e salvato i dati acustici e ottici ad una frequenza di campionamento presumibilmente costante.

Ipotizzando che vi sia un confronto con un sistema di riferimento esatto solo nei vertici del quadrato, soffermandosi più tempo in tali vertici è stato possibile avere più dati che sono stati poi confrontati proprio con un riferimento certo per un calcolo della dispersione dei dati. I dati intermedi tra un vertice e l'altro sono serviti solo da un punto di vista grafico.

In questo modo è stato possibile confrontare i soli dati acustici e ibridi registrati durante le acquisizioni, rispetto alla ground-truth. La tab.1 mostra un elenco degli esperimenti eseguiti con il tipo di trasduttore acustico utilizzato, e il numero totale di giri sulla ground-truth.

Tab. 1 – Descrizione di test eseguiti con sistema SBL.

Experiment	Acoustic transducer	ARKit	Complete number of laps of the route	Number of points acquired
Dive session 1	SBL (Seamodem)	Yes	3	15643
Dive session 2	SBL (Seamodem)	Yes	3	12602

3.5.2.3 - Descrizione della ground-truth

La ground-truth costruita, a causa della difficoltà di lavorare sott'acqua, ha presentato una forma diversa da quella di un quadrato. Le misure riportate al termine della sua costruzione sott'acqua, sono state le seguenti:

- Lato AB: 30 [m];
- Lato BC: 30 [m];
- Lato CD: 30 [m];
- lato AD: 29,26 [m];
- diagonale BD: 43,3 [m];
- diagonale AC: 41 [m];

Costruendo un modello geometrico si sono ricavate le posizioni dei quattro vertici, e di conseguenza le inclinazioni dei lati. Nella fase di post-elaborazione sono stati analizzati i video. Guardando con attenzione i video delle prove successive, considerando che sull'interfaccia del tablet c'è l'informazione sulla bussola, si è notato dallo screen-recorder che la ground-truth era costruita con il lato AC inclinato di circa 11 gradi rispetto alla direzione nord, quindi l'intera ground-truth è stata ruotata di 11 gradi in senso antiorario. In fig.3.26 è mostrata la ground-truth costruita.

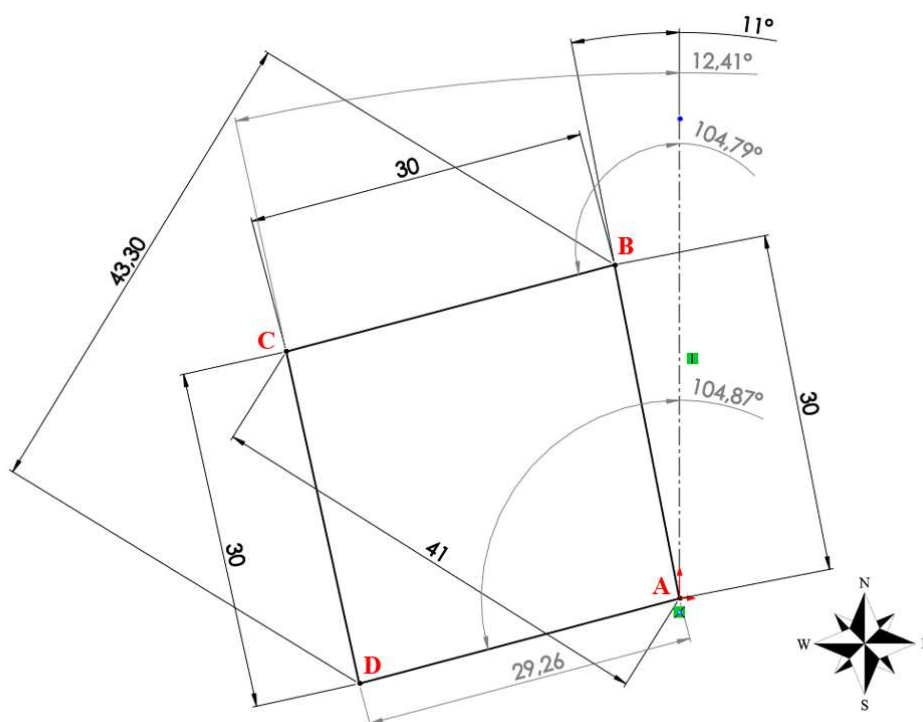


Figura 3.26 – Ground-truth costruita.

Considerando che sull'interfaccia dell'App c'era anche il tempo trascorso, è stato possibile individuare il tempo in cui il subacqueo stazionava sui vertici della ground-truth. In particolare, ci si è concentrati sulla marcatura di inizio e fine per ogni vertice.

3.5.3 - Descrizione dei files di log

Tutte le informazioni relative alle prove sperimentali sono state salvate in files di formato Json, uno per ogni Dive-session, che sono stati successivamente post-elaborati. Nel caso in questione,

il file Json è strutturato in righe, ciascuna per ogni istante di tempo campionato. In particolare, ogni riga contiene le seguenti informazioni:

- timestamp: informazioni sulla data e sull'ora, con una precisione fino al millesimo di secondo; questi valori sono stati poi resi disponibili in un sistema di riferimento temporale assoluto.
- seamodem_position/divyboard_position: informazioni sulla posizione X, Y e Z del sistema SBL. La convenzione utilizzata è stata la seguente:
 - +Z : nord
 - +X : est
 - +Y : direzione verso la superficie.

Bisogna tenere presente che:

- per la Dive-session 1: la piattaforma inerziale ha misurato un errore di orientamento di circa 30 gradi. I dati di log relativi alle posizioni del ricevitore sono stati quindi affetti dall'errore della piattaforma inerziale. Quindi è stata fatta una correzione software sulle coordinate usando una matrice di trasformazione.
- per la Dive-session 2: stesso test della Dive-session 1, ma le posizioni del ricevitore all'interno del log sono state corrette preliminarmente via software applicando la correzione di 30 gradi.
- seamodem_ranges: queste informazioni si riferiscono alla configurazione SBL, ma sono risultate essere errate. Le informazioni reali sugli intervalli sono state incluse tramite un altro file fornito da Applicon s.r.l., e sono state adattate alla dimensione del file JSON nel pieno rispetto dei periodi di tempo in cui sono state acquisite. Osservare la tab.2.

Tab. 2 – Informazioni riguardanti i ranges.

Experiment	Name of JSON files	N° of points acquired in JSON files	Name of Applicon files	N° of point aquired in Applicon files
Dive session 1	Seamodem_1	15643	Kalman01	5739
Dive session 2	Seamodem_2	12602	Kalman44	4794

- seamodem_quality: questa informazione si riferisce alla configurazione SBL. La qualità del segnale acustico è stata suddivisa in 4 categorie:
 - Nessuno=0;
 - Basso=1/2;
 - Medio=3;
 - Alto=4;

I numeri rappresentano il numero di trasduttori che hanno ricevuto il segnale in quel momento.

- arkit_position: informazioni sulla posizione X, Y e Z del sistema ARKit. La convenzione utilizzata è stata la stessa di quella adottata dal sistema acustico. Il sistema di riferimento ARKit è stato inizializzato per ogni sessione di immersione puntando il tablet verso nord lungo il lato AB.
- arkit_state. La qualità del segnale acustico è stata suddivisa in 3 categorie:
 - ARTrackingStateNotAvailable
 - ARTrackingStateLimited
 - ARTrackingStatoNormale
- arkit_state_reason utile per monitorare le seguenti informazioni:

- ARTrackingStateReasonNone;
- ARTrackingStateReasonInsufficientFeatures.
- hybrid1_position e hybrid2_position: a causa degli errori della piattaforma inerziale dei vari sistemi di tracciamento, le posizioni di Hybrid1 e Hybrid2 sono state ricalcolate a partire dai dati di posizionamento di ARKit.
- sensor: nella sottosezione bussola sono state rese disponibili le informazioni relative alla bussola.

3.5.4 - Descrizione del software di elaborazione dati

Nella fase di post-elaborazione dei file JSON si è deciso di utilizzare il software Matlab/Simulink per riorganizzare i dati e apportare tutte le correzioni necessarie sia per i grafici sia per le strategie. Il software riconosce automaticamente se si tratta di un'architettura SBL oppure USBL (per l'architettura USBL fare riferimento al Capitolo 5). In particolare, il software prevede l'utilizzo di due sezioni:

- Sezione di pre-elaborazione: in questa sezione è creata una struttura ".mat" per ogni sessione di immersione, all'interno della quale sono presenti tutti i dati corretti in base ai vari problemi elencati nel paragrafo precedente, come ad esempio quello sull'orientazione e l'inserimento dei ranges. Pertanto, vengono salvati i dati acustici e ottici corretti, con le relative qualità dei segnali. Per ogni Dive-session è stata prevista la costruzione di una ground-truth dinamica che tiene conto con esattezza, in base al tempo, dello stato di permanenza del sub nei vertici del quadrato. Sono stati quindi costruiti dei diagrammi temporali t-X e t-Z. Tutti i dati sono stati convertiti in un sistema di riferimento temporale assoluto che inizia a 0 secondi per ogni sessione di immersione. Sono state inoltre salvate le informazioni relative alla lunghezza, al tempo e al tipo di trasduttore utilizzato.
- Sezione di Plot e sviluppo di strategie: in questa sessione è possibile caricare una sessione di immersione partendo dalla struttura specifica (.mat), ed è possibile generare una serie di grafici in modo automatizzato.

In particolare, sia per l'acustico sia per l'ottica, ed anche per gli ibridi, è possibile ottenere:

- percorso X-Z;
- percorso X-Z colorato in base alla qualità (numero di transponder);
- percorso X-Z colorato gradualmente secondo la direzione di avanzamento;
- coordinata X rispetto al tempo (colorata in base alla qualità solo per SBL);
- coordinata Z rispetto al tempo (colorata in base alla qualità solo per SBL);
- grafici Intervalli di lunghezza rispetto al tempo (valido solo per SBL);
- grafici ground-truth sia nei diagrammi X-Z che nei diagrammi t-X e t-Z;
- per ogni data set è possibile creare sottoinsiemi per analizzare i dati in specifici intervalli inseriti dagli utenti;
- è possibile visualizzare i dati relativi all'intero percorso, e solo nei momenti in cui il subacqueo è fermo nei singoli vertici;
- grafici di errore nei diagrammi t-X e t-Z;
- calcolo della frequenza di ricezione dei dati acustici ed ottici, anche tra due limiti di tempo predeterminati dall'utente.

3.5.5 – Risultati preliminari

Durante la prima sessione di immersione sono stati registrati i seguenti dati: in particolare, in figura 3.27(a) e figura 3.27(b) sono rappresentati tre giri mediante colorazione progressiva sul

piano X-Z, rispettivamente per i dati ottici e per i dati acustici. Con linea rossa tratteggiata è indicata la ground-truth.

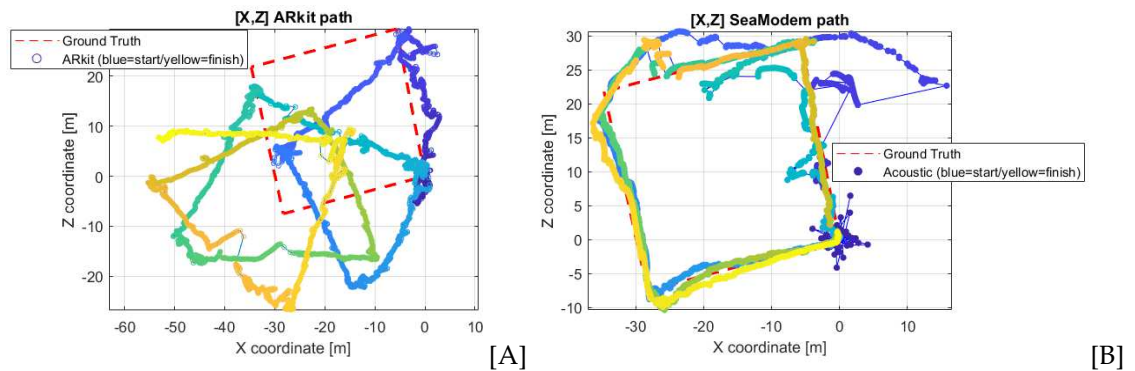


Figura 3.27 – Dati ottici (a) e dati acustici (b) nelle coordinate X-Z per la Dive Session 1.

In figura 3.28(a) e figura 3.28(b) è stato rappresentato l'andamento delle coordinate X e Z rispetto al numero di punti acquisiti. Con la linea continua rossa è rappresentata la ground-truth.

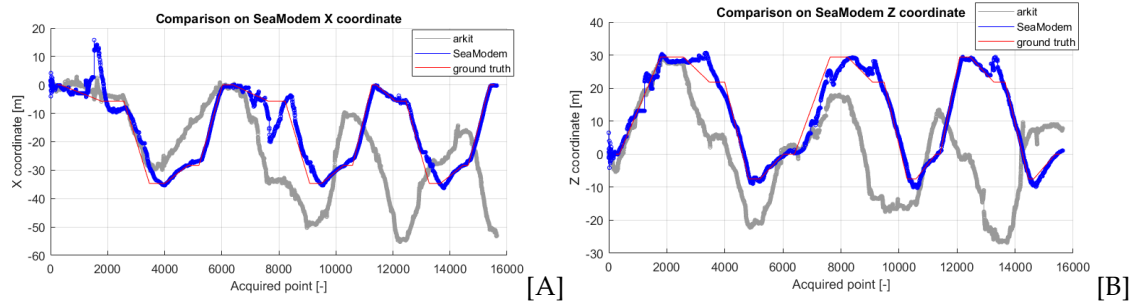


Figura 3.28 – Dati ottici ed acustici in coordinate NumberAcquisitionPoint-X (a) e NumberAcquisitionPoint-Z (b) per la Dive Session 1.

Per la Dive session 2 sono stati ricreati gli stessi grafici rispettivamente riportati nelle figure 3.29 e 3.30.

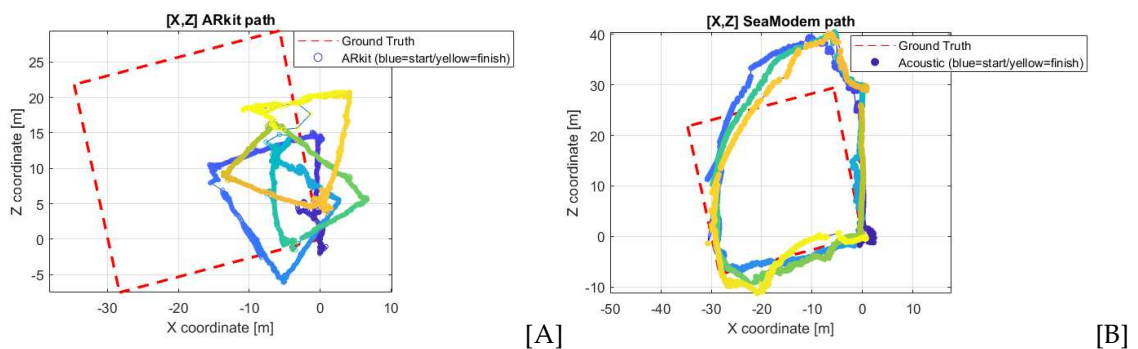


Figura 3.29 – Dati ottici (a) e dati acustici (b) nelle coordinate X-Z per la Dive Session 2.

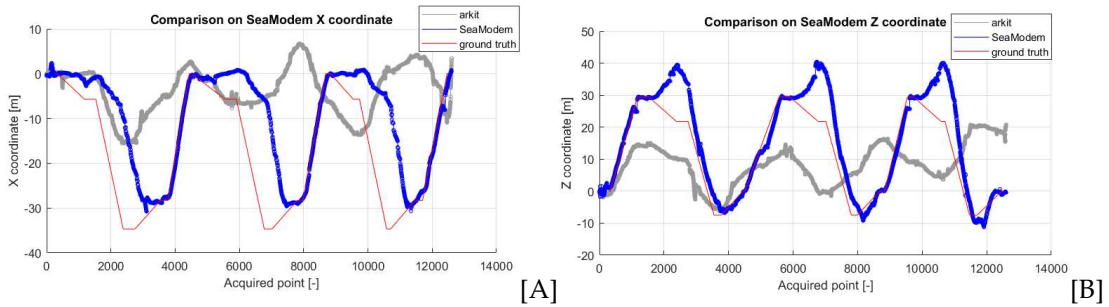


Figura 3.30 – Dati ottici ed acustici in coordinate NumberAcquisitionPoint-X (a) e NumberAcquisitionPoint-Z (b) per la Dive Session 2.

In figura 3.31(a) e figura 3.31(b), per la configurazione SBL, le misure dei ranges sono mostrate in un diagramma Tempo-LunghezzaRanges, rispettivamente per Dive Session 1 e Dive Session 2.

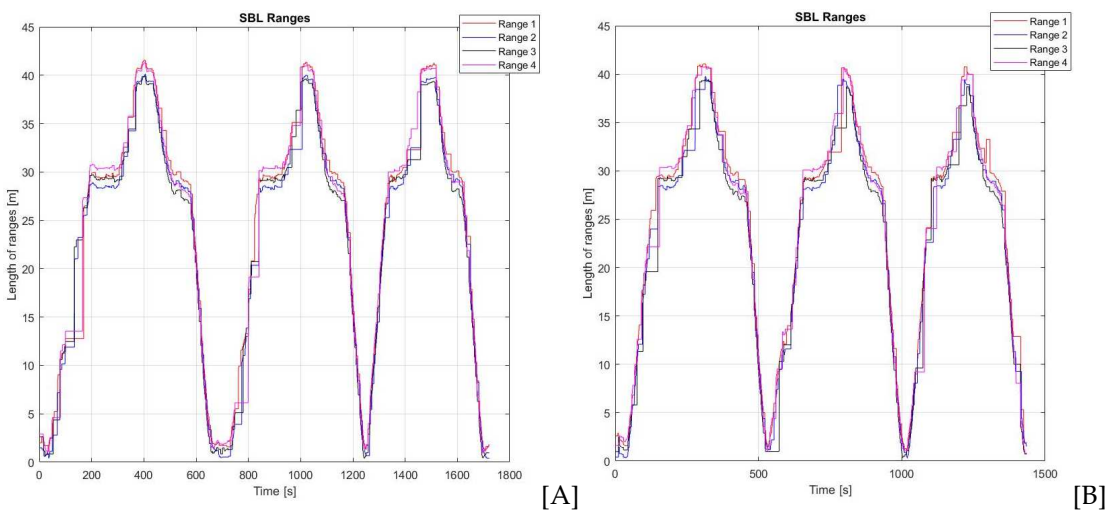


Figura 3.31 – Grafici Tempo-LengthRanges per la Dive Session 1 (a) e la Dive Session 2 (b).

Nella figura 3.32(a) e 3.32(b), rispettivamente per la Dive Session 1 e per la Dive Session 2, in un diagramma X-Z i dati acustici sono rappresentati solo nei momenti in cui il subacqueo è fermo nei vertici della ground-truth. Questi dati sono stati suddivisi in base alla qualità: rosso per segnale NESSUNO, blu per segnale BASSO, nero per segnale MEDIO e verde per segnale ALTO. Con una linea tratteggiata rossa è rappresentata la ground-truth.

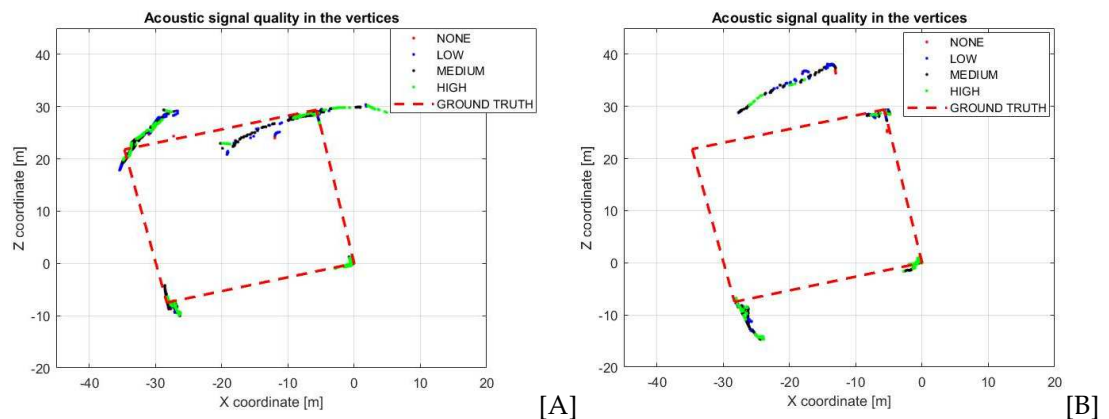


Figura 3.32 – Dati acustici classificati in base alla qualità in un grafico X-Z in prossimità dei vertici della ground-truth; (a) dati per la Dive Session 1, (b) dati per la Dive Session 2.

3.5.6 - Descrizione della strategia di hybrid tracking 2.0

Si è deciso di implementare una strategia attraverso l'utilizzo simultaneo di dati acustici e ottici, con l'obiettivo di ottenere un miglioramento complessivo nella precisione della posizione del sub rispetto alla ground-truth. Il ricevitore acustico utilizzato ha una velocità di aggiornamento di circa 0,1 secondi per ricevere un dato, e considerando che per il ricevitore è possibile avere informazioni sulla qualità del segnale, è stata prevista una strategia che ne tenesse conto.

La posizione ibrida è stata calcolata come media ponderata tra la posizione assoluta del sistema acustico e quello ottico di ARKit. I pesi si basano sulla qualità attuale dei due sistemi di posizionamento. La posizione locale di ARKit è trasformata in posizione assoluta, e tale posizione viene calcolata con riferimento all'ultima posizione acustica di alta qualità. In figura 3.33 il risultato è presentato in un diagramma X-Z. La ground-truth è tracciata con una linea rossa tratteggiata. I dati ibridi sono rappresentati in progressione temporale mediante un colore scatter che va dal blu al giallo, mentre i dati acustici sono presentati con dei punti neri. Si è scelto di rappresentare la strategia applicata solo al primo giro della Dive-session 1.

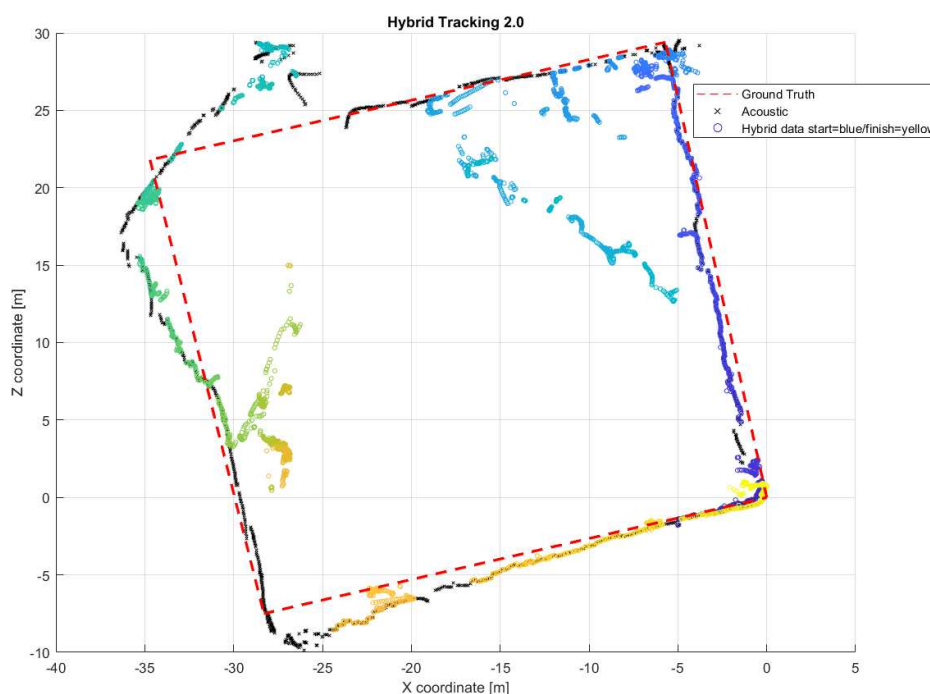


Figura 3.33 – Strategy 2.0 applicata al primo giro della Dive session 1.

Il sistema di localizzazione ibrido, basato sull'utilizzo del sistema acustico SBL, ha presentato risultati interessanti, sia per il fatto che l'architettura acustica ha una buona frequenza di aggiornamento, sia perché è possibile sfruttare le informazioni sulla qualità del segnale per elaborare una strategia più performante. In generale, i dati acquisiti da ARKit in questi test non sono stati eccellenti, quindi i risultati delle prove sono stati accettabili ma sicuramente migliorabili nei successivi test futuri. Alcune future strategie potrebbero prevedere l'introduzione del filtro di Kalman esteso per la fusione dei dati.

Capitolo 4

Integrazione del sistema SBL con l'ASV.

A seguito degli studi effettuati in simulazione, si è passati allo sviluppo del software, realizzato in ambiente ROS (Robot Operating System), per l'interfacciamento del veicolo autonomo di superficie ASV con il sistema acustico SBL. È descritto l'algoritmo di inseguimento adottato dal veicolo per inseguire il diver equipaggiato da tablet. È stata prevista un'interfaccia grafica realizzata sotto forma di web-app per il controllo degli elementi in azione; questa interfaccia permette la visualizzazione dei vari dispositivi su una mappa, e permette l'attuazione di comandi ed il monitoraggio di dati, con l'obiettivo di facilitare la gestione informatica del processo che altrimenti andrebbe condotta a "basso livello". Sono mostrati successivamente i risultati sperimentali ottenuti a seguito di alcuni test con l'obiettivo di appurare il corretto funzionamento degli algoritmi implementati. Sono inoltre presentati protocolli e metodologie per i test sperimentali dinamici e statici da condurre con il sistema finale.

4.1 - Interfacciamento del sistema SBL su piattaforma ROS

Il veicolo ASV H2Omni-X è un veicolo autonomo di superficie che può essere controllato sia in modo manuale sia in modo automatico. Esso è stato programmato dagli sviluppatori utilizzando un framework di nome ROS (Robot Operating System) basato su Linux. Il sistema acustico SBL di nome AULoS, progettato da Applicon s.r.l., è stato programmato dagli sviluppatori utilizzando un altro framework. Uno scopo preliminare del lavoro è stato quello di sviluppare un package che permettesse l'interfacciamento tra le due tecnologie, all'interno del quale sono state poi aggiunte altre funzionalità utili alla navigazione autonoma. All'interno di questo paragrafo è spiegata tutta la parte che è stata sviluppata sul sistema operativo Linux, partendo inizialmente da una panoramica generale del framework ROS per metterne in risalto le potenzialità, per poi passare alla descrizione dell'interfacciamento con le varie periferiche e dell'implementazione di un algoritmo di inseguimento che il veicolo H2Omni-X adotta per inseguire dalla superficie il tablet subacqueo, attraverso informazioni acustiche.

4.1.1 - ROS (Robot Operating System)

In questa sezione è descritto il sistema ROS in generale, sottolineando le sue potenzialità e la sua importanza nell'ambito della robotica.

4.1.1.1 - Generalità

ROS (Robot Operating System) è il sistema operativo su cui girano tutti i software di controllo del robot oggetto di studio. Se si immagina di dover svolgere un task molto complesso, possiamo idealmente andare a dividere quest'ultimo in una serie di sotto-task più semplici, addirittura elementari. Il concetto di ROS si basa proprio su questo: ognuna di queste azioni è eseguita da un programma autonomo che gira in contemporanea con tutti gli altri. Questi programmi in ROS vengono denominati "nodi" e si scambiano dei messaggi con una logica peer-to-peer, in assenza

quindi di un central routing. Da questo si capisce come il sistema sia di tipo tool-based: non vi è un ambiente di sviluppo integrato, ma ogni azione è eseguita da un programma separato. Ogni nodo può essere scritto in uno qualsiasi dei linguaggi di programmazione per cui esista un client library (C++, Python, Java, MATLAB, ecc.), e la comunicazione avviene in maniera del tutto indipendente dal linguaggio in cui i due nodi che scambiano messaggi sono scritti.

ROS è definito dai suoi stessi creatori come un “framework flessibile per la scrittura di software per robot”, una definizione sicuramente molto generica ma che coglie a pieno l’anima del software. È possibile vedere ROS come una raccolta di tool e librerie general purpose che ricoprono tutti i vari aspetti del software necessario al controllo di un robot, qualsiasi sia la tipologia di quest’ultimo. A questi tool separati è inoltre fornito un modo di comunicare estremamente semplice ed intuitivo. Questa semplificazione nei protocolli di comunicazione permette allo sviluppatore di risparmiare quel tempo che altrimenti sarebbe necessario per generare una struttura di comunicazione tra le varie componenti che possono trovarsi a lavorare contemporaneamente durante lo svolgimento di task altamente complessi.

Uno degli obiettivi di ROS è quello di “facilitare lo sviluppo tramite codice robusto e general purpose”. Le varie parti di codice scritte per una specifica applicazione possono infatti essere riusate per una diversa in modo estremamente semplificato. Un’altra delle caratteristiche chiave è quella di permettere un approccio allo sviluppo software di tipo collaborativo; le varie parti possono infatti essere sviluppate da persone o gruppi diversi e non è necessario che ognuno conosca a fondo il funzionamento di tutte quelle restanti. Ciò rende anche semplice integrare nuove funzioni o sostituire una parte del codice in sistemi già finiti e funzionanti.

Ciò che ha contribuito in maniera fondamentale alla diffusione di ROS è il fatto che sia rilasciato in maniera del tutto gratuita ed open source sotto una licenza BSD, che ne permette l’uso a scopo commerciale e non. Ovviamente ciò considera il pacchetto base di ROS, che consiste di tutti gli strumenti per gestire la comunicazione fra i nodi e tutta una serie di tool utili per le più svariate funzioni di un robot; ogni creatore di pacchetti potrà scegliere in modo indipendente se rilasciare o meno al pubblico quanto creato. La user-base di ROS è così ampia da permettere di trovare in rete quasi sempre una serie di pacchetti già pronti per le applicazioni che si desidera.

4.1.1.2 - Infrastruttura di comunicazione

Lo scambio di messaggi di cui si è parlato, caratteristica chiave di ROS, avviene tramite i topic; un topic avrà uno o più publisher che andranno appunto a pubblicare su di esso un messaggio, ed avrà poi uno o più subscriber che andranno a leggere i messaggi pubblicati; ogni qual volta un subscriber legge un nuovo messaggio dal topic, esso esegue un’azione specificata da una funzione di callback. Questo sistema è del tutto asincrono, poiché un messaggio può essere pubblicato anche in assenza di qualcuno che lo riceva, e anonimo, poiché chi legge un messaggio non ha informazioni su chi lo ha pubblicato e viceversa. I nodi, definiti nel paragrafo precedente, sono proprio i publisher e subscriber; ogni nodo in particolare può pubblicare su più topic e può sottoscrivere a più topic. Questa metodologia di comunicazione permette al sistema di essere damage tolerant, in quanto se anche un nodo nel sistema dovesse avere dei malfunzionamenti tutti gli altri continuerebbero a funzionare.

Una specifica libreria di ROS permette di eseguire le operazioni di recording e playback dei messaggi di uno specifico topic.

L’infrastruttura appena descritta può funzionare in maniera distribuita, cioè i nodi possono girare su piattaforme diverse connesse allo stesso network. Ciò è permesso dal roscore, cioè un programma che all’avvio di ogni nodo gli fornisce le informazioni sugli indirizzi nella rete degli altri nodi con cui comunicare. Questo agisce soltanto all’avvio dei nodi, poi il funzionamento è

puramente peer-to-peer come già descritto. In un ambiente con più piattaforme, su una sola dovrà essere in esecuzione un'istanza del roscore, mentre nelle altre si va a definire una variabile di ambiente che fornisce l'indirizzo della macchina dove reperirlo.

In parallelo al sistema publisher/subscriber è possibile utilizzare una serie di remote procedure calls. Queste sono dette *service*, ed hanno un funzionamento di tipo sincrono basato su una richiesta ed una successiva risposta. In genere si usano queste per azioni che devono essere eseguite una tantum. Esistono inoltre delle *actions*, che sono simili ai *service* ma permettono di monitorare il processo prima del suo completamento.

4.1.1.3 - *Namespace e remapping*

Ciò che rende il software scritto in ROS riusabile e general purpose sono i due concetti di namespace e remapping. Nodi identici possono esistere in più istanze e funzionare in contemporanea, ognuno sotto un diverso namespace. Ciò permette quindi di svolgere le funzioni di un particolare nodo per più componenti del robot o per più ambiti, avendo però scritto il codice una sola volta. Il remapping permette di sostituire una qualunque stringa all'interno di un nodo prima di eseguirlo. In questo modo si può andare a sostituire i valori specifici per il robot di interesse in un nodo scritto in maniera del tutto generale.

4.1.1.4 - *Feature specifiche per robot*

Su ogni topic vengono pubblicati messaggi con una formattazione ben precisa, specificata in fase di pubblicazione del messaggio stesso. Il pacchetto base di ROS fornisce una libreria di messaggi standard con le formattazioni utili a tramettere dati dei sensori, dati sulla geometria, informazioni di navigazione ed altro. Ciò non toglie che l'utente può definire i propri messaggi "customizzati" in base alle necessità.

Una libreria di ROS molto utilizzata è quella denominata "Tf". Questa si occupa di gestire le trasformazioni di coordinate tra i sistemi di riferimento di corpi e robot differenti in modo del tutto automatico, e funziona sia per trasformazioni statiche che dinamiche.

Al fine di simulare il comportamento dei robot è necessario fornirne una descrizione; ciò è reso immediato in ROS dall'utilizzo di un formato standardizzato, l'URDF, che consiste in un documento XML con informazioni su dimensioni, apparenza, inerzia, posizioni dei sensori e dei trasduttori, e tutte gli altri dati utili.

4.1.1.5 - *Tools*

ROS è utilizzabile totalmente da linea di comando. Ciò rende agevole l'accesso da remoto ad un robot, ma sono comunque messi a disposizione dell'utente dei tool con interfaccia grafica. Il più utilizzato di questi è Rviz, un software general-purpose per la visualizzazione 3D dei modelli dei robot e dei dati dei sensori.

Un altro tool utilizzato spesso è Rqt, definito come un generatore custom di interfaccia grafica. Questo permette di usufruire di tutta una serie di funzioni per monitorare i messaggi sui topic, per pubblicare messaggi, per richiamare *service*, per visualizzare dati e molto altro.

Nel pacchetto base è poi presente Gazebo, un tool di simulazione 3D con un completo motore fisico che permette di inserire i modelli dei robot in ambienti altamente complessi sia indoor che outdoor.

È possibile eseguire l'infrastruttura software ROS che gestisce il veicolo su un computer con sistema operativo Ubuntu, e la compagnia ha fornito una versione contenente dei nodi di simulazione numerica che sostituissero il robot fisico nella ricezione dei comandi e nella restituzione dei dati dei sensori; in questo modo è stato possibile lavorare anche non essendo in

presenza del robot fisico. Il grafico di questa infrastruttura prevede che i nodi siano rappresentati con forme ellittiche ed i topic con forme rettangolari. Un robot commerciale come quello considerato porta ad avere una struttura estremamente complessa, con decine di nodi operanti in parallelo ed uno scambio continuo di messaggi su svariati topic. Il punto di forza di ROS in questo caso è proprio che non è necessario conoscere il funzionamento completo di ognuno di questi nodi per andare ad inserire una nuova funzionalità o sostituirla con una esistente. È bastata infatti l'indicazione su quali nodi eseguissero la simulazione e sui topic contenenti i dati di interesse.

4.1.2 - Controllore ai potenziali virtuali

La fase di inseguimento del diver richiede che il veicolo H2Omni-X rimanga al di sotto di una certa distanza massima sulla superficie ma che allo stesso tempo non stazioni nella sua immediata vicinanza per evitare che possa fare da ostacolo al diver stesso nella fase di risalita. In fase preliminare, un altro veicolo H2Omni-X (1) equipaggiato dal ricevitore acustico (Seamodem) ed azionato mediante radiocomando, ha sostituito il diver per esigenze pratiche e di costi. In figura 4.1 è possibile trovare una rappresentazione di quanto descritto.

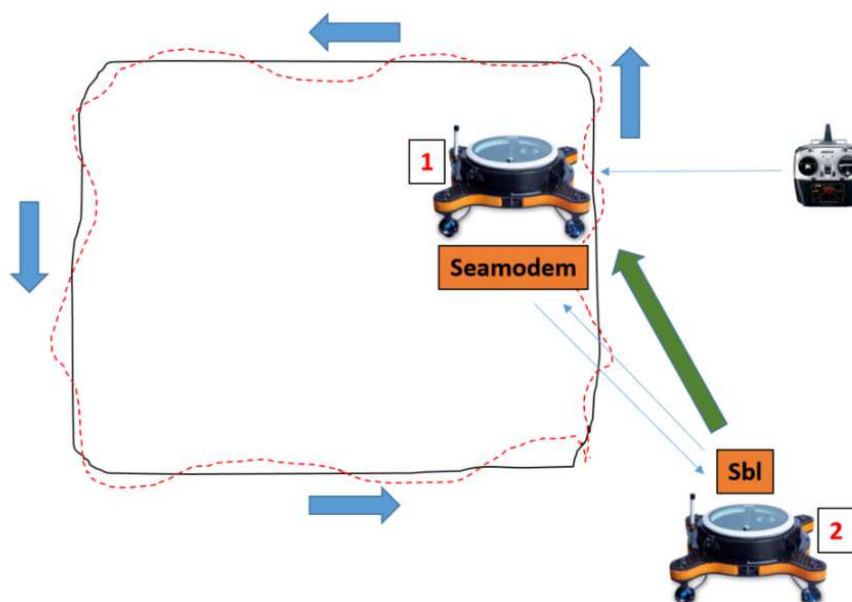


Figura 4.1 – Rappresentazione dell'architettura preliminare utilizzata per testare gli algoritmi di configurazione e di inseguimento.

È stato studiato il controllore ai potenziali virtuali che presuppone un controllo sulle velocità del veicolo inseguente, numero 2; in particolare queste velocità sono calcolate dall'inverso del gradiente di una funzione potenziale totale (funzione della posizione del diver) che è somma di due contributi: uno attrattivo ed uno repulsivo. Questo permette al veicolo 2 di essere attratto nell'area definita da una corona circolare intorno al veicolo 1. In altre parole, si calcola la velocità desiderata come l'inverso del gradiente della funzione nel punto rappresentato dalla posizione del veicolo. Ciò permette di avere un intervallo attorno al diver in cui la velocità desiderata è nulla, con un evidentemente rilassamento dei vincoli di controllo in quanto non si insegue un punto preciso ma tutta una corona circolare. Se il veicolo dovesse avvicinarsi o allontanarsi oltre il limite il gradiente tenderebbe a crescere in modo da far spostare il veicolo verso una nuova posizione. Definite con $[D_x D_y]$ le posizioni del diver, e con $[X Y]$ le posizioni del robot, valgono le seguenti equazioni:

$$r = \sqrt{(X - D_x)^2 + (Y - D_y)^2} \quad (4.1)$$

$$U_{att} = \begin{cases} 0 & \text{se } r \leq r_{max} \\ A(r - r_{max})^2 & \text{se } r > r_{max} \end{cases} \quad (4.2)$$

$$U_{rep} = \begin{cases} -R(r - r_{min})^4 & \text{se } r < r_{min} \\ 0 & \text{se } r \geq r_{min} \end{cases} \quad (4.3)$$

$$U = U_{att} + U_{rep} \quad (4.4)$$

$$v = -\nabla U \quad (4.5)$$

La funzione potenziale (figura 4.2 c) è ricavata come somma di due termini, uno attrattivo (figura 4.2 a) ed uno repulsivo (figura 4.2 b). Il primo è definito come una funzione parabolica che vale 0 in prossimità del diver e tende a crescere quando si supera un certo limite. Quella repulsiva invece ha una pendenza molto marcata in prossimità del diver e tende a zero oltre una certa distanza.

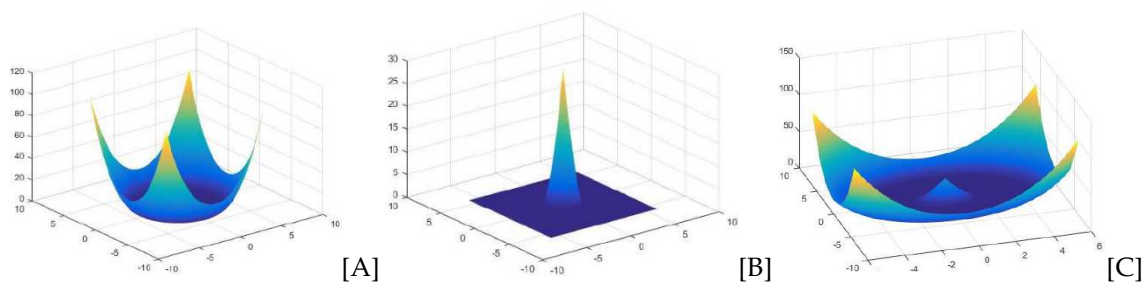


Figura 4.2 – (a) Funzione potenziale attrattiva, (b) funzione potenziale repulsiva, (c) funzione potenziale totale.

Dal grafico in figura 4.3, che rappresenta l'inverso del gradiente della funzione potenziale, possiamo vedere come la velocità desiderata sia nulla in una corona circolare con centro il diver e dimensioni r_{min} e r_{max} , che sarà proprio la regione in cui il veicolo rimarrà durante tutte le operazioni. Questi due valori devono essere adeguatamente calibrati in base alle necessità.

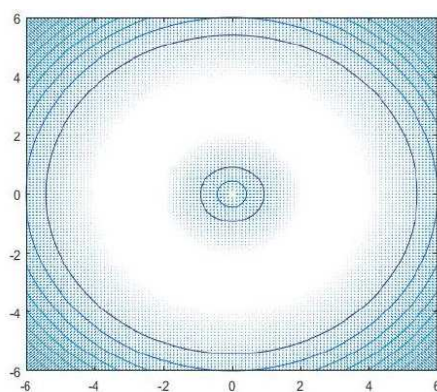


Figura 4.3 – Inverso del gradiente.

Altri due valori che definiscono il funzionamento sono i coefficienti A e R, rispettivamente relativi al potenziale attrattivo e repulsivo. Un valore alto genera velocità maggiori nelle zone di rispettiva competenza, quindi con una risposta più pronta; valori troppo alti generano però delle oscillazioni non desiderate nel moto del veicolo. Valori bassi invece implicano un controllo più

rilassato, ma si rischia di rimanere a distanze troppo grandi o troppo piccole. Anche questi dovranno quindi essere opportunamente calibrati in fase di utilizzo ed in base alle esigenze. Derivando la funzione potenziale attrattiva rispetto a X e rispetto a Y, si ottengono le relazioni (4.6) e (4.7) che sono state corrette con un segno negativo in fase di implementazione. Esse valgono nel caso in cui la distanza r del veicolo rispetto al target sia maggiore del raggio massimo.

$$V_{NORTH} = \frac{-2A(r - r_{max})}{r} (X - D_X) \quad (4.6)$$

$$V_{EAST} = \frac{-2A(r - r_{max})}{r} (D_Y - Y) \quad (4.7)$$

Se il veicolo staziona nella corona circolare con una distanza r rispetto al target compresa tra il raggio massimo ed il raggio minimo, valgono le relazioni (4.8) e (4.9).

$$V_{NORTH} = 0 \quad (4.8)$$

$$V_{EAST} = 0 \quad (4.9)$$

Derivando la funzione potenziale repulsiva rispetto ad X e rispetto ad Y, si ottengono le relazioni (4.10) e (4.11) che sono state corrette con un segno negativo in fase di implementazione. Esse valgono nel caso in cui la distanza r del veicolo rispetto al target sia minore del raggio minimo.

$$V_{NORTH} = \frac{-4R}{r} (r - r_{min})^3 (X - D_X) \quad (4.10)$$

$$V_{EAST} = \frac{-4R}{r} (r - r_{min})^3 (D_Y - Y) \quad (4.11)$$

Considerando il sistema solidale al veicolo in coordinate locali NED (north, east, down), con il nord rivolto in avanti, east a destra e down verso sotto, mediante la (4.12) è possibile risalire dalle velocità nel sistema di riferimento globale alle velocità nel sistema di riferimento locale, denominate "surge" e "sway", tenendo conto di un segno meno sull'angolo di heading.

$$\begin{bmatrix} V_{SURGE} \\ V_{SWAY} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} V_{NORTH} \\ V_{EAST} \end{bmatrix} \quad (4.12)$$

4.1.3 - Simulazione in ambiente Matlab del controllore ai potenziali virtuali

La strategia di controllo descritta è stata testata e studiata in ambiente Matlab. Sono state testate le matrici di rotazione che permettono un passaggio da coordinate locali in coordinate globali e viceversa, sia per quanto riguarda posizione ed orientamento, sia per quanto riguarda le velocità lungo nord e lungo est del veicolo. La simulazione impostata ha previsto di calcolare le velocità del veicolo in coordinate globali. Dalle simulazioni è stato possibile testare le equazioni che sono state successivamente implementate in linguaggio Python ai fini di integrarle all'interno del package sviluppato per gli inseguimenti, descritto nei paragrafi successivi. In figura 4.4 è rappresentato il piano tangente globale che riporta le posizioni del tablet (ricevitore), quelle del veicolo inseguente, la sua orientazione e le informazioni sulle circonferenze.

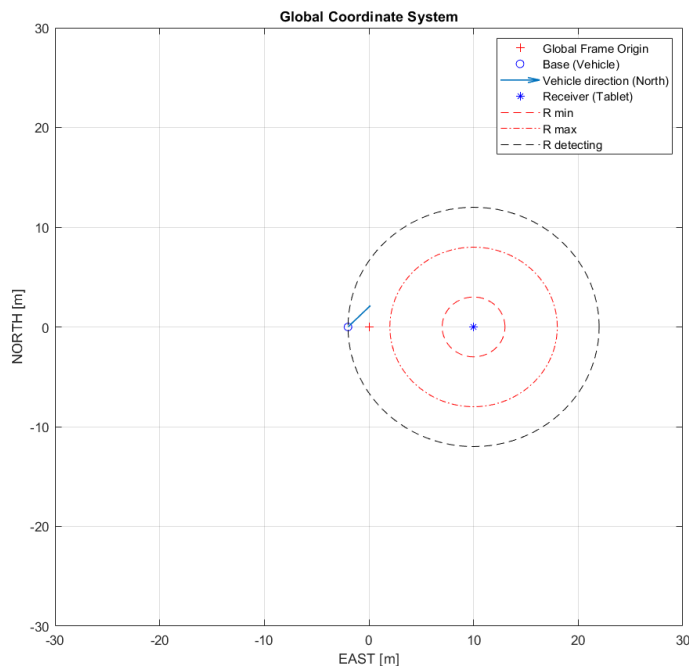


Figura 4.4 – Rappresentazione di una posizione ed orientazione generica del veicolo di superficie sul piano tangente globale.

Nel grafico di figura 4.5 è stata quindi plottata la velocità teorica che dovrebbe avere il veicolo per raggiungere il target nel punto (10 0); in questo caso è stato riportato solo il grafico della velocità lungo East a titolo di esempio, anche perché in riferimento all’esempio di figura 4.4 le velocità lungo North è chiaramente nulla. Si noti come una volta raggiunto il punto (2 0) le velocità tendono a decrescere man mano che ci si avvicina al punto (10 0).

Per ragioni pratiche, si è imposto un limite di saturazione oltre il quale le velocità del veicolo devono essere assunte pari alla velocità massima del veicolo stesso. Sotto tali limiti quindi ciò che si verifica è che le velocità seguono un andamento polinomiale come descritto nel paragrafo 4.1.2. Velocità negative rappresentano cambiamenti di verso.

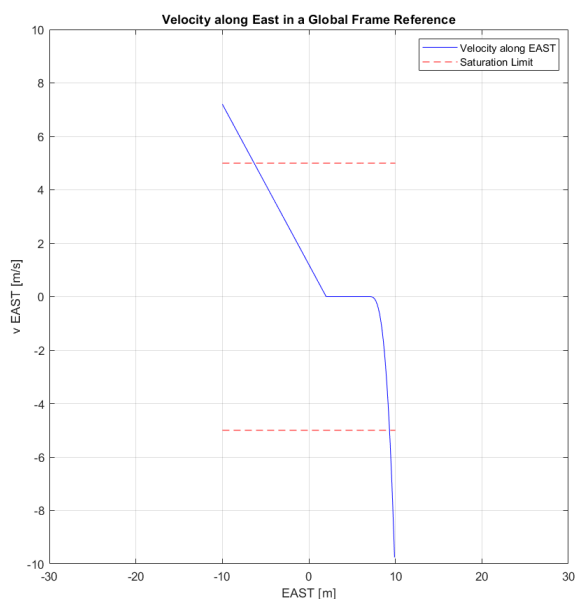


Figura 4.5 – Andamento della velocità del veicolo lungo east nel piano tangente globale.

4.1.4 - Descrizione del software in ROS implementato sull' ASV H2Omni-X

Il veicolo H2Omni-X ha un computer di bordo (Intel NUC) all'interno del quale gira il sistema operativo Ubuntu, quindi basato su LINUX. Come già accennato, la logica di funzionamento della platform è stata sviluppata sul framework ROS. Tutta la sensoristica pubblica su dei topic specifici i vari valori di GPS ed IMU.

In sintesi, è stato creato un package che permette in primis l'interfacciamento tra le varie periferiche. Inoltre, esso consente l'iscrizione ad alcuni topic specifici del veicolo per prelevare determinate informazioni, le quali vengono elaborate e vengono restituite al veicolo stesso in termini di velocità che esso deve esibire lungo le due direzioni del suo sistema di riferimento locale per una corretta applicazione della strategia di controllo automatico.

4.1.5 - Sviluppo del package di controllo

Sono stati sviluppati dei moduli software, in linguaggio Python, da integrare in ambiente ROS (Robot operating system) per l'interfacciamento hardware tra l'unità di localizzazione acustica (trasmettitore) ed il veicolo di superficie. Tali moduli riguardano il parsing dei dati di posizione ed orientazione ricevuti dal veicolo, l'interfacciamento con la base mediante protocollo TCP, e l'algoritmo ai potenziali virtuali che calcola in real-time le velocità da assegnare al veicolo che inseguire il target. La comunicazione tra tablet subacqueo e ricevitore acustico è gestito con protocollo UDP ed è controllabile in ambiente Linux solo nelle fasi di debug (in questo caso il ricevitore acustico viene attaccato alla rete locale mediante cavo ethernet); in tutti gli altri casi l'unico modo per interfacciarsi con il ricevitore acustico è quello di scambiare segnali acustici. In figura 4.6 è rappresentato uno schema.

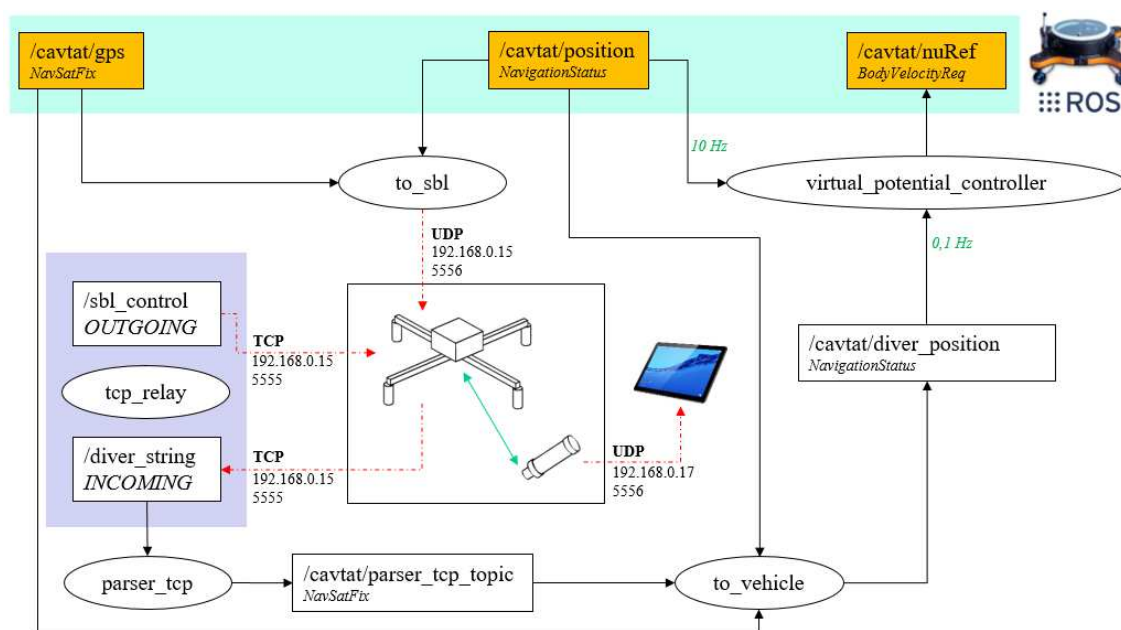


Figura 4.6 – Architettura del package ROS implementato.

In seguito è riportata una descrizione dei vari script implementati all'interno del package, sviluppati in linguaggio Html/Python. L'attività ha previsto anche una fase di sviluppo preliminare in simulazione attraverso il tool Gazebo, per meglio comprendere il significato di alcuni nodi e di alcuni topic presenti nella rete ROS.

4.1.5.1 - Launch file

Il launch file è utilizzato in ROS per avviare contemporaneamente più nodi o anche altri file launch. In particolare, le sezioni denominate "include" si riferiscono ad altri launch, mentre quelle denominate "node" si riferiscono a dei nodi. Nel caso studio condotto vengono avviati i nodi che permettono di reindirizzare i dati di sensori e trasduttori, ed il nodo relativo al controllore per l'inseguimento del diver. Inoltre, è avviato il nodo "tcp_relay" del pacchetto esterno "transport_bridges" utilizzato per il settaggio dell'indirizzo IP in rete locale e della porta locale e remota; questi parametri sono specificati all'interno del file launch stesso.

4.1.5.2 - to_sbl.py

In questo nodo, che contiene al suo interno una classe denominata ToSBL, i valori dei sensori GPS ed IMU del veicolo vengono formattati secondo lo standard NMEA e poi inviati su un socket UDP creato per la comunicazione con l'unità di superficie dell'SBL. Gli oggetti di classe in questo caso sono i subscriber ai topic relativi ad i due sensori, i dati su indirizzo IP e porte da utilizzare per creare il socket UDP ed il socket stesso creato tramite la libreria "socket". Nel callback per il sensore GPS i dati vengono convertiti in stringhe e poi viene utilizzata la libreria "pynmea2" per creare la stringa GPGGA. Questa viene poi mandata al socket UDP creato in precedenza. Per la IMU invece non era presente una funzione nella libreria, quindi la stringa viene creata manualmente come concatenazione delle stringhe contenenti i dati necessari. Anche questa viene in seguito inviata al socket UDP. Si ricordi che GPGGA si riferisce al GPS mentre PRDID si riferisce all'orientazione.

4.1.5.3 - Package per TCP

Si è deciso di utilizzare un package ROS per la creazione e la gestione delle interfacce TCP tra il mondo ROS ed il framework utilizzato dal sistema acustico. Con questo package è possibile creare delle porte TCP per la comunicazione. I parametri sono settati all'interno del launch file come precedentemente descritto.

4.1.5.4 - parser_tcp.py

Lo scopo di questo nodo, che contiene al suo interno una classe di nome ParserTCP, è di leggere i messaggi ricevuti dall'unità di superficie dell'SBL, formattati in stringhe csv (comma separated value e farne un parsing, cioè riconoscere le varie parti e creare dei messaggi formattati secondo una certa logica. Ciò avviene per i messaggi del GPS. Anche in questo caso si è creato una classe, denominata "Parser", con gli oggetti subscriber e publisher per i due sensori. Si noti come i nomi dei topic utilizzati siano generici, si è poi eseguito un remapping ai nomi reali in fase di avvio del nodo sul robot; è questo un esempio interessante di come il software scritto in ROS sia general-purpose. È stato utilizzato un nodo ROS già esistente che leggesse i valori pubblicati dall'SBL sull'interfaccia TCP e li pubblicasse sul topic che poi si usa in questo nodo. In entrambi i callback ogni elemento presente nella stringa viene inserito in un oggetto list tramite una funzione della libreria csv. Da questi "list" vengono poi letti i dati di interesse, le cui posizioni sono note avendo una formattazione NMEA. Con questi valori vengono creati nuovi messaggi di tipo NavSatFix per il GPS, che sono in seguito pubblicati.

4.1.5.5 - to_vehicle.py

In questo script è stata implementata una classe di nome ToVehicle, che acquisisce dati direttamente sia dal veicolo mediante sotto-iscrizioni dirette ai topic di H2Omni-X, sia a quelli creati nel package in questione. L'obiettivo è di fornire la posizione del diver nel sistema di riferimento globale (piano tangente) esprimendo le coordinate in formato NED (north, east,

down). A seguito di opportune conversioni, il nodo pubblica il risultato sul topic `diver_position` con un formato dei messaggi di tipo `NavigationStatus`.

4.1.5.6 - *virtual_potential_controller.py*

Questo nodo si occupa dell'implementazione del controllore ai potenziali virtuali la cui base teorica è stata descritta in precedenza. Nella pratica, viene creato un nodo che sottoscrive ai topic di posizione del veicolo e del diver, e ad ogni aggiornamento della posizione del diver viene calcolata la velocità obiettivo del veicolo che viene pubblicata sul relativo topic. Dei controlli a basso livello per i motori si occupano i controllori già esistenti implementati sul veicolo.

Per il corretto funzionamento dell'algoritmo, come visto nella sezione 4.1.2, occorrono le coordinate globali NED del diver e del veicolo. La classe `VirtualPotentialController` ha quindi due subscriber, uno per la posizione del diver ed uno per quella del veicolo, ed un publisher sul topic `"/cavtat/nuRef"`, cioè quello della velocità obiettivo. Vi sono poi le variabili relative alle due posizioni, che sono aggiornate con ogni callback dei due subscriber, e le costanti utilizzate nella formulazione matematica del controllore già descritta. Nel callback del subscriber alla posa del diver non si fa altro che aggiornare la variabile della classe con la posa attuale. In quello relativo alla posa del `H2Omni-X`, invece, vengono effettuati i calcoli sulle velocità obiettivo; questo avviene ovviamente solo se si è in una delle zone a gradiente non nullo. Viene poi creato un nuovo messaggio con le componenti di velocità di tipo `BodyVelocityReq` e l'informazione sul tempo attuale, che viene pubblicato tramite il publisher.

4.1.5.7 - *Servizi*

Il controllo originario del veicolo ASV prevedere l'utilizzo manuale di un joystick di comando, azionabile in automatico mediante un servizio. Si è pensato di creare un altro servizio al fine di consentire un facile passaggio dalla modalità di controllo manuale a quella automatica. In seguito è presente una lista dei messaggi da pubblicare sul service `/cavtat/ConfigureVelocityController` per selezionare la modalità di controllo del veicolo:

- [1 1 0 0 0 1] comando manuale con joystick;
- [2 2 0 0 0 0] comando automatico in velocità mediante controllore ai potenziali virtuali;
- [0 0 0 0 0 0] comando di STOP.

Si tenga presente che ciascuna posizione all'interno di questi vettori corrisponde ad un asse.

4.2 - Sviluppo della webapp di controllo

In questo paragrafo sono descritte le fasi dello sviluppo di una webapp di controllo per l'architettura modulare oggetto di studio. L'intervento software apportato in questa fase ha consentito di migliorare le connessioni tra i diversi dispositivi e di facilitare la gestione informatica dell'intero processo che altrimenti andrebbe condotto a "basso livello".

4.2.1 - Descrizione della problematica

Il sistema acustico di localizzazione subacqueo (sia il trasmettitore sia il ricevitore), impostati come Server dagli sviluppatori, non consente la connessione multiclient. Inoltre, dopo aver condotto diversi test sperimentali al mare o in alcuni laghi, è nata l'esigenza pratica di sviluppare

un sistema grafico facilmente utilizzabile per semplificare le operazioni di lettura, salvataggio ed interpretazione dei dati acquisiti durante le diverse prove.

L'hardware da dover gestire, si riduce essenzialmente a 5 elementi, ognuno dei quali si compone di una scheda dotata di sistema operativo proprio. In particolare, si dispone di:

- pc di bordo: computer portatile generico, dotato di sistema operativo Linux;
- veicolo ASV H2Omni-X: equipaggiato con PC Intel NUC;
- trasmettitore acustico: attaccato fisicamente sotto il veicolo autonomo, collegato alla stessa rete locale del pc, e dotato di BeagleBone con Linux;
- ricevitore acustico: dotato di BeagleBone, sulla quale gira sistema operativo Linux;
- tablet subacqueo: attaccato sul ricevitore acustico, e dotato di sistema operativo Linux.

Pc di bordo e veicolo ASV comunicano tramite rete locale wi-fi. Il trasmettitore acustico è collegato al veicolo mediante cavo di rete. Quindi tutti e 3 possono comunicare. Tablet subacqueo e ricevitore acustico stanno sott'acqua, quindi non possono connettersi via wi-fi alla rete locale generale, ma essendo molto vicini possono instaurare tra di loro una rete locale propria a parte. Il ricevitore acustico può comunicare con il trasmettitore acustico attraverso comunicazione acustica, quindi tra le due reti locali ci può essere uno scambio di informazioni. In fase preliminare di test, per ragioni di debug, il ricevitore subacqueo è stato connesso con un cavo ad una boa posta in superficie, dotata di antenna wi-fi. In questo modo il ricevitore acustico ed il tablet hanno avuto la possibilità di collegarsi alla rete locale principale. Dal pc di bordo, dunque, è stato possibile controllare tutto lo scambio dei dati. Il package sviluppato in ROS, precedentemente descritto, ha consentito la comunicazione dei dati da parte del sistema acustico verso il veicolo H2Omni-X, il quale è in grado di elaborare delle strategie di inseguimento.

Il passaggio successivo è stato quello di sviluppare una web-app di controllo per gestire tutti gli attori in gioco ai fini del monitoraggio dei parametri di interesse, dell'attuazione di alcuni comandi e della visualizzazione su una mappa dei dati di posizione. L'architettura software ha previsto una programmazione in linguaggio Html, Javascript e Css ai fini della creazione dell'interfaccia grafica, ma c'è stata la necessità di dover risolvere 4 problematiche:

- connessione multi-client;
- gestione dei web-socket per la comunicazione con la web-app;
- processamento dei dati, calcoli e conversioni;
- gestione della rete locale.

Questi punti sono stati gestiti da alcuni script sviluppati in linguaggio Python e Bash script di Linux, che fondamentalmente hanno il compito di fare da tramite tra l'hardware (ASV, SBL, Tablet e Ricevitore) e l'ambiente grafico della webapp di controllo.

4.2.2 - Presentazione della webapp di controllo

In questa sezione è presentata l'idea della webapp di controllo con una sua descrizione grafica e di funzionamento, quindi sia dal punto di vista del Front-end, sia del Back-end.

4.2.2.1 - Front-end

La fase di ideazione della webapp ha previsto un mockup la cui rappresentazione grafica è riportata in figura 4.7. Si è pensato di riportare tutte le informazioni utili in un'unica schermata per avere una visualizzazione completa di tutti i parametri.

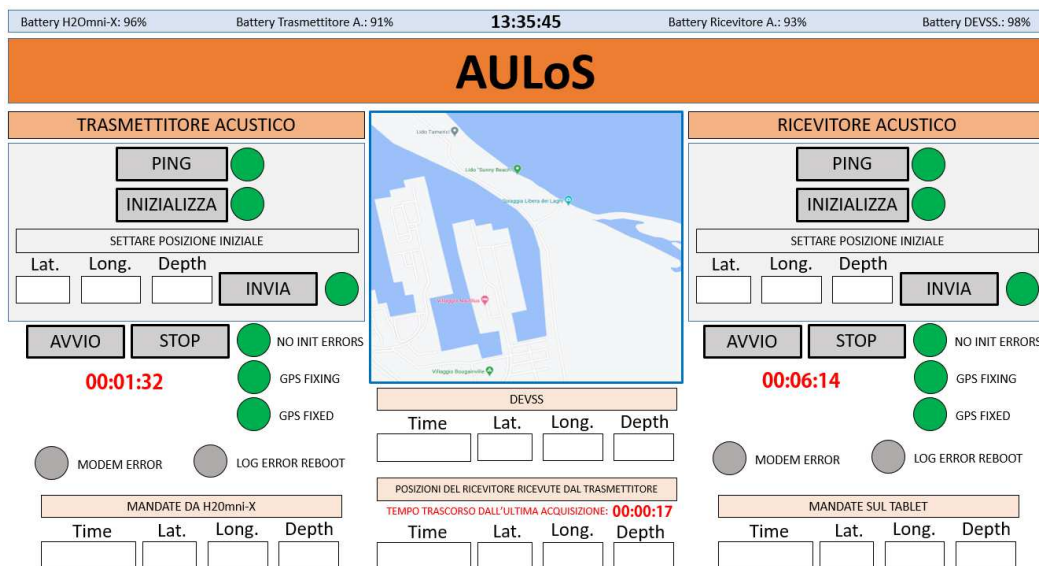


Figura 4.7 – Ideazione del mockup per l'interfaccia di controllo.

L'interfaccia è stata pensata con una struttura che prevede essenzialmente da 5 aree:

- barra superiore (sopra la scritta AULoS);
- colonna "Trasmettitore acustico";
- colonna "Ricevitore";
- colonna centrale superiore (dove risiede la mappa);
- colonna centrale inferiore.

Il risultato finale nella sua versione 1.0, sviluppato in linguaggio HTML e CSS, è riportato nella figura 4.8.

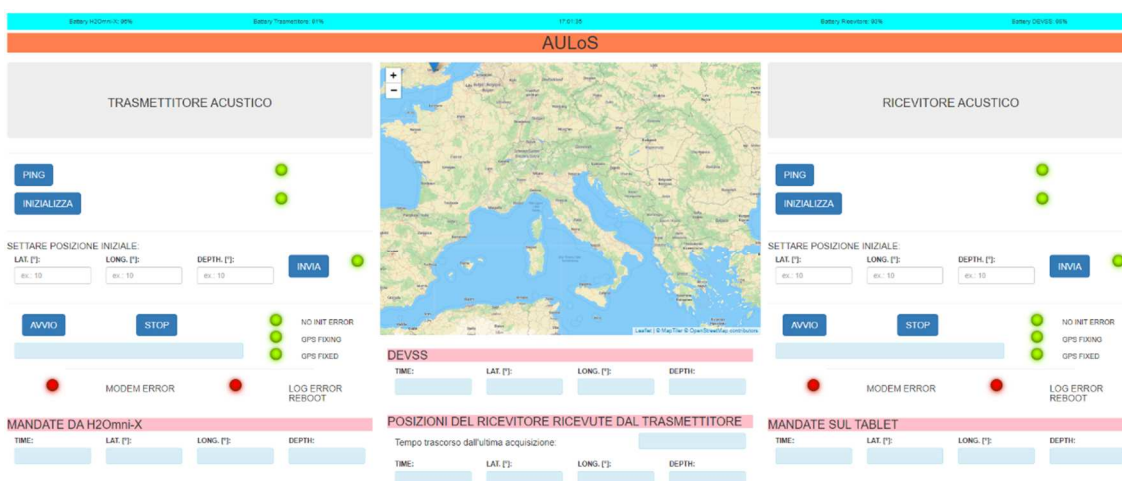


Figura 4.8 – Versione 1.0 dell'interfaccia di controllo.

Successivamente, dopo alcune fasi di test, è emersa l'esigenza di aggiungere delle funzionalità ulteriori. La versione 2.0 dell'interfaccia grafica, meno fedele al mockup iniziale, è presentata in figura 4.9. Oltre alla riga in basso riguardante la messaggistica dei websocket, sono state aggiunte delle funzionalità nella colonna centrale inferiore e la colonna del ricevitore acustico.

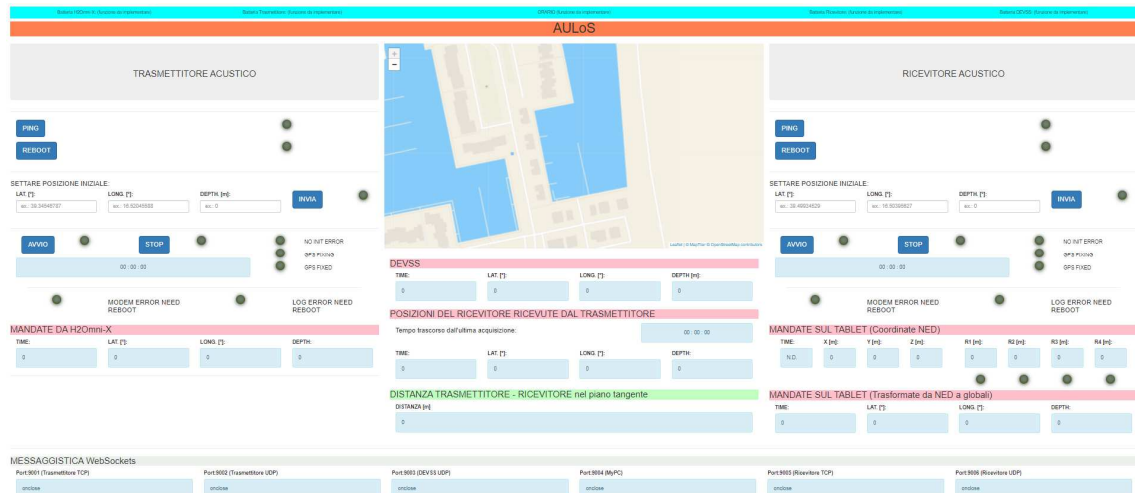


Figura 4.9 – Versione 2.0 dell'interfaccia di controllo.

Nella barra in alto, di colore azzurro, si trovano informazioni relative al livello di batteria del trasmettitore acustico, del ricevitore acustico, del veicolo DEVSS (introdotto in fase di debug e quindi non sempre presente) ed il veicolo H20mni-X. Nella parte centrale c'è un orologio digitale. Nella barra arancione è riportato semplicemente il nome AULoS.

Nella parte inferiore, identificata dalla barra grigia MESSAGGISTICA Websockets, si trovano delle celle per visualizzare la messaggistica di avvenuta apertura delle porte, con eventuali messaggi di errore.

Analizzando la parte sinistra della webapp, sotto la scritta TRASMETTITORE ACUSTICO, ci sono dei pulsanti che permettono di effettuare un Ping sulla rete locale del trasmettitore acustico, ed un Reboot dello stesso. Il tutto è correlato da Led per ottenere un feedback sul buon esito, o meno, dell'operazione. Scendendo sempre nella parte sinistra, si trova una sezione per settare le posizioni iniziali del trasmettitore acustico. Scendendo sempre di più, risiedono due pulsanti di AVVIO e di STOP del sistema, con un cronometro che tiene conto del tempo di durata del test al momento dell'avvio. Ci sono dei led per avere un feedback che non ci siano stati errori sul sistema acustico, che il GPS sia connesso e pronto per il fixing, ed un altro di avvenuto fixing da parte del GPS. Nella sezione sottostante ci sono altri due led di stato che permettono di monitorare un'eventuale necessità di dover effettuare un riavvio del modem acustico o eventualmente monitorare un errore di log. Nella sottosezione MANDATE DA H20mni-X è possibile monitorare le posizioni del trasmettitore acustico che sono quelle fornite dal veicolo H20mni-X. Nella parte di destra dell'interfaccia, sotto la scritta RICEVITORE ACUSTICO, si ritrovano gli stessi comandi visti per il trasmettitore, quindi Ping, Reboot, invio delle posizioni iniziali, avvio e stop del test con relativo cronometro, led vari per avere un feedback sulle operazioni condotte. Subito sotto, nella sottosezione MANDATE SUL TABLET (coordinate NED) si hanno le coordinate locali che il ricevitore acustico manda sul tablet. Nella sottosezione MANDATE SUL TABLET (Trasformate da NED a globali) si hanno le coordinate globali del ricevitore.

Nella parte centrale dell'interfaccia è presente la mappa di Leaflet gestita in questo progetto grazie al fatto che sono state fornite le API (Application Graphics Interface) da parte degli sviluppatori; ciò rende integrabile le mappe su applicazioni di questo genere. La parte della mappa è gestita sia dal file html sia da quello di Javascript, descritti in seguito; su di essa è possibile osservare, in tempo reale, dove sono posizionati il ricevitore acustico, il trasmettitore acustico, il veicolo DEVSS, la platform H20mni-X, ed il tablet subacqueo.

Nella sezione immediatamente sotto, si trovano le posizioni di DEVSS. Ancora più giù, ci sono le posizioni del ricevitore acustico ricevute dal trasmettitore acustico. In questa sezione è presente un cronometro, che si resetta ogni volta che il trasmettitore acquisisce una nuova posizione dal ricevitore mediante comunicazione acustica subacquea. Infine, nella sezione DISTANZA TRASMETTITORE – RICEVITORE NEL PIANO TANGENTE, si ha la distanza relativa tra il trasmettitore acustico ed il ricevitore acustico sul piano tangente al trasmettitore.

4.2.2.2 - Back-end

La schematizzazione in figura 4.10 rappresenta come sono state gestite le connessioni Client-Server tra i diversi dispositivi. Sono stati creati degli script in linguaggio Python all'interno dei quali sono implementate una serie di classi volte allo scopo. Spesso, un singolo componente deve essere in grado di comportarsi sia da client sia da server in base a dove esso è collegato. La descrizione dettagliata dei vari script sarà trattata nelle sezioni successive. Si evidenziano, in sintesi, i 5 macro-blocchi ciascuno corrispondente ad una unità hardware. In ciascun macro-blocco, su sistema operativo Linux, vengono fatti eseguire gli script. Si precisa, anche in questo caso, che il ricevitore acustico è connesso alla rete locale solo in fase di debug mediante cavo.

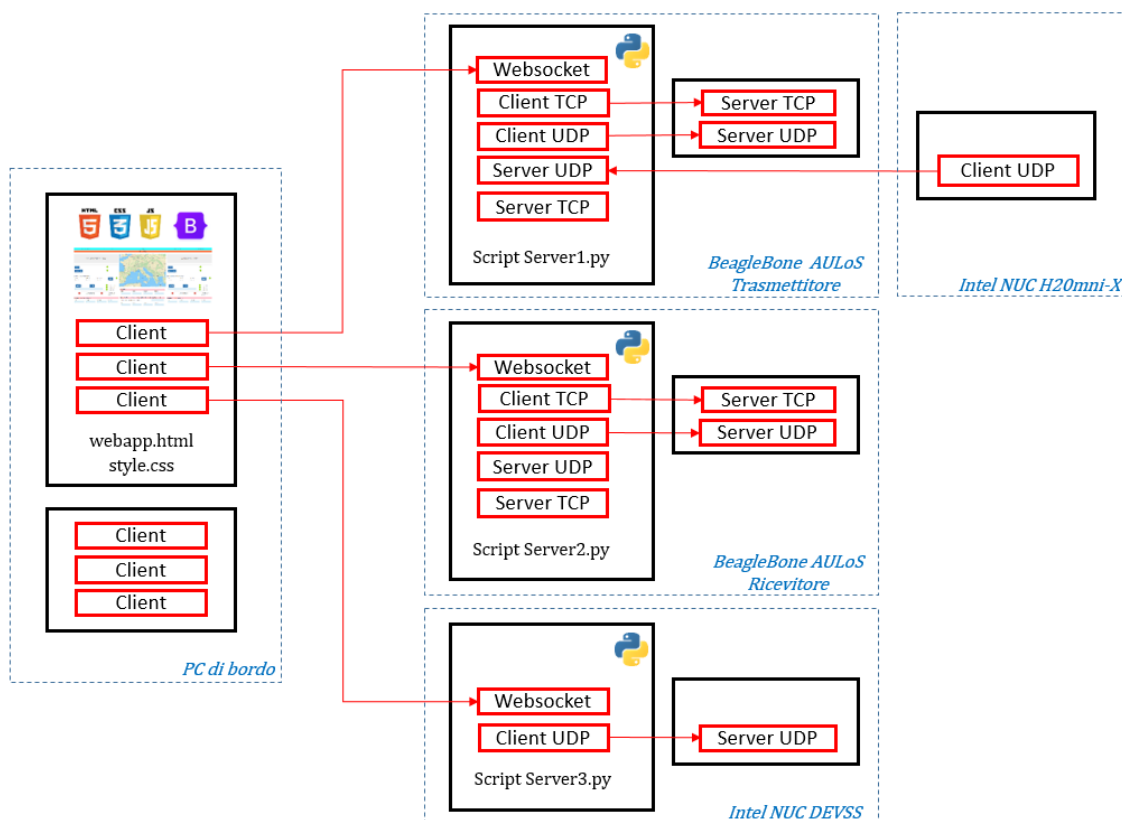


Figura 4.10 – Rappresentazione schematica della logica di back-end per le connessioni tra dispositivi.

4.2.3 - Gestione delle connessioni di rete

L'architettura prevede essenzialmente il seguente layout di connessione:

- il computer portatile per il controllo della webapp e del framework ROS è connesso ad uno switch mediante cavo. A questo switch è connessa la prima antenna MikroTik;
- la seconda antenna MikroTik è connessa allo switch dell'ASV, al quale switch è connesso anche il trasmettitore AULoS;
- la terza antenna, qualora ci si ritrovasse in fase di debug, è connessa al veicolo DEVSS o alla Boa, al cui switch è connesso il ricevitore acustico mediante cavo.

Le tre antenne wireless comunicano sulla stessa rete locale. In particolare, l'antenna alla quale è connesso il Pc portatile è stata settata come Ground control Station, mentre le altre due sono degli Access-point. In figura 4.11 è riportato uno schema delle connessioni, con gli indirizzi IP utilizzati.

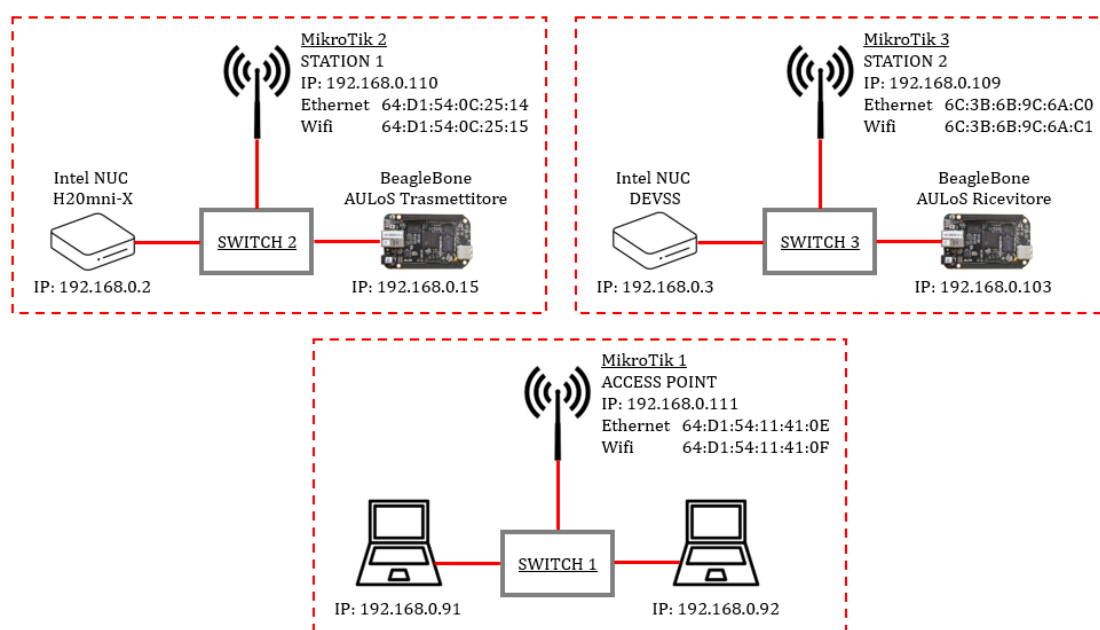


Figura 4.11 – Gestione delle connessioni di rete.

4.2.4 - Infrastruttura di comunicazione della webapp

In questa sezione è descritto come è stata impostata l'infrastruttura della comunicazione della webapp. Vediamo da qui in avanti un elenco delle porte che sono state create per gestire la comunicazione attraverso i websocket. È importante inoltre sottolineare che i dati scambiati in ingresso ed in uscita della webapp adoperano il formato JSON.

4.2.4.1 - Porta 9001 per il trasmettitore acustico e protocollo TCP

Lo script TCP_server.py gira sul trasmettitore acustico. In questa prima sezione è analizzato lo script centrale, il TCP_server.py che è essenzialmente composto da 3 classi. La classe principale, chiamata Server, consente di stabilire tutte le comunicazioni con il veicolo ASV ed i pc portatili che si connettono. Una seconda classe chiamata Client consente di stabilire una connessione con il trasmettitore acustico AULoS che per sua natura progettuale è un Server e non si può cambiare. Infine, una terza classe consente la connessione con la webapp mediante websocket. È chiaro che all'interno di questa micro-architettura tutti i dati si scambiano tra le classi.

In particolare è previsto da webapp (o dal terminale del pc) l'azionamento di alcuni comandi che passando dal TCP.server.py vanno a comandare il sistema AULoS. Esso periodicamente manda la posizione del ricevitore acustico, ottenuta mediante onde acustiche, verso il TCP_server.py che a sua volta le comunica al veicolo ASV, quindi al framework ROS, il quale attua le strategie di inseguimento. Tutte le modalità di scambio dati sono effettuate mediante protocollo TCP. In figura 4.12 è riportato uno schema per la porta 9001.

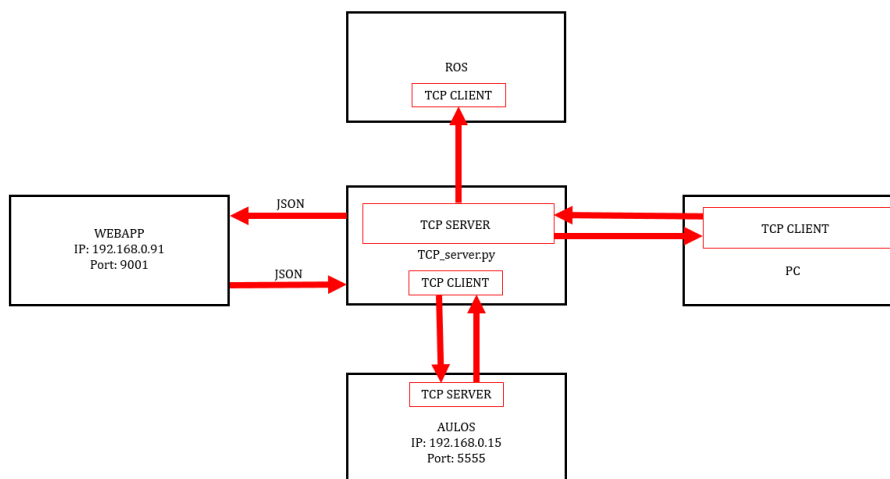


Figura 4.12 – Schema per lo scambio dei dati che interessano la porta 9001 mediante websocket.

4.2.4.2 - Porta 9002 per il trasmettitore acustico e protocollo UDP

In questa seconda sezione si analizza lo scambio dei dati mediante protocollo UDP. In particolare, l'esigenza è che il veicolo deve comunicare i dati del GPS e della IMU a tutta l'architettura. Grazie alla presenza dell'UDP server su un altro script UDP_server.py tutto ciò è concesso. L'unità centrale smisterà i dati al trasmettitore acustico ed alla webapp sulla porta 9002. Eventualmente qualsiasi computer connesso sulla rete locale potrà accedere a queste informazioni. Come nel caso precedente, lo script UDP_server.py ha una classe per gestire la connessione con il veicolo sotto forma di server, un'altra per il trasmettitore sotto forma di client, e poi attraverso i websocket per la comunicazione dei dati alla webapp sulla porta 9002. In figura 4.13 è riportato uno schema.

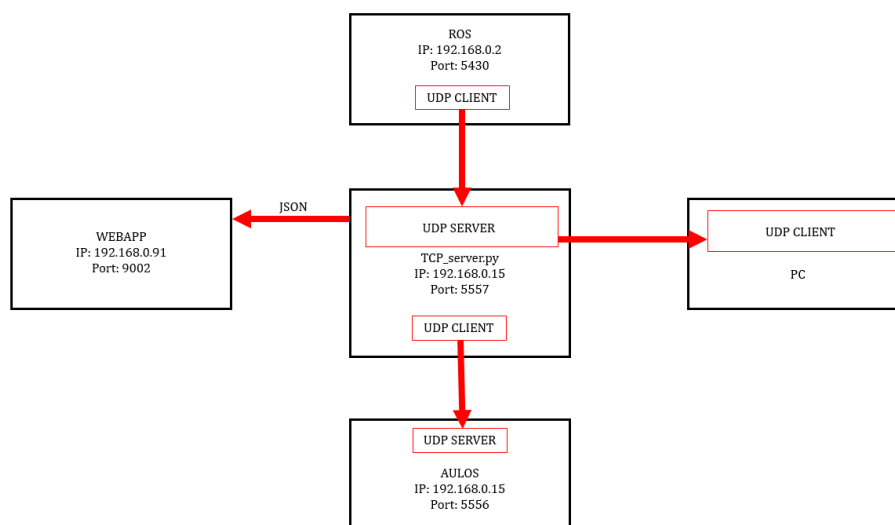


Figura 4.13 – Schema per lo scambio dei dati che interessano la porta 9002 mediante websocket.

4.2.4.3 - Porta 9003 per DEVSS (o boa) e protocollo UDP

In fase di debug del sistema, è essenziale che il ricevitore acustico comunichi in qualche modo la sua posizione non solo mediante onde acustiche al trasmettitore, ma anche mediante un cavo ethernet impermeabile connesso ad una boa oppure ad un secondo veicolo che nella situazione attuale è rappresentato da DEVSS, il veicolo ASV sviluppato da 3DResearch. Tale veicolo è connesso alla rete locale, pertanto è possibile avere il tempo reale il dato di posizione del ricevitore acustico senza aspettare che esso lo comunichi al trasmettitore con una certa frequenza. In questa sezione è analizzato un caso più semplice. Uno script UDP_server.py gira sul veicolo DEVSS il quale semplicemente trasferisce i dati disposizione da DEVSS alla webapp. In figura 4.14 è riportato uno schema del funzionamento.

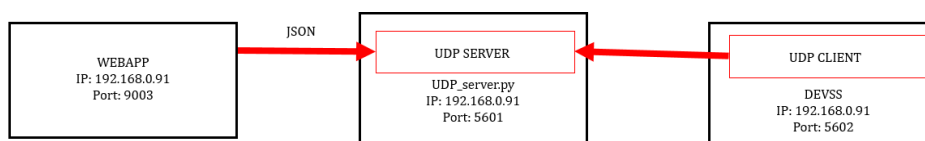


Figura 4.14 – Schema per lo scambio dei dati che interessano la porta 9003 mediante websocket.

4.2.4.4 - Porta 9004 per H2Omni-X locale

In questa sezione è analizzato un quarto script che gira sulla stessa scheda sulla quale gira la webapp, ovvero quella nel veicolo H2Omni-X. Questo script è stato creato per poter effettuare dei ping sulla rete locale, per assicurarsi che nulla sia disconnesso, e visualizzare le risposte sulla webapp di controllo mediante la porta 9004. In particolare, i ping avvengono verso il veicolo H2Omni-X, verso DEVSS (o la boa), verso il trasmettitore acustico e verso il ricevitore acustico in fase di debug. Tutte le risposte sono veicolate sulla webapp. Inoltre è possibile mandare dei messaggi per effettuare un riavvio del trasmettitore acustico e del ricevitore acustico. Tutti gli script, ad eccezione di webapp.py hanno estensione .sh ovvero sono degli script programmati in linguaggio Bash di Linux. In figura 4.15 è riportata una rappresentazione schematica.

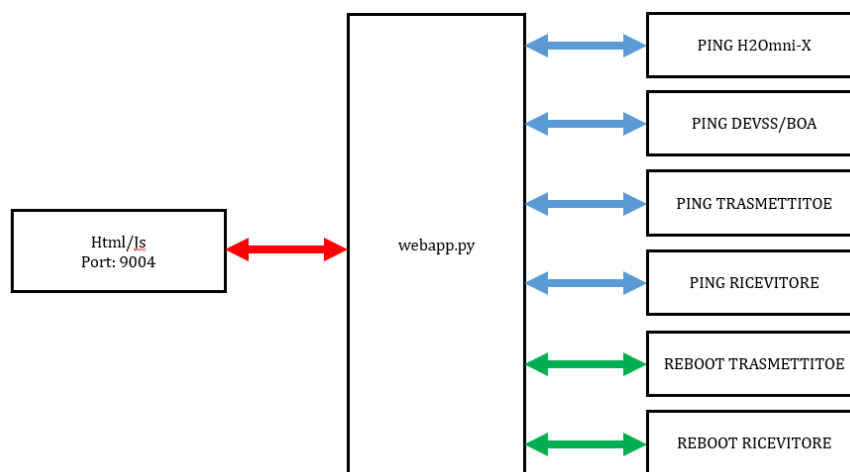


Figura 4.15 – Schema per lo scambio dei dati che interessano la porta 9004 mediante websocket.

4.2.4.5 - Porta 9005 per il ricevitore acustico e protocollo TCP

In questa quinta sezione si analizza come avviene la comunicazione con il ricevitore acustico, ammesso che sia connesso alla rete locale in fase di debug. Il principio di funzionamento è lo stesso del trasmettitore ma ci sono meno unità coinvolte. Infatti i comandi sono inviati dal Pc

oppure dalla webapp e sono veicolati sul ricevitore acustico attraverso il file TCP_server.py che gira sul ricevitore acustico. In figura 4.16 è riportato uno schema.

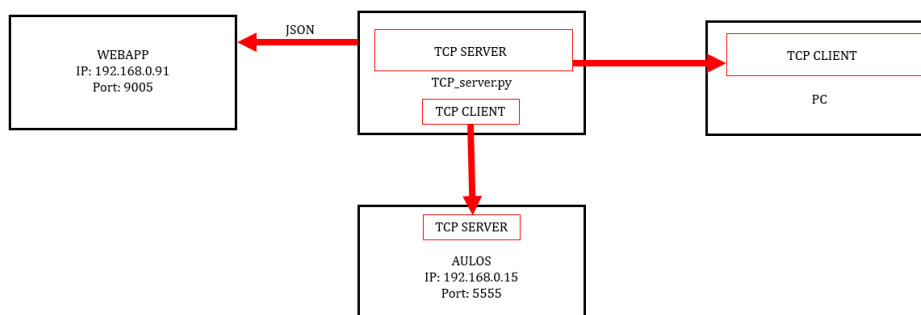


Figura 4.16 – Schema per lo scambio dei dati che interessano la porta 9005 mediante websocket.

4.2.4.6 - Porta 9006 per il ricevitore acustico e protocollo UDP

In questa parte, il ricevitore acustico comunica con alta frequenza la sua informazione di posizione la quale viene veicolata attraverso lo script UDP_server.py verso il tablet subacqueo, verso la webapp mediante web-socket attraverso la porta 9006, e verso un eventuale PC connesso in rete locale attraverso terminale. In figura 4.17 è riportato uno schema.

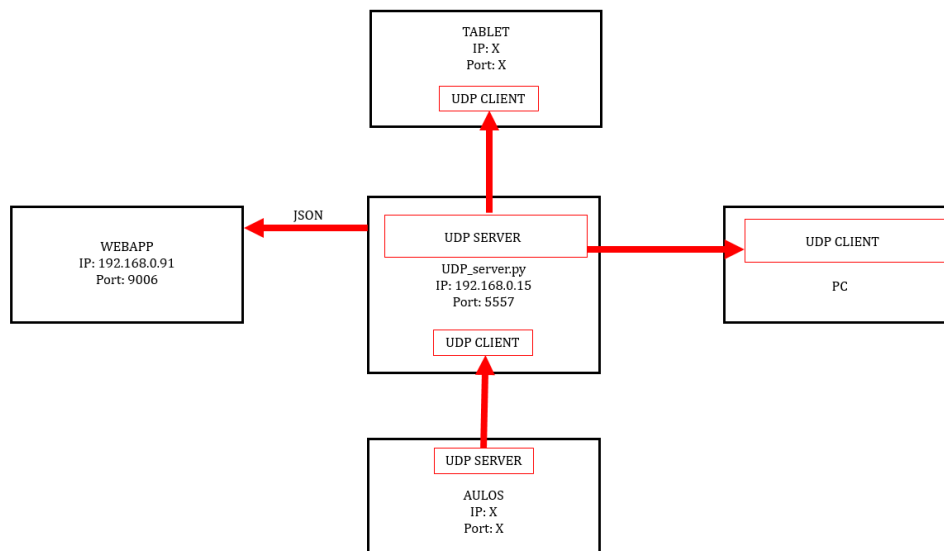


Figura 4.17 – Schema per lo scambio dei dati che interessano la porta 9006 mediante websocket.

4.2.5 - Files della webapp

In questa sezione sono brevemente commentati gli script realizzati per il corretto funzionamento della webapp di controllo.

4.2.5.1 - Veicolo H2Omni-X

Si iniziano a descrivere i file implementati all'interno del computer di bordo del veicolo ASV. Sul veicolo gira un sistema operativo Linux al quale è connesso il framework ROS, quindi con tutta la sua logica descritta nei paragrafi precedenti. Si sono quindi successivamente implementati i file relativi all'interfacciamento con la webapp. In particolare si hanno:

- ***Index.html***: in questo file html è implementata tutta la struttura dell'interfaccia grafica della webapp. Tale file si avvale dell'uso di Bootstrap, un framework Css per la realizzazione dei pulsanti, delle finestre di monitoraggio, ecc. In questo file è richiamata anche la mappa Leaflet [4.01] il cui funzionamento sarà descritto nella parte relativa alla dinamica della pagina, quindi nel file Javascript, ma una parte delle linee di codice sono state implementati in questo file;
- ***style.css***: in questo file di stile Css abbastanza ridotto, sono implementate alcune definizioni per i led presenti all'interno della webapp;
- ***script.js***: il file programmato in Javascript consente di dare dinamicità alla pagina web. In esso sono contenute tutta una serie di informazioni per la mappa, come ad esempio la creazione dei punti iniziali dei quattro attori protagonisti (ASV, DEVSS, tablet e sub), nonché il richiamo delle varie icone. Inoltre, sono definite tutte le funzioni per lo scambio della messaggistica tramite il formato JSON. Sono inoltre presenti alcune funzioni per la creazione dei websocket. Sono definite le funzioni dei timer di avvio delle simulazioni, nonché indicazioni temporali dell'ultimo dato ricevuto da parte del ricevitore acustico. Vi sono inoltre funzioni per la generazione dell'orario, e poi delle funzioni per la conversione del dato di posizione locale in globale del ricevitore acustico, visualizzabili sulla mappa in coordinate globali dopo aver fornito le coordinate iniziali del punto. Questo file è composto dalle seguenti macro-sezioni:
 - SEZIONE 1: sono definite tutte le variabili. In particolare, vengono definite le variabili per la creazione dei websocket, e poi le variabili per inizializzare latitudine e longitudine del trasmettitore acustico, del ricevitore acustico mandati dal trasmettitore, del ricevitore acustico mandate sul tablet, di DEVSS, variabili per la comunicazione con comandi LINUX su PC. Sono inoltre definite le varie icone che sono visualizzabili sulla mappa;
 - SEZIONE 2: qui avviene la creazione dei websocket utilizzati per la trasmissione dei vari dati dell'architettura con le webapp di controllo. In particolare, si ricorda che ci sono 6 websocket. Sono inoltre definite una serie di funzioni per l'apertura, la chiusura ed il monitoraggio di errori per quanto riguarda i websocket;
 - SEZIONE 3: la sezione 3 riguarda la parte di software per gestire il primo websocket, quello con la porta 9001. Gestisce la ricezione dei messaggi mediante protocollo TCP. Gestisce inoltre il comportamento dei led presenti nella parte sinistra della webapp;
 - SEZIONE 4: questa sezione si occupa di gestire la porta 9002 attraverso la quale avviene la trasmissione dati mediante protocollo UDP per il trasmettitore acustico;
 - SEZIONE 5: questa sezione si occupa di gestire la porta 9003. Si ricevono i messaggi da parte del veicolo DEVSS;
 - SEZIONE 6: questa sezione si occupa di gestire la porta 9004 attraverso la quale è possibile azionare comandi di Ping e Reboot sia del ricevitore acustico sia del trasmettitore acustico;
 - SEZIONE 7: questa sezione di occupa di gestire la porta 9005 attraverso la quale è possibile coordinare la parte di comunicazione del ricevitore acustico mediante protocollo TCP. Sono gestiti i led relativi allo start, all'invio delle coordinate iniziali, e quelli di errore;
 - SEZIONE 8: questa sezione si occupa di gestire la porta 9006 attraverso la quale è possibile coordinare la parte di comunicazione UDP del ricevitore acustico. È

possibile avere delle informazioni relative ai 4 ranges acustici, e quelli relativi alle posizioni X, Y, Z del ricevitore;

- SEZIONE 9: si utilizzano le funzioni per interagire con la pagina HTML;
- SEZIONE 10: in questa sezione ci sono una serie di funzioni per richiamare i comandi di Ping e di Reboot del trasmettitore acustico e del ricevitore acustico. Si hanno inoltre avvio e stop dei cronometri per il trasmettitore e per il ricevitore. Infine le ultime due funzioni permettono di inviare i dati di posizione iniziale del trasmettitore e del ricevitore;
- SEZIONE 11: in questa sezione vengono gestite le varie funzioni per i cronometri di ricevitore e trasmettitore. Inoltre, vi è un cronometro che si resetta ogni volta che il trasmettitore riceve un nuovo dato di posizione da parte del ricevitore acustico;
- SEZIONE 12: in quest'ultima sezione viene calcolata la distanza che c'è tra il trasmettitore ed il ricevitore, per una esigenza di debug per i test sperimentali. La funzione ottiene la latitudine e la longitudine sia del ricevitore sia del trasmettitore, mediante un modello di approssimazione della terra con una sfera;
- *webapp.py*: questo script permette l'interazione tra webapp e gli script bash di Linux ai fini del Reboot dei dispositivi acustici e dei Ping di rete. Viene stanziata la porta 9004 per il websocket, sull'indirizzo di rete 127.0.0.1 che è quello locale della macchina su cui verrà fatto girare l'eseguibile. In esso sono implementati delle funzioni che danno dinamicità ai led della webapp. In particolare, se il ping di rete va a buon fine, i led lampeggeranno per tre volte consecutive. Nel caso di reboot avvenuto correttamente, i led corrispondenti si accenderanno di verde, altrimenti di rosso;
- *file Bash Linux*: diversi file bash di Linux permettono l'interazione a livello di sistema operativo con i componenti della rete locale. In particolare, nella cartella "FileBash" sono presenti i seguenti file:
 - *pingRIC.sh*: è utilizzato per fare un ping a livello di rete locale al ricevitore acustico;
 - *pingTR.sh*: è utilizzato per fare un ping a livello di rete locale al ricevitore acustico;
 - *rebootRIC.sh*: è utilizzato per effettuare un riavvio del ricevitore acustico;
 - *rebootTR.sh*: è utilizzato per effettuare un riavvio del trasmettitore acustico;
- *files ausiliari*: tra i file ausiliari ci sono gli assets di bootstrap, la libreria per il funzionamento dei websocket e le immagini. Nella cartella "immagini" sono presenti 4 file .png (figura 4.18) utilizzati nella webapp per identificare graficamente sulla mappa le posizioni in real-time rispettivamente di:
 - coordinate globali di H2Omni-X;
 - coordinate globali del ricevitore acustico mandati sul trasmettitore attraverso comunicazione acustica (bassa frequenza di dati); in questo modo si fa un debug dell'avvenuta comunicazione della posizione del ricevitore al trasmettitore mediante scambio di messaggi acustici;
 - coordinate globali che il tablet subacqueo riceve ad alta frequenza dal ricevitore acustico. Sono gli stessi dati che il sub monitora in real-time sulla mappa del tablet;
 - coordinate globali di DEVSS (o eventualmente di una boa) con le sole esclusive finalità di debug.

Nella cartella "assets" sono invece presenti alcuni file css e javascript di bootstrap, ed i file css e javascript della mappa Leaflet. Nella cartella "websocket_server" sono presenti le librerie per la creazione e l'interazione con i websocket.

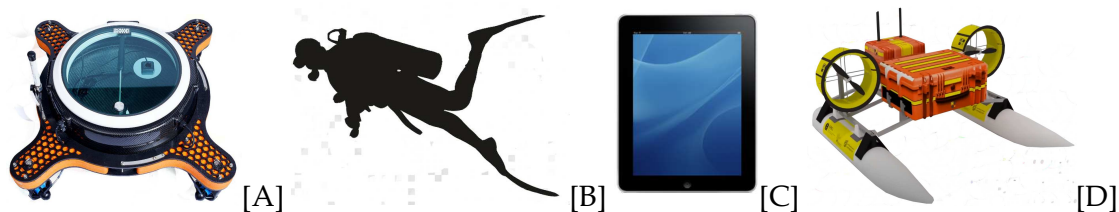


Figura 4.18 – Immagini utilizzate all’interno della webapp di controllo per identificare le coordinate globali rispettivamente di: (a) H2Omni-X, (b) ricevitore acustico con informazioni provenienti da H2Omni-X, (c) ricevitore acustico e quindi tablet subacqueo, (d) posizione della boa o del veicolo DEVSS.

Tutti i files che sono stati descritti in questo paragrafo sono collocati all’interno della cartella `usr/share/nginx/html/aulosserver`. Nginx è un webserver leggero e ad alte prestazioni, precedentemente installato dagli sviluppatori di H2Omni-X.

4.2.5.2 - Trasmettitore acustico

Nel trasmettitore acustico sono implementati dei servizi che consentono di far eseguire in modo automatico due script Python per comunicazione TCP ed UDP nel momento dell’accensione del sistema. Affinchè ciò sia possibile, i servizi si avvalgono di due script `.sh` ai quali è delegato il compito di richiamare gli script Python. In generale, questi script sono:

- `startTCP.sh`
- `startUDP.sh`
- `aulos-server-tcp.service`
- `aulos-server-udp.service`

I file `.service` sono collocati nella cartella `/etc/systemd/system` di Ubuntu Linux, mentre i file `.sh` sono collocati nella cartella della webapp. Inoltre, sono presenti altri due script molto importanti:

- `TCPythonServer.py` per trasmettitore acustico: questo script programmato in Python racchiude una serie di classi che permettono di instaurare delle comunicazioni client-server tra platform e sistema acustico. Inoltre è gestita la connessione con la webapp mediante l’utilizzo di `websocket`. Questo script è composto dalle seguenti classi:
 - Class `Server`;
 - Class `AulosClient`;
 - Class `WebSocket`.

La classe `Server` essenzialmente permette di inizializzare il server principale che permette la connessione multi-client con altri dispositivi. La classe `AulosClient` è invece una parte di codice che si comporta come client per il sistema trasmettitore AULoS. Il trasmettitore AULoS è un server e non si poteva trasformare come client, altrimenti si sarebbe semplificato tutto. La classe `WebSocket` invece permette di instaurare una comunicazione con la webapp di controllo del sistema. Nella classe `WebSocket` confluiscono tutti i messaggi dalla webapp verso i dispositivi connessi e viceversa, mediante delle funzioni implementate nella classe. I messaggi vengono parserizzati, ed avvengono le conversioni in JSON. Anche in questo caso, è essenziale la libreria `websocket_server` per il funzionamento dei `websocket`, utile anche per lo script `UDPPythonServer.py`.

- ***UDPPythonServer.py*** per trasmettitore acustico: la struttura generale di questo script segue quello descritto precedentemente. Anche qui abbiamo essenzialmente 3 classi che svolgono funzioni analoghe a quelle già descritte:
 - Class Server;
 - Class ClientForAulos;
 - Class WebSocket.

4.2.5.3 - Ricevitore acustico

Senza perdere di generalità neanche in questa circostanza, ci si limita ad indicare semplicemente quali sono gli script programmati per il ricevitore acustico

- ***TCP_Python_Server.py*** per ricevitore acustico;
- ***UDP_Python_Server.py*** per ricevitore acustico.

Valgono le stesse considerazioni in merito a quanto detto per il trasmettitore acustico. In fase di debug è sempre possibile connettersi alla webapp pertanto anche in questo caso sono implementate le classi relative alla connessione tramite websocket. Rimane inoltre da precisare che questi script sono attivati manualmente da terminale shell, pertanto non sono previsti script per l'avvio automatico al momento dell'accensione del dispositivo.

4.2.5.4 - DEVSS (o boa) per debug

Il file descritto in questa sezione è utile solo ai fini di debug. Si presuppone che nella rete locale ci sia la presenza di un secondo veicolo come DEVSS (oppure di una boa mobile in superficie trainata da un secondo diver). Lo script è il seguente:

- ***UDP_DEVSS_Python_Server.py***: l'unico file utilizzato su DEVSS permette uno scambio di dati mediante protocollo UDP. Tali dati sono i valori di posizione prelevati dal GPS del veicolo (o della boa) e che sono trasmessi alla webapp di controllo per la visualizzazione sulla mappa. L'utilità è quella di avere un confronto tra la posizione calcolata dal ricevitore acustico (agganciato a DEVSS o alla boa) ed il dato GPS prelevato appunto da DEVSS o dalla boa.

4.2.5.5 - Pc di bordo

Sul pc di bordo non è richiesto l'avvio di nessuno script. Inizialmente il pc di bordo gestiva l'esecuzione della webapp quindi tutti i files giravano su di esso. Nella configurazione finale, per visualizzare la webapp, basta essere connessi alla stessa rete locale del computer di bordo del veicolo, ed inserendo l'indirizzo IP nel browser è possibile visualizzare la webapp. Inoltre, è possibile interagire con le varie periferiche connesse alla rete locale mediante le shell di Linux.

4.3 - Test sperimentali e protocolli

In questa sezione sono descritti una serie di test sperimentali condotti, ed i relativi protocolli. Si sono effettuate delle prove sperimentali in diverse giornate, ciascuna delle quali ha permesso di validare sia il corretto funzionamento dei codici implementati sia lato ROS sia per la webapp di

controllo, ma anche per valutare le performance dell'intero sistema, in particolar modo del sistema di localizzazione acustico SBL AULoS.

Sono state condotte inizialmente delle prove presso il lago Jarun di Zagabria per validare il corretto funzionamento del sistema e dell'algoritmo di inseguimento. Sul veicolo ASV è stata installata la base del sistema di localizzazione acustica SBL nella sua prima versione. In particolare sono state previste due modalità di esecuzione:

- i principali test effettuati hanno riguardato il caso in cui il ricevitore acustico stesse fisso. Dal momento che viene reso attivo il controllore ai potenziali virtuali sul veicolo H2Omni-X, quest'ultimo si posiziona in una corona circolare con raggio medio di circa 5 metri intorno al ricevitore;
- una seconda tipologia di test (figura 4.19) ha previsto invece che il ricevitore venisse montato su un ROV, quello celestino, comandato da un joystick. È stata prevista la modalità automatica del veicolo H2Omni-X che mediante informazioni acustiche scambiate tra trasmettitore (montato sotto di esso) e ricevitore (montato sul ROV) è stato in grado di adattarsi ai cambiamenti delle distanze relative.

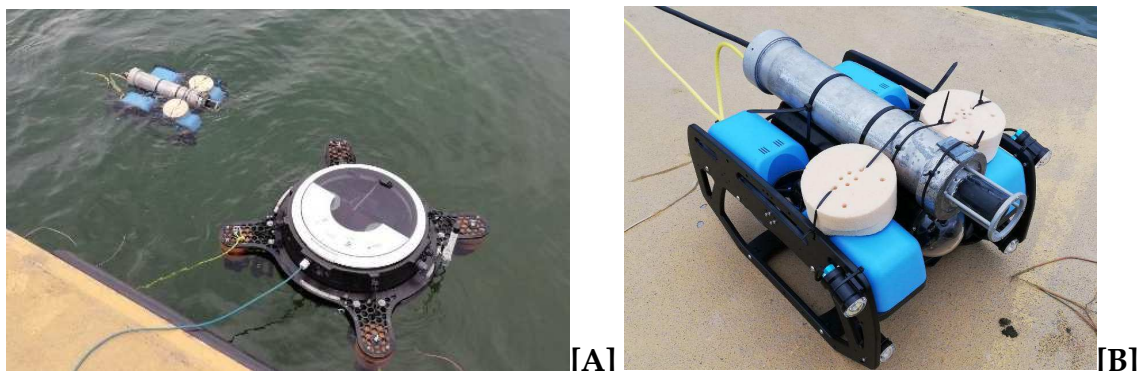


Figura 4.19 – Test sperimentali presso il lago Jarun di Zagabria. (a) ROV connesso al ricevitore acustico e veicolo H2Omni-X connesso al trasmettitore acustico; (b) ROV e ricevitore acustico.

Questi test svolti a Zagabria hanno interessato il sistema acustico SBL nella sua prima versione. Sono serviti principalmente a testare il software implementato e l'intera architettura in un contesto diverso da quello della piscina; è stato possibile fare debug con scambio dati a maggiore frequenza, condizione che la piscina per fenomeni legati al riverbero acustico non permette.


Nel corso dei mesi successivi sono stati effettuati altri test in Italia che hanno interessato zone come il porto di Campora, il porto di Vibo Valentia, ed i laghi di Sibari. Il software generale per la comunicazione (ROS) è stato arricchito di funzionalità e quindi i test sono serviti per testare di volta in volta nuove situazioni, anche in presenza della nuova versione del ricevitore acustico AULoS. Parallelamente è stata sviluppata la webapp di controllo che ha richiesto dei test in aree ampie per avere maggiori spazi di manovra, e soprattutto allo scoperto, per avere un buon segnale GPS.

La strumentazione interessata è sempre stata costituita da due veicoli ASV, uno dei quali è H2Omni-X, e dal sistema acustico AULoS (trasmettitore e ricevitore). Il trasmettitore acustico è sempre rimasto solidale ad H2Omni-X, mentre il ricevitore a volte è stato ancorato al secondo ASV, oppure si è utilizzato da solo.

Le tipologie di test effettuate potrebbero essere sintetizzate schematicamente nel modo seguente, facendo riferimento ai protocolli utilizzati:

• TEST 1:

Descrizione: entrambi i veicoli sono ancorati con delle corde da qualche parte, ad una distanza di circa 10 metri. Prendiamo quindi la posizione assoluta del ricevitore.
Durata: 5 minuti



LUOGO:
 Acqua, luogo aperto, area estesa, profondità adeguata

N.B. 1 : le prove devono essere condotte a diverse distanze, come ad esempio a 10[m], a 25[m] ed a 50[m]

N.B. 2 : conviene fissare un'origine alla base acustica, altrimenti l'impatto dell'errore del gps sul risultato finale si farebbe sentire troppo, e noi vogliamo solo capire quanto è l'errore del sistema acustico. Analogamente, dal lato ricevitore bisogna assicurarsi che il filtro di Kalman non vada a fare previsioni dato che è fermo (provare con richiesta posizioni ogni 5 secondi ed ogni 30 secondi ad esempio).

Nome file ROS	
Distanza effettiva	
Ora di inizio	
Ora di fine	

HARDWARE NECESSARIO:

- Ricevitore AULoS (**fixed**)
- Platform Zagabria con SBL
- ATLAS

PROTOCOLLO:
 La platform è fissata;
 AULoS è fissato;
 L'ATLAS è solidale con AULoS;
 Si calcola la posizione assoluta di AULoS tramite l'SBL e si confronta con la stima dell'ATLAS.

OUTPUT:

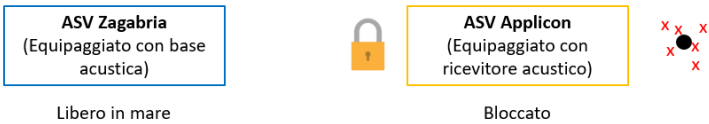
- Posizione di AULoS calcolata tramite SBL Platform

OBBIETTIVO:
 Localizzare AULoS usando l'SBL della Platform e stimare la qualità della localizzazione.

N.B.: fare test al variare della distanza. (CARATTERIZZAZIONE)

• TEST 2:

Descrizione: Il veicolo di Zagabria è libero in mare, mentre il veicolo Applicon è ancorato; la distanza è di circa 20 metri. Prendiamo quindi la posizione assoluta del ricevitore.
Durata: 5 minuti



Nome file ROS	
Distanza effettiva	
Ora di inizio	
Ora di fine	

• TEST 3:

Descrizione: Il veicolo di Zagabria è libero in mare ma con la funzione Hovering attiva, mentre il veicolo Applicon è ancorato; la distanza è di circa 20 metri. Prendiamo quindi la posizione assoluta del ricevitore.

Durata: 5 minuti

ASV Zagabria
(Equipaggiato con base
acustica)

Hovering



ASV Applicon
(Equipaggiato con
ricevitore acustico)

Bloccato



Nome file ROS	
Distanza effettiva	
Ora di inizio	
Ora di fine	

LUOGO:

Acqua, luogo aperto, area estesa,
profondità adeguata

HARDWARE NECESSARIO:

- Ricevitore AULoS (**fixed**)
- Platform Zagabria con SBL
- ATLAS

OUTPUT:

- Posizione di AULoS calcolata tramite SBL Platform

OBIETTIVO:

Localizzare AULoS usando l'SBL della Platform e stimare la qualità della localizzazione.

PROTOCOLLO:

La platform è in hovering;
AULoS è fissato;
L'ATLAS è solidale con AULoS;
Si calcola la posizione assoluta di AULoS tramite l'SBL e si confronta con la stima dell'ATLAS.

N.B.: fare test al variare della distanza. (CARATTERIZZAZIONE)

• TEST 4:

Descrizione: Il veicolo di Zagabria viene fatto muovere con il radiocomando lungo un percorso, o anche semplicemente lungo una retta, mentre il veicolo Applicon è ancorato. Prendiamo quindi la posizione assoluta del ricevitore.

Durata: 5 minuti



ASV Zagabria
(Equipaggiato con base
acustica)

Radiocomandato



ASV Applicon
(Equipaggiato con
ricevitore acustico)

Bloccato



Nome file ROS	
Distanza effettiva	
Ora di inizio	
Ora di fine	

LUOGO:

Acqua, luogo aperto, area estesa,
profondità adeguata

HARDWARE NECESSARIO:

- Ricevitore AULoS (**fixed**)
- Platform Zagabria con SBL

OUTPUT:

- Confronto tra localizzazione GPS e localizzazione AULoS

HARDWARE NECESSARIO:

- Ricevitore AULoS (**fixed**)
- Platform Zagabria con SBL
- ATLAS

OUTPUT:

- Posizione di AULoS calcolata tramite SBL Platform

OBIETTIVO:

Localizzare AULoS usando l'SBL della Platform e stimare la qualità della localizzazione.

PROTOCOLLO:

La platform segue un path guidato (mission planner*);
AULoS è fissato;
L'ATLAS è solidale con AULoS;
Si calcola la posizione assoluta di AULoS tramite l'SBL e si confronta con la stima dell'ATLAS.

***: non è necessario però potrebbe essere un valore aggiunto. In realtà l'obiettivo non è fare il tracking di AULoS.**

• TEST 5:

Descrizione: Il veicolo di Zagabria è ancorato, mentre il veicolo Applicon viene fatto muovere con il radiocomando lungo un percorso, o anche semplicemente lungo una retta. Prendiamo quindi la posizione assoluta del ricevitore.

Durata: 5 minuti



ASV Zagabria
(Equipaggiato con base acustica)

Bloccato



ASV Applicon
(Equipaggiato con ricevitore acustico)

Radiocomandato



LUOGO:

Acqua, luogo aperto, area estesa, profondità adeguata

Nome file ROS	
Distanza effettiva	
Ora di inizio	
Ora di fine	

HARDWARE NECESSARIO:

- Ricevitore AULoS (in movimento)
- Platform Zagabria con SBL
- ATLAS
- DEVSS oppure diver*

OUTPUT:

- Posizione di AULoS calcolata tramite SBL Platform

OBIETTIVO:

Localizzare (tracking) AULoS usando l'SBL della Platform e stimare la qualità della localizzazione.

PROTOCOLLO:

La platform è ferma;
AULoS è in movimento;
AULoS deve essere localizzato oltre che dal sistema acustico anche tramite un altro sistema per poter avere una ground truth (alternative):

- Diver: riferimenti ottici* (timer sincronizzato tra AULoS e sistema di acquisizione);
- DEVSS: GPS differenziale;

Si calcola la posizione assoluta di AULoS tramite l'SBL e si confronta con la ground truth.

* Servono dei marker georeferenziati per determinare la pos. del diver.

• TEST 6:

Descrizione: Il veicolo di Zagabria è in modalità di navigazione automatica, mentre il veicolo Applicon viene fatto stazionare in mare. Prendiamo quindi la posizione assoluta del ricevitore, e vediamo come si comporta il controllore sull'ASV di Zagabria.

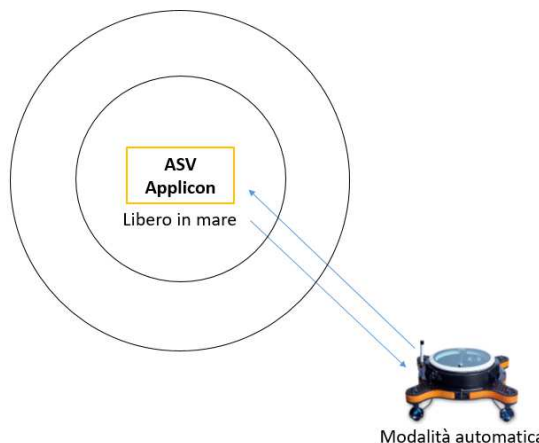
Durata: 5 minuti

Il veicolo di Zagabria prenderà periodicamente delle nuove posizioni e si adatterà alla corona circolare. Sarebbe interessante variare i parametri del controllore (r_{min} , r_{max} , A, R)

LUOGO:

Acqua, luogo aperto, area estesa, profondità adeguata (non necessariamente in fase preliminare)

Nome file ROS	
Distanza effettiva	
Ora di inizio	
Ora di fine	



HARDWARE NECESSARIO:

- Ricevitore AULoS (fissato)
- Platform Zagabria con SBL
- ATLAS
- DEVSS oppure diver

PROPEDEUTICO:

Il ricevitore deve funzionare bene.

INPUT:

Distanza relativa tra ricevitore e platform.

PROTOCOLLO:

La platform è in modalità automatica (inseguimento del ricevitore); AULoS è in fissato in un punto georeferenziato;

Si calcola la posizione assoluta della platform rispetto ad AULoS.

OBIETTIVO:

Studiare il comportamento del sistema considerando l'algoritmo dei PV.

N.B.: La corona circolare va dimensionata in base all'errore del ricevitore. Più l'area della corona è grande più è incerta la localizzazione.

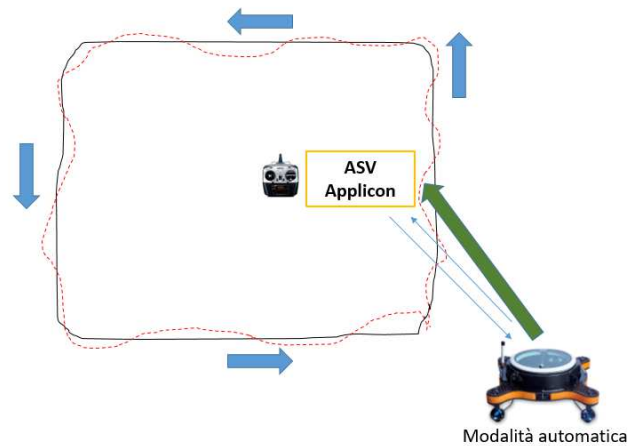
• TEST 7:

Descrizione: Il veicolo di Zagabria è in modalità di navigazione automatica, mentre il veicolo Applicon viene fatto muovere con il radiocomando lungo un percorso. Prendiamo quindi la posizione assoluta del ricevitore, e vediamo come si comporta il controllore sull'ASV di Zagabria.

Durata: 5 minuti

LUOGO:
Acqua, luogo aperto, area estesa,
profondità adeguata

Nome file ROS	
Distanza effettiva	
Ora di inizio	
Ora di fine	



HARDWARE NECESSARIO:

- Ricevitore AULoS
- Platform Zagabria con SBL
- ATLAS
- DEVSS oppure diver*

OUTPUT:

- Posizione di AULoS calcolata tramite SBL Platform

OBIETTIVO:

Studiare le performance di tracking del sistema.

PROTOCOLLO:

La platform è in modalità automatica (inseguimento del ricevitore); AULoS viene movimentato da DEVSS (alla giusta profondità) oppure dal diver;

AULoS deve essere localizzato oltre che dal sistema acustico anche tramite un altro sistema per poter avere una ground truth (alternative):

- Diver: riferimenti ottici* (timer sincronizzato tra AULoS e sistema di acquisizione);
- DEVSS: GPS differenziale;

Si calcola la posizione assoluta di AULoS rispetto all'SBL della platform.

* Servono dei marker georeferenziati per determinare la pos. del diver.

Capitolo 5

Il sistema commerciale USBL Seatrac della Blueprint: soluzioni adottate per il miglioramento delle sue performance.

In questo capitolo è descritto lo studio condotto sul miglioramento delle performance di un sistema di localizzazione acustica di tipo USBL (Ultra Short Baseline) commerciale. A seguito di una breve descrizione di questo sistema, sono presentati una serie di test sperimentali che hanno dato luogo a dei risultati non troppo soddisfacenti relativamente all'accuratezza. In seguito sono descritte una serie di soluzioni adottate per il miglioramento delle performance del sistema. Inizialmente si è pensato ad una soluzione di tipo meccanico, ovvero si è studiato un modello multi-body di uno stabilizzatore meccanico per gestire la compensazione delle oscillazioni del sistema USBL. Successivamente, si è pensato di intervenire via software implementando un filtro statistico a finestra mobile in grado di interpolare i dati al fine di fornire una stima più precisa della posizione. Come terza strategia di miglioramento, si è infine pensato di implementare una soluzione di tipo ibrido che ha previsto la fusione dei dati di posizione del sistema acustico USBL con quelli ottici provenienti dal sistema ARKit di un tablet della Apple.

5.1 - Il sistema USBL Seatrac e test preliminari

Durante la fase di sperimentazione, si è pensato di utilizzare un sistema USBL (Ultra-short baseline) per effettuare alcuni test. Si è scelto di acquistare il sistema USBL Seatrac della Blueprint [5.01] che ha consentito, nelle varie fasi di studio, di effettuare dei confronti con il sistema SBL, quindi è risultato essere molto utile.

5.1.1 - Descrizione dell'USBL Blueprint Seatrac

La tecnologia USBL Seatrac è comprensiva di due moduli hardware: in particolare, sono stati utilizzati i moduli X150 e X010 (figura 5.1). Durante i test sperimentali, il primo, che rappresenta il cuore del sistema di riferimento globale, è stato tenuto insieme da un filo teso per ridurre al minimo le oscillazioni causate dalle correnti marine, mentre il secondo è stato portato dal sub durante l'esecuzione dei percorsi subacquei. Interrogando acusticamente il trasduttore X010 con l'X150, e utilizzando algoritmi basati su tecniche di triangolazione, è possibile risalire alla posizione X010 con una certa frequenza, rispetto ad un sistema di riferimento noto. La posizione può essere geo-referenziata con un GPS e può essere gestita anche localmente con coordinate NED (nord, est, down).



Figura 5.1 – Moduli X150 e X010 del sistema di localizzazione acustico subacqueo USBL Seatrac della Blueprint.

Volendo fare un breve confronto, USBL Seatrac e SBL Seamodem (Applicon s.r.l.) sono tecnologie diverse, e l'obiettivo è stato di testarle entrambe per misurare le prestazioni e valutarne il rendimento durante gli esperimenti condotti. I due sistemi hanno velocità di aggiornamento differenti. Il Seamodem è infatti in grado di acquisire più punti nello stesso intervallo di tempo. Il Seatrac è compatto e molto facile da trasportare, ma allo stesso tempo ha una precisione inferiore.

5.1.2 - Test sperimentali per la localizzazione dei sub

Il test descritto in questo paragrafo ha avuto come obiettivo quello di migliorare le performance di geo-localizzazione dei sub in ambiente subacqueo, legando le funzionalità dei GPS con particolari strumenti acustici, gli USBL (Ultra Short Baseline), estremamente compatti e versatili, che hanno sostituito gli SBL e LBL in numerose applicazioni. In questo lavoro è stata utilizzata la tecnologia degli USBL sviluppata dall'azienda Blueprint, comprensiva di due moduli hardware e dei relativi software proprietari. In particolare, sono stati adoperati i moduli Seatrac X150 ed X010: il primo solidale all'imbarcazione, quindi ad sistema di riferimento globale, mentre il secondo trasportato da un operatore subacqueo nello scenario marino. Mediante l'interrogazione acustica del trasduttore X010 da parte del X150, ed avvalendosi di algoritmi che si basano su tecniche di triangolazione, è possibile risalire alla posizione del X010 con una certa frequenza, rispetto ad un sistema di riferimento noto. La posizione può essere geo-referenziata con un GPS, e può essere anche gestita in locale con coordinate NED (nord, east, down), ricavabili dal post-processing di stringhe NMEA. In particolare, nel test in questione, si è adoperato un GPS Applanix (figura 5.2).

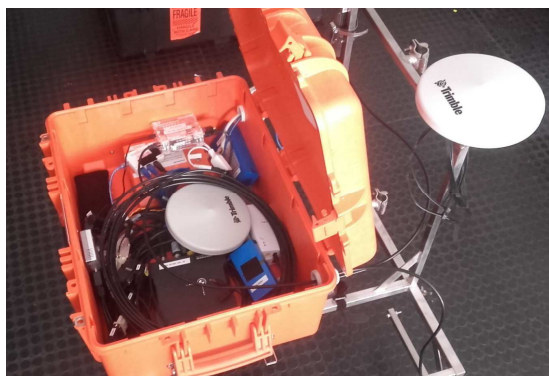


Figura 5.2 – Sistema di geo-localizzazione Applanix.

In figura 5.3 è mostrato lo schema della strumentazione utilizzata per l'esecuzione del test. Tutta la strumentazione è connessa ad un mini-pc dal quale è possibile sincronizzare le operazioni e salvare i dati.

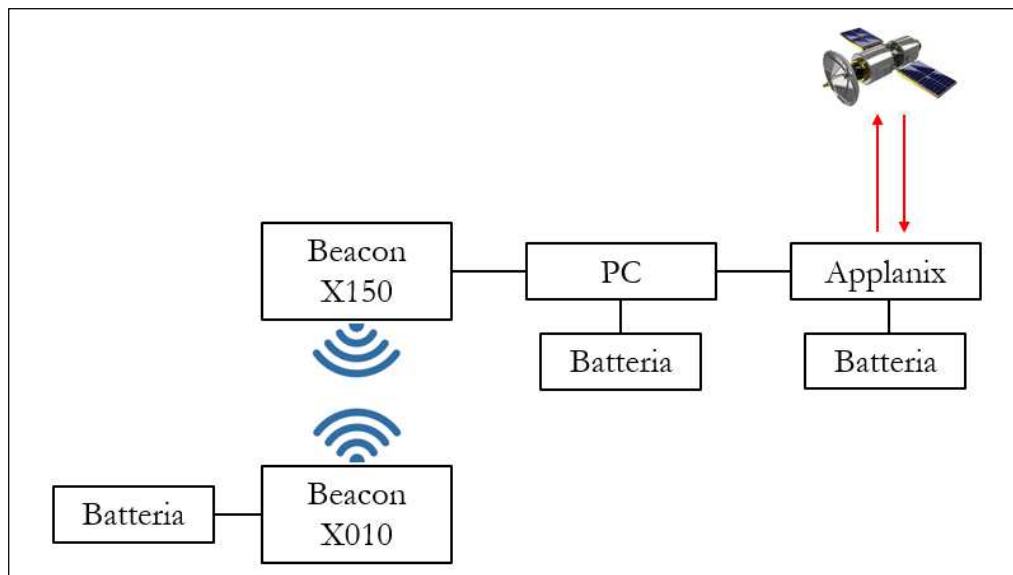


Figura 5.3 – Schema semplificato dell'hardware utilizzato.

Sono stati effettuati una serie di test preliminari in una piscina per la comprensione dei software di controllo e per un giusto settaggio dei parametri operativi, nonché per la corretta integrazione del sistema GPS con la strumentazione USBL. Successivamente, ci si è recati in un'area balneare per escludere dalle acquisizioni gli effetti di riverbero dei segnali acustici causati dalle pareti della piscina. Si è pensato di delimitare l'area del fondale marino, a circa 6 metri di profondità, con 4 marker di riferimento come quello mostrato in figura 5.4, rappresentante uno dei 4 vertici di un quadrato di lato 10 metri.



Figura 5.4 – Uno dei 4 marker posizionati sul fondale, con relativo galleggiante.

L'obiettivo della fase sperimentale ha previsto che il sub ripercorresse i 4 vertici del quadrato, e mediante il monitoraggio delle posizioni, si tenesse traccia del percorso. In particolare, è stato eseguito il tragitto in due configurazioni: nella prima si è assunto che il modulo X010 stesse dietro la bombola dell'ossigeno (figura 5.5 a), mentre nella seconda che stesse davanti al sub (figura 5.5 b); in questo modo è stato possibile valutare l'entità del disturbo delle bolle d'aria, prodotte dalla respirazione del sub, sulla trasmissione dei segnali acustici.

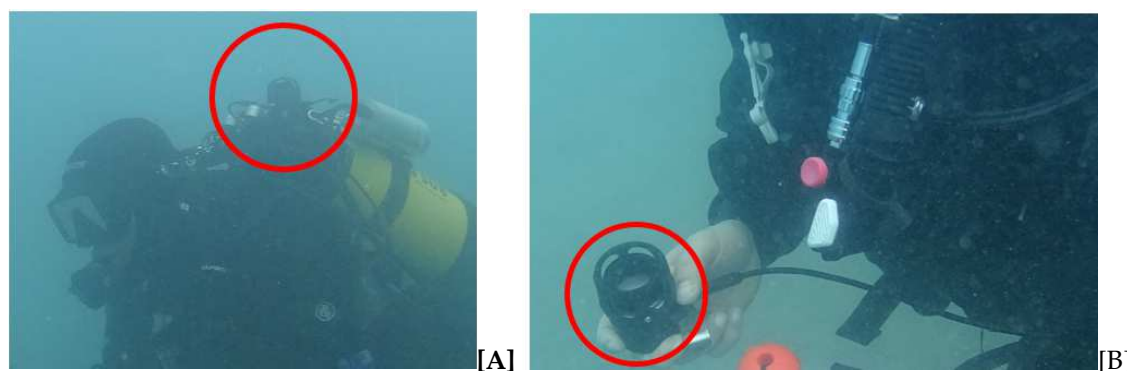


Figura 5.5 – Posizione del X010 dietro la bombola (a) e davanti (b).

Per entrambe le configurazioni, si è partiti dal marker iniziale (1) e, passando per i marker (2)(3)(4), si è ritornati a quello di partenza (5); per tener traccia delle posizioni e sincronizzarle con la storia temporale registrata dei segnali acustici, si è scattata una foto al raggiungimento del sub su ogni marker. Questo modo di procedere è risultato comodo per riordinare i dati del test sfruttando il fatto che è possibile estrapolare successivamente l'orario dalle fotografie. La fase di elaborazione dati ha previsto quindi il post-processing di stringhe NMEA ottenute durante il test, generate con una frequenza di campionamento pari ad 1Hz. Si è risaliti alle coordinate spaziali del sub all'interno dello scenario marino rispetto ad un sistema di riferimento globale solidale all'imbarcazione.

Come previsto, e come mostrato in tabella 5.1, è stato inevitabile che una parte di dati si sia persa durante la comunicazione dei trasduttori. Si noti che la perdita di stringhe derivanti dalla configurazione con beacon davanti la bombola, è stata pari al 96,43%.

Tabella 5.1 – I due test a confronto con relativi dati persi durante l'esecuzione delle prove.

Configurazione	Campioni teorici	% Campioni persi
Beacon dietro la bombola	151	59,60%
Beacon davanti la bombola	252	96,43%

Sono stati ritenuti validi soltanto i dati relativi alla configurazione con beacon dietro la bombola. È stato ricostruito il percorso ed è stato confrontato con quello teorico (figura 5.6).

5.1.4 - Elaborazione dati del GNSS

In questa sezione sono descritti i risultati provenienti da test statici che hanno previsto che il ricevitore acustico stesse fermo nell'ambiente subacqueo. Dopo una fase di post-processing in Matlab, si è potuto caratterizzare l'errore massimo che si ha sulla posizione. In particolare, nelle figure 5.8 e 5.9 sono riportati i risultati in termini di latitudine e longitudine.

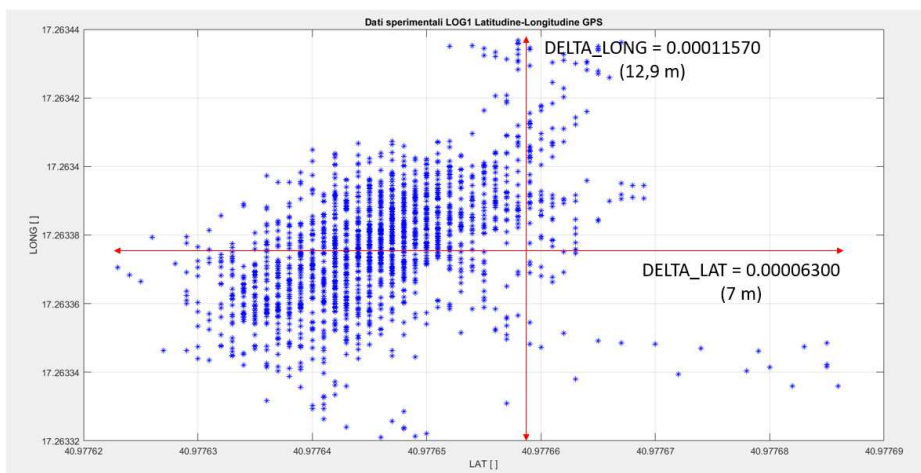


Figura 5.8 – Dispersione dei dati di posizione del ricevitore subacqueo (Log1).

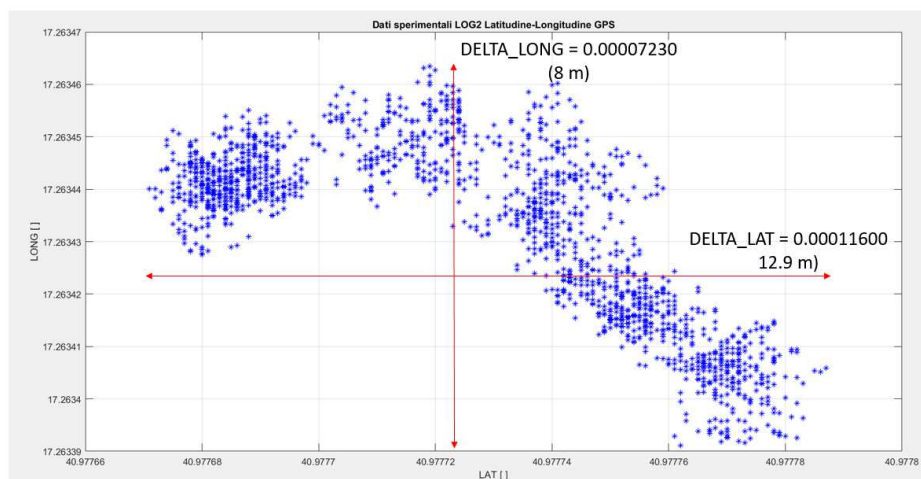


Figura 5.9 – Dispersione dei dati di posizione del ricevitore subacqueo (Log2).

5.1.5 - Individuazione delle principali fonti di errore

A valle dei test eseguiti in diversi contesti, è stato possibile individuare le principali sorgenti di errore che vanno ad alterare il risultato finale di posizionamento. Queste sorgenti sono:

- GPS;
- IMU;
- presenza di bolle.

Queste sorgenti di errore sono state analizzate criticamente e sono state proposte due soluzioni, descritte nei paragrafi seguenti, per migliorare le performance del sistema, intervenendo sia dal punto di vista hardware sia software.

5.2 - Studio di un sistema per la riduzione delle oscillazioni dell'USBL

Il primo approccio che è stato studiato per la riduzione dell'errore di localizzazione dei dispositivi USBL tenta di ridurre meccanicamente le oscillazioni della base del dispositivo che è tipicamente montata su un palo solidale con l'imbarcazione di appoggio dei divers o su un ASV come descritto in precedenza. L'idea è quindi di provare a mantenere la base dell'USBL in posizione quanto più verticale possibile attraverso uno smorzatore per il sistema USBL in questione, che possa minimizzare le oscillazioni prodotte dal moto ondoso andando quindi a migliorare l'accuratezza complessiva del posizionamento.

5.2.1 - Analisi della problematica

Come analizzato nei paragrafi precedenti l'aumentare dell'inclinazione del Seatrac, seppur avvenga la correzione via software grazie all'unità inerziale (AHRS), genera un rumore e quindi un peggioramento dell'accuratezza della localizzazione. Le condizioni ideali sarebbero che il Seatrac stesse perfettamente perpendicolare alla superficie dell'acqua. In figura 5.10 è rappresentato il layout di partenza, sul quale è stato necessario intervenire, mentre in figura 5.11 è rappresentato uno schema nel quale si evince come un'inclinazione dell'unità X150 introduca un certo errore proporzionale alla sua inclinazione.



Figura 5.10 – Layout di partenza.

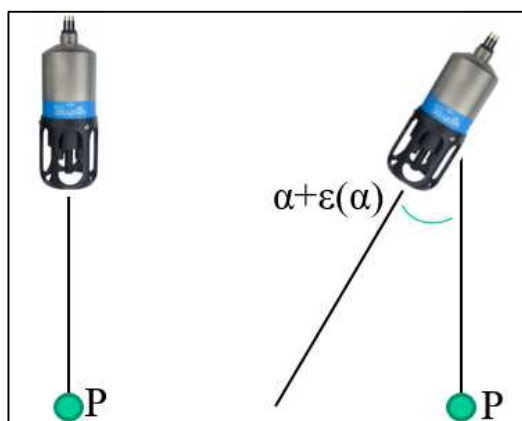


Figura 5.11 – Introduzione di un errore a seguito dell'inclinazione dell'unità X150.

L'intervento di compensazione numerica che si può ottenere utilizzando un sensore IMU solidale con il dispositivo non è sufficiente perché l'errore di misura dell'inclinazione si ripercuote sull'errore di posizione del ricevitore in maniera proporzionale alla distanza fra i due trasduttori. Lo studio presentato di seguito ipotizza quindi di poter intervenire fisicamente sul sistema per garantire una stabilizzazione dinamica dell'unità X150.

5.2.2 Modello multi-body

In questa sezione è presentata l'idea del progetto dello smorzatore passivo, il suo funzionamento e quali sono le parti che lo compongono; successivamente, è descritto lo sviluppo di un modello multi-body che ha permesso di individuare una serie di parametri per il suo dimensionamento. È stato utilizzato l'ambiente Matlab/Simulink per procedere alla caratterizzazione del sistema, con il particolare uso del pacchetto di Simulink relativo alla meccanica (Simscape multi-body). Riferendoci alla figura 5.12, che rappresenta il sistema nell'ambiente multi-body, è possibile individuare 4 parti:

- 1) l'imbarcazione, soggetta al movimento delle onde;
- 2) parte di aggancio dello smorzatore passivo all'imbarcazione (sistema di riferimento in blu), che sarà sempre solidale ad essa ed asseconderà tutti i suoi movimenti;
- 3) prima parte mobile dello smorzatore, in particolar modo connessa alla parte 2 mediante un giunto passivo lineare (sistema di riferimento in verde) che permette di smorzare il movimento del Seatrac lungo l'asse Z;
- 4) Seatrac (cilindro) con asta di galleggiamento e massa concentrata (sferetta), è tutto un unico corpo. Tutta la parte 4 è connessa alla parte 3 mediante un giunto passivo sferico (sistema di riferimento celeste) che permette di smorzare le rotazioni del Seatrac rispetto agli assi X e Y. Sull'estremità superiore è prevista l'installazione di un galleggiante.

Complessivamente, tutto il sistema è in grado di ridurre una traslazione e due rotazioni del Seatrac quando l'imbarcazione si muove. Il baricentro del sistema asta-massa-Seatrac, in fase di simulazione, è stato fatto coincidere con il centro dei volumi.

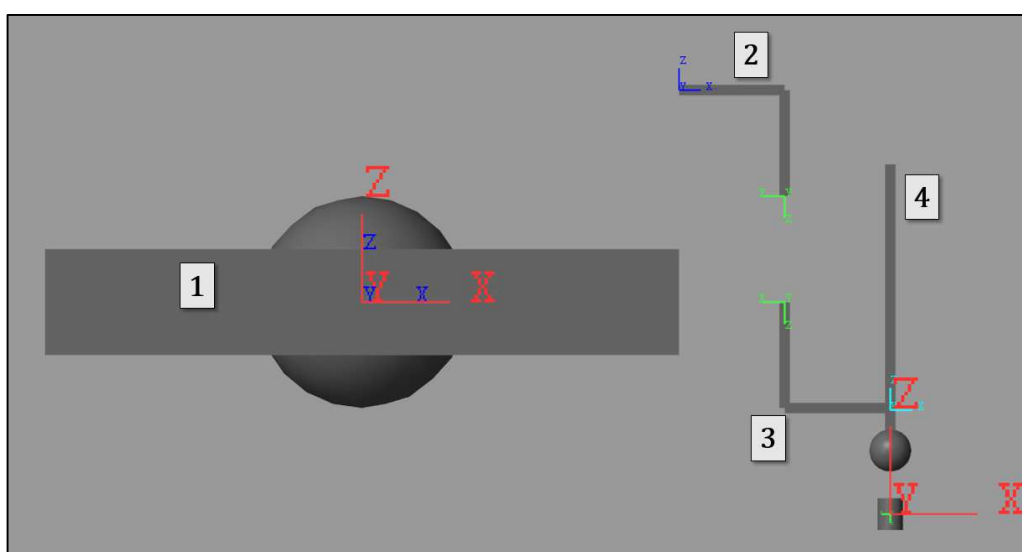


Figura 5.12 – Elementi del modello grafico multi-body.

In figura 5.13 è raffigurata una rappresentazione 3D del modello multi-body con i relativi sistemi di riferimento.

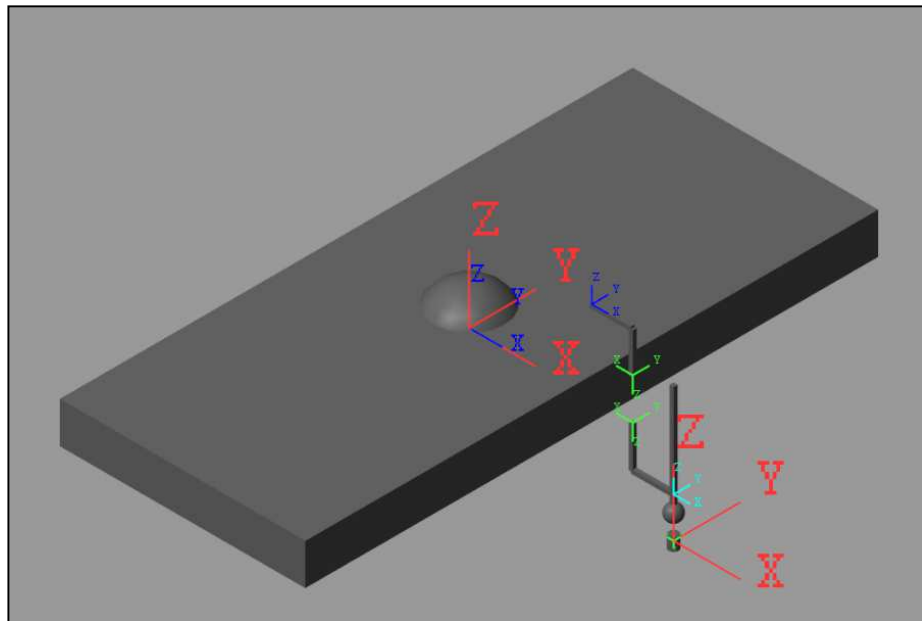


Figura 5.13 – Sistemi di riferimento del modello multi-body.

Per creare il modello grafico appena descritto si è adoperato lo schema Simulink presente in figura 5.14, nel quale sono presenti i blocchetti per l'attuazione, per i giunti, per le conversioni, per il monitoraggio dei parametri di interesse, ed i blocchetti nei quali sono implementate le varie matrici di roto-traslazione per passare da un generico sistema di riferimento al successivo o al precedente.

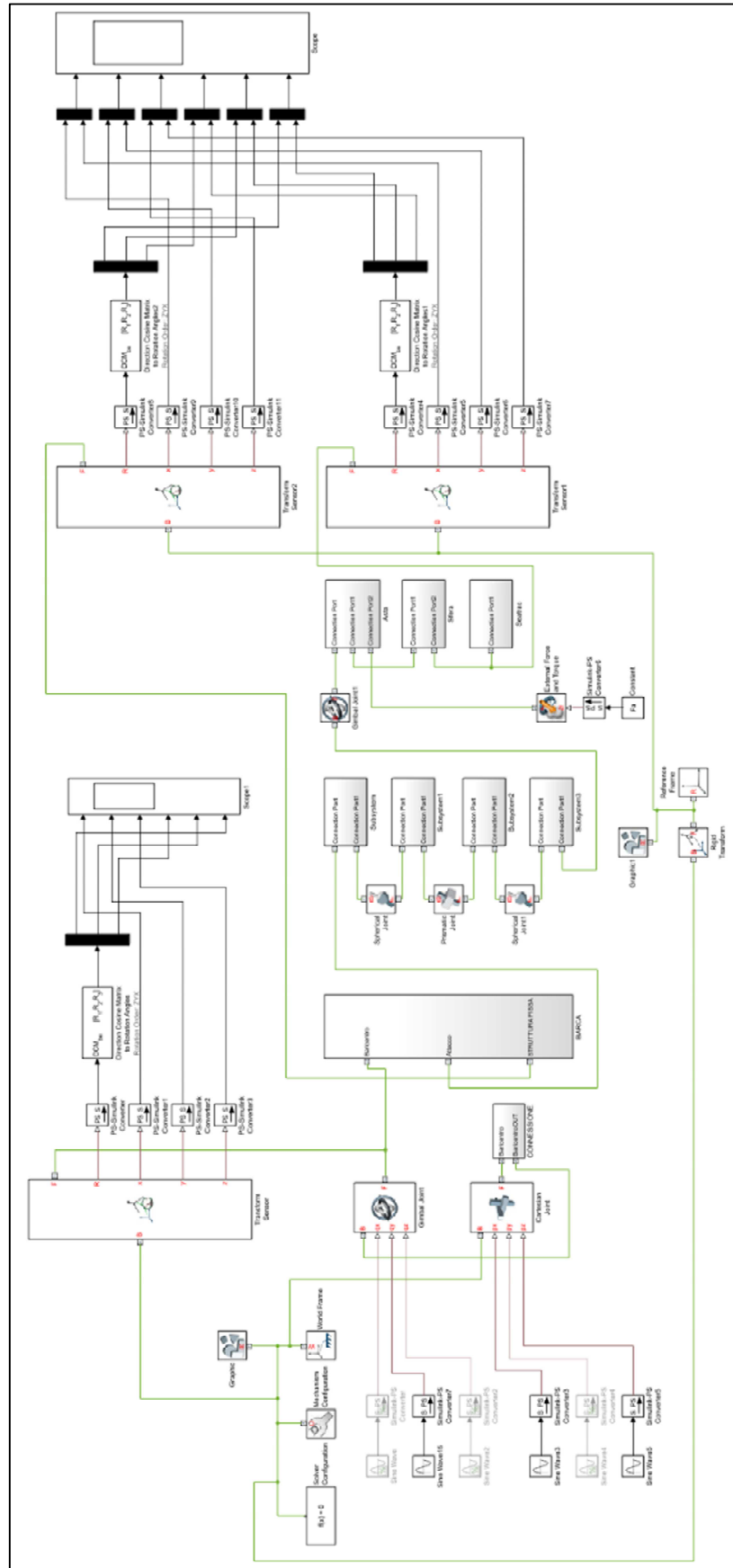


Figura 5.14 – Schema Simulink per la creazione del modello multi-body.

5.2.3 - Risultati analitici

Le simulazioni condotte hanno previsto l'utilizzo dei parametri riportati in tabella 5.2.

Tabella 5.2 – Parametri del modello multi-body.

Parametro	Valore	Descrizione
d_1	1 [m]	lunghezza asta parte 1
d_2	0.5 [m]	lunghezza asta parte 2
d_3	0.2 [m]	lunghezza asta parte 3
d_asta	1.35 [m]	lunghezza asta
ro_s	2700 [Kg/m ³]	densità media della struttura
ro_f	1027 [Kg/m ³]	densità del fluido
K_l	1000 [N/m]	rigidezza del giunto lineare
C_l	10 [N/(m/s)]	coefficiente di smorzamento del giunto lineare
K_r	0.001 [N*m/deg]	rigidezza del giunto sferico
C_r	0.01 [N*m/(deg/s)]	coefficienti di smorzamento del giunto sferico

Le simulazioni tengono conto del fatto che l'intero sistema sia collocato in acqua. Per modellare la risposta dinamica in modo realistico, sono stati assunti dei coefficienti di smorzamento abbastanza bassi per il giunto lineare e per il giunto sferico. Questo garantisce che l'intero sistema abbia un comportamento sovra-smorzato. Pertanto, in questa fase di studio non è stato necessario procedere con analisi CFD per il calcolo della resistenza idrodinamica in quanto il suo effetto è già intrinseco nei parametri di smorzamento del sistema. Il modello prevede inoltre che sia presente il contributo della forza di Archimede agente sul sistema asta-massa-Seatrac.

Da alcune simulazioni, dopo aver individuato dei parametri realistici e ottimali per quanto riguarda i coefficienti di smorzamento dei giunti e dei coefficienti di rigidezza, nonché dopo aver dimensionato adeguatamente le masse in gioco, si è pervenuti al risultato di figura 5.15. È evidente il netto miglioramento dell'oscillazione del Seatrac il cui angolo è stato identificato con la lettera Beta. L'oscillazione gialla corrisponde al caso in cui non sia presente lo smorzatore passivo, mentre la curva blu quando è presente lo smorzatore passivo. Il miglioramento è netto.

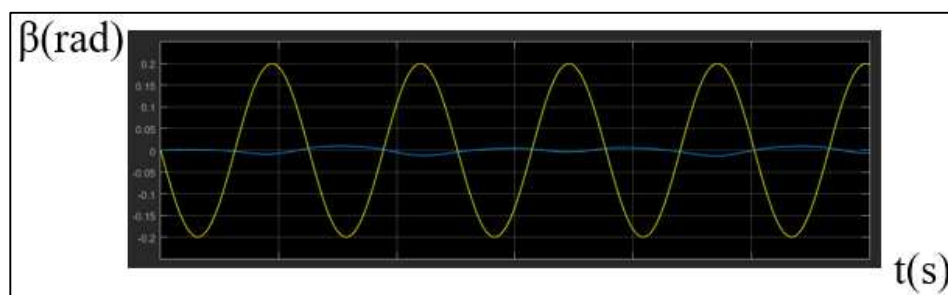


Figura 5.15 – Risultati analitici; in giallo l'oscillazione del sistema senza smorzatore passivo, in blu con la presenza dello smorzatore passivo.

5.2.4 - Dall'analisi numerica alla creazione di un banco prova sperimentale

Incoraggiati dai risultati ottenuti, prima di passare alla realizzazione dello stabilizzatore passivo, si è pensato di dimensionare un banco prova per poter caratterizzare l'errore del Seatrac attraverso una campagna di misurazioni. Si è pensato di condurre le prove facendo oscillare il Seatrac in acqua; il Seatrac è fissato ad un'asta verticale, la quale è azionata attraverso un motore elettrico, fissato a bordo piscina su una struttura. Il motore è controllabile in posizione, e l'idea è di effettuare le prove ad angolo imposto, a diverse frequenze di oscillazione. L'obiettivo del motore elettrico è quindi quello di simulare le oscillazioni del palo dovute al moto ondoso.

Per poter scegliere il motore responsabile dell'attuazione adatto allo scopo, si è dovuto creare un secondo modello in Matlab/Simulink (figura 5.16) affiancato da un software di simulazione numerica CFD. Il modello creato tiene conto di:

- gravità e spinta di Archimede (nel centro dei volumi immersi);
- effetto dell'attrito con l'acqua (ipotizzando regime laminare).

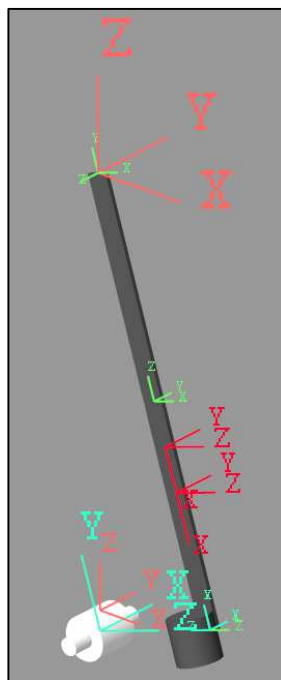


Figura 5.16 – Sistemi di riferimento del modello Simulink implementato per dimensionare il motore del setup sperimentale.

Il modello presentato è stato realizzato mediante la creazione del seguente schema (figura 5.17) in Matlab/Simulink avvalendosi del pacchetto Simscape/Multi-body di Simulink.

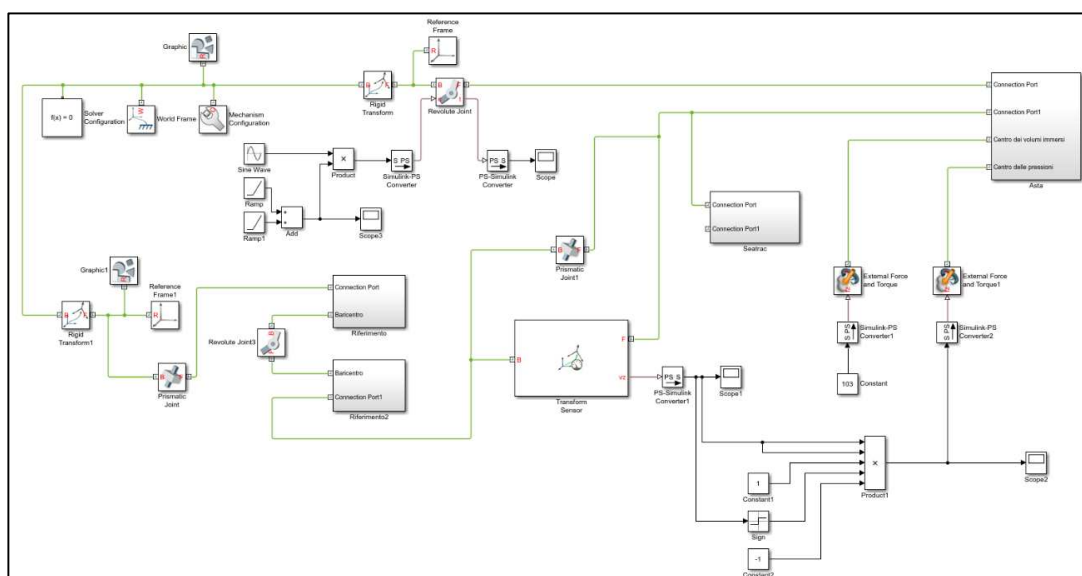


Figura 5.17 – Modello Simulink implementato per dimensionare il motore del setup sperimentale.

5.2.5 - Analisi CFD

È stata avviata una simulazione CFD in SolidWorks per simulare l'interazione che avviene tra il fluido (acqua) ed il sistema USBL Seatrac montato sul suo palo di sostegno (figura 5.18).

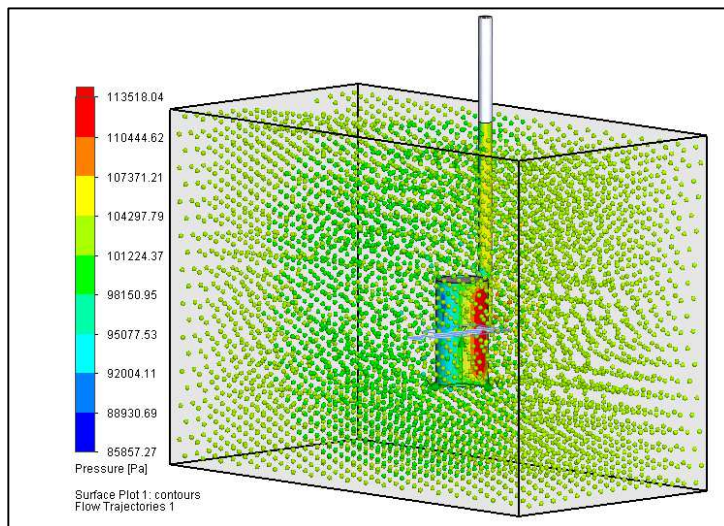


Figura 5.18 – Simulazione CFD in SolidWorks.

La simulazione CFD ha consentito di determinare il coefficiente di forma dell'assieme costituito dal sistema USBL e dal palo, che è stato successivamente utilizzato nel modello Matlab/Simulink per la determinazione delle forze e delle coppie in gioco. In riferimento alla figura 5.19, sono state utilizzate le seguenti equazioni per la determinazione delle grandezze fisiche di interesse, utilizzando un modello analitico approssimativo: la (5.1) permette di calcolare la forza di attrito partendo dai parametri relativi al coefficiente che dipende dalla forma del corpo, al coefficiente di viscosità dinamica del fluido, alla dimensione trasversale del corpo e alla velocità, mentre la (5.2) è la nota formula che permette di calcolare il coefficiente di resistenza (drag coefficient) partendo dai parametri relativi alla forza di trascinamento (drag), densità, velocità al quadrato ed area di impatto.

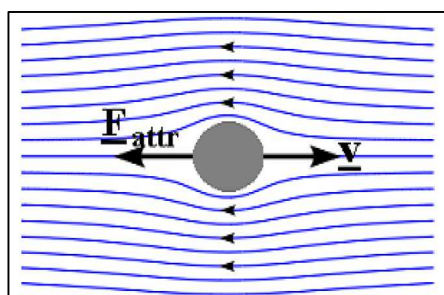


Figura 5.19 – Corpo investito da un fluido in regime laminare.

$$F_{attr} = -k\eta Lv \tag{5.1}$$

$$C_D = \frac{F_D}{\frac{1}{2}\rho v^2 A} \tag{5.2}$$

Il dimensionamento del sistema in simulazione ha permesso di stabilire quali devono essere i parametri meccanici/elettrici del motore da utilizzare in fase di test. In particolare, il valore minimo della sua coppia deve essere pari a 60 [Nm] per garantire l'oscillazione controllata del Seatrac ad una frequenza che sia fedele a quella reale del moto ondoso del mare.

5.3 - Filtro statistico a finestra mobile dinamica

In questo paragrafo è trattato uno studio su un filtro statistico a finestra mobile con grandezza di finestraggio dinamica, applicato ai dati di posizionamento del sistema USBL, con l'obiettivo di migliorare la precisione del tablet subacqueo solidale ad esso. Ci si è ispirati all'articolo scientifico [5.02] il cui fine è quello di proporre delle tecniche di filtraggio per migliorare il dato di posizionamento di un sistema GNSS, quindi per applicazioni 2D terrestri. Gli stessi principi sono stati trattati in questo lavoro di tesi con degli adattamenti ed accortezze per il caso 3D subacqueo. L'attività è stata svolta presso 3DResearch s.r.l..

5.3.1 - Descrizione del setup

L'algoritmo è stato programmato e testato su Matlab/Simulink, per poi passare all'implementazione vera e propria all'interno di un sistema sperimentale sviluppato da 3DResearch s.r.l. (figura 5.20), composto da USBL, tablet e due Raspberry-Pi. L'algoritmo, opportunamente convertito in linguaggio C++, è stato eseguito sulla Raspberry-Pi connessa al ricevitore acustico. In sintesi, l'algoritmo di correzione riceve le coordinate NED (nord, east, down) e le restituisce corrette.

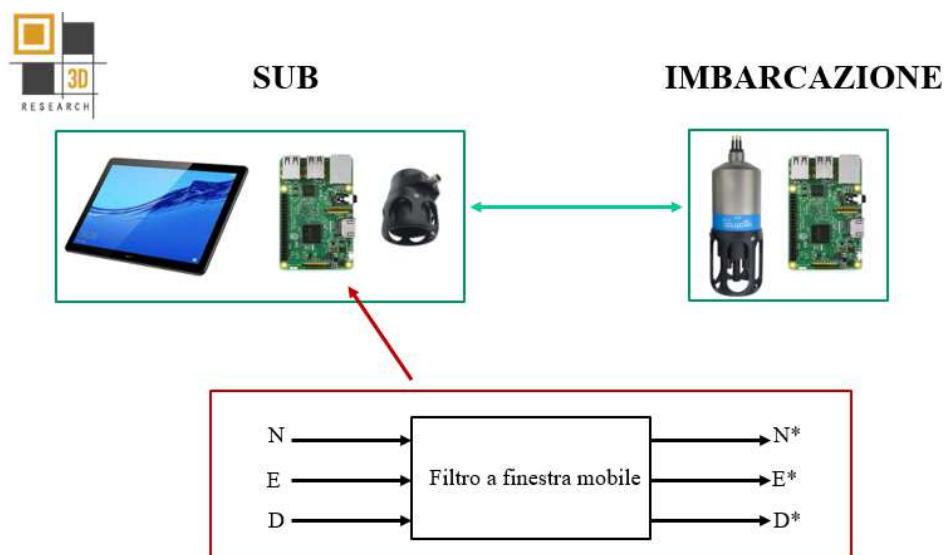


Figura 5.20 – Sistema sperimentale sviluppato da 3DResearch s.r.l. per l'implementazione del filtro statistico a finestra mobile /sistema USBL Seatrac).

5.3.2 - Descrizione dei parametri di input

L'algoritmo presuppone che l'utente definisca i parametri iniziali riportati in tabella 5.3.

Tabella 5.3 – Parametri iniziali dell’algoritmo definiti dall’utente.

IWS	Dimensione minima della finestra
MWS	Dimensione massima della finestra
ϵ	
ET_D	Tolleranza d’errore della distanza
MINSPEED	Velocità minima
$MAX_{acceleration}$	Massima accelerazione possibile

5.3.3 - Descrizione dell’algoritmo

Se si volesse esprimere in sintesi il funzionamento di questo algoritmo, si potrebbe dire che quando si riceve un nuovo valore di posizione $i+1$ (figura 5.21), vengono calcolate velocità ed accelerazioni. Se tali valori calcolati sono anormali rispetto ad una situazione fisicamente accettabile, vuol dire che la nuova posizione acquisita non è valida e deve essere gestita dall’algoritmo mediante alcuni criteri.

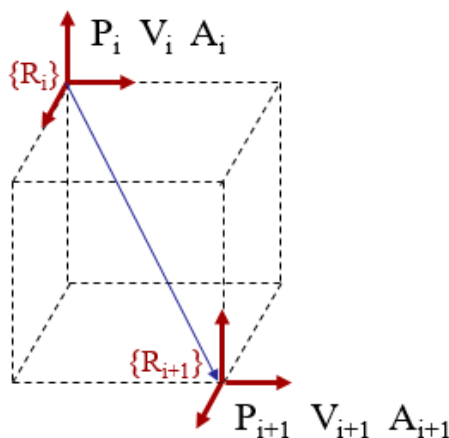


Figura 5.21 – Acquisizione di un nuovo punto.

I parametri di velocità ed accelerazione di soglia sono impostati dall’utente, e sono proprio questi parametri che vengono confrontati con quelli acquisiti dal sistema acustico. Ciascuna tupla, ovvero l’insieme di X, Y, Z e tempo, può in generale essere marcata come:

- default (0)
- filtrata (1);
- filtrata a causa delle accelerazioni (2);
- ripescata (3);
- interpolata (4).

L’algoritmo si occupa di effettuare in successione una serie di calcoli, riportati brevemente in seguito:

- acquisizione di almeno una tupla completa;
- acquisizione coordinate posizione NED;
- calcolo delle velocità;

- calcolo delle accelerazioni;
- calcolo velocità ed accelerazioni nel 3D;
- calcolo delle velocità medie MAX, MAY e MAz nell'intervallo di osservazione.

L'algoritmo è predisposto quindi per ricevere, come parametri di ingresso, una tupla completa indicata con P_{i+1} . Sono inoltre impostabili dall'utente dei parametri di supporto come la dimensione della finestra dinamica n , ed il numero consecutivo di errori NCE.

Questo filtro ha la caratteristica di espandere e contrarre dinamicamente una finestra di osservazione dei dati in tempo reale, a seconda di determinate condizioni; è in grado di fornire una posizione attuale considerando una media mobile la quale tiene conto dei dati precedenti che ricadono nella finestra di osservazione, ed effettua delle previsioni qualora il dato di posizionamento rilevato possa ritenersi poco attendibile. A seconda delle circostanze, il filtro è in grado di ripescare dei dati nel passato e correggere il percorso registrato. In figura 5.22 è riportata la rappresentazione della finestra mobile dinamica, utile per l'osservazione e memorizzazione dei dati.

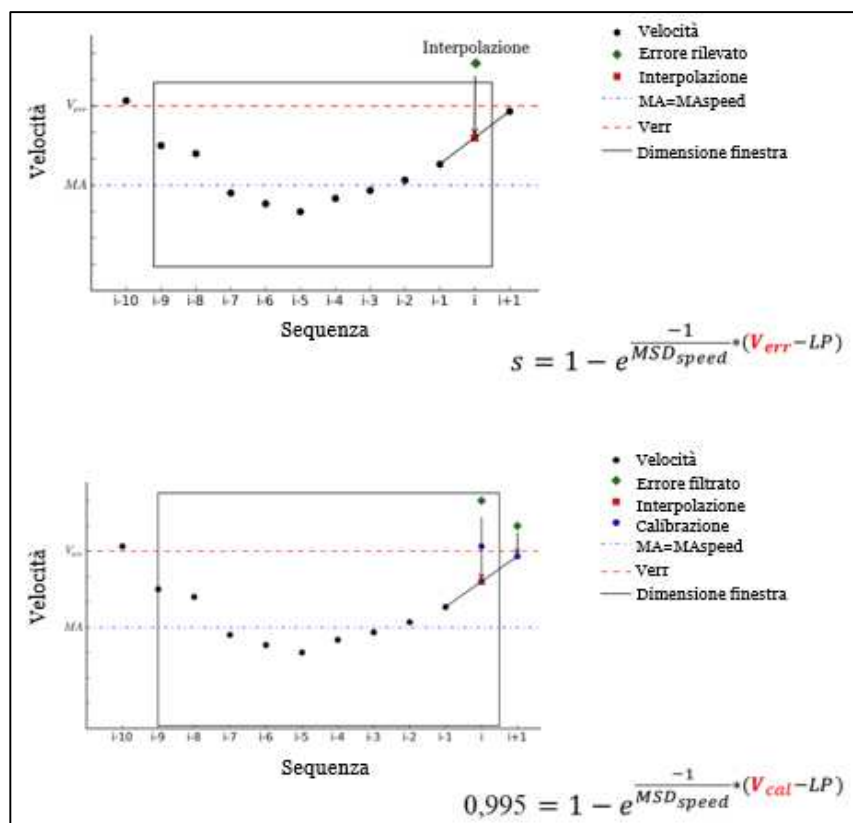


Figura 5.22 – Osservabilità dei dati di velocità attraverso una finestra mobile dinamica.

Il calcolo della velocità minima è reso disponibile dalle equazioni (5.3) e (5.4). La (5.3) è applicata se il valore di velocità rilevato dal sistema acustico è troppo alto, quindi si utilizza il valore di velocità impostata dall'utente che è un valore di soglia; la (5.4) si applica nel caso opposto.

$$MIN_{speed} = MINSPEED \tag{5.3}$$

$$MIN_{speed} = \frac{ET_D}{Epoch_i - Epoch_{i-1}} \quad (5.4)$$

L'algoritmo di calcolo consente di:

- scartare valori di posizione anomali (criterio sulle accelerazioni);
- individuare dei valori di posizione insoliti, con il conseguente ricalcolo a partire dall'interpolazione numerica delle velocità;
- estendere e contrarre la finestra di osservazione;
- ripescare dei dati precedenti (backtracking);
- gestire le diverse operazioni basandosi su una certa affidabilità statistica.

5.3.4 - Risultati in simulazione

In figura 5.23 sono riportati alcuni risultati inerenti ad una simulazione su un percorso quadrato. Sono riportati i diagrammi spazio-tempo per mettere in risalto come la maggior parte dei pallini verdi, ovvero le posizioni filtrate, siano migliori rispetto agli asterischi blu, quanto meno nella maggior parte dei casi. In particolare, si apprezza questa differenza nel grafico della Z. È stato calcolato un indice di performance dell'algoritmo che, su un centinaio di simulazioni condotte, restituisce un valore complessivo di miglioramento pari al 9% rispetto al segnale di posizione non filtrato.

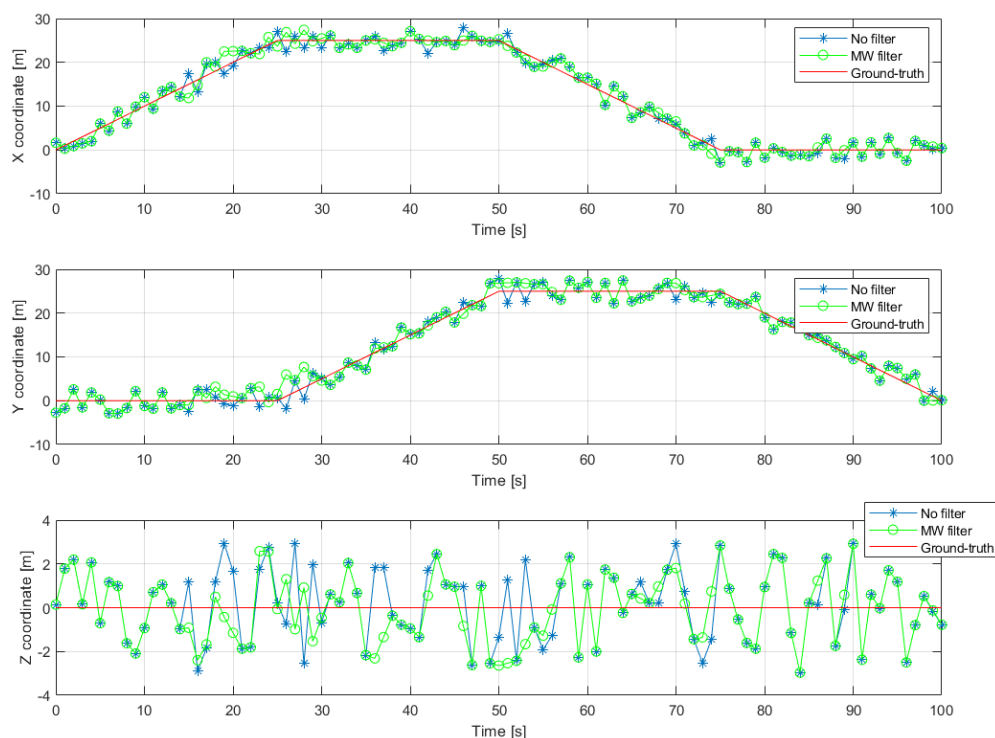


Figura 5.23 – Confronto tra segnale non filtrato e filtrato nei diagrammi tempo-X, tempo-Y, tempo-Z.

Una rappresentazione in un grafico 3D è fornita in figura 5.24:

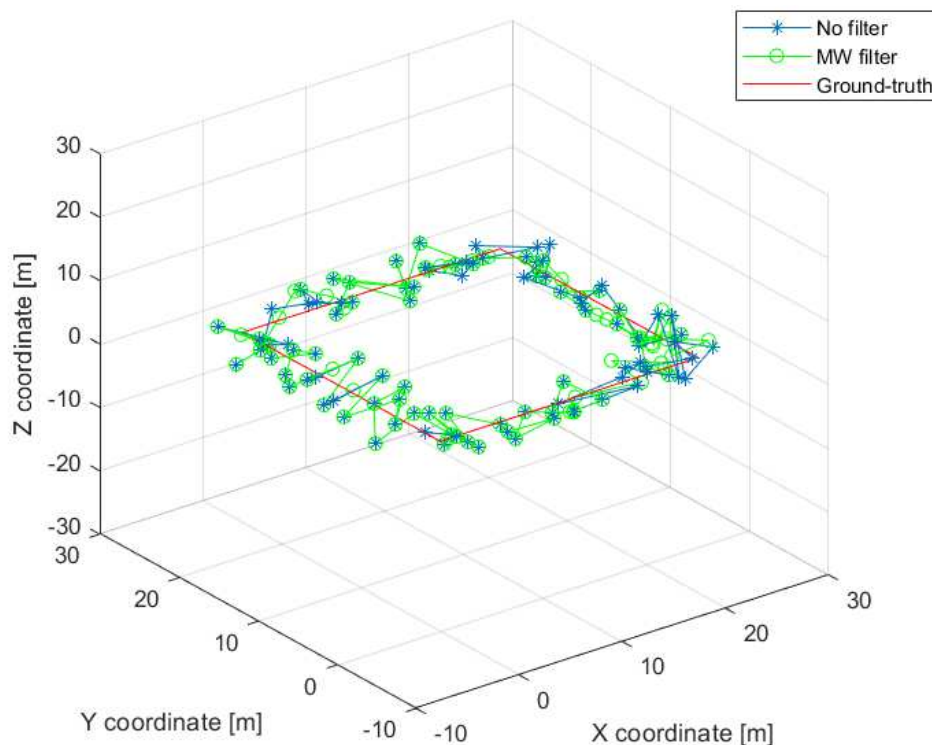


Figura 5.24 – Confronto tra segnale non filtrato e segnale filtrato in un grafico 3D.

I parametri utilizzati per eseguire le simulazioni in questione sono riportati nella tabella 5.4.

Tab. 5.4 – Parametri utilizzati per eseguire le simulazioni.

IWS	7
MWS	12
ε	0,5
ET_D	1
MINSPEED	7 [m/s]
$MAX_{acceleration}$	3 [m/s ²]

5.4 – Sensor-fusion con dati ottici (visual-inertial odometry)

Nel paragrafo 3.5 sono stati descritti alcuni test sperimentali condotti nella zona di Le Castella, in provincia di Crotone. Si è indagato come tecnologie acustiche ed ottiche possano “fondersi” per dar luce a delle soluzioni ibride che migliorano la localizzazione subacquea di un determinato target. Per l’applicazione seguente le condizioni operative, i protocolli dei test e la strumentazione utilizzata sono le stesse, con l’unica differenza che in questo paragrafo sono stati riportati i risultati adottando il sistema di localizzazione USBL sviluppato da Blueprint, invece nel paragrafo 3.5 quelli ricavati dal sistema SBL sviluppato da Applicon s.r.l.. Diverse strategie sono state implementate per il Seatrac (Blueprint) e il Seamodem (Applicon), perché hanno caratteristiche diverse. In particolare, il Seatrac ha una velocità di aggiornamento inferiore, circa 5 secondi per ricevere un dato contro gli 0,1 secondi del Seamodem. Le strategie messe in atto per

il Seatrac hanno avuto l'obiettivo di colmare le lacune tra i dati acustici, e per esse non è possibile avere informazioni sulla qualità del segnale.

Riepilogando in breve gli esperimenti condotti, il sub porta con sé un tablet subacqueo durante l'immersione. Sul tablet viene eseguita un'app sviluppata da 3DResearch che consente di visualizzare una serie di informazioni utili per l'immersione, ed inoltre è fatto eseguire il framework ARKit che consente di fornire una stima della posizione del diver attraverso tecniche VIO (Visual-Inertial Odometry) mediante la camera posta sul retro del tablet. Il modulo X010 Seatrac è compatto e molto facile da trasportare, ma allo stesso tempo tutto il sistema ha una precisione inferiore rispetto al sistema SBL.

Nella tabella 5.5 sono riportate alcune informazioni utili relative ai test condotti, ovvero numero di giri completi e numero di punti acquisiti.

Tab. 5.5 – Descrizione delle Dive-Session eseguite con sistema USBL.

Experiment	Acoustic transducer	ARKit	Complete number of laps of the route	Number of points acquired
Dive session 3	USBL (Seatrac)	Yes	3	12264
Dive session 4	USBL (Seatrac)	Yes	1	3851

I file Json acquisiti durante le prove seguono lo stesso schema di quelli per il sistema SBL, e la loro descrizione è riportata nel paragrafo 3.5.3. Per le Dive session 3 e 4 si è riscontrato un errore di 135 gradi sulla piattaforma inerziale che è stato successivamente corretto in post-processing. Si ribadisce che, a differenza della configurazione SBL, per il Seatrac Blueprint USBL non è stato possibile estrapolare dati riguardanti la qualità dei segnali trasmessi. Inoltre, non si hanno informazioni sui ranges.

5.4.1 – Risultati preliminari

In questo paragrafo sono presentati i risultati derivanti dal post-processing dei test sperimentali condotti. Per la Dive-session 3, sono stati ricreati i seguenti grafici (per la spiegazione del software vedere il paragrafo 3.5.4) riportati in figura 5.25 e 5.26.

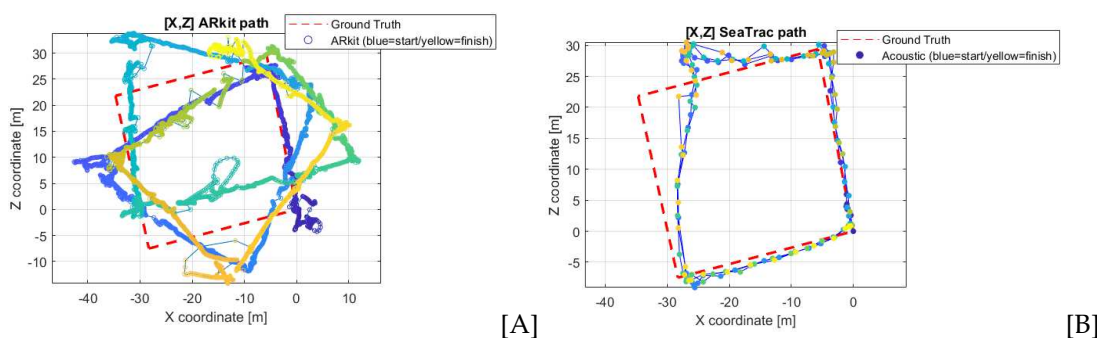


Figura 5.25 – Dati ottici (a) e dati acustici (b) nelle coordinate X-Z per la Dive Session 3.

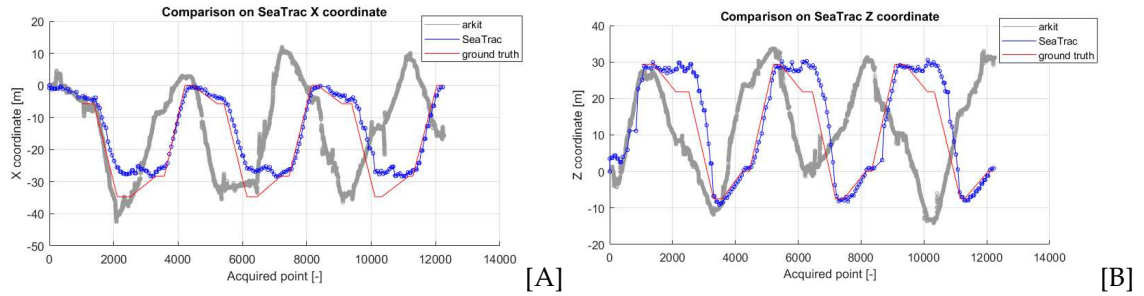


Figura 5.26 – Dati ottici e dati acustici in coordinate NumberAcquisitionPoint-X (a) e NumberAcquisitionPoint-Z per la Dive Session 3.

Per la Dive session 4 gli stessi grafici sono stati riportati nelle figure 5.27 e 5.28.

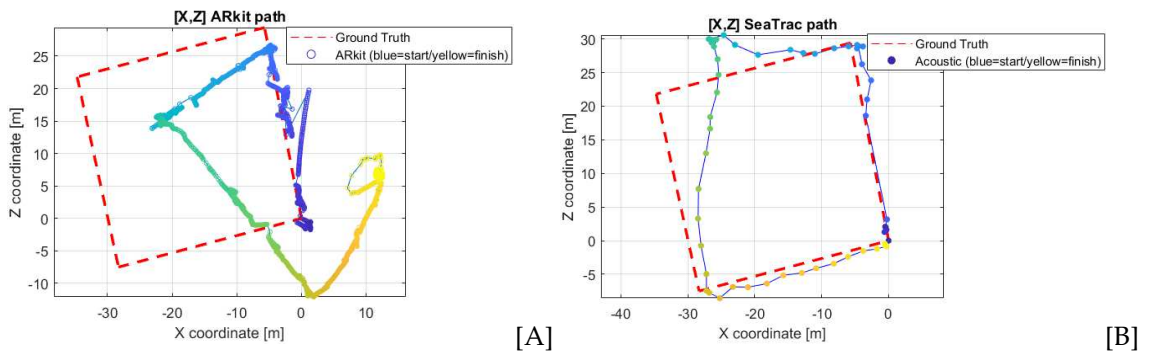


Figura 5.27 – Dati ottici (a) e dati acustici (b) nelle coordinate X-Z per la Dive Session 4.

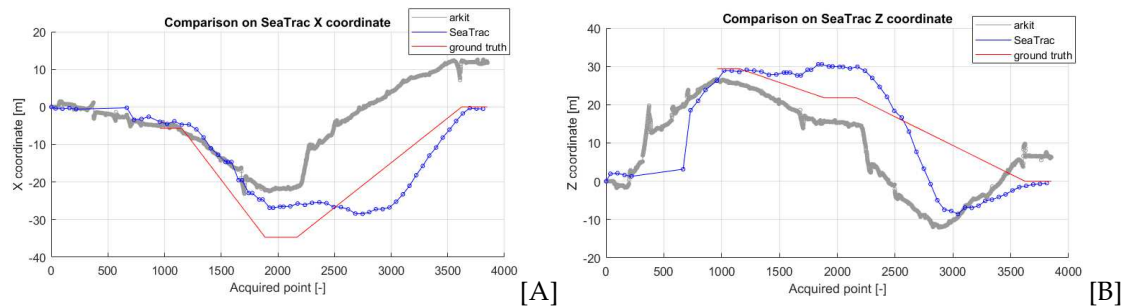


Figura 5.28 – Dati ottici e dati acustici in coordinate NumberAcquisitionPoint-X (a) e NumberAcquisitionPoint-Z per la Dive Session 4.

Nelle figure 5.29(a) e 5.29(b), rispettivamente per le Dive Session 3 e 4, in un diagramma X-Z i dati acustici sono rappresentati solo nel momento in cui il sub è fermo nei vertici della ground-truth; con una linea tratteggiata rossa è rappresentata la ground-truth.

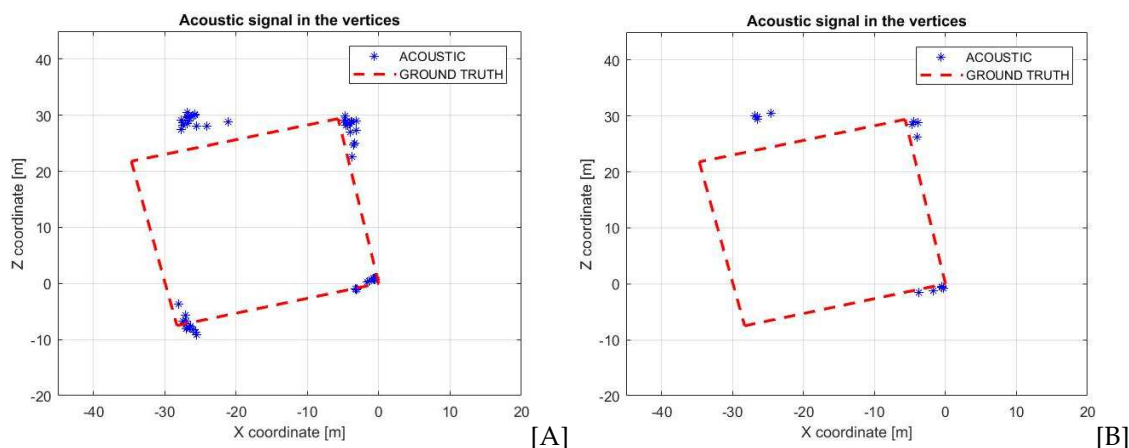


Figura 5.29 – Dati del segnale acustico nei vertici della ground-truth per la Dive Session 3 (a) e per la Dive Session 4 (b).

5.4.2 - Strategie

Partendo dall’analisi dei dati precedentemente acquisiti, sono descritte in seguito le due diverse strategie implementate.

5.4.2.1 - Strategia 1.0

Questa strategia per sistema USBL consiste nel considerare solo il posizionamento acustico quando disponibile e il movimento delta di ARkit quando l’acustica non è disponibile. Il problema di questa strategia è che la direzione dei dati di ARkit non è coerente con i dati acustici. In figura 5.30 il risultato è presentato in un diagramma X-Z. In particolare, viene presentata l’applicazione della strategia solo per il primo giro della Dive-Session 3 per evitare confusione. La ground-truth è rappresentata da una linea tratteggiata rossa, mentre i dati ARkit sono rappresentati in progressione temporale mediante un colore scatter che va dal blu al giallo. I dati acustici sono presentati dalla croce nera. Le linee azzurre rappresentano le connessioni tra i dati acustici e i dati di ARkit, e hanno lo scopo di dare direzionalità al percorso.

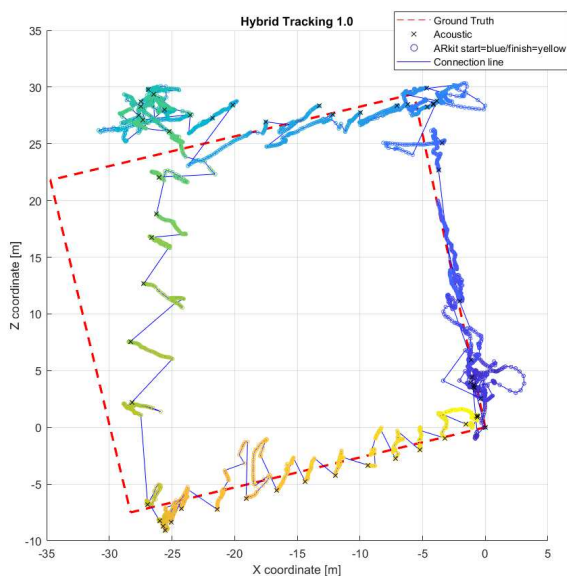


Figura 5.30 – Strategy 1.0 applicata al primo giro della Dive-Session 3.

5.4.2.2 - Strategia 1.1

Questa strategia è sempre per sistema USBL e consiste nel considerare solo il posizionamento acustico quando è disponibile e il movimento delta di ARkit quando l'acustica non è disponibile. La differenza rispetto alla strategia 1.0 è che il delta di ARkit viene ruotato rispetto alla bussola per compensare l'errore di direzione di ARkit. In figura 5.31 è presentata l'applicazione della strategia al primo giro della Dive-Session 3. Le stesse caratteristiche grafiche descritte nella strategia 1.0 sono valide anche per la strategia 1.1.

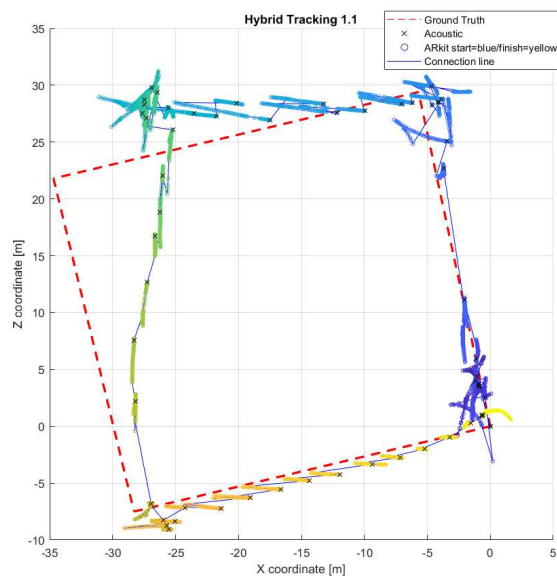


Figure 5.31 – Strategy 1.1 applicata al primo giro della Dive-Session 3.

In generale, i risultati del sistema USBL sono stati più scadenti rispetto a quelli ottenuti dal sistema SBL (paragrafi 3.5.5 e 3.5.6). Il sistema USBL ha una frequenza di aggiornamento dei dati inferiore a quella del sistema SBL, inoltre la qualità è risultata essere globalmente più bassa.

Capitolo 6

Ricostruzione 3D online in ambiente subacqueo utilizzando una camera di profondità a basso costo.

Incoraggiati dal beneficio che potenzialmente è possibile ottenere dalle tecnologie ibride, in particolar modo dai sistemi di localizzazione acustici ed ottici, si è pensato di approfondire meglio gli aspetti legati alle stereo-camere subacquee, che in determinate circostanze potrebbero riuscire a fornire una localizzazione abbastanza precisa se fatte lavorare insieme ad un sistema di localizzazione acustico. Sebbene le tecniche di visione artificiale siano ampiamente ed efficacemente utilizzate per applicazioni terrestri, l'adozione di queste tecniche in scenari subacquei è spesso un compito impegnativo. Ciò è dovuto al mezzo acquoso che degrada la qualità delle immagini acquisite richiedendo l'utilizzo di metodologie specifiche per la loro correzione ed il loro miglioramento. In questo capitolo, si presenta un sistema a basso costo per la ricostruzione 3D subacquea e per misurazioni 3D basato su una fotocamera di profondità standard. Il sistema è costituito da una stereo-camera ZED della Stereolabs, e dal suo alloggiamento impermeabile progettato per essere fabbricato tramite tecnologie di produzione additiva. È stato sviluppato un software dedicato per gestire tutte le fasi del processo di acquisizione, partendo dalla fase di calibrazione e proseguendo per la rettifica, il calcolo della mappa di profondità, la ricostruzione della scena 3D e la misurazione 3D utilizzando le librerie software fornite dal produttore della camera. È stato effettuato un test di laboratorio preliminare in un ambiente controllato per quantificare le prestazioni del sistema nell'accuratezza della misurazione.

6.1 - Introduzione

Le applicazioni della visione artificiale si stanno rapidamente diffondendo in un ampio spettro di operazioni subacquee che richiedono informazioni ambientali estremamente accurate. I sistemi di visione, infatti, sono sempre più utilizzati come strumento primario per l'ispezione di siti sottomarini, in discipline che vanno dall'archeologia alla biologia, all'ingegneria offshore e all'ispezione di condotte [6.01]. Ciò è stato facilitato dalla crescente adozione da parte del settore dei veicoli a comando remoto (ROV) e dei veicoli subacquei autonomi (AUV), che hanno portato a una serie di nuove sfide e applicazioni per la visione artificiale.

Invece di una semplice proiezione bidimensionale, come fa una fotocamera standard, i sistemi di visione 3D subacquei consentono agli operatori di ottenere dati 3D quantitativi e acquisire forme 3D di elementi strutturali subacquei che possono essere utilizzati in molte applicazioni, come la manipolazione robotica subacquea [6.02], e la fluidodinamica computazionale di geometrie 3D ricostruite ottenute dal rilievo subacqueo. Tutti gli scenari e i campi di applicazione sopra menzionati richiedono dati 3D di alta qualità dell'ambiente operativo subacqueo. Questi tipi di dati possono essere ottenuti solo utilizzando sistemi di visione artificiale 3D perché i tradizionali

sistemi di rilevamento acustico non sono in grado di fornire il livello di dettaglio e accuratezza richiesto per tali applicazioni. Tra le varie soluzioni di visione artificiale, la tecnologia di visione stereo binoculare ha dimostrato di essere una soluzione valida e affidabile che ha provato la sua efficacia in molti settori come la ricostruzione di scene 3D, la navigazione con visione robotica, il riconoscimento e il tracciamento di un target, il rilevamento industriale, ecc. [6.03]. Inoltre, potrebbe essere anche una soluzione conveniente, dal momento che, negli ultimi anni, questa tecnologia ha beneficiato della presenza sul mercato di diverse fotocamere a basso costo standard [6.04]. Sebbene l'utilizzo di tali soluzioni a basso costo sia ormai consolidato per applicazioni terrestri, in scenari subacquei la loro adozione è ancora ostacolata dalle limitazioni intrinseche del mezzo che degrada la qualità delle immagini acquisite richiedendo l'utilizzo di metodologie specifiche per la loro correzione e miglioramento. A questo proposito, è importante sottolineare che le immagini catturate da qualsiasi fotocamera subacquea non seguono i principi della proiezione prospettica [6.05] a causa dell'effetto di rifrazione all'interfaccia trasparente dell'alloggiamento della fotocamera.

Uno degli esempi più interessanti descritti in letteratura riguarda l'utilizzo di una stereo-camera Stereolabs® ZED per lo sviluppo di un sistema di visione subacquea in tempo reale per la presa robotica [6.06] e un sistema di visione stereo per il rilevamento di oggetti subacquei e la ricostruzione 3D [6.07]. Ma entrambi i lavori forniscono solo risultati qualitativi.

In questo contesto, il lavoro sviluppato si basa su un sistema a basso costo per la ricostruzione 3D online in ambienti subacquei. In particolare, il sistema proposto è costituito da una fotocamera standard, ovvero una fotocamera Stereolabs® ZED Stereo, un software dedicato ad-hoc e una custodia impermeabile realizzata con tecnologie di produzione additiva. Un primo prototipo del sistema è stato caratterizzato in ambiente controllato al fine di misurarne le prestazioni, in termini di accuratezza, e verificare poi se potesse essere idoneo per misure 3D e attività di ispezione svolte in ambiente subacqueo.

6.2 - Sviluppo del sistema e prototipazione

Come illustrato nella figura 6.1 (a), l'hardware è costituito dalla stereo-camera e dal suo alloggiamento impermeabile. In particolare, la custodia è stata progettata per ridurre al minimo gli ingombri e per essere utilizzata sott'acqua fino a 30 metri di profondità. La custodia si compone di due parti principali collegate mediante viti per consentire l'accesso alla stereo-camera quando si è fuori dall'acqua, mentre un passacavo è stato utilizzato per permettere il collegamento della stereo-camera tramite USB 3.0 a un PC esterno.

La stereo-camera è una telecamera stereo ZED di Stereolabs® [6.08]. Questa è una telecamera di profondità stereo passiva che riproduce il modo in cui funziona la visione umana utilizzando due telecamere RGB da 4 MP in grado di fornire immagini ad alta definizione fino a 2K a 15 fps con un FOV di 110°.

Al fine di adottare il modello fotogrammetrico standard evitando l'introduzione di errori sistematici [6.09] dovuti principalmente all'effetto di rifrazione all'interfaccia aria-acqua, la fotocamera è stata posizionata nella custodia su un supporto appositamente progettato per ridurre al minimo la distanza tra la sua lente d'ingresso ed il vetrino piatto (spessore 4 mm).

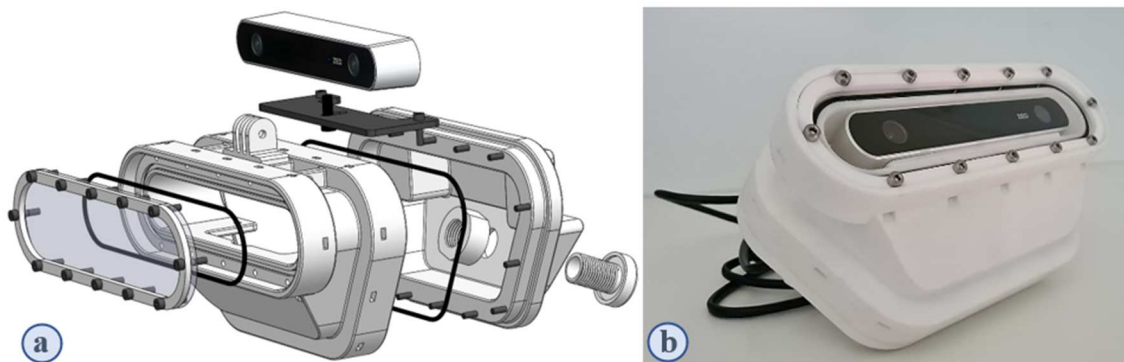


Figura 6.1 - Fotocamera di profondità con vista esplosa della sua custodia impermeabile (a) e del suo prototipo fisico (b).

Per quanto riguarda l'architettura software (figura 6.2), essa si compone di cinque moduli principali che partendo dalle immagini grezze acquisite consentono di eseguire attività di misurazione e ricostruzione 3D subacquee.

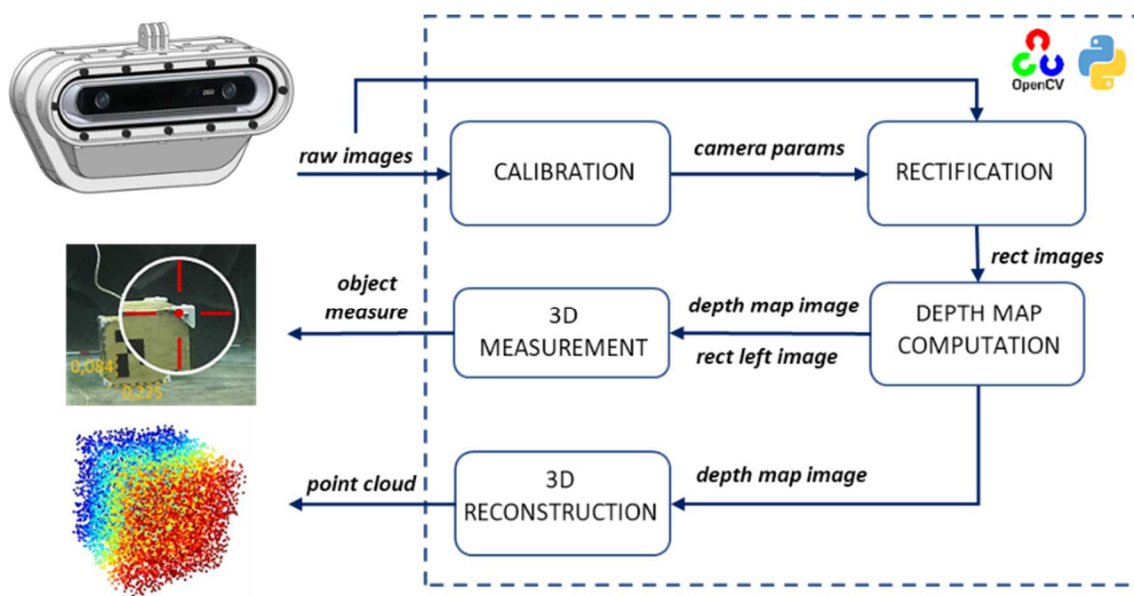


Figura 6.2 - Architettura SW.

I moduli software sono stati programmati in Python, integrando librerie OpenCV open source [6.10], per eseguire una ricostruzione di oggetti 3D online e misurazioni 3D della scena subacquea acquisita. In particolare, come illustrato nella figura 6.2, il software acquisisce coppie di immagini grezze dal dispositivo stereo, a un frame-rate definito, che vengono poi adottate per eseguire procedure di rettifica dell'immagine e stima della mappa di profondità prima di generare la nuvola di punti dell'oggetto 3D risultante. Per quanto riguarda il modulo di calibrazione, esso richiede coppie di immagini a scacchiera come input per calcolare i parametri di orientamento interno ed esterno (cioè parametri intrinseci ed estrinseci) [6.11] della fotocamera secondo il modello pinhole [6.12]. La procedura di calibrazione viene eseguita automaticamente implementando operazioni di filtraggio e una funzione *sub-pixel finder* [6.13] fornita da OpenCV che consente di rilevare punti con una precisione superiore alla dimensione del pixel. I punti rilevati sono quindi utilizzati per eseguire il processo di calibrazione della fotocamera basato

sull'algoritmo Bouguet [614], trovando i parametri di orientamento interno ed esterno della fotocamera, che vengono usati come input dal modulo di rettifica. Questo modulo corregge le immagini grezze, inviandole al modulo di calcolo della mappa di profondità per la generazione della mappa di profondità. In particolare, l'algoritmo semi-global block matching (SGBM) per la corrispondenza stereo [6.15] viene utilizzato per convertire le immagini rettificate sia a sinistra che a destra in una singola mappa di disparità. Per evitare rumori significativi, sono stati implementati un metodo WLS (Weighted Minimum Square) e un algoritmo di post-filtraggio guidato, forniti da OpenCV. In particolare, il filtro WLS consente di rimuovere i buchi nella mappa di disparità dovuti alle semi-occlusioni riempiendo l'immagine per ottenere una segmentazione uniforme tra i livelli, quindi viene applicato il filtro guidato per lo smoothing preservando i bordi dell'immagine. La mappa di disparità viene quindi riproiettata per generare la mappa di profondità, ottenendo il valore della profondità Z per ogni pixel dell'immagine. Infine, la mappa di profondità bidimensionale risultante e la corrispondente immagine RGB vengono inviate al modulo di ricostruzione 3D, per la generazione della nuvola di punti 3D desiderata, e al modulo di misurazione 3D che consente di calcolare la distanza 3D tra due punti scelti sulle immagini 2D. In particolare, le coordinate tridimensionali dei due punti vengono smussate utilizzando i punti più vicini in un'area circolare definita, quindi viene calcolato il punteggio z di ciascun punto per scartare i valori anomali e viene calcolata la media delle coordinate dei punti consolidati rimanenti. Il filtraggio dei punti selezionati dall'utente riduce l'imprecisione causata da una mappa di profondità non corrispondente a livello locale o da una selezione dei punti non precisa.

La figura 6.1 b mostra il prototipo del sistema di visione stereo subacqueo proposto. Prima di procedere con la prototipazione fisica, è stata eseguita un'analisi agli elementi finiti (FEA) per valutarne l'affidabilità e l'integrità nell'ambiente sottomarino. I risultati hanno dimostrato che il sistema mantiene una piena funzionalità, con un fattore di sicurezza minimo di 1,3 a una profondità di 30 m sotto il livello del mare, con una pressione dell'acqua di 0,4 MPa. La custodia è stata poi prodotta in materiale PA2200 tramite tecnologie SLS (Selective Laser Sintering) mediante una stampante EOS Formiga P110. Il lavoro di stampa ha richiesto 15 ore, e le parti sono state pulite utilizzando un media blaster.

6.3 - Analisi statica FEM sulla custodia impermeabile

Prima di procedere con la prototipazione fisica, la custodia impermeabile e la parte trasparente anteriore (flat-port) sono state sottoposte all'analisi statica con il metodo agli elementi finiti (FEM). A tale scopo è stato utilizzato il modulo Simulazione, integrato nel software SolidWorks. Il sistema è stato simulato nelle condizioni tipiche dell'ambiente subacqueo e ne sono state verificate l'affidabilità e l'integrità. L'analisi è stata condotta ipotizzando che il dispositivo raggiunga una profondità di 30 metri, pertanto è stato impostato un carico idrostatico di circa 0,4 MPa. Considerata la presenza di carichi idrostatici che aumentano con l'aumentare della profondità, sono state condotte due tipologie di valutazioni: verificare l'integrità delle parti e la deformazione delle sedi degli o-ring, che potrebbero causare la tenuta idrodinamica. L'assemblaggio è stato semplificato per ridurre l'onere computazionale. Per simulare correttamente il carico idrostatico, l'assieme è stato vincolato internamente (Fig. 6.3 a).

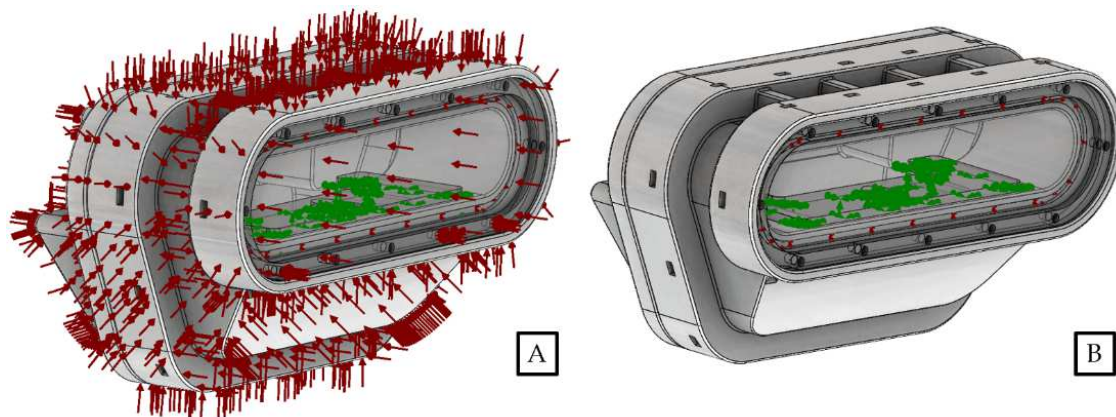


Fig. 6.3 – (A) Carichi idrostatici e vincoli sull'assieme, indicati rispettivamente con frecce rosse e verdi; (B) pressione concentrata per simulare l'effetto dell'o-ring, indicato con le frecce rosse.

Per simulare l'effetto del precarico delle viti sull'o-ring anteriore, è stata esercitata una pressione risultante sulla flat-port (Fig. 6.3 b). La poliammide PA2200 ($\sigma_y=39$ MPa) è stata assegnata ai componenti realizzati con tecnologia SLS, mentre il PMMA ($\sigma_y=70$ MPa) alla flat-port, a contatto con la stereocamera. Lo spessore minimo della custodia è di 4,5 mm, mentre per la flat-port è di 4 mm. In Fig. 6.4 è mostrato che il carico massimo è 30,88 MPa, determinato con i criteri di Von Mises. I risultati dimostrano che il fattore di sicurezza (FoS) dell'intero case è di circa 1,3, considerando che i componenti critici sono le parti realizzate in PA2200. Per la sola flat-port il FoS è 2.

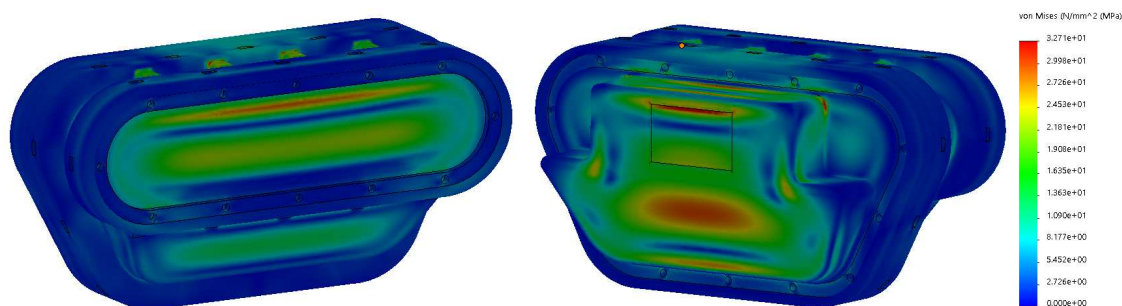


Figura 6.4 – Tensioni risultanti utilizzando il criterio di Von Mises.

Ulteriori analisi sono state effettuate per verificare l'efficacia della tenuta stagna che risente del campo di pressione a cui lavorano gli o-ring. La scelta degli o-ring e il progetto delle relative sedi sono stati effettuati adottando le regole di progettazione e seguendo la documentazione tecnica fornita dal produttore dell'o-ring. In Fig. 6.5 e Fig. 6.6 sono mostrate una sezione longitudinale ed una sezione trasversale sia in configurazione non deformata che deformata. Gli spostamenti lungo la coordinata Y, in prossimità delle sedi, sono contenuti rispetto alle tolleranze prescritte dal costruttore. Pertanto, i risultati dell'analisi mostrano che l'impermeabilità della custodia non è compromessa.

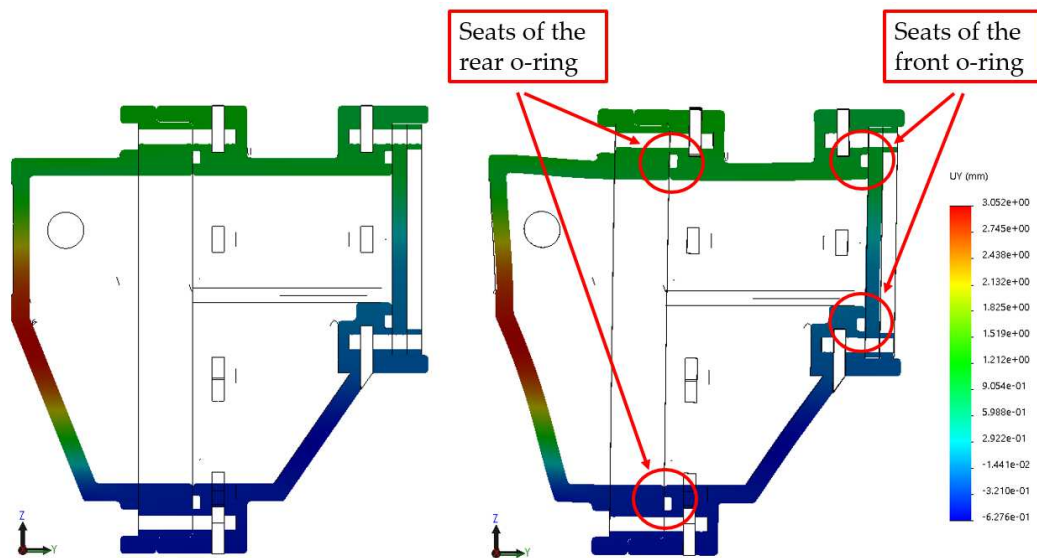


Figura 6.5 - Spostamenti su un piano longitudinale; sezioni non deformate e deformate sono presentate rispettivamente a sinistra e a destra.

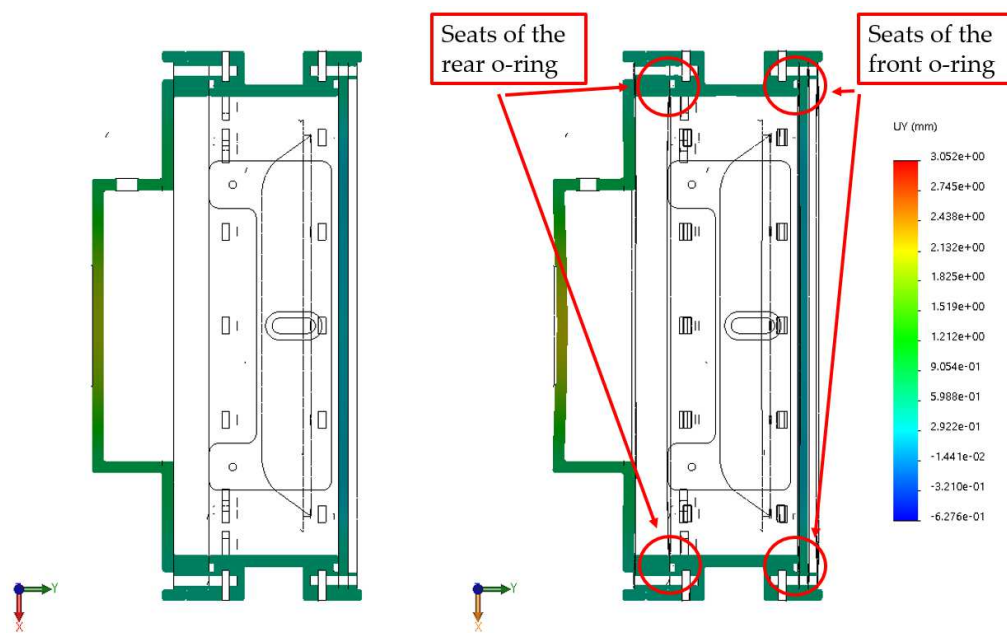


Figura 6.6 - Spostamenti su un piano trasversale; sezioni non deformate e deformate sono presentate rispettivamente a sinistra e a destra.

6.4 - Test sperimentali

Nei laboratori del DIMEG (Dipartimento di Ingegneria Meccanica, Energetica e Gestionale) dell'Università della Calabria è stata effettuata una prova sperimentale per valutare le prestazioni della telecamera. Il test è stato eseguito sia sott'acqua che in aria al fine di valutare il degrado delle prestazioni dovuto al mezzo acquoso. Come illustrato nella figura 6.7 a, il test subacqueo è stato eseguito in un ambiente controllato, ovvero una vasca circolare di 3 metri di diametro e 1 metro di profondità. Il sistema di visione stereo è montato su una staffa scorrevole, che gli consente di rimanere in una posizione fissa durante il processo di acquisizione delle immagini.

La fotocamera è collegata tramite USB 3.0 a un PC esterno per il debug del processo e le operazioni di post-elaborazione. Tutte le immagini durante i test sono state acquisite a una risoluzione di 2K pixel e 15 fotogrammi al secondo.

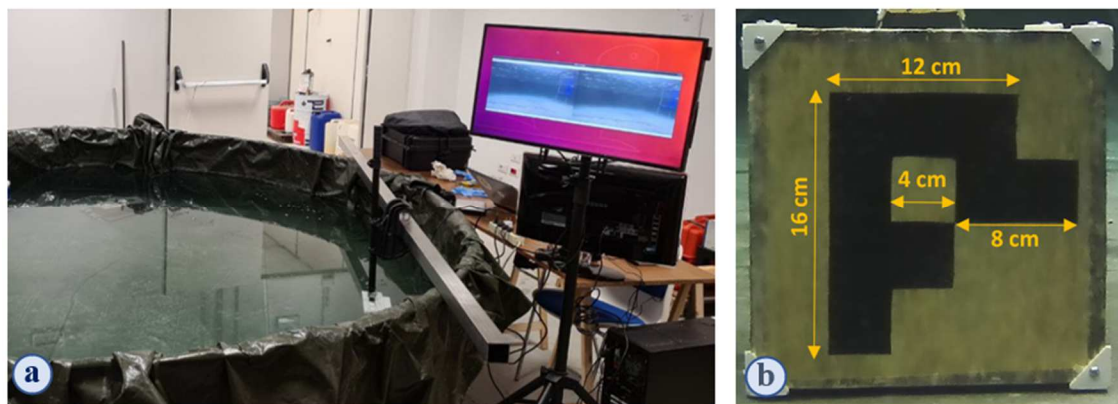


Figura 6.7 – Setup sperimentale (a) e oggetto cubico per il test di ricostruzione 3D (b).

In particolare, il test è stato effettuato utilizzando un oggetto cubico (fig. 6.7 b), con motivi neri costituiti da figure quadrate con lato lungo 4 cm, su ciascuna faccia.

La procedura adottata per la prova sperimentale, sia sott'acqua che in aria, è illustrata in figura 6.8. Sono state effettuate diverse acquisizioni, con diverse posizioni e orientamenti dell'oggetto, che sono state poi elaborate tramite il software sviluppato per eseguire la ricostruzione 3D. Per ciascuna delle quattro distanze di acquisizione, cioè 1 m, 1,5 m, 2 m e 2,5 m, il cubo è stato orientato con angoli di 0°, 25°, 50° e 75°.

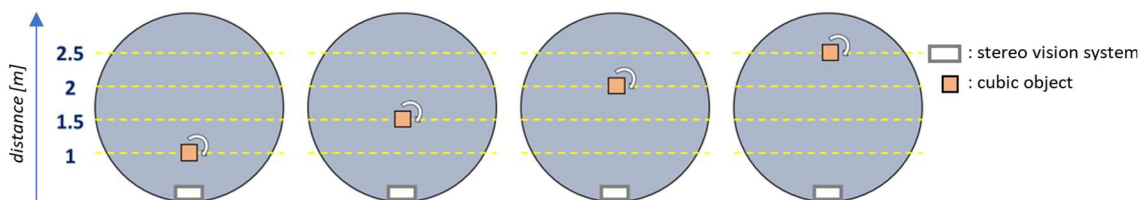


Figura 6.8 – Procedura dei test sperimentali.

Il processo di calibrazione è stato effettuato in ambiente sommerso mediante uno schema a scacchiera piatto. Cinquanta coppie di immagini della scacchiera sono state rilevate dal modulo di calibrazione e quindi elaborate automaticamente: è stata eseguita la posizione degli angoli interni della scacchiera e le informazioni sui parametri stereoscopici sono state salvate per la successiva fase di rettifica. I parametri risultanti della telecamera stereo sono stati utilizzati per eseguire la ricostruzione di oggetti 3D acquisendo, per ogni configurazione di target cubico, una coppia di immagini dal dispositivo stereo e quindi inviandola al software per l'elaborazione offline. In particolare, il modulo di rettifica ha preso come input i parametri di calibrazione e le immagini grezze acquisite per eseguire il processo di rettifica. Le immagini corrette sono state quindi trasmesse al modulo di calcolo della mappa di profondità, che ha eseguito la corrispondenza stereo per calcolare la mappa di disparità richiesta per la mappa di profondità. Quindi, la mappa di profondità risultante è stata utilizzata dal modulo di misurazione 3D per eseguire misurazioni 3D di oggetti.

La tabella seguente (tabella 6.1) mostra i risultati degli errori dell'analisi ottenuti per le misurazioni 3D eseguite sott'acqua e sopra l'acqua. In particolare, l'analisi dell'errore è stata

effettuata quantificando l'accuratezza, calcolata come percentuale dell'errore medio tra le misurazioni delle caratteristiche degli oggetti e le loro dimensioni reali, e l'errore assoluto medio (MAE).

Tabella 6.1 - Risultati della misurazione 3D per ogni distanza di acquisizione in condizioni subacquee e aeree.

Distance [mm]	Accuracy [%] (underwater)	MAE [mm] (underwater)	Accuracy [%] (air)	MAE [mm] (air)
1000	6.91	11.035	Disregarded	Disregarded
1500	5.06	5.217	2.37	3.385
2000	5.97	7.142	4.068	4.614
2500	5.73	6.121	3.43	4.385

I risultati mostrano che sott'acqua è possibile ottenere un errore medio inferiore a un centimetro per distanze superiori a un metro, arrivando a una precisione del 5% per oggetti a 1,5 metri. L'andamento dell'errore non segue una funzione monotona sulle distanze. Ciò è dovuto alla natura aleatoria dell'algoritmo del modulo di misurazione 3D utilizzato per ridurre l'imprecisione causata da una selezione del punto non precisa. L'algoritmo di misura fornisce errori accettabili per un intervallo di distanza operativa da 1,5 m a 2,5 m; ciò non è verificabile per una distanza operativa di 1 m, il che rende inefficiente l'algoritmo proposto. Pertanto, all'esterno dell'acqua, la precisione di misura del sistema viene calcolata alle stesse distanze a partire da 1,5 m; come mostrato nella tabella precedente, l'accuratezza della misura migliora notevolmente. La figura seguente (fig. 6.9 a) mostra la nuvola di punti dell'oggetto risultante calcolata dal modulo di ricostruzione 3D per la condizione subacquea.

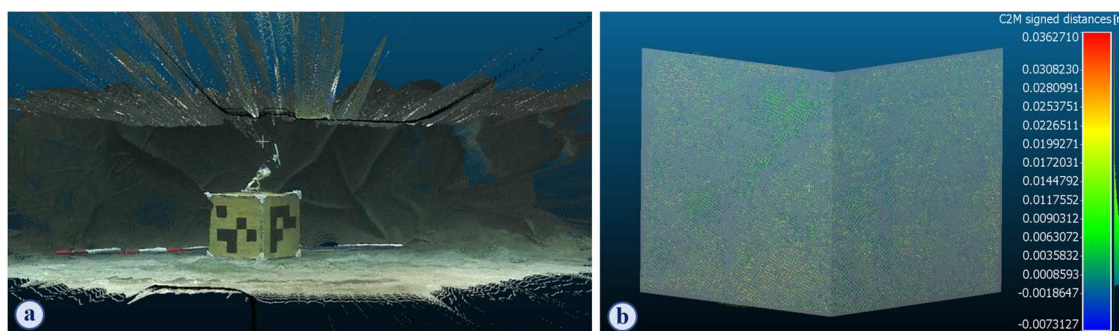


Figura 6.9 - Ricostruzione di oggetti 3D subacquei (a) e scala cromatica del campo scalare risultante che rappresenta l'errore di distanza in metri (b).

Tale nuvola di punti è stata poi confrontata con un modello di riferimento, ovvero un modello CAD con le dimensioni reali dell'oggetto cubico, per verificare le prestazioni di ricostruzione 3D del sistema proposto. L'errore di ricostruzione è stato misurato tramite il software open source CloudCompare [6.16]. Il processo di allineamento è stato eseguito selezionando coppie di punti equivalenti in entrambe le entità. La scala di colori del campo scalare risultante (fig. 6.9 b) dell'errore di distanza tra la nuvola di punti generata e l'oggetto mesh di riferimento, e la relativa distribuzione (fig. 6.10 a), vengono infine calcolate. Si ottiene un errore medio di 5,5 millimetri sott'acqua, convalidando l'accuratezza della nuvola di punti generata.

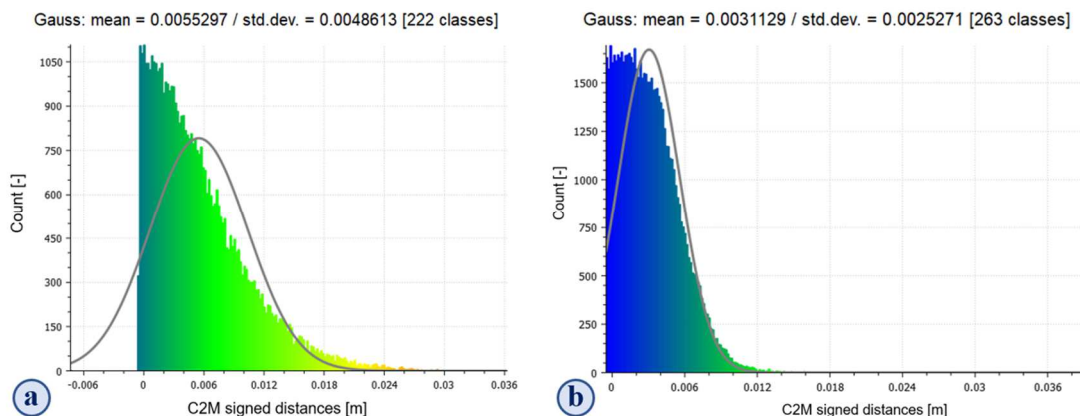


Figura 6.10 - Distribuzione di Gauss che rappresenta l'errore di distanza in metri in condizioni sott'acqua (a) e in aria (b).

La figura 6.10 (b) mostra la distribuzione dell'errore calcolata in aria; in particolare, la qualità della ricostruzione 3D migliora notevolmente con minime distorsioni planari e un errore medio di 3,12 millimetri. Sulla base di questi risultati, è possibile evidenziare che il sistema proposto è adatto per attività di misurazione 3D di oggetti e ricostruzione 3D da svolgere sott'acqua per applicazioni industriali e scientifiche. Ad esempio, il sistema proposto potrebbe essere montato su ROV e AUV per attività di ispezione di condotte, con una distanza operativa fino a 3 metri a causa della torbidità dell'acqua. Oppure può essere adottato per la documentazione e ricostruzione 3D di ambienti sommersi per applicazioni in campo ambientale e archeologico.

6.5 - Riepilogo

In questo capitolo è stata testata una soluzione a basso costo per la ricostruzione 3D subacquea adoperando una fotocamera di profondità standard. In particolare, è stato sviluppato un software dedicato per l'elaborazione online utilizzando librerie open-source. È stato condotto un test di laboratorio in ambiente controllato per caratterizzare il sistema proposto e quantificarne le prestazioni in termini di accuratezza della misurazione, variando la distanza relativa tra la telecamera e gli oggetti inquadrati. È stata condotta un'analisi quantitativa: l'accuratezza e l'errore medio assoluto (MAE) sono stati impiegati nella valutazione dell'oggetto ricostruito in 3D e sono stati forniti i risultati dell'analisi dell'errore, ottenendo l'errore più basso per una distanza operativa di 1,5 m. Gli sviluppi futuri si concentreranno su miglioramenti per rafforzare la procedura di calibrazione e ulteriori sperimentazioni saranno effettuate in mare in condizioni operative reali.

Conclusioni

La ricerca svolta nell'ambito del Dottorato ha permesso di migliorare alcuni dei più innovativi strumenti per l'archeologia subacquea, impiegando sistemi di localizzazione acustica, veicoli robotici e tablet subacquei per supportare il complesso lavoro di ispezione, documentazione e monitoraggio di siti archeologici sommersi da parte degli archeologi subacquei, incrementando il livello generale di sicurezza per gli stessi.

È stata svolta un'analisi sui sistemi di localizzazione acustica subacquea al fine di migliorare le performance di geo-localizzazione di un target subacqueo. Lo studio ha previsto l'utilizzo di sistemi per la localizzazione acustica di tipo SBL (short baseline) ed USBL (ultra-short baseline), di veicoli autonomi di superficie (ASV) e a controllo remoto subacquei (ROV), e di un tablet subacqueo. La comunicazione tra i diversi dispositivi è stata resa possibile attraverso lo sviluppo di un software in ambiente ROS (*Robot Operating System*) che ha riguardato l'implementazione e la validazione di un *package*; è stata inoltre sviluppata una *webapp* di controllo per l'attuazione dei comandi ed il monitoraggio dei parametri tecnici di interesse. Diverse sono state le analisi condotte in ambiente simulativo attraverso lo sviluppo di modelli statici e dinamici in ambiente *Matlab/Simulink*. Differenti soluzioni hanno visto l'integrazione dei sistemi acustici con sensori inerziali e camere subacquee, con l'obiettivo di migliorare il dato di posizione attraverso tecniche di sensor-fusion, in particolare di *visual-inertial odometry*, spesso rese possibili grazie all'utilizzo di un filtro di Kalman. Esteso data la non linearità del modello.

I punti di forza dell'architettura in analisi sono l'integrazione del sistema di navigazione sul tablet subacqueo che fornisce all'utilizzatore delle preziose informazioni sull'immersione e sulla sua localizzazione; il sistema SBL utilizzato presenta, infatti, una configurazione a funzionamento inverso che permette al ricevitore acustico di calcolare la sua posizione e di fornirla direttamente al tablet subacqueo; in questo modo il sub è in grado di orientarsi localizzandosi su una mappa visualizzabile sul tablet subacqueo.

Si sono indagate, inoltre, diverse strategie per il miglioramento della posizione, sia dal punto di vista hardware sia software attraverso lo sviluppo di algoritmi, che a seconda della configurazione hanno fornito una riduzione degli errori sulla localizzazione. È opportuno sottolineare l'importanza del ruolo svolto dal veicolo autonomo di superficie ASV sotto il quale è stata montata la base acustica (trasmettitore): esso è in grado di ricoprire il ruolo di hub di comunicazione tra il mondo sottomarino e quello terrestre; esso segue il sub durante la sua immersione, facendo sì che l'area di lavoro sia teoricamente estendibile all'infinito con il vantaggio di avere sempre una robusta comunicazione tra base acustica e ricevitore acustico, e quindi un dato affidabile di geo-localizzazione fino ad una certa profondità.

Le criticità emerse hanno riguardato l'accuratezza del sistema acustico nella stima delle distanze. Infatti, così come studiato in simulazione, l'impatto che tale errore ha sul calcolo della posizione non è per nulla trascurabile e potrebbe rappresentare un punto di partenza di studio per sviluppi futuri del progetto.

Successivi sviluppi del lavoro potrebbero riguardare ulteriori analisi sull'integrazione dei sistemi di localizzazione acustica con stereo-camere subacquee, adoperando un filtro di Kalman esteso per pervenire a dei risultati migliori in termini di posizionamento. In particolare, nella tesi è presentata un'analisi di ricostruzione 3D in ambiente subacqueo condotta per mezzo di una stereo-camera commerciale per applicazioni terrestri, che ha restituito risultati interessanti.

L'impiego di filtri particellari, ai fini della localizzazione, potrebbe essere un altro punto interessante da considerare, a patto che si abbia una batimetria del fondale marino già acquisita,

attraverso la quale poter eseguire gli algoritmi di localizzazione in tempo reale. È interessante anche pensare come l'impiego di DVL (*Doppler Velocity Logger*) possa andare a migliorare la precisione del dato di posizionamento, fornendo al filtro di Kalman informazioni sulla velocità. Anche aumentare la baseline dei trasduttori montati a croce sotto il veicolo potrebbe certamente fornire dei risultati migliori, così come riscontrato più volte in simulazione, pertanto uno degli interventi possibili da effettuare potrebbe essere quello di trovare delle soluzioni hardware per estendere la baseline senza compromettere la dinamica del veicolo, e ridurre le oscillazioni prodotte dalle onde.

Infine, un altro possibile sviluppo futuro potrebbe riguardare l'utilizzo di una unità IMU esterna, di qualità nettamente superiore a quella integrata nel tablet, montata ad esempio sulle bombole del sub, ed inserita all'interno di un sistema stabilizzante; in questo modo l'integrazione delle accelerazioni per ottenere le velocità (o le posizioni) da veicolare nel filtro di Kalman Esteso potrebbe fornire dei dati più accurati e quindi affetti da meno rumore, cosa che invece avviene se si considerano le accelerazioni del tablet subacqueo che è soggetto a ripetuti movimenti bruschi e vibrazioni durante il suo utilizzo in acqua.

Lo sviluppo di questo lavoro di tesi ha previsto anche delle attività sperimentali, quindi dei test reali sull'intero sistema, differenziandosi da molti lavori presenti in letteratura che hanno affrontato la problematica soltanto proponendo delle idee oppure affrontando il problema solo in simulazione. I test condotti, seppur non abbiano portato a dei risultati definitivi, hanno seguito dei protocolli abbastanza rigidi e ben strutturati, pensati per essere sfruttati anche per eventuali test futuri.

Il contesto applicativo che interesserà i risultati di questo lavoro di tesi riguarda la creazione di parchi acquatici archeologici sommersi; all'interno di essi, tanti sub equipaggiati da tablet e ricevitore acustico potranno apprezzare le meraviglie del passato, navigando in totale autonomia per mezzo di una mappa di navigazione presente sul tablet stesso. Un'accurata geolocalizzazione permetterà di avere informazioni precise sulla storia del sito archeologico, che saranno visualizzabili direttamente sul display del tablet subacqueo dopo aver raggiunto uno specifico punto di interesse. Il passo successivo riguarderà l'utilizzo delle tecniche di realtà aumentata per visualizzare sul tablet ricostruzioni di reperti archeologici distrutti o deteriorati dal tempo e dalle acque, e per continuare a sviluppare tali tecnologie è necessario una geolocalizzazione molto precisa.

Bibliografia e sitografia

- [I.01] Denoble, P. J., Caruso, J. L., Dear, G. de L., Pieper, C. F., Vann, R. D.: Common causes of open-circuit recreational diving fatalities. In: *Undersea & Hyperbaric Medicine* Bethesda, vol. 35, no. 6, pp. 393-406 (2008).
- [I.02] Cardia, C., Gjanci, P., Petrioli, C., Saturni, G., Spaccini, D., Tomaselli, D.: The Internet of Underwater Things: From Nemo to underwater Whatsapp. Conference: *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 409-410. (2019).
- [I.03] Scaradozzi, D., Zingaretti, S., Ciuccoli, N., Costa, D., Palmieri, G., Bruno, F., Ritacco, G., Cozza, M., Raxis, P., Tzifopanopoulos, T.: Lab4Dive Mobile Smart Lab for Augmented Archaeological Dives. In: *IOP Conference Series Materials Science and Engineering*, vol. 364, no. 1 (2018).
- [I.04] Bernardi, M., Cardia, C., Gjanci, P., Monterubbiano, A., Petrioli, C., Picari, L., Spaccini, D.: The Diver System: Multimedia Communication and Localization Using Underwater Acoustic Networks. Conference: *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"* pp. 1-8 (2019).
- [I.05] Reis, J., Morgado, M., Batista, P., Oliveira, P., Silvestre, C.: Design and Experimental Validation of a USBL Underwater Acoustic Positioning System. In: *Sensors*, vol. 16, no. 9 (2016).
- [I.06] Bruno, F., Barbieri, L., Lagudi, A., Mangeruga, M., Pupo, F., Casavola, A.: A Cooperative Monitoring System for Diver Global Localization and Operation Support. In book: *Design Tools and Methods in Industrial Engineering, Proceedings of the International Conference on Design Tools and Methods in Industrial Engineering*, pp. 410-421 (2019).
- [I.07] Bruno, F., Barbieri, L., Mangeruga, M., Cozza, M., Lagudi, A., Čejka, J., Liarokapis, F., Skarlatos, D.: Underwater augmented reality for improving the diving experience in submerged archaeological sites. In: *Ocean Engineering*, vol. 190, pp. 106487 (2019).
- [I.08] Miskovic, N., Nad, D., Rendulic, I.: Tracking Divers: An Autonomous Marine Surface Vehicle to Increase Diver Safety. In: *Robotics & Automation Magazine*, vol. 22, pp. 72-84. (2015)
- [I.09] Vasilijevic, A., Nad, D., Miskovic, N.: Autonomous Surface Vehicles as Positioning and Communications Satellites for the Marine Operational Environment – Step toward Internet of Underwater Things. Conference: *2018 IEEE 8th International Conference on Underwater System Technology: Theory and Applications*, pp. 1-5 (2018).

-
- [I.10] Miskovic, N., Pascoal, A., Bibuli, M., Caccia, M., Neasham, J., Birk, A., Egi, S., Grammer, K., Marroni, A., Vasilijevic, A., Nad, D., Vukić, Z.: CADDY project, year 3: The final validation trials. In: OCEANS 2017, pp. 1-5 (2017).
- [I.11] www.imareculture.eu
- [I.12] www.diveinhistory.com
- [I.13] www.interreg-med.eu
- [I.14] https://labust.fer.hr/zari/labust/research/projects/ongoing_projects/cromarx
- [I.15] www.caddy-fp7.eu
- [I.16] Bruno, F., Lagudi, A., Muzzupappa, M., Lupia, M., Cario, G., Barbieri, L., Passaro, S., Saggiomo, R.: Project VISAS: Virtual and Augmented exploitation of submerged archaeological sites—Overview and first results (2016). In: Marine Technology Society Journal vol. 50, no. 4, pp. 119-129 (2016).
- [I.17] Stojanovic, M., Preisig J.: Underwater acoustic communication channels: Propagation models and statistical characterization. In: IEEE Communications Magazine, vol. 47, no. 1, pp. 84-89 (2009).
- [1.01] Gode T., Woolsey, C. A., Stilwell, D. J, Wicks, A. L.: Long Baseline Ranging Acoustic Positioning System. Master thesis in Electrical and Computer Engineering at Virginia Polytechnic Institute, February 2015.
- [1.02] Groves, P. D. In book: Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems (second edition).
- [1.03] Kamila, N. K. In book: Handbook of Research on Wireless Sensor Network Trend, Technologies, and Applications (2017) IGI Global, India.
- [1.04] <https://www.slideshare.net/codevintec/gli-strumenti-50199875>
- [1.05] [https://it.wikipedia.org/wiki/Trieste_\(batiscafo\)#Altre_immersioni_del_Trieste](https://it.wikipedia.org/wiki/Trieste_(batiscafo)#Altre_immersioni_del_Trieste)
- [1.06] http://www.wikiwand.com/en/Underwater_acoustic_positioning_system
- [1.07] Tan, H. P., Diamant, R., Seah, W. K. G., Waldmeyer, M.: A survey of techniques and challenges in underwater localization. In: Ocean Engineering, vol. 38, no. 14–15, pp. 1663-1676 (2011).
- [1.08] <http://craigmooreantarctica.blogspot.it/2009/10/scini.html>
- [1.09] E. Carnevale, Sensori e trasduttori per industria e automazione (2[^]ED).
- [1.10] Evologics, <https://www.evologics.de/en/products/acoustics/index.html>
-

-
- [1.11] http://www1.na.infn.it/fisicainbarca/GPS_INFN.pdf
- [1.12] Navtech, <https://www.navtechgps.com/>
- [1.13] Advanced Navigation, <http://www.advancednavigation.com.au/>
- [1.14] Xsens, <https://www.xsens.com/>
- [1.15] Mackenzie, K.V.: Nine-term equation for the sound speed in the oceans. In: The Journal of the Acoustical Society of America, vol. 70, no. 3, pp 807-812 (1981).
- [1.16] Coppens, A. B.: Simple equations for the speed of sound in Neptunian waters. In: The Journal of the Acoustical Society of America, vol. 69, no. 3, pp. 862-863 (1981).
- [1.17] Chen, C. T., Millero, F. J.: Speed of sound in seawater at high pressures. In: The Journal of the Acoustical Society of America, vol. 62, no. 5, pp. 1129-1135 (1977).
- [1.18] Brinker, R. C. In book: The Surveying Handbook, Springer Science Business Media, LLC (1987).
- [2.01] Bruno, F., Mišković, N., Nađ, Đ., Kapetanović, N., Lagudi A., Aiello, R., Lupia, M., Cario, G.: New technologies for improving the diver experience in underwater cultural sites. In: Proceedings of the International Conference on Archaeology and Tourism Sense and sustainability, pp. 1-6 (2019)
- [2.02] Microsemi, "Quantum SA.45s Chip Scale Atomic Clock", http://www.microsemi.com/document-portal/doc_download/133305_quantum-sa_45scsac, [Online; accessed February 2017].
- [2.03] Cario, G., Casavola, A., Djapic, V., Gjanci, P., Lupia, M., Petrioli, C., Spaccini, D.: Clock Synchronization and Ranging Estimation for Control and Cooperation of Multiple UUVs. In: Proceedings of MTS/IEEE OCEANS, pp. 1-9 (2016).
- [2.04] Miskovic, N., Triska, Z., Nad, D., Vukic, Z.: Guidance of a small-scale overactuated marine platform - Experimental results. In: MIPRO 2011 Proceedings of the 34th International Convention, pp. 684-689 (2011).
- [3.01] Cario, G., Casavola, A., Gagliardi, G., Lupia, M., Severino, U., Bruno, F.: Analysis of error sources in underwater localization systems. In: OCEANS 2019, pp. 1-6 (2019).
- [3.02] Cario, G., Casavola, A., Gagliardi, G., Lupia, M., Severino, U.: Accurate Localization in Acoustic Underwater Localization Systems. In: Sensors, vol. 21, no. 3 (2021).
- [3.03] <https://www.advancednavigation.com/solutions/spatial/>
- [3.04] Bruno, F., Barbieri, L., Mangeruga, M., Cozza, M., Lagudi, A., Čejka J., Liarokapis, F., Skarlatos, D.: Underwater augmented reality for improving the diving experience in submerged archaeological sites. In: Ocean Engineering, vol. 190, pp. 106487 (2019).
-

-
- [4.01] www.leafletjs.com
- [5.01] Neasham, J. A., Goodfellow G., Sharpouse, R.: Development of the “Seatrac” miniature acoustic modem and USBL positioning units for subsea robotics and diver applications. In: *OCEANS 2015*, , pp. 1-8 (2015).
- [5.02] Song H. Y., Lee J. S.: Detecting Positioning Errors and Estimating Correct Positions by Moving Window. In: *PLoS ONE*, vol. 10, no. 12 (2015).
- [6.01] Bruno, F., Bianco, G., Muzzupappa, M., Barone, S., Razionale, A. V.: Experimentation of structured light and stereo vision for underwater 3D reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing* 66(4), 508-518 (2011).
- [6.02] Bruno, F., Lagudi, A., Barbieri, L., Rizzo, D., Muzzupappa, M. and De Napoli, L.: Augmented reality visualization of scene depth for aiding ROV pilots in underwater manipulation. In: *Ocean Engineering*, vol. 168, pp.140-154 (2018).
- [6.03] Vitali, A., Regazzoni, D., Rizzi, C., Lupi, G.: Low Cost Markerless Motion Capture Systems: A Comparison Between RGB Cameras and RGB-D Sensors. In: *ASME International Mechanical Engineering Congress and Exposition*, vol. 84539. American Society of Mechanical Engineers (2020).
- [6.04] Digumarti, S. T., Chaurasia, G., Taneja, A., Siegart, R., Thomas, A., Beardsley, P.: Underwater 3D capture using a low-cost commercial depth camera. In: *IEEE Winter Conference on Applications of Computer Vision (WACV) 2016*, pp. 1-9. IEEE (2016).
- [6.05] Andrew, A.M.: *Multiple View Geometry in Computer Vision*, Kybernetes, pp. 1333-1341 (2001).
- [6.06] Wang, Y., Tang, C., Cai, M., Yin, J., Wang, S., Cheng, L., Tan, M.: Real-Time Underwater Onboard Vision Sensing System for Robotic Gripping. *IEEE Transactions on Instrumentation and Measurement* 70, 1-11 (2020).
- [6.07] Wang, C., Zhang, Q., Lin, S., Li, W., Wang, X., Bai, Y., Tian, Q.: Research and experiment of an underwater stereo vision system. In: *OCEANS Marseille 2019*, pp. 1-5. IEEE (2019).
- [6.08] Stereolabs Homepage, <https://www.stereolabs.com/zed/>, last acceded 2021/03/13
- [6.09] Menna, F., Nocerino, E., Remondino, F.: Photogrammetric modelling of submerged structures: influence of underwater environment and lens ports on three-dimensional (3D) measurements. *Latest developments in reality-based 3D surveying and modelling*. MDPI Basel, Switzerland, pp. 279-303 (2018).
- [6.10] OpenCV Homepage, <https://opencv.org/>, last acceded 2021/03/13.
- [6.11] Granshaw, S.: *Photogrammetric Terminology*. 3rd edn. *The Photogrammetric Record*, pp. 210-252 (2016).
-

- [6.12] Forsyth, D. A., Ponce, J.: Computer vision: a modern approach. 2nd edn. Pearson (2012).
- [6.13] Forstner, M.A., Gulch, E.: A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centers of Circular Features. In: Proceedings of the ISPRS Intercommission Conference on Fast Processing of Phonogrammic Data, pp. 281-305, Interlaken (1987).
- [6.14] Zhang, Z.: A flexible new technique for camera calibration. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334. IEEE (2000).
- [6.15] Hirschmuller, H.: Stereo Processing by Semiglobal Matching and Mutual Information. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pp. 328-341. IEEE (2008).
- [6.16] CloudCompare, <http://www.cloudcompare.org/>, last acceded 2021/03/13.