

**PAOLO ANTONIO FICHERA**

Dipartimento Tecnologie Energetiche  
Smart Energy  
Intelligenza Distribuita e Robotica per l'Ambiente e la Persona  
Centro Ricerche Casaccia

# VENUS

Veicolo per Navigazione Subacquea e Sorveglianza  
Sistema di Controllo – Prove Sperimentali

RT/2020/9/ENEA



AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE,  
L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

PAOLO ANTONIO FICHERA

Dipartimento Tecnologie Energetiche  
Smart Energy  
Intelligenza Distribuita e Robotica per l'Ambiente e la Persona  
Centro Ricerche Casaccia

# VENUS

Veicolo per Navigazione Subacquea e Sorveglianza  
Sistema di Controllo – Prove Sperimentali

RT/2020/9/ENEA



AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE,  
L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

## Ringraziamenti

Al Comando e al personale del Comando Aeroporto Vigna di Valle,  
che hanno ospitato e supportato le prove di Venus nel lago di Bracciano

Ai colleghi del Laboratorio di Robotica,  
che hanno condiviso - e sopportato - le tante 'prove al lago'

*Rem tene, verba sequuntur*

I rapporti tecnici sono scaricabili in formato pdf dal sito web ENEA alla pagina [www.enea.it](http://www.enea.it)

I contenuti tecnico-scientifici dei rapporti tecnici dell'ENEA rispecchiano  
l'opinione degli autori e non necessariamente quella dell'Agenzia

The technical and scientific contents of these reports express the opinion  
of the authors but not necessarily the opinion of ENEA.

## **VENUS**

Veicolo per Navigazione Subacquea e Sorveglianza  
Sistema di Controllo – Prove Sperimentali

Paolo Antonio Fichera

### **Riassunto**

Venus è un robot autonomo interamente sviluppato in ENEA – DTE-SEN-IDRA per missioni di esplorazione/monitoraggio/sorveglianza subacquea. Dopo una breve descrizione generale della struttura del veicolo e degli scopi per cui è stato pensato viene descritta l'architettura software del Sistema di Controllo, basata sul Robot Operating System. Il Controllo è organizzato in blocchi indipendenti (nodi) e intercomunicanti. Vengono descritti i singoli nodi e le strutture dati di I/O che ne consentono l'interazione, garantendo al contempo la modularità del Sistema. Viene quindi illustrata l'Interfaccia Uomo Macchina sviluppata per la teleguida e la gestione delle missioni. Segue il resoconto delle prove sperimentali e dei risultati conseguiti. Le prove sono state effettuate nel lago di Bracciano, con il supporto dell'Aeronautica Militare. Sono infine discusse le criticità emerse e i possibili sviluppi del progetto.

**Parole chiave:** Robotica autonoma, Robotica subacquea, AUV, Sistema di controllo, PID.

### **Abstract**

*Venus is an autonomous underwater robot fully developed at ENEA – DTE-SEN-IDRA to carry out search/monitoring/surveillance tasks. A brief description of the vehicle and of its purposes is given. The software architecture of the Control System is described. The Control is based on Robot Operating System and it is built of independent, communicating blocks (nodes). Individual nodes are described along with I/O data structures that implement intercommunication, assuring at the same time System modularity. The Man Machine Interface is then explained, which was developed to allow robot remote control and tasks management. The report of experimental tests and of achieved outcomes follows. The tests were carried out in Bracciano lake, with the support of Italian Air Force. Criticalities and future developments are lastly discussed.*

**Keywords:** Autonomous robots, Underwater robotics, AUV, Control system, PID.



# INDICE

1. Introduzione	7
2. Struttura HW	9
2.1 Cenni sulla struttura del veicolo	9
2.2 Unità di calcolo	10
2.3 Comunicazione	10
2.4 Sensori	10
2.5 Attuatori	10
2.6 Stazione di Controllo	11
3. Controllo - Architettura SW	12
3.1 ROS	13
3.2 Schema dell'architettura - definizione nodi	13
3.3 Sistema di Controllo	15
3.4 Registrazione (log) dati	18
4. Descrizione nodi - tabelle I/O	20
4.1 GPS	20
4.2 Sonar	21
4.3 Video	24
4.4 Listener	26
4.5 Ottico	27
4.6 Bussola	29
4.7 Pressostato	30
4.8 IMU	32
4.9 Listener	34
4.10 Navigatore	36
4.11 PID	40
4.12 Motore	44
4.13 Timoni	45
5. Interfaccia uomo-macchina	46
6. Collaudi – prove di navigazione	51
6.1 Navigazione GPS	52
6.2 Navigazione in quota	56
6.3 Navigazione sul fondo	63
7. Criticità e miglioramenti	70
8. Appendici	72
9. Bibliografia	74



## 1. Introduzione

Venus – Veicolo per la Navigazione Subacquea e la Sorveglianza – è un robot autonomo sottomarino interamente progettato e realizzato presso l'Unità DTE-SEN-IDRA dell'ENEA – Centro Ricerche Casaccia.

Sono attualmente operanti due prototipi, denominati Venus 2 e Venus 3, entrambi collaudati in navigazione.

Gli impieghi – effettivi e potenziali - dei robot sottomarini sono ben noti, spaziando dalla ricerca relitti all'ispezione/sorveglianza di strutture/manufatti sommersi, alla rilevazione del profilo dei fondali, al monitoraggio di inquinanti, etc.

Altrettanto nota è la differenza fra ROV (Remotely Operated Vehicles), robot teleguidati mediante cordoni ombelicali, lunghi anche diverse centinaia di metri, e AUV (Autonomous Underwater Vehicles), robot completamente autonomi, che non richiedono la presenza, nelle immediate vicinanze, di una stazione di guida/alimentazione (tipicamente una nave appoggio). Senza volersi dilungare sui vantaggi e sugli svantaggi di ciascuna categoria, si nota soltanto che per missioni di una certa complessità la presenza del cavo ombelicale dei ROV è un ostacolo insormontabile; d'altra parte la navigazione in autonomia di un veicolo subacqueo presenta problematiche notevolmente superiori.

Una linea di pensiero teorizza l'impiego di sciame (swarm) di robot autonomi: gruppi più o meno numerosi - a seconda del particolare task – di veicoli identici, caratterizzati da una bassa complessità realizzativa (quindi da un basso costo) e cooperanti secondo regole ispirate in parte al comportamento sociale di talune specie animali. Il vantaggio di disporre di più AUV operanti in parallelo appare evidente per determinati tipi di missione, come ad esempio l'esplorazione di vaste aree di fondale, mentre la particolare modalità di interazione dovrebbe aumentare l'efficienza delle operazioni.

Venus è stato originariamente pensato per operare come elemento di uno sciame. Da qui la particolare cura posta nel progettare una macchina dai costi contenuti, pur dotandola, naturalmente, di tutte le funzionalità necessarie alla navigazione in sciame, e in particolare di un efficace sistema di comunicazione subacquea, essenziale per un tipo di controllo basato su un continuo scambio di informazioni fra i componenti del gruppo. Un avanzato sistema di comunicazione ottica è in fase di sviluppo, parallelamente al veicolo, presso l'Unità di Robotica - DTE-SEN-IDRA.

Nel corso dello sviluppo ci si è resi conto della necessità di dotare il robot della capacità di operare in totale autonomia, e del notevole valore aggiunto che una tale soluzione comporta.

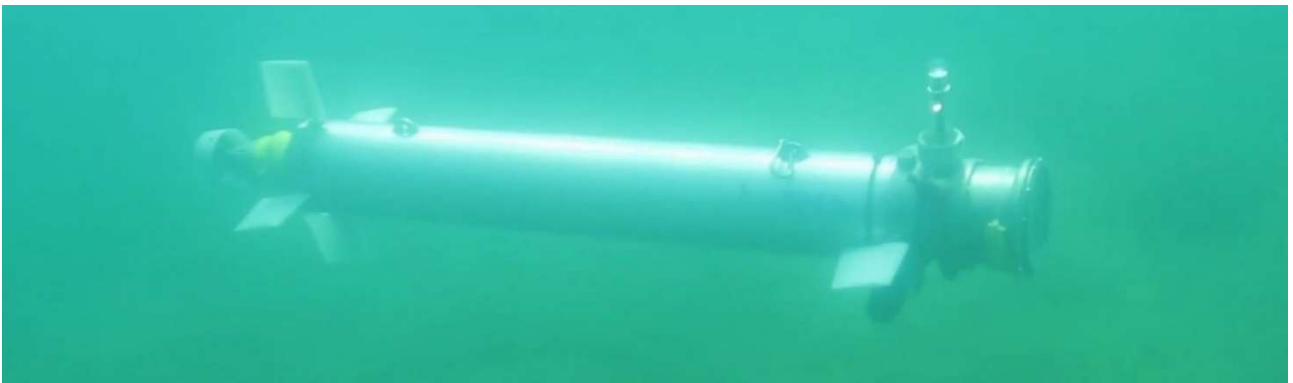
Il Venus attuale è il risultato di questa evoluzione. È in grado, come si vedrà, di svolgere autonomamente elementari missioni di navigazione/esplorazione, la cui combinazione consente in prospettiva l'esecuzione dei compiti complessi per cui è stato pensato.

Con la sensoristica attualmente ospitata a bordo (§2.4) possono essere svolte missioni di navigazione a punto prestabilito (GPS), profilazione fondale (sonar) ed esplorazione (videocamera). Un'altra importante categoria



di missioni per cui Venus è concepito, il monitoraggio dello stato delle acque, potrebbe essere testata semplicemente installando gli opportuni sensori.

Nel seguito, dopo un breve richiamo alla struttura hardware (nel seguito: HW), verrà discusso il Sistema di Controllo implementato e i relativi aspetti di architettura software (nel seguito: SW). Verranno quindi illustrate le prove di navigazione in acque libere effettuate nel lago di Bracciano, per fare infine il punto sullo stadio attuale e sulle prospettive di sviluppo di Venus.



**Figura 1.1 – Venus in immersione nel lago di Bracciano**

## 2. Struttura HW

La struttura elettro-meccanica di Venus è descritta in [1] - progetto meccanico originale - e in [2] - descrizione della struttura meccanica e dell'elettronica di bordo.

Di seguito verrà data, per comodità di lettura, una panoramica degli elementi costituenti Venus e che verranno richiamati nei capitoli successivi. Per una trattazione dettagliata, si veda [2].

### 2.1 Cenni sulla struttura del veicolo

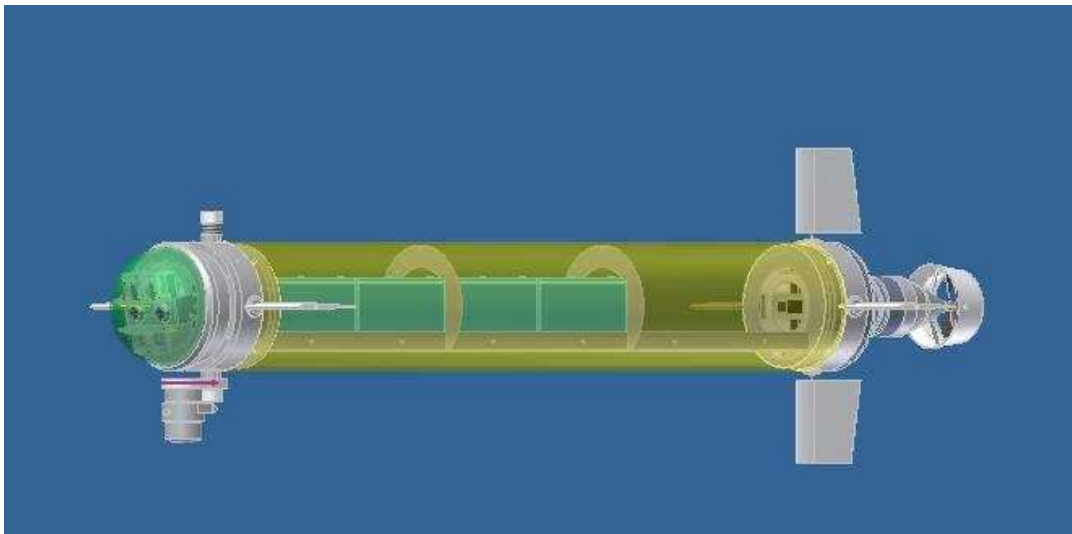


Figura 2.1 – Struttura di base

Ai fini della trattazione seguente, Venus può essere schematicamente suddiviso in tre sezioni: Testa, Coda e corpo centrale. Tutte in lega d'alluminio, le prime due sono composte da elementi realizzati da macchine a controllo numerico, mentre il corpo è ricavato, in accordo con i criteri di economicità discussi nell'Introduzione, da una tubazione industriale standard.

L'alloggiamento dell'elettronica di controllo è suddiviso fra la Testa e la Coda; quest'ultima ospita anche il sistema di propulsione, mentre nel corpo centrale trovano posto le batterie e i cavi di connessione tra i differenti sottosistemi.

Venus viene governato mediante due tipi di attuatori: un motore (propulsione) e sei timoni (profondità e direzione), disposti a coppie nelle sezioni di Testa e di Coda.

Sono operanti due coppie di timoni di profondità (anteriori e posteriori) e una coppia di timoni di direzione (superiore e inferiore; situati in coda, in corrispondenza ai timoni di profondità posteriori - **Fig. 2.1**).

Non sono presenti casse di compensazione né eliche di manovra (thruster trasversali): il controllo di profondità e di direzione è dinamico, e non è prevista l'operazione in hovering.

L'assetto statico di Venus è leggermente positivo, per ovvie ragioni di sicurezza (in caso di arresto del motore il veicolo torna in superficie).

## 2.2 Unità di calcolo

Sono attualmente presenti due embedded computer (nel seguito e. c.), alloggiati rispettivamente nella Testa e nella Coda, e costituiti ciascuno da una CPU montata su una scheda che alloggia anche i dispositivi periferici e le connessioni per l'I/O. Il sistema operativo impiegato è Linux – Linaro.

È in fase avanzata l'inserimento di una terza unità di calcolo, un Single Board Computer (nel seguito SBC) che sarà situato nella Coda. Le motivazioni verranno discusse in §4.3 e nel Cap. 7.

## 2.3 Comunicazione

La comunicazione fra le unità di calcolo interne è realizzata mediante un hub Ethernet situato nella Testa.

La comunicazione con la Stazione di Controllo remota (§2.6) consiste attualmente in un link WiFi, controllato dall'embedded computer di Testa. La relativa antenna è alloggiata in un cilindro di plexiglas montato su uno degli elementi della Testa.

Il link ottico per la comunicazione subacquea, citato nell'Introduzione, verrà anch'esso montato sulla Testa.

## 2.4 Sensori

Necessari al Controllo del veicolo e allo svolgimento delle missioni, sono installati:

### sulla Testa

- un modulo GPS con antenna integrata
- un sonar panoramico, montato su un piano verticale per svolgere le funzioni di rilevamento fondale
- una coppia di videocamere

### sulla Coda

- una bussola
- un pressostato, con funzione di profonditàmetro
- una Inertial Measurement Unit (IMU)

## 2.5 Attuatori

### sulla Testa

- una coppia di timoni di profondità

### sulla Coda

- una coppia di timoni di profondità
- una coppia di timoni di direzione
- il propulsore (motore + riduttore)

## 2.6 Stazione di Controllo

È costituita da un PC portatile dotato di link WiFi. Vi è installata l'interfaccia uomo-macchina (Man Machine Interface, MMI – **Cap. 5**), che consente all'operatore di avviare/arrestare i nodi su Venus, pilotarlo in teleguida, visualizzare i sensori di bordo e impostare e lanciare le missioni, nonché di scaricare i dati di log a fine missione.

### 3. Controllo - Architettura SW

Nel Sistema Venus possono essere distinti due strati di SW:

- SW c.d. 'di base', costituito dai demoni che governano varie funzioni di setup, comunicazione e sicurezza del sistema e dai driver che gestiscono i singoli dispositivi HW;
- SW c.d. 'di alto livello', che implementa il Sistema di Controllo.

Il primo strato è esaurientemente trattato in [3].

Il secondo costituisce l'argomento del presente Capitolo.

Il software di controllo di Venus è organizzato in blocchi funzionali indipendenti (Nodi), ciascuno dei quali effettua una parte del processamento dei dati necessari al funzionamento del Sistema.

Lo scopo della scelta di questo tipo di architettura è da una parte la semplificazione della scrittura e della manutenzione del SW, dall'altra la realizzazione di un sistema modulare, con possibilità di sostituire un singolo Nodo senza dover modificare i rimanenti.

Ciò è utile nel caso di sostituzione di componenti HW o di aggiornamento di driver SW: è possibile p. es. sostituire il componente IMU con un modello differente, oppure cambiare il pacchetto SW di gestione delle telecamere, modificando solamente i relativi Nodi, con la sola avvertenza di lasciare inalterati i dati di I/O, cioè l'interfacciamento col resto del sistema.

Sono previsti tre tipi di Nodi:

- gestione sensori
- gestione attuatori
- gestione processi Controllo.

Per ciascun Nodo sono definiti e codificati i dati di ingresso/uscita (I/O). La connessione fra i diversi Nodi avviene mediante lo scambio di tali dati, eventualmente dopo opportuna sincronizzazione.

I dati scambiati appartengono ad una delle seguenti tipologie:

- comandi (es: navigazione in controllo di direzione)
- parametri dei comandi (es: setpoint direzione)
- dati sensoriali (es: lettura bussola)
- file log/immagini (da Venus a MMI)

I comandi sono delle stringhe contenenti un codice identificativo seguito, eventualmente, da uno o più parametri. Vengono inviati da MMI o scambiati fra i Nodi, per richiedere al Nodo ricevente l'esecuzione di una particolare azione. Il Nodo ricevente si avvale di un semplice interprete dei comandi, appositamente sviluppato. La lista dei comandi implementati è riportata in **App. 2**.

Il flusso dei dati verrà discusso nel seguito del Capitolo.

### 3.1 ROS

L'architettura del Sistema di Controllo di Venus è basata sul ROS (Robot Operating System).

Il ROS è una infrastruttura flessibile per lo sviluppo di software dedicato a sistemi robotici. È una collezione di regole, strumenti e librerie per la costruzione di sistemi robotici complessi, orientata a favorire la cooperazione di gruppi di lavoro indipendenti. Può anche essere pensato come un meta-sistema operativo, poiché fornisce servizi tipici di un sistema operativo quali, fra gli altri, astrazione dall'hardware, controllo di dispositivi low-level, gestione di processi indipendenti e della comunicazione fra processi ([www.ros.org](http://www.ros.org)).

La configurazione in fase di esecuzione (runtime) è costituita da una rete di processi (Nodi) indipendenti connessi mediante l'infrastruttura di comunicazione dello stesso ROS.

È sempre presente un Nodo 'Master' con funzioni di supervisione e di coordinamento, in particolare riguardo allo scambio dati.

Le comunicazioni fra i diversi Nodi consistono in flussi asincroni di dati che viaggiano su 'code' dette Topic. Il concetto di coda sta ad indicare che i dati immessi su una Topic dal Nodo scrivente vengono impilati e letti, sulla stessa Topic e nello stesso ordine (FIFO), dal Nodo ricevente, secondo la sincronizzazione fornita dal ROS mediante le funzioni **spin()** (lettura continua) e **spinOnce()** (lettura singola).

Secondo la terminologia ROS un Nodo scrivente 'pubblica' su una Topic (Publisher), mentre un Nodo ricevente 'sottoscrive' una Topic (Subscriber).

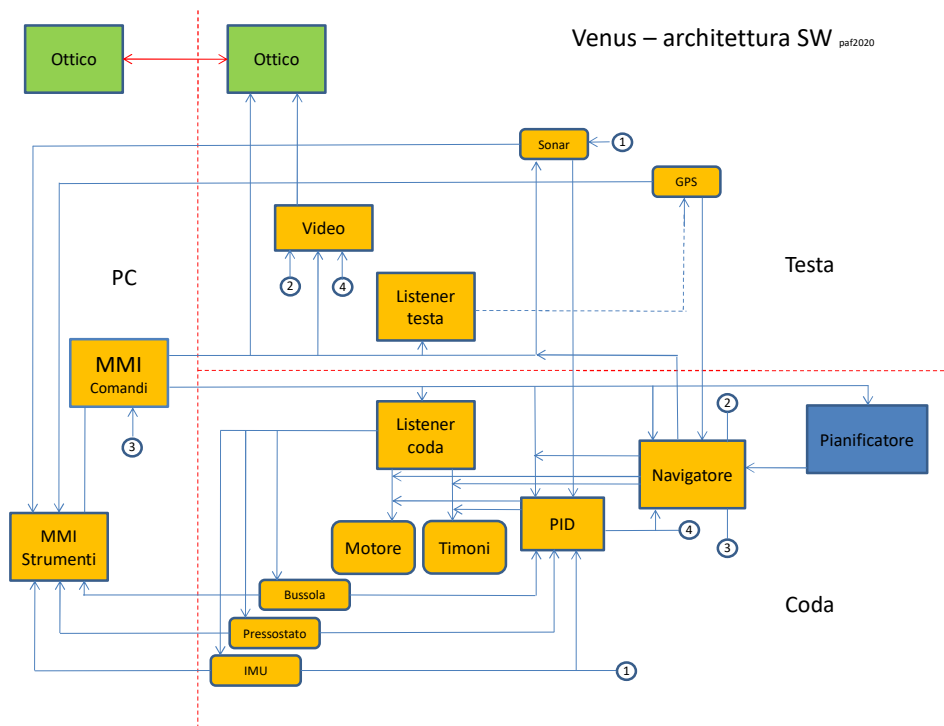
I Nodi possono essere distribuiti su differenti strutture HW.

Appare evidente la sovrapposizione di una simile impostazione con l'architettura scelta per Venus.

Delle numerose funzionalità di ROS quelle usate in Venus riguardano essenzialmente il set-up del sistema (installazione Nodo master, dichiarazione degli indirizzi TCP/IP dei singoli Nodi), le comunicazioni fra Nodi (gestione Topic), le funzioni di temporizzazione e alcune funzioni per il monitoraggio di Nodi e Topic.

### 3.2 Schema dell'architettura - definizione nodi

Lo schema dell'Architettura è riportato in **Fig. 3.1**.



**Figura 3.1 – Architettura SW**

La colorazione dei blocchi, ciascuno rappresentante un Nodo, segue la seguente codifica:

- giallo: nodo implementato, operativo
- verde: nodo parzialmente sviluppato, non operativo
- blu: nodo non ancora sviluppato

Le frecce indicano i flussi di dati fra i vari Nodi, schematizzando le Topic. Queste sono elencate, nodo per nodo, nel **Cap. 4**.

I Nodi sono distribuiti fra le due strutture HW 'Testa' e 'Coda' in modo da minimizzare, per quanto possibile, la distanza fisica fra l'unità di calcolo ed i sensori/attuatori con cui il singolo Nodo si interfaccia, al fine di ottimizzare la lunghezza dei cavi di collegamento, migliorando la qualità dei segnali trasmessi.

Sulla Testa sono installati i Nodi:

- GPS (sensori)
- Sonar (sensori)
- Video (sensori/processo)
- Listener (processo)
- Ottico (processo/comunicazione)

Sulla Coda sono installati/previsti i Nodi:

- Bussola (sensori)
- Pressostato (sensori)
- IMU (sensori)
- Listener (processo)
- [Pianificatore] (processo)
- Navigatore (processo)
- PID (processo)
- Motore (attuatori)
- Timoni (attuatori)

Sulla Stazione di Controllo è installato il Nodo

- MMI (Man Machine Interface, §2.6 - Cap. 5).

### 3.3 Sistema di Controllo

Viene di seguito riportata la struttura del Sistema di Controllo, con la descrizione delle funzioni dei Nodi che lo implementano e del flusso di dati fra di essi.

I singoli Nodi verranno descritti in dettaglio nel **Cap. 4**.

Scopo del Controllo è la realizzazione delle tipologie di missioni autonome indicate nel **Cap. 1** a partire da direttive, quanto più possibile sintetiche e user-friendly, impartite dall'operatore.

È inoltre possibile la teleguida del mezzo. Tale funzione è condizionata dalla presenza di un link fisico di comunicazione, e pertanto limitata attualmente alla navigazione in superficie. La teleguida subacquea sarà possibile – senza peraltro richiedere significative modifiche al Sistema di Controllo - non appena diventerà operativo il sistema di comunicazione ottica attualmente in fase di sviluppo presso l'Unità di Robotica.

Il metodo prescelto per l'esecuzione della singola missione prevede la suddivisione del relativo percorso in un numero finito di tratti elementari riconducibili ad un ristretto insieme di tipologie, pensate per consentire, con la loro combinazione, la copertura di un ampio spettro di casi operativi.

Le tipologie di tratto attualmente implementate sono:

- navigazione (generica) (bussola, profondimetro)



- navigazione in superficie (bussola)
- navigazione gps (bussola, GPS)

All'interno del singolo tratto va inserita, con le necessarie sincronizzazioni, l'attività dei sensori richiesti dalla particolare missione; attualmente è possibile programmare riprese foto/video.

Il flusso dei dati di controllo procede, attraverso i Nodi, nel seguente ordine:

MMI/[Pianificatore] → Navigatore → PID → attuatori (Motore e Timoni).

- il nodo MMI consente all'operatore di comporre, tratto per tratto, le missioni (**Cap. 5**). I tratti vengono quindi trasmessi, sulla Topic 'coda', al nodo Navigatore. Questo è attualmente l'unico metodo per l'inserimento delle missioni.
- il nodo Pianificatore non è ancora stato realizzato. È previsto che abbia la funzione di interpretare le direttive di alto livello impartite dall'operatore, generando le opportune missioni e suddividendole in tratti elementari, come discusso precedentemente. I tratti verranno quindi trasmessi, sulla Topic 'coda', al nodo Navigatore.
- Il nodo Navigatore riceve in ingresso, sulla Topic 'coda', un certo numero di tratti elementari, provenienti attualmente da MMI e in futuro dal Pianificatore.

**Nota:** la congruenza della sequenza dei tratti (intesa come costruzione di una missione compiuta) deve essere verificata a monte, nei nodi trasmettenti (MMI/Pianificatore). Il Navigatore effettua solo un controllo di congruenza fra il numero di tratti dichiarato e quello effettivamente ricevuto.

Per i tratti di **navigazione in superficie/profondità** il Navigatore determina i setpoint di direzione/profondità, inviandoli poi, con i relativi comandi (**App. 2**), al nodo successivo (PID).

Provvede inoltre, nel caso in cui siano previste riprese foto/video, ad inviare i relativi segnali di sincronizzazione al nodo Video.

Per i tratti di **navigazione con gps** il Navigatore attiva un timer specifico. Questo, ad ogni scadenza (il periodo attualmente impostato è pari a 2 sec), avvia l'esecuzione di una routine dedicata che, in sequenza

- converte i dati di longitudine e latitudine, ricevuti dal nodo GPS, in coordinate gaussiane, secondo un algoritmo ricavato da Mazzucato – Hirvonen, in [4]
- verifica il raggiungimento della posizione impostata, terminando nel caso il tratto in esecuzione ed arrestando il timer
- se la posizione non è ancora raggiunta, ma Venus è giunto in un intorno predefinito della stessa, diminuisce la potenza del motore, per favorire un avvicinamento controllato (§4.10 e commento

alle Figure **6.2, 6.3**)

- se la posizione non è ancora raggiunta calcola la direzione da impostare (setpoint direzione) in base ai nuovi dati di posizione, inviandola ai nodi PID e MMI

Per ogni tipo di tratto il Navigatore invia infine al nodo Motore il comando di accensione, insieme al valore numerico della potenza da impostare.

- il nodo PID riceve in ingresso, sulla Topic ‘coda’, i comandi, provenienti dal Navigatore (missioni programmate, discusse al punto precedente) o da MMI (teleguida), relativi ai controlli di direzione e di profondità.

La sua funzione è quella di calcolare le leggi di controllo dei timoni di direzione e profondità, determinandone i setpoint di inclinazione, che vengono quindi forniti come input al nodo Timoni.

Si nota qui, per inciso, che attualmente non è implementata nessuna legge di controllo del propulsore, non essendo disponibile alcuna retroazione. La potenza motore viene fissata per ogni tratto di missione. Ciò può costituire un limite in alcune situazioni, p. es. nel controllo dell’assetto (vedi commento alla Fig. **6.13** e **Cap. 7**).

PID è l’acronimo di controllo Proporzionale – Integrabile – Derivativo. Si tratta, come noto, di un controllo semplice e robusto, che si è dimostrato adatto a gestire in modo soddisfacente i profili impostati di profondità e di direzione (**Cap. 6**).

L’esperienza sul campo fin qui condotta ha mostrato che per il controllo di direzione è sufficiente un Proporzionale puro, mentre per il controllo di profondità viene applicata una leggera correzione Derivativa. Per il calcolo delle leggi di controllo il nodo PID impiega i dati di bussola, profundimetro, IMU e sonar, provenienti dai rispettivi Nodi sulle Topic dedicate.

I dati di direzione e profondità entrano direttamente nei rispettivi loop di controllo.

Dalla IMU viene ricavato l’angolo di beccheggio (pitch), impiegato nel controllo delle fasi di discesa alla profondità impostata e di risalita – fasi effettuate ad assetto costante (§**4.11**) - e nella determinazione della direzione verticale nel sistema di riferimento di Venus, necessaria per il calcolo della distanza dal fondo (§**4.2**).

La lettura del sonar viene attualmente impiegata per la determinazione della posizione del Venus rispetto al fondale, in modo da poter sempre garantire il rispetto di una distanza minima di sicurezza.

Si nota in proposito che sono possibili due modalità di navigazione in profondità:

- in quota (quota costante, distante dal fondo; es: monitoraggio inquinamento o microplastiche)
- sul fondo (distanza dal fondo costante e minima; es: ricognizione video del fondale)

Nel primo caso il setpoint di profondità è fisso; solo in presenza di innalzamenti del fondale non previsti il controllo provvede ad aggiustarlo per riportarlo alla distanza minima di sicurezza.

La seconda modalità è ottenuta semplicemente impostando una quota sicuramente inferiore a quella del fondale che si intende esplorare; il controllo provvede, operando come sopra, a modificare il setpoint per mantenere il veicolo alla distanza di sicurezza, e l’aggiustamento è in questo caso continuo (§**4.11** e **Cap. 6**). In futuro è previsto l’impiego del sonar anche per funzioni di obstacle avoidance (§**4.2** e **Cap. 7**).

Gli attuatori sono azionati, rispettivamente, dai nodi Motore e Timoni, il cui output agisce direttamente sull'elettronica di controllo dei dispositivi [2].

- Il nodo Motore è in ascolto sulla Topic 'motore', da cui legge la potenza impostata (setpoint), espressa come percentuale della potenza massima.  
Riceve l'input dal nodo Navigatore (come discusso) o da MMI tramite il nodo Listener (teleguida); può ricevere un comando di arresto di emergenza dal nodo PID.
- Il nodo Timoni è in ascolto sulla Topic 'timoni', da cui legge, in sequenza, le posizioni impostate (setpoint) delle singole pinne.  
Riceve l'input dal nodo PID (come discusso) o da MMI tramite il nodo Listener (teleguida); riceve dal nodo Navigatore il comando di reset dei timoni a fine missione.

Entrambi i nodi sono anche in ascolto sulla Topic 'strumnav', sulla quale ricevono, provenienti da MMI, comandi operativi quali avvio/arresto del Nodo e gestione della registrazione dei dati di funzionamento (log su file/standard output - §3.4).

### 3.4 Registrazione (log) dati

Sono implementate due modalità di registrazione dati, rispettivamente su file e su standard output.

- La prima modalità riguarda i dati relativi alla navigazione, ai sensori e all'impostazione degli attuatori. Ogni nodo scrive, su file dedicati, i dati di propria pertinenza (**Cap. 4**). I file vengono di norma scaricati sulla Stazione di Controllo, tramite MMI, alla fine della missione (o di un gruppo di missioni, o di un periodo di teleguida), per essere esaminati off-line. Da essi vengono tratti i grafici che visualizzano gli aspetti salienti dell'andamento della missione, come verrà mostrato nel **Cap. 6**.  
Particolare cura è stata posta nel velocizzare l'estrazione dei dati e la realizzazione dei grafici, per cui è possibile controllare direttamente sul campo l'andamento della singola missione, per poter procedere alle correzioni di parametri che si ritenessero opportune e verificarne subito l'efficacia.
- La seconda modalità riguarda essenzialmente i dati relativi al funzionamento del SW di controllo, nei suoi vari aspetti. I dati vengono scritti su standard output, per essere visualizzati in tempo reale su terminali (shell) aperti sulla Stazione di Controllo.  
La visualizzazione è ovviamente possibile in presenza del link fisico di comunicazione, come discusso nel §3.3; sono quindi attualmente esclusi i tratti in immersione. È però possibile registrare su file anche questa tipologia di dati, mediante reindirizzamento dello standard output, consentendone l'esame off-line.  
D'altra parte l'apertura di terminali di controllo è propria dell'attuale fase di sviluppo di Venus, mentre a regime l'unica interfaccia impiegata sarà MMI. Verrà valutata l'utilità di rendere stabile la registrazione di

parte dei dati in questione, riconducendoli alla prima modalità.

Si nota infine che, a causa della notevole mole di dati interessati dalla seconda modalità, si è reso necessario parametrizzarne l'acquisizione. Sono previsti a tal fine, per ciascun Nodo, tre livelli di registrazione (parametro 'log\_lev': 0/1/2 <-> solo dati essenziali/insieme intermedio/tutti i dati previsti) e, per i dati prodotti a frequenze elevate (2 Hz e oltre), un campionamento variabile (parametri 'log\_frq', 'log\_cnt'). Livello e frequenza di registrazione sono modificabili in un'apposita sezione di MMI (**Cap. 5**).

La gestione della registrazione dati è completata da una serie di comandi, impartiti da MMI, comprendenti l'avvio e l'arresto delle registrazioni, il download sulla Stazione di Controllo e il reset (cancellazione) dei file di log.

La possibilità di arrestare e riavviare la registrazione consente di scaricare ordinatamente i dati dopo ogni missione, permettendone, come discusso, l'esame immediato; consente inoltre di non sovraccaricare la memoria di sistema e di effettuare download non eccessivamente lunghi (vedi, nel **Cap. 7**, le criticità nel trasferimento dati su WiFi).

La possibilità di cancellare periodicamente i file di log ha l'ovvio fine di ottimizzare lo spazio di archiviazione.

## 4. Descrizione nodi - tabelle I/O

Sotto la voce **MMI** sono elencati, per ciascun Nodo, i parametri modificabili in fase di esecuzione da MMI.

### Testa

#### 4.1 GPS

**Input:** sensore GPS - interfaccia RS232, protocollo NMEA

**Output:** Topic 'fix'

Formato: sensor\_msgs::NavSatFix

**Log file:** → effettuato nel nodo Navigatore

Il nodo GPS legge, in continuo, i dati generati dal modulo GPS, trasmettendoli sulla Topic 'fix'.

I dati impiegati nel controllo sono le coordinate geografiche latitudine e longitudine.

Il SW nmea\_navsat\_driver [E. Perko e S. Martin], open source, proviene da ros.org e viene impiegato senza modifiche.

## 4.2 Sonar

<b>Input:</b>	sonar Trittech Micron	- interfaccia RS232
	Topic 'imu'	
	Formato: 3 float in	std_msgs::String
	Topic 'testa'	
	Formato:	std_msgs::String
<b>Output:</b>	Topic 'sonar'	
	Formato: 4 float in	std_msgs::Float32MultiArray
<b>Log file:</b>	sonar_nav.txt	- parametri configurazione sonar
	sonar_log.txt	- distanza dal fondo
	dfmed_log.txt	- distanza dal fondo mediata
	ob_av_log.txt	- obstacle avoidance
<b>MMI:</b>	range	- profondità massima di scansione
	gain	- guadagno %
	numBins	- numero impulsi/scansione
	min_dist	- distanza di sicurezza dal fondo
	leftLim	- limite angolare sinistro scansione
	rightLim	- limite angolare destro scansione
	filt_flg	- filtro distanza fondo ON/OFF
	log_lev	- livello log
	log_frq	- frequenza log

Il nodo Sonar legge, in continuo, i dati generati dal sonar, trasmettendoli, dopo le necessarie elaborazioni, sulla Topic 'sonar'.

Il SW è derivato da quello sviluppato da M. Martinelli per [5], cui sono stati apportate alcune correzioni nella inizializzazione/gestione dei parametri e le modifiche necessarie al diverso tipo di operazione richiesto, in accordo con le specifiche Trittech [6].

Il Trittech Micron è un sonar panoramico, che opera normalmente su un piano orizzontale acquisendo dati su 360° in rotazione continua. Il range dell'angolo di acquisizione è modificabile a piacimento. Se minore di 360°, il range viene scansionato alternativamente nelle due direzioni possibili (oraria – antioraria).

Il range è suddiviso in step angolari. Ad ogni step corrisponde una scansione.

Sul Venus il sonar è stato montato sotto la prua, sul piano verticale comprendente l'asse del veicolo, in modo

che, nel proprio sistema di riferimento,  $90^\circ$  coincidano con la direzione di marcia e  $180^\circ$  con la verticale verso il basso.

Il range di acquisizione è attualmente limitato fra  $90^\circ$  e  $210^\circ$ .

È così possibile rilevare sia dati relativi a possibili ostacoli in direzione di marcia ( $90^\circ$  - obstacle avoidance), sia dati relativi alla distanza dal fondo, sulla verticale.

Si noti che il fondo viene letto a  $180^\circ$  in caso di assetto orizzontale, mentre nelle fasi di immersione ed emersione, considerando un'inclinazione massima del Venus di  $\pm 30^\circ$ , la verticale varia fra  $150^\circ$  e  $210^\circ$ ; da qui l'estensione del range a  $210^\circ$ , mentre, volendo limitare l'acquisizione alla sola distanza dal fondo (escludendo obstacle avoidance), il range verrebbe direttamente limitato fra  $150^\circ$  e  $210^\circ$ , con benefici sulla velocità di scansione e quindi sulla continuità della rilevazione.

Nel caso si prevedessero angoli di inclinazione maggiori di  $30^\circ$  in immersione/emersione il range andrebbe modificato di conseguenza, mantenendo preferibilmente la simmetria intorno ai  $180^\circ$ .

Attualmente le funzioni di obstacle avoidance non sono implementate (**Cap. 7**).

Di seguito viene brevemente esaminato il flusso dei dati, che è organizzato in un loop infinito (in ogni ciclo viene processata una scansione del sonar). Vengono tralasciate le chiamate specifiche alle funzioni del SW Tritech (per eventuali approfondimenti si rimanda a [6]).

Nel ciclo viene dapprima acquisita la relativa linea di scansione, della quale viene letto l'angolo (nel sistema di riferimento del sonar), approssimandolo all'intero più vicino. Viene quindi calcolata la verticale, sommando a  $180^\circ$  l'angolo di beccheggio (pitch) letto sulla Topic 'imu'.

La linea di scansione è composta da un certo numero di impulsi ('Bin'), attualmente impostato a 200; con una distanza massima di scansione (range) attualmente impostata a 10 m, si ha una risoluzione di 5 cm.

Ogni impulso corrisponde ad una determinata distanza dal sonar ed è caratterizzato dall'ampiezza dell'eco di ritorno.

Per determinare la distanza effettiva del bersaglio (nel nostro caso il fondale), è necessario individuare l'impulso corrispondente. Possono essere impiegati diversi algoritmi, in dipendenza dalla necessità di discriminare eventuali echi spuri. Nella presente applicazione è stato usato il più semplice, consistente nel considerare l'impulso associato alla massima ampiezza del segnale di ritorno. La validità della scelta è stata confermata dai risultati sperimentali, ottenuti, come verrà descritto in seguito (**Cap. 6**), nel lago di Bracciano; differenti tipi di fondale potrebbero richiedere scelte diverse.

Per tenere in conto la non omogeneità del fondale del lago (sabbia + alghe) si è deciso di non considerare la sola scansione lungo la verticale, ma tutte quelle che ricadono in un intorno della verticale stessa, di ampiezza pari a sei volte la dimensione angolare dello step di scansione (che è un parametro del sonar, attualmente impostato a  $0.9^\circ$ ). Di queste scansioni viene effettuata la media semplice, che costituisce il valore calcolato della distanza dal fondo, da impiegare nel controllo di profondità.

Tale valore viene infine scritto sulla Topic 'sonar' insieme alla relativa ampiezza dell'eco (anch'essa mediata come sopra) ed alla distanza minima dal fondo, che è un parametro della navigazione (§3.3) ed è impostabile

da MMI (valore di default attuale pari a 1 m).

I primi due valori vengono anche registrati, insieme all'angolo della verticale, nel file di log `dfmed_log.txt`.

In modo analogo vengono acquisite e registrate le altre linee di scansione comprese nei limiti impostati (file di log `sonar_log.txt` e, per angoli fra  $90^\circ$  e  $100^\circ$ , `ob_av_log.txt`, dedicato all'obstacle avoidance in direzione di marcia).

Poiché, come si è detto, non sono attualmente implementate funzioni di obstacle avoidance, i dati relativi non vengono trasmessi sulla Topic 'sonar', per non sovraccaricare la banda WiFi.



### 4.3 Video

<b>Input:</b>	webcam Logitech C615 1&2	- interfaccia USB
	Topic 'syncfv'	
	Formato:	std_msgs::Int16
	Topic 'quota'	
	Formato:	std_msgs::Float32
	Topic 'testa'	
	Formato:	std_msgs::String
<b>Output:</b>	Video streaming su Stazione Controllo (se presente link fisico comunicazione)	
	Registrazione foto/video su file	
	Topic 'ottica'	
	Formato:	std_msgs::String
<b>Log file:</b>	miss_t.txt	- tratti missioni (Testa)
<b>MMI:</b>	log_lev	- livello log
	log_frq	- frequenza log

Il nodo Video gestisce le videocamere installate su Venus, permettendo l'acquisizione, la registrazione e l'eventuale trasmissione in streaming dei video (o dei singoli scatti) richiesti dalla missione in corso.

Lo streaming è visualizzato sulla Stazione di Controllo ed è, al solito, condizionato dalla presenza di un link fisico di comunicazione; quindi, nella fase attuale, è limitato alla navigazione in superficie.

Il SW è basato sulla utility Mplayer.

**Nota:** al momento della stesura del presente documento è in avanzata fase di realizzazione una radicale modifica del sistema di acquisizione video, che comporterà l'inserimento di una terza struttura HW dedicata esclusivamente allo scopo, nonché l'impiego di un SW più efficiente [Ffmpeg]. L'argomento verrà brevemente discusso nei **Cap. 6 e 7**.

L'acquisizione video è governata da comandi sia diretti (in teleguida) che derivanti da direttive in una missione autonoma.

Il nodo Video rimane quindi in ascolto sulla Topic 'testa', in attesa del singolo comando.

Oltre ai comandi di gestione log (registrazioni e parametri missione), quelli propriamente riguardanti i video comprendono (**App. 2**):

- gestione fari
- foto singole (con e senza attivazione fari)
- streaming video/registrazione video da videocamera sinistra/destra
- programmazione foto/video (sequenze di foto/video con intervalli predeterminati)
- trasmissione su link ottico (\*)
- gestione missioni

(\*) il link ottico, come già accennato, è in fase di sviluppo. Il software realizzato - ed inserito nel nodo Ottico (§4.5) - è stato impiegato in fase di test del prototipo, dimostrando il corretto funzionamento della trasmissione dati, ma non può considerarsi definitivo.

Il comando di gestione missioni provvede all'esecuzione delle acquisizioni foto/video programmate nei singoli tratti. Vengono impiegate le procedure della programmazione foto/video sopra indicata (nelle prove sul campo si è preferito in genere eseguire diverse brevi riprese nel singolo tratto, anche in considerazione del fatto che effettuare una ripresa continua, per l'intera durata del tratto, potrebbe essere assai dispendioso in termini di consumo di memoria e di tempo di trasmissione dei file video a MMI a fine missione - vedi, nel **Cap. 7**, problemi di banda WiFi).

Nella gestione missioni assume importanza la sincronizzazione delle acquisizioni. Infatti mentre il nodo Video viene eseguito sulla Testa, tutti gli altri aspetti della missione sono gestiti dal nodo Navigatore, che viene eseguito sulla Coda. Le due strutture HW Testa e Coda non sono di per sé sincronizzate (si ricorda che lo scambio dati fra i nodi è asincrono, come permesso dal sistema delle Topic ROS).

Nel caso dell'esecuzione di una missione la sincronizzazione dei video nei rispettivi tratti è ovviamente essenziale. Essa viene effettuata mediante l'invio di appositi segnali dal Navigatore al Video, sulla Topic 'syncfv'. Il Navigatore invia un solo segnale di sincronizzazione non appena il singolo tratto è effettivamente iniziato, dopodiché le differenti acquisizioni previste procedono secondo la programmazione foto/video impostata, come discusso in precedenza.

Per facilitare l'archiviazione e il successivo esame dei file foto/video vengono inseriti, nel nome del file, le informazioni di orario e profondità immediatamente precedenti l'inizio della registrazione.

L'informazione di tempo è presa direttamente dal sistema operativo, mentre quella di profondità viene letta dalla Topic 'quota', sulla quale è stata scritta dal nodo PID.

## 4.4 Listener

<b>Input:</b>	Topic 'testa'		
	Formato:	std_msgs::String	
<b>Log file:</b>	logV<Venus id>testa.txt	- Standard Output Testa	(es: logV2testa.txt)
<b>MMI:</b>	log_lev	- livello log	
	log_frq	- frequenza log	

Il nodo Listener contiene un semplice interprete dei comandi che vengono inviati da MMI o scambiati fra i nodi (**Cap. 3**). È presente anche sulla Coda.

In realtà i Nodi più complessi hanno un proprio interprete dei comandi (peraltro identico a quello di Listener), per ragioni di modularità e di comodità di scrittura.

Per quanto riguarda, in particolare, la Testa, sia Sonar che Video hanno il proprio interprete, mentre il nodo GPS non riceve comandi dal Sistema.

Le funzioni di Listener sono quindi piuttosto limitate. Esse comprendono:

- Avvio nodi Testa
- arresto nodi GPS e Listener
- aggiornamento strutture parametri log nodi Testa
- gestione file log Standard Output (\*)

(\*) allo stato attuale la registrazione su Standard Output avviene solo parzialmente. Ciò è dovuto alla 'interferenza' causata dal nodo Video nelle fasi di acquisizione. La modifica accennata nel §4.3 – **Nota** (vedi anche **Cap. 6 e 7**) dovrebbe risolvere questo problema.

Il nodo Listener rimane in ascolto sulla Topic 'testa', in attesa del singolo comando.

È l'unico Nodo a partire automaticamente all'accensione della macchina; provvede all'avviamento degli altri nodi della Testa non appena riceve il comando di Start, inviato da MMI (**Cap. 5**).

## 4.5 Ottico

<b>Input:</b>	micro Arduino	- interfaccia seriale [RS232]
	Topic 'ottica'	
	Formato:	std_msgs::String
	Topic 'testa'	
	Formato:	std_msgs::String
<b>Output:</b>	micro Arduino	- interfaccia seriale [RS232]
	Topic 'testa'	
	Formato:	std_msgs::String
	Topic 'coda'	
	Formato:	std_msgs::String
<b>MMI:</b>	log_lev	- livello log
	log_frq	- frequenza log

Il nodo Ottico ha il compito di gestire le comunicazioni subacquee fra Venus e MMI (o fra più Venus) sul link ottico.

Il SW è basato sul driver implementato da G. Cupertino [3].

**Nota:** come accennato a più riprese (**Cap. 1 – §2.3**), il link ottico è attualmente in fase di realizzazione. Il nodo Ottico è stato impiegato per collaudarne il sottosistema di interfaccia con l'HW di bordo del Venus, già realizzato e basato su microcontrollore Arduino (nei test il link ottico è stato emulato con un link seriale). Durante il collaudo è stato trasferito un file contenente un'immagine (jpg), dimostrando l'operatività del sottosistema, ma naturalmente la realizzazione definitiva potrà comportare modifiche HW/SW. Per questa ragione si considera il Nodo 'parzialmente implementato'. L'output previsto sulle Topic 'testa' e 'coda' (trasmissione comandi per via ottica) non è inoltre riportato in Fig. 3.1.

Analogamente alla trasmissione in aria, mediante WiFi, possono essere scambiati dati appartenenti ad una delle seguenti tipologie:

- Comandi, con i relativi parametri (MMI -> Venus, Venus <-> Venus)
- dati sensoriali (Venus -> MMI)
- file log/immagini (Venus -> MMI)

Il flusso dati è organizzato in un loop infinito, il cui singolo ciclo è costituito da una fase di Ricezione e da una

fase di Trasmissione, che vengono eseguite consecutivamente.

### Ricezione

Il ciclo inizia verificando l'eventuale presenza di un carattere di controllo in ingresso sul link seriale (futuro link ottico). In caso affermativo si ha, secondo il valore trovato:

- 1: presenza di un file da leggere (es: foto/video)
- 2: presenza di una stringa da leggere (es: comando)
- 4: comunicazione della corretta trasmissione file da parte della stazione trasmittente

Nei casi (1) e (2) si controlla innanzitutto se la trasmissione è indirizzata al Venus su cui ci troviamo, o se si tratta di una trasmissione broadcast. In caso affermativo vengono lanciate rispettivamente le routine di acquisizione file/stringa.

Al termine dell'acquisizione, o in caso di assenza di dati in ingresso, si passa alla fase di

### Trasmissione

La presenza di dati in uscita è segnalata da Venus mediante il primo carattere di una stringa dedicata ('line').

Se diverso da \0 si ha:

- '\$': Arduino in modalità modem
- '1': Arduino in modalità configurazione [in corso di implementazione]
- '2': trasmissione stringa (es: comando)
- '3': trasmissione file (es: foto/video)
- '4' riserva (a disposizione)

Nei casi '2' e '3' la trasmissione inizia con due byte indicanti rispettivamente il tipo di trasmissione ((2) e (1), che verranno letti dal Venus destinatario come specificato al punto precedente 'Ricezione') e l'identità del Venus destinatario (eventualmente broadcast). Vengono quindi lanciate rispettivamente le routine di trasmissione stringa/file.

A conclusione del ciclo, la stringa 'line' viene azzerata.

## Coda

### 4.6 Bussola

<b>Input:</b>	bussola elettronica	- interfaccia i2c
	Topic 'strumnav'	
	Formato:	std_msgs::Int16
<b>Output:</b>	Topic 'bussola'	
	Formato:	std_msgs::Float32
<b>Log file:</b>	buss_log.txt	- direzione moto
<b>MMI:</b>	log_lev	- livello log
	log_frq	- frequenza log

Il nodo Bussola legge, in continuo, i dati generati dalla bussola elettronica, trasmettendoli sulla Topic 'bussola'. Rimane inoltre in ascolto sulla Topic 'strumnav', sulla quale riceve i comandi operativi di avvio/arresto nodo e gestione della registrazione dei dati di funzionamento (log su file/standard output).

Il SW è basato sul driver implementato da G. Cupertino [3].

## 4.7 Pressostato

<b>Input:</b>	pressostato	- segnale in corrente; interfaccia analogica (ADC e. c. Testa)
	Topic 'strumnav'	
	Formato:	std_msgs::Int16
<b>Output:</b>	Topic 'pressostato'	
	Formato:	std_msgs::Float32
<b>Log file:</b>	press_log.txt	- quota baricentro, velocità verticale
<b>MMI:</b>	log_lev	- livello log
	log_frq	- frequenza log

Il nodo Pressostato legge, in continuo, i dati generati dal pressostato (qui impiegato come profondimetro), elaborandoli e trasmettendo sulla Topic 'pressostato' i valori di quota e velocità verticale calcolati.

Rimane inoltre in ascolto sulla Topic 'strumnav', sulla quale riceve i comandi operativi di avvio/arresto nodo e gestione della registrazione dei dati di funzionamento (log su file/standard output).

Il SW è basato sul driver implementato da G. Cupertino [3].

Il dispositivo restituisce un valore di tensione decrescente all'aumentare della pressione idrostatica, quindi della profondità.

All'inizio delle operazioni dev'essere determinata la tensione di riferimento, corrispondente alla quota 0, cioè alla superficie dell'acqua.

È bene ripetere periodicamente l'operazione di 'zero', per contrastare eventuali fenomeni di deriva. Per tale motivo lo 'zero', effettuato automaticamente per la prima volta all'accensione del veicolo, viene ripetuto dopo ogni arresto/riavviamento del log dati (file press\_log.txt); a sua volta la sequenza di arresto/riavviamento viene tipicamente eseguita ad ogni download del suddetto file sulla Stazione di Controllo, tramite MMI.

Durante lo 'zero' il veicolo deve rimanere rigorosamente immobile, su una superficie acquatica calma. È bene quindi avviare il Venus quando è già in acqua, attendendo il tempo necessario alla sua stabilizzazione, ed effettuare, per quanto possibile, il download dei dati di log a veicolo immobile in superficie.

La tipologia del dispositivo (segnale di uscita in corrente) ha reso necessario utilizzare, per il calcolo del singolo valore di tensione, la media effettuata su un certo numero di acquisizioni (attualmente impostato a 100). Inoltre all'avvio del Venus i primi quattro cicli di lettura vengono scartati, e la tensione di riferimento è quella fornita dal quinto ciclo.

Nel generico ciclo di funzionamento la profondità calcolata è proporzionale alla differenza fra la tensione

acquisita e la tensione di riferimento. La velocità verticale è proporzionale alla derivata discreta della tensione acquisita.



## 4.8 IMU

<b>Input:</b>	modulo IMU Phidget	- interfaccia USB
	Topic 'strumnav'	
	Formato:	std_msgs::Int16
<b>Output:</b>	Topic 'imu'	
	Formato:	std_msgs::String
<b>Log file:</b>	imu_log.txt	- roll, pitch, yaw
<b>MMI:</b>	log_lev	- livello log
	log_frq	- frequenza log
	rec_all_flg	- flag scrittura continua su buffer (*)
	t_file	- periodo stampa completa su file (*)

(\*) la scrittura su buffer/file del set completo dei dati IMU non è attualmente utilizzata.

Il nodo IMU legge, in continuo, i dati generati dal modulo IMU, trasmettendo sulla Topic 'imu' i valori degli angoli di roll, pitch e yaw (rollio, beccheggio e imbardata).

Rimane inoltre in ascolto sulla Topic 'strumnav', sulla quale riceve i comandi operativi di avvio/arresto nodo e gestione della registrazione dei dati di funzionamento (log su file/standard output).

Il SW è basato sulle specifiche Phidget [7]

Il set completo dei dati acquisiti consiste in

- tempo acquisizione (intero, secondi)
- tempo acquisizione (frazione, microsecondi)
- accelerazione asse x
- accelerazione asse y
- accelerazione asse z
- velocità angolare asse x
- velocità angolare asse y
- velocità angolare asse z
- campo magnetico asse x

- campo magnetico asse y
- campo magnetico asse z

Il dispositivo richiede una calibrazione iniziale. I relativi valori vengono calcolati al decimo ciclo di acquisizione a partire dall'avviamento, mentre i primi nove vengono scartati.

Il set completo di dati viene inviato dal dispositivo all'interfaccia – e quindi al nodo IMU – con una frequenza preimpostata (il periodo deve essere un multiplo di 8 ms; quello attualmente impostato è pari a 256 ms).

Allo stato attuale, i dati di accelerazione lineare non vengono direttamente impiegati nel Controllo di Venus. A regime vengono calcolati gli angoli di roll, pitch, e yaw, a partire dalle accelerazioni lineari acquisite; i valori calcolati vengono in seguito convertiti in gradi sessagesimali e inviati sulla Topic 'imu' con una temporizzazione preimpostata (il periodo è attualmente pari a 1 sec).

## 4.9 Listener

<b>Input:</b>	Topic 'coda'		
	Formato:	std_msgs::String	
<b>Output:</b>	Topic 'motore'		
	Formato:	std_msgs::Int16	
	Topic 'timoni'		
	Formato:	std_msgs::Int16	
	Topic 'strumnav'		
	Formato:	std_msgs::Int16	
<b>Log file:</b>	logV<Venus id>coda.txt	- Standard Output Coda	(es: logV2coda.txt)
<b>MMI:</b>	log_lev	- livello log	
	log_frq	- frequenza log	

Il nodo Listener contiene un semplice interprete dei comandi che vengono inviati da MMI o scambiati fra i nodi (**Cap. 3**).

A differenza di quanto avviene sulla Testa (§4.4), solo i nodi Navigatore e PID hanno un proprio interprete dei comandi, mentre i nodi Bussola, Pressostato, IMU, Motore e Timoni dipendono (\*) da Listener.

Le funzioni di Listener comprendono:

- Avvio nodi Coda
- arresto nodi dipendenti (\*) e Listener
- comandi diretti motore (da MMI)
- comandi diretti timoni direzione e profondità (da MMI)
- test pinne
- gestione log dati nodi dipendenti (\*)
- aggiornamento strutture parametri log nodi Coda
- gestione file log Standard Output
- gestione parametri modificabili da MMI nodi dipendenti (\*)

Il gruppo di comandi diretti agli attuatori (motore e timoni) implementa la teleguida di Venus (**Cap. 5**). I comandi vengono trasmessi rispettivamente sulle Topic 'motore' e 'timoni'.

Il test delle pinne, originariamente sviluppato per verificarne il funzionamento, conserva l'importante funzione di calibrazione ('zero'), che si rende necessaria dopo eventuali interventi di manutenzione/sostituzione degli

attuatori dei timoni, o in caso di disallineamenti dovuti ad urti delle pinne contro ostacoli esterni. I comandi vengono trasmessi sulla Topic ‘timoni’.

Il gruppo di comandi relativi alla gestione dei dati di log consente di attuare quanto discusso in §3.4. I comandi vengono trasmessi sulla Topic ‘strumnav’.

Il nodo Listener ha in carico la gestione di due timer di sicurezza, ‘stopTim’ e ‘emrgTim’. L’azione di entrambi consiste nell’arresto del motore, ed entrambi vengono inizializzati ed avviati ogni volta che il motore è azionato in teleguida, quindi da MMI:

- stopTim: la teleguida non può iniziare se prima non ne è impostata la durata, il cui valore, in sec, è trasmesso da MMI a Listener come parametro, mediante il comando di stop (**Cap. 5**). Il blocco della partenza in caso di mancata impostazione è effettuato a livello di MMI (rifiuto trasmissione comando di start – **Cap. 5**)
- emrgTim: viene impostato ad ogni movimento della barra potenza motore su MMI (**Cap. 5**), ma ha una durata fissa, non modificabile da MMI, che stabilisce il tempo massimo previsto per un singolo lancio in teleguida.

I due timer sono evidentemente ridondanti, scelta giustificata dalla serietà delle conseguenze di una perdita di controllo in navigazione subacquea. In questo caso infatti viene a mancare, come più volte discusso, l’unico link di comunicazione attualmente disponibile (WiFi), e con esso la possibilità di arrestare il veicolo in emergenza, con conseguente possibile perdita del veicolo stesso.

L’arresto del motore causa la risalita in superficie del Venus, a causa dell’assetto positivo (§2.1).

Si tratta naturalmente di una soluzione legata all’attuale fase di sviluppo. In presenza del link ottico, e quindi della possibilità di comunicazione subacquea, la strategia illustrata potrà essere rivista, sebbene si ritenga utile il mantenimento di almeno un timer di sicurezza.

Si nota infine che la ridondanza è ulteriormente rafforzata da un timer inserito nel SW di base [3].

Il nodo Listener rimane in ascolto sulla Topic ‘coda’, in attesa del singolo comando.

È l’unico Nodo a partire automaticamente all’accensione della macchina; provvede all’avviamento degli altri nodi della Coda non appena riceve il comando di Start, inviato da MMI (**Cap. 5**).

## 4.10 Navigatore

<b>Input:</b>	Topic 'fix'		
	Formato:	sensor_msgs::NavSatFix	
	Topic 'quota'		
	Formato:	std_msgs::Float32	
	Topic 'coda'		
	Formato:	std_msgs::String	
<b>Output:</b>	Topic 'coda'		
	Formato:	std_msgs::String	[-> PID]
	Topic 'testa'		
	Formato:	std_msgs::String	[-> Listener -> GPS]
	Topic 'syncfv'		
	Formato:	std_msgs::Int16	
	Topic 'motore'		
	Formato:	std_msgs::Int16	
	Topic 'timoni'		
	Formato:	std_msgs::Int16	
	Topic 'feedback'		
	Formato:	std_msgs::String	
<b>Log file:</b>	miss_c.txt	- tratti missioni (Coda)	
	gps_nav.txt	- latitudine e longitudine (da GPS)	
<b>MMI:</b>	log_lev	- livello log	
	log_frq	- frequenza log	

Il ruolo del nodo Navigatore nel controllo di Venus è stato illustrato in §3.3.

Il nodo, dotato di proprio interprete dei comandi, rimane in ascolto sulla Topic 'coda'.

Oltre ai soliti comandi per l'arresto nodo e per la gestione del log dati, quelli specifici per la navigazione sono relativi alle seguenti modalità:

- Navigazione a loop aperto
- Navigazione con gps
- Gestione missioni

La prima modalità non è più in uso. È stata sviluppata per effettuare le prime prove di navigazione di Venus in acque libere (\*), ed è caratterizzata dall'assenza di retroazione di direzione e profondità. Consiste nell'impostare (su MMI) una serie di tratti prefissando, per ciascuno di essi, l'inclinazione dei singoli timoni e il tempo di esecuzione. I tratti vengono eseguiti consecutivamente.

Considerando p. es. tre tratti, e programmando i timoni di profondità in modo che nel primo siano rivolti verso il basso, nel secondo siano in posizione orizzontale e nel terzo siano rivolti verso l'alto, si ottiene un percorso subacqueo (immersione – tratto orizzontale – riemersione) elementare ma utile per il collaudo in condizioni operative degli attuatori e per la verifica della tenuta stagna e dell'assetto del veicolo.

(\*). Tutte le prove in acque libere si sono svolte nel lago di Bracciano (**Cap. 6**).

La seconda modalità consente di navigare, con l'ausilio del GPS, fino ad un punto di coordinate geografiche (latitudine e longitudine) note, impostando queste ultime direttamente da MMI o come parte di una missione. Per raggiungere il punto prefissato ('bersaglio') è necessario innanzitutto trasformare le coordinate geografiche dei punti di partenza (posizione corrente) e di arrivo in coordinate Gaussiane. Ottenuta così una mappatura dei due punti in un sistema di riferimento piano XY, è possibile calcolarne le distanze lungo gli assi ( $\Delta X$ ,  $\Delta Y$ ) e quindi, mediante una semplice operazione trigonometrica, individuare la direzione della congiungente. A questo punto si procede con una normale navigazione in controllo di direzione, con l'ausilio della bussola, assumendo come setpoint la direzione calcolata.

Col procedere del moto lo spostamento del punto corrente, restituito dal GPS sulla Topic 'fix', fa sì che cambi la direzione della congiungente col bersaglio. Si ricorre quindi ad un algoritmo iterativo, in cui i passi sopra descritti vengono ripetuti – assumendo ogni volta il punto corrente come punto di partenza – fino al raggiungimento del punto di arrivo. L'eventuale raggiungimento viene verificato ad ogni iterazione, e comunicato a MMI sulla Topic 'feedback'.

Parametri del suddetto algoritmo sono gli errori massimi accettabili lungo X e Y e il periodo dell'iterazione. I primi definiscono un rettangolo intorno al punto di arrivo, all'interno del quale il bersaglio si considera raggiunto; la minima ampiezza possibile del rettangolo dipende dalla precisione del GPS.

Il periodo di iterazione deve tener conto dell'ampiezza dell'errore ammesso (il rettangolo appena definito) e della velocità di Venus: un'iterazione troppo lenta in presenza di elevata precisione richiesta ed elevata velocità potrebbe causare il 'salto' del bersaglio (veicolo che attraversa il rettangolo senza che la posizione venga rilevata; vedi commento a **Fig. 6.3**).

Per diminuire la probabilità di una tale evenienza, è stata introdotta un'ulteriore fase nell'avvicinamento: all'arrivo del Venus in un intorno significativamente più ampio del precedente, la velocità viene abbassata portando la potenza del motore al 50% (**Fig. 6.2**).

La conversione delle coordinate geografiche (GPS) in coordinate Gaussiane, effettuata ad ogni iterazione, è realizzata mediante un algoritmo ricavato da [4].

La temporizzazione del ciclo è affidata ad un timer dedicato, 'gpsLoopTim'.

Le coordinate geografiche e le distanze lungo gli assi X e Y dal bersaglio vengono registrate ad ogni iterazione, eventualmente insieme alle indicazioni 'posizione prossima' e 'posizione raggiunta'. La registrazione è soggetta alla verifica della quota di navigazione (dev'essere in superficie), che viene letta dalla Topic 'quota', sulla quale è stata scritta dal nodo PID.

La terza modalità realizza l'implementazione della gestione missioni, così come descritta nel §3.3.

I tratti in cui la missione è suddivisa vengono trasmessi da MMI, in sequenza, sulla Topic 'coda'. Si ricorda qui, per comodità, che possono attualmente essere di uno dei seguenti tipi:

- navigazione (generica) (bussola, profundimetro)
- navigazione in superficie (bussola)
- navigazione gps (bussola, GPS)

È previsto inoltre un tratto 'fittizio' di stop, che consiste nell'arresto e nel reset di tutti gli attuatori (**App. 1**).

I parametri associati a ciascun tratto sono:

- numero d'ordine progressivo
- numero totale tratti costituenti la missione
- tipo del tratto
- durata (sec)
- direzione desiderata (°)
- profondità desiderata (m)
- potenza motore (%)
- latitudine bersaglio (° sessadecimali)
- longitudine bersaglio (° sessadecimali)
- richiesta foto/video
- numero f/v
- durata video (sec)
- intervallo fra f/v consecutivi (sec)
- ritardo della prima acquisizione dall'inizio tratto (sec)
- angolo di pitch desiderato (°)

Appena completata la trasmissione da parte di MMI viene effettuato il controllo di congruenza sui tratti trasmessi. L'eventuale esito negativo causa l'annullamento della missione, e viene comunicato a MMI, sulla Topic 'feedback', perché provveda alla ritrasmissione.

In caso di esito positivo i tratti trasmessi vengono processati in sequenza. Nel primo tipo (navigazione 'generica') vengono impostati entrambi i controlli di direzione e profondità, inviando al nodo PID, sulla Topic 'coda', i comandi per i rispettivi timoni (\*), seguiti dai setpoint. Nel secondo tipo (navigazione in superficie) viene impostato il solo controllo di direzione, mentre nel terzo tipo (navigazione gps) viene direttamente lanciata la routine che implementa la navigazione con GPS, descritta al punto precedente.

Nel primo caso (la navigazione ‘generica’ è l’unica in cui sia prevista l’immersione di Venus) il nodo GPS viene arrestato, per essere successivamente riavviato all’inizio di un qualsiasi tratto di superficie. Il relativo comando è inviato sulla Topic ‘testa’.

Nel secondo e terzo caso (navigazione obbligata in superficie) i timoni di profondità vengono automaticamente posti nella configurazione ‘a barchetta’ (anteriori orientati verso l’alto, posteriori verso il basso, inclinazione prefissata – attualmente 15°), che conferisce al veicolo un assetto dinamico tale da impedire immersioni indesiderate e cavitazione dell’elica (**Cap. 6**), oltre ad una buona visibilità per le antenne WiFi e GPS.

Il passo successivo consiste nell’avvio del timer ‘navStepTim’, che regola la durata del tratto. Il timer viene impiegato anche nei tratti GPS, nonostante questi abbiano, nel bersaglio, un preciso limite spaziale. Si tratta di una misura di sicurezza per eventuali casi di malfunzionamento del sensore GPS o di errata impostazione delle coordinate del bersaglio.

Solo a questo punto viene inviato, nel caso la missione preveda riprese foto/video, l’apposito segnale di sincronismo al nodo Video, sulla Topic ‘syncfv’ (§4.3).

Viene infine inviato, sulla Topic ‘motore’, il comando di avvio motore, insieme alla potenza impostata.

Il tratto ‘fittizio’ di stop, inserito al termine della sequenza della missione, provvede all’arresto ordinato dei controlli attivi (direzione e profondità) e del motore, nonché al reset (‘zero’) dei timoni. I relativi comandi vengono trasmessi rispettivamente sulle Topic ‘coda’, ‘motore’ e ‘timoni’.

Una procedura analoga viene comunque effettuata automaticamente (anche se il tratto di stop non è stato inserito) a fine missione, subito dopo l’esecuzione dell’ultimo tratto previsto, per ragioni di sicurezza.

(\*) I comandi che attuano una navigazione in controllo di direzione o profondità possono anche essere inviati ‘manualmente’ (al di fuori di una missione) da MMI (**Cap. 5**). La navigazione così impostata è comunque soggetta ai timer di sicurezza descritti in §4.9.



## 4.11 PID

<b>Input:</b>	Topic 'bussola'	
	Formato:	std_msgs::Float32
	Topic 'pressostato'	
	Formato:	std_msgs::Float32
	Topic 'imu'	
	Formato:	std_msgs::String
	Topic 'sonar'	
	Formato:	4 float in std_msgs::Float32MultiArray
	Topic 'coda'	
	Formato:	std_msgs::String
<b>Output:</b>	Topic 'motore'	
	Formato:	std_msgs::Int16
	Topic 'timoni'	
	Formato:	std_msgs::Int16
	Topic 'quota'	
	Formato:	std_msgs::Float32
<b>Log file:</b>	buss_nav.txt	- setpoint bussola
	prof_nav.txt	- setpoint profondimetro/parametri navigazione
	cntr_log.txt	- quote centro geometrico
<b>MMI:</b>	max_ang_prf	- massima inclinazione timoni profondità
	reg_rngpc	- range regolazione
	const_prf	- compensazione costante spinta positiva
	pit_stp_sat	- setpoint pitch fasi immersione/emersione
	k_vvl	- k velocità verticale (termine derivativo PID)
	k_ass	- k controllo assetto
	prf_range	- prof. massima raggiungibile
	log_lev	- livello log
	log_frq	- frequenza log

Il ruolo del nodo PID nel controllo di Venus è stato illustrato in §3.3.

Il nodo, dotato di proprio interprete dei comandi, rimane in ascolto sulla Topic 'coda'.

Oltre ai soliti comandi per l'arresto nodo, la ricezione dei parametri modificabili da MMI e la gestione del log dati, quelli specifici per il controllo sono relativi alle seguenti modalità:

- Controllo di direzione
- Controllo di profondità

Nel nodo è attivo un loop continuo, nel quale vengono effettivamente calcolate le leggi di controllo.

Come discusso precedentemente, i comandi di direzione e di profondità possono essere ricevuti direttamente da MMI o come parte di una missione. Insieme ai comandi vengono trasmessi, rispettivamente, i setpoint di direzione e di profondità, quest'ultimo insieme al setpoint di assetto e al coefficiente di proporzionalità della legge di controllo.

I comandi attivano, rispettivamente, gli algoritmi di controllo di direzione e profondità; nel secondo caso viene anche avviato il timer di sicurezza 'stopPrfTim', che, analogamente ai timer descritti in §4.9, imposta un limite temporale al singolo tratto in navigazione subacquea.

Il controllo di direzione consta del solo termine proporzionale.

L'errore di direzione, pari alla differenza fra il relativo setpoint e la lettura della bussola (Topic 'bussola'), viene ridotto, in valore assoluto, a meno di 180°.

L'angolo dei timoni di direzione viene quindi impostato ad una frazione del massimo angolo consentito ('MAX\_ANG\_TIM', predefinito); la frazione è pari al rapporto fra l'errore e 180°.

L'angolo così calcolato viene quindi trasmesso al nodo Timoni, sulla Topic 'timoni'.

Il controllo di profondità è più complesso. Prima di descriverlo sono necessarie alcune premesse.

- Il tipo di missioni per cui Venus è stato progettato – senza volere, per questo, escludere differenti futuri sviluppi – prevede tipicamente l'effettuazione di percorsi ad una certa quota d'immersione, lungo i quali effettuare operazioni di ricognizione o campionamento (**Cap. 1**). In questa ottica le fasi di immersione alla quota prescelta e di riemersione a fine missione non richiedono profili particolari, e sono state sviluppate, tenendo conto semplicemente della rapidità e dell'efficienza, come tratti ad assetto costante.
- Caso particolare della missione-tipo è la navigazione a 'contatto' col fondale, che ha generalmente lo scopo di effettuare una ricognizione con riprese foto/video. In questo caso è richiesto il mantenimento di una distanza costante dal fondo, per ovvi motivi di sicurezza; per consentire la migliore riuscita delle riprese anche in presenza di acque non particolarmente limpide, tale distanza dovrà essere la minore possibile compatibilmente con le citate esigenze di sicurezza. Come accennato in §3.3 il mantenimento della distanza minima dal fondo è ottenuto mediante modifica continua del setpoint di profondità.
- È stata presa in considerazione la possibilità che l'assetto statico positivo di Venus (§2.1) possa portare in saturazione il controllo, diminuendone la sensibilità. Per questo è stato previsto un termine costante di compensazione della spinta di Archimede, da applicare in caso di necessità.

- Va infine notato che vengono controllati sia la quota baricentrica del Venus che il suo assetto di marcia (angolo di beccheggio, o pitch); quest'ultimo può influenzare, in particolare, la buona riuscita delle riprese foto/video (un'inclinazione anche lievemente positiva non consente di effettuare riprese del fondale).

Il primo passo dell'algoritmo di calcolo della legge di controllo di profondità è la verifica del setpoint.

La profondità del fondale è calcolata, ad ogni ciclo, sommando i dati della profondità del Venus e della distanza di Venus dal fondo, pubblicati rispettivamente dai nodi Pressostato e Sonar sulle Topic 'pressostato' e 'sonar'. Se la differenza fra le quote del setpoint e del fondale è inferiore alla distanza di sicurezza, il setpoint viene eguagliato a quest'ultima. Se nel ciclo successivo la profondità del fondale aumenta, il setpoint viene riportato verso il valore originariamente impostato – non bruscamente, ma in un certo numero di iterazioni.

Vengono così gestite con un unico algoritmo le diverse modalità di navigazione (in quota/sul fondo, §3.3), comprendendo il caso di un imprevisto innalzamento del fondale durante una navigazione a quota costante. L'unica situazione non controllabile è quella di innalzamento brusco del fondale (ostacolo frontale), per cui bisognerà ricorrere ad algoritmi di obstacle avoidance.

Si passa quindi al calcolo degli errori:

- di profondità, pari alla differenza fra il setpoint effettivo e la lettura del profondimetro (Topic 'pressostato'), corretta mediante l'angolo di pitch (Topic 'imu') per ottenere la quota del baricentro geometrico;
- di assetto, pari alla differenza fra il relativo setpoint e l'angolo di pitch.

Il calcolo della legge di controllo è differente secondo la fase dell'immersione, come enunciato in premessa.

Fase a quota costante o 'a contatto' col fondo: l'immersione è considerata in questa fase quando Venus si trova in un intorno della quota del setpoint (c. d. 'intervallo di regolazione') scelto, arbitrariamente (\*), come l'inverso del coefficiente di proporzionalità della legge di controllo. In questa condizione viene dapprima calcolata la componente proporzionale del controllo della quota: l'angolo delle due coppie di timoni di profondità viene provvisoriamente eguagliato al prodotto del massimo angolo consentito ('MAX\_ANG\_PRF', predefinito) per l'errore di profondità, ulteriormente moltiplicato per il coefficiente di proporzionalità.

Per come è stato scelto (\*) l'intorno di calcolo, l'angolo così determinato è minore o uguale, in modulo, all'angolo massimo consentito.

Viene quindi aggiunta la componente derivativa (termine di smorzamento), pari al prodotto fra il relativo coefficiente di proporzionalità e la velocità verticale (Topic 'pressostato') cambiata di segno.

Nel caso venga rilevata la tendenza, dovuta alla spinta positiva, a permanere stabilmente al di sopra della quota di setpoint, viene aggiunto, come discusso in premessa, un termine costante di compensazione.

A parte viene calcolato il contributo relativo al mantenimento dell'assetto di marcia, pari al prodotto dell'errore di assetto per il relativo coefficiente di proporzionalità. Tale contributo viene applicato preferibilmente ai soli timoni anteriori, in quanto se ne è constatata la maggiore efficacia nella regolazione dell'assetto; in caso di

superamento dell'angolo massimo, viene invece ripartito fra gli anteriori e i posteriori.

Fasi di discesa e risalita: al di fuori dell'intervallo di regolazione sopra definito si adotta, come discusso in premessa, un controllo ad assetto costante.

Il contributo relativo al mantenimento della quota viene quindi azzerato, mentre quello relativo all'assetto è calcolato come il precedente, ma considerando un diverso setpoint, predefinito e modificabile da MMI.

L'ultimo passo nel calcolo della legge di controllo consiste nella verifica dell'inclinazione di ciascuna coppia di timoni, dovuta alla somma dei vari contributi: in caso risulti superiore al valore massimo consentito viene uguagliata a quest'ultimo (situazione di saturazione).

Le inclinazioni dei timoni così calcolate vengono infine inviate, a coppie (anteriore – posteriore), al nodo Timoni, sulla Topic 'timoni'.

## 4.12 Motore

<b>Input:</b>	Topic 'motore'	
	Formato:	std_msgs::Int16
	Topic 'strumnav'	
	Formato:	std_msgs::Int16
<b>Output:</b>	motore Maxon	- interfaccia i2c
<b>Log file:</b>	mot_log.txt	- potenza impostata [%]
<b>MMI:</b>	i2c_offset	- offset (potenza motore)
	i2c_range	- range di variazione (potenza motore)
	log_lev	- livello log
	log_frq	- frequenza log

Il nodo Motore rimane in ascolto sulla Topic 'motore', da cui legge la potenza impostata (setpoint), espressa come percentuale della potenza massima.

Rimane inoltre in ascolto sulla Topic 'strumnav', sulla quale riceve i comandi operativi di avvio/arresto nodo e gestione della registrazione dei dati di funzionamento (log su file/standard output).

Il SW è basato sul driver implementato da G. Cupertino [3].

All'arrivo di un nuovo setpoint vengono trasmessi al motore i comandi di abilitazione/disabilitazione, blocco/sblocco freni, marcia avanti/indietro/folle e infine la potenza (%) da impostare.

i2c\_offset e i2c\_range sono parametri numerici usati dal driver del motore. Non vanno di norma modificati.

#### 4.13 Timoni

<b>Input:</b>	Topic 'timoni'	
	Formato:	std_msgs::Int16
	Topic 'strumnav'	
	Formato:	std_msgs::Int16
<b>Output:</b>	step motor pinne	- interfaccia RS232
<b>Log file:</b>	pinne_log.txt	- inclinazioni impostate (°)
<b>MMI:</b>	log_lev	- livello log
	log_frq	- frequenza log

Il nodo Timoni rimane in ascolto sulla Topic 'timoni', da cui legge, in sequenza, le posizioni impostate (setpoint) delle singole pinne.

Rimane inoltre in ascolto sulla Topic 'strumnav', sulla quale riceve i comandi operativi di avvio/arresto nodo e gestione della registrazione dei dati di funzionamento (log su file/standard output).

Il SW è basato sul driver implementato da G. Cupertino [3].

All'arrivo di un nuovo blocco di setpoint vengono trasmessi agli attuatori delle pinne i comandi di abilitazione/disabilitazione e gli angoli da impostare.

È implementato un controllo di congruenza, per cui il set di dati è considerato valido, e impiegato, solo alla lettura di sei valori validi consecutivi.

## 5. Interfaccia uomo-macchina

L'interfaccia uomo-macchina (Man Machine Interface, MMI) è lo strumento che consente all'operatore il controllo completo dell'attività del robot.

Nell'attuale fase di sviluppo il solo avvio a freddo della macchina è ancora demandato a comandi diretti impartiti da terminali (shell) aperti sulla Stazione di Controllo; a regime tale fase verrà inglobata nelle procedure automatiche di avviamento [3].

MMI consente in particolare di:

- avviare/arrestare i Nodi su Venus
- gestire il robot in teleguida (pilotaggio, acquisizioni foto/video)
- visualizzare in tempo reale i sensori di bordo
- impostare e lanciare le missioni
- gestire l'acquisizione e il download dei dati di log

L'interfaccia è basata sulla libreria open source Qt e su ROS (§3.1): questa scelta ha consentito di configurarla come un nodo ROS, con evidenti vantaggi nella comunicazione con i Nodi del Sistema di Controllo di Venus.

È installata su un PC portatile dotato di link WiFi (Stazione di Controllo, §2.6). La comunicazione fra MMI e Venus è pertanto possibile, attualmente, solo in superficie. Il link ottico più volte citato consentirà, una volta completato, la comunicazione subacquea.

È possibile collegare un joystick alla Stazione di Controllo, per la gestione della teleguida.

L'interfaccia si presenta all'operatore suddivisa in due sezioni, ciascuna visualizzata in una propria finestra. Nella prima sono raggruppati i comandi principali, la teleguida, la gestione diretta delle acquisizioni foto/video e una finestra alfanumerica in cui vengono visualizzati i comandi effettivamente inviati a Venus.

La seconda è dedicata ai sensori, alla navigazione 'diretta' (direzione, profondità, GPS – vedi nota in fondo a §4.10) e alla gestione del log dati.

Diverse 'dialog box' (finestre a scomparsa) possono essere aperte per la gestione di particolari comandi, mediante i relativi pulsanti.

### **Comandi**

La finestra principale è mostrata in **Fig. 5.1**.

Partendo dall'alto, e da sinistra verso destra, si hanno:

- pulsante di connessione allo specifico Venus (attualmente V2 e V3 - **Cap. 1**); 'localhost' per test SW dell'interfaccia, senza collegamenti esterni

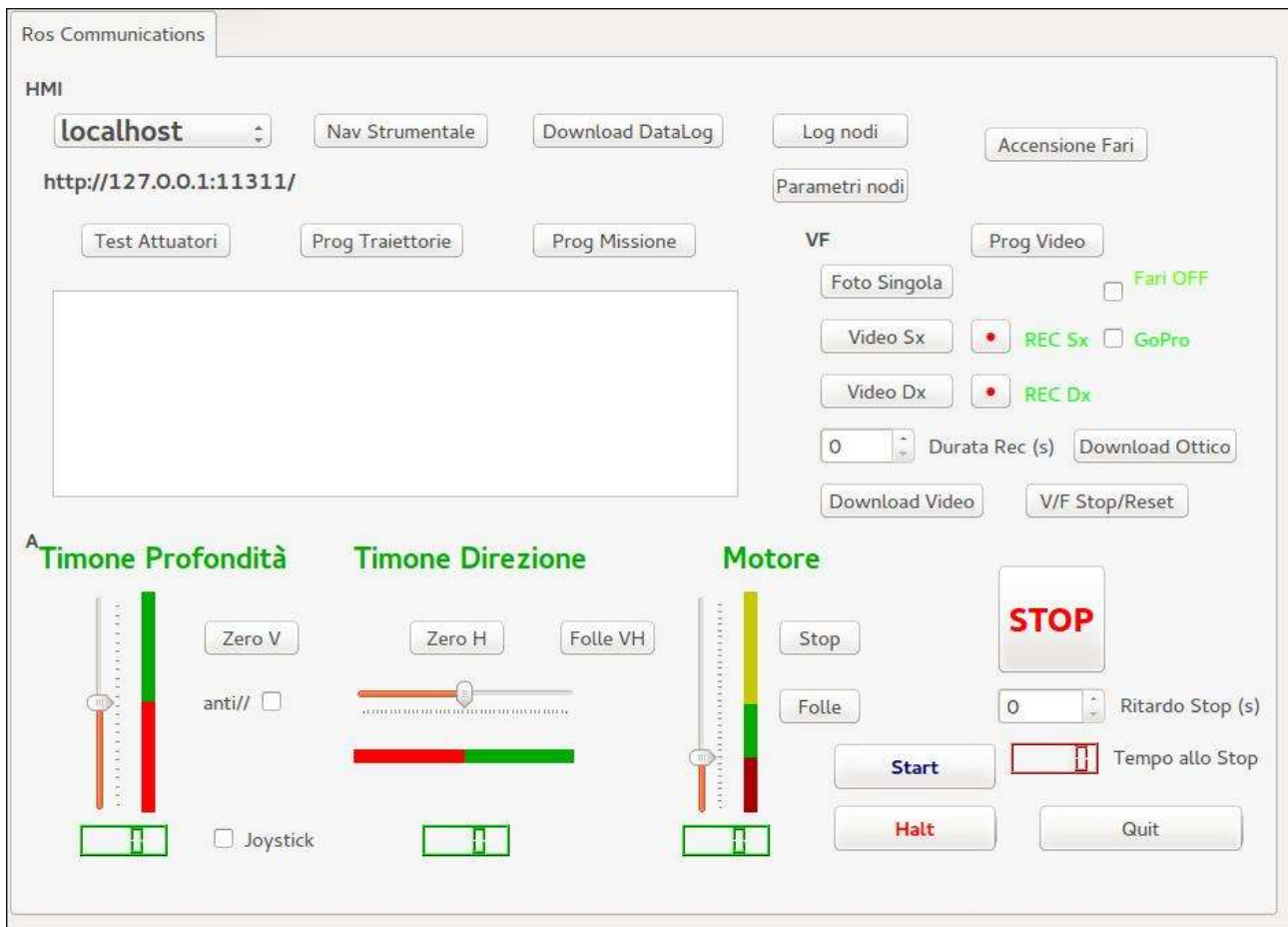


Figura 5.1

- pulsante di avvio della finestra sensori ('Nav Strumentale')
- pulsante per il download dei dati da standard output (§3.4)
- pulsante per i parametri di log dati (§3.4)
- pulsante per i parametri specifici dei Nodi modificabili da MMI (Cap. 4)
- pulsante per il comando diretto dei fari anteriori
- pulsante per il test pinne (§4.9)
- pulsante per la programmazione tratti a loop aperto (§4.10)
- pulsante per la programmazione missioni (§4.10)
- finestra per la visualizzazione delle stringhe di comandi e parametri inviati a Venus
- gruppo comandi diretti video: avvio riprese (registrazione o streaming), download (anche su canale ottico - §4.5), reset memoria (cancellazione dei file foto/video)
- gruppo comandi teleguida: i timoni di profondità/direzione e il motore vengono comandati dai cursori ('barre'), mediante mouse, o dal joystick collegato al PC. L'azionamento della barra Motore provoca l'avvio del timer di sicurezza 'emrgTim' (§4.9). Sono presenti pulsanti per il riallineamento allo 'zero' dei timoni e l'arresto motore. I timoni di profondità anteriori e posteriori possono funzionare in configurazione parallela o antiparallela (§4.10)



- gruppo comandi avviamento/arresto: comprende i pulsanti di avvio/arresto Nodi, Stop di emergenza e arresto di MMI
  - Start/Halt: i nodi possono essere avviati/arrestati indipendentemente sulla Testa e sulla Coda. È possibile arrestare anche i nodi Listener, che partono automaticamente all'avvio di Venus (§4.4 e §4.9)
  - Stop: ha una doppia modalità di funzionamento: se il campo 'Ritardo Stop' viene lasciato in bianco, provoca l'arresto rapido di Venus (spegnimento motore e tutte le pinne a 90°); se viceversa il 'Ritardo Stop' viene impostato (in sec), avvia il timer di sicurezza 'stopTim' (§4.9), provvedendo all'arresto del motore allo spirare del timer. La mancata impostazione del 'Ritardo Stop' impedisce l'invio al robot del comando di start, quindi la partenza del veicolo (§4.9), avvisando l'operatore mediante apposito messaggio di warning.
  - Quit: arresto di MMI e chiusura della finestra

Vengono di seguito mostrate due istanze della finestra programmazione missioni, che si apre col pulsante 'Prog. Missione' della finestra principale.

Il tipo di tratto da impostare (§3.3 e App. 1) viene scelto mediante il pulsante 'Nuovo Step'.

Figura 5.2

In Fig. 5.2 l'immissione di un tratto di navigazione 'generica'; in pratica si usa questa finestra per programmare

un tratto in immersione, mentre per i tratti in superficie è disponibile l'apposita schermata. Oltre ai parametri necessari alla navigazione (si noti la profondità, attualmente limitata a 10 m – **Cap. 6**) è possibile impostare quelli relativi alle eventuali riprese foto/video.

Prima di passare all'immissione del tratto successivo, è necessario confermare col pulsante 'Conferma Step', altrimenti il tratto attuale viene scartato. Il pulsante 'Reset' azzerla la schermata in corso.

The screenshot shows a software interface for mission programming. At the top, it says 'Programmazione Missione'. Below this, there are several sections: 'Step inseriti' with a text box containing the number '2'; 'Nuovo Step' with a dropdown menu currently showing 'Nav Gps'; 'Navigazione' with two numeric input fields for 'Durata (0 - 3600 sec)' and 'Potenza (0 - 100%)', both set to '0'; 'GPS' with four numeric input fields for latitude and longitude (all set to '0') and checkboxes for 'N/S' and 'E/W', both of which are checked; and a 'File' section with a 'Nome file:' text box, a 'Scrittura su file' button, and 'Cancel' and 'OK' buttons at the bottom right. Additionally, there are 'Conferma Step' and 'Reset Step' buttons on the right side of the interface.

**Figura 5.3**

In **Fig. 5.3** l'immissione di un tratto di navigazione gps. Latitudine e longitudine vengono immesse in gradi sessagesimali, con l'aggiunta dei centesimi di secondo.

Una volta completata l'immissione dei tratti, è possibile avviare la missione (pulsante 'OK') o registrarla su file, dopo averne indicato il nome, e poi, eventualmente, lanciarla.

Per lanciare una missione già registrata su file va scelta l'opzione 'file' (**App. 1**) dal pulsante 'Nuovo Step'.

## **Sensori**

La finestra sensori è mostrata in **Fig. 5.4**.

È suddivisa in cinque settori, uno per ogni strumento, più una sezione in basso per la gestione del log dei dati. Oltre alla visualizzazione delle letture dei sensori è possibile, nei primi tre settori, impostare delle navigazioni

rispettivamente in controllo di direzione, di punto d'arrivo e di profondità (pulsanti 'StartNav'/'StopNav' – vedi anche nota (\*) in fondo a §4.10).

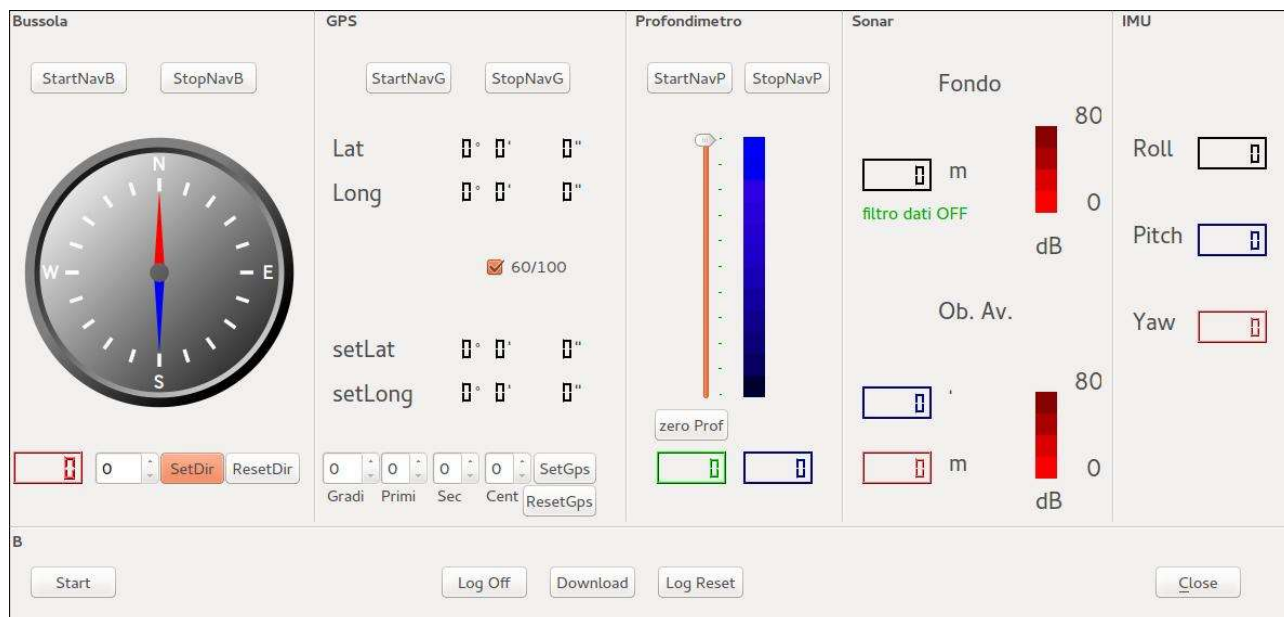


Figura 5.4

L'impostazione del tempo di navigazione e della potenza del motore vengono sempre effettuate nella finestra principale, così come l'avvio del motore. Si tratta di una funzionalità in pratica non più impiegata, che ha avuto la funzione di collaudare i vari tipi di navigazione (le missioni impiegano i comandi sviluppati in questa modalità – App. 2).

L'effettiva trasmissione dei dati da parte di Venus dipende, al solito, dal link disponibile.

Le coordinate geografiche nel settore GPS possono essere visualizzate sia in gradi sessagesimali che in gradi sessadecimali.

Nel settore Sonar i dati di distanza dal fondo e di obstacle avoidance sono mostrati da due indicatori distinti. È visualizzata anche l'intensità del segnale, in decibel.

Nella sezione in basso, da sinistra a destra, si hanno i pulsanti:

- avvio 'strumenti' (avvio visualizzazione dati)
- avvio/arresto registrazione dati su Venus (file di log)
- download file di log -> Stazione di Controllo
- cancellazione file di log su Venus
- chiusura finestra sensori

## 6. Collaudi – prove di navigazione

Le prove di navigazione e di collaudo del Sistema di Controllo sono state effettuate nel lago di Bracciano, in località Vigna di Valle, nello specchio d'acqua antistante la base dell'Aeronautica Militare "Luigi Bourlot", presso la quale il team ENEA è stato cortesemente ospitato e supportato per gli aspetti logistici.

Le prove hanno dapprima riguardato la determinazione del corretto assetto statico e la capacità di effettuare brevi immersioni, con verifica, in particolare, dell'impermeabilità del veicolo. Tutte le prove di questo tipo si sono svolte in teleguida (§3.3) o in loop aperto (§4.10).

Si è quindi passati ai test sulla gestione delle missioni, con lo scopo di verificare la bontà del metodo di suddivisione in tratti (§3.3), la corretta concatenazione nell'esecuzione degli stessi e, più in generale, l'adeguatezza del sistema Venus allo svolgimento dei compiti per cui è stato pensato.

Nel primo gruppo di prove si è constatata la notevole sensibilità dell'assetto a variazioni anche piccole della ripartizione dei pesi fra prua e poppa. Si noti che in una fase di sviluppo come l'attuale gli interventi sul veicolo per manutenzione o piccole modifiche sono frequenti, e possono comportare, appunto, variazioni nella distribuzione dei pesi.

Si è visto che è preferibile un assetto leggermente inclinato verso prua, che favorisce l'entrata in immersione; viceversa una prua troppo leggera può rendere problematica tale fase, arrivando a causare, paradossalmente, cavitazione dell'elica (quando il robot comincia a inclinarsi ma non riesce a immergersi). Si segnala in proposito che, proprio in seguito ai primi risultati sperimentali, l'elica originale (in plastica) è stata sostituita con una più efficiente (in metallo), che ha complessivamente migliorato le prestazioni di Venus.

Un assetto appurato richiede peraltro alcune cautele, in quanto può dar luogo all'immersione dello scafo anche con i timoni di profondità in posizione orizzontale (0°). Ciò può causare, in caso di teleguida, un'imprevista perdita di controllo dovuta alla impossibilità di funzionamento del link WiFi. Per fronteggiare una tale evenienza sono stati introdotti i due timer di sicurezza descritti in §4.9 e la configurazione dei timoni di profondità 'a barchetta' descritta in §4.10.

Delle problematiche relative all'impermeabilità si tratta in [2]; qui si ricorda solo che tutte le prove fin qui effettuate si sono svolte entro la profondità di 8 m, per rispettare le specifiche dei premistoppa dei timoni – ma anche per elementari considerazioni di prudenza, a fronte del rischio di danneggiare irrimediabilmente l'elettronica (test statici dello scafo vuoto [2] ne avevano dimostrato l'impermeabilità fino a 25 m).

Nel programmare la seconda serie di prove si è partiti dalla definizione di una missione-tipo, pensata in modo da simulare la generica missione per cui Venus è stato progettato (**Cap. 1**):

1. raggiungimento di un punto di coordinate geografiche (latitudine e longitudine) prefissate;
2. svolgimento delle attività previste dalla missione vera e propria;
3. rientro al punto di partenza.

Scopo dei test è la dimostrazione che una missione così impostata possa essere realizzata mediante la concatenazione di ‘tratti’ appartenenti alle tipologie definite in §3.3 e richiamate in §4.10, e precisamente:

- per i punti 1 e 3: navigazione gps;
- per il punto 2: navigazione subacquea a quota costante e/o sul fondo, eventualmente in combinazione con tratti di navigazione in superficie.

Le missioni vengono impostate in MMI, eventualmente registrate su file (cosa che ne velocizza l’avvio, specie in caso di ripetizione della stessa missione) e trasmesse a Venus.

Appena effettuato il controllo di congruenza (§3.3 e §4.10) da parte del nodo Navigatore, il Venus entra in modalità autonoma e vi permane fino al termine della missione.

L’esito della missione viene verificato mediante l’esame dei log dati (come discusso in §3.4 i Nodi registrano su file i dati salienti, man mano che questi vengono elaborati); come si vedrà i dati vengono utilmente impiegati nel tracciamento di grafici (realizzati mediante l’utility Gnuplot) che forniscono un quadro sintetico dell’andamento dei parametri di interesse. Nel caso di missioni in superficie, o di visibilità subacquea particolarmente buona, anche l’esame visivo può contribuire alla valutazione dell’esito.

La produzione dei grafici è automatizzata e può essere effettuata immediatamente dopo la fine della singola missione, permettendo così di effettuare correzioni e nuove verifiche direttamente sul campo.

Di seguito vengono illustrati, con l’ausilio dei grafici prodotti, alcuni dei ‘lanci’ effettuati nel corso delle sessioni di prova, che ben dimostrano il livello di efficacia raggiunto dal Controllo di Venus, oltre ad alcuni dei casi in cui le prove sul campo hanno contribuito direttamente alla individuazione e alla soluzione di funzionamenti non corretti.

Per comodità di lettura i grafici sono riportati trasversalmente, e i commenti raggruppati a parte.

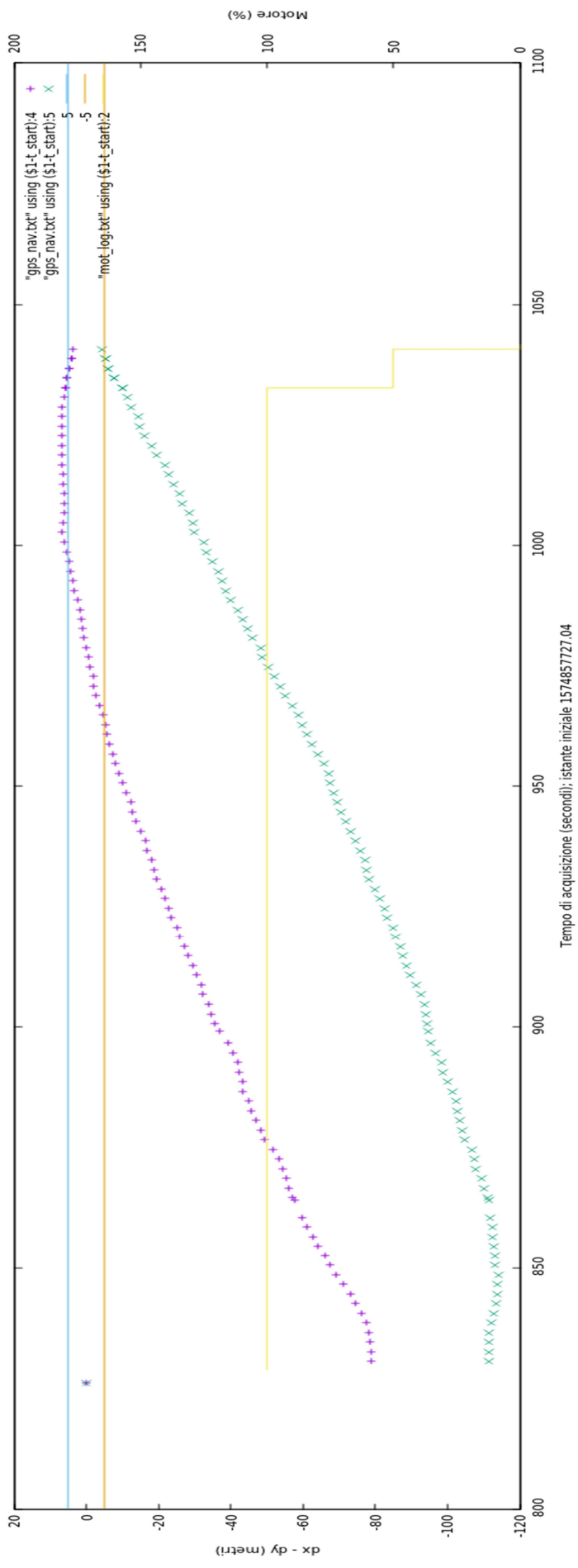
In alto a destra è indicato, a fianco di ciascuna curva, il file di log che l’ha generata.

## 6.1 Navigazione GPS

In Fig. 6.1 è mostrato l’andamento della distanza dal bersaglio (errori lungo gli assi x, y) fino a convergenza, in un lancio del 27 Novembre 2019.

In Fig. 6.2 la parte finale dello stesso lancio, in cui è evidenziato l’arrivo sul bersaglio (§4.10: errori massimi  $\Delta x = \Delta y = 5$  m; step temporale = 3 sec; motore al 50% in prossimità del bersaglio).

Per confronto, a conferma di quanto notato in §4.10, si riporta in Fig. 6.3 la parte finale di un lancio del 18 Giugno 2019, in cui la convergenza non è stata raggiunta: qui gli errori massimi erano  $\Delta x = \Delta y = 3$  m, lo step temporale 10 sec e non era previsto nessun rallentamento in prossimità del bersaglio. Il Venus ‘balla’ intorno al bersaglio senza riuscire a fermarsi (dx e dy non sono mai contemporaneamente inferiori agli errori massimi impostati): è troppo veloce in confronto al tempo di campionamento ed alla precisione richiesta.



**Figura 6.1 – Convergenza GPS**  
(27/11/2019)

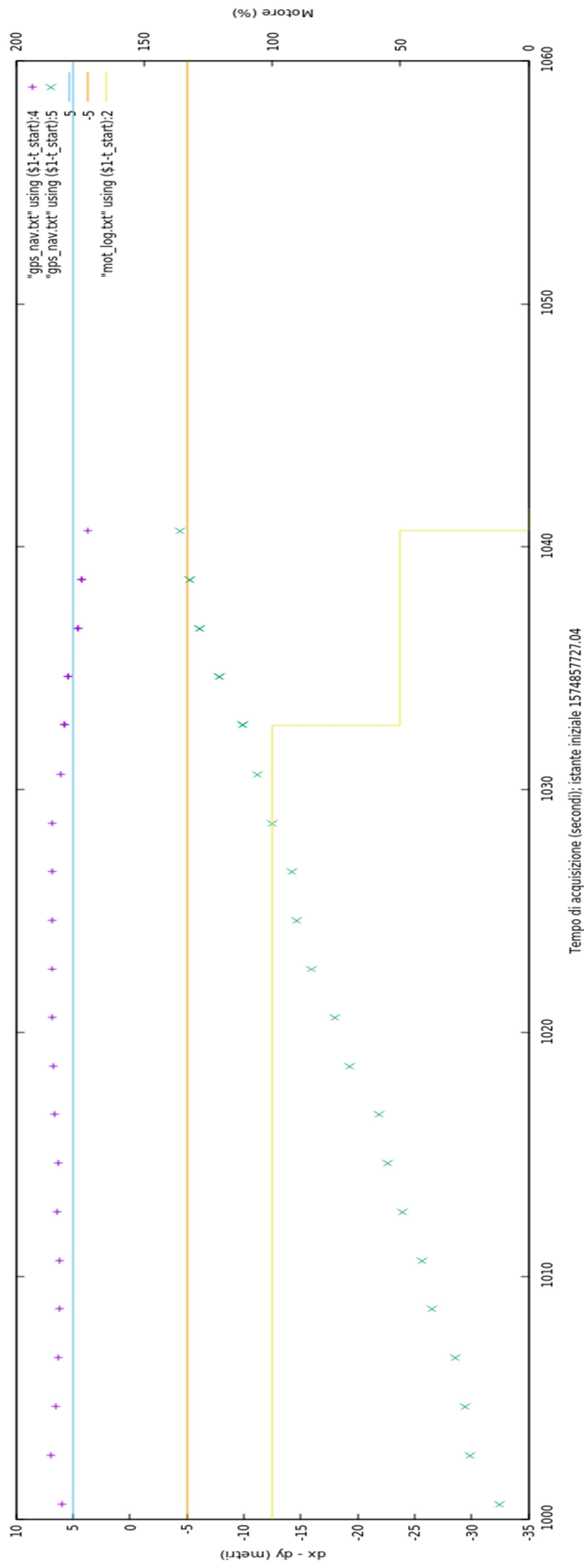


Figura 6.2 – GPS: bersaglio centrato (particolare di Fig. 6.1)

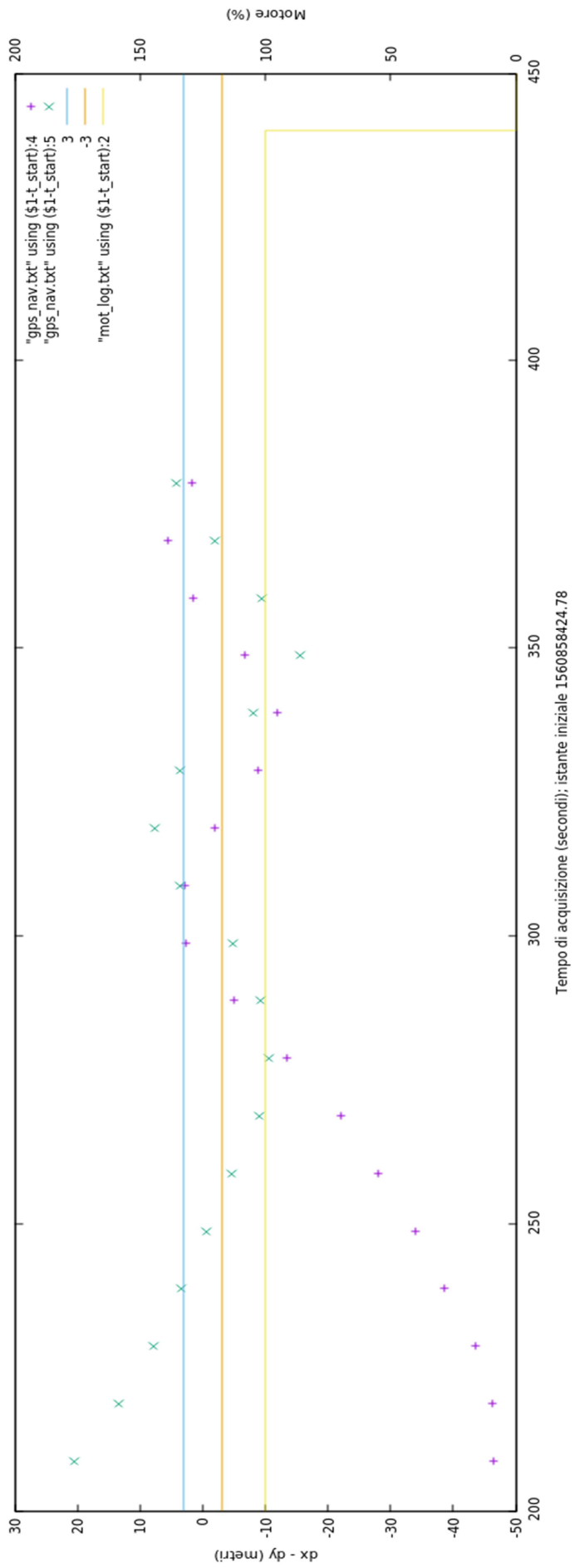


Figura 6.3 – GPS: bersaglio mancato  
(18/06/2019)



## 6.2 Navigazione in quota

In **Fig. 6.4** è illustrata una navigazione in quota, cioè a una distanza dal fondo superiore alla quota di sicurezza (18 Luglio 2019). Il setpoint di profondità (**prof\_nav:2**), impostato a -5 m, resta infatti costante per tutta la durata del tratto in immersione (60 sec). La missione è completata da un tratto con setpoint 0 m e durata 30 sec (risalita). Il profilo in quota è mostrato da **cntr\_log:2**, e evidenzia un buon inseguimento del setpoint, a dimostrazione della validità della legge di controllo.

In **Fig. 6.5** viene mostrata, sullo stesso grafico, l'azione dei timoni di profondità (anteriori: **pinne\_log:2**, posteriori: **pinne\_log:4**). Come discusso in §4.11, nelle fasi di discesa e risalita viene controllato soltanto l'angolo di pitch (beccheggio), nella fase centrale la quota e il pitch.

Nella fase centrale si possono distinguere due distinti interventi dei timoni: dapprima per 'raddrizzare' il Venus dopo la 'picchiata', poi per mantenere quota e pitch impostati (quest'ultimo pari a 10°). Come già notato, la quota viene ben mantenuta, e lo stesso si può dire, in questo caso, per l'assetto; in altri lanci il controllo dell'assetto ha dato risultati meno soddisfacenti.

Un discorso a parte merita la rilevazione del profilo del fondale (**cntr\_log:4**) che, come si nota, è alquanto discontinuo. Detto profilo è dato, come si intuisce facilmente, dalla somma delle letture del profondimetro e del sonar. Come discusso in §4.2 e mostrato in **Fig. 6.6** (particolare della missione in esame) la rilevazione del fondo da parte del sonar avviene solo al passaggio del fascio sulla verticale che ha luogo, col range di lettura attualmente impostato (90° - 210°), a intervalli di 1 e 3 sec circa. Il fondale viene calcolato, nel nodo PID, ogni 0,5 sec (la frequenza del ciclo di controllo di PID è di 2 Hz), con dati sempre 'freschi' dal profondimetro ma non, evidentemente, dal sonar: si ha così la discontinuità descritta.

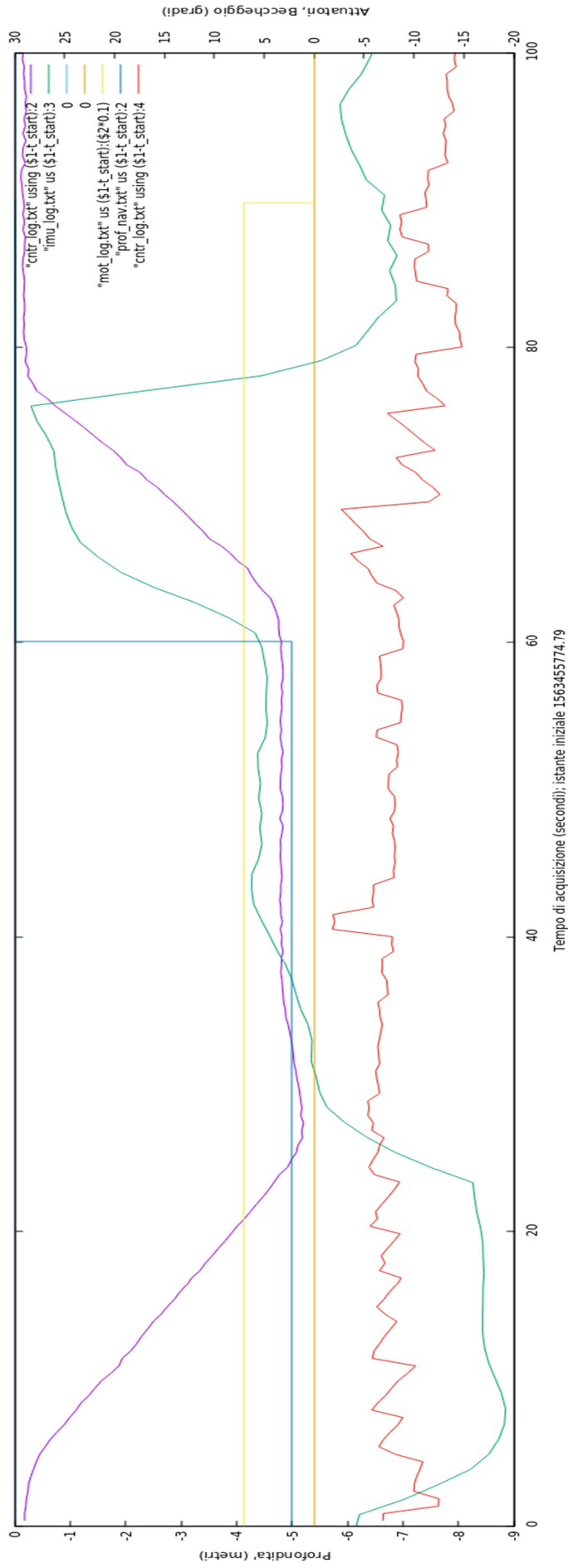
Nelle versioni successive del SW si è provveduto ad effettuare il calcolo della profondità del fondale solo in presenza di dati aggiornati del sonar, ottenendo così un profilo meglio raccordato.

Nell **Fig. 6.7** e **Fig. 6.8** sono riportate due missioni multilivello (entrambe del 16/02/2018) che differiscono in pratica solo per il valore della costante di proporzionalità della legge di controllo, pari rispettivamente a 2 e 1.25.

Si nota facilmente che nella seconda i tempi di discesa/risalita sono circa doppi (la velocità di discesa è riportata in **cntr\_log:3**).

Sempre nel secondo grafico si notano l'assenza di sovraelongazione al raggiungimento della quota di setpoint, variazioni di assetto molto contenute e il lavoro dei timoni di profondità, alquanto inferiore.

Infine, in **Fig. 6.9**, una navigazione in quota (03/07/2019) nella quale è stato incontrato un rialzo imprevisto del fondale (fra i sec 50 e 80); il setpoint di profondità (**prof\_nav:2**) viene automaticamente innalzato in modo da mantenere la distanza di sicurezza dal fondo (**cntr\_log:4**).



**Figura 6.4 – Navigazione 'in quota'**  
(18/07/2019)

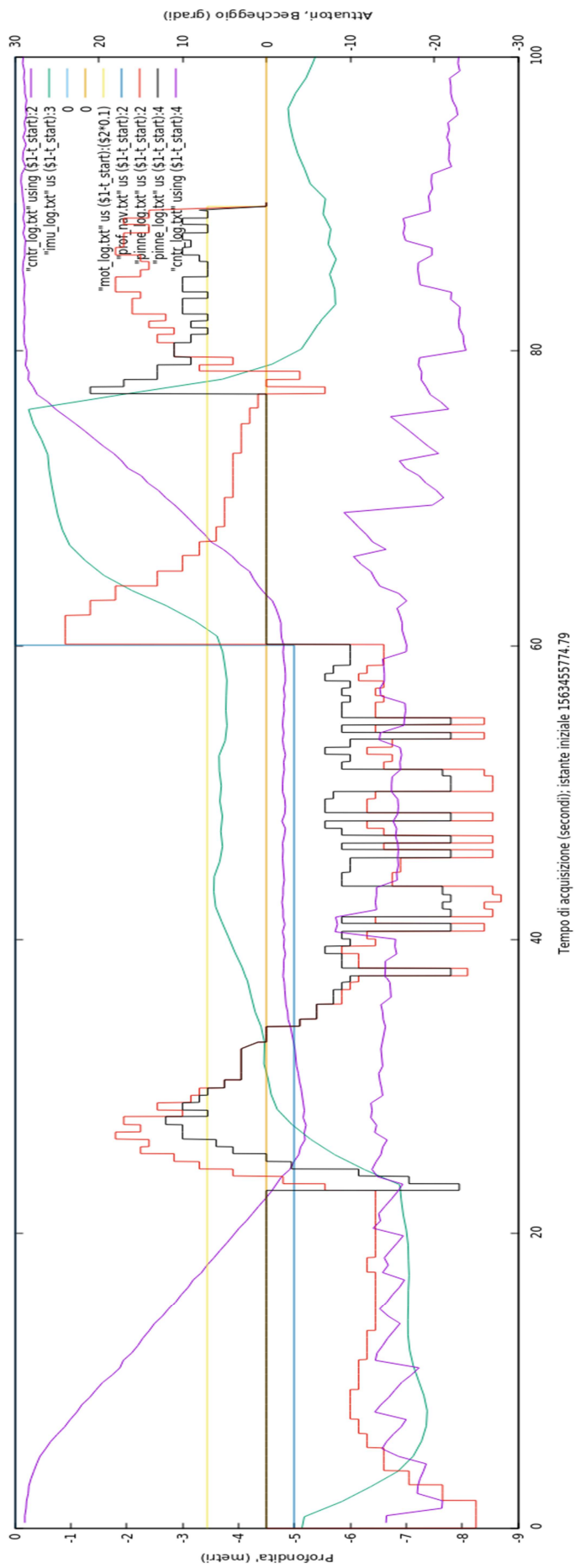


Figura 6.5 – Azione dei timoni di profondità nell'immersione di Fig. 6.4

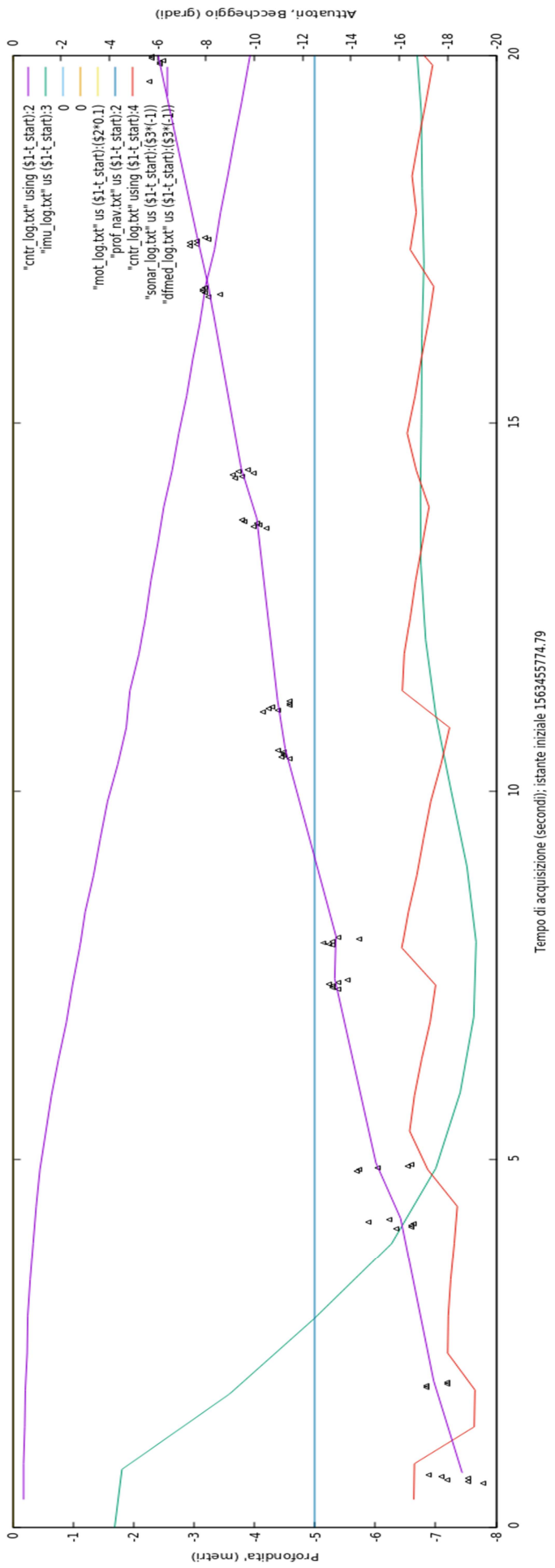
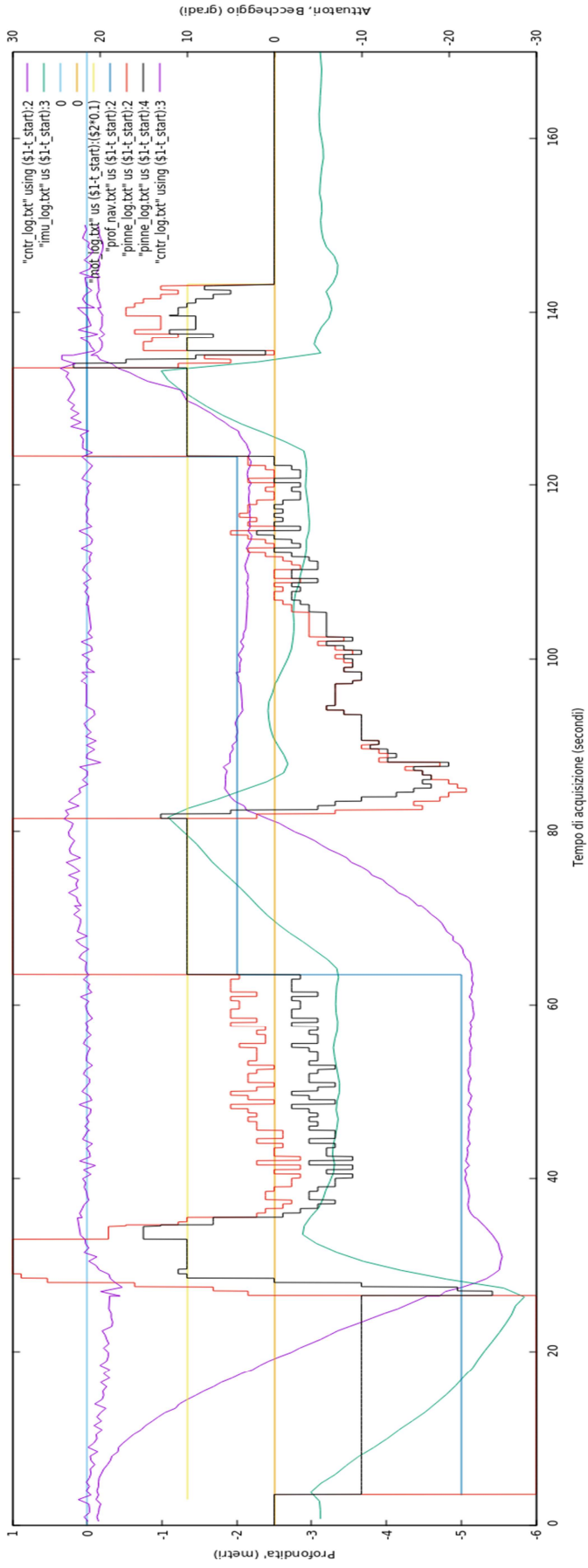


Figura 6.6 – Sonar: campionamento fondale (particolare di Fig. 6.4)



**Figura 6.7 – Multilivello,  $k_p = 2$   
(16/02/2018a)**

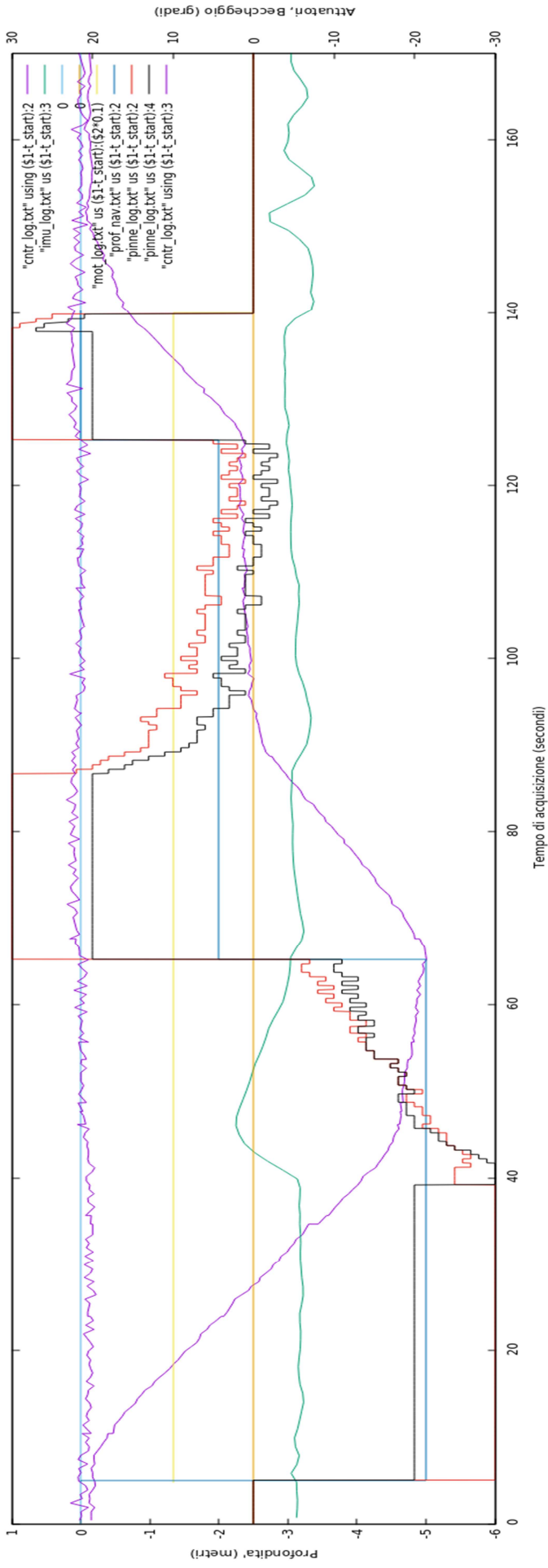


Figura 6.8 - Multilivello,  $k_p = 1,25$   
(16/02/2018b)

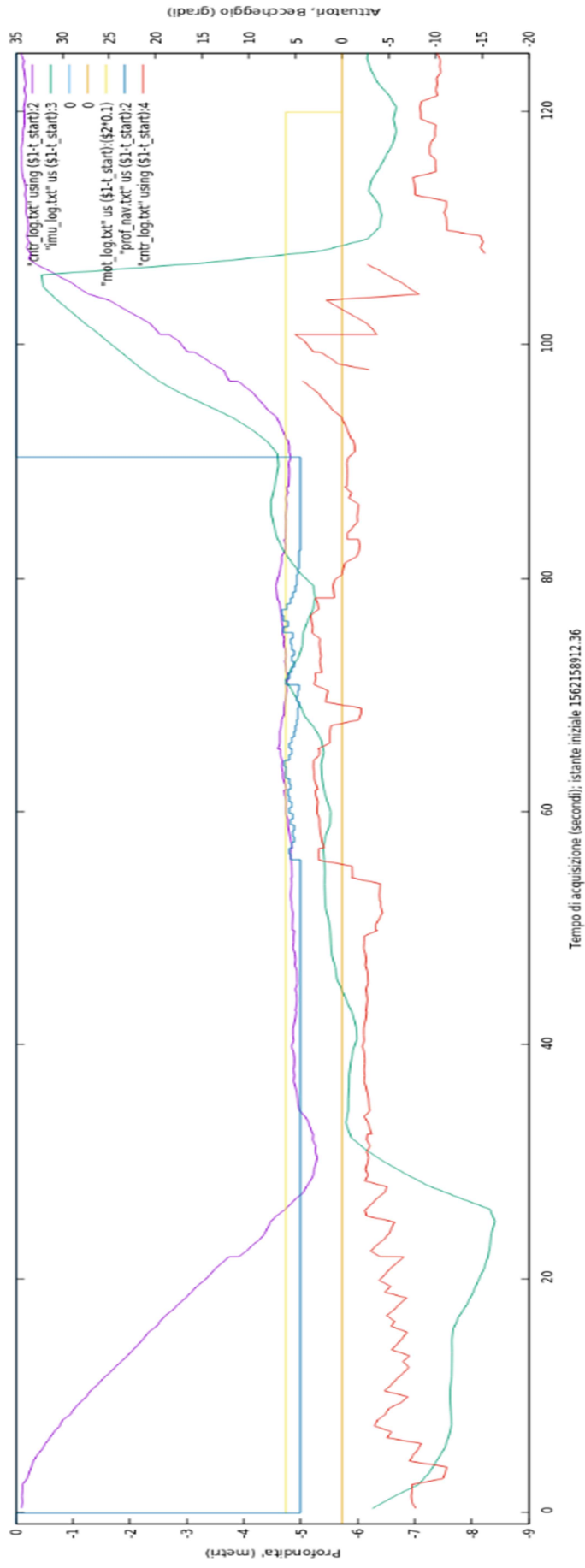


Figura 6.9 – Correzione setpoint per rialzo del fondale  
(03/07/2019)

### 6.3 Navigazione sul fondo

In **Fig. 6.10** è illustrata una navigazione (12/09/2019) in prossimità del fondale (**cntr\_log:4**), in cui il setpoint di profondità (**prof\_nav:2**) è costantemente regolato per mantenere la distanza di sicurezza.

L'interesse del grafico sta nel mostrare l'influenza, discussa in precedenza, di piccoli spostamenti di peso sull'assetto. Proprio per correggere quest'ultimo, sulla base dei test precedenti, prima del lancio era stata appesantita leggermente la prua del veicolo, ma l'intervento è stato evidentemente eccessivo. Analizzando l'andamento del pitch (**imu\_log:3**) si nota infatti che questo nella fase di navigazione sul fondo si mantiene ben al di sotto del setpoint, che era impostato a 10°, nonostante il notevole lavoro dei timoni di profondità (**pinne\_log:2** e **:4**); durante la fase di risalita si mantiene su valori inferiori alla norma (allo spegnimento del motore torna bruscamente negativo). Come conseguenza il tempo assegnato per la risalita (20 sec, normalmente sufficienti dalla quota di 5 m) non basta per la riemersione, e allo spegnimento del motore (**mot\_log:2**) Venus si trova ancora a 2 m di profondità.

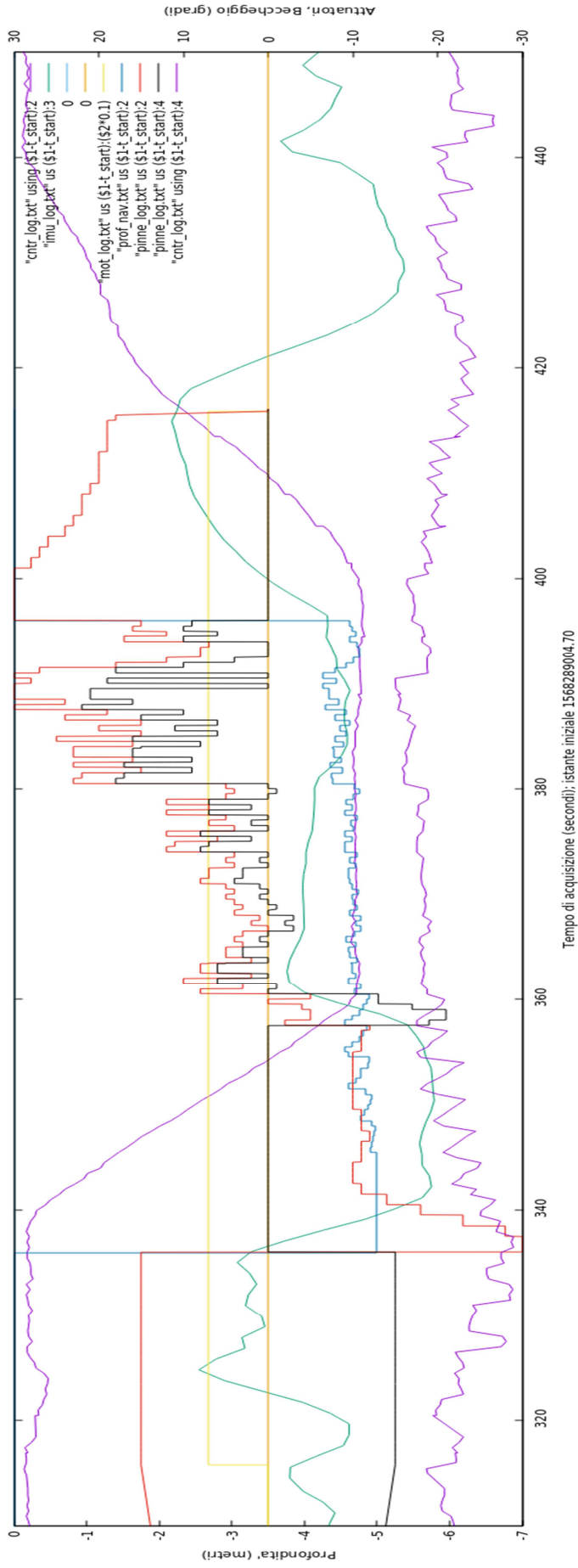
In **Fig. 6.11** è mostrata una missione (23/10/2019) del tutto analoga alla precedente, ma con una più efficace equilibratura del veicolo. Lo si nota soprattutto nei tempi e nell'andamento del pitch nella fase di risalita. Nella fase di navigazione sul fondo il pitch si mantiene invece leggermente al di sopra del setpoint, che qui è impostato a 0°; si nota, anche in questo caso, il lavoro dei timoni di profondità, che mantengono molto bene la quota impostata, mentre faticano a correggere l'assetto.

Nel grafico è riportato anche il tracciato delle misure del fondale da parte del sonar (**sonar\_log:3**), che fornisce un interessante esempio del contributo dei test all'individuazione di funzionamenti non corretti.

Oltre ai normali intervalli temporali fra le rilevazioni del fondo (passaggi del fascio sonar sulla verticale), che danno luogo alle 'seghettature' discusse nel commento alla **Fig. 6.6** e visibili anche in questo caso, si nota una lacuna ben più ampia in corrispondenza della fase di navigazione sul fondo, che causa un'errata rilevazione dello stesso. Si ha in pratica un 'dente di sega' fittizio (la situazione torna alla normalità al riprendere delle misurazioni sonar). Nel caso in oggetto ciò non ha causato problemi di sicurezza: il fondo percepito sale e il Venus lo segue, allontanandosi così da quello reale. Se viceversa fosse stata rilevata una profondità superiore a quella effettiva, il veicolo avrebbe potuto entrare in collisione col fondale, pur 'credendo' di rispettare la distanza di sicurezza.

Un comportamento del genere ('lacuna' sonar) è stato rilevato in numerosi altri lanci, sempre in corrispondenza all'effettuazione di riprese video. È stato quindi attribuito ad una 'interferenza' causata dal nodo Video nelle fasi di acquisizione (l'acquisizione video assorbe tutte le risorse HW di calcolo, bloccando di fatto gli altri nodi – vedi anche §4.4). La modifica accennata nel §4.3 – **Nota** (vedi anche **Cap. 7**) dovrebbe risolvere anche questo problema.





**Figura 6.10 – Navigazione ‘sul fondo’, prua appesantita (12/09/2019)**

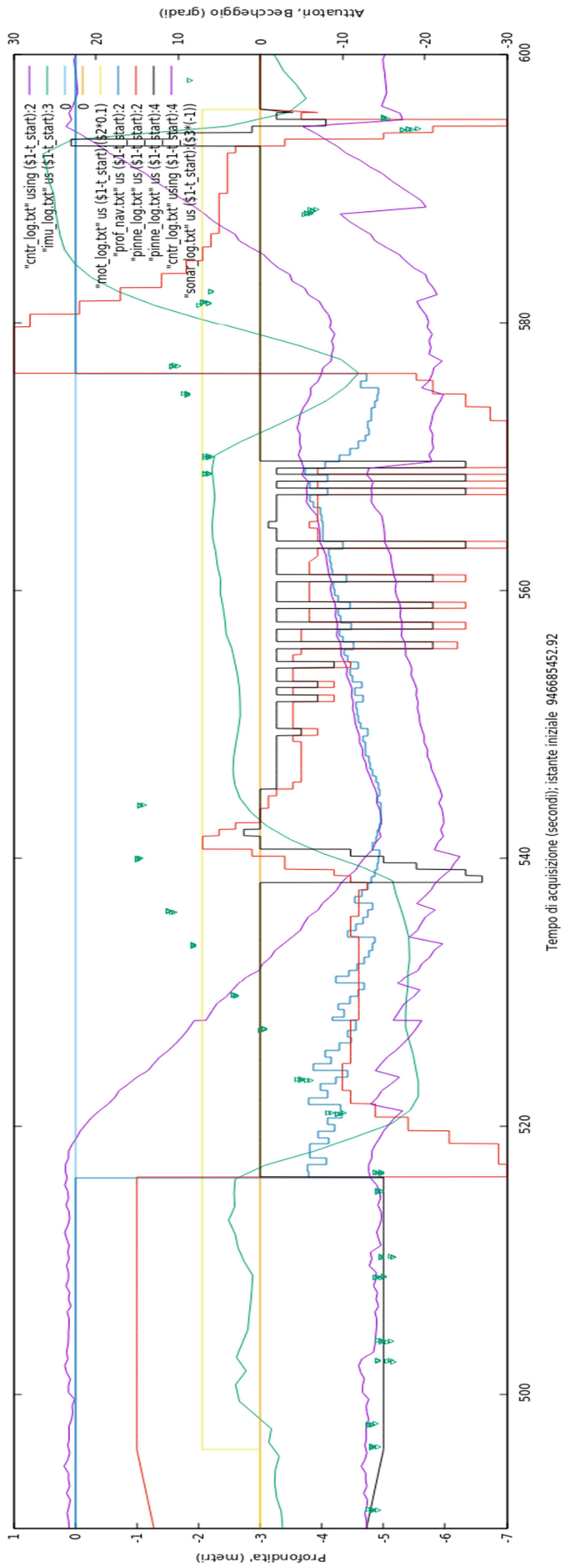


Figura 6.11 – 'Lacuna' sonar  
(23/10/2019)

In **Fig. 6.12** è illustrata una navigazione 'tipo' su fondale (29/10/2019). Il setpoint di profondità è 'copiato' con la consueta precisione, e anche il controllo di assetto svolge in maniera accettabile il suo compito (pitch lievemente negativo, con un setpoint di 0°).

È stata impiegata la nuova modalità di calcolo del fondale (vedi commento alla **Fig. 6.6**), per cui questo appare più regolare, sebbene la discontinuità sia sempre presente.

La **Fig. 6.13** mostra un lancio effettuato nella stessa giornata (29/10/2019). La navigazione si è svolta senza intoppi fino a circa 20 sec prima del termine del tratto in immersione, quando Venus si è incagliato in un ammasso di alghe. Si ricorda in proposito che non è tuttora disponibile un algoritmo di obstacle avoidance che, se presente, avrebbe potuto evitare l'impatto.

L'interesse del tracciato è però nell'andamento del pitch, che oscilla bruscamente intorno al valore di setpoint (che era di 0°) per poi diventare decisamente negativo col Venus incagliato. La differenza rispetto alla precedente missione sta nel valore di potenza impostato per il motore: 80% invece che 70%. A tale differenza si è attribuito il comportamento osservato.

Si ribadisce (§3.3) che l'inserimento del motore nel loop di controllo sarebbe sicuramente utile nella gestione di comportamenti del genere (**Cap. 7**).

La **Fig. 6.14** illustra infine il comportamento del controllo di direzione. Il relativo lancio è stato effettuato nella stessa giornata dei due precedenti (29/10/2019), ed è consistito in un 'quadrato' in superficie, con lati della durata di 60 sec ciascuno.

Nel grafico sono riportati il setpoint di direzione (**buss\_nav:2**), la risposta di Venus (lettura bussola, **buss\_log:2**) e l'azione dei timoni di direzione (**pinne\_log:6**).

Il controllo è soddisfacente, e infatti non ha mai dato problemi nello svolgimento dei test.

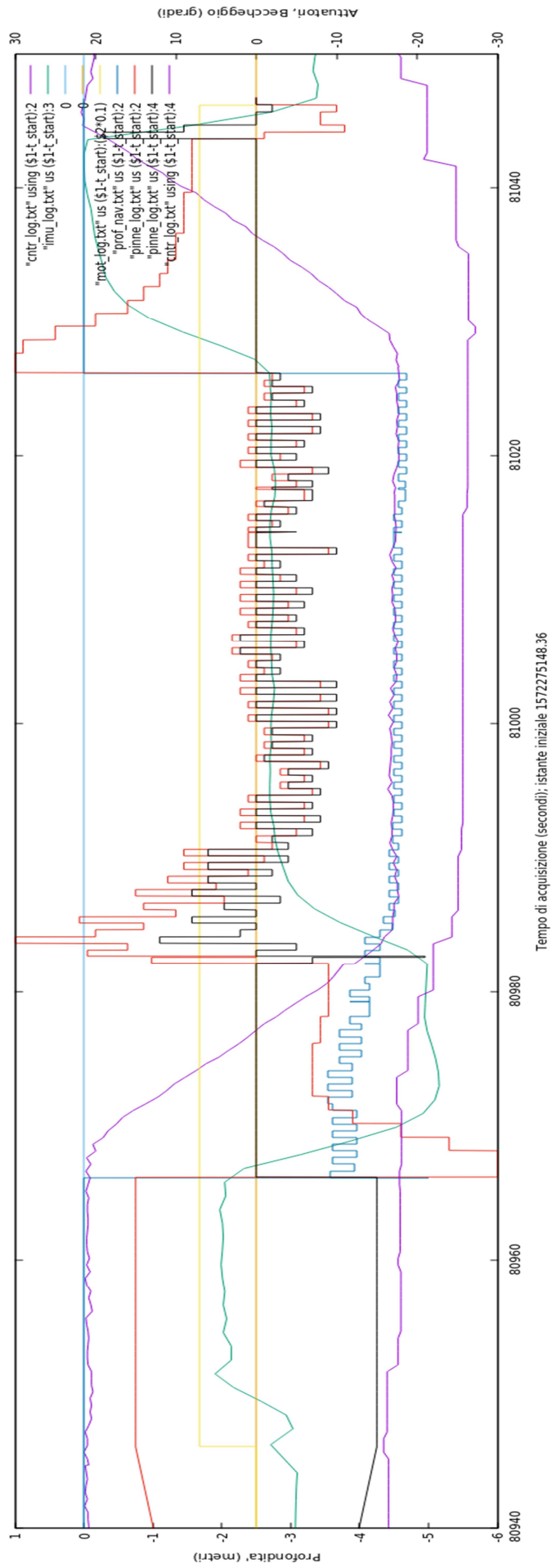


Figura 6.12 – Navigazione 'sul fondo', potenza 70%  
(29/10/2019a)

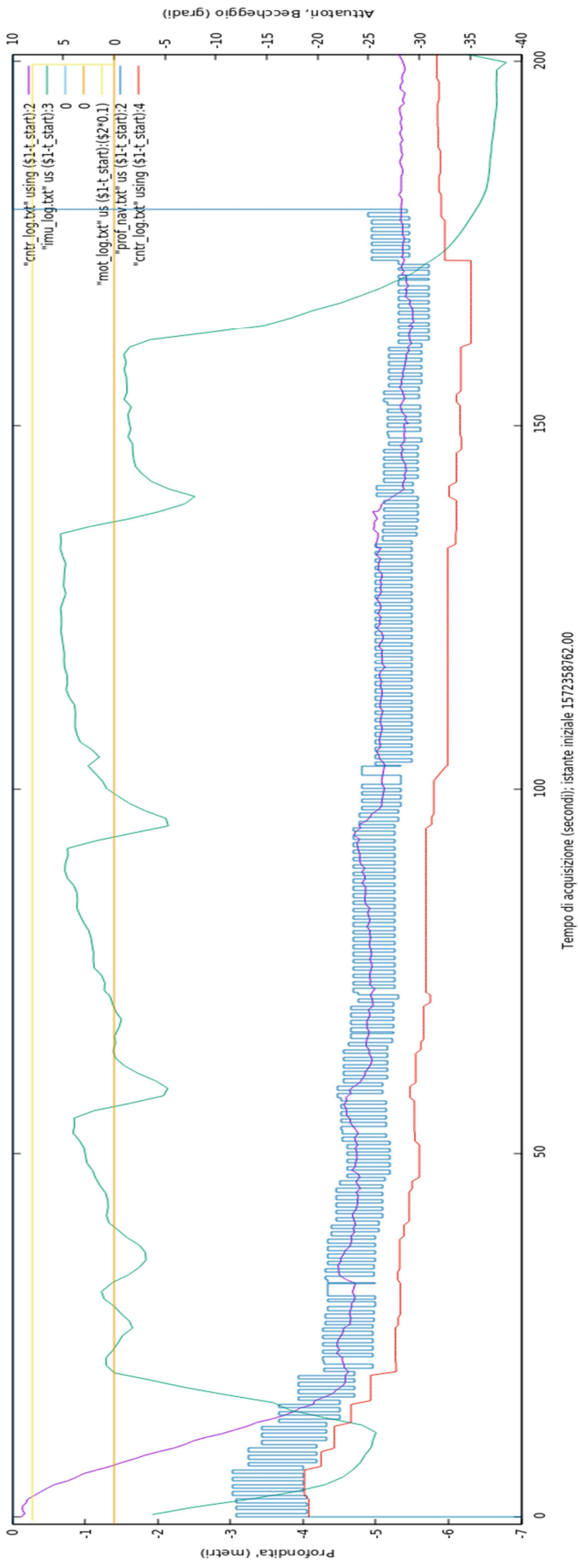
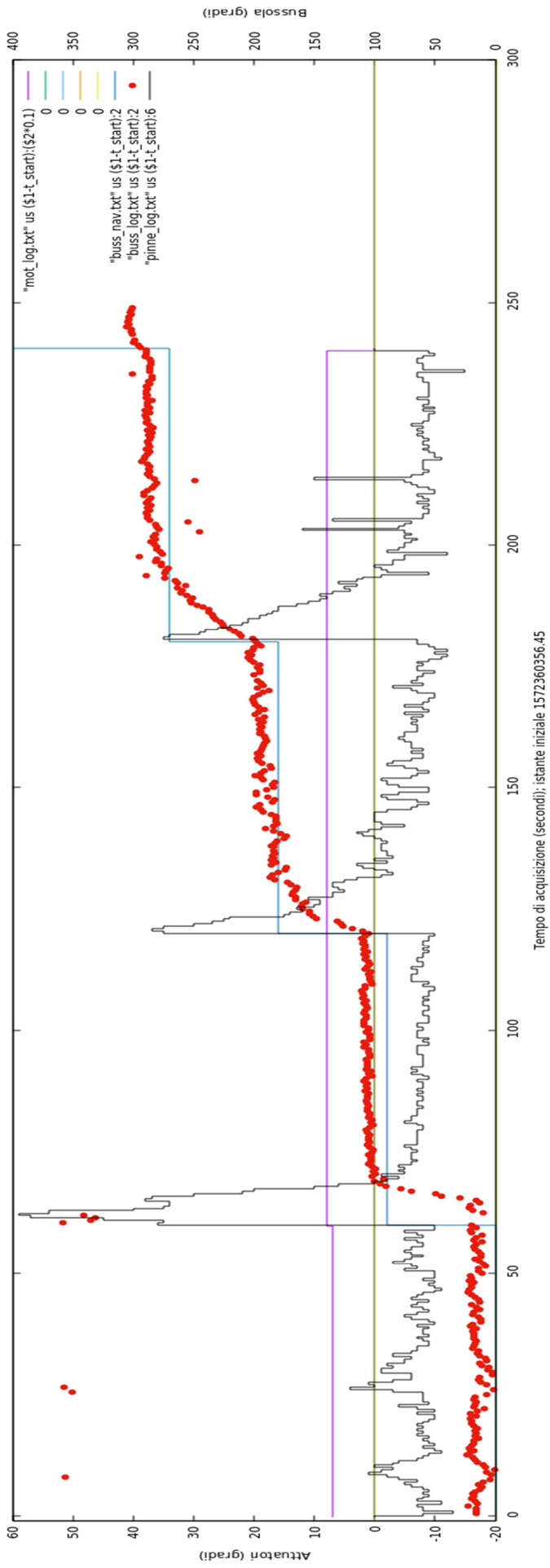


Figura 6.13 – instabilità assetto, potenza 80%  
(29/10/2019b)



**Figura 6.14 – Controllo di direzione  
(29/10/2019c)**

## 7. Criticità e miglioramenti

Nella prima fase delle prove sul campo sono emersi, com'era ragionevole attendersi, alcuni malfunzionamenti nella componentistica e nell'assemblaggio, ormai risolti e peraltro inevitabili data la natura prototipale di Venus.

Attualmente è stato raggiunto un buon livello di affidabilità generale. Come mostrato nel **Cap. 6** Venus è in grado di svolgere semplici missioni programmate, che vanno intese come elementi la cui combinazione consentirà di svolgere i compiti complessi per cui Venus è stato pensato (**Cap. 1**).

Sono tuttavia emerse alcune criticità non trascurabili, che hanno causato un certo rallentamento nell'attività sul campo.

Le principali riguardano l'acquisizione video e il link WiFi fra Venus e la Stazione di Controllo (MMI).

Per quanto riguarda la prima si è constatato che l'effettuazione di una ripresa da parte del nodo Video blocca in pratica l'attività degli altri Nodi operanti sulla Testa e lo scambio dati sulle Topic interessate.

Nel **Cap. 6** (commento alla **Fig. 6.11**) si è discussa la presenza, e le possibili serie conseguenze, di lacune - anche estese - nel tracciato sonar, osservando che si sono sempre verificate in concomitanza a riprese video, e mai in missioni che non prevedessero acquisizioni.

La responsabilità del malfunzionamento viene di conseguenza attribuita al nodo Video; come controprova si è notato che il fenomeno non si verifica nei tratti di navigazione in superficie/GPS, nei quali non vengono di norma effettuate acquisizioni. Oltre al sonar, anche il nodo Listener accusa inconvenienti, vedendo compromessa la capacità di registrare (su standard output) i dati previsti (§4.4).

Si rileva infine una scarsa qualità dei video acquisiti, consistente nella perdita di serie consecutive di frame (video 'a scatti').

Le cause del malfunzionamento sono state individuate nella insufficiente potenza di calcolo dell'HW e, in parte, nel funzionamento non particolarmente efficiente del SW impiegato.

La modifica che dovrebbe comportare la radicale soluzione di queste problematiche è in avanzata fase di realizzazione. Prevede sia l'installazione di una terza CPU (su SBC), caratterizzata da migliori prestazioni, dedicata almeno inizialmente alla sola gestione delle videocamere, che l'impiego di un SW più efficiente (Ffmpeg). L'insieme è stato già testato off-line (G. Cupertino), e si attende l'installazione a bordo di Venus.

Alle problematiche del link WiFi si è fin qui soltanto accennato, in quanto non comportano problemi di sicurezza, almeno nell'esecuzione delle missioni. Si vuole qui ricordare che la missione rappresenta la modalità principale di impiego di Venus, e che dal momento del suo inizio il funzionamento è completamente autonomo e non necessita di link con la Stazione di Controllo.

Rilevanti sono state invece le conseguenze in termini di ritardi nell'esecuzione delle prove, sia per difficoltà nella trasmissione dei comandi, sia per lo scarico a fine missione dei dati registrati, che come discusso sono essenziali per il prosieguo delle prove stesse.

La causa dei malfunzionamenti è stata individuata nell'antenna installata a bordo di Venus. Questa è situata in una struttura cilindrica di plexiglas insieme al sensore GPS e ad un certo numero di LED che fungono da luci di segnalazione.

Si ritiene che sarebbe opportuno separare innanzitutto le due antenne, per evitare possibili interferenze; la modifica ipotizzata prevede la realizzazione di una seconda struttura che ospiti il GPS, e il miglioramento dell'alloggiamento dell'antenna WiFi nel cilindro esistente.

Oltre alle problematiche descritte, che riguardano componenti del sistema già operativi, si ritiene di poter indicare almeno due possibili miglioramenti consistenti nell'aggiunta di nuove funzionalità.

Nonostante il controllo di tipo PID svolga bene il proprio compito, come più volte sottolineato nell'esame dei grafici di **Cap. 6**, in particolari situazioni il controllo dell'assetto non è del tutto soddisfacente (vedi, p. es., la **Fig. 6.13**). Si ritiene che in casi del genere un accurato modello dinamico potrebbe portare, mediante la determinazione in tempo reale della distribuzione di forze sul veicolo, ad una azione complessivamente più efficace, permettendo l'inserimento nel loop di controllo della regolazione del motore (§3.3).

Attualmente Venus è impossibilitato a individuare ostacoli sulla linea di marcia, e si è visto nel **Cap. 6** come ciò abbia comportato (in realtà in più di un'occasione) l'incagliamento in masse di alghe che, in certe stagioni, abbondano sul fondale del lago di Bracciano, alle quote interessate dalle immersioni di prova.

Ciò rende evidente, qualora ce ne fosse bisogno, la necessità dell'implementazione di un algoritmo di obstacle avoidance. Si potrebbe pensare di realizzarlo impiegando il solo sonar (già predisposto (§4.2) ad effettuare rilevazioni in linea di marcia), ma proprio il caso commentato in **Fig. 6.13** mostra che questo non sarebbe sufficiente all'individuazione di certi tipi di ostacoli, come appunto un ammasso di alghe. Infatti nel caso in esame il tracciato sonar nell'intorno di 90° (direzione di marcia – file **ob\_av\_log.txt**) è stato esaminato qualitativamente, senza poter distinguere con chiarezza l'ostacolo presente.

Si ritiene quindi che un algoritmo efficace dovrebbe prevedere l'impiego congiunto di sonar e videocamera.



## 8. Appendici

### 1 - Missioni

```
enum HMI_mss_t { NO_mst, // null mission step
                rec_mst, // missione da file
                nav_mst, // navigazione (buss/prof)
                sup_mst, // navigazione superficie (buss)
                gps_mst, // navigazione gps (superficie)
                stp_mst}; // arresto motore/reset timoni
```

## 2 – Comandi

```
enum HMI_cmd_t { NO_cmd, // stop nodi
ON_cmd, // start nodi
mtB_cmd, // potenza motore
mtS_cmd, // stop motore
mtD_cmd, // disabilitazione motore (folle)
hlB_cmd, // timone direzione - helm
hlS_cmd, // helm -> 0°
dvB_cmd, // timone profondità - dive
dvS_cmd, // dive -> 0°
prg_cmd, // programmazione traiettorie
mss_cmd, // programmazione missione
fll_cmd, // pinne in folle
tst_cmd, // test pinne
bss_cmd, // navigazione con bussola
prf_cmd, // navigazione con profondimetro
gps_cmd, // navigazione con gps
lON_cmd, // log strumenti ON
lOF_cmd, // log strumenti OFF
sdl_cmd, // download log strumenti
srs_cmd, // reset log strumenti
ddl_cmd, // download log generale
lpr_cmd, // parametri log generale
npr_cmd, // parametri nodi
fON_cmd, // fari ON
fOF_cmd, // fari OFF
sON_cmd, // foto singola - fari ON
sOF_cmd, // foto singola - fari OFF
vdL_cmd, // start video Sx
vdR_cmd, // start video Dx
rcL_cmd, // start REC video Sx
rcR_cmd, // start REC video Dx
vpr_cmd, // programmazione video
fpr_cmd, // programmazione foto
vdl_cmd, // download video
odl_cmd, // download video su canale ottico
vrs_cmd}; // reset video
```

## 9. Bibliografia

- [1] F. Cavallini: “Small AUV – VENUS (Veicolo per la Navigazione sUbacquea e la Sorveglianza)” – ENEA, Febbraio 2010
- [2] G. Cupertino, P. A. Fichera: “VENUS – Veicolo per Navigazione Subacquea e Sorveglianza – Descrizione elettro-meccanica – Test Preliminari” – ENEA, 2020
- [3] G. Cupertino: “VENUS – Veicolo per Navigazione Subacquea e Sorveglianza – Programmazione di base - Test Preliminari” – ENEA, 2020
- [4] M. T. Mazzucato: “Coordinate geografiche e gaussiane, un metodo di trasformazione” – Matematicamente.it Magazine, n. 20, Settembre 2013
- [5] M. Martinelli: “Integrazione di sensori sonar per underwater SLAM” – Tesi di Laurea, Università ‘La Sapienza’ - Roma, 2012
- [6] Trittech: Software Notes for controlling and operating RS-232 Sonar Heads – 2003/2010
- [7] Phidget: [https://www.phidgets.com/docs/OS\\_-\\_Linux](https://www.phidgets.com/docs/OS_-_Linux)

ENEA  
Servizio Promozione e Comunicazione  
[www.enea.it](http://www.enea.it)

Stampa: Laboratorio Tecnografico ENEA - C.R. Frascati  
maggio 2020