



UNIVERSITA' DEGLI STUDI DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Centro Interdipartimentale di Studi e Attività Spaziali, "G. Colombo"

SCUOLA DI DOTTORATO DI RICERCA IN: Scienze Tecnologie e Misure Spaziali

INDIRIZZO: Astronautica e Scienze da Satellite

CICLO: XXI

**SIMULTANEOUS LOCALIZATION AND MAPPING
APPLIED TO AN AIRSHIP WITH INERTIAL NAVIGATION SYSTEM
AND CAMERA SENSOR FUSION**

**SIMULTANEOUS LOCALIZATION AND MAPPING
APPLICATO AD UN DIRIGIBILE MEDIANTE SENSOR FUSION
TRA SISTEMA DI VISIONE E PIATTAFORMA INERZIALE**

Direttore della Scuola: Ch.mo Prof. Cesare Barbieri

Supervisore: Ch.mo Prof. Stefano Debei

Co-supervisore: Ch.mo Prof. Roberto Da Forno

Dottorando: Nicola La Gloria

Contents

Abstract	1
Sommario	3
1.1 Introduzione	3
1.2 Modello Dinamico dell’Airship	5
1.3 Controllo dell’Airship	8
1.4 I Sensori per la navigazione	10
1.4.1 La piattaforma inerziale	10
1.4.2 Il sistema di visione	11
1.5 La formulazione del problema dello SLAM e la sua natura probabilistica	14
1.5.1 Requisiti per la soluzione del problema	15
1.5.2 Soluzione basata su EKF	16
1.6 Le sottomappe	17
1.7 Monocamera SLAM	19
1.8 Risultati	19
Introduction	3
1 Aerobots: Concepts and Applications	5
1.1 Introduction	5
1.2 Background and Motivation	5
1.3 Simultaneous Localization and Mapping (SLAM)	6
1.4 Aerobot Achitectures Concepts Overview	7
1.4.1 LTA Vehicles	7
1.4.2 HTA Vehicles	10
1.5 Aerobot Applications	10
1.5.1 Venus Exploration	10
1.5.2 Titan Exploration	11

1.5.3	Mars Exploration	12
1.5.4	Cooperative Autonomous Ground-Air-Space Robots	13
1.5.5	Terrestrial Application	14
2	Vehicle Dynamics Modeling	15
2.1	Coordinate Systems and Trasformations	16
2.1.1	ECEI Reference Frame	16
2.1.2	ECEF Reference Frame	16
2.1.3	Topocentric Reference Frame	17
2.1.4	North East Down Reference Frame	18
2.1.5	Vehicle Reference Frame	18
2.2	Rigid Body Dynamics	18
2.3	Rigid Body Kinematics	20
2.3.1	Attitude parametrizations	21
2.4	General 6 DOF Rigid Body Equations of Motion	23
2.4.1	Kirchoff Equations	24
2.4.2	Coriolis and Centripetal Matrix	25
2.5	General Principles of Hydrodynamics	26
2.6	Added Mass and Inertia	29
2.6.1	Coriolis and Centripetal Matrix	34
2.6.2	Added Mass of the fins	34
2.7	Hydrodynamic Forces acting on an Airship Hull	36
2.8	Viscous Forces	39
2.8.1	Viscous Lateral Force	39
2.8.2	Axial Drag	42
2.9	Force and Moments Due to Control Surface Deflection	43
2.9.1	Restoring Forces and Moments	43
2.10	Airship Equations of Motions	43
2.11	Wind	44
2.12	Control Strategy	44
3	Numerical Dynamics Implementation and Results	47
3.1	MASS: MATLAB Airship Slam Simulator	47
3.2	Earth Airship Setup	48
3.3	Earth Simulations Results	51
3.3.1	Airship in Neutral Flight	51

3.3.2	Airship in Constant Wind	53
3.3.3	Airship in Constant Wind with Stern Engine	55
3.4	Titan Airship Setup	57
3.5	Titan Simulations Results	60
3.5.1	Way-Point Following	61
3.5.2	Leading Way-Point Following	63
4	Navigation Sensors	67
4.1	Inertial Navigation System	67
4.1.1	Accelerometers	68
4.1.2	Gyroscopes	68
4.1.3	Full 6-DOF INS	68
4.1.4	Strap Down Navigation Equations	70
4.1.5	Biases	72
4.1.6	IMU Error Modeling	73
4.2	Vision Navigation System	75
4.2.1	Pinhole Camera Image Model	75
4.2.2	Camera Calibration	77
4.2.3	Stereo Camera Error Modeling	77
4.2.4	Correspondence Problem	79
4.2.5	Basic Matching Strategies	79
4.2.6	SLAM Detection and Matching Hypotesis	82
4.2.7	SLAM VNS Set Up	82
5	The SLAM Problem	87
5.1	Formulation of the SLAM problem	87
5.1.1	Probabilistic Nature of SLAM	88
5.1.2	SLAM Solution Requirements	90
5.2	Non-Linear Systems	92
5.3	EKF Solution	93
5.3.1	System State	94
5.3.2	Vehicle and Landmark Model	94
5.3.3	Sensor model	95
5.3.4	Estimation Process	95
5.4	Managing Features	98
5.4.1	Feature Initialization and State Augmentation	98

5.4.2	Features Extraction	99
5.4.3	Data Association	99
5.5	Computational Complexity	100
5.6	Compressed Extended Kalman Filter (CEKF)	100
5.7	Monocular SLAM	103
6	Airship SLAM Implementation	105
6.1	Basics Remarks	105
6.1.1	1-DOF EKF Slam	105
6.1.2	2-DOF EKF Slam: a rover with encoders	108
6.1.3	Measurement Process	110
6.1.4	Simulation Results	112
6.2	Airship Slam	115
6.2.1	State Estimation Module (SEM)	115
6.2.2	Map Management	116
6.2.3	Computational Cost	117
6.2.4	IMU Measurement	117
6.2.5	EKF Prediction Stage	118
6.2.6	Stereo Camera Measurement	121
6.2.7	Data association and State Augmentation	122
6.2.8	Update Stage	123
6.3	Simulations results	125
6.3.1	Operations Flight	126
6.3.2	Monocamera Slam	136
	Appendix	148
A	Rotations Kinematics	149
A.1	Kinematic form of Rotation Matrix	149
A.2	Time dependence of Rotation Matrix	150
B	MATHCORE Library Reference	151
B.1	Functions	151
B.1.1	Function Documentation	153
	Bibliography	163

Abstract

This thesis handles the problem of Simultaneous Localization and Mapping (SLAM) applied to airships by sensor fusion between Inertial Measurement Unit (IMU) and a Vision system (mono or stereo cameras).

The simultaneous localization and mapping problem asks if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and to incrementally build a consistent map of this environment while simultaneously determining its location within this map. The “solution” of the SLAM problem has been one of the notable successes of the robotics community over the past decade. SLAM has been formulated and solved as a theoretical problem in a number of different forms. SLAM has been implemented in a number of different scenarios from indoor robots to outdoor, but in the domain of flying vehicles, only recent works have demonstrated its feasibility. Despite the progress in the area of autonomous vehicles with on-board control and navigation systems capable of planning and executing trajectories, there is still an untapped potential in the form of LTA vehicles (lighter-than-atmosphere) as unmanned robotic platforms.

Lighter-than-atmosphere (LTA) systems provide significant advantages for planetary exploration due to their potential for extended mission duration, long traverse, and extensive surface coverage capabilities. Robotic airships, in particular, are ideal platforms for airborne planetary exploration. Airships have modest power requirements, and combine the extended airborne capability of balloons with the maneuverability of airplanes or helicopters. Their controllability allows precise flight path execution for surveying purposes, long-range as well as close-up ground observations, station-keeping for long-term monitoring of high-value science sites, transportation and deployment of scientific instruments and in-situ laboratory facilities across vast distances to key science sites, and opportunistic flight path replanning in response to the detection of relevant science sensor signatures. Furthermore, robotic airships provide the ability to conduct extensive surveys over both solid terrain and liquidcovered areas, and to reconnoiter sites that are inaccessible to ground vehicles. Implementation of these capabilities requires achieving a high degree of vehicle autonomy across a broad spectrum of operational scenarios.

The main difficulty to build a consistent map for an autonomous vehicle is to precisely determine the sensor position and orientation as it moves. Dead reckoning techniques, that integrate over time the data provided by motion estimation sensors, such as wheel encoders for rovers or inertial sensors, are not sufficient for that purpose because they’re intrinsically prone to generate position estimates with unbounded error growth. On the other hand relative position sensors as cameras (VNS) have an operating frequency too low to provide a position estimation with a certain degree of reliability and bounded uncertainty. Here a fusion of the two system is proposed to build a solution of the localization and mapping problem using the estimation theory.

Airship dynamics and control is a crucial point to model a realistic navigation system which takes into account the characteristics of the motions of this kind of vehicle in a selected environment. A consistent dynamical model has been developed which include aerodynamic, actuators and environmental modelling. The aerodynamic model proposed uses the integration of the added

mass contribution to estimate pressure forces and moments due to potential flow, while explicit form for aerodynamic forces has been derived to compute viscous effects on the hull. These effects has been estimated by determining a station point from which the flow ceases to be potential.

An airship simulator has been developed and two different kind of veichles have been modelled, for terrestrial and planetary applications. Also a control strategy, for path planning, has been design with a particular attention to its robustness in case of wind disturbances. Terrestrial and Titan atmosphere, gravity and dynamical viscosity were consirered.

The question of the autonomous navigation of the airship in an unknown enviroment has been solved with SLAM technique and the solution proposed is based on the Extended Kalman filter where the system state merge vehicle kinematic paramters and feature location.

The state-based formulation of the SLAM involves the estimation of a joint state and its computational complexity scales quadratically with the number of landmarks.

First implementations adopted the classic EKF to build a consistent map of a unknown landscape by IMU measurement as inputs for the system and stereo camera to provide bearing and range measurements. As the number of landmark increased less performant become the estimation process, so in order to manage large maps a variation of the EKF has been implemented: the Compressed Extented Kalman Filter.

Both EKF and CEKF solutions has been tested vitually in a Titan simulated enviroment, using the dynamical data of the Titan Explorer airship produced by the dynamics and control simulator.

In order to go beyond the limits of a stereo system given by increasing errors at higher altitudes, a SLAM techique based on single camera measurement has been developed. Here for every new landmark measured, a subfiltering process is invoked to estimate the depth of the landmark.

Sommario Esteso

1.1 Introduzione

Questo lavoro di tesi affronta il problema dello SLAM (Simultaneous Localization and Mapping) come strumento di navigazione autonoma per dirigibili (detti in generale veicoli LTA, *lighter-than-atmosphere*). La problematica dello SLAM è verificare se è possibile per un robot essere collocato all'interno di un luogo sconosciuto, in una posizione sconosciuta, e di costruirne incrementalmente, muovendosi all'interno di esso, una mappatura consistente determinando la sua posizione all'interno di essa. Lo SLAM è stato formulato e risolto come problema teorico in modi differenti; la sua implementazione è orientata prevalentemente all'interno della robotica 2D sia indoor che outdoor.

Per quanto riguarda la sua applicazione per la navigazione di veivoli, che si muovono quindi nello spazio tridimensionale, recenti studi ne hanno dimostrato l'effettiva fattibilità; le principali piattaforme di test sono state piccoli veicoli autonomi come aerei ed elidroni. Un certo interesse ha suscitato l'implementazione dello SLAM sui dirigibili dato il loro vasto impiego sia in operazioni militari e civili. I recenti programmi NASA ed ESA inoltre hanno proprio questi veicoli come attori principali per le future missioni in situ, missioni che possono costituire una nuova frontiera per l'esplorazione planetaria dopo l'impiego di lander e rover.

I dirigibili sono delle piattaforme strategiche in quanto hanno bassi requisiti energetici, permettono missioni più longeve e possono attraversare vasti territori inaccessibili ai rover. La loro controllabilità permette la pianificazione di traiettorie ad-hoc per lo studio delle superfici planetarie per lunghe distanze potendo variare la propria altezza operativa a seconda delle necessità.

L'applicazione di tecniche SLAM permetterebbe una localizzazione del dirigibile in un qualche sistema di riferimento quando misure assolute di posizione (ad esempio GPS) non siano disponibili (come nel caso dei pianeti) e quindi la possibilità di poter utilizzare queste informazioni non solo per la propria navigazione ma eventualmente anche di altri veicoli cooperanti.

Pensiamo infatti ad un veicolo che si trovi in un ambiente sconosciuto senza alcun supporto GPS, prima che esso possa prendere alcuna decisione all'interno di esso ne deve conoscere la topologia; per spostarsi da un punto ad un altro deve sapere la sua posizione all'interno di una mappa dove i target possono essere localizzati con un sistema di coordinate.

Muovendosi all'interno di un landscape e acquisendo informazioni su di esso grazie ad un qualche sensore di bordo, come ad esempio un sistema di visione basato su telecamera oppure lidar, è possibile per il veicolo localizzarsi e localizzare le informazioni sull'ambiente circostante.

Ciò che lo strumento di visione misura è strettamente legato con lo stato cinematico del veicolo la cui misurazione diventa di grande importanza: per misurare il proprio stato dinamico e cinematico con una certa frequenza è necessaria una piattaforma inerziale (IMU, *Inertial Measurement System*).

Il problema della localizzazione sembra quindi banale in quanto se il sistema di visione, ad esempio un sistema stereo, è in grado di fornire una misura tridimensionale di una feature del

terreno e la piattaforma inerziale la posizione del veicolo da un certo istante iniziale, abbiamo tutte le informazioni per localizzare sia il veicolo che la feature.

La questione cruciale è che sia la piattaforma inerziale e sia il sistema di visione sono soggetti ad incertezza nel processo di misura! L'incertezza della IMU nel misurare le forze specifiche e le velocità angolari porta nel tempo ad effetti catastrofici nella determinazione di velocità e posizione a causa della propagazione dell'incertezza nel processo di integrazione; lo stesso discorso vale per l'assetto. Analogamente, un sistema stereo ha un'incertezza variabile dal tipo di risoluzione e ottiche impiegate che può portare a stimare una misura di range con un errore di 5m da un'altezza di 100m !

Lo SLAM è una tecnica basata su un processo di stima e filtraggio statistico, mediante il quale le variabili di stato del sistema come assetto, velocità, posizione del veicolo e le feature che costituiscono la mappa, vengono stimate in un processo iterativo in tempo reale da un computer di bordo mediante le informazioni provenienti dalle misure affette da incertezza; tale incertezza è modellata come un rumore bianco gaussiano.

I particolare noi affronteremo in questo lavoro una configurazione in cui le misure del sistema di visione costituiscono gli output del nostro processo mentre quelle della piattaforma inerziale ne costituiscono gli input; ecco perchè parleremo di sensor fusion tra IMU e sistema di visione.

I dirigibili sono delle ottime piattaforme sperimentali per testare la fusione tra un sistema di navigazione inerziale e uno basato sulla visione. Uno dei fattori determinanti è la loro bassa velocità in quanto è possibile la determinazione dello stato dinamico del veicolo in tempo reale mediante il processo di stima, sia per la compatibilità con le frequenze di campionamento tra sensori di visione e il tempo computazionale richiesto per il processamento dei dati, sia per la migliore identificazione e tracking delle features (dette anche *landmark*); di contro i sensori inerziali hanno una grande incertezza in caso di dinamica lenta.

Per progettare e implementare la navigazione di un dirigibile mediante SLAM è necessario un ambiente di simulazione. Questo significa che va studiata la dinamica del veicolo per poter simulare l'acquisizione dalla IMU da un modello più realistico possibile.

Proprio in questa direzione si svolge la prima parte di questo lavoro, dove è stata analizzata la dinamica del dirigibile mediante la scrittura e la risoluzione numerica delle equazioni del moto. La letteratura in merito si è rivelata alquanto ostica in quanto molto materiale risale alle prime applicazioni dell'idrodinamica per lo studio di questi veicoli da parte di Max Munk negli anni 20, in cui alcuni aspetti formali sono a volte poco chiari e non sufficientemente motivati. Successivamente fino alla metà degli anni 50 vi sono una serie di lavori in cui si cerca di raffinare gli aspetti teorici integrandoli con i dati sperimentali mediante relazioni semi-empiriche.

In questo lavoro sono stati validati alcuni risultati salienti implementando un modello dinamico proveniente da recenti pubblicazioni sullo studio dei veicoli marini; le forze aerodinamiche "potenziali" dovute alla pressione del fluido sono modellate mediante l'integrazione della dinamica della massa virtuale, mentre i contributi viscosi sono stati ottenuti mediante relazioni semi-empiriche note dalla letteratura. Si è inoltre considerata all'interno del modello l'azione del vento.

Accanto alla dinamica-cinematica del dirigibile, l'implementazione di una tecnica di controllo è un requisito funzionale importante per poter realizzare la *loop closure*; il fatto di poter osservare nuovamente una stessa feature migliora la stima dell'intera mappa. Si è cercato quindi di poter realizzare traiettorie intelligenti mediante un algoritmo di controllo sufficientemente robusto anche in caso di vento laterale costante, pari ad un mezzo della velocità del veicolo.

L'artefatto prodotto da questa prima fase dello studio è un modulo software per la simulazione della dinamica e controllo del dirigibile. Sono stati modellati due tipi di veicolo, uno per usi terrestri (di 9 m di lunghezza) e uno per l'esplorazione planetaria (di 17 m di lunghezza). Quest'ultimo è stato implementato secondo le specifiche del "Titan Explorer", architettura pro-

posta per l’esplorazione in situ di Titano dalla NASA.

Sono state fatte quindi diverse simulazioni sia in environment terrestre sia in atmosfera di Titano per studiare la dinamica del veicolo evidenziando peculiarità dovute a diversa gravità, pressione atmosferica e viscosità dinamica.

Come già accennato, la soluzione del problema dello SLAM è di natura probabilistica dove gioca un ruolo fondamentale la correlazione tra la posizione dei landmark e la cinematica del veicolo; questa correlazione cresce monotonicamente con le osservazioni.

Lo *stato* del sistema in esame include sia le informazioni relative alla cinematica del veicolo sia la posizione di ogni singolo landmark; il peso computazionale di tale configurazione all’interno di un processo di stima è sicuramente elevato.

E’ stato affrontato lo studio e l’implementazione della soluzione al problema dello SLAM sia mediante l’utilizzo del filtro di Kalman esteso classico, (*Extended Kalman Filter*), che mediante la sua variante compressa CEKF (*Compressed Extended Kalman Filter*). Quest’ultima è di grande applicabilità nel caso la mappa contenga un numero di landmark tale da rendere troppo elevato il costo computazionale e quindi si renda necessario la suddivisione in sottomappe del landscape in esame.

Il sistema di visione è stato studiato in due diverse configurazioni: stereo e mono camera. La prima fornisce misure di bearing e range e quindi un’informazione tridimensionale della feature, mentre la seconda solamente la misura di bearing. In tal caso è necessaria l’implementazione di una tecnica diversa basata sull’utilizzo di un sottofiltro per ogni nuova feature osservata per la stima della profondità.

L’artefatto prodotto dallo studio delle diverse problematiche dello SLAM sono stati dei moduli software per la simulazione della navigazione in che in cooperazione con quelli relativi alla dinamica costituiscono quello che è stato chiamato MASS: Matlab Airship Slam Simulator.

Mediante questo strumento sono state effettuate simulazioni di navigazione autonoma in ambiente terrestre e planetario (Titano) mediante misure provenienti da stereo e mono camera con risultati incoraggianti per l’applicazione di quanto prodotto su un veicolo reale.

1.2 Modello Dinamico dell’Airship

Nella navigazione in genere uno degli aspetti funzionali essenziali è il poter descrivere la posizione di un veicolo rispetto all’ambiente circostante. Per i nostri scopi è più pratico riferirci ad un veicolo che naviga all’interno del campo gravitazionale generato da una sfera perfetta.

Come è noto, i pianeti non sono sfere perfette ma degli sferoidi irregolari; sicuramente questa è la descrizione più accurata. Tuttavia per la navigazione è più utile una descrizione più semplice anche soprattutto a livello. A fine di costruire una mappa che descriva l’ambiente dove un veicolo sta navigando ci servono principalmente due sistemi di riferimento; uno inerziale o pseudo-inerziale e un sistema solidale al veicolo; avremo inoltre il sistema di riferimento solidale al sensore, ma si tratta semplicemente di una rototraslazione costante nel tempo rispetto al sistema di riferimento del veicolo.

La scelta del sistema di riferimento dove referenziare i punti di una mappa è di tipo geodetico, ovvero che consideri accuratamente la forma non sferica del geoide. Ciò nonostante questo procedimento può essere considerato successivo al processo di ricostruzione di una mappa locale mediante tecnica SLAM. Il veicolo in esame, di tipo LTA, può essere considerato solidale all’atmosfera (ha una velocità relativa rispetto al terreno molto bassa) per cui eventuali effetti inerziali dovuti al moto della terra possono essere trascurati; inoltre essendo la zona locale mappata piccola rispetto alle dimensioni del pianeta, possiamo trascurare le variazioni di forma del geoide. Si provvederà successivamente a referenziare la mappa locale rispetto ad un punto di essa di cui si conoscono anche le coordinate geodetiche, mediante una misura di tipo assoluto

(GPS o un Orbiter in caso di missione planetaria).

In definitiva per i nostri scopi necessitiamo di tre soli sistemi di riferimento; uno che chiameremo impropriamente "inerziale" solidale ad un punto arbitrario del landscape su cui stiamo volando, uno solidale al veicolo, e uno solidale allo strumento di visione¹. In generale il sistema solidale al veicolo è legato a quello inerziale dalla trasformazione omogenea

$${}^i\mathbf{H}_b = \begin{bmatrix} {}^i\mathbf{R}_b & {}^iT_b \\ \mathbf{0} & 1 \end{bmatrix} \quad (1.1)$$

dove ${}^i\mathbf{R}_b$ è descritta dai parametri di Eulero

$$\begin{aligned} {}^i\mathbf{R}_b &= {}^b\mathbf{R}_i^T(\phi, \theta, \psi) \\ &= \begin{bmatrix} \cos \phi \cos \theta & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \sin \phi \cos \theta & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi \\ -\sin \theta & \cos \theta \sin \psi & \cos \theta \cos \psi \end{bmatrix} \end{aligned}$$

dove viene considerata la sequenza 3-2-1 (yaw ϕ , pitch θ , roll ψ).

Molto utile per il calcolo cinematico è la relazione tra velocità angolare e variazione temporale degli angoli di Eulero

$${}^b\boldsymbol{\omega} = \begin{bmatrix} -\sin \theta & 0 & 1 \\ \sin \psi \cos \theta & \cos \psi & 0 \\ \cos \psi \cos \theta & -\sin \psi & 0 \end{bmatrix} \begin{bmatrix} \omega_\phi \\ \omega_\theta \\ \omega_\psi \end{bmatrix} = \mathbf{E}(\phi, \theta, \psi) \begin{bmatrix} \omega_\phi \\ \omega_\theta \\ \omega_\psi \end{bmatrix} \quad (1.2)$$

e anche la sua inversa per calcolare la derivata temporale degli angoli dalla velocità angolare

$$\mathbf{H} = \mathbf{E}^{-1} = \begin{bmatrix} 0 & \frac{\sin \psi}{\cos \theta} & \frac{\cos \psi}{\cos \theta} \\ 0 & \cos \psi & -\sin \psi \\ 1 & \sin \psi \tan \theta & \cos \psi \tan \theta \end{bmatrix} \quad (1.3)$$

Le equazioni che descrivono la dinamica di un corpo rigido possono essere scritte nella forma

$$\begin{bmatrix} m\mathbb{I}_{3 \times 3} & -m\tilde{\mathbf{r}}_G \\ m\tilde{\mathbf{r}}_G & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 - m[\boldsymbol{\omega} \times \mathbf{v} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_G)] \\ \mathbf{m}_0 - [\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + m\mathbf{r}_G \times (\boldsymbol{\omega} \times \mathbf{v})] \end{bmatrix} \quad (1.4)$$

o nella forma compatta usata per gli airships

$$\mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_{RB} \quad (1.5)$$

dove

$$\mathbf{M}_{RB} = \begin{bmatrix} m\mathbb{I}_{3 \times 3} & -m\tilde{\mathbf{r}}_G \\ m\tilde{\mathbf{r}}_G & \mathbf{I} \end{bmatrix} \quad (1.6)$$

Qui $\mathbb{I}_{3 \times 3}$ è la matrice identità $\mathbf{I}$ è il tensore di inerzia rispetto ad O (localizzato nel centro di volume del hull) e $\tilde{\mathbf{r}}_G \in SO(3)$; la matrice di massa generalizzata ha le proprietà

$$\mathbf{M}_{RB} = \mathbf{M}_{RB}^T > 0; \quad \dot{\mathbf{M}}_{RB} = 0$$

Notiamo che nello studio delle equazioni del moto del veicolo si è usata la notazione SNAME dove

$$\mathbf{f}_0 = \boldsymbol{\tau}_1 = [X, Y, Z]^T$$

¹In realtà si dovrebbe considerare anche il sistema di riferimento della IMU che però consideriamo solidale al sistema di riferimento del veicolo

$$\mathbf{m}_0 = \boldsymbol{\tau}_2 = [K, M, N]^T$$

$$\mathbf{v} = \boldsymbol{\nu}_1 = [u, v, w]^T$$

$$\boldsymbol{\omega} = \boldsymbol{\nu}_2 = [p, q, r]^T$$

$$\mathbf{r}_G = [x_G, y_G, z_G]^T$$

La forma della matrice di Coriolis, $\mathbf{C}_{RB}(\boldsymbol{\nu})$, si ottiene mediante le equazioni di Kirchhoff usate nello studio della dinamica dei corpi rigidi immersi in un fluido

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T}{\partial \boldsymbol{\nu}_1} \right) + \boldsymbol{\nu}_2 \times \frac{\partial T}{\partial \boldsymbol{\nu}_1} &= \boldsymbol{\tau}_1 \\ \frac{d}{dt} \left(\frac{\partial T}{\partial \boldsymbol{\nu}_2} \right) + \boldsymbol{\nu}_2 \times \frac{\partial T}{\partial \boldsymbol{\nu}_2} + \boldsymbol{\nu}_1 \times \frac{\partial T}{\partial \boldsymbol{\nu}_1} &= \boldsymbol{\tau}_1 \end{aligned} \quad (1.7)$$

che rappresentano l'analogo delle equazioni della dinamica (1.4) scritte in forma Lagrangiana dove l'energia cinetica è data da

$$T = \frac{1}{2} \boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\nu} \quad (1.8)$$

Si ottiene l'espressione

$$\mathbf{C}(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathbf{S}(\mathbf{M}_{11}\boldsymbol{\nu}_1 + \mathbf{M}_{12}\boldsymbol{\nu}_2) \\ -\mathbf{S}(\mathbf{M}_{11}\boldsymbol{\nu}_1 + \mathbf{M}_{12}\boldsymbol{\nu}_2) & -\mathbf{S}(\mathbf{M}_{21}\boldsymbol{\nu}_1 + \mathbf{M}_{22}\boldsymbol{\nu}_2) \end{bmatrix} \quad (1.9)$$

dove $\mathbf{S}$ è l'operatore *skew*. Il moto di un corpo rigido in un fluido (o aria) induce un moto nel fluido stesso. Di conseguenza anche quest'ultimo possiede una energia cinetica altrimenti nulla se il corpo fosse in quiete. E' quindi ovvio che le equazioni del moto debbano considerare l'energia cinetica che il moto del corpo trasferisce al fluido; nel caso il moto sia accelerato il lavoro compiuto sul fluido è diverso da zero. In questo scenario viene introdotto il concetto di *massa virtuale* o *massa aggiunta*.

La massa virtuale nelle equazioni del moto gioca il ruolo delle *forze di pressione* normali indotte dal fluido sul corpo in moto che dipendono dall'accelerazione di quest'ultimo per il quale impropriamente si devono considerare della variazioni nella massa e nell'inerzia quando in moto in un fluido.

E' importante notare che integrando i termini che includono la massa aggiunta otteniamo i contributi aerodinamici dovuti alla pressione del fluido, quelli cioè che possiamo associare al *flusso potenziale*². E' chiaro che andranno quindi aggiunti nelle equazioni del moto i contributi viscosi.

Le equazioni del moto dell'airship possono scriversi nella forma

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu}) = \boldsymbol{\tau} \quad (1.10)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}\boldsymbol{\nu} \quad (1.11)$$

dove $\boldsymbol{\nu} = [u, v, w, p, q, r]^T$ $\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^T$ e

$$\mathbf{J} = \begin{bmatrix} {}^b\mathbf{R}_i & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{E}(\phi, \theta, \psi) \end{bmatrix} \quad (1.12)$$

$$\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A \quad \mathbf{C}(\boldsymbol{\nu}) = \mathbf{C}(\boldsymbol{\nu})_{RB} + \mathbf{C}(\boldsymbol{\nu})_A \quad (1.13)$$

$$\boldsymbol{\tau} = \boldsymbol{\tau}_B + \boldsymbol{\tau}_P + \boldsymbol{\tau}_A + \boldsymbol{\tau}_G$$

In particolare abbiamo:

²Il flusso è prodotto unicamente dal moto del corpo in un fluido non viscoso e l'unica forza agente sulle particelle del fluido è il gradiente della pressione

τ_G = vettore della forza generalizzata di gravità;

τ_A = vettore delle forze generalizzate aerodinamiche dovute ad effetti viscosi

τ_P = vettore delle forze generalizzate dovute agli attuatori;

τ_B = vettore delle forze generalizzate dovute al galleggiamento.

E' importante sottolineare come gli effetti viscosi siano stati implementati considerando relazioni semi-empiriche note dalla letteratura.

La massa virtuale generalizzata M_A include il contributo dell'hull, calcolato analiticamente dalle sue proprietà geometriche, e delle code posteriori, calcolto dalla loro geometria mediante la tecnica di integrazione per *striscie*.

L'azione del vento viene modellata considerando nelle equazioni del moto la velocità relativa del veicolo rispetto a quella del vento

$$\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_w \quad (1.14)$$

dove $\boldsymbol{\nu}_w = [u_w, v_w, w_w, 0, 0, 0]$ e quindi assunto costante la velocità del vento (o lentamente variabile) in modo che

$$\dot{\boldsymbol{\nu}}_w = 0 \Rightarrow \dot{\boldsymbol{\nu}}_r = \dot{\boldsymbol{\nu}} \quad (1.15)$$

otteniamo le equazioni del moto *relative*

$$M\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \tau_B + \tau_P + \tau_A(\boldsymbol{\nu}_r) + \tau_G \quad (1.16)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}\boldsymbol{\nu} \quad (1.17)$$

1.3 Controllo dell'Airship

Il controllo del veicolo è una condizione necessaria se si vuole operare la *loop closure* nel processo di SLAM. Infatti il poter riosservare una determinata feature apporta dei considerevoli benefici nella sua stima in quanto correlata al resto della mappa e alla cinematica del veicolo.

Anche se il sistema dinamico è fortemente non lineare è stato implementato un controllo di tipo PD in due differenti varianti:

- *leading way point*: questa strategia si basa sulla pianificazione di una determinata traiettoria. Mediante un tool esterno appositamente sviluppato viene generata una poligonale a tratti rettilinei raccordati da spirali cubiche nello spazio delle fasi; il braccio delle spirali è ovviamente una funzione del raggio di curvatura che si vuole considerare, che dipende dalle dimensioni e velocità del veicolo. Tale traiettoria piana (supponiamo che il veicolo abbia effettuato il take-off) è costituita in realtà da una serie di punti (way points) a distanza ravvicinata che il veicolo deve seguire³. Questo approccio si è rivelato particolarmente efficace quando la velocità del vento è elevata (circa metà della velocità del veicolo); infatti la derivata della curva tra due punti successivi varia con lentezza (o per nulla nei tratti rettilinei) per cui il veicolo può variare il suo assetto mediante gli attuatori in modo graduale. Inoltre data la vicinanza tra way point successivi la spinta fornita dagli attuatori dal parametro proporzionale del controllo PD risulta minore dando maggiore stabilità al veicolo in fase di curvatura.
- *orienting way point*: in questo caso i way point vengono posizionati sequenzialmente ad una certa distanza (anche grande) e uniti mediante segmenti; per ognuno di essi viene specificata una velocità target.

³Questi punti rappresentano il campionamento ad una certa frequenza della traiettoria continua

In entrambi i modi è possibile disegnare traiettorie chiuse.

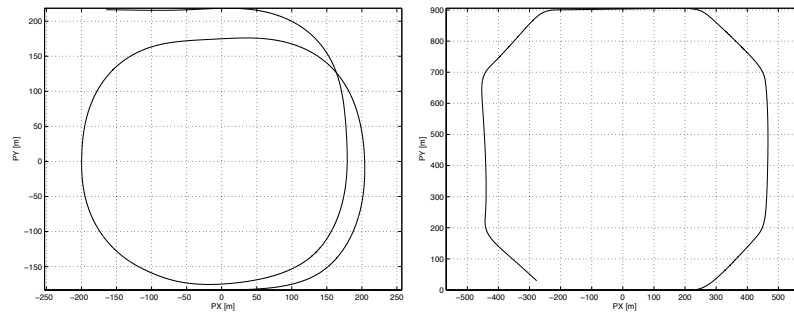


Figure 1.1: Traiettorie di tipo leadind way point e orientiring way point

Gli attuatori scelti per il controllo del veicolo sono:

- motore longitudinale;
- angolo di tilt del motore longitudinale;
- motore di stern posto in coda in direzione ortogonale all'asse longitudinale del veicolo;
- angolo del deflettore di coda per il controllo del pitch;
- angolo del deflettore di coda per il controllo dello yaw;

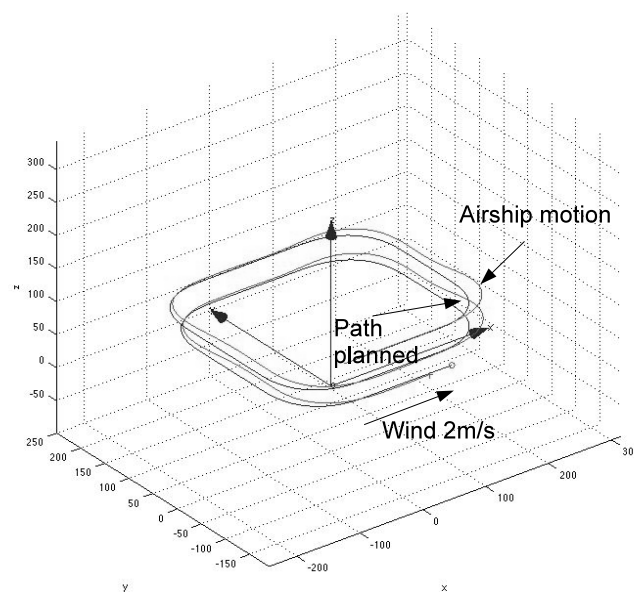


Figure 1.2: MASS software motion visualization

L'artefatto prodotto dallo studio della dinamica e del controllo del dirigibile è un modulo software (DCM, *Dynamic and Control Module*) che permette di simulare il comportamento dinamico del veicolo su traiettorie pianificate. Sono stato modellati due diversi veicoli, uno per uso terrestre (un classico mini-zeppelin) e uno per uso planetario basato su uno studio di missione effettuato dalla NASA per l'esplorazione di Titano.

Il software parametrizza le caratteristiche ambientali che condizionano il moto del dirigibile ad una arbitraria altezza, come la gravità, vento, densità e viscosità dinamica dell'atmosfera. Sono state effettuate diverse simulazioni nei differenti scenari in differenti situazioni di vento:

- simulazione in ambiente terrestre: la velocità di crociera del dirigibile è stata scelta pari a 5 m/s con vento unidirezionale fino a 2 m/s, e altezza di volo pari a 100 m. Il controllo usato è di tipo *leading way point*. Sono stati riportati i profili delle variabili dinamiche e cinematiche.
- simulazione in ambiente planetario (Titano): la velocità di crociera del dirigibile è di 5 m/s con vento unidirezionale di 0.5 m/s (venti longitudinali) e altezza 100 m. Sono stati studiati i profili delle variabili cinematiche e dinamiche in caso di controllo *leading way point* e *orienting way point*

1.4 I Sensori per la navigazione

1.4.1 La piattaforma inerziale

La navigazione e la guida sono definite come il processo di determinare e controllare la posizione di un veicolo.

L'integrazione di differenti tipi di sistemi di navigazione è molto frequente nei sistemi autonomi in quanto permette di minimizzare gli errori di misura, di differente natura, presenti in ogni singolo strumento.

E' molto diffusa l'integrazione tra sensori per la navigazione inerziale e altri sistemi come il Global Positioning System (GPS).

I sistemi di navigazione basati su strumenti di visione (Visual Navigation System) sono anch'essi molto popolari nell'integrazione con le piattaforme inerziali; essi forniscono misure di *bearing* (latitudine e longitudine in un sistema polare centrato nel sensore) e di *range* nel caso il sistema sia stereo.

Un sistema di navigazione inerziale completo consiste in almeno tre accelerometri e tre giroscopi tra loro ortogonali che forniscono misure di accelerazione e velocità angolare rispetto ai tre assi principali.

Una particolare configurazione utilizzata è la "strap-down", dove i sensori sono solidali allo strumento (e al veicolo a meno di una rototraslazione arbitraria), il sistema deve computare il cambio di riferimento dal sistema "locale" a quello inerziale o viceversa a seconda del tipo di meccanizzazione usata.

Sia la velocità (ed eventualmente la posizione) che l'orientazione del veicolo sono determinati mediante un processo di integrazione a partire dalla misura delle accelerazioni specifiche (a meno della gravità) e dalle velocità angolari. Nella pratica questa integrazione propaga nel tempo gli errori dovuti al rumore di misura e alla non linearità dei sensori; maggiore è il tempo di integrazione maggiori saranno questi errori.

Quindi le informazioni provenienti dalla IMU sono utili per fornire posizione e orientazione solo per corti periodi di tempo. Quanto a lungo possiamo utilizzare tali informazioni di posizione e

assetto dipendono unicamente dalla magnitudine di tali errori (*drift rate*).

La traiettoria predetta è naturalmente funzione della calibrazione iniziale e allineamento della piattaforma. Con calibrazione si intende la determinazione dei bias degli accelerometri e dei giroscopi. L'allineamento consiste nel determinare l'orientazione iniziale del sistema solidale alla piattaforma rispetto ad un opportuno sistema di riferimento.

L'allineamento è molto importate in quanto gli algoritmi interni al sistema di navigazione inerziale useranno tale orientazione iniziale per computare le grandezze di interesse.

Nel contesto applicativo in esame la piattaforma inerziale, di tipo strap-down, fornisce le entrate (input) nel sistema dinamico utilizzato nel processo di stima dello stato. Si usano le misure di velocità angolare e accelerazione per computare le predizioni di velocità, posizione e assetto; l'incertezza nella misurazione viene modellata come rumore bianco gaussiano (scorrelato).

Le forza specifica misurata dalla piattaforma di tipo strap-down è rappresentata nel sistema solidale da

$${}^b\bar{\mathbf{S}} = {}^b\dot{\mathbf{v}} + {}^b\dot{\boldsymbol{\omega}} \times {}^b\mathbf{r}_S + {}^b\boldsymbol{\omega} \times ({}^b\boldsymbol{\omega} \times {}^b\mathbf{r}_S) - {}^b\mathbf{g} \quad (1.18)$$

dove ${}^b\boldsymbol{\omega}$ è la velocità angolare del veicolo, ${}^b\mathbf{r}_S$ è la distanza del sensore dall'origine del body frame e ${}^b\dot{\mathbf{v}}$ rappresenta la derivata totale (rispetto agli assi fissi) dell velocità del veicolo. Quindi l'equazione per il modello della piattaforma inerziale sarà

$$\partial {}^b\mathbf{v} = {}^b\bar{\mathbf{s}} + {}^b\mathbf{g} - {}^b\boldsymbol{\omega} \times {}^b\mathbf{v} - {}^b\dot{\boldsymbol{\omega}} \times {}^b\mathbf{r}_S - {}^b\boldsymbol{\omega} \times ({}^b\boldsymbol{\omega} \times {}^b\mathbf{r}_S) \quad (1.19)$$

dove abbiamo messo in evidenza la derivata $\partial_t {}^b\mathbf{v}$ fatta rispetto al sistema mobile dove viene effettuata la misura.

1.4.2 Il sistema di visione

Il sistema di visione come ulteriore strumento di misura integrato permette di acquisire una misura assoluta a frequenze minori della piattaforma inerziale e con un errore contenuto; le misure effettuate con questo sensore sono gli output del nostro processo e dipendono dallo stato dinamico del veicolo, mentre le misure della IMU rappresentano gli input. Un sistema di visione è costituito da una o più telecamere; nel caso di monocamera la misura sarà una il solo bearing (due angoli), mentre in presenza di sistema stereo avremo anche misure di profondità.

La telecamera viene descritta matematicamente mediante quello che viene chiamato modello *pinhole*

Se un punto P ha coordinate ${}^c\mathbf{P} = [{}^cx, {}^cy, {}^cz]^T$ relativamente al sistema di riferimento della camera Σ_c che ha origine nel centro ottico, con l'asse z lungo l'asse ottico della lente, allora abbiamo che le coordinate di un punto P e la sua rappresentazione nel piano immagine $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$ sono collegate da quella che viene chiamata *proiezione prospettica ideale*

$$x = f \frac{{}^cx}{{}^cz} \quad y = f \frac{{}^cy}{{}^cz} \quad (1.20)$$

dove f è riferita alla *focale*.

Questo modello è chiamato *ideal pinhole*. Essa è una idealizzazione del modello delle lenti sottili poichè quando l'apertura decresce l'effetto dovuto alla diffrazione diventa predominante e tale approssimazione non vale più.

Si può utilizzare questo modello per simulare le misure di bearing provenienti dal sistema di

visione dove

$$\begin{aligned}\frac{{}^c x}{{}^c z} &= \tan \alpha \\ \frac{{}^c y}{{}^c z} &= \tan \beta\end{aligned}\tag{1.21}$$

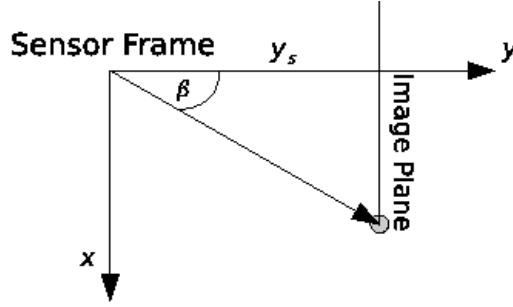


Figure 1.3: Sistema di riferimento canonico del modello pinhole.

Un sistema di due telecamere ci fornisce anche la ${}^c z$ per cui possono essere ricavate le coordinate tridimensionali della scena.

Nel momento in cui si dovrà operare con immagini reali acquisite la relazione che lega i pixel dell'immagine al punto espresso nel sistema di riferimento della camera è

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^c \mathbf{P} \\ 1 \end{bmatrix}\tag{1.22}$$

dove vengono messi in evidenza i parametri intrinseci della camera. Quando essi sono noti la camera è calibrata.

E' importante notare come nel presente lavoro dove si vuole studiare e implementare una soluzione algoritmica del problema dello SLAM, non si sono considerate le problematiche della *features detection* e il *features tracking* che sono comunque stati parzialmente studiati come problema disgiunto.

Un sistema di visione deve poter riconoscere lo stesso landmark in differenti frame acquisiti. Il problema non è semplice in quanto la forma e la luminosità possono variare a causa di trasformazioni (nel caso semplice affini) per effetto del moto e dell'ambiente circostante.

Idealmente un modello per la corrispondenza è fornito dall'equazione

$$I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_2)) \quad \forall \mathbf{x}_1 \in \Omega \cap h^{-1}(\Omega) \subset \mathbb{R}^2$$

dove la funzione h descrive la trasformazione del dominio dell'immagine.

In verità una descrizione più realistica considera un rumore aggiuntivo

$$I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_2)) + \nu(h(\mathbf{x}_1))\tag{1.23}$$

dove

$$\hat{h} = \arg \min_h \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} \|I_1(\tilde{\mathbf{x}}_1) - I_2(h(\tilde{\mathbf{x}}_2))\|^2\tag{1.24}$$

Definiamo *feature* quella regione $W(\mathbf{x})$ dove l'equazione

$$I_1(\tilde{\mathbf{x}}) = I_2(h(\tilde{\mathbf{x}}, \alpha)) \quad \forall \tilde{\mathbf{x}} \in W(\mathbf{x}) \quad (1.25)$$

determina unicamente il parametro α che rappresenta il tipo di trasformazione. In caso di semplice traslazione il *flusso ottico* $\mathbf{u}$ viene determinato dalla minimizzazione dell'equazione

$$E_b(\mathbf{u}) = \sum_{W(\mathbf{x})} [\nabla I^T \mathbf{u} + I_t]^2 \quad (1.26)$$

o analogamente⁴ il *feature tracking*

$$E_t(dx, dy) = \sum_{W(\mathbf{x})} [I(x + dx, y + dy, t + dt) - I(x, y, t)]^2 \quad (1.27)$$

Quando le features devono essere poste in corrispondenza dopo un tempo sufficientemente lungo, conviene trovare tali corrispondenze tra il frame iniziale e quello finale, senza l'analisi dei frame intermedi in quanto gli errori verrebbero accumulati. La deformazione tra la prima e l'ultima immagine non può essere semplicemente modellata da una traslazione; si usa invece una trasformazione affine.

Possiamo stimare i parametri della trasformazione minimizzando

$$E_a(A, \mathbf{d}) = \sum_{W(\mathbf{x})} [I_2(A\tilde{\mathbf{x}} + \mathbf{d}) - I_1(\tilde{\mathbf{x}})]^2 \quad (1.28)$$

e quindi applicare il metodo NCC (*normalized cross correlation*)

$$NCC(h) = \frac{\sum_{W(\mathbf{x})} (I_1(\tilde{\mathbf{x}}) - \hat{I}_1)(I_2(h(\tilde{\mathbf{x}})) - \hat{I}_2)}{(\sum_{W(\mathbf{x})} (I_1(\tilde{\mathbf{x}}) - \hat{I}_1)^2 \sum_{W(\mathbf{x})} (I_2(h(\tilde{\mathbf{x}})) - \hat{I}_2)^2)^{1/2}} \quad (1.29)$$

dove

$$\hat{I}_1 = \frac{1}{N} \sum_{W(\mathbf{x})} I_1(\tilde{\mathbf{x}})$$

$$\hat{I}_2 = \frac{1}{N} \sum_{W(\mathbf{x})} I_2(h(\tilde{\mathbf{x}}))$$

Come sensori da usare in differenti operazioni sono presi in considerazione sia un sistema stereo che un sistema mono camera; sarebbe infatti auspicabile considerare due diversi scenari.

Se le features devono essere stimate con un certo grado di precisione (ad esempio nel caso in cui il dirigibile acquisisce immagini per fotogrammetria su una striscia di superficie) sarebbe opportuno compatibilmente con alla quota operativa usare un sistema stereo. Con il setup riportato in tabella ??, focale 6mm e baseline 2m ed altezza operativa di 80 abbiamo una incertezza sulla determinazione delle coordinate del punto pari a $[1.840m, 1.297m, 1.085m]$.

	px	pixel size [10^{-6} m]	sensor size [mm]	FoV
sensor x	2050	3.45	7.07	61.03
sensor y	2450	3.45	8.45	70.32

Table 1.1: Camera setup

⁴Il problema posto in termini di flusso ottico è analogo a quello del tracking di una particolare feature

Se invece la stima delle features non richiede grande precisione, ad esempio nel caso in cui il veicolo debba solo effettuare uno spostamento da un sito ad un altro, volando ad altezza maggiore dove il sistema stereo non è utilizzabile, possiamo usare una sola camera con caratteristiche analoghe con un'ottica da 25 mm, e considerare pessimisticamente l'incertezza pari al footprint di 3 pixel ottenendo un errore di 0.02 gradi. I due casi differiscono anche nella tecnica di SLAM da applicare per la navigazione in quanto le misure fornite dai due sistemi non hanno lo stesso grado di informazione.

1.5 La formulazione del problema dello SLAM e la sua natura probabilistica

Il principale vantaggio dello SLAM è che elimina la necessità della conoscenza topologica a priori di un certo scenario. Se il dirigibile si muove attraverso un landscape prendendo misure di posizione relativa tra landmark mediante il sistema di visione è ovvio che le diverse stime di posizione sono necessariamente correlate l'una all'altra a causa degli errori sulla posizione del veicolo (Fig 1.4). L'implicazione di questo è molto profonda in quanto una soluzione consistente al problema della localizzazione del veicolo e della produzione della mappa richiedono uno *stato congiunto* composto dalla dinamica del veicolo e dalla posizione di ogni singolo landmark. Ciò richiede che lo stimatore implementi un vettore di stato di dimensioni ragguardevoli con un costo computazionale che va come il quadrato del numero del landmark.

Nella sua forma probabilistica il problema dello SLAM richiede che venga computata ad ogni nuova misura

$$p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})$$

la densità di probabilità congiunta

$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$$

dove

- $\mathbf{x}_k$: è il vettore di stato che descrive la posizione e l'orientazione del veicolo.
- $\mathbf{m}_i$: è il vettore che descrive la posizione dell' i -esimo landmark supposta tempo invariante.
- $\mathbf{z}_{ik}$: è l'osservazione dell' i -esimo landmark al tempo k
- $\mathbf{X}_{0:k} = [\mathbf{x}_0, \dots, \mathbf{x}_k] = [\mathbf{X}_{0:k-1}, \mathbf{x}_k]$: la storia passata delle posizioni del veicolo
- $\mathbf{U}_{0:k} = [\mathbf{u}_0, \dots, \mathbf{u}_k] = [\mathbf{U}_{0:k-1}, \mathbf{u}_k]$: la storia degli input del sistema
- $\mathbf{m} = [\mathbf{m}_1, \dots, \mathbf{m}_k]$: l'insieme di tutti i landmark
- $\mathbf{Z}_{0:k} = [\mathbf{z}_0, \dots, \mathbf{z}_k] = [\mathbf{Z}_{0:k-1}, \mathbf{z}_k]$: l'insieme di tutte le osservazioni dei landmark

Ovviamente una soluzione ricorsiva è desiderabile e sarà fornita dall'applicazione di un filtro di Kalman esteso.

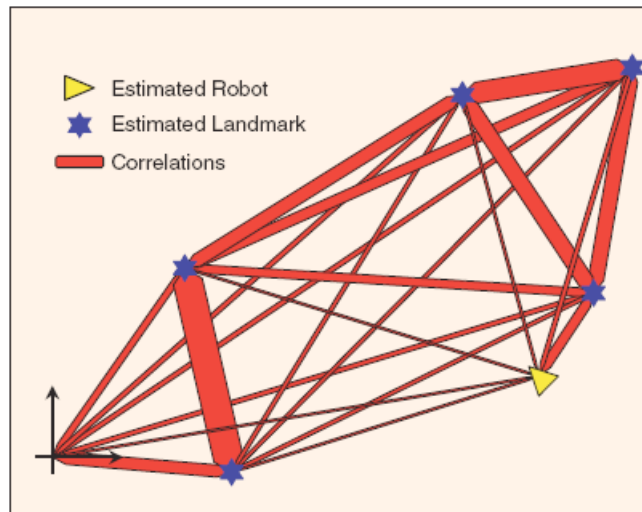


Figure 1.4: Diagramma che illustra la correlazione tra i diversi landmark [T. Bailey and H.F. Durant-Whyte, 2006].

1.5.1 Requisiti per la soluzione del problema

La soluzione del problema dello SLAM è basata sui seguenti risultati

- 1) Il determinante di ogni minore della matrice di covarianza della mappa decresce in maniera monotona ad ogni osservazione.
- 2) Nel limite in cui il numero delle osservazioni cresce le stime delle posizioni dei landmark diventano sempre più correlate.
- 3) La covarianza associata alla posizione di ogni singolo landmark è determinata unicamente dalla covarianza sulla stima della posizione del veicolo.

In particolare questi risultati mostrano che

- L'intera struttura dello SLAM dipende in maniera critica dal mantenere la completa conoscenza della cross-correlazione tra le stime della posizione dei landmark e del veicolo.
- Al procedere del veicolo attraverso il landscape gli errori nella stima di ogni coppia di landmark diventa più correlata.
- Gli errori nella stima dei landmark sono tra loro correlati; questo significa che data la conoscenza nella mappa di un singolo landmark tutti gli altri possono essere determinati.
- Al procedere del veicolo che misura ogni singolo landmark, l'errore nella stima della distanza relativa tra essi diminuisce monotonamente.
- Al convergere della mappa, gli errori sulla posizione relativa arrivano ad un limite inferiore determinato dall'incertezza sulla prima osservazione.

1.5.2 Soluzione basata su EKF

Si definiscono il vettore di stato

$$\mathbf{x}_k = [\mathbf{x}_v(k), \mathbf{m}(k)]^T = [\mathbf{x}_v(k), \mathbf{m}_1(k), \dots, \mathbf{m}_i(k)]^T \quad (1.30)$$

e un modello dinamico a tempo discreto

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k)) + \mathbf{w}(k) \quad (1.31)$$

Il modello dinamico dei landmark è stazionario

$$\mathbf{m}_i(k) = \mathbf{m}_i(k-1) \quad (1.32)$$

La fase di predizione del filtro è data da

$$\begin{bmatrix} \hat{\mathbf{x}}_v(k|k-1) \\ \hat{\mathbf{m}}(k|k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\hat{\mathbf{x}}(k-1|k-1), \mathbf{u}(k)) \\ \hat{\mathbf{m}}(k-1|k-1) \end{bmatrix} \quad (1.33)$$

per il vettore di stato, mentre per la matrice di covarianza

$$\mathbf{P}(k|k-1) = \nabla_{\mathbf{x}} \mathbf{f}(k) \mathbf{P}(k-1|k-1) \nabla_{\mathbf{x}} \mathbf{f}^T(k) + \nabla_{\mathbf{u}} \mathbf{f}(k) \mathbf{U}(k) \nabla_{\mathbf{u}} \mathbf{f}^T(k) + \mathbf{Q}(k) \quad (1.34)$$

L'update del filtro necessita il calcolo del modello della misura

$$\hat{\mathbf{z}}(k|k-1) = \mathbf{h}(\hat{\mathbf{x}}(k|k-1))$$

il calcolo del vettore innovazione e della covarianza dell'innovazione, rispettivamente

$$\boldsymbol{\gamma}(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k|k-1) \quad (1.35)$$

$$\mathbf{S}(k) = \nabla_{\mathbf{x}} \mathbf{h}(k) \mathbf{P}(k|k-1) \nabla_{\mathbf{x}} \mathbf{h}^T(k) + \mathbf{R}(k) \quad (1.36)$$

Le equazione per l'aggiornamento del vettore di stato e della covarianza sono

$$\mathbf{W}(k) = \mathbf{P}(k|k-1) \nabla_{\mathbf{x}} \mathbf{h}^T(k) \mathbf{S}^{-1}(k) \quad (1.37)$$

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{W}(k) \boldsymbol{\gamma}(k) \quad (1.38)$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k) \mathbf{S}(k) \mathbf{W}^T(k) \quad (1.39)$$

Quando viene effettuata una osservazione il sistema deve poter decidere se una feature è stata precedentemente osservata; in caso contrario l'osservazione può essere scartata o inizializzata come nuova feature. Questo procedimento viene chiamato *data association*.

Se $\mathbf{y}_j$ è una nuova misura, calcoliamo il vettore innovazione

$$\boldsymbol{\gamma}_{i,j} = \mathbf{y}_j - \mathbf{h}(\mathbf{m}_i) \quad (1.40)$$

dove $\mathbf{h}(\mathbf{m}_i)$ è la predizione della misura per tutte le feature presenti nella mappa.

Quindi si calcola la covarianza dell'innovazione

$$\mathbf{S}_i(k) = \nabla_{\mathbf{x}_a} \mathbf{h}(\mathbf{m}_i) \mathbf{P}(k|k-1) \nabla_{\mathbf{x}_a} \mathbf{h}(\mathbf{m}_i)^T + \mathbf{R}(k) \quad (1.41)$$

definiamo *normalized innovation square*, $n_{k,j}$

$$n_{i,j} = \boldsymbol{\gamma}_{i,j}^T \mathbf{S}_i(k)^{-1} \boldsymbol{\gamma}_{i,j} \quad (1.42)$$

e la *normalized distance*

$$d_{i,j} = n_{i,j} + \log(\det \mathbf{S}_i(k)) \quad (1.43)$$

Se

$$\min_i d_{i,j} < \tau \quad (1.44)$$

la misura è associata, oppure se

$$\min_i n_{i,j} > \tau \quad (1.45)$$

la misura rappresenta una nuova feature.

In quest'ultimo caso, lo stato del sistema va *aumentato* e una nuova matrice di covarianza inizializzata. La nuova feature viene inizializzata mediante una opportuna funzione di inizializzazione

$$\hat{\mathbf{m}}_i(k) = \mathbf{g}_i(\hat{\mathbf{x}}_v(k|k-1), \mathbf{z}(k)) \quad (1.46)$$

lo stato aumentato diventa quindi

$$\begin{bmatrix} \mathbf{x}_v(k|k-1) \\ \mathbf{m}(k|k-1) \\ \mathbf{g}(\mathbf{x}_v(k|k-1), \mathbf{z}(k)) \end{bmatrix} \quad (1.47)$$

mentre la covarianza

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{vv}(k|k-1) & \mathbf{P}_{vm}(k|k-1) & \mathbf{P}_{vv}^T(k|k-1)\nabla_{\mathbf{v}}\mathbf{g}^T(k) \\ \mathbf{P}_{vm}^T(k|k-1) & \mathbf{P}_{mm}(k|k-1) & \mathbf{P}_{vm}(k|k-1)\nabla_{\mathbf{v}}\mathbf{g}^T(k) \\ \nabla_{\mathbf{v}}\mathbf{g}(k)\mathbf{P}_{vv}^T(k|k-1) & \nabla_{\mathbf{v}}\mathbf{g}(k)\mathbf{P}_{vm}(k|k-1) & \nabla_{\mathbf{v}}\mathbf{g}(k)\mathbf{P}_{vv}(k|k-1)\nabla_{\mathbf{v}}\mathbf{g}^T(k) + \nabla_{\mathbf{z}}\mathbf{g}(k)\mathbf{R}(k)\nabla_{\mathbf{z}}\mathbf{g}^T(k) \end{bmatrix}$$

Il processo ricorsivo è illustrato nella figura 1.5

1.6 Le sottomappe

Come abbiamo già accennato, il costo computazionale del processo di update dello stato del sistema va come $O(n^2)$, dove n è il numero di features. Se la mappa da gestire non è di grandi dimensioni, la tecnica basata sul EKF ha un tempo di processamento accettabile, ma nel caso n cominciasse a crescere le performances degraderebbero velocemente. Per ovviare a questo problema e non limitare lo SLAM a mappe di piccole dimensioni, si utilizza una particolare variante del filtro di Kalman, il *filtro di Kalman esteso compresso* (CEKF). L'idea è di suddividere l'intera area interessata in regioni più piccole. Indichiamo con A , la regione R in cui si trova il veicolo e le otto regioni adiacenti, mentre con B il resto della mappa.

All'interno di A il processo di stima si applica allo stato $\mathbf{x}_A$ ed è identico al filtro EKF classico con la variante che il processo di predizione e update includono tre nuove quantità da aggiornare, le matrici Φ, Θ, Ψ . Queste matrici contengono in un certo senso le informazioni necessarie al filtro per aggiornare globalmente la mappa, lo stato $\mathbf{x}_B$ e la sua covarianza, quando il veicolo esce dalla zona R .

Con questa variante del filtro le performance del processo di computazione migliorano notevolmente.

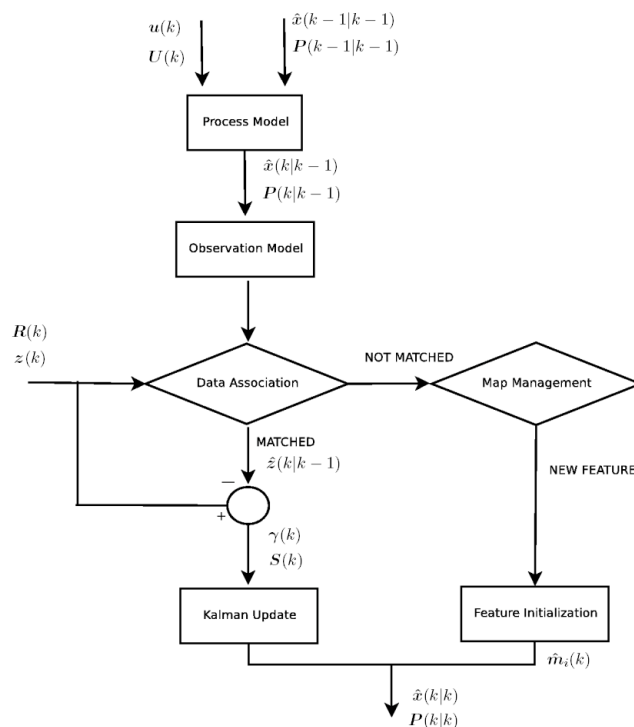


Figure 1.5: Diagramma a blocchi del processo di stima

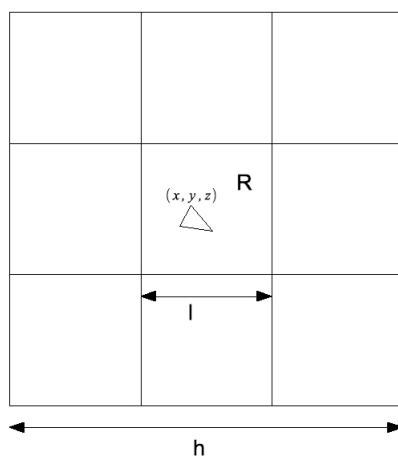


Figure 1.6: Le sottomappe; quando l'airship naviga nella regione R il filtro compresso include nello stato $\mathbf{x}_A$, lo stato del veicolo e gli stati relativi ai landmark che appartengono a R e alle otto regioni adiacenti. Un update globale è richiesto solamente quando il veicolo lascia la regione R e lo stato $\mathbf{x}_A$ cambia relativamente alla nuova regione R e le otto adiacenti.

1.7 Monocamera SLAM

Come già accennato ci possono essere delle situazioni in cui il sistema stereo non è utilizzabile per la navigazione, ad esempio se l'airship ha la necessità di aumentare la propria quota per potersi trasferire in un'altro sito di interesse. In questo caso per limitare gli errori di misura sarebbe necessaria una baseline troppo grande. Per questo motivo si può applicare una tecnica SLAM basata sull'utilizzo di una sola camera.

La singola camera fornisce solo misure di bearing, l'idea è quindi di usare un sottofiltro che viene invocato ogni volta che una nuova feature viene misurata, diciamo al tempo τ , in modo da stimare in qualche la profondità incognita; solo dopo aver determinato la misura di profondità la feature viene inclusa nel filtro principale, oppure viene scartata.

Il sistema per il sottofiltro segue il modello

$$\mathbf{x}_\tau^i(t) = \mathbf{x}_\tau^i(t-1) + \mathbf{w}_x \quad (1.48)$$

$$\lambda_\tau^i(t) = \lambda_\tau^i(t-1) + \mathbf{w}_\lambda \quad (1.49)$$

$$\mathbf{x}^i(t) = {}^i\mathbf{H}_s(t) {}^i\mathbf{H}_s(\tau)^s \mathbf{x}^i(t) \lambda^i(t) = \mathbf{h}({}^s\mathbf{x}^i(t), \lambda^i(t), \tau) \quad (1.50)$$

dove

- $\mathbf{x}^i$ è la misure di bearing della i -feature;
- λ^i è la misura di range incognita;
- $\mathbf{w}_x$ e $\mathbf{w}_\lambda$ rappresentano rumore gaussiano;

La terza equazione rappresenta il modello della misura.

L'idea di base è che quando il sottofiltro viene inizializzato al tempo τ e la feature viene riosservata in $t > \tau$, il sottofiltro mette in relazione lo stato cinematico del veicolo al tempo t con la misura in τ , dando una predizione quindi della misura al tempo t . Se la covarianza dell'errore di stima è minore di una certa soglia allora la feature viene inserita nel filtro principale.

1.8 Risultati

L'artefatto prodotto dallo studio delle tecniche SLAM per la navigazione è stato un modulo software (*State Estimation Module*) da integrare con il modulo relativo alla dinamica che fornisce i dati di input al sistema nel processo di filtraggio.

Come primo esperimento, è stata simulata la navigazione del veicolo, dotato di sistema stereo, nell'atmosfera di Titano ad una altezza di 80 m con vento 0.5 m/s. E' stata disegnata una traiettoria ottimale per un processo di mapping accurato in una zona di 800x800 metri (la scelta è stata guidata anche dai tempi di simulazione in ambiente MATLAB).

Si consideri che l'altezza scelta è quasi il limite di applicabilità della visione stereo con le ottiche e il CCD scelti.

I risultati sono riportati in Fig. 1.7 dove si può apprezzare il risultato della simulazione. L'errore minimo è di 0.21 m, quello medio di 2.82, mentre quello massimo è di 13.93 m. Il 94% delle features è stata stimata con un errore minore di 5 m. La seconda simulazione (fig.1.8) illustra i risultati dello SLAM con una camera sola. Nelle stesse condizioni ambientali della prima simulazione, si è ipotizzato di spostarsi ad una altezza di 200 m lungo una striscia di superficie di 2 km; è da notare come in questa situazione non ci sia alcuna loop-closure.

I risultati sono molto soddisfacenti in quanto l'errore medio è di circa 5 m. Il 58% delle features ha un errore tra i 5 e i 10 m mentre il 39% inferiore ai 5 m.

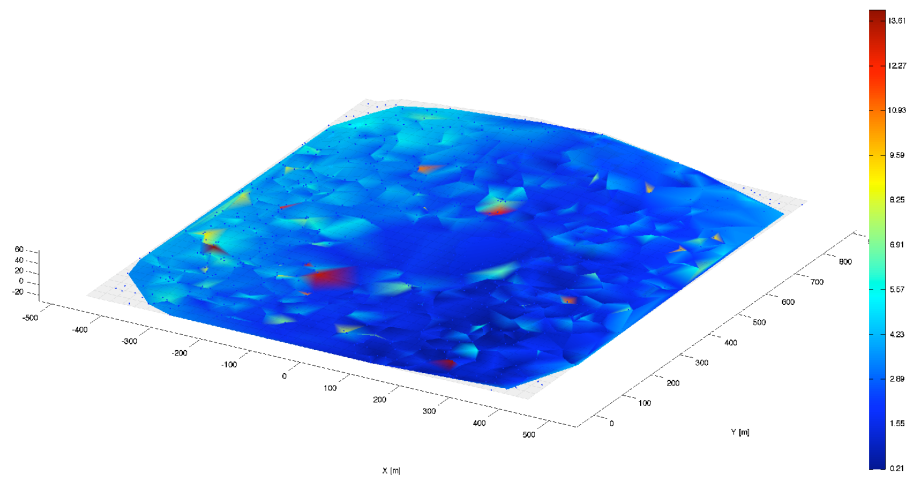


Figure 1.7: Stima delle feature mediante stereo SLAM in un grafico a gradiente di colore, l'errore medio è di 2.82 m

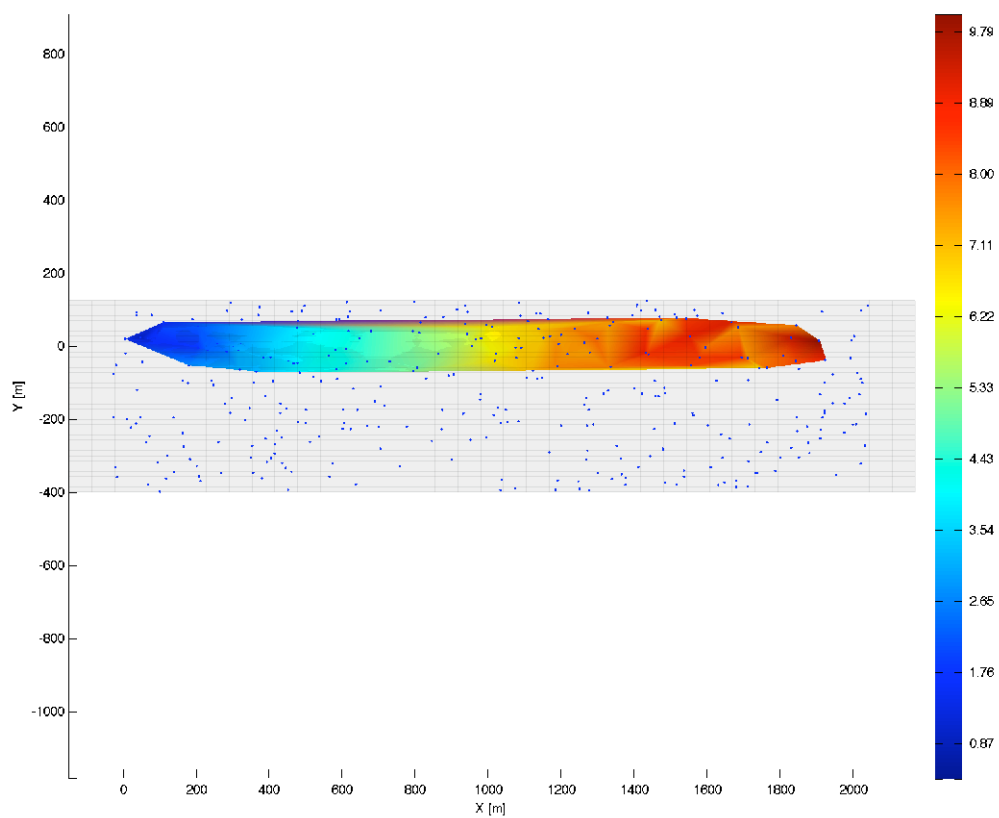


Figure 1.8: Stima delle feature mediante mono SLAM in un grafico a gradiente di colore, l'errore medio è di 2.82 m

Introduction

Unmanned robotic vehicle systems have evolved rapidly in the past two decades, with new technologies and applications being discovered at an ever increasing pace. Unmanned vehicle systems have been successfully developed and deployed for agricultural, military, nuclear, and other applications that are either hazardous in nature, or require an exacting level of precision or repeatability. Vehicle control and sensor systems have become more advanced and reliable with the increasing availability of smaller, more powerful computers. The modern computing capability and other technological advancements have led to autonomous vehicles equipped with global positioning systems capable of centimeter accuracy, detailed real-time updating terrain maps, and obstacle avoidance systems using three-dimensional vision processing and laser range finding sensors.

There are countless areas in which autonomous mobile agents might help to remove human operators from dangerous or hostile environments. This is especially true in the context of field robotic applications. Unmanned vehicles promise to allow often dangerous tasks to be performed from remote locations in a range of application domains such as mining, defence and sub sea exploration.

Early work in field robotics concentrated on remote piloting of platforms, and there is still a considerable amount of innovative work undertaken in this area.

With the advent of newer technologies, including a host of relatively cheap sensors and increases in computational speed, there has been a recent push to increase the level of autonomy with which remote agents are allowed to operate. This is seen in numerous application domains where systems are required to operate for long periods with little or no input from a human operator. From the landing of spacecraft on distant planets to submersible vehicles operating deep under our planet's oceans there is a need for systems capable of making decisions and taking control actions in an independent manner. A number of groups around the world have been concentrating their efforts on the development of field robotic applications and these are being taken up in a variety of industrial sectors. The deployment of autonomous systems in field environments demands high levels of robustness and system integrity. In order for these systems to be adopted into real world applications they must be shown to present a significant advantage in terms of safety, productivity and reliability.

One of the fundamental competencies required for a truly autonomous agent is the ability to navigate successfully within an environment. Navigation is the science of determining the course of a vehicle on the basis of information from a variety of sensors. Traditionally, mariners relied on sightings of stars to aid them navigate their ships reliably in unknown waters. This form of navigation relies on a known map of stellar bodies and is analogous to many map based localisation schemes that are prevalent in the autonomous robotics community. This scenario is similar to that encountered by many of today's mobile robotic systems that operate in previously unexplored environments. In the absence of prior map information, the vehicle must use its on-board sensing to aid in localisation. Autonomous navigation remains one of the

fundamental building blocks of these systems. This need for reliable, long term and autonomous navigation has motivated a considerable amount of research.

The navigation techniques are still designed based on dead reckoning (INS) and absolute information as GPS system. By fusing data from a GPS and INS, utilizing the two systems complimentary properties, a robust navigation system with both high accuracy and update rate may be obtained.

There is considerable research into dead reckoning and absolute sensors that can work reliably in different environments and weather conditions. Different sensors suites are required for each application, such as navigation in underground and open cut mining, underwater vehicles, ships and space vehicles. Each system will require a particular combination of dead reckoning and absolute sensors. In many robotics applications however, a vehicle (aerial and ground vehicle) needs to perform a task within environments where GNSS (Global Navigation Satellite System) information may not be available, such as indoors, in forest, underground, or other such locations where GNSS is naturally denied.

Autonomous operation of interplanetary exploration vehicles is essential due to the large time delays in communications, Mars for examples, may have a 35 minute round trip communication time. These time delays mean that teleoperation although very useful for rovers would not be suitable for aerobots because of the uncertainty in the aerobot's position between communications. This means that any aerobot will need to be fully autonomous, accepting high level commands, such as go to a given location and conduct these sets of experiments. In such cases autonomous navigation and localization is required; starting from the last decade a lot of works have been made to improve the capabilities of a new generation of sensors based on vision systems.

Visual system, in broad term, is a collection of devices that transform measurement of lights into information about spatial and material properties of a scene. Among these devices we need photosensitive sensors (camera or a retina) as well as computational mechanism that allow us to extract information from the raw sensory readings.

The complexity of the physical world is infinitely superior of the complexity of measurement of its images, so vision is more than difficult. We cannot simply invert the image formation process and reconstruct the true scene from a number of image. What we can reconstruct is a best *model* of the world or a its representation.

For instance, what model of the scene to infer depends on whether we want to use it to move within the environment, to visualize it from novel viewpoints or to recognize objects or materials. If we want to navigate in an unknown environment, we care for the shape and motion of obstacles, not so much for their material or the ambient light. Nevertheless, the latter influence the measurement, and have to be dealt with somehow.

The idea of endowing with a sense of vision to have them interact with human, robots and the environment in a dynamic fashion is not new. First, until just over a decade ago, there was no commercial hardware available to transfer a full-resolution image into the memory of a computer at frame rate, let alone to process it and do anything useful with it. This step has now been overcome; digital camera are ubiquitous, and so are powerful computers that can process their output in real time.

Visual and inertial sensing are two sensory modalities that can be explored to give robust solutions on image segmentation and recovery of 3D structure from images, increasing the capabilities of robotic systems and enlarging the application potential of vision systems. The "beauty" of combining these two sensor modalities is the complementary characteristics of camera and inertial sensors. On one hand, the inertial sensors have large measurement

uncertainty at slow motion and lower relative uncertainty at high velocities. Inertial sensors can measure very high velocities and accelerations. On the other hand, the cameras can track features very accurately at low velocities. With increasing velocity tracking is less accurate since the resolution must be reduced to obtain a larger tracking window with same pixel size and, hence, a higher tracking velocity.

A lot of work has done in 2-D autonomous navigation to improve capabilities in the localization and mapping of ground robots, but aerial applications are very young. As an aerobot can move freely in 3-D cartesian space, the solutions to navigation become more complex.

The simultaneous localization and mapping (SLAM) problem asks if is possible for a mobile robot to be placed in unknow location in an unknown enviroment and for the robot to incrementally build a consistent map of this enviroment while simulteneously determining its location within this map.

Accurate localisation is arguably the most fundamental competence required by autonomous vehicle systems. It forms the basis for most navigation and control decisions. Without accurate localisation, a mobile robot is essentially left to wander its environment with no notion of where it is and where it is going.

Another important competence for a mobile system is the ability to build and maintain maps of initially unknown environments. Maps allow the vehicle to plan its movement within the environment in order to achieve its goals. The automatic creation of maps allows the vehicle to model the world using its sensory resources. In some instances, the creation of an accurate map of the environment is a goal in itself while in many other circumstances the maps are used as a tool for performing other higher level tasks. Given accurate positioning information, the creation of a map of the environment is a straightforward undertaking. Given a current position estimate and an observation of the environment the observation can be fused into the existing map to create an updated map estimate.

Interest on the utilization of unmanned aerial vehicles (aerobot), has grown in the past few years, due to their potential utilization in surveillance, exploration, monitoring, and transportation tasks. In this context, efforts, have been directed to the development of semi-autonomousvehicles with on-board control and navigation systems capable of planning and executing trajectories based on high level human-planned missions. Despite the progress in this area, there is still an untapped potential in the form of airships, also known as blimps or lighter-than-air (LTA) vehicles, as unmanned robotic platforms. For our purposes LTA vehicles, as airships, the are an usefull experimetal platform for testing inertial navigation positioning and visual sensor.

Chapter 1

Aerobots: Concepts and Applications

1.1 Introduction

Aerobot technology is generating a good deal of interest in planetary and terrestrial exploration circles. Balloon based aerobots have much to offer to ESA's Aurora programme, e.g. high resolution mapping, landing site selection, rover guidance, data relay, sample site selection, payload delivery, and atmospheric measurement. Aerobots could be used in a variety of configurations from uncontrolled free-flying to tethered rover operation, and are able to perform a range of important tasks which other exploration vehicles cannot. Technically, a lighter than air (LTA) aerobot concept is attractive because it is low risk, low-cost, efficient, and much less complex than heavier than air (HTA) vehicles such as fixed wing gliders, and crucially, much of the required technology 'building blocks' currently exist. Smart imaging and localisation is a key enabling technology for remote aerobots. Given the current lack of comprehensive localisation and communication systems, it is important that aerobots are equipped with the ability to determine their location, with respect to a planet's surface, to a suitable accuracy and in a self-sufficient way. The availability of a variety of terrain feature extraction, point tracking, and image compression algorithms means that such a self-reliant system is now achievable.

1.2 Background and Motivation

For those planets and moons that support an atmosphere, flying robots are likely to provide a practical solution to the problem of extended planetary surface coverage for terrain mapping, and surface composition surveying. Not only could such devices be used for suborbital mapping of terrain regions, but they could be used to transport and deploy science packages or release microrovers at different geographically separated sites. While much attention has been given to the use of rovers for planetary exploration, the use of flying robots, or aerobots, for planetary exploration represent a highly innovative concept. Rovers are capable of travelling relatively small distances and much of a planet's terrain is impossible to small wheeled vehicles; aerobots in comparison have no such limitations.

Aerobots (balloon, blimp, small aircraft or helicopters), fill the gap between satellites and rovers. Where satellites typically obtain full planetary coverage with medium resolution, aerobots can cover a small area but with superior resolution. The advantages of aerobots over landers/rovers is that aerobots can land if is designed to, and perform typically the same measurement as rovers, whilst having the capability of travelling distances, non accessible by rovers.

Advances in technology areas will be assessed whether these technologies can improve the exist-

ing technical feasibility significantly. If we consider the latest advances in extremely light weight and flexible solar panels we have a dramatic improvement of the power over mass ratio of solar powered aeroglidors and blimps.

We can summarize the typical tasks of an aerobot for planetary or terrestrial exploration as follows

- acquire and store images of a surface at various resolution
- construct and update a 3D model of the surface (DEM)
- constantly estimate its position (latitude, longitude and altitude), attitude and motion respect to the surface for any control strategy
- decide on the base of the communication budget and information content which images and at which resolution/compression need to be transmitted to earth, satellite or ground station
- transport scientific payloads

The principal challenges for aerobot planetary exploration include:

- large communication latencies
- extended communication blackout periods
- extended mission duration
- operation in substantially unknown environments

These challenges impose the following capability requirements:

- *vehicle safing*, so that the safety and integrity of the aerobot can be ensured over the full duration of the mission and during extended communication blackout;
- *accurate safe and autonomous flight control* (for active aerobots), including deployment lift-off, landing, hovering/station-keeping, surface approach, and long traverses;
- *spatial mapping and self-localization* in the absence of a global positioning system;
- *advanced perceptual servoing*, allowing the aerobot to detect and avoid atmospheric and topographic hazards, and also to identify, home in, and keep station over pre-defined science targets or terrain features.

1.3 Simultaneous Localization and Mapping (SLAM)

When the environmental conditions are appropriate, terrain mapping of an unknown environment is desirable. In fact from various planetary scenarios, a lot of critical operations could be done safely if that kind of information is available.

High resolution terrain mapping can be the main payload of flying devices in a wide variety of terrestrial applications too: fine geographic survey, environmental analysis, mine detection and localization, exploring the nature of and the hazard to aviation of dust plumes of active volcanoes, military surveillance and intelligence gathering, and high altitude communication platforms.

The simultaneous localization and mapping (SLAM) problem asks if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map.

Terrain mapping is also a way to achieve precise localization of the flying robot, by providing environment references, thus enabling a position estimation with bounded errors as the robot flies.

Finally, mapping is a prerequisite to the development of cooperative air/ground robotics ensembles.

The main difficulty to build high resolution terrain maps is to precisely determine the sensor position and orientation as it moves. Dead reckoning techniques, that integrate over time the data provided by motion estimation sensors, such as wheel encoders for rovers or inertial sensors, are not sufficient for that purpose.

Indeed, not only the position estimate they provide between successive data acquisitions may be not precise enough, but they are intrinsically prone to generate position estimates with unbounded error growth.

Motion estimation techniques that use vision for features tracking or matching have recently been proposed in the context of ground rovers but applications to aerial vehicle are in a intense phase of investigation.

These techniques allow to get a more precise position estimate between successive data acquisitions, but their errors also cumulate over time, since they do not memorize any environment feature.

The only solution to guarantee bounded errors on the position estimates is to rely features that are detected on a stable environment and memorize them as the vehicle moves. It has early been understood in the robotic community that the problems of mapping such features and estimating the robot location are intimately tied together, and that they must therefore be solved in a unified manner. For our purposes LTA vehicles are a useful experimental platform for testing inertial navigation positioning and visual sensor. In vehicle such balloon or blimp the SLAM problem can be solved using the same techniques as for indoor and land vehicles, because their low dynamics. So, we can trade between different sensors drivers (single camera, stereo camera, radar altimeter) having more time than higher speed vehicle, for on board real time computation.

1.4 Aerobot Architectures Concepts Overview

There is a strategic gap in current exploration technologies for systems that can combine extensive coverage with high-resolution data collection and in-situ science capabilities. For planets and moons with an atmosphere, this gap can be addressed through aerial vehicles. In the Solar System, in addition to Earth, the planets Venus and Mars, and the Saturn moon Titan have significant atmospheres. Aerial vehicles that have been considered for planetary exploration include airplanes and gliders, helicopters, balloons and airships.

1.4.1 LTA Vehicles

Lighter-than-atmosphere (LTA) systems provide significant advantages for planetary and terrestrial exploration due to their potential for extended mission durations, long traverses and

extensive surface coverage capabilities. Unfortunately they cannot fly directly toward target of interest, on the other hand they can perform surface feature tracking because image capture frequency is compatible with their low baseline speed.

The environment in which LTA vehicles operate will dictate its capabilities and will be the determining factor in the LTA vehicle's feasibility. Items such as atmospheric density, atmospheric composition, temperature, and solar radiation are the main characteristics that will their design and mission capabilities. The initial and most basic environmental requirement for an LTA vehicle is the presence of an atmosphere. Now let's have an overview of the principal architecture of LTA vehicles.

1.4.1.1 Balloons

Balloons, have as its main source of lift what is called aerostatic lift, i.e., lift that is independent of flight speed. Unlike the lift force generated over a wing surface which is directly proportional to the square of the flight speed, aerostatic lift comes from Archimedes' Principle: it can be calculated by multiplying the volume of air displaced by the lifting gas, by the difference in density between such gas and air. That is also why such force is still known today as buoyancy. Therefore, only gases that are lighter than air, i.e., whose densities are lower than air at a given temperature and pressure, can be used. One of the earliest concepts for aerobots to be used as exploration devices for planetary surfaces is balloons. During the last couple of decades different kinds of balloons were proposed[E.Laan et al., 2004].

The *Solar Montgolfiere balloon* uses solar heat. The balloon's envelope is made from a material that traps heat from sunlight or from a planet surface. The Solar Montgolfiere balloon will sink at night and during the day the balloon will rise again due to solar heating. A guiderope attached to the bottom of the gondola containing the payload will anchor the balloon during the night. The main disadvantage of the Solar Montgolfiere balloon is the short mission duration time.

Another concept uses *superpressure balloons*. This kind of balloon is ideally suited for long duration flights. The balloon will fly at a constant atmospheric density altitude during any diurnal change in temperature, i.e. the balloon will not lose altitude during the night like the Solar Montgolfiere balloon. The superpressure balloon is completely sealed and as the lifting gas of the balloon is heated by radiation, convection and conduction during the flight the envelope material must withstand the increased gas pressure inside the balloon. Superpressure balloons work because they are essentially constant volume systems and during the night after the gas has cooled the balloon remains pressurised. No ballasting is required to maintain altitude as long as the balloon remains pressurised. The altitude of the balloon is dictated by the strain in the balloon envelope due to pressure changes. As long as the balloon is pressurised the mission will continue.

In the *reversible fluid balloon*, the envelope is connected to a reservoir containing a fluid which is easily vaporised. The balloon will gain altitude when the fluid is vaporised into gas and drops in altitude by condensing the gas back into the fluid phase. The main advantage of this concept to the other two balloons is the possibility to control the altitude of the balloon. The cost of having this capability is increased system complexity and therefore an increase in risk.

1.4.1.2 Airships

The airship is a direct descendant from balloon technology but it could control its trajectory by fuel burning engines. In order to retain the same buoyancy force upwards when ascends through the atmosphere (where decreases ambient pressure) it expands the lifting gas volume by letting out air (in Earth flies, but in general a gas more dense than the lifting gas) out of the “ballonets” which are simply bags inside the main envelope.

Like balloons airships suffer the temperature effects: decreasing of aerostatic lift as the temperature increases or superheating after long exposure to direct sunlight or a rapid ascent through the atmosphere [S.B.V. Gomes and J.Ramos, 1998].

When the airship is powered by fuel-burning engines, it will undergo a mass decrease over a given period of operation. If not properly accounted for, it can come to land with the buoyancy force far exceeding the weight of the complete vehicle. In that case, the landing operation can become quite a hazardous activity, to be avoided at all costs, since it could ultimately mean valving off expensive lifting gas. There are different strategies to prevent this occurrence as using ballast or deflecting thrust propulsors.

Airships use the same aerodynamic controls as conventional aircraft as aerodynamic control surfaces

- rudders and elevators attached to the empennage surfaces for yaw and altitude control respectively;
- vectored thrust which is simply a rotation of the propulsion units about a horizontal axis
- bow and/or stern thrusters for fine control features for landing and docking operations

Much has been said about an airship’s apparent lack of maneuverability under strong winds or in the face of weather fronts. This usually derives from a wrong perception: a buoyant vehicle, whose overall mass is very close to that of the displaced air would behave like a bubble of soap, floating and drifting away carried by the elements. In fact, that may happen to an airship that is strongly underpowered as it does with conventional aircraft as well.

Realistically, the airship is vulnerable to winds much in the same circumstances as conventional aircraft are, i.e., during take-off and landing operations when the speed is low and the winds are described as light and variable.

To achieve long duration flight i.e for planetary explorations, not only will innovative power and propulsion systems be needed but also innovative airship designs that take advantage of the environment in which they are operating. A long duration airship for planetary exploration will need to be matched to and take advantage of the environment in which it is to fly [A. Colozza , 2004].

The requirements for the airship were that it would need to be capable of station-keeping at a given location as well as operating for an extended period of time of 50 earth days or more. Two types of propulsion / power systems were at the moment considered. These were a solar photovoltaic system and a nuclear radioisotope system. The isotope system can be considered for all of the potential planetary locations whereas the photovoltaic system would only be applicable to Venus and Mars. Beyond Mars the solar intensity is The ratio between airship’s fin area and volume guides its design. For the solar powered airship, it is assumed [A. Colozza , 2004] that portions of the solar array would be located on the upper surface of the lower two fins and on both sides of the vertical fin.

1.4.2 HTA Vehicles

Heavier-than-atmosphere unmanned systems (HTA-UAV) like small aircraft or helicopters are used in different terrestrial applications. They have proven their usefulness in military reconnaissance i.e aerial photography used for the research on wildlife habitats, but their practical applications have been expanding to more than military uses. Various sizes of UAVs are designed to different levels of performance depending on their application. HTA vehicles are desirable for their controllability but their operating time depends obviously on propellant availability. The power required to fly at a constant height, does not only relate to the lift to drag ratio (the L/D ratio determines how far an aircraft can fly without propulsion) but also has a linear relation to the relative velocity of the aircraft to the air. So i.e flying in thin air does not only require more thrust (rocket engine), but also more power (propeller).

The military has shown the most recent interest in small UAVs (SUAVs) for many reasons. A SUAV is much more portable than its large counterparts and requires only one operator. A smaller reconnaissance plane can assess ground targets at a closer range without being detected. Therefore, most SUAVs use electric motors as a propulsion system, which allows for a stealthier and more reliable flight with little engine failure. Also an SUAV is less expensive and can be considered a disposable asset. This factor allows SUAV pilots to navigate hostile areas and focus on their primary mission, rather than plane recovery. In addition to military applications, size and cost advantages are attracting civilian and private uses. Therefore, SUAVs are most suitable for use in non-military applications because they are less expensive and less dangerous.

Many SLAM implementations use small aircraft that could fly at approximately 50 m/s at 100 meters of altitude and have an endurance of the order of 45 min with 20 kg of payload. In this scenario ground features were installed with enough separation ($\geq 50\text{m}$) to relieve the on-line data association problem for the high velocity. SLAM prediction and update are performed in real-time with a low number of features but if the latency of the filter update becomes significant then accumulated inertial/vision data are processed as a block.

Within the context of SLAM, “loop closing” is the task of deciding whether or not a vehicle has, after an excursion of arbitrary length, returned to a previously visited area. Reliable loop closing is both essential and hard. It is without doubt one of the greatest impediments to long term, robust SLAM. For a vehicle with high velocity it is hard to estimate with bounded uncertainty the feature position by successive re-observations in a single fly over, because FOV limitations; so further observations of the same feature are needed and so loop-closure is in some way necessary. Results show that the uncertainties of the vehicle and map further decrease monotonically due to successive re-observations but also to loop closure [J.Kim and S. Sukkarieh, 2007].

For vehicles as balloons or airships loop closure is not a binding constraint, this is the reason for which they are useful experimental platforms for testing inertial navigation and vision system.

1.5 Aerobot Applications

1.5.1 Venus Exploration

The original motivation for developing a new class of robotically controlled balloon was to advance the exploration of Venus. Following the exploration of the surface of Venus by short lived Soviet landers and by the VEGA balloons, the U.S.A. carried out the Magellan mission which mapped the surface of Venus using radar sensors. The radar revealed a surface with a great variety of structural and volcanic features. Venus, Earth’s estranged sister planet, has a dense atmosphere exceeding 92 bars in pressure and surface temperatures in excess of 470°C . Its surface is obscured from view at visible wavelengths by high altitude haze and clouds as

well as the molecular scattering of the clear atmosphere beneath. The Soviet landers VEGA functioned for less than two hours on the surface. With advanced thermal techniques and the use of vacuum insulation, it may be possible to extend surface lifetime to 1-2- weeks. Much longer lived systems, however, will require radioisotopic power and temperature control systems which will be costly, expensive and present environmental concerns.

An aerobot can turn the environmental challenges of Venus to advantage, it would make brief excursions to the hot surface environment of Venus to acquire data and return to the cooler, higher altitudes to telemeter those data to an orbiting relay station or directly to earth. A solar powered airship can be considered for flight on Venus [A. Colozza , 2004]. For a solar powered airship, since there is a thick atmosphere and cloud cover, is assumed that all solar radiation below the clouds is diffuse. Therefore below the cloud layer, there is no variation in array output based on array or airship position relative to the location of the sun. With the solar powered airship, below the cloud layer, the availability of power is the driving factor for the airship design.

Due to the high-density environment of the Venus atmosphere, the lift produced by the envelope volume was more than sufficient to lift the airship or balloons and their associated systems. In fact much of the envelope would need to be filled with atmospheric gas with only a small volume utilized by the lighter lifting gas.

NASA's road map includes a Venus Mobile Explorer mission (VME) addressed for 2020. This later launch date would provide adequate time to develop the extreme environment technologies, such high-temperature electronics and power systems, needed for this long-lived mission operating near the surface. VME would be a precursor of a new frontier for Venus in-situ exploration missions.

Although NASA is still considering various mobility options for VME, the long traversing capabilities of an air mobility platform (e.g. situated in a blimp or balloon's gondola) and its ability to explore terrain with a variety of topographic conditions and bearing strength offer major advantages over a surface rover.

1.5.2 Titan Exploration

Titan [A. Coustenis and R. D. Lorenz, 1998] is the largest satellite of Saturn, and unique in the solar system in that it is the only satellite with a substantial atmosphere. This atmosphere is both interesting, in that it is the only significant nitrogen atmosphere in the solar system other than that of Earth and also is host to extensive organic photochemistry; these photochemical products form a thick haze which until recently has impeded remote sensing of the surface.

Its density has been determined at 1880 kg/m^3 . The surface gravitational acceleration is 1.35 ms^{-2} at the surface, or about $1/7$ that of Earth. The surface atmospheric pressure is 1.5 bar with a temperature of 94K and a density of 5.4 kg/m^3 .

The composition is predominantly molecular nitrogen, with a few per cent methane, an undetermined amount (less than a few per cent) of argon, and, interesting for its exploration, traces of many organic compounds.

Even ignoring the crucial trafficability concerns on Titan, rovers can't satisfy the global access requirement so a near-surface aerial vehicle is indicated, LTA or HTA, (HTA with a mass of $\simeq 100 \text{ kg}$ [R. D. Lorenz, 2000]).

Even well before the present Cassini-Huygens era, the possibility of using balloons to explore Titan had been an appealing concept, perhaps with expendable *drop-sonde* to perform surface science.

Possible trajectories of passive balloons in Titan's troposphere are simulated with the instan-

taneous wind field predicted by a GCM (General Circulation Model). In most areas the basic motion of a balloon is a predominantly eastward or westward drift, depending on altitude, latitude and season of the balloon release point. Some meridional oscillation is always superposed on this basic motion, resulting in a wavy trajectory, with a maximum extent (of 40°) at high latitudes of the winter hemisphere[R. D. Lorenz and T. Tokano, 2006].

A preliminary analysis[R. D. Lorenz, 2000] underscore that for a Titan's mission, *data selection and compression* are perhaps more critical technology requirements than improvements in propulsion or aerodynamics.

Also an helicopter (HTA vehicle) or tilt-rotor architecture has been proposed; it offers excellent precision landing capability, although has flight power too high for continuous flight[R. D. Lorenz, 2000]. TANDEM is an L-class ESA mission to explore in situ Titan and Enceladus, two of the most interesting bodies in the Saturnian system, indeed in the solar system overall.

It's as a multi-component exploration system to be built with international partners for a launch in 2021 or later. The baseline mission concept is for two moderately-sized spacecraft: a Titan-Enceladus Orbiter (carrying the Enceladus landers), a the Titan in situ investigation elements (the Titan hot air balloon and three mini-probes).

1.5.3 Mars Exploration

The atmosphere of Mars is very thin. Near the surface on Mars the atmospheric density is similar to the density of Earth's atmosphere at 30 km. The atmosphere is made up almost entirely of carbon dioxide. The temperature on Mars is on average much colder than on Earth. Although at certain times of the year and locations the temperature will rise above freezing, most of the time temperatures are well below the freezing point of water. The gravitational force on Mars is about $1/3$ what it is on Earth. Therefore the lift that an HTA vehicle must generate is only $1/3$ what is needed on Earth. This reduced gravity (and hence lower lift requirement) is a large benefit for this type of vehicles; with an atmosphere barely dense enough it's possible an HTA flight even at very low altitudes.

However the environmental conditions on Mars are not all beneficial. The low atmospheric density poses a significant problem for a LTA vehicles. Due to the low atmospheric density on Mars, designing and flying an airship that relies on buoyancy for lift is a difficult task. This is especially true for a solar powered airship because of the low solar intensity at Mars. To meet the station-keeping mission, the airship would need to collect enough energy during the day to provide power throughout the nighttime period. This also requires an energy storage system such as a regenerative fuel cell system. It was assumed that the airship, if feasible on Mars, would operate near the surface where the wind velocity is 2m/s. This is a representative wind speed near the Martian surface based on data returned from the various landers and rovers that have operated on the surface. Buoyancy control schemes have been identified including: superpressure balloons, chemical interactions with the CO_2 atmosphere to produce ballast, hot air balloons, and secondary balloons using periodic buoyancy gas replenishment from reservoirs and gas release and balloon deflation. A new architecture for Mars exploration is based on Directed Aerial Robot Explorer (DARE) platforms: autonomous balloons with path guidance capabilities that can carry heavy scientific payloads and deploy swarms of miniature robotic probes over multiple target areas[Alexey A. Pankine et al., 2005].

1.5.4 Cooperative Autonomous Ground-Air-Space Robots

Several different kinds of robotic explorers are used today, or will be used in the future, those include orbiters, ground rovers and aerobots. The potential advantages of heterogeneous robots teams include the potential to share sensors, explore large areas simultaneously.

Space or air robot are physically bigger than rovers and therefore will have greater ability to generate power that can accommodate more computing resources and higher communication bandwidth; they have a wider field of view and are therefore in a better position to make longer-term plans compared to rovers.

A new heterogeneous ground-air-space robot team architecture is proposed [S. Joshi et al., 2004] in which ground rovers cooperate with intelligent overhead robots to work on a previously *unknown terrain* in such a way that past experience of one rover can be used to help later rovers.

Specific air-ground coordination issues in multiple robot teams also comprise a new research area. The localization of both ground robots and air robots using communication has been identified as an important challenge.

We can see a possible robot-team architecture in figure 2.1 The aerobot serves the role of

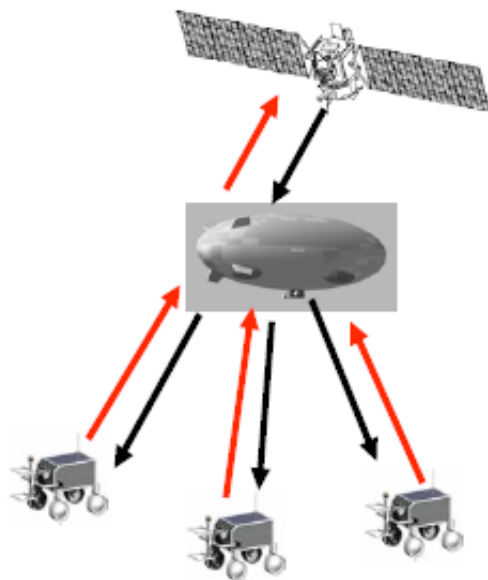


Figure 1.1: Artwork of a multiple heterogeneous robot system for planetary exploration mission. Photo: NASA/JPL UC Davis

creating maps from several different sources of information within the multiple robot team: overhead spacecraft images, aerobot images, and ground rover obstacle detection.

The aerobot will be able to capture middle resolution images of ground features and its cameras may be able to zoom in on specific sites to get more information about obstacles and science sites. The entire aerobot may climb to a different altitude for additional viewing opportunities [A. Elfes et al., 2003]. Overhead spacecraft low resolution images/wide field of view could be used to plan navigation for the aerobot itself, or for longer range planning for the ground rovers. The aerobot instead may then obtain the entire space-based image and parse out the section it needs to consider.

Space, air, and ground vehicle information is fused onboard the aerobot to create a map of the terrain under consideration. Localization is an important consideration for ground-air-space robot teams. In the proposed architecture, the aerobot localizes the rovers (within some

uncertainty tolerance) using its imaging capabilities. Also, the rover could perform its own self localization and transmits this information to the aerobot. The aerobot then uses optimal information fusion techniques to combine the estimates into a better estimate.

Using the aerobot map and the localization of rovers and obstacles, the aerobot then computes an optimal path for the rover to follow to reach its goal and avoid obstacles. Common methods of path planning in robotics include graph search techniques, artificial potential field approaches, and heuristic rule-based approaches.

The overhead spacecraft is used to store images, maps, navigational routes created by the aerobot, backup, redundancy in case of failure and so a critical role for lightening the aerobot storage system.

1.5.5 Terrestrial Application

There is undoubtedly a regain of interest in developing aerobots for terrestrial application.

Military and civil interests in developing HTA aerobots are especially addressed towards territory monitoring, surveillance, transportation and exploration tasks which require displacements in a short time as well as precise control of the vehicle.

On the other hand, airships and balloons are today employed in various missions such as observation of urban areas or the battleship; they play a significant role in surveillance, monitoring, transportation and communications.

Other tasks that an LTA aerobot could perform are fire detection, rescue in general, transport payload for science purposes i.e. for climate studies, atmosphere and ground (terrain and ice) analysis or exploring the nature of dust plumes of active volcanoes.

They require less energy compared to HTA vehicles so they are feasible for low cost missions.

For our purposes, LTA vehicles are a useful experimental platform for testing inertial navigation and visual sensor fusion. In fact, vehicles such as balloons or blimps, the SLAM problem can be solved using the same techniques as for indoor and land vehicles, because their low dynamics and so having more time than higher speed vehicles, for on-board real time computation.

Chapter 2

Vehicle Dynamics Modeling

The use of unmanned aerial vehicles has grown in the past years with an increasing interest in LTA vehicles both for planetary exploration and for terrestrial application. The motivation behind this is that LTA vehicles offer more advantages in power budgeting and longevity in general for a space mission but also offer a better implementation of the autonomous navigation techniques that include a vision system and an IMU sensor enabling a better feature tracking with SLAM. Furthermore their low speed is very useful for station keeping tasks, environment monitoring, climate research etc.

Airship dynamics and control is a crucial point to model a realistic navigation system which takes into account the characteristics of the motions of this kind of vehicle in a selected environment. Literature in this field is rich and poor at the same time. Many studies are dated from the beginning of the 20th century where first applications of hydrodynamics take place. Authors as Max Munk provided simplified methods to describe the dynamical behaviour of forces and moments acting on airship hull under certain simplifications in the physical conditions. Most of efforts of the various authors were addressed to determine the aerodynamic forces and moments due to pressure and viscosity. While an estimate of pressure induced forces could be derived and modelled from hydrodynamic theory, the determination of viscous forces required to deal with experimental data. In this sense a lot of works dealt with force and pressure distribution measurements on number of different fuselage forms of varying slenderness ratio, varying rearward position of maximum thickness and varying nose ratio. With these profiles a lot of semi-empirical relations were derived to establish the limit of applicability of the potential theory and where a model for viscous forces has to be applied instead, for normal forces and axial forces.

The concept of added mass has played an important role in all theoretical approach, which has to be considered when a body moves through a fluid otherwise at rest; there's a certain amount of kinetic energy of the fluid caused by the motion of the body and when it's not constant (accelerated motion) the body makes a positive work on the fluid. Added mass is the principal cause of pressure induced forces on airships.

With the contribution of numerical integrators it's possible to compute the pressure forces and moments without an explicit analytical expression for it, simply including in the integration of the equation of motions the added mass contribution.

2.1 Coordinate Systems and Trasformations

To navigate succesfully, there must be a way for describing the position with respect to the enviroment in general. For our purposes it is more practical to refer to a vehicle navigating on a perfect sphere shape planet even if a lumpy spheroid is a more accurate description of a planet. However this description is not very useful for navigation. A smooth model amenable to a relatively simple mathematical description is needed, so we refer to a reference ellipsoid. The constants defining the reference ellipsoid (size and shape) are determined using extensive geodetic and satellite observations that allow to obtain the spherial harmonic coefficients (SAC) defining the gravity field; such data are well known for Earth but less for other Solar System Planets. The reference ellipsoid could be biaxial, as in the case of Earth, or triaxial as in case of Mercury.

Usually we refer, for semplicity, to a vehicle navigating on a rotating spherical Earth; it's possible to extend these definitions to another spherical rotating planet.

2.1.1 ECEI Reference Frame

ECEI system (*Earth-Centered-Earth-Inertial*) originates at the center of the Earth and is designated with versors ${}^i\hat{e}_x, {}^i\hat{e}_y, {}^i\hat{e}_z$. The fundamental plane is Earth's equator. The ${}^i\hat{e}_x$ versor points to vernal equinox; the ${}^i\hat{e}_y$ is $\pi/2$ to the East in the equatorial plane; ${}^i\hat{e}_z$ extends through the North Pole.

The equinox and the plane of the equator move very slightly over the time so the term *pseudo-inertial* could be used.

2.1.2 ECEF Reference Frame

ECEF system (Earth Centered Earth Fixed) remains fixed to the rotating Earth, and rotates relative to the ECEI frame at the rotation rate of the Earth ω_e . We recall that Greenwich sideral time is the rigth ascension of Greenwich

$$\alpha_g = \omega_e t$$

Vector from ECEI and ECEF systems are related with the relation

$${}^i\mathbf{r} = R_3(\alpha_g) {}^f\mathbf{r} = {}^i\mathbf{R}_f {}^f\mathbf{r} \quad (2.1)$$

where

$${}^i\mathbf{R}_f = \begin{bmatrix} \cos \alpha_g & -\sin \alpha_g & 0 \\ \sin \alpha_g & \cos \alpha_g & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If a point P is described in a spherical parametrization (ϕ, λ, r) where $\phi \in [-\pi/2, \pi/2]$ is the geographic latitude and $\lambda \in [0, 2\pi]$ the geographic longitude, we have

$${}^f x = r \cos \phi \cos \lambda \quad (2.2)$$

$${}^f y = r \cos \phi \sin \lambda \quad (2.3)$$

$${}^f z = r \sin \phi. \quad (2.4)$$

2.1.3 Topocentric Reference Frame

This system (SEZ, South East,Zenith) is useful if observing the veichle from Earth's surface and is used extensively with sensor systems.

The SEZ system rotates with the site and we indicate it as ${}^t\hat{e}_x, {}^t\hat{e}_y, {}^t\hat{e}_z$.

The ${}^t\hat{e}_y$ points to South from the site (even in the Southern Emisphere). The ${}^t\hat{e}_z$ points East, while ${}^t\hat{e}_x$ points radially outward from the site, aligned the site's local vertical.

In the SEZ system we can define as “look angles“ the angles to view the vehicle from a ground station. The *azimuth*, β , is the angle measured from north, **clockwise** to the location beneath the vehicle, and conventionally has values to be from 0° to 360° .

The *elevation*, el , is the angle measured from the local horizon plane, positive up to the object,

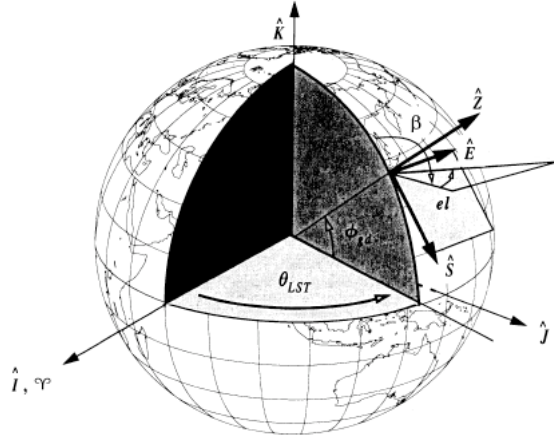


Figure 2.1: SEZ Topocentric System. Image: Vallado[D.A. Vallado, 2001]

it has values from -90° to 90° and sometimes is called *altitude*. A *zenith distance* is also defined as the complementary of elevation, $\pi/2 - el$. The transformation matrix from SEZ system to inertial is

$${}^i\mathbf{R}_t = {}^i\mathbf{R}_f {}^f\mathbf{R}_a {}^a\mathbf{R}_t \quad (2.5)$$

where the auxiliary transformation is

$${}^f\mathbf{R}_a = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

and then

$${}^a\mathbf{R}_t = \begin{bmatrix} \cos(\pi - \phi) & -\sin(\pi - \phi) & 0 \\ \sin(\pi - \phi) & \cos(\pi - \phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$${}^i\mathbf{R}_f = \begin{bmatrix} \cos \alpha_g & -\sin \alpha_g & 0 \\ \sin \alpha_g & \cos \alpha_g & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The convention is dictated to simplify the construction of multiple rotation. In fact is known from theory that the columns of a generic rotation matrix ${}^a\mathbf{R}_b$ are the versors of the frame (b) expressed in the frame (a).

If we remember only the z-axis rotation matrix we can manage rotations around a reference axis considering a transition matrix which represent a transformation between an initial frame and a final frame that has the z-axis along the direction of rotation. This is typical the typical approach adopted in robotics applications[R.Da forno, 2001].

2.1.4 North East Down Reference Frame

The North East Down Reference frame (NED), has versors ${}^n\hat{e}_x$, ${}^n\hat{e}_y$, ${}^n\hat{e}_z$ where ${}^n\hat{e}_z$ points toward East, ${}^n\hat{e}_y$ points toward North and ${}^n\hat{e}_x$ towards down (nadir).

The transformation matrix from NED system to inertial is

$${}^i\mathbf{R}_n = {}^i\mathbf{R}_f {}^f\mathbf{R}_a {}^a\mathbf{R}_n \quad (2.6)$$

where

$${}^f\mathbf{R}_a = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

and

$${}^a\mathbf{R}_n = \begin{bmatrix} \cos(\pi - \phi) & -\sin(\pi - \phi) & 0 \\ \sin(\pi - \phi) & \cos(\pi - \phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.1.5 Vehicle Reference Frame

The vehicle frame (VF) is the reference frame fixed to the vehicle. To simplify the writing of rigid body equation of motion we choose versors ${}^b\hat{e}_x$, ${}^b\hat{e}_y$, ${}^b\hat{e}_z$ of VF aligned with the principal axis of the vehicle.

2.2 Rigid Body Dynamics

In this section we will apply Euler first and second axioms to derive the rigid body equation of motions. The problem is reported in many books so it will be only mentioned here in order to allow us to describe in detail the motion of a rigid body in a fluid. Consider a body fixed coordinate system rotating with angular velocity ${}^b\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ about an Earth-fixed coordinate system.

It is assumed that the mass is constant in time ($\dot{m} = 0$). For a rigid body satisfying this condition the distance from the origin of the body fixed frame to vehicle center of gravity can be defined as

$${}^b\mathbf{r}_G = \frac{1}{m} \int_V {}^b\mathbf{r} \rho dV \quad (2.7)$$

For marine and LTA vehicles it is desirable to derive the equations of motion for an arbitrary origin in a local body fixed coordinate system to take advantage of the vehicle geometrical properties. Since the hydrodynamic and kinematic forces and moments are given in the body fixed reference frame it is convenient to formulate Newton's law in this frame.

When deriving the equations of motion it will be assumed that: (1) the vehicle is rigid and (2) the earth-fixed reference frame is inertial. In guidance and control applications in space it is

usual to use star-fixed reference system or a reference frame rotating with the Earth, while the marine vehicles usually are related to an Earth-fixed reference frame. To derive with respect to time a vector in a body fixed rotating frame we have the well known relation

$${}^b\dot{\mathbf{v}} = \partial_t {}^b\mathbf{v} + {}^b\boldsymbol{\omega} \times {}^b\mathbf{v} \quad (2.8)$$

where the symbol ∂_t is the time derivative in the moving body frame. It's obvious that

$${}^b\dot{\boldsymbol{\omega}} = \partial_t {}^b\boldsymbol{\omega} + {}^b\boldsymbol{\omega} \times {}^b\boldsymbol{\omega} = \partial_t {}^b\boldsymbol{\omega} \quad (2.9)$$

For the translational motion we have that the force equation from Newton's Second Law is¹

$$m(\partial_t \mathbf{v}_0 + \boldsymbol{\omega} \times \mathbf{v}_0 + \partial_t \boldsymbol{\omega} \times \mathbf{r}_g + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_G)) = \mathbf{f}_0 \quad (2.10)$$

where $\mathbf{v}_0$ is the velocity of the body reference frame. From the *Euler's second Axiom* we obtain after some algebra the classical equations of rotational motion

$$I\partial_t \boldsymbol{\omega} + \boldsymbol{\omega} \times (I\boldsymbol{\omega}) + m\mathbf{r}_G \times (\partial_t \mathbf{v} + \boldsymbol{\omega} \times \mathbf{v}_0) = \mathbf{m}_0 \quad (2.11)$$

Equations could be written as

$$(m\dot{\mathbf{v}} - m\tilde{\mathbf{r}}_G\dot{\boldsymbol{\omega}}) = \mathbf{f}_0 - m[\boldsymbol{\omega} \times \mathbf{v} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_G)] \quad (2.12)$$

$$(I\dot{\boldsymbol{\omega}} + m\tilde{\mathbf{r}}_G\dot{\mathbf{v}}) = \mathbf{m}_0 - [\boldsymbol{\omega} \times I\boldsymbol{\omega} + m\mathbf{r}_G \times (\boldsymbol{\omega} \times \mathbf{v})] \quad (2.13)$$

or in a more compact form

$$\begin{bmatrix} m\mathbb{I}_{3 \times 3} & -m\tilde{\mathbf{r}}_G \\ m\tilde{\mathbf{r}}_G & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_0 - m[\boldsymbol{\omega} \times \mathbf{v} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_G)] \\ \mathbf{m}_0 - [\boldsymbol{\omega} \times I\boldsymbol{\omega} + m\mathbf{r}_G \times (\boldsymbol{\omega} \times \mathbf{v})] \end{bmatrix} \quad (2.14)$$

where in the second member we have pointed out the inertial forces. Since we express all quantities in the body frame and the time derivative are always expressed with respect to the moving frame, we'll indicate for semplicity of notations

$$\partial_t {}^b\mathbf{v} = \dot{\mathbf{v}}$$

¹Now we express all equation in the body frame; the covariant reference simbol is omitted for simplicity

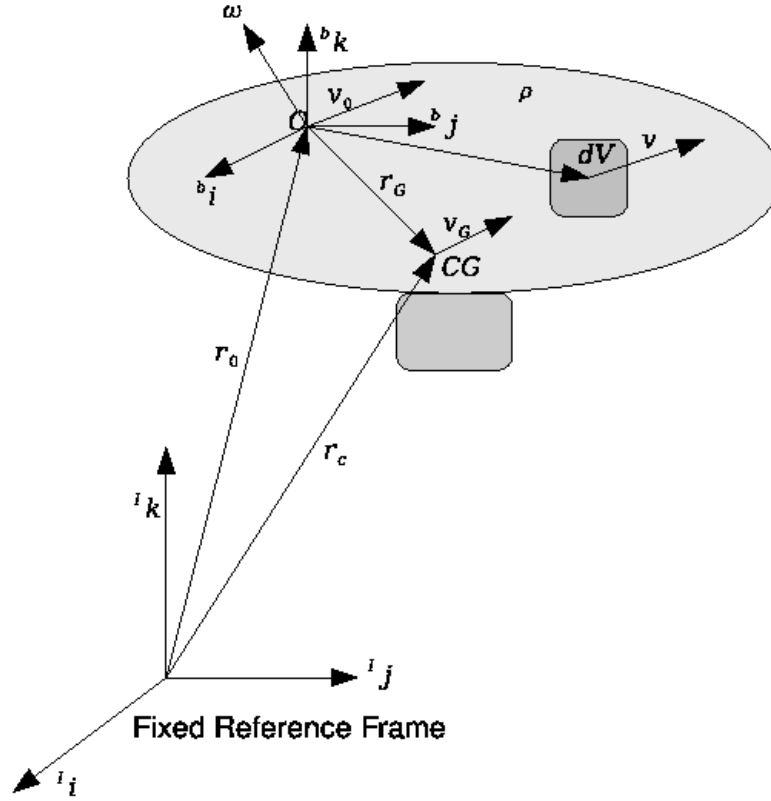


Figure 2.2: Rigid body reference frame and fixed reference frame

2.3 Rigid Body Kinematics

There are many ways to express the attitude of the vehicle. In particular we could describe an arbitrary orientation of the body frame respect an inertial one with a sequence of basic rotations. We refer to *Yaw*(ϕ), *Pitch*(θ), *Roll*(ψ) angles to represent the attitude of the veicle in the senquence, from inertial frame to the body frame,

$$\phi \rightarrow \theta \rightarrow \psi$$

So the rotation matrix from body to inertial is

$${}^1\mathbf{R}_b = R(\phi)R(\theta)R(\psi) \quad (2.15)$$

with

$$R(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

$$R(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.17)$$

$$R(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{bmatrix} \quad (2.18)$$

Now we have to find the relation between the rate of change of Euler angles and the angular velocity of the vehicle.

The idea is to consider small changes of the Euler angle, and determine the effects of the rotation vector, so we could write

$${}^b\boldsymbol{\omega} = {}^b\boldsymbol{\omega}_\phi + {}^b\boldsymbol{\omega}_\theta + {}^b\boldsymbol{\omega}_\psi \quad (2.19)$$

$${}^b\boldsymbol{\omega} = \begin{bmatrix} \dot{\psi} \\ 0 \\ 0 \end{bmatrix} + R(\psi)^\top \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\psi)^\top R(\theta)^\top \begin{bmatrix} 0 \\ 0 \\ \dot{\phi} \end{bmatrix} \quad (2.20)$$

so we get

$$\dot{\boldsymbol{\phi}} = \begin{bmatrix} -\sin \theta \\ \sin \psi \cos \theta \\ \cos \psi \cos \theta \end{bmatrix} \quad \dot{\boldsymbol{\theta}} = \begin{bmatrix} 0 \\ \cos \psi \\ -\sin \psi \end{bmatrix} \quad \dot{\boldsymbol{\psi}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (2.21)$$

or in matrix form

$${}^b\boldsymbol{\omega} = \begin{bmatrix} -\sin \theta & 0 & 1 \\ \sin \psi \cos \theta & \cos \psi & 0 \\ \cos \psi \cos \theta & -\sin \psi & 0 \end{bmatrix} \begin{bmatrix} \omega_\phi \\ \omega_\theta \\ \omega_\psi \end{bmatrix} = \mathbf{E}(\phi, \theta, \psi) \begin{bmatrix} \omega_\phi \\ \omega_\theta \\ \omega_\psi \end{bmatrix} \quad (2.22)$$

It is useful to calculate the inverse transformation to express the time derivative of Euler angles from angular velocity

$$\mathbf{H} = \mathbf{E}^{-1} = \begin{bmatrix} 0 & \frac{\sin \psi}{\cos \theta} & \frac{\cos \psi}{\cos \theta} \\ 0 & \cos \psi & -\sin \psi \\ 1 & \sin \psi \tan \theta & \cos \psi \tan \theta \end{bmatrix} \quad (2.23)$$

2.3.1 Attitude parametrizations

Euler symmetric parameters provide a very convenient parametrization of the attitude. They are more compact than the direction cosine matrix, because only four parameters, rather than nine, are needed.

They are more convenient than the Euler axis and angle parametrization because the expression for the direction cosine matrix in terms of Euler symmetric parameters do not involve trigonometric functions which require time-consuming computer operations. In fact we have that the rotation matrix from a frame (b) to a frame (a) is

$${}^a\mathbf{R}_b = {}^b\mathbf{R}_a^T(\phi, \theta, \psi)$$

where ${}^a\mathbf{R}_b(\phi, \theta, \psi)$ could be expressed by Euler angles ϕ, θ, ψ which refers to the rotation sequence "ZXZ" or "313".

$$\begin{aligned} {}^a\mathbf{R}_b &= {}^b\mathbf{R}_a^T(\phi, \theta, \psi) \\ &= \begin{bmatrix} \cos \psi \cos \phi - \cos \theta \sin \psi \sin \phi & -\sin \psi \cos \phi - \cos \theta \cos \psi \sin \phi & \sin \theta \sin \phi \\ \cos \phi \sin \phi + \cos \theta \sin \psi \cos \phi & -\sin \psi \sin \phi + \cos \theta \cos \psi \cos \phi & -\sin \theta \cos \phi \\ \sin \theta \sin \psi & \sin \theta \cos \psi & \cos \theta \end{bmatrix} \end{aligned}$$

Alternatively using quaternions we write

$${}^a\mathbf{R}_b(\mathbf{q}) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_3q_0) & 2(q_1q_3 + q_2q_0) \\ 2(q_1q_2 + q_3q_0) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_1q_0) \\ 2(q_1q_3 - q_2q_0) & 2(q_2q_3 + q_1q_0) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.24)$$

Usually Euler Angles are more "human readable", because they give a visual representation of the attitude. So in most cases it would be desirable to have, before integrations, initial attitude conditions expressed by Euler parameters.

We need an algorithm that allows, from the knowledge of Euler matrix elements, to derive quaternions parameters before CPU calculations.

Summing diagonal elements of (2.24) we have

$$\begin{aligned} 2(q_0^2 + q_1^2) - 1 + 2(q_0^2 + q_2^2) + 2(q_0^2 + q_3^2) - 1 &= \text{Tr}({}^a\mathbf{R}_b) \\ \Rightarrow 4q_0^2 + 2(q_0^2 + q_1^2 + q_2^2 + q_3^2) - 2 &= \text{Tr}({}^a\mathbf{R}_b) + 1 \\ \Rightarrow q_0^2 &= \frac{\text{Tr}({}^a\mathbf{R}_b) + 1}{4} \end{aligned} \quad (2.25)$$

$$(2.26)$$

where Tr is the matrix trace.

We have to distinguish two cases. 1) $q_0 \neq 0$

From (2.25) summing and subtracting in the left side the quantity r_{11} of (2.24) we obtain

$$\begin{aligned} 4q_0^2 - 2r_{11} + 2r_{11} &= \text{Tr}({}^a\mathbf{R}_b) + 1 \\ \Rightarrow 4q_0^2 - 4(q_0^2 + q_1^2) &= \text{Tr}({}^a\mathbf{R}_b) - 1 - 2r_{11} \\ \Rightarrow q_1^2 &= \frac{1 + 2r_{11} - \text{Tr}({}^a\mathbf{R}_b)}{4} \end{aligned} \quad (2.27)$$

similary we have

$$q_2^2 = \frac{1 + 2r_{22} - \text{Tr}({}^a\mathbf{R}_b)}{4} \quad (2.28)$$

$$q_3^2 = \frac{1 + 2r_{33} - \text{Tr}({}^a\mathbf{R}_b)}{4} \quad (2.29)$$

$$(2.30)$$

subtracting the diagonal terms of (2.24) we have for example

$$r_{32} - r_{23} = 4q_0q_1 \quad (2.31)$$

from which

$$q_1 = (r_{32} - r_{23})/4q_0 \quad (2.32)$$

$$q_2 = (r_{13} - r_{31})/4q_0$$

$$q_3 = (r_{21} - r_{12})/4q_0$$

Equations (2.32), once we fix the sign of q_0 , could be used to calculate the other parameters; when q_0 tends to zero these equations diverge. So in a more stable an algorithm that uses (2.27), (2.28), (2.29) where the sign ambiguity is solved by (2.32).

For example if $r_{32} > r_{23}$ then $q_1 > 0$ and we take the positive roots of (2.27).

2) $q_0 = 0$

Summing out diagonal terms of (2.24)

$$\begin{aligned} r_{21} - r_{12} &= 4q_1q_2 \\ r_{31} - r_{13} &= 4q_1q_3 \\ r_{32} - r_{23} &= 4q_2q_3 \end{aligned} \quad (2.33)$$

while for diagonal element of (2.24), considering that $q_0 = 0$, we have

$$\begin{aligned} q_1^2 &= (r_{11} + 1)/2 \\ q_2^2 &= (r_{22} + 1)/2 \\ q_3^2 &= (r_{33} + 1)/2 \end{aligned} \quad (2.34)$$

From quaternions normalization we have that at least one is different from zero and so we can choose, with arbitrary sign, from (2.34), the one which has greater value (for a more stable algorithm) and calculate the other parameters from (2.33)

2.4 General 6 DOF Rigid Body Equations of Motion

Equation (2.10) and (2.11) are usually written, according to the SNAME notation, as:

$$\begin{aligned} \mathbf{f}_0 &= \boldsymbol{\tau}_1 = [X, Y, Z]^\top \\ \mathbf{m}_0 &= \boldsymbol{\tau}_2 = [K, M, N]^\top \\ \mathbf{v} &= \boldsymbol{\nu}_1 = [u, v, w]^\top \\ \boldsymbol{\omega} &= \boldsymbol{\nu}_2 = [p, q, r]^\top \\ \mathbf{r}_G &= [x_G, y_G, z_G]^\top \end{aligned}$$

Applying this notation to (2.10) and (2.11) yields:

$$\begin{aligned} X &= m[\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] \\ Y &= m[\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] \\ Z &= m[\dot{w} - uq + vp - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] \\ K &= I_x\dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\ &\quad + m[y_G(\dot{w} - uq - vp) - z_G(\dot{v} - wp + ur)] \\ M &= I_y\dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \\ &\quad + m[z_G(\dot{u} - vr - wq) - x_G(\dot{w} - uq + vp)] \\ N &= I_z\dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} \\ &\quad + m[x_G(\dot{v} - wq - ur) - y_G(\dot{u} - vr + wq)] \end{aligned} \quad (2.35)$$

These equations can be expressed in a more compact vectorial form as:

$$\mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_{RB} \quad (2.36)$$

where $\boldsymbol{\nu} = [\boldsymbol{\nu}_1 \ \boldsymbol{\nu}_2] = [u, v, w, p, q, r]$ expresses in the body fixed frame the linear and angular velocity and $\boldsymbol{\tau}_{RB} = [X, Y, Z, K, M, N]$ is the generalized vector of external forces and moments.

The 6x6 generalized *inertia matrix* can be isolated easily from (2.35) and its parametrization is *unique* and satisfies

$$\mathbf{M}_{RB} = \mathbf{M}_{RB}^T > 0; \quad \dot{\mathbf{M}}_{RB} = 0$$

where

$$\mathbf{M}_{RB} = \begin{bmatrix} m\mathbb{I}_{3 \times 3} & -m\tilde{\mathbf{r}}_G \\ m\tilde{\mathbf{r}}_G & \mathbf{I} \end{bmatrix} \quad (2.37)$$

Here $\mathbb{I}_{3 \times 3}$ is the identity matrix, $\mathbf{I}$ is the inertia tensor with respect to O (usually the center of volume of the hull) and $\tilde{\mathbf{r}}_G \in SS(3)$.

2.4.1 Kirchoff Equations

Consider a vehicle with body fixed linear velocity $\boldsymbol{\nu}_1 = [u, v, w]$ and angular velocity $\boldsymbol{\nu}_2 = [p, q, r]$. Hence the force $\boldsymbol{\tau}_1$ and moment $\boldsymbol{\tau}_2$ are related to kinetic energy

$$T = \frac{1}{2} \boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\nu} \quad (2.38)$$

by the vector equations

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial T}{\partial \boldsymbol{\nu}_1} \right) + \boldsymbol{\nu}_2 \times \frac{\partial T}{\partial \boldsymbol{\nu}_1} &= \boldsymbol{\tau}_1 \\ \frac{d}{dt} \left(\frac{\partial T}{\partial \boldsymbol{\nu}_2} \right) + \boldsymbol{\nu}_2 \times \frac{\partial T}{\partial \boldsymbol{\nu}_2} + \boldsymbol{\nu}_1 \times \frac{\partial T}{\partial \boldsymbol{\nu}_1} &= \boldsymbol{\tau}_2 \end{aligned} \quad (2.39)$$

These are the *Kirchhoff's equations of motion*. Notice that (2.39) does not include the gravitational forces.

Now we want to prove the equivalence of the Eulerian (2.10),(2.11) and Lagrangian (2.39) approach to the equations of motion of a rigid body. Consider

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}$$

as the inertia matrix and express the kinetic energy of the body as

$$T = \frac{1}{2} \boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\nu} = \frac{1}{2} (\boldsymbol{\nu}_1^T \mathbf{M}_{11} \boldsymbol{\nu}_1) + \frac{1}{2} (\boldsymbol{\nu}_1^T \mathbf{M}_{12} \boldsymbol{\nu}_2) + \frac{1}{2} (\boldsymbol{\nu}_2^T \mathbf{M}_{21} \boldsymbol{\nu}_1) + \frac{1}{2} (\boldsymbol{\nu}_2^T \mathbf{M}_{22} \boldsymbol{\nu}_2) \quad (2.40)$$

Then we get

$$\frac{\partial T}{\partial \boldsymbol{\nu}_1} = \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\nu}_1} (\boldsymbol{\nu}_1^T \mathbf{M}_{11} \boldsymbol{\nu}_1) + \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\nu}_1} (\boldsymbol{\nu}_1^T \mathbf{M}_{12} \boldsymbol{\nu}_2) + \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\nu}_1} (\boldsymbol{\nu}_2^T \mathbf{M}_{21} \boldsymbol{\nu}_1) + \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\nu}_1} (\boldsymbol{\nu}_2^T \mathbf{M}_{22} \boldsymbol{\nu}_2) \quad (2.41)$$

The first term of the sum could be written as²

$$\begin{aligned} \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\nu}_1} (\boldsymbol{\nu}_1^T \mathbf{M}_{11} \boldsymbol{\nu}_1) &= \frac{1}{2} \left[\frac{\partial \boldsymbol{\nu}_1^T}{\partial \boldsymbol{\nu}_1} \mathbf{M}_{11} \boldsymbol{\nu}_1 + \boldsymbol{\nu}_1^T \frac{\partial (\mathbf{M}_{11} \boldsymbol{\nu}_1)}{\partial \boldsymbol{\nu}_1} \right] \\ &= \frac{1}{2} (\mathbf{M}_{11} \boldsymbol{\nu}_1 + \boldsymbol{\nu}_1^T \mathbf{M}_{11}) \\ &= \mathbf{M}_{11} \boldsymbol{\nu}_1 \end{aligned} \quad (2.42)$$

$$(2.43)$$

²Remember that $\mathbf{M} = \mathbf{M}^T$ so any sub-matrix of $\mathbf{M}$ is orthogonal

and so the others

$$\frac{1}{2} \frac{\partial}{\partial \boldsymbol{\nu}_1} (\boldsymbol{\nu}_1^\top \mathbf{M}_{12} \boldsymbol{\nu}_2) = \frac{1}{2} \mathbf{M}_{12} \boldsymbol{\nu}_2 \quad (2.44)$$

$$\frac{1}{2} \frac{\partial}{\partial \boldsymbol{\nu}_1} (\boldsymbol{\nu}_2^\top \mathbf{M}_{21} \boldsymbol{\nu}_1) = \frac{1}{2} \boldsymbol{\nu}_2^\top \mathbf{M}_{21} = \frac{1}{2} \mathbf{M}_{12} \boldsymbol{\nu}_2 \quad (2.45)$$

Finally we have

$$\frac{\partial T}{\partial \boldsymbol{\nu}_1} = \mathbf{M}_{11} \boldsymbol{\nu}_1 + \mathbf{M}_{12} \boldsymbol{\nu}_2 \quad (2.46)$$

and in the same manner

$$\frac{\partial T}{\partial \boldsymbol{\nu}_2} = \mathbf{M}_{22} \boldsymbol{\nu}_2 + \mathbf{M}_{21} \boldsymbol{\nu}_1 \quad (2.47)$$

Now, using the last two relations and the definition of the inertia matrix (2.37) the first Kirhhoff's equation becomes

$$\begin{aligned} & \mathbf{M}_{11} \dot{\boldsymbol{\nu}}_1 + \mathbf{M}_{12} \dot{\boldsymbol{\nu}}_2 + \boldsymbol{\nu}_2 \times (\mathbf{M}_{11} \boldsymbol{\nu}_1 + \mathbf{M}_{12} \boldsymbol{\nu}_2) \\ &= m \dot{\boldsymbol{\nu}}_1 - m \tilde{\mathbf{r}}_G \times \dot{\boldsymbol{\nu}}_2 + m(\boldsymbol{\nu}_2 \times \boldsymbol{\nu}_1) + m \boldsymbol{\nu}_2 \times (\boldsymbol{\nu}_2 \times \mathbf{r}_G) \\ &= \boldsymbol{\tau}_1 \end{aligned} \quad (2.48)$$

The latter equation is the euler equation of translational motion.

Similarly for the second Kirhhoff's equation we obtain

$$\begin{aligned} & \mathbf{M}_{22} \dot{\boldsymbol{\nu}}_2 + \mathbf{M}_{21} \dot{\boldsymbol{\nu}}_1 + \boldsymbol{\nu}_2 \times (\mathbf{M}_{22} \boldsymbol{\nu}_2 + \mathbf{M}_{21} \boldsymbol{\nu}_1) + \boldsymbol{\nu}_1 \times (\mathbf{M}_{11} \boldsymbol{\nu}_1 + \mathbf{M}_{12} \boldsymbol{\nu}_2) \\ &= \mathbf{I} \boldsymbol{\nu}_2 + m \mathbf{r}_G \times \dot{\boldsymbol{\nu}}_1 + \boldsymbol{\nu}_2 \times (m \mathbf{r}_G \times \boldsymbol{\nu}_1) + \boldsymbol{\nu}_2 \times (\mathbf{I} \boldsymbol{\nu}_2) - m \boldsymbol{\nu}_1 \times (\mathbf{r}_G \times \boldsymbol{\nu}_2) \\ &= \mathbf{I} \boldsymbol{\nu}_2 + m \mathbf{r}_G \times \dot{\boldsymbol{\nu}}_1 + \boldsymbol{\nu}_2 \times (\mathbf{I} \boldsymbol{\nu}_2) + m \mathbf{r}_G \times (\boldsymbol{\nu}_2 \times \boldsymbol{\nu}_1) \\ &= \boldsymbol{\tau}_2 \end{aligned} \quad (2.49)$$

where we used the property of the triple product so that

$$\begin{aligned} & \boldsymbol{\nu}_2 \times (m \mathbf{r}_G \times \boldsymbol{\nu}_1) - m \boldsymbol{\nu}_1 \times (\mathbf{r}_G \times \boldsymbol{\nu}_2) \\ &= \mathbf{r}_G (\boldsymbol{\nu}_2 \cdot \boldsymbol{\nu}_1) - \boldsymbol{\nu}_1 (\boldsymbol{\nu}_2 \cdot \mathbf{r}_G) - \mathbf{r}_G (\boldsymbol{\nu}_2 \cdot \boldsymbol{\nu}_1) + \boldsymbol{\nu}_2 (\boldsymbol{\nu}_1 \cdot \mathbf{r}_G) \\ &= \boldsymbol{\nu}_2 (\boldsymbol{\nu}_1 \cdot \mathbf{r}_G) - \boldsymbol{\nu}_1 (\boldsymbol{\nu}_2 \cdot \mathbf{r}_G) \\ &= \mathbf{r}_G \times (\boldsymbol{\nu}_2 \times \boldsymbol{\nu}_1) \end{aligned} \quad (2.50)$$

2.4.2 Coriolis and Centripetal Matrix

Let $\mathbf{M}$ be an inertia matrix defined as

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}$$

From equations (2.14), (2.39), (2.48) and (2.49) it is easy to see that we can define a matrix $\mathbf{C}(\boldsymbol{\nu})$ such as

$$\mathbf{C}(\boldsymbol{\nu}) \boldsymbol{\nu} \triangleq \begin{bmatrix} \boldsymbol{\nu}_2 \times \frac{\partial T}{\partial \boldsymbol{\nu}_1} \\ \boldsymbol{\nu}_2 \times \frac{\partial T}{\partial \boldsymbol{\nu}_2} + \boldsymbol{\nu}_1 \times \frac{\partial T}{\partial \boldsymbol{\nu}_1} \end{bmatrix} \quad (2.51)$$

and using the *skew operator* $\mathbf{S} \in SS(3)$ ³ we have

$$\mathbf{C}(\boldsymbol{\nu}) \boldsymbol{\nu} \triangleq \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathbf{S}(\frac{\partial T}{\partial \boldsymbol{\nu}_1}) \\ -\mathbf{S}(\frac{\partial T}{\partial \boldsymbol{\nu}_1}) & -\mathbf{S}(\frac{\partial T}{\partial \boldsymbol{\nu}_2}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \end{bmatrix} \quad (2.52)$$

³We also use the $\tilde{\cdot}$ operator instead

The $C(\boldsymbol{\nu})$ matrix expresses the inertial terms (Coriolis and Centripetal) of (2.14). Finally using relations (2.46) and (2.47) we obtain

$$C(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -S(M_{11}\boldsymbol{\nu}_1 + M_{12}\boldsymbol{\nu}_2) \\ -S(M_{11}\boldsymbol{\nu}_1 + M_{12}\boldsymbol{\nu}_2) & -S(M_{21}\boldsymbol{\nu}_1 + M_{22}\boldsymbol{\nu}_2) \end{bmatrix} \quad (2.53)$$

Application of (2.53) with $\mathbf{M} = \mathbf{M}_{RB}$ yields the expression for $\mathbf{C}_{RB}$ where

$$\mathbf{C}_{RB}(\boldsymbol{\nu}) = -\mathbf{C}_{RB}(\boldsymbol{\nu})^\top \quad \forall \boldsymbol{\nu} \in \mathbb{R}^6$$

2.5 General Principles of Hydrodynamics

In all the practical problems to be discussed, only the most general principles of hydrodynamics are used and in practically all cases the problems are reduced to questions involving only energy and momentum. It may be worth while to deduce the few equations necessary. Since air is a fluid, the pressure is everywhere perpendicular to any surface through which it is transferred. If u, v, w are components of the velocity of flow at any point it's worth the continuity principle⁴,

$$\nabla \cdot \mathbf{v} = 0$$

The entire energy is kinetic and therefore,

$$T = 1/2\rho \int (u^2 + v^2 + w^2) d\tau$$

where $d\tau$ is an element of volume of the fluid.

By Newton's law of motion, considering a flux parallel to the x -axis and so the mean pressure p acting on the face $dydz$ and $p + \partial_x p dx$ in the opposite side, we have

$$\rho \frac{du}{dt} dx dy dz = [p - (p + \partial_x p dx)] dydz = -\partial_x p dx dy dz \quad (2.54)$$

or

$$\rho \frac{du}{dt} = -\partial_x p \quad (2.55)$$

This may be written

$$\rho du = -\partial_x (p dt) \quad (2.56)$$

The impulse per unit area in the time dt is, by definition $p dt$. So the infinitesimal change in velocity du can be considered as produced by the infinitesimal impulse $p dt$, and a finite velocity u may be considered as produced from a state of rest by the finite impulse $P = \int p dt$.

Then

$$\rho u = -\partial_x P \quad (2.57)$$

or

$$u = \partial_x \left(-\frac{P}{\rho} \right) \quad (2.58)$$

Similarly, the other two components of the velocity of the flow at any point will be defined by

$$v = \partial_y \left(-\frac{P}{\rho} \right), \quad w = \partial_z \left(-\frac{P}{\rho} \right) \quad (2.59)$$

⁴In practice it is a law of conservation of the mass. If we consider a volume element $dV = dx dy dz$, delimited by faces parallel to the coordinate axes, the principle of conservation of the mass expresses that the net flux of mass through any faces of the cube in the interval dt is equal to the variation of the mass inside the cube. The equation for this variation is $\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0$; if $\rho = \text{const}$ the equation becomes $\nabla \cdot \mathbf{v} = 0$

Defining the *potential flow*

$$\phi = -\frac{P}{q} \quad (2.60)$$

we have

$$u = \partial_x(\phi), \quad v = \partial_y(\phi) \quad w = \partial_z(\phi) \quad (2.61)$$

from which follows

$$\phi = \int (u dx + v dy + w dz) \quad (2.62)$$

A second differentiaion of (2.61) gives

$$\partial_y u = \partial_{xy} \phi = \partial_{yx} \phi = \partial_x v \quad (2.63)$$

The substitution of (2.61) into the continuity equation gives

$$\nabla^2 \phi = 0 \quad (2.64)$$

The last equation (Laplace equation) assures that the sum of any solutions is a solution again. This is equivalent to the superposition of flows: sum of the potential, of the impulsive pressures, or of the velocity components of several potential flows give a potential flow again. This refers originally to the case in which the flow is created by one impulsive preasure from rest position. Each continuous and changing pressure can be replaced by infinitely small impulsive pressures; the resultant flow is the superposition of the flows created by each impulsive pressure. As a superposition the potential flows originates a potential flow again. Most of all aerodynamics flows are potential flows, but considering the entire hull as we'll see next.

Now we have to compute the pressure at each point of the potential flow. The acceleration of each particle is given by eq (??); the acceleration is therefore a function of the space coordinates like the pressure.

Each component fo the acceleration, say $\frac{du}{dt}$ has to be espressed by the local rate of change of the velocity component $\partial_t u$ in a certain point and by the velocity components and their local derivatives.

We have $u = u(x, y, z, t)$ so

$$du = \partial_x u dx + \partial_y u dy + \partial_z u dz + \partial_t u dt \quad (2.65)$$

and

$$\frac{du}{dt} = u \partial_x u + v \partial_y u + w \partial_z u + \partial_t u = -1/\rho \partial_x p \quad (2.66)$$

From eq.(2.63) we have

$$u \partial_x u + v \partial_x v + w \partial_x w + \partial_t u = -1/\rho \partial_x p \quad (2.67)$$

and

$$\partial_t(\partial_x \phi) + 1/2(u^2 + v^2 + w^2) = -1/\rho \partial_x p \quad (2.68)$$

which integrated with respect to dx gives

$$\partial_t \Phi + \frac{\rho}{2}(u^2 + v^2 + w^2) = -\frac{p}{\rho} + C \quad (2.69)$$

where C is a constant.

The equations for the two other components of the acceleration would give the same equation. Hence the pressure can be divided into two parts superimposed; the first part $\rho \partial_t \Phi$ is due to the pressure building up or changing the potential flow. It's zero if the flow is steady.

The second part, called *Bernoulli pressure*

$$-\frac{\rho}{2} V^2$$

is the pressure necessary to maintain and keep up the steady potential flow. It depends only on the velocity and density of the fluid; the greater the velocity, the smaller the pressure. The pressure acts permanently without changing the flow and hence without changing the kinetic energy. It follows therefore that the Bernoulli pressure acting on the surface of a moving body can not perform any mechanical work. Hence in the case of the straight motion of a body, the component of the resultant force parallel to the motion is zero. In general however a momentum imparted to the fluid is not parallel to the motion of the body but it possesses a lateral component. In the case of steady motion we have

$$P + \frac{1}{2}\rho V^2 = P_0 + \frac{1}{2}\rho V_0^2 \quad (2.70)$$

Different is the case if an impulsive pressure, $-\rho\Phi$, is present. The distribution of this impulsive pressure over the surface of a body in a fluid is characterized by a resultant impulsive force and moment. As further characteristic there is the mechanical work performed by the impulsive pressure during the creation of the flow, absorbed by the air and contained afterwards in the flow as kinetic energy of all particles.

The work done by an impulse is proved in mechanics to be the product of the impulse by the average of the initial and final velocities in the direction of the impulse

$$T = \mathbf{I} \cdot \mathbf{v}$$

If a solid is moving through a fluid otherwise at rest we have the average,

$$v = (v_i + v_f)/2 = v/2$$

and if the existing fluid motion is considered as having been produced from rest by impulses applied by the surface of the body, the velocity normal to any element of surface is $\partial_n\Phi$ where n is drawn from the body to the fluid. The mean value of the velocity is hence $1/2\partial_n\Phi$.

If $v_n = \partial_n\Phi$ from eq.(2.57) we obtain

$$\partial_n(\rho\Phi) = -\partial_n P$$

so the work done by the impulse is

$$dL = (-\rho\Phi) \cdot 1/2(\partial_n\Phi) dS \quad (2.71)$$

hence

$$L = -\frac{\rho}{2} \int \Phi \partial_n\Phi dS \quad (2.72)$$

When a body moves through a fluid otherwise at rest, there is a certain amount of kinetic energy of the fluid caused by the motion of the body. If the latter is moving with a velocity $\mathbf{v}$ in a definite direction, T is the kinetic energy of the fluid due to the motion of the body, and ρ is the density of the fluid, by definition

$$K\rho = \frac{T}{1/2v} \quad (2.73)$$

is the *apparent mass* of the body for the motion in that particular direction which has to be added to the body mass.

A similar concept is used if we have a rotating body, where the body has an apparent moment of inertia.

In general the moment imparted to the fluid is not parallel to the motion of the body, but it possesses a lateral component. The body in general has different apparent masses with respect to the motions in different directions, and that makes the mechanics of a body surrounded by a

perfect fluid different from that of one moving in a vacuum. We can represent the generalized added mass by a 6x6 matrix

$$\mathbf{M}_A = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \quad (2.74)$$

where we include the added mass and added inertia tensor.

2.6 Added Mass and Inertia

Any motion of the body induces a motion in the otherwise stationary fluid; the fluid must move aside and then close in behind the body in order to allow the body to make a passage through it. As a consequence, the fluid possesses kinetic energy that it would lack if the body were not in motion. The body has to impart the kinetic energy to the fluid by doing work on the fluid, i.e., the work done is equal to the change in kinetic energy. Any adequate equations of motion for the body must take into account this kinetic energy given to the fluid by the body. *This is the function of the added mass terms in the equations.* When the motion of the body is steady, the corresponding fluid motion is steady and the kinetic energy in the fluid is constant. Hence no work is done on the fluid as long as the motion of the body remains steady. It follows that if only steady motions are to be studied, the added mass terms can be omitted in the equations of motion. The concept of added mass is usually misunderstood to be a finite amount of air connected to the vehicle such that the vehicle and the fluid represent a new system with mass larger than the original system. This is not true since the vehicle motion will force the whole fluid to oscillate with different fluid particle amplitudes in phase with the forced motion of the vehicle.

Added (virtual) mass should be understood as *pressure-induced* forces and moments due to a forced harmonic motion of the body which are proportional to acceleration of the body. For an aiship (like completely submerged vehicles) we will assume that the added mass coefficients are constant and thus independent of the wave circular frequency. Moreover, any motion of the vehicle will induce a motion in the otherwise stationary fluid. In order to allow the vehicle to pass through the fluid, the fluid must move aside and then close behind the vehicle. As a consequence, the fluid passage possesses kinetics energy that it would lack if the vehicle was not in motion.

The expression for the fluid kinetic energy T_A can be writtten [Lamb, 1945] as a quadratic form of the body axis velocity vector components

$$T_A = \frac{1}{2} \boldsymbol{\nu}^T \mathbf{M}_A \boldsymbol{\nu}$$

Here $\mathbf{M}_A$ is a 6×6 added inertia matrix defined as

$$\mathbf{M}_A = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \triangleq - \begin{bmatrix} X_{\ddot{u}} & X_{\ddot{v}} & X_{\ddot{w}} & X_{\ddot{p}} & X_{\ddot{q}} & X_{\ddot{r}} \\ Y_{\ddot{u}} & Y_{\ddot{v}} & Y_{\ddot{w}} & Y_{\ddot{p}} & Y_{\ddot{q}} & Y_{\ddot{r}} \\ Z_{\ddot{u}} & Z_{\ddot{v}} & Z_{\ddot{w}} & Z_{\ddot{p}} & Z_{\ddot{q}} & Z_{\ddot{r}} \\ K_{\ddot{u}} & K_{\ddot{v}} & K_{\ddot{w}} & K_{\ddot{p}} & K_{\ddot{q}} & K_{\ddot{r}} \\ M_{\ddot{u}} & M_{\ddot{v}} & M_{\ddot{w}} & M_{\ddot{p}} & M_{\ddot{q}} & M_{\ddot{r}} \\ N_{\ddot{u}} & N_{\ddot{v}} & N_{\ddot{w}} & N_{\ddot{p}} & N_{\ddot{q}} & N_{\ddot{r}} \end{bmatrix} \quad (2.75)$$

The notation of SNAME is used in this expression; for instance the hydrodynamic added mass force Y_A along the y -axis due to an acceleration $\ddot{u}$ in the x -direction is written as

$$Y_A = Y_{\ddot{u}} \text{ where } Y_{\ddot{u}} \triangleq \frac{\partial Y}{\partial \ddot{u}} \quad (2.76)$$

For a rigid body at rest under the assumption of an ideal fluid, the added inertia matrix is definite positive [Lamb, 1945]

$$\mathbf{M}_A = \mathbf{M}_A^T > 0$$

In a real fluid the 36 elements of $\mathbf{M}_A$ may all be distinct but still $M_A > 0$. Experience has shown that the numerical values of the added mass derivatives in a real fluid are usually in good agreement with those obtained from ideal theory. Hence $\mathbf{M}_A = \mathbf{M}_A^T > 0$ is a good approximations. Consider a symmetrical added inertia matrix having 21 distinct hydrodynamic derivatives; expanding the kinetic energy yields:

$$\begin{aligned} 2T_A = & -X_{\dot{u}}u^2 - Y_{\dot{v}}v^2 - Z_{\dot{w}}w^2 - 2Y_{\dot{w}}uw - 2X_{\dot{w}}wu - 2X_{\dot{v}}uv \\ & -K_{\dot{p}}p^2 - M_{\dot{q}}q^2 - N_{\dot{r}}r^2 - 2M_{\dot{r}}qr - 2K_{\dot{r}}rp - 2K_{\dot{q}}pq \\ & -2p(X_{\dot{p}}u + Y_{\dot{p}}v + Z_{\dot{p}}w) \\ & -2p(X_{\dot{q}}u + Y_{\dot{q}}v + Z_{\dot{q}}w) \\ & -2p(X_{\dot{r}}u + Y_{\dot{r}}v + Z_{\dot{r}}w) \end{aligned} \quad (2.77)$$

To derive the added mass force and moments we use the Kirchhoff's equations in component form

$$\begin{aligned} \frac{d}{dt} \frac{\partial T_A}{\partial u} &= r \frac{\partial T_A}{\partial v} - q \frac{\partial T_A}{\partial w} - X_A \\ \frac{d}{dt} \frac{\partial T_A}{\partial v} &= p \frac{\partial T_A}{\partial w} - r \frac{\partial T_A}{\partial u} - Y_A \\ \frac{d}{dt} \frac{\partial T_A}{\partial w} &= q \frac{\partial T_A}{\partial u} - p \frac{\partial T_A}{\partial v} - Z_A \\ \frac{d}{dt} \frac{\partial T_A}{\partial p} &= w \frac{\partial T_A}{\partial v} - v \frac{\partial T_A}{\partial w} + r \frac{\partial T_A}{\partial q} - q \frac{\partial T_A}{\partial r} - K_A \\ \frac{d}{dt} \frac{\partial T_A}{\partial q} &= u \frac{\partial T_A}{\partial w} - w \frac{\partial T_A}{\partial u} + p \frac{\partial T_A}{\partial r} - r \frac{\partial T_A}{\partial p} - M_A \\ \frac{d}{dt} \frac{\partial T_A}{\partial r} &= v \frac{\partial T_A}{\partial u} - u \frac{\partial T_A}{\partial v} + q \frac{\partial T_A}{\partial p} - p \frac{\partial T_A}{\partial q} - N_A \end{aligned} \quad (2.78)$$

Substituting (2.77) into (2.78) gives the following expression for the added mass terms

$$\begin{aligned} X_A &= X_{\dot{u}}\dot{u} + X_{\dot{w}}(\dot{w} + uq) + X_{\dot{q}}\dot{q} + Z_{\dot{w}}wq + Z_{\dot{q}}q^2 \\ &\quad + X_{\dot{v}}\dot{v} + X_{\dot{p}}\dot{p} + X_{\dot{r}}\dot{r} - Y_{\dot{v}}vr - Y_{\dot{p}}rp - Y_{\dot{r}}r^2 \\ &\quad - X_{\dot{v}}ur - Y_{\dot{w}}wr \\ &\quad + Y_{\dot{w}}vq + Z_{\dot{p}}pq - (Y_{\dot{q}} - Z_{\dot{r}})qr \\ Y_A &= X_{\dot{v}}\dot{u} + Y_{\dot{w}}\dot{w} + Y_{\dot{q}}\dot{q} \\ &\quad + Y_{\dot{v}}\dot{v} + Y_{\dot{p}}\dot{p} + Y_{\dot{r}}\dot{r} + X_{\dot{v}}vp - Y_{\dot{w}}wp + X_{\dot{r}}r^2 + (X_{\dot{p}} - Z_{\dot{r}})rp - Z_{\dot{p}}p^2 \\ &\quad - X_{\dot{v}}(up - wr) + X_{\dot{u}}ur - Z_{\dot{w}}wp \\ &\quad - Z_{\dot{q}}pq + X_{\dot{q}}qr \\ Z_A &= X_{\dot{w}}(\dot{u} - wq) + Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} - X_{\dot{u}}uq - X_{\dot{q}}q^2 \\ &\quad + Y_{\dot{w}}\dot{v} + Z_{\dot{p}}\dot{p} + Z_{\dot{r}}\dot{r} + Y_{\dot{v}}vp + Y_{\dot{r}}rp + Y_{\dot{p}}p^2 \\ &\quad + X_{\dot{v}}up - Y_{\dot{w}}wp \\ &\quad - X_{\dot{v}}vq - (X_{\dot{p}} - Y_{\dot{q}})pq - X_{\dot{r}}qr \end{aligned} \quad (2.79)$$

$$\begin{aligned}
K_A &= X_{\dot{p}}\dot{u} + Z_{\dot{w}}\dot{w} + K_{\dot{q}}\dot{q} - X_{\dot{v}}uw + X_{\dot{r}}uq + Y_{\dot{w}}w^2 - (Y_{\dot{q}} - Z_{\dot{r}})wq + M_{\dot{r}}q^2 \\
&\quad + Y_{\dot{p}}\dot{v} + K_{\dot{p}}\dot{p} + K_{\dot{r}}\dot{r} + Y_{\dot{w}}v^2 - (Y_{\dot{q}} - Z_{\dot{r}})vr + Z_{\dot{p}}vp - M_{\dot{r}}r^2 - K_{\dot{q}}rp \\
&\quad + X_{\dot{w}}uv - (Y_{\dot{v}} - Z_{\dot{w}})wv - (Y_{\dot{r}} - Z_{\dot{q}})wr - Y_{\dot{p}}wp - X_{\dot{q}}ur \\
&\quad + (Y_{\dot{r}} - Z_{\dot{q}})vq - (M_{\dot{q}} - N_{\dot{r}})rq + K_{\dot{r}}pq \\
M_A &= X_{\dot{q}}(\dot{u} + wq) + Z_{\dot{q}}(\dot{w} + uq) + M_{\dot{q}}\dot{q} - X_{\dot{w}}(u^2 - w^2) - (X_{\dot{w}} - X_{\dot{u}})wu \\
&\quad + Y_{\dot{q}}\dot{v} + K_{\dot{q}}\dot{p} + M_{\dot{r}}\dot{r} + Y_{\dot{p}}vr - Y_{\dot{r}}vp - K_{\dot{r}}(p^2 - r^2) + (K_{\dot{p}} - N_{\dot{r}})rp \\
&\quad - Y_{\dot{w}}uv + X_{\dot{v}}vw - (X_{\dot{r}} + Z_{\dot{p}})(up - wr) + (X_{\dot{p}} - Z_{\dot{r}})(wp + ur) \\
&\quad - M_{\dot{r}}pq + K_{\dot{q}}qr \\
N_A &= X_{\dot{r}}\dot{u} + Z_{\dot{r}}\dot{w} + M_{\dot{r}}\dot{q} + X_{\dot{v}}u^2 + Y_{\dot{w}}wu - (X_{\dot{p}} - Y_{\dot{q}})uq - Z_{\dot{p}}wq - K_{\dot{q}}q^2 \\
&\quad + Y_{\dot{r}}\dot{u} + K_{\dot{r}}\dot{p} + N_{\dot{r}}\dot{r} - X_{\dot{v}}v^2 - X_{\dot{r}}vr - (X_{\dot{p}} - Y_{\dot{q}})vp + M_{\dot{r}}rp + K_{\dot{q}}p^2 \\
&\quad - (X_{\dot{u}} - Y_{\dot{v}})uv - X_{\dot{w}}vw + (X_{\dot{q}} + Y_{\dot{p}})up + Y_{\dot{r}}ur + X_{\dot{q}}wp \\
&\quad - (X_{\dot{q}} + Y_{\dot{p}})vq - (K_{\dot{p}} - M_{\dot{q}})pq - K_{\dot{r}}qr
\end{aligned} \tag{2.80}$$

Equations (2.79) are in six parts. Each part is arranged with longitudinal components of the motion on the first line and lateral components on the second line. The third line contains mixed terms involving the u or w velocity as one factor. Often one or both of these velocities are large enough to be treated as constants during the motion, thus permitting the affected terms on the third line to be treated as additional terms in the lateral components of motion. The fourth line contains mixed components of motion of such a nature that they usually can be neglected as second order terms. Many of the 21 added mass derivatives contained in the general expressions for added mass are either zero or mutually related when the body has various symmetries. Bodies of practical interest almost invariably have a vertical plane of symmetry. Bodies used as underwater vehicles or airships usually are even more symmetrical and a finned prolate spheroid is a good approximation of those bodies.

It's always possible to choose the coordinate axes along the directions of permanent translations [Lamb, 1945] so that

$$Y_{\dot{w}} = X_{\dot{w}} = X_{\dot{v}} = 0$$

Assume a prolate spheroid with the origin of a Cartesian set of body axes located at any arbitrary point on the axis of revolution, which is also the x -axis. Mutually perpendicular y - and z -axes will also be perpendicular to the x -axis. Assume that the spheroid is fitted with a dorsal fin near the maximum diameter of the body and lying in the xz -plane. In addition, assume that the body has four fins that are identical and are located at the rear of the body. Let these four fins each have the same x -coordinate and be placed in cruciform fashion around the x -axis. Furthermore, let the plane of each fin pass through the x -axis. This finned body can provide a rather good approximation to a submarine or airship, as far as the theoretical values for added mass is concerned, when the spheroid has the same fineness ratio and displacement as the actual body.

It's evident that this body has one plane of symmetry, the XZ -plane. The direction normal to this plane is a direction of *permanent traslation* [Lamb, 1945] so the energy does not change if w change in sign; it is also evident from stream lines around the body that the energy of the fluid does not change if p, r change in sign. So from (2.77) we have for example

$$M_{\dot{r}}qr = M_{\dot{r}}q(-r) \Leftrightarrow M_{\dot{r}} = 0$$

following the same considerations we have

$$K_{\dot{q}} = X_{\dot{p}} = M_{\dot{p}} = Y_{\dot{q}} = X_{\dot{r}} = Z_{\dot{r}} = 0$$

The energy becomes

$$\begin{aligned}
 2T_A = & -X_{\dot{u}}u^2 - Y_{\dot{v}}v^2 - Z_{\dot{w}}w^2 \\
 & -K_{\dot{p}}p^2 - M_{\dot{q}}q^2 - N_{\dot{r}}r^2 \\
 & -2K_{\dot{r}}rp - 2pY_{\dot{p}}v - 2p(X_{\dot{q}}u + Z_{\dot{q}}w) - 2pY_{\dot{r}}v
 \end{aligned} \tag{2.81}$$

Equations (2.79) become more simpler

$$\begin{aligned}
 X_A &= X_{\dot{u}}\dot{u} + X_{\dot{q}}\dot{q} + Z_{\dot{w}}wq + Z_{\dot{q}}q^2 - Y_{\dot{v}}vr - Y_{\dot{p}}rp - Y_{\dot{r}}r^2 \\
 Y_A &= Y_{\dot{v}}\dot{v} + Y_{\dot{p}}\dot{p} + Y_{\dot{r}}\dot{r} + X_{\dot{u}}ur - Z_{\dot{w}}wp - Z_{\dot{q}}pq + X_{\dot{q}}qr \\
 Z_A &= Z_{\dot{q}}\dot{q} - X_{\dot{u}}uq - X_{\dot{q}}q^2 + Y_{\dot{v}}vp + Y_{\dot{r}}rp + Y_{\dot{p}}p^2 \\
 K_A &= Y_{\dot{p}}\dot{v} + K_{\dot{p}}\dot{p} + K_{\dot{r}}\dot{r} - (Y_{\dot{v}} - Z_{\dot{w}})vw - (Y_{\dot{r}} - Z_{\dot{q}})wr - Y_{\dot{p}}wp - X_{\dot{q}}ur \\
 &\quad + (Y_{\dot{r}} - Z_{\dot{q}})vq - (M_{\dot{q}} - N_{\dot{r}})rq + K_{\dot{r}}pq \\
 M_A &= X_{\dot{q}}(\dot{u} + wq) + Z_{\dot{q}}(\dot{w} + uq) + M_{\dot{q}}\dot{q} - X_{\dot{u}}wu + Y_{\dot{p}}vr - Y_{\dot{r}}vp - K_{\dot{r}}(p^2 - r^2) + (K_{\dot{p}} - N_{\dot{r}})rp \\
 N_A &= Y_{\dot{r}}\dot{u} + K_{\dot{r}}\dot{p} + N_{\dot{r}}\dot{r} - (X_{\dot{u}} - Y_{\dot{v}})uv + (X_{\dot{q}} + Y_{\dot{p}})up + Y_{\dot{r}}ur + X_{\dot{q}}wp \\
 &\quad - (X_{\dot{q}} + Y_{\dot{p}})vq - (K_{\dot{p}} - M_{\dot{q}})pq - K_{\dot{r}}qr
 \end{aligned} \tag{2.82}$$

Removing the dorsal fin we have a body with two plane of simmetry the XZ -plane and the the XY -plane. The simmetry with respect to the latter introduces the invariance of the energy for a change in sign of w, p, q and so

$$X_{\dot{q}} = Y_{\dot{p}} = K_{\dot{r}} = 0$$

Furthermore the resulting body is a body fo revolution around x with symmetric fins we have

$$Y_{\dot{v}} = Z_{\dot{w}} \quad M_{\dot{q}} = N_{\dot{r}}$$

and applying a rotation of $\pi/2$ around the axis of simmetry, the energy does not change. The rotations applied to linear and angular velocities gives

$$[u, v, w] \rightarrow [u, w, -v]; \quad [p, q, r] \rightarrow [p, r, -q]$$

so the energy is invariant when we write $w, -v, r, q$ in order to v, w, q, r , this yelds

$$Z_{\dot{q}} = -Y_{\dot{r}}$$

The resulting expression for the energy is

$$\begin{aligned}
 2T_A = & -X_{\dot{u}}u^2 - Y_{\dot{v}}v^2 - Y_{\dot{v}}w^2 \\
 & -K_{\dot{p}}p^2 - M_{\dot{q}}q^2 - M_{\dot{q}}r^2 \\
 & -2pZ_{\dot{q}}w + 2pZ_{\dot{q}}v
 \end{aligned} \tag{2.83}$$

and the component equations for added mass force and moment are

$$\begin{aligned}
 X_A &= X_{\dot{u}}\dot{u} + Y_{\dot{v}}(rv - qw) - Y_{\dot{r}}(q^2 + r^2) \\
 Y_A &= X_{\dot{u}}ru + Y_{\dot{v}}(\dot{v} - pw) + Y_{\dot{r}}(\dot{r} + pq) \\
 Z_A &= -X_{\dot{u}}qu + Y_{\dot{v}}(\dot{w} + pv) - Y_{\dot{r}}(\dot{q} - pr) \\
 K_A &= K_{\dot{p}}\dot{p} \\
 M_A &= (X_{\dot{u}} - Y_{\dot{v}})uw - Y_{\dot{r}}(\dot{w} + pv - qu) + N_{\dot{r}}\dot{q} - (N_{\dot{r}} - K_{\dot{p}})pr \\
 N_A &= -(X_{\dot{u}} - Y_{\dot{v}})vu + Y_{\dot{r}}(\dot{v} + ru - pw) + N_{\dot{r}}\dot{r} + (N_{\dot{r}} - K_{\dot{p}})pq
 \end{aligned} \tag{2.84}$$

The added mass matrix is

$$M_A = - \begin{bmatrix} X_{\ddot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\ddot{v}} & 0 & 0 & 0 & -Z_{\dot{q}} \\ 0 & 0 & Z_{\ddot{w}} & 0 & Z_{\dot{q}} & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & Z_{\dot{q}} & 0 & M_{\dot{q}} & 0 \\ 0 & -Z_{\dot{q}} & 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \quad (2.85)$$

Normally $K_{\dot{p}}$ would be negligible compared to $N_{\dot{r}}$. If there were no fins, $K_{\dot{p}}$ and $Z_{\dot{q}}$ would be zero⁵ and with the usual sizes of tail fin, can be approximated safely to zero.

It is important to remark another simplification: the added mass effect does not depend on the acceleration, per se, but on the nature of the body motion, so the added mass effect may differ under two sets of circumstances where the acceleration is the same. For example, a prolate spheroid, moving in a perfect fluid in the direction of the x-axis, and experiencing an acceleration $\ddot{u}$, would have streamlines different from ones of the same spheroid making a steady turn at a fixed angle of drift (e.g due to Coriolis Forces). The body would react in the same way to either longitudinal and Coriolis Force. So the added mass effect would definitely be different in the two cases, however, because of the difference in the flow patterns attending the two different motions of the body. Recalling that a given flow pattern requires a unique velocity potential to produce it, the kinetic energy is derivable from the velocity potential and that the added mass effects are the result of changes in the kinetic energy of the fluid; it is reasonable to expect different added mass effects from the two different flow patterns.

2.6.0.1 Added Mass for Prolate Ellipsoid

A simple approach to obtain the generalized added mass matrix is to approximate the hull as an ellipsoid of revolution (prolate) with $a > b = c$ where a is the major semiaxis along the centerline of the ship.

If the body frame is located at the centre of volume (CV) of the vehicle⁶ with the centerline along the x axis and positively toward the nose and the z axis positively upward, then all the off diagonal terms of (2.74) are set to zero as previously seen.

The calculation of the six coefficients of the generalized mass matrix is given by [Lamb, 1945]

$$\begin{aligned} X_{\ddot{u}} &= -4/3\pi\rho_A ab^2 k_1 \\ Y_{\ddot{v}} = Z_{\ddot{w}} &= -4/3\pi\rho_A ab^2 k_2 \end{aligned} \quad (2.86)$$

where

$$k_1 = \frac{\alpha_0}{2 - \alpha_0}$$

$$k_2 = \frac{\beta_0}{2 - \beta_0} \quad (2.87)$$

$$(2.88)$$

⁵This is the case of a vehicle with three plane of symmetry, where the added mass matrix is diagonal. The diagonal structure is highly attractive since off diagonal elements are difficult to determine from experiments as well as theory. In practice, the diagonal approximation is found to be quite good for many applications. This is due to the fact that the off-diagonal elements will be much smaller than their diagonal counterparts.

⁶We also set, in the derivation of the equations of the motion, the centre of volume coincident with the centre of buoyancy

and ρ_A is the density of the fluid. For a prolate spheroid ($a > b = c$) we have

$$\alpha_0 = \frac{2(1-e^2)}{e^3} \left(\frac{1}{2} \ln \frac{1+e}{1-e} - e \right) \quad (2.89)$$

$$\beta_0 = \gamma_0 = \frac{1}{e^2} - \frac{1-e^2}{2e^3} \ln \frac{1+e}{1-e} \quad (2.90)$$

$$(2.91)$$

and the additional moment of inertia of the rotational potential flow are given by

$$K_{\dot{\mathbf{r}}} = 0$$

$$N_{\dot{\mathbf{r}}} = M_{\dot{\mathbf{q}}} = -4/3\pi\rho_A ab^2 \frac{b^2 + a^2}{5} k' \quad (2.92)$$

$$(2.93)$$

where

$$k' = \frac{e^4(\beta_0 - \alpha_0)}{(2-e^2)[2e^2 - (2-e^2)(\beta_0 - \alpha_0)]} \quad (2.94)$$

$$(2.95)$$

2.6.1 Coriolis and Centripetal Matrix

For a rigid body moving through an ideal fluid the hydrodynamic Coriolis and Centripetal Matrix $\mathbf{C}_A(\boldsymbol{\nu})$ can always be parameterized such that

$$\mathbf{C}_A(\boldsymbol{\nu}) = -\mathbf{C}_A(\boldsymbol{\nu})^\top \quad (2.96)$$

by defining

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & -\mathbf{S}(\mathbf{A}_{11}\boldsymbol{\nu}_1 + \mathbf{A}_{12}\boldsymbol{\nu}_2) \\ -\mathbf{S}(\mathbf{A}_{11}\boldsymbol{\nu}_1 + \mathbf{A}_{12}\boldsymbol{\nu}_2) & -\mathbf{S}(\mathbf{A}_{21}\boldsymbol{\nu}_1 + \mathbf{A}_{22}\boldsymbol{\nu}_2) \end{bmatrix} \quad (2.97)$$

This results could easily be proved by 2.4.2 substituting

$$\mathbf{M}_A = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

2.6.2 Added Mass of the fins

The added mass and moment of inertia of the fins can be computed by integrating the 2-D added mass of the cross section over the fin region (Strip theory). In our case fins are placed in a cruciform geometry as shown in figure and the 2-D transverse added mass in the y and z direction can be computed using potential flow theory [Nielsen, 1988]. The contribution of the fins to these 2-D added mass terms (surface fins) can be written as

$$m_{22}^{SF} = m_{33}^{SF} = \rho\pi(b - R^2/b)^2, \quad m_{44}^{SF} = 2k_{44}\rho b^4/\pi \quad (2.98)$$

where R is the body cross-sectional radius of the hull where fins are installed and b is the fins semispan. The factor k_{44} depends on $(2b_e)^2/S_f$ ratio (??). The non zero elements in the added mass matrix of the fins $\mathbf{M}_{AF}$ are obtained by integrating (2.98) [Fossen, 1994].

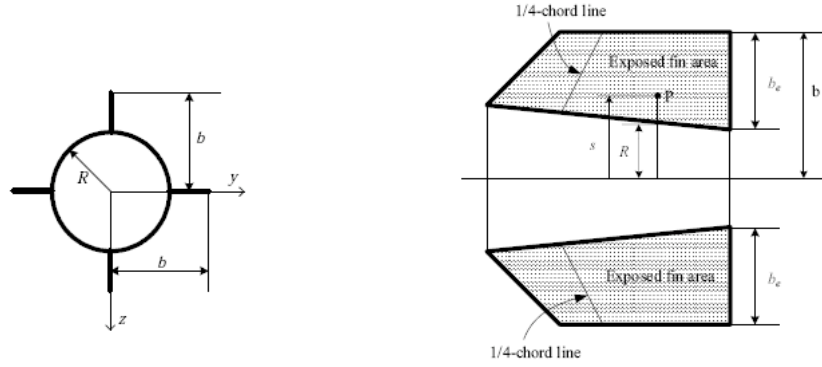
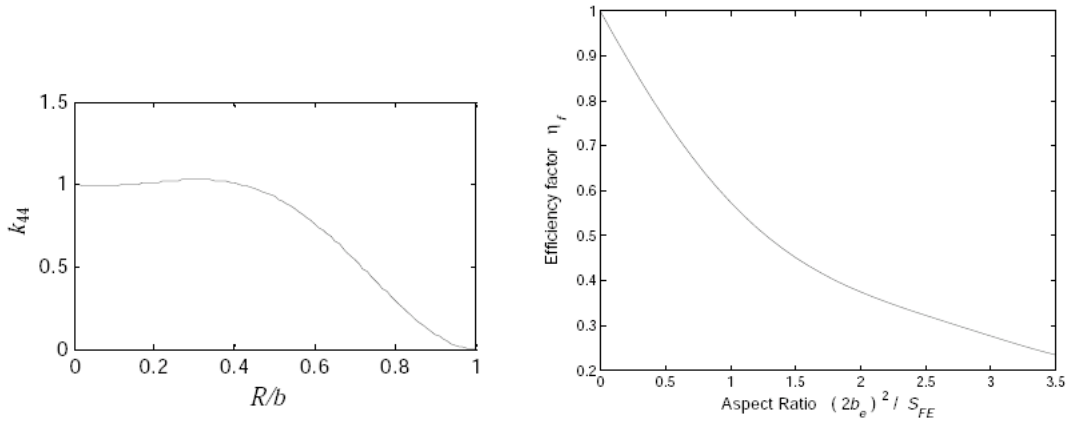


Figure 2.3: Cruciform fins configuration and geometry

Figure 2.4: Body-fins factor k_{44} and efficiency factor η_f

For symmetry of the finned body the total added mass has the form (2.85) and so the added mass of the fins is

$$\mathbf{M}_{AF} = - \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & -Z_{\dot{q}} \\ 0 & 0 & Z_{\dot{w}} & 0 & Z_{\dot{q}} & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & Z_{\dot{q}} & 0 & M_{\dot{q}} & 0 \\ 0 & -Z_{\dot{q}} & 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \quad (2.99)$$

The coefficients are determined by

$$\begin{aligned} Y_{\dot{v}} &= Z_{\dot{w}} = -\eta_p \int_{x_0}^{x_1} m_{22}^{SF} dx, & Z_{\dot{q}} &= -\eta_p \int_{x_0}^{x_1} m_{22}^{SF} x dx \\ K_{\dot{p}} &= -\eta_p \int_{x_0}^{x_1} m_{44}^{SF} dx & M_{\dot{q}} &= N_{\dot{r}} = -\eta_p \int_{x_0}^{x_1} m_{22}^{SF} x^2 dx \end{aligned} \quad (2.100)$$

where x_0, x_1 are the coordinates of the starting and ending positions of the fins and η_p is the fin's efficiency factor (??).

2.7 Hydrodynamic Forces acting on an Airship Hull

A determination of pressure distribution in a wind tunnel is usually tedious, complicate and expensive; calculations of pressure coefficients by CFD software could not be applied in a dynamical airship simulator easily. The aerodynamic forces on an airship hull have been invetigated by different methods based on hydrodynamic theory of flow. Enough has been done experimentally to show that the normal pressure is substantially unaffected by viscosity and skin friction except near the stern, especially for small airship with low velocity.

In this section we present the formulation for computing the hydrodynamic forces for both straight and curvilinear flight. Integrating numerically the equations for the added mass we have the same results but if we want to calculate the viscous contribution we need explicit integral formulas to remove the potential contribution againsts the viscous one.

For curvilinear flight, the most general case, the preassure at any point can be expressed as

$$\begin{aligned} \frac{p}{q} = & \left(\cos \alpha - \frac{r}{R} \sin \phi \right)^2 + \left(\frac{x}{R} + \sin \alpha \right)^2 \\ & - \left\{ A \cos \alpha \cos \beta + B \sin \alpha \sin \beta \sin \phi + [2S(C-1) - CS_{max}] \frac{\cos \beta \sin \phi}{\pi r R} \right\}^2 \end{aligned} \quad (2.101)$$

$$- \left(B \sin \alpha + \frac{x C}{R} \right)^2 \cos^2 \phi \quad (2.102)$$

where we introduce the symbols

P = any point of the surface

P_0 = point of highest pressure (and zero velocity)

p = pressure at generic point P of the surface of the hull

q_0 = pressure at point P_0

$A = 1 + k_1$ (where k_1 is the Lamb Coefficient)

$B = 1 + k_2$ (where k_2 is the Lamb Coefficient)

$$C = k' \frac{n^2 + 1}{n^2 - 1}$$

n = ellipsoid fineness ratio a/b

$$\alpha = \tan^{-1} \left(\sqrt{v^2 + w^2}/u \right) \text{ where velocity is computed in the C.V}$$

R = trajectory curvature radius

x, y = profile coordinate of the point P in the body frame

S = cross sectional area of ellipsoid at point P

S_{max} = maximum cross sectional area of ellipsoid at point

$$\frac{dS}{dx} = \text{rate of change of cross section } 2\pi r \tan \beta$$

(vol) = Airhip volume

Consider an element on the surface of an ellipsoid of revolution. The force due to aerodynamic pressure on a small arc

$$dF = pr d\phi \quad (2.103)$$

This force can be subdivided into three compoments

a) Transverse (or normal) to the centerline

$$dF_T = pr \sin \phi d\phi \quad (2.104)$$

b) Longitudinal

$$dF_L = pr \tan \beta d\phi \quad (2.105)$$

c) Perpendicular to the plane of symmetry; this component is evidently balanced by the component on the other side of the hull and is here disregarded

The resultant transverse force per unit of length of any section perpendicular to the axis, is in general

$$\begin{aligned} \frac{dQ}{dx} &= \Delta F_T = 2 \int_{-\pi/2}^{\pi/2} pr \sin \phi d\phi \\ &= q \left\{ AB \frac{dS}{dx} \sin \alpha - \frac{2}{R} [(2AC - 2A + \sec^2 \beta^2 \alpha) S - ACS_{max}] \right\} \cos^2 \beta \cos \alpha \end{aligned} \quad (2.106)$$

Now if we parametrize the ellipse with the equation

$$y = \pm b \left(1 - \frac{x^2}{a^2} \right)^{1/2} \quad (2.107)$$

and the cross section al ares is $S = \pi y^2$, we have that

$$\frac{dS}{dx} = -4\pi \frac{b^2}{a^2} x \quad (2.108)$$

and

$$\cos^2 \beta = \left[1 - 4 \frac{b^2 x^2}{a^2 (a^2 - x^2)} \right]^{-1} = \frac{1}{\sec^2 \beta} \quad (2.109)$$

We can express the transversal force with the formula

$$\begin{aligned} F_T &= q_0 \frac{AB}{2} \sin 2\alpha \int_{-a}^a 4\pi \frac{b^2}{a^2} x \left[1 - 4 \frac{b^2 x^2}{a^2 (a^2 - x^2)} \right]^{-1} \\ &\quad - \frac{2}{R} \cos \alpha \int_{-a}^a \left[(2AC - 2A + \sec^2 \beta) \pi b^2 \left(1 - \frac{x^2}{a^2} \right) - AC \pi b^2 \right] \cos \beta dx \end{aligned} \quad (2.110)$$

The general expression of longitudinal force is, for unit of lenght

$$\begin{aligned} \Delta F_L &= 2 \int_{-\pi/2}^{\pi/2} pr \tan \alpha d\phi \\ &= q_0 \left[2 \cos^2 \alpha + 2 \left(\frac{x}{R} + \sin \alpha \right)^2 + \left(\frac{r}{R} \right)^2 - 2 (A \cos \alpha \cos \beta)^2 - \left(B \sin \alpha + \frac{x}{R} C \right)^2 \right] \pi r \tan \beta \\ &\quad - \left[B \sin \alpha \sin \beta + (2CS - CS_{max} - 2S) \frac{\cos \beta}{\pi r R} \right]^2 \pi r \tan \beta \end{aligned} \quad (2.111)$$

The longitudinal force per unit of length if integrated over the length of the ellipsoid will give the total longitudinal aerodynamic force, and it will be equal to [R.H. Upson and W.A. Klikoff, 1931]

$$F_L = 2q_0k_2 \left(\frac{vol}{R} \right) \sin \alpha \quad (2.112)$$

so there's no resultant longitudinal force during pitched flight in an ideal fluid. By integrating the shear curve, Q over the total length we can obtain the aerodynamic moment due to transverse force around the center of volume [R.H. Upson and W.A. Klikoff, 1931]

$$M_T = q_0(k_2 - k_1)(vol) \frac{n^2}{n^2 - 1} \sin \alpha \quad (2.113)$$

The longitudinal moment will always have a sign opposite respect to the lateral moment and has the expression

$$M_L = q_0(k_2 - k_1) \frac{(vol)}{n^2 - 1} \sin 2\alpha \quad (2.114)$$

Combining the two moments we have the total moment around the center of volume

$$M = q_0(k_2 - k_1)(vol) \sin 2\alpha \quad (2.115)$$

which is the *Munk moment*. The fundamental difference between the circular and pitched flight in an ideal fluid is that in the case of pitched flight (and also for $\theta = 0$) there's no resultant transverse or longitudinal forces but we can have a resultant moment.

It's important to point out how the two representation of hydrodynamic pressures, one given by dynamics of added mass and the one given by integrated pressure forces, produce the same results. We can get the aerodynamical forces by

$$\tau_A = -M_A \dot{\nu} - C_A(\nu) \nu \quad (2.116)$$

where M_A is the generalized added mass matrix and $C_A(\nu)$ is the Coriolis matrix. So in explicit form we have from 2.97

$$\tau_A = - \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \dot{\nu}_1 \\ \dot{\nu}_2 \end{bmatrix} - \begin{bmatrix} \nu_2 \times (\mathbf{A}_{11}\nu_1 + \mathbf{A}_{12}\nu_2) \\ \nu_1 \times (\mathbf{A}_{11}\nu_1 + \mathbf{A}_{12}\nu_2) + \nu_2 \times (\mathbf{A}_{21}\nu_1 + \mathbf{A}_{22}\nu_2) \end{bmatrix} \quad (2.117)$$

When the airship is in steady translation we have only the term $-\nu_1 \times (\mathbf{A}_{11}\nu_1)$ which is really the *Munk moment*.

Now it's useful to derive an expression to compute the curvature radius R to use in (2.110).

If the airship is in steady turn the forces acting in the hull are

- a) transverse component of centrifugal force due to the mass of the airship itself, distributed through the hull, with resultant acting at center of volume. Transverse velocity v is made up of two terms, $V_0 \sin \alpha$ due to translation, plus the term due to rotation where x is measured along the axis from the aerodynamic center,

$$\frac{dx}{dt} = \omega \times x = \frac{V_0}{R} \hat{j}$$

and so

$$\frac{dv}{dt} = \frac{dv}{dx} \frac{dx}{dt} = \frac{V_0^2}{r} \cos \alpha \quad (2.118)$$

The centrifugal force is

$$F_c = \frac{mV_0^2}{R} \cos \alpha \quad (2.119)$$

- b) transverse component to the centrifugal force due to added mass of air set in motion around the hull which appears as aerodynamic pressure

$$F_{c,a} = \frac{mV_0^2}{R} k_1 \cos \alpha \quad (2.120)$$

- c) distributed aerodynamic force which does not produce any resultant force but produces an unballancing moment (*Munk moment*)
- d) a stern force acting at distance l_h from the center of gravity whose magnitude ballances the *Munk moment*

$$F_s = \frac{mV_0^2}{2l} (k_2 - k_1) (vol) \sin 2\alpha \quad (2.121)$$

Now all these forces should be ballanced so,

$$\frac{mV_0^2}{R} (1 + k_1) \cos \alpha = \frac{mV_0^2}{2l} (k_2 - k_1) (vol) \sin 2\alpha \quad (2.122)$$

thus,

$$\frac{(1 + k_1)}{R} = \frac{(k_2 - k_1)}{l} \sin \alpha \quad (2.123)$$

$$R = \frac{Al}{(B - A) \sin \alpha} \quad (2.124)$$

The parameter l is the "fin arm" and could be taken as the distance from the center of gravity of the ship to the center of pressure of the fin l_h . A more precise value of l could be determined by wind tunnel experiments.

2.8 Viscous Forces

2.8.1 Viscous Lateral Force

Wind-tunnel tests on the aerodynamics of bodies of revolution at different angles of attack have shown that a prediction based on potential flow assumption can cause considerable errors because of the effects of viscosity, especially at the rear of the body. A study of the low speed pressure distribution of inclined bodies with greatly different nose countours [Edward J. Hopkins, 1951] showed that transverse forces acting on the expanding portions of these bodies agree well with those predicted by potential theory. However, for the contracting portions of these bodies where the effects of the viscosity become more important, the transverse forces predicted by potential theory does not agree with experiments. In [Edward J. Hopkins, 1951] it's presented a semi-empirical estimation approach for the calculation of aerodynamics of bodies of revolution. In his work experimental data for 15 body of revolution where analized with an angle of attack in a range from 0° to 20° to determine the portion of the body for which potential theory should be employed to attain optimum agreement between the calculated and experimental data.

They found that exists a longitudinal distance ϵ_0 from the nose at which potential theory doesn't match experimental data. The point at ϵ_0 could be correlated with the longitudinal position on the body at which the rate of change of cross-sectional area has a maximum negative value with longitudinal distance

$$\epsilon_0 = 0.378L + 0.527\epsilon_1 \quad (2.125)$$

where ϵ_1 denotes the position at which $\frac{dS}{d\epsilon}$ has a maximum negative value. For us it is more convenient to express ϵ_0 in the body frame with the origin in the CV, so we have

$$\xi_0 = a - \epsilon_0 \quad (2.126)$$

The normal force (or transverse) to the centerline can be computed now considering non-potential effects as

$$F_T = \int_{\xi_0}^a \Delta F_T dx + q_0 \eta_h (C d_c)_h \sin^2 \gamma \int_{-a}^{\xi_0} 2r dx \quad (2.127)$$

where $(C d_c)$ is the crossflow drag coefficient of an infinite-length circular cylinder [J.H. Allen and E.W. Perkins, 1952], η is an efficiency factor accounting for the finite length of the body and determined from the fineness ratio of the body, r is the local cross-sectional radius. The corresponding moment is

$$M_T = \int_{\xi_0}^a \Delta F_T x dx + q_0 \eta_h (C d_c)_h \sin^2 \gamma \int_{-a}^{\xi_0} 2r x dx \quad (2.128)$$

The dynamic pressure $q_0 = 1/2 \rho v^2$ and the angle γ between the centerline and the velocity vector is computed from the local velocity $[u, v, w]^T$ at the position ξ_0 that is,

$$\gamma = \tan^{-1} \left(\sqrt{v^2 + w^2} / u \right) \quad (2.129)$$

From this results it's necessary to compute the viscous effects to remove the potential flow contribution downstream of ξ_0 and replace it with a viscous flow contribution. So the viscous contribution acting on the hull, normal to the center line is

$$F_{N_h} = - \int_{-a}^{\xi_0} \Delta F_T dx + q_0 \eta_h (C d_c) \sin^2 \gamma \int_{-a}^{\xi_0} 2r dx \quad (2.130)$$

where $C d_c$ is the cross flow drag coefficient of an infinite-length circular cilinder and η_h is an efficiency factor accounting for a finite length of the body and determined from the fineness ratio of the body. The components of the normal viscous force are

$$(F_{N_h})_y = -F_{N_h} \sin \theta = -F_{N_h} \frac{v}{\sqrt{v^2 + w^2}} \quad (2.131)$$

$$(F_{N_h})_z = -F_{N_h} \cos \theta = -F_{N_h} \frac{w}{\sqrt{v^2 + w^2}} \quad (2.132)$$

The corresponding moment about the origin of the body frame due to normal force is

$$M_{N_h} = - \int_{-a}^{\xi_0} \Delta F_T x dx + q_0 \eta_h (C d_c) \sin^2 \gamma \int_{-a}^{\xi_0} 2r x dx \quad (2.133)$$

The components of the moment vector are (fig. 2.5) (note that $M_{N_h} < 0$)

$$(M_{N_h})_y = M_{N_h} \cos(\pi/2 - \theta) = M_{N_h} \sin \theta = M_{N_h} \frac{w}{\sqrt{v^2 + w^2}} < 0 \quad (2.134)$$

$$(M_{N_h})_z = -M_{N_h} \sin(\pi/2 - \theta) = -M_{N_h} \cos \theta = -M_{N_h} \frac{v}{\sqrt{v^2 + w^2}} > 0 \quad (2.135)$$

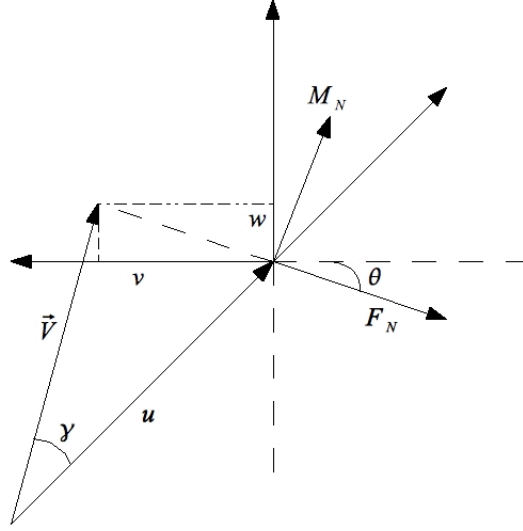


Figure 2.5: Normal force and moment decomposition

Normal force and moments due to the fins [S.P. Jones and J.D De Lournier, 1981] is given respectively by

$$F_{N_f} = q_0 S_f \left[\frac{1}{2} (Cn_\alpha^*)_f \eta_f \sin(2\gamma) + (C_{dc})_f \sin \gamma \sin |\gamma| \right] \quad (2.136)$$

$$M_{N_f} = q_0 S_f \left[\frac{1}{2} (Cn_\alpha^*)_f \eta_f (l_f)_1 \sin(2\gamma) + (C_{dc})_f S_f (l_f)_2 \sin \gamma \sin |\gamma| \right] \quad (2.137)$$

where

$(Cn_\alpha^*)_f$ = derivative of the isolated fins normal force coefficients with respect to α at $\alpha = 0$ and referenced to S_f

η_f , fin efficiency factor accounting for the effect of the hull on the fins

$(C_{dc})_f$, fins cross-flow drag coefficient. We use the drag coefficient of a thin bidimensional plate

$(l_f)_1$, distance from the origin to the aerodynamic center of fins

$(l_f)_2$, distance from the origin to the geometric center of fins

The vector decomposition for the normal force is

$$(F_{N_f})_y = -F_{N_f} \cos \theta = -F_{N_f} \frac{v}{\sqrt{v^2 + w^2}} \quad (2.138)$$

$$(F_{N_f})_z = -F_{N_f} \sin \theta = -F_{N_f} \frac{w}{\sqrt{v^2 + w^2}} \quad (2.139)$$

$$(2.140)$$

and for the moment

$$(M_{N_f})_y = -M_{N_f} \sin(\theta) = -M_{N_f} \frac{w}{\sqrt{v^2 + w^2}} \quad (2.141)$$

$$(M_{N_f})_z = M_{N_f} \cos(\theta) = M_{N_f} \frac{v}{\sqrt{v^2 + w^2}} \quad (2.142)$$

$$(2.143)$$

2.8.2 Axial Drag

Principale sources of drag on streamlined bodies are

- a) **form drag**: the drag on the body resulting from the integrated effect of the static pressure acting normal to its surface resolved in the drag direction
- b) **skin friction drag**: the drag resulting from viscous shearing stresses over its wetted surface.

For the axial drag, Hoerner [G.A. Khoury and J.D. Gillett, 1999] analysed experimental data and compared them with theory to produce empirical formulae which may be used to asses the drag of bodies. For most airships the flow over the hull is turbulent and for these conditions it was deduced that

$$(C_{d_h})_0/C_s = 4(l/d)^{1/3} + 6(d/l)^{1.2} + 24(d/l)^{2.7} \quad (2.144)$$

Hoerner suggests that C_s (skin drag coefficients) for practical levels of surface and for $Re > 5 \cdot 10^6$ varies as

$$C_s = 0.043 \cdot Re^{-1/6} \quad (2.145)$$

where Re is the Reynolds number.

Combining equations (2.144) and (2.145) gives the zero-angle axial drag coefficient

$$(C_{d_h})_0 = [0.172(l/d)^{1/3} + 0.252(d/l)^{1.2} + 1.032(d/l)^{2.7}] \cdot Re^{-1/6} \quad (2.146)$$

So the total axial force induced by fins and hull is given by

$$F_D = q_0[(C_{d_h})_0 S_h + (C_{d_f})_0 S_f] \cos^2 \alpha \quad (2.147)$$

where

$$\alpha = \tan^{-1} \left(\sqrt{v^2 + w^2}/u \right) \text{ angle of attack computed in the C.V}$$

$$S_h = (V)^{2/3}, \text{ hull cross reference area}$$

$$S_f \simeq S_h/3, \text{ fins reference area}$$

$$(C_{d_f})_0, \text{ fin zero-angle axial drag coefficient (adimensional)}$$

$$(C_{d_h})_0, \text{ hull zero-angle axial drag coefficient (adimensional)}$$

We can arrange all terms to form the vector of viscous forces and moments (to put in the right side of the airship equation of motion)

$$\tau_A = \begin{bmatrix} -F_D \\ -(F_{N_f} + F_{N_h}) \cos \theta \\ -(F_{N_f} + F_{N_h}) \sin \theta \\ 0 \\ (M_{N_f} - M_{N_h}) \sin \theta \\ (-M_{N_f} - M_{N_h}) \cos \theta \end{bmatrix} \quad (2.148)$$

2.9 Force and Moments Due to Control Surface Deflection

To enable pitch control it is useful to implement in the vehicle's model the action of the lateral rudders. Force and moment applied by a couple of lateral (pitch) rudders could be expressed as

$$F_z(\delta_\theta) = -q_0 \frac{\partial C_L}{\partial \delta} S_f \eta_f \delta_\theta \quad (2.149)$$

$$M_y(\delta_\theta) = -q_0 \frac{\partial C_L}{\partial \delta} S_f \eta_f (l_f)_1 \delta_\theta \quad (2.150)$$

where δ is the deflection angle of the pitch-rudder and θ is the effect on the pitch angle. Similar expressions are obtained for yaw rudders

$$F_y(\delta_\phi) = -q_0 \frac{\partial C_L}{\partial \delta} S_f \eta_f \delta_\phi \quad (2.151)$$

$$M_z(\delta_\phi) = q_0 \frac{\partial C_L}{\partial \delta} S_f \eta_f (l_f)_1 \delta_\phi \quad (2.152)$$

where δ is the deflection angle of the yaw-rudders, ϕ is the effect on the yaw angle, and $\frac{\partial C_L}{\partial \delta}$ is the derivative of the fin lift coefficient with respect to the angle of attack at zero incidence [J.B. Mueller and M.A Paluszek,]; it can be obtained by experiments or from CFD results.

2.9.1 Restoring Forces and Moments

In the hydrodynamic terminology the gravitational and buoyant forces are called restoring forces. The gravitational force $\mathbf{f}_g$ will act in the center of gravity $\mathbf{r}_g$ of the vehicle. Similarly the buoyant force $\mathbf{f}_b$ will act through the center of buoyancy, considered coincident to the origin of the body frame. The weight of the airship is $P = mg$, while the buoyancy force is defined as $B = \rho g V$, where V is the volume of the fluid displaced by the vehicle. In the body frame we have

$${}^b \mathbf{f}_G = {}^b \mathbf{R}_i \begin{bmatrix} 0 \\ 0 \\ P \end{bmatrix} \quad {}^b \mathbf{f}_B = -{}^b \mathbf{R}_i \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} \quad (2.153)$$

So the restoring force and moment vector in the body frame is

$$\boldsymbol{\tau}_B = \begin{bmatrix} {}^b \mathbf{f}_G + {}^b \mathbf{f}_B \\ {}^b \mathbf{r}_G \times {}^b \mathbf{f}_G \end{bmatrix} \quad (2.154)$$

if ${}^b \mathbf{f}_G = -{}^b \mathbf{f}_B$ (valid for LTA vehicle) we arrive at the simpler form:

$$\boldsymbol{\tau}_B = \begin{bmatrix} 0 \\ {}^b \mathbf{r}_G \times {}^b \mathbf{f}_G \end{bmatrix} \quad (2.155)$$

2.10 Airship Equations of Motions

The equations of motion for dynamics and kinematics in the body frame could be written respectively as:

$$M\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu}) = \boldsymbol{\tau} \quad (2.156)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}\boldsymbol{\nu} \quad (2.157)$$

where $\nu = [u, v, w, p, q, r]^T$ $\eta = [x, y, z, \phi, \theta, \psi]$ and

$$J = \begin{bmatrix} {}^b R_i & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & E(\phi, \theta, \psi) \end{bmatrix} \quad (2.158)$$

$$M = M_{RB} + M_A \quad C(\nu) = C(\nu)_{RB} + C(\nu)_A \quad (2.159)$$

and

$$\tau = \tau_B + \tau_P + \tau_A + \tau_G$$

where

τ_G = generalized gravity force vector;

τ_A = generalized aerodynamic force vector

τ_P = generalized propulsion force vector;

τ_B = generalized buoyancy force vector.

2.11 Wind

This section shows how the wind-induced force and moments can be included in the equation of motion. The method is based on the assumption that the equation of motion can be represented in term of *relative velocity*

$$\nu_r = \nu - \nu_w \quad (2.160)$$

where $\nu_w = [u_c, v_c, w_c, 0, 0, 0]$ is a vector of irrotational body fixed wind velocities.

Let $[u_c^i, v_c^i, w_c^i]$ be the Earth (inertial) fixed wind velocity vector, in the body frame it will be

$${}^b \nu_w = {}^b R_i {}^i \nu_w \quad (2.161)$$

Let assume that the body-fixed wind velocity is constant or at least slowly-varying such that the following holds

$$\dot{\nu}_w = 0 \Rightarrow \dot{\nu}_r = \dot{\nu} \quad (2.162)$$

Hence the nonlinear relative equations of motions take the form

$$M\dot{\nu} + C(\nu_r)\nu_r = \tau_B + \tau_P + \tau_A(\nu_r) + \tau_G \quad (2.163)$$

$$\dot{\eta} = J\nu \quad (2.164)$$

2.12 Control Strategy

To control the airship we use a PD linear based control; essentially we are interested to control the longitudinal motion with a main thruster, a lateral (turning) motion using a stern thruster and yaw rudder, the pitch angle with lateral rudders and altitude with a tilt angle of the main thruster; so

$$u \in \mathbb{R}^5$$

It has been chosen to implement a control strategy using a reference trajectory designed a priori, this is a good approach especially in presence of wind (in a ratio 1/2 of the vehicle's velocity). This trajectory is not a set of high frequency sampled points (in the phase space) but a set of

waypoints sampled at low frequency; this is an hybrid approach between a nominal trajectory planning and the *orienting* technique. For each way point we have the *nominal* quantities

$$[{}^i\mathbf{r}_{nom}, {}^i\mathbf{v}_{nom}, \phi_{nom}] \quad (2.165)$$

that is the nominal inertial position, velocity and yaw angle. So we can define

$$\delta {}^i\mathbf{r} = {}^i\mathbf{r}_{nom} - {}^i\mathbf{r} \quad (2.166)$$

and in the body frame becomes

$$\delta {}^b\mathbf{r} = {}^b\mathbf{R}_i(\phi, \theta, \psi) \delta {}^i\mathbf{r} \quad (2.167)$$

Similarly we define

$$\delta {}^b\mathbf{v} = {}^b\mathbf{R}_i(\phi, \theta, \psi) {}^i\mathbf{v}_{nom} - {}^b\mathbf{v} \quad (2.168)$$

Now we can compute the norm of the position error in the xy plane

$$\delta r_{xy} = \|\delta {}^i\mathbf{r}(x, y)\| \quad (2.169)$$

Let's define the angular altitude error as

$$\delta \alpha = -\tan^{-1}(\delta r_z / \delta r_{xy}) \quad (2.170)$$

the vehicle's pitch error as

$$\delta \theta = (\theta - \delta \alpha) \quad (2.171)$$

and the vehicle's yaw error as

$$\delta \phi = \tan^{-1}(\delta r_y / \delta r_x) \quad (2.172)$$

We also define the quantity

$$\sigma = K_\sigma \tan^{-1}(\delta v_y / \delta v_z) \quad (2.173)$$

In order to apply a PD based control we have to define the generalized position error vector as

$$\delta \mathbf{X}(t) = [\delta r_{xy}(t), \delta \theta(t), \delta \phi(t), \delta r_z(t)]^T \quad (2.174)$$

and a generalized velocity vector

$$\delta \dot{\mathbf{X}}(t) = [\delta v_x(t), q(t), r(t) - \sigma, \delta v_z(t)]^T \quad (2.175)$$

If we define the proportional control matrix $\mathbf{K}_P$ as

$$\mathbf{K}_P = \begin{bmatrix} Kp_x & 0 & 0 & 0 \\ 0 & Kp_\theta & 0 & 0 \\ 0 & 0 & -Kp_{\phi,thr} & 0 \\ 0 & 0 & 0 & Kp_z \\ 0 & 0 & Kp_{\phi,rud} & 0 \end{bmatrix} \quad (2.176)$$

and the derivative control matrix $\mathbf{K}_D$

$$\mathbf{K}_D = \begin{bmatrix} Kd_x & 0 & 0 & 0 \\ 0 & Kd_\theta & 0 & 0 \\ 0 & 0 & Kd_{\phi,thr} & 0 \\ 0 & 0 & 0 & -Kd_z \\ 0 & 0 & -Kd_{\phi,rud} & 0 \end{bmatrix} \quad (2.177)$$

the input control vector is given by

$$\mathbf{u}(t) = \mathbf{D}_P \delta \mathbf{X}(t) + \mathbf{K}_D \delta \dot{\mathbf{X}}(t) \quad (2.178)$$

For each control element we define a saturation value such that

$$|u_i(t)| < u_{i,sat} \quad (2.179)$$

In order to make the response of the control not impulsive (so not realistic) the outputs are processed by a shaping filter to make the response smoother. The shaping filter process the control output following the model described by the differential equation

$$\frac{dg(t)}{dt} = -\alpha [g(t) - w(t)] \quad (2.180)$$

where $w(t)$ is the input function, and $\alpha = 1/\tau$ where τ is the time constant. To implement the filter in a discrete time system we have to discretize the above equation

$$g_{k+1} = -\Delta t \alpha (g_k - w_k) + g_k \quad (2.181)$$

Two control strategies has been developed from what discussed above, they are very similar in the concept but with different robustness:

- *leading way point*: here a *plane trajectory planning* is designed using an external tool developed on purpose. By fixing some consecutive points and specifying the desired cruise velocity to maintain, a curvature radius, and a sampling frequency, the tool returns an array of way points (in the phase space), sampled at the specified frequency, to form a trajectory that interpolates the initially user's fixed points. When the vehicle reaches a way point (with a tolerance) then it'll chase the next one (at a short distance ahead). It has been observed that this solution is more robust in case of high wind when the Airship is turning; this because the tangent slope in two consecutive way points varies slowly;
- *orienting way point*: here spare way points are sequentially placed in a plane and the vehicle has to reach them; for each way point a target velocity is specified. In this manner it's possible to design closed trajectory loops.

Chapter 3

Numerical Dynamics Implementation and Results

3.1 MASS: MATLAB Airship Slam Simulator

MASS is the software artifact produced by this work, it provides a physically based simulation environment for Airship kinematics, dynamics and the autonomous navigation using SLAM technique. The software, developed under MATLAB framework, could be used to evaluate dynamics and control strategies of the vehicle in different environmental conditions characterized i.e by wind and atmosphere density; the software implements SLAM algorithms to map an unknown environment by simulated IMU and Cameras measurements.

Basically the software is composed by two modules:

- Dynamic and Control Module (Airship dynamics and control for airship)
- State Estimation Module (SLAM algorithms)

The airship's software module developed is based on system modeling, which includes aerodynamic, airship actuators, and environmental modeling; the model bring together all discussed in the previous chapter; the aerodynamic model developed has the ability to simulate the cruise, hovering, and landing phase. The software has been developed in MATLAB from scratch, without using any special toolbox; it's has been possible to implement the equations of motion with simplified programming formalism, create a 3D trajectory and attitude visualizations and different parameters plots. In order to make the model as versatile as possible for changing flight conditions and optimization studies, all ship and environments are given parametric definitions. In order to verify the accuracy of the model the results were compared to literature.

With a user-defined vehicle flying in a user-defined environments it has been possible to simulate an airship flying on Earth and on the moon Titan.

For our purposes, the results obtained are very important to understand the kinematic behaviour of the vehicle in different environments conditions and to collect data for SLAM simulations; they also are very useful for future investigations on vehicle resources budgeting in term of propeller and power.

The flow diagram of the module reported in fig. 3.31; inputs for the dynamical model are:

- controlled thrusters (main and stern) engine, which responds to control feedback.
- controlled stern and pitch rudders;

- aerodynamic forces (pressure forces and viscous forces)
- wind
- coriolis and centrifugal forces
- gravity and buoyancy

3.2 Earth Airship Setup

The first vehicle used for our investigation is a typical commercial airship for terrestrial operations with length of 9 m and diameter 4 m.

the body frame axes are placed with the x -axis is along the center line of the vehicle in the direction of the motion, the z -axis directed upward and the y -axis complete the triad. The physical characteristics of the single parts of the airship are reported below; the body frame is centered in the center of buoyancy which coincide with the hull's center of volume (CV).

Airship is controlled by the following actuators:

- a main thruster for longitudinal motion;
- a tilt engine which turn the main thruster of a maximum angle of 20° ;
- a stern thruster for turning;
- pitch and yaw rudders, placed on the fins.

Fin Mass Property (Part Configuration)

Density = 80.00 kg/m³

Mass = 1.04 kg

Volume = 0.01 m³

Surface area = 2.65 m²

Gondola Mass Property (Part Configuration)

Density = 7.00 kg/m³

Mass = 7.88 kg

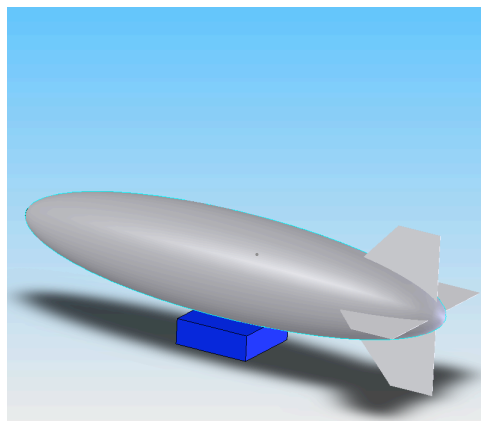


Figure 3.1: 3D model of the Earth Blimp

Volume = 1.13 m^3

Surface area = 7.50 m^2

Ellipsoid Gas Mass Property (Part Configuration)

Density = 0.18 kg/m^3

Mass = 3.36 kg

Volume = 18.83 m^3

Surface area = 45.31 m^2

Vehicle Mass Property (Assembly Configuration)

Density = 1.08 kg/m^3

Mass = 21.68 kg

Volume = 20.06 m^3

Surface area = 154.55 m^2

Center of the mass: (m)

X = -0.030018

Y = 0.000000

Z = -0.45

Moment of Inertia: ($\text{kg} \cdot \text{m}^2$)

With respect to Body Frame

$I_{xx} = 21.76$ $I_{xy} = 0.0006$ $I_{xz} = 0$

$I_{yx} = 0.0006$ $I_{yy} = 176.03$ $I_{yz} = 0$

$I_{zx} = 0$ $I_{zy} = 0$ $I_{zz} = 165.04$

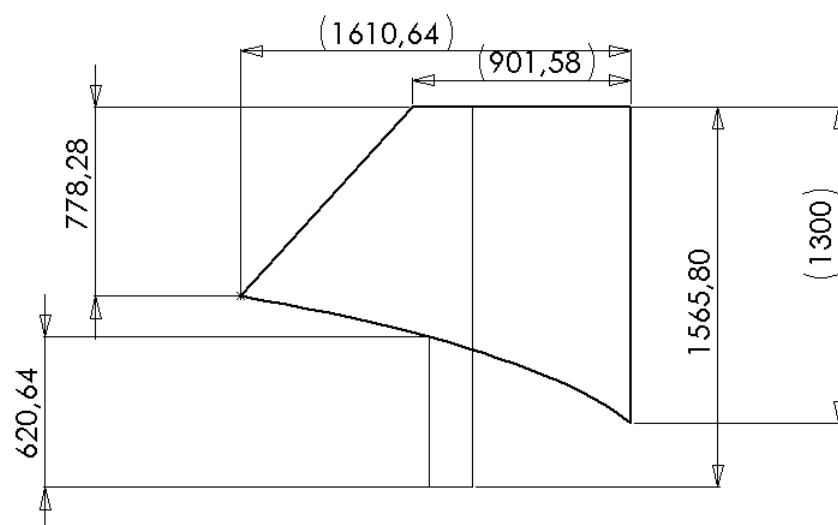


Figure 3.2: Design of the fin

The hull and fins added mass matrix coefficients are so determined from their shape

$$(\mathbf{M}_A)_h = \begin{bmatrix} 1.55 & 0 & 0 & 0 & 0 & 0 \\ 0 & 19.88 & 0 & 0 & 0 & 0 \\ 0 & 0 & 19.88 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 63.31 & 0 \\ 0 & 0 & 0 & 0 & 0 & 63.31 \end{bmatrix} \text{ Kg m}^2 \quad (3.1)$$

and the added mass matrix of the fins is

$$(\mathbf{M}_A)_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6.34 & 0 & 0 & 0 & -5.10 \\ 0 & 0 & 6.34 & 0 & 5.10 & 0 \\ 0 & 0 & 0 & 2.67 & 0 & 0 \\ 0 & 0 & 5.10 & 0 & 5.48 & 0 \\ 0 & -5.10 & 0 & 0 & 0 & 5.48 \end{bmatrix} \text{ Kg m}^2 \quad (3.2)$$

The aerodynamical parameters used to compute the viscous damping are

Parameter	Definition	Value
C_{d_c}	Hull Cross Flow Drag Coefficient	1.2
$(C_{d_h})_0$	Hull zero-angle axial drag coefficient	0.028
S_h	Hull reference area	7.08 m ²
η_{hf}	Hull efficiency factor due to hull-fins coupling	1.0
η_h	Hull efficiency factor due to finite length of the body	0.60
η_f	Fins efficiency factor due to hull-fins coupling	0.20
S_f	Fins reference area	0.4 · S_h m ²
$(C_{d_f})_0$	Fins zero angle axial drag coefficient	0.006
$(C_{d_c})_f$	Fins cross flow drag coefficient	1.8
$(\frac{\partial C_L}{\partial \alpha})_f$	Derivative of fin-lift coefficient wrt the angle of attack at zero incidence	5.73
$(l_f)_1$	Distance from the CV to the fins geometric center	3.8 m
$(l_f)_2$	Distance from the CV to the fins cross flow force center	3.0 m
$(\frac{\partial C_L}{\partial \delta})_f$	Derivative of fin lift-coefficient wrt the flap deflection angle	0.7

Table 3.1: Aerodynamic coefficients used for Earth airship

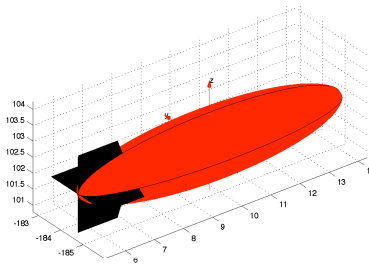


Figure 3.3: MASS: 3D Airship attitude visualization

3.3 Earth Simulations Results

Simulations on Earth environment presented in this section are based on the following hypothesis:

- the air density was supposed constant, $\rho = 1.2$;
- the local gravity vector was supposed constant;
- the air dynamic viscosity was considered constant $\mu = 1.78 \cdot 10^{-5}$ kg/m*s;
- the wind was supposed to act continuously in one direction (worst case) with constant velocity;
- no atmosphere turbulence was considered;
- the Reynolds number was supposed constant;
- vehicle altitude is 100 m;
- control adopted is of the type *leading way-point following*;

3.3.1 Airship in Neutral Flight

In this simulation the vehicle is supposed to fly without wind and yaw-controlled only by tail rudder. The 3D and 2D trajectory are reported in fig 3.4 and 3.27, following trends of attitude (roll pitch and yaw), angular velocity , position , linear velocity, and control engine.

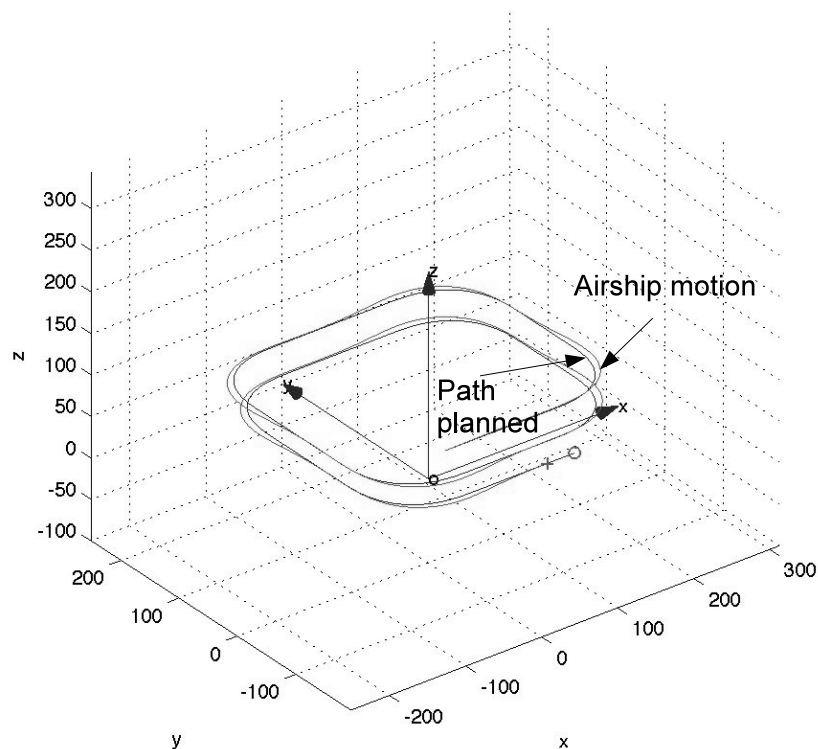


Figure 3.4: Blimp Neutral Flight, 3D trajectory

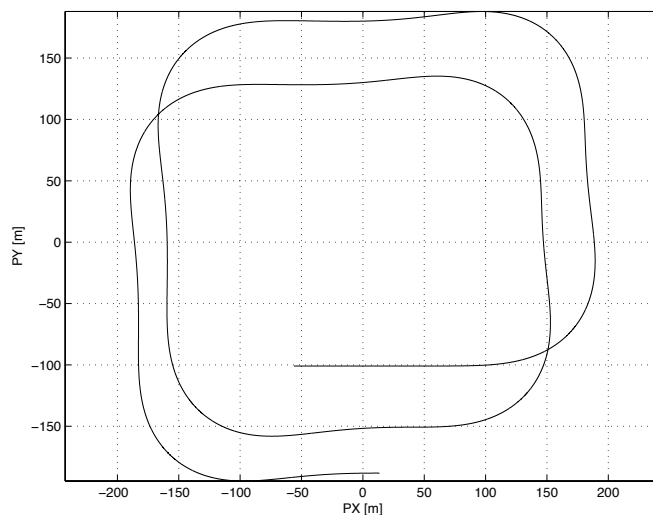
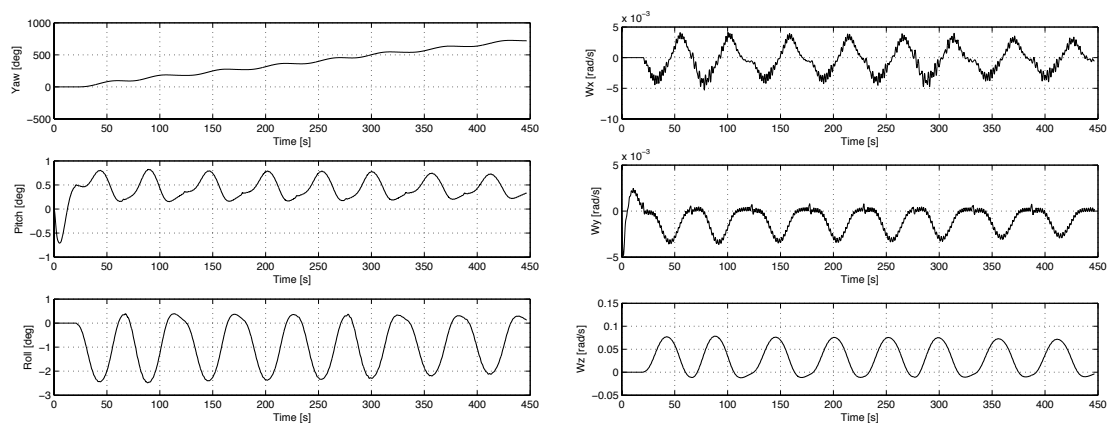
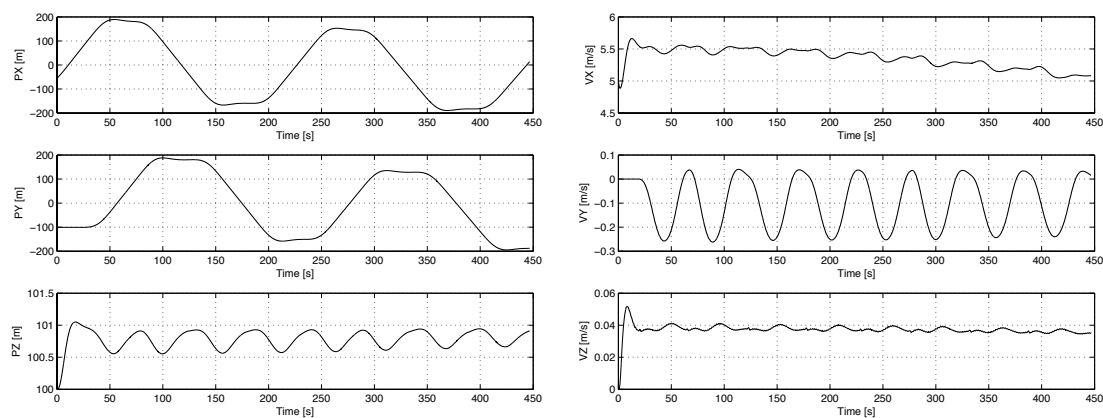


Figure 3.5: Blimp Neutral Flight, 2D trajectory

Figure 3.6: Airship attitude (Roll, Pitch, Yaw), angular velocity ${}^b\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$ Figure 3.7: Airship coordinates in the inertial frame, ${}^I\hat{\mathbf{r}}[x, y, z]$ and velocity ${}^b\mathbf{v} = [v_x, v_y, v_z]$

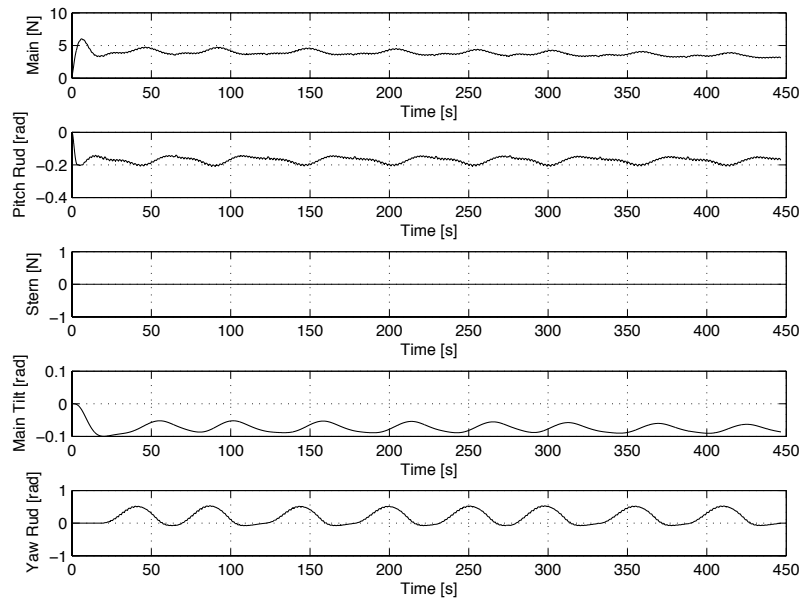


Figure 3.8: Control Engine Outputs

3.3.2 Airship in Constant Wind

In this simulation the vehicle is supposed to fly in a windy environment characterized by a 2m/s constant wind in the direction showed in fig. 3.9. Airship velocity is 5 m/s and it curves only by the actions of tail rudders. This is a typical setup for Earth mini-zeppelin airships.

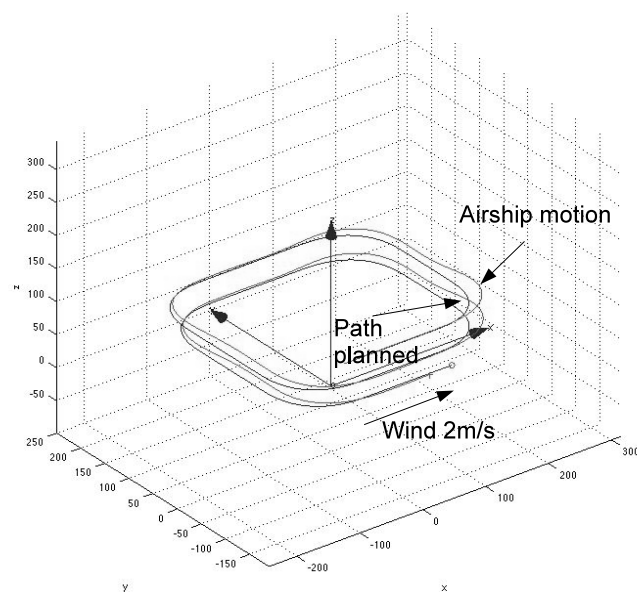


Figure 3.9: Airship in constant wind flight, 3D trajectory

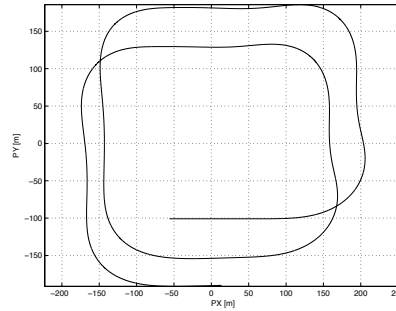
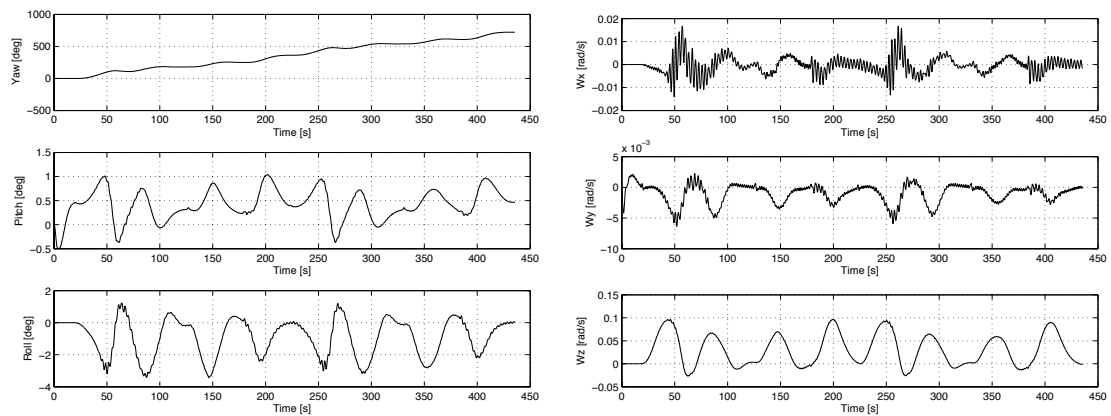
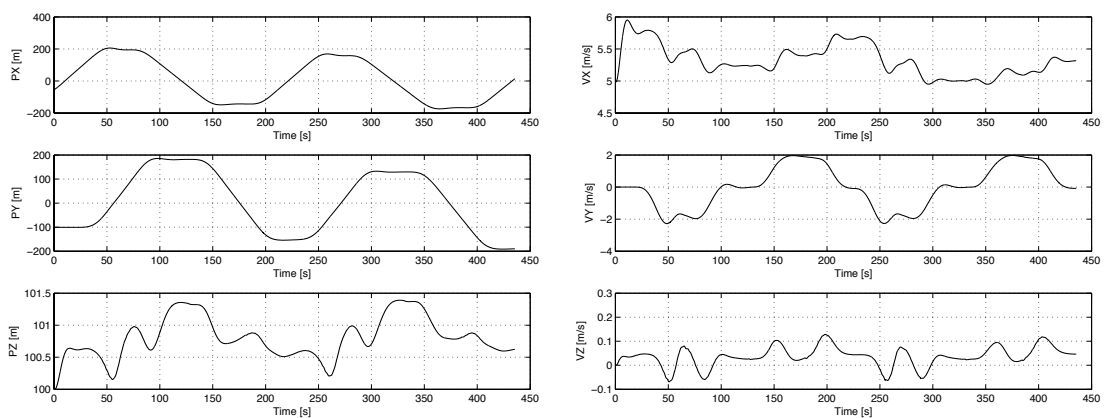


Figure 3.10: Airship in constant wind flight, 2D trajectory

Figure 3.11: Airship attitude (Roll, Pitch, Yaw), angular velocity ${}^b\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$ Figure 3.12: Airship coordinates in the inertial frame, ${}^I\hat{\mathbf{r}}[x, y, z]$ and velocity ${}^b\mathbf{v} = [v_x, v_y, v_z]$

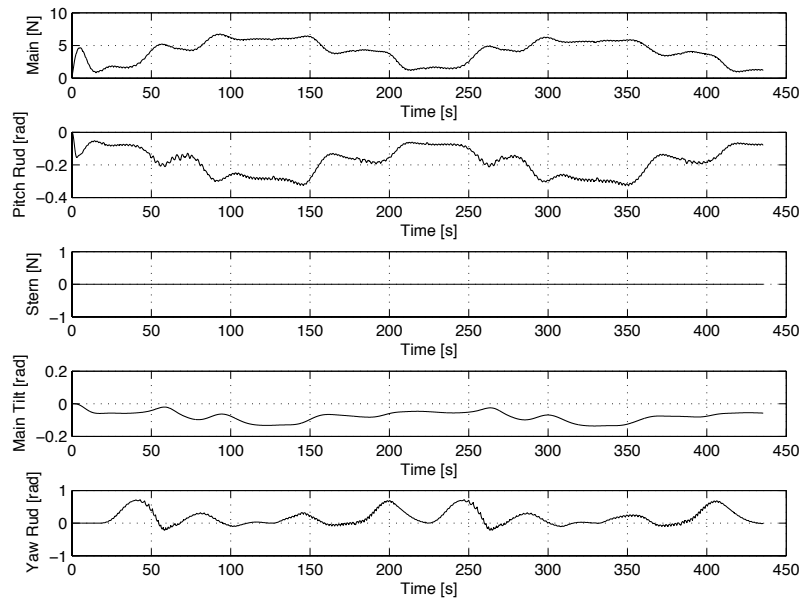


Figure 3.13: Control Engine Outputs

3.3.3 Airship in Constant Wind with Stern Engine

In this simulation the vehicle is supposed to fly under a 2m/s constant wind, in the direction showed in figure 3.14, with a longitudinal velocity of 5 m/s; the vehicle turns under the action of tail rudder and stern thruster. The latest is placed in the rear of the hull orthogonal to the longitudinal axis of the airship.

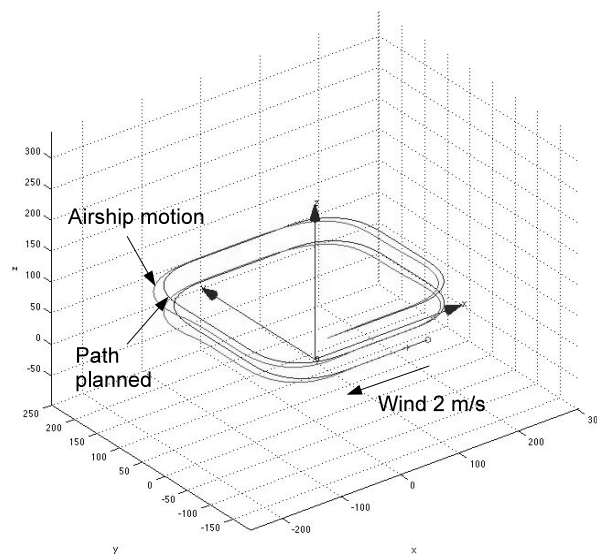


Figure 3.14: Airship in constant wind flight, 3D trajectory

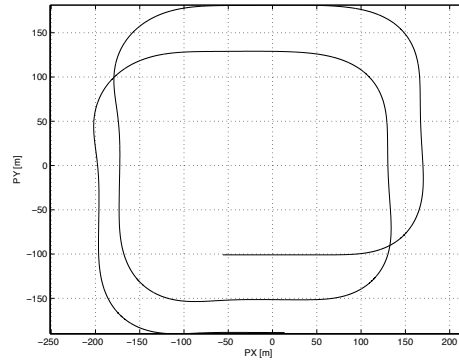
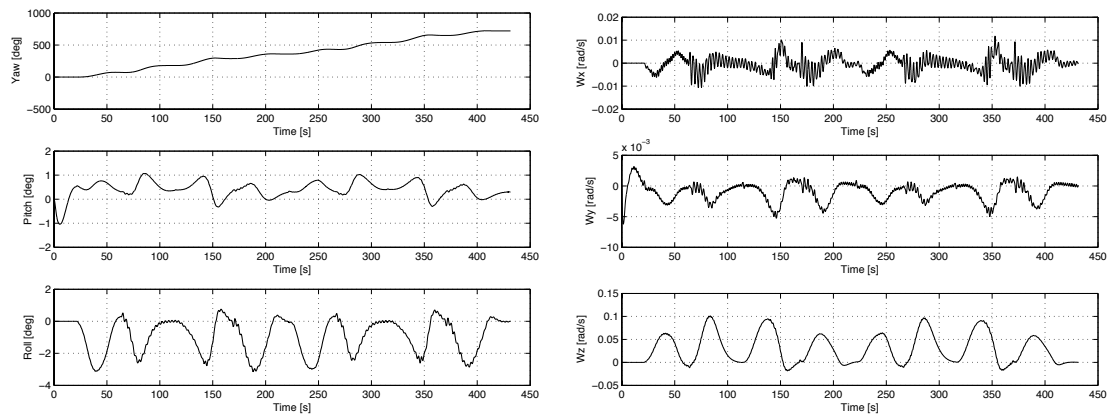
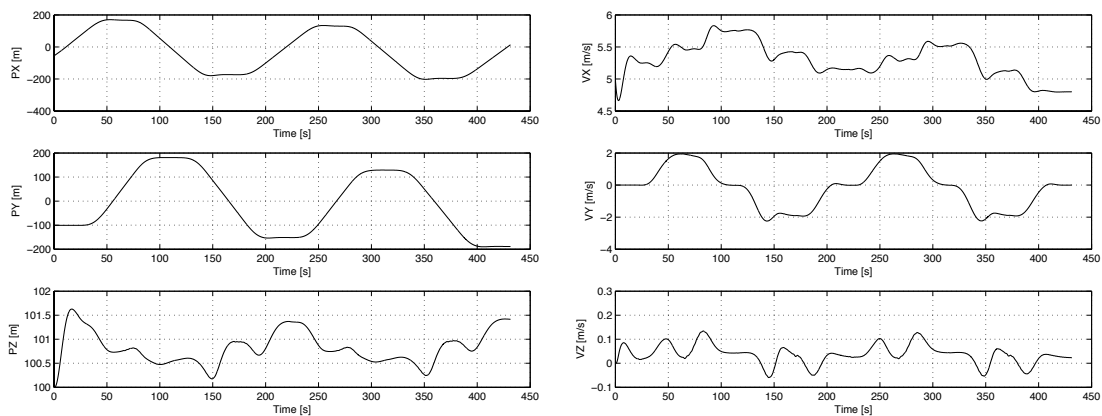


Figure 3.15: Airship in constant wind flight, 2D trajectory

Figure 3.16: Airship attitude (Roll, Pitch, Yaw), angular velocity ${}^b\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$ Figure 3.17: Airship coordinates in the inertial frame, ${}^I\hat{\mathbf{r}}[x, y, z]$ and velocity ${}^b\mathbf{v} = [v_x, v_y, v_z]$

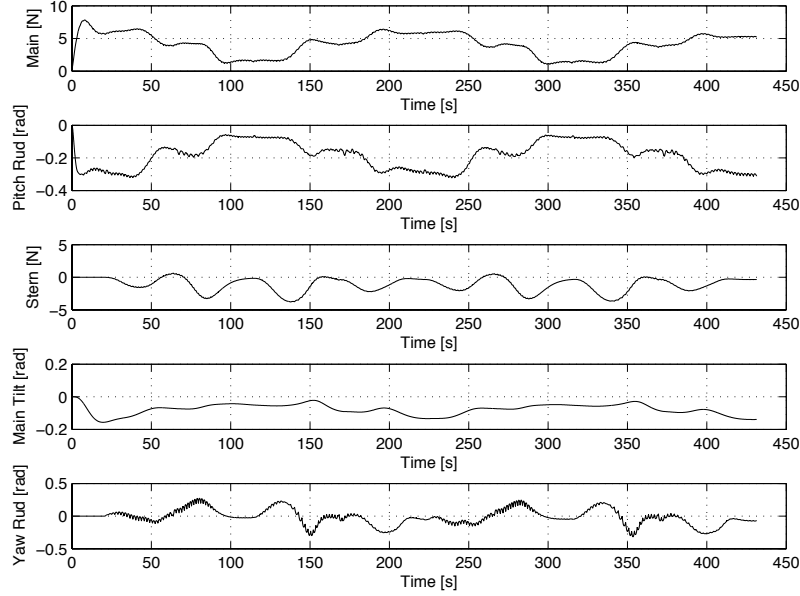


Figure 3.18: Control Engine Outputs

3.4 Titan Airship Setup

Our knowledge and understanding of Titan, Saturn's largest moon, have increased significantly as a result of measurements obtained from the Cassini spacecraft following its orbital insertion around Saturn on June 30, 2004 and even more recently with the measurements obtained during the descent of the Huygens probe through the atmosphere and onto the surface of Titan on January 14, 2005. The Titan Explorer Mission proposed by NASA is the next step in the exploration of this mysterious world. The Titan Explorer Mission consists of a Titan Orbiter and a Titan Airship that traverses the atmosphere of Titan and can land on its surface.

Titan's atmosphere is ideally suited for atmospheric flight; with the low gravity and the high atmospheric density, flight is readily achieved. An airship has been selected as the primary (or baseline) science delivery platform for this study. Airships combine the advantages of mobility of investigating multiple regions, with the ability for varying altitude and ultimately going repeatedly to the surface.

The vehicle showed in Figure (3.19). The physical characteristics of the single parts of the airship are reported below; from the original design proposed by NASA only the fins have been reshaped for better control issue.

The control feedback acts on

- a main thruster for longitudinal motion;
- a tilt engine which turn the main thruster of a maximum angle of 20° ;
- a stern thruster for turning;
- pitch and yaw rudders, placed on the fins.

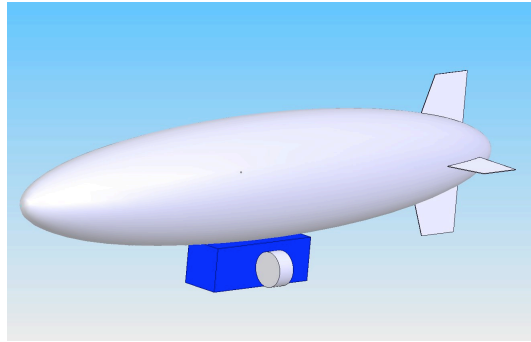


Figure 3.19: 3D model of the Titan Explorer

Fin Mass Property (Part Configuration)

Density = 90.00 kg/m³

Mass = 2.44 kg

Volume = 0.03 m³Surface area = 3.70 m²

Gondola Mass Property (Part Configuration)

Density = 37.00 kg/m³

Mass = 194.25 kg

Volume = 5.25 m³Surface area = 20.15 m²

Ellipsoid Helium Gas Mass Property (Part Configuration)

Density = 0.60 kg/m³

Mass = 48.13 kg

Volume = 80.83 m³Surface area = 129.36 m²

Airship Mass Property (Assembly Configuration)

Density = 3.62 kg/m³

Mass = 312.93 kg

Volume = 86.43 m³Surface area = 477.95.55 m²

Center of the mass: (m)

X = -0.21

Y = 0.00

Z = -1.55

Moment of Inertia: (kg * m²)

With respect to Body Frame

Ixx = 1413.41 Ixy = - 0.81 Ixz = 0

Iyx = -0.81 Iyy = 3571.05 Iyz = 0

Izx = 0 Izy = 0 Izz = 2424.66

Parameter	Definition	Value
C_{dc}	Hull Cross Flow Drag Coefficient	1.2
$(C_{dh})_0$	Hull zero-angle axial drag coefficient	0.016
S_h	Hull reference area	23.29 m ²
η_{hf}	Hull efficiency factor due to hull-fins coupling	1.0
η_h	Hull efficiency factor due to finite length of the body	0.62
η_f	Fins efficiency factor due to hull-fins coupling	0.20
S_f	Fins reference area	6.47 m ²
$(C_{df})_0$	Fins zero angle axial drag coefficient	0.006
$(C_{dc})_f$	Fins cross flow drag coefficient	1.8
$(\frac{\partial C_L}{\partial \alpha})_f$	Derivative of fin-lift coefficient wrt the angle of attack at zero incidence	5.73
$(l_f)_1$	Distance from the CV to the fins geometric center	8.0 m
$(l_f)_2$	Distance from the CV to the fins cross flow force center	8.0 m
$(\frac{\partial C_L}{\partial \delta})_f$	Derivative of fin lift-coefficient wrt the flap deflection angle	1.2

Table 3.2: Aerodynamic Coefficients for the Titan Explorer

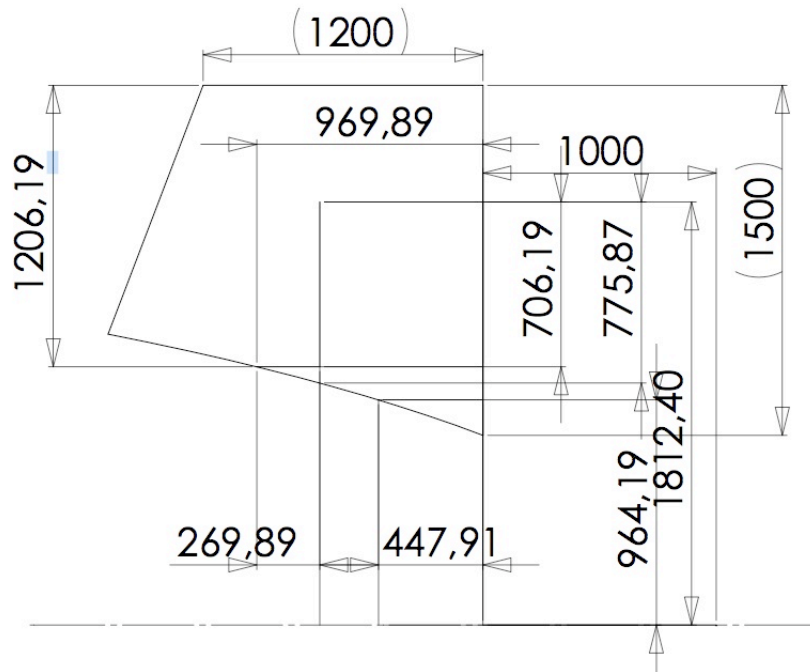


Figure 3.20: Design of the fin

The total Airship added mass matrix is

$$(\mathbf{M}_A) = \begin{bmatrix} 36.10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 605.68 & 0 & 0 & 0 & -43.73 \\ 0 & 0 & 60.56 & 0 & 43.73 & 0 \\ 0 & 0 & 0 & 69.41 & 0 & 0 \\ 0 & 0 & 43.73 & 0 & 6849.37 & 0 \\ 0 & -43.73 & 0 & 0 & 0 & 6849.37 \end{bmatrix} \text{ Kg m}^2 \quad (3.3)$$

3.5 Titan Simulations Results

Simulations on Titan environment presented in this section are based on the following hypotheses:

- the air density was supposed altitude dependent

$$\rho = 5.4627 - 0.21851h + 0.00294h^2 - 1.2054 \cdot 10^{-5}h^3 \text{ [kg/m}^3\text{]} \quad (3.4)$$

- the air dynamic viscosity

$$\mu = 6.43 \cdot 10^{-6} - 1.52 \cdot 10^{-7}h + 1.03 \cdot 10^{-8}h^2 - 3.28 \cdot 10^{-8}h^3 \text{ [kg/m} \cdot \text{s]} \quad (3.5)$$

- the wind was supposed to act continuously in one direction (longitudinal titan wind) with constant velocity 0.5 m/s
- no atmosphere turbulence was considered
- the Reynolds number was supposed constant, $6.67 \cdot 10^7$
- vehicle altitude is 100 m
- take off phase is not considered, the vehicle start with the cruise velocity of 5 m/s.
- airship uses stern thruster and rudders to turn

Both control strategies described in §2.12 are presented. Being the gravity less than in Earth, the gondola restoring moment is not sufficient to maintain low roll angles if the control engine acts too much in stern thrusters. In *leading way point follow* method, the proportional coefficient is low respect the classical way point follow technique to maintain the same velocity because consecutive points are nearer. This reduces the thrust necessary especially during turning where the airship tends to roll. Anyway the difference is limited to some degrees only. The engine cost for longitudinal flight of our model is in agreement with the preliminar study by NASA/JPL [J.S. Levine and H.S. Wright, 2005].

3.5.1 Way-Point Following

In this simulation a closed-loop trajectory is obtained by placing eight way-points; the curvature radius of the "circular" path is 200 m over an area of approximately 1 km².

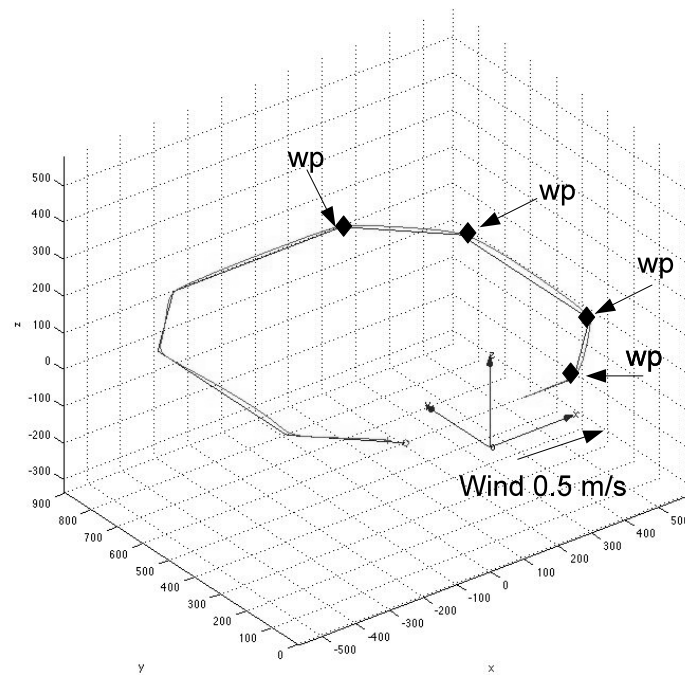


Figure 3.21: Titan Explorer controlled by classical way point placement to form a closed path

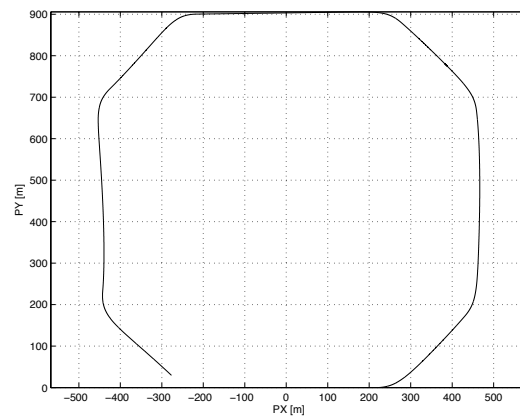


Figure 3.22: 2D trajectory

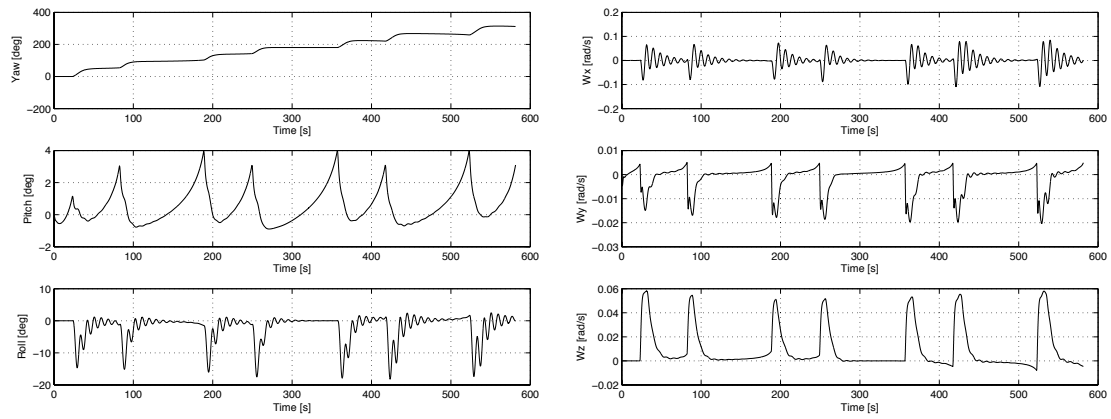


Figure 3.23: Airship attitude (Roll, Pitch, Yaw), angular velocity ${}^b\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$

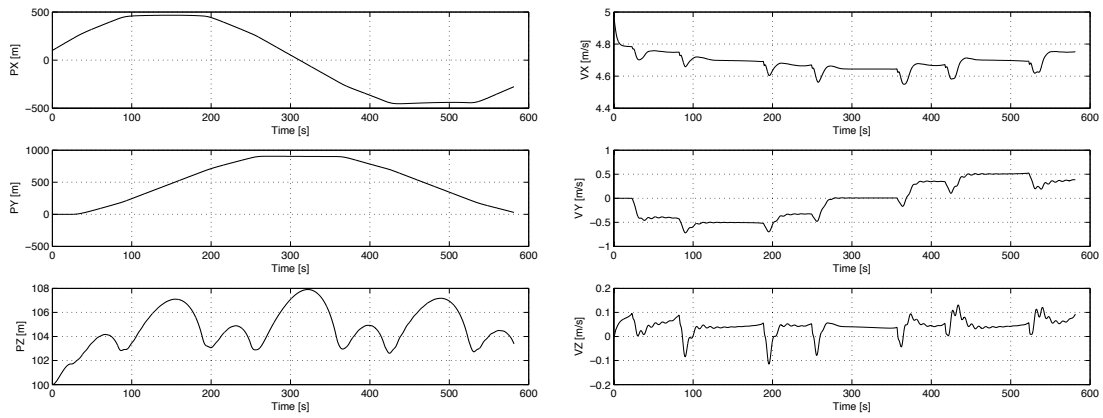


Figure 3.24: Airship coordinates in the inertial frame, ${}^I\hat{\mathbf{r}}[x, y, z]$ and velocity ${}^b\mathbf{v} = [v_x, v_y, v_z]$

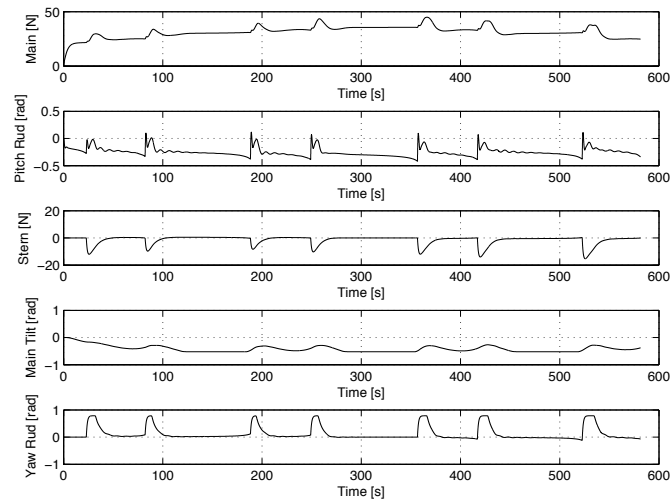


Figure 3.25: Control Engine Outputs

3.5.2 Leading Way-Point Following

The reference trajectory is similar to that drawn for Earth flight but with a higher curvature radius of the interpolating curve. The roll angle is reduced respect the previous simulation like the variation of the airship altitude during the flight due to lift force.

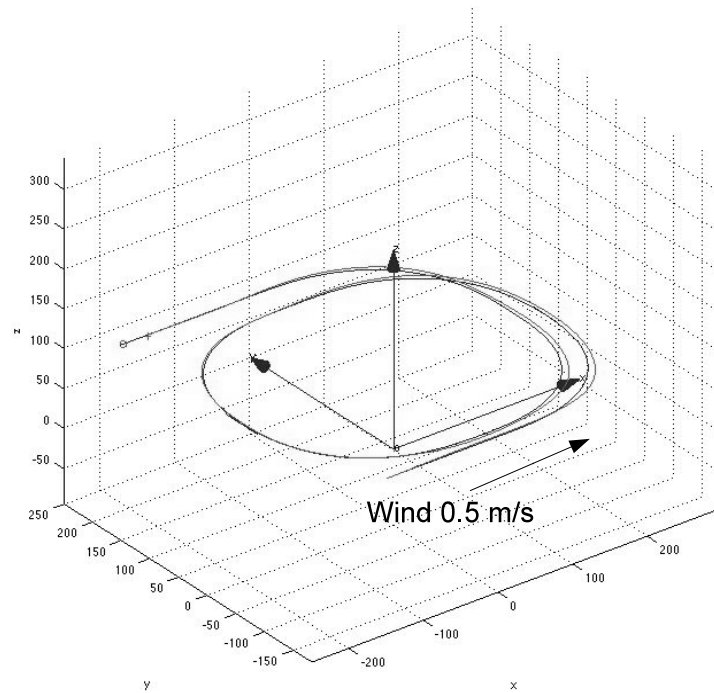


Figure 3.26: With leading way-point control trajectory is more smooth

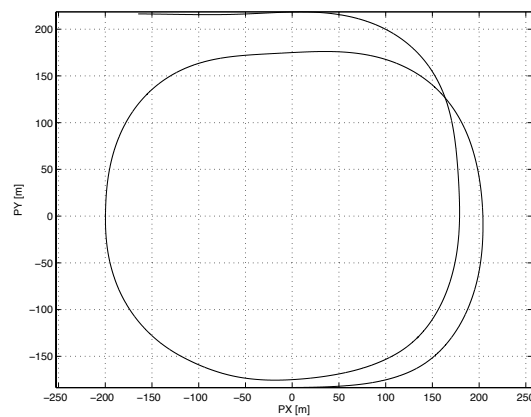


Figure 3.27: 2D trajectory

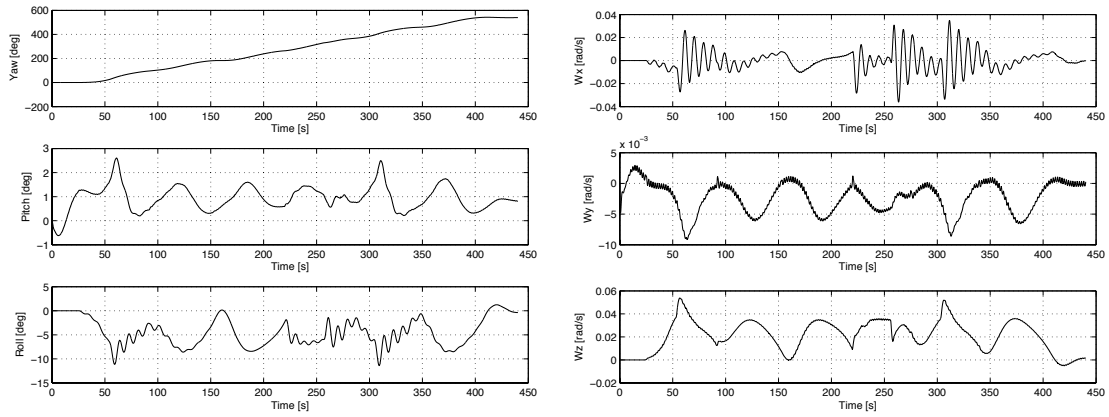


Figure 3.28: Airship attitude (Roll, Pitch, Yaw), angular velocity ${}^b\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$

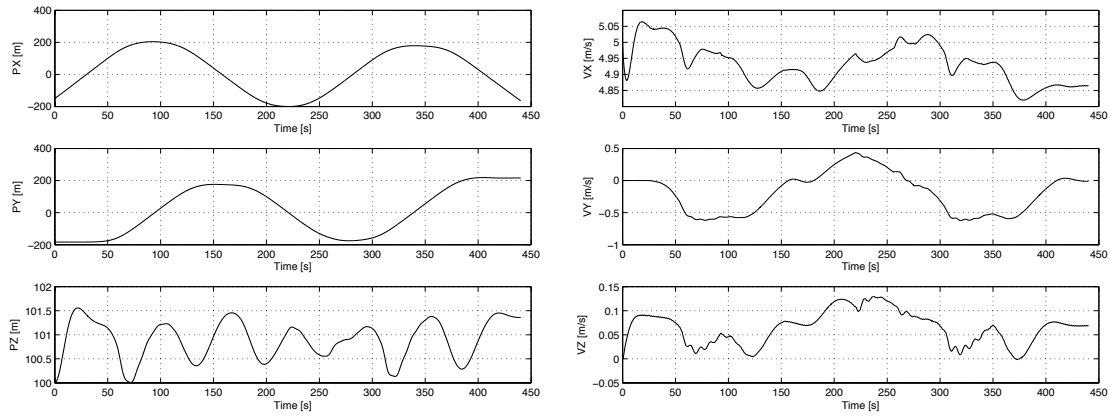


Figure 3.29: Airship coordinates in the inertial frame, ${}^I\hat{\mathbf{r}}[x, y, z]$ and velocity ${}^b\mathbf{v} = [v_x, v_y, v_z]$

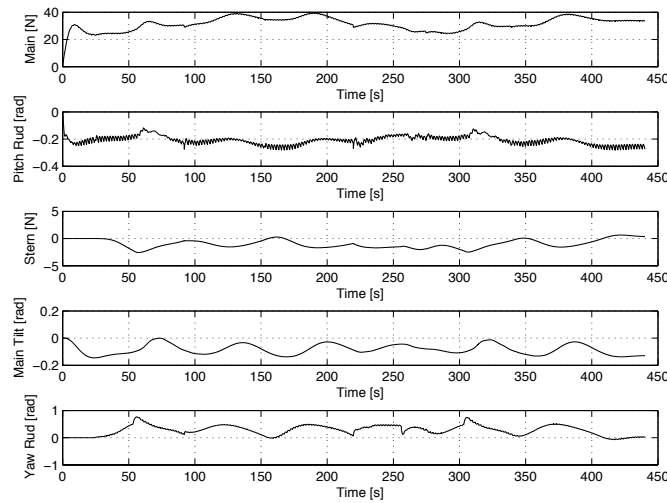


Figure 3.30: Control Engine Outputs

Dynamic and Control Module (DCM)

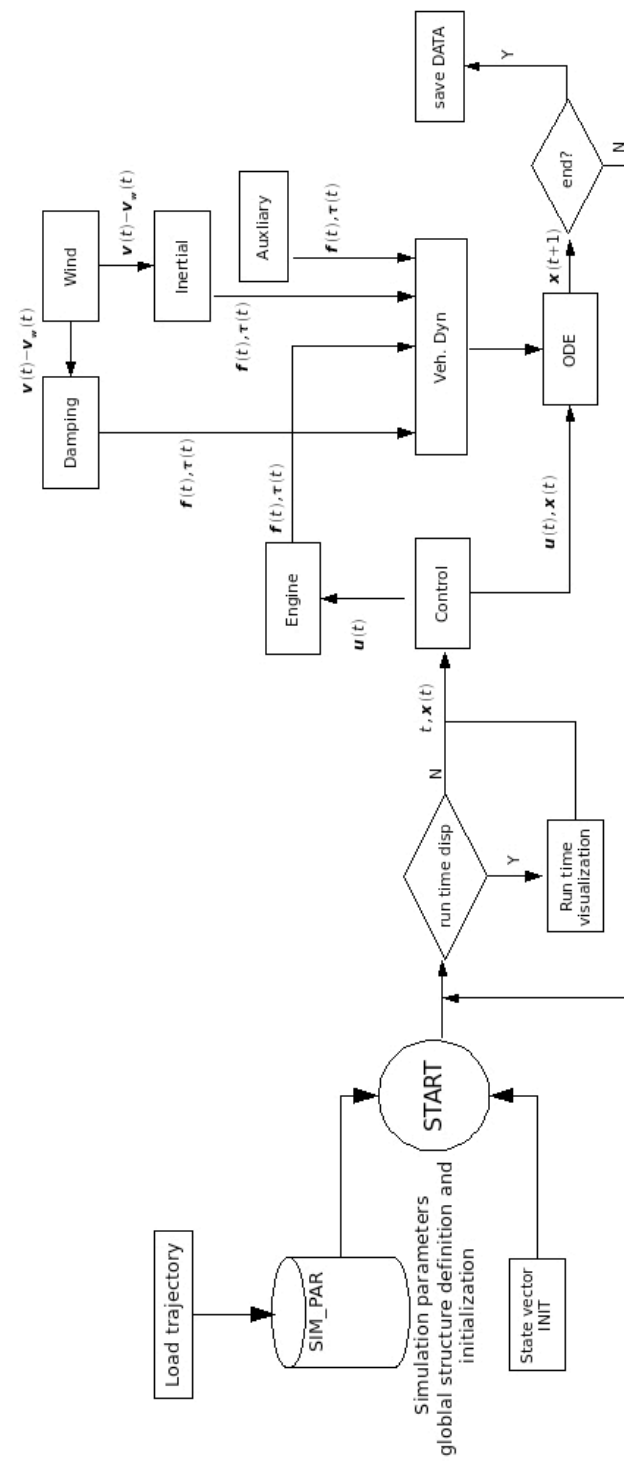


Figure 3.31: MASS DCM flow diagram

Chapter 4

Navigation Sensors

Navigation and guidance are defined as the processes of determining and controlling the position of a vehicle.

The integration of different types of navigation systems is frequently used in autonomous systems due to the fact that particular errors existing in anyone of them are usually of different physical natures.

Systems based on inertial sensors (rate gyros and linear accelerometers) are often coupled with other absolute position sensors as GPS in order to control biases and drift whose corrupt the measure over the time.

The vision navigation systems (VNS) provide measurement of range and bearing at a low frequency that bound positioning errors; the basic idea for an autonomous navigation system consists in coupling high frequency IMU measurement with vision sensors to reduce position errors and localize a vehicle in a local map by SLAM technique. In this chapter we analyze the property of both sensors to have an efficient and realistic model to use in the estimation process discussed in chapter 5. In particular are discussed the "strap-down" and the pinhole camera as measurement models for IMU and VNS respectively.

4.1 Inertial Navigation System

Inertial sensors make measurements of the internal state of the vehicle. A major advantage of inertial sensors is that they are non-radiating and non-jammable and may be packaged and sealed from the environment. This makes them potentially robust in harsh environmental conditions. Historically, Inertial Navigation Systems have been used in aerospace vehicles, military applications such as ships, submarines, missiles, and to a much lesser extent, in land vehicle applications. Only a few years ago the application of inertial sensing was limited to high performance high cost aerospace and military applications. However, motivated by requirements for the automotive industry, a whole variety of low cost inertial systems have now become available in diverse applications involving heading and attitude determination.

The most common type of inertial sensors are accelerometers and gyroscopes. Accelerometers measure acceleration with respect to an inertial reference frame. Gyroscopes measure the rate of rotation independent of the coordinate frame.

4.1.1 Accelerometers

The accelerometers measure the inertia force generated when a mass is affected by change in velocity. This force may change the tension of a string or cause a deflection of beam or may even change the vibrating frequency of a mass. The accelerometers are composed of three main elements: a mass, a suspension mechanism that positions the mass and a sensing element that returns an observation proportional to the acceleration of the mass. A basic one-degree of freedom accelerometer is shown in figure 4.1. This accelerometer is usually referred to as an open loop since the acceleration is indicated by the displacement of the mass. There are other different

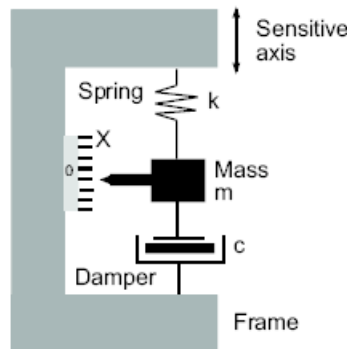


Figure 4.1: Open loop accelerometer

type of accelerometer as *pendulus integrating gyros accelerometer*, *vibrating string accelerometer*, *fiber optic accelerometer*.

4.1.2 Gyroscopes

These devices return an output proportional to the rotational velocity. There is a large variety of gyroscopes that are based on different principles. The price and quality of these sensors varies significantly. A gyroscope is a device for measuring or maintaining orientation, based on the principle of conservation of angular momentum. The essence of the device is a spinning wheel on an axle. The device, once spinning, tends to resist changes to its orientation due to the angular momentum of the wheel.

A common type of gyro is the single-degree-of-freedom gyro. The gyro rotor is mounted on a single gimbal (fastened to the spin axis and gyro case) and allows freedom of inner gimbal motion about only one axis.

If you attempt to change the gyro's plane of rotor spin by rotating the case about the input axis (figure 4.2), the gyro will precess; turning the gyro case has the same effect as applying a torque on the spin axis. An angle measurement device could be mounted in the output axes. Three single degree-of-freedom gyros are required for three axis stabilization.

4.1.3 Full 6-DOF INS

A full inertial navigation system (IMU) consists of at least three (triaxial) accelerometers and three orthogonal gyroscopes that provide measurements of acceleration in three dimensions and rotation rates about three axes. An IMU system assembled from low cost solid-state components is almost always constructed in a "strap-down" configuration. This term means that the gyros and accelerometers are fixed to a common chassis and are not actively controlled on gimbals to align

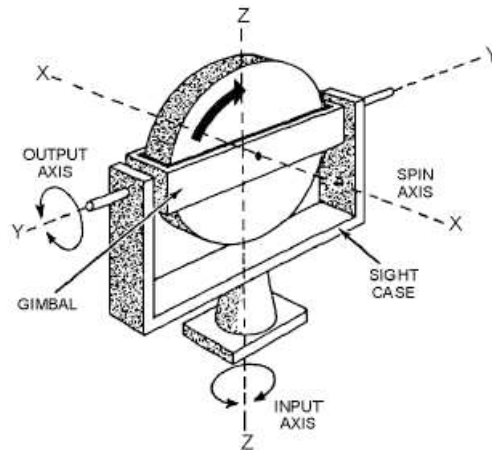


Figure 4.2: Single-degree-of-freedom gyro

themselves in a pre-specified direction. This design has the advantage of eliminating all moving parts. The strap-down construction, however, needs substantially more complex algorithms in order to compute and distinguish true linear acceleration from angular acceleration and body roll or pitch with respect to gravity.

Once true linear acceleration has been determined, vehicle position may be obtained, in principle, by double integration of the acceleration.

Vehicle orientation and attitude may also be determined by integration of the rotation rates of the gyros. In practice this integration leads to unbounded growth in position errors with time due to the noise associated with the measurement and the non-linearity of the sensors. Consequently, IMU information is useful for determining position and orientation only over short periods of time. How long one can rely on IMU information to provide a measure of position and orientation depends ultimately on the magnitude of the measurement error or *drift rate*.

The predicted trajectory will mainly be a function of the initial calibration and alignment of the platform. By *calibration* is meant the determination of the biases on the accelerometers and gyros¹, the *alignment* process consists of determining the initial orientation of the platform. This is of fundamental importance since the dead-reckoning algorithm will use this initial orientation to update the attitude information and then evaluate the accelerations in the navigation frame.

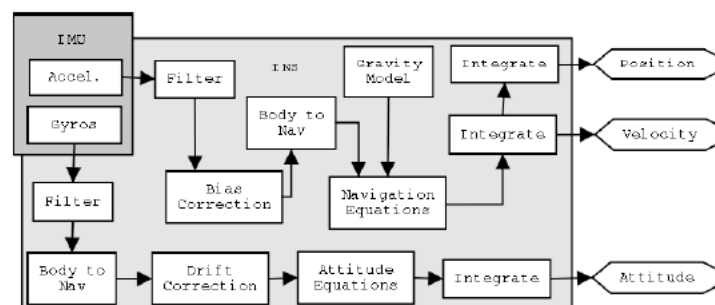


Figure 4.3: Inertial Navigation System

¹Many IMU platform are self-calibrating systems

4.1.4 Strap Down Navigation Equations

In this section we derive the fundamental equations to model a strap-down navigation sensor; this model will be used into the SLAM filtering process as an input of the system and to simulate specific force measurement on the vehicle from dynamical data. The fundamental equation of inertial navigation is

$${}^i\ddot{\mathbf{P}} = -\frac{GM}{P^3} {}^i\mathbf{P} + {}^i\mathbf{S} = {}^i\mathbf{g} + {}^i\mathbf{S} \quad (4.1)$$

where ${}^i\mathbf{g}$ is the gravity acceleration vector and the vector ${}^i\mathbf{S}$ included specific force due to thrust, lift, drag etc.

In inertial navigation, accelerometers and gyro detect accelerations and moments due to forces exerted on the body. These forces are typically referred to as specific force. Thus reading from the IMU will be referred to as specific forces, which are independent of the mass. The navigation equations for the Earth Centered Earth Fixed (ECEF) system are now obtained below.

For the position vector we have

$${}^f\mathbf{P} = {}^f\mathbf{R}_i {}^i\mathbf{P}$$

solving by ${}^i\mathbf{P}$ and take the time derivative we get

$${}^i\dot{\mathbf{P}} = {}^f\tilde{\omega}_e {}^i\mathbf{R}_f {}^f\mathbf{P}$$

where

$${}^f\tilde{\omega}_e = \begin{bmatrix} 0 & -\omega_e & 0 \\ \omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Differentiating another time we have

$${}^i\ddot{\mathbf{P}} = {}^i\mathbf{R}_f ({}^f\tilde{\omega}_e^2 {}^f\hat{\mathbf{P}} + 2{}^f\tilde{\omega}_e {}^f\dot{\mathbf{P}} + {}^f\ddot{\mathbf{P}}) \quad (4.2)$$

Substituting (4.2) in the fundamental navigation equation

$${}^i\ddot{\mathbf{P}} = {}^i\mathbf{g} + {}^i\mathbf{S}$$

we get

$${}^i\mathbf{R}_f ({}^f\tilde{\omega}_e {}^f\dot{\mathbf{P}} + 2{}^f\tilde{\omega}_e {}^f\dot{\mathbf{P}} + {}^f\ddot{\mathbf{P}}) = {}^i\mathbf{R}_f ({}^f\mathbf{g} + {}^f\mathbf{S})$$

or

$${}^f\ddot{\mathbf{P}} + {}^f\tilde{\omega}_e {}^f\tilde{\omega}_e {}^f\hat{\mathbf{P}} + 2{}^f\tilde{\omega}_e {}^f\dot{\mathbf{P}} = {}^f\mathbf{g} + {}^f\mathbf{S} \quad (4.3)$$

We can write this second order differential equation as a system of a first order by setting

$${}^f\dot{\mathbf{P}} = {}^f\mathbf{V}$$

and so write

$$\begin{bmatrix} {}^f\dot{\mathbf{V}} \\ {}^f\dot{\mathbf{P}} \end{bmatrix} = \begin{bmatrix} -2{}^f\tilde{\omega}_e {}^f\mathbf{V} & -{}^f\tilde{\omega}_e {}^f\tilde{\omega}_e \\ \mathbb{I} & 0 \end{bmatrix} \begin{bmatrix} {}^f\mathbf{V} \\ {}^f\mathbf{P} \end{bmatrix} + \begin{bmatrix} {}^f\mathbf{g} + {}^f\mathbf{S} \\ 0 \end{bmatrix} \quad (4.4)$$

In strap-down systems, specific forces are measured in the body frame, so the last equation become

$$\begin{bmatrix} {}^f\dot{\mathbf{V}} \\ {}^f\dot{\mathbf{P}} \end{bmatrix} = \begin{bmatrix} -2{}^f\tilde{\omega}_e {}^f\mathbf{V} & -{}^f\tilde{\omega}_e {}^f\tilde{\omega}_e \\ \mathbb{I} & 0 \end{bmatrix} \begin{bmatrix} {}^f\mathbf{V} \\ {}^f\mathbf{P} \end{bmatrix} + \begin{bmatrix} {}^f\mathbf{R}_b & {}^f\mathbf{R}_b \\ 0 & 0 \end{bmatrix} \begin{bmatrix} {}^b\mathbf{g} \\ {}^b\mathbf{S} \end{bmatrix} \quad (4.5)$$

Now considering attitude kinematics (see §2.3) we have

$$\begin{bmatrix} {}^f\dot{\mathbf{V}} \\ {}^f\dot{\mathbf{P}} \\ \dot{\Phi} \end{bmatrix} = \begin{bmatrix} -2{}^f\tilde{\omega}_e {}^f\mathbf{V} & -{}^f\tilde{\omega}_e {}^f\tilde{\omega}_e & 0 \\ \mathbb{I} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} {}^f\mathbf{V} \\ {}^f\mathbf{P} \\ \Phi \end{bmatrix} + \begin{bmatrix} {}^f\mathbf{R}_b & {}^f\mathbf{R}_b & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathbf{H} \end{bmatrix} \begin{bmatrix} {}^b\mathbf{g} \\ {}^b\mathbf{S} \\ {}^b\omega \end{bmatrix} \quad (4.6)$$

where ${}^b\mathbf{S}$ and ${}^b\boldsymbol{\omega}$ are acceleration and rotation rate obtained from the system.

A more simplified model could be derived under some hypothesis. An airship could be considered as a vehicle that moves respect the atmosphere, and its velocity respect the ground is small so any inertial effect due to Earth rotation could be neglect. Moreover in most slam applications the landscape dimension is small compared with the planet and so also variation in the geoid shape could be neglected.

Finally only three reference frames are required for local navigation; the vehicle reference frame, the sensor reference frame and a pseudo-inertial frame, with origin in a arbitrary point of the surface, where reference points in the map. In general the vehicle frame and the inertial frame are related by the homogeneous transformation

$${}^i\mathbf{H}_b = \begin{bmatrix} {}^i\mathbf{R}_b & {}^iT_b \\ \mathbf{0} & 1 \end{bmatrix} \quad (4.7)$$

Under these assumptions we proceede to find out the proper *system mechanization* to which our model will refer to.

The different choice of the reference frame results in a variation of the mechanization; for our purposes it has been implemented the *body frame mechanization* where it's required to calculate vehicle speed respect the body axes.

The equation of the inertial motion expressed in the body frame is ²

$${}^b\dot{\mathbf{v}} = \partial_t {}^b\mathbf{v} + {}^b\boldsymbol{\omega} \times {}^b\mathbf{v} \quad (4.8)$$

where the ∂ symbol refers to differentiation with respect the moving axes.

Using the navigation equation

$${}^b\dot{\mathbf{v}} = {}^b\mathbf{S} + {}^b\mathbf{g} \quad (4.9)$$

In this equation, ${}^b\mathbf{S}$ represent the *specific force acceleration* to which ther navigation system is subjected while ${}^b\boldsymbol{\omega} \times {}^b\mathbf{v}$ is the acceleration caused by the moving frame; so we have

$${}^b\mathbf{S} = \partial_t {}^b\mathbf{v} + {}^b\boldsymbol{\omega} \times {}^b\mathbf{v} - {}^b\mathbf{g} \quad (4.10)$$

where ${}^b\mathbf{g}$ is the local gravity vector. If the system isn't located in the origin of the body frame we have to consider the apparent term.

If ${}^b\mathbf{r}_S$ is the radius vector of the system in the body frame we have

$${}^b\ddot{\mathbf{r}}_S = {}^b\dot{\boldsymbol{\omega}} \times {}^b\mathbf{r}_S + {}^b\boldsymbol{\omega} \times ({}^b\boldsymbol{\omega} \times {}^b\mathbf{r}_S) \quad (4.11)$$

so the navigation equation become

$${}^b\bar{\mathbf{S}} = {}^b\dot{\mathbf{v}} + {}^b\dot{\boldsymbol{\omega}} \times {}^b\mathbf{r}_S + {}^b\boldsymbol{\omega} \times ({}^b\boldsymbol{\omega} \times {}^b\mathbf{r}_S) - {}^b\mathbf{g} \quad (4.12)$$

where ${}^b\bar{\mathbf{S}}$ represent the specific force measured on board. The model equation to integrate by the IMU is

$$\partial_t {}^b\mathbf{v} = {}^b\bar{\mathbf{S}} + {}^b\mathbf{g} - {}^b\boldsymbol{\omega} \times {}^b\mathbf{v} - {}^b\dot{\boldsymbol{\omega}} \times {}^b\mathbf{r}_S - {}^b\boldsymbol{\omega} \times ({}^b\boldsymbol{\omega} \times {}^b\mathbf{r}_S) \quad (4.13)$$

it's the strapdown IMU model equation.

Basically to implement inertial navigation in SLAM algorithms we need:

- simulate the acquisition from body mounted accelerometers, this is done by

$${}^b\bar{\mathbf{S}} = {}^b\mathbf{S} + {}^b\ddot{\mathbf{r}}_S \quad (4.14)$$

where ${}^b\mathbf{S}$ is given by (4.10) if the sensor is placed in CV, and the $\partial_t {}^b\mathbf{v}$, at any time, is given by the airship dynamical equation implemented in the Dynamics and Coltrol Module of the software simulator and loaded by the navigation software.

²Note that $\mathbf{r}$ is the radius from the body frame origin and the inertial. So we suppose that the system is placed in the origin of the body frame

- simulate the acquisition from body mounted gyroscopes
- derive the IMU model equation to compute the jacobian in the predictive stage of the Kalman Filter by

$${}^b\dot{\mathbf{v}} = {}^b\mathbf{S} + {}^b\mathbf{g} - {}^b\boldsymbol{\omega} \times {}^b\mathbf{v} - {}^b\dot{\boldsymbol{\omega}} \times {}^b\mathbf{r}_S - {}^b\boldsymbol{\omega} \times ({}^b\boldsymbol{\omega} \times {}^b\mathbf{r}_S) \quad (4.15)$$

In the MASS simulator a fiber optic IMU is considered for error modeling for navigation; in particular main features of the device are reported in table 4.1:

Update rate[Hz]	>125
Angular Rate	
Angular rate[°/s]	> ± 200
Bias [°/hr]	< ± 20
Resolution [°/s]	<0.025
Bandwidth [Hz]	> 100
Acceleration	
Range [g]	> ± 4
Bias [mg]	> ± 12
Resolution [mg]	<0.6
Bandwidth [Hz]	> 75

Table 4.1: Fiber Optic IMU700CB-200

4.1.5 Biases

The accelerometers measure the absolute acceleration with respect to an inertial frame. We are interested in the translational acceleration hence the algorithms used should compensate for all other accelerations. For practical purposes it can be considered that gravity is the only other acceleration present. The orientation of the accelerometer has to be known with very good accuracy in order to compensate for gravity without introducing errors comparable to the actual translation acceleration. This has been the main limitation that has prevented the widespread use of inertial sensors being used as position sensors in low cost implementations.

The orientation of the accelerometer can be tracked using a gyroscope. These sensors provide an output proportional to the rotation speed but are always contaminated by noise and drift.

The IMU's accelerometers and gyros are sensitive to temperature as shown in [E.Nebot and H.Durrant-Whyte, 1997]. Thus as the temperature of the IMU changes, the associated bias and drift will change until the temperature reaches steady state or remains the same. This is not critical in our application, we just wait for the IMU to reach steady state before trusting the readings. However if this system was mounted in an aircraft which changed altitude and temperatures, this would be a problem.

Vibration in a strap-down system can cause many problems with the INS. Generally great care must be taken to isolate the IMU from any resonance frequencies. In high precision systems, various tests must be done to try to identify what these frequencies are then design elaborate mounts to hold the IMU.

The drift rates and accelerometer biases tend to change each time the unit is switched on. This is due to the fact that measurements are noisy. Typically there is a low pass filter used to remove some of this noise before the measurements are used in the navigation equations (also realistically, there tends to be low pass filtering somewhere in the system due to hardware limitation because not everything has infinite bandwidth). When random noise is filtered, this

produces what is called a random walk. The integration of this random walk will result in velocity and positions moving at different rates during different runs even though the IMU (and vehicle) are in the same orientation and experiencing the same accelerations during each run (hysteresis).

Constant gyro bias intruduces an error in position determination [E.Nebot and H.Durrant-Whyte, 1997] that grows as

$$\epsilon_{\omega} \propto Bt^3 \quad (4.16)$$

The bias in the accelerometer will also increase the error position too, as

$$\epsilon_{acc} \propto Bt^2 \quad (4.17)$$

The calibration procedure must incorporate the identification of gyro and accelerometer biases. In addition to the prefiltering of the IMU data, an extended Kalman filter could be used, with another sensor measurements, to estimate the biases and drifts of the system and then update the navigational solution [K.J.Walchko and P.A.C.Manson, 2002],[M.S.Grewal et al., 1990].

Usually in terresrtial application it is made by using absolute GPS measurement, but with the lack of GPS signal others techniques have to be used.

4.1.6 IMU Error Modeling

We start writing the navigation equation with “yaw-pitch-roll“ ϕ, θ, ψ attitude parametrization in the EICF reference frame

$$\begin{bmatrix} {}^f\dot{\mathbf{V}} \\ {}^f\dot{\mathbf{P}} \\ {}^f\dot{\mathbf{\Phi}} \end{bmatrix} = \begin{bmatrix} -2{}^f\tilde{\omega}_e {}^f\mathbf{V} & -{}^f\tilde{\omega}_e {}^f\tilde{\omega}_e & 0 \\ \mathbb{I} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} {}^f\mathbf{V} \\ {}^f\mathbf{P} \\ {}^f\mathbf{\Phi} \end{bmatrix} + \begin{bmatrix} {}^f\mathbf{R}_b & {}^f\mathbf{R}_b & 0 \\ 0 & 0 & 0 \\ 0 & 0 & {}^f\mathbf{R}_b\mathbf{H} \end{bmatrix} \begin{bmatrix} {}^b\mathbf{g} \\ {}^b\mathbf{S} \\ {}^b\boldsymbol{\omega} \end{bmatrix} \quad (4.18)$$

If we differentiate the firts equation of the above system and taking into account that terms containing $\delta {}^f\tilde{\omega}_e$ could be omitted, since considered extremely accurate quantities, and in first approximation no error is accounted for ${}^f\mathbf{g}$, we get

$$\delta {}^f\dot{\mathbf{V}} = -2{}^f\tilde{\omega}_e \delta {}^f\mathbf{V} + -{}^f\tilde{\omega}_e {}^f\tilde{\omega}_e \delta {}^f\mathbf{P} + \delta ({}^f\mathbf{R}_b {}^b\mathbf{S}) \quad (4.19)$$

where [A.B.Chatfield, 1997]

$$\delta ({}^f\mathbf{R}_b {}^b\mathbf{S}) = -{}^f\mathbf{R}_i \delta {}^i\tilde{\Phi} {}^i\mathbf{R}_b {}^b\mathbf{S} \quad (4.20)$$

and

$$\delta {}^i\tilde{\Phi} = \begin{bmatrix} 0 & -\delta\phi & \delta\theta \\ \delta\phi & 0 & -\delta\psi \\ -\delta\theta & \delta\psi & 0 \end{bmatrix} \quad (4.21)$$

Now, we have

$$\begin{aligned} \delta ({}^f\mathbf{R}_b {}^b\mathbf{S}) &= -{}^f\mathbf{R}_i \delta {}^i\tilde{\Phi} {}^i\mathbf{R}_b {}^b\mathbf{S} \\ &= -{}^f\mathbf{R}_i \delta \mathbf{\Phi} \times {}^i\mathbf{R}_b {}^b\mathbf{S} \\ &= -{}^f\delta \mathbf{\Phi} \times {}^f\mathbf{S} \\ &= {}^f\tilde{\mathbf{S}} {}^f\delta \mathbf{\Phi} \end{aligned} \quad (4.22)$$

where the tilde $\tilde{S}$ indicate the skew operator applied to S , and finally

$$\delta^f \dot{V} = -2 {}^f \tilde{\omega}_e \delta^f V + - {}^f \tilde{\omega}_e {}^f \tilde{\omega}_e \delta^f P + {}^f \tilde{S} \delta^f \Phi + {}^f R_b \delta^b S \quad (4.23)$$

second equation of (4.18) is trivial. For the third equation we use the fact that [A.B.Chatfield, 1997]

$$\delta^i \dot{\Phi} \approx {}^i \tilde{\omega}_{b,i} \delta^i \Phi - \delta^i \omega_{b,i} \quad (4.24)$$

and so premultiplying for ${}^f R_i$ we obtain

$$\begin{aligned} \delta^f \dot{\Phi} &\approx {}^f \tilde{\omega}_{b,i} \delta^f \Phi - \delta^f \omega_{b,i} \\ &= {}^f \tilde{\omega}_{b,i} \delta^f \Phi - {}^f R_b \delta^b \omega_{b,i} \end{aligned} \quad (4.25)$$

If now consider a linear trend for accelerometer bias and for gyros' drift with a gaussian noise w_S, w_ω added with we can write the linear model

$$\begin{bmatrix} \delta^f \dot{V} \\ \delta^f \dot{P} \\ \delta^f \dot{\Phi} \\ \delta^b \dot{S} \\ \delta^b \dot{\omega} \end{bmatrix} = \begin{bmatrix} -2 {}^f \tilde{\omega}_e {}^f V & - {}^f \tilde{\omega}_e {}^f \tilde{\omega}_e & {}^f \tilde{S} & {}^f R_b & 0 \\ \mathbb{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & {}^f \tilde{\omega}_{b,i} & 0 & - {}^f R_b \\ \vdots & & \dots & 0 & \vdots \\ 0 & & \ddots & & 0 \end{bmatrix} \begin{bmatrix} \delta^f V \\ \delta^f P \\ \delta^f \Phi \\ \delta^b S \\ \delta^b \omega \end{bmatrix} + B \begin{bmatrix} 0 \\ 0 \\ 0 \\ w_S \\ w_\omega \end{bmatrix} \quad (4.26)$$

and

$$B = \begin{bmatrix} \mathbf{0}_{9 \times 9} & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{9 \times 9} & \mathbf{1}_{6 \times 6} \end{bmatrix} \quad (4.27)$$

Using the body frame mechanization introduced in §4.1.4 assuming ${}^s r = \mathbf{0}$ is easy to prove that the model could be represented

$$\begin{bmatrix} \delta^b \dot{V} \\ \delta^b \dot{P} \\ \delta^b \dot{\Phi} \\ \delta^b \dot{S} \\ \delta^b \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\partial_v ({}^b \tilde{\omega} \cdot {}^b v) & \mathbf{0} & \partial_\Phi ({}^b R_i {}^i g) & \mathbf{0} & -\partial_\omega ({}^b \tilde{\omega} \cdot {}^b v) \\ \mathbb{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & \partial_\Phi (H \cdot {}^b \omega) & 0 & \partial_\omega (H \cdot {}^b \omega) \\ \vdots & & \dots & 0 & \vdots \\ 0 & & \ddots & & 0 \end{bmatrix} \begin{bmatrix} \delta^b V \\ \delta^b P \\ \delta^b \Phi \\ \delta^b S \\ \delta^b \omega \end{bmatrix} + B \begin{bmatrix} 0 \\ 0 \\ 0 \\ w_S \\ w_\omega \end{bmatrix} \quad (4.28)$$

and

$$B = \begin{bmatrix} \mathbf{0}_{9 \times 9} & \mathbf{0}_{6 \times 6} \\ \mathbf{0}_{9 \times 9} & \mathbf{1}_{6 \times 6} \end{bmatrix} \quad (4.29)$$

The derivative terms are easily determined (see Appendix B)

This model is quite simple but functional, other models could have states with much more parameters as in [M.S.Grewal et al., 1990].

However the model derived above is been used in SLAM techniques where the linear Kalman filter (indirect appropach) is implemented [J.Kim and S. Sukkarieh, 2007].

4.2 Vision Navigation System

4.2.1 Pinhole Camera Image Model

If a point P has coordinate ${}^c\mathbf{P} = [{}^cX, {}^cY, {}^cZ]^T$ relative to a camera reference frame Σ_c centered at the optical center, with its z-axis being the optical axis (of the lens) then we have that the coordinates of P and its representation in the image plane $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$ are related by *the ideal-perspective-projection*

$$x = -f \frac{{}^cX}{{}^cZ} \quad y = -f \frac{{}^cY}{{}^cZ} \quad (4.30)$$

where f is referred to a *focal lenght*; the projection is a map

$$\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2; \quad \mathbf{X} \rightarrow \mathbf{x}$$

This imaging model is called an *ideal pinhole camera model*. It's an idealization of the thin lens model, since when the aperture decreases, diffraction effects become dominant and therefore the thin lens model does not hold.

To eliminate the negative sign in (4.30) that makes the image of an object appear to be upside-down, we can flip the image, corresponding of placing the image plane in front of the optical center.

Let us consider now a generic point ${}^w\mathbf{P} = [{}^wX, {}^wY, {}^wZ]^T \in \mathbb{R}^3$ relative to the *World-reference-frame* (WRF), we could write the coordinate transformation to the camera-frame Σ_c with a rigid body transformation

$${}^c\mathbf{P} = {}^c\mathbf{T}_w {}^w\mathbf{P}$$

where

$${}^c\mathbf{T}_w = \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{P}_w \\ 0 & 1 \end{bmatrix}$$

is the roto-traslation operator³.

So adopting the pinhole camera model in homogeneous coordinate we get with some algebra the geometric model of an ideal camera

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{P}_w \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^w\mathbf{P} \\ 1 \end{bmatrix} \quad (4.31)$$

of in matrix form

$$\lambda \mathbf{x} = K_f \Pi_0 {}^c\mathbf{P} = K_f \Pi_0 {}^c\mathbf{T}_w {}^w\mathbf{P}^+ \quad (4.32)$$

If the focal lenght is known and hence can be normalized to 1, this model reduces to

$$\lambda \mathbf{x} = \Pi_0 \mathbf{X} = \Pi_0 {}^c\mathbf{T}_w {}^w\mathbf{P}^+ \quad (4.33)$$

where the matrix Π_0 is the *standard or canonical projection matrix*. The ideal model of equation (4.32) is specified relative to a particular choice of reference frame, the *canonical retinal frame*, centered at the optical center with one axis aligned with the optical axis. In practice, when one captures image the measurement are obtained in terms of pixel (i, j) , with the origin of the image coordinate frame typically in the upper-left corner of the image.

So the image coordinate could be obtained in the new frame by

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4.34)$$

³ ${}^c\mathbf{P}_w$ if the radius vector between the origins of the two frame expressed in the frame Σ_c

where s_x, s_y are scaling factors and represent the pixel dimension in metric units, o_x, o_y is the center of the retinal frame in the pixel frame. If pixels are not rectangular a more general form of the *scaling matrix* can be considered

$$\begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

Now, combining the projection model with the scaling and translation, we obtain a more realistic model of a transformation between homogeneous coordinate of a 3-D point relative to a camera frame and homogeneous coordinate of its image expressed in terms of pixel

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & s_\theta & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^c\mathbf{P} \\ 1 \end{bmatrix} \quad (4.35)$$

or in matrix form

$$\lambda \mathbf{x}' = K_s K_f \Pi_0 {}^c\mathbf{P} = K \Pi_0 {}^c\mathbf{P}^+ \quad (4.36)$$

where

$$K = \begin{bmatrix} f s_x & f s_\theta & o_x \\ 0 & f s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

is the *intrinsic parameter matrix* or *calibration matrix* and depends to a particular camera. The entries of the matrix K have the following geometric interpretation

- o_x : x-coordinate of the principal point in pixels
- o_y : y-coordinate of the principal point in pixels
- $f s_x = \alpha_x$: size of unit length in orizontal pixels
- $f s_y = \alpha_y$: size of unit length in vertical pixels
- α_x/α_y : aspect ratio
- $f s_\theta$: skew of the pixel, often close to zero

When the calibration matrix K is known, the *calibrated* coordinates $\mathbf{x}$ can be obtained from the pixel coordinate $\mathbf{x}'$ by a simple inversion of K

$$\lambda \mathbf{x} = \lambda K^{-1} \mathbf{x}' = \Pi_0 {}^c\mathbf{P}^+ \quad (4.37)$$

Camera frame and Wold frame transformation is characterized by a rigid body motion $g : (R, T)$ where R, T are called *extrinsic calibration parameter*. The overall model for image formation is captured by the following equation

$$\lambda \mathbf{x}' = K \Pi_0 {}^c\mathbf{T}_w {}^w\mathbf{P}^+ \quad (4.38)$$

4.2.2 Camera Calibration

Camera calibration determines the camera model that defines the image formation geometry between 3-D coordinates of a point in the scene and its corresponding 2-D coordinates in the camera image. Vision researchers have traditionally used fiducials with known 3-D geometry in some external reference frame.

Methods for calibrating the camera are:

- a) *Photogrammetric calibration*: calibration is performed by observing a calibration object whose geometry in 3-D space is known with very good precision. The calibration object usually consists of two or three planes orthogonal to each other. These approaches require an expensive calibration apparatus, and an elaborate setup.
- b) *Self-calibration*: techniques in this category do not use any calibration object and the metric properties of the camera and of the imaged scene are recovered from a set of uncalibrated images. Just by moving a camera in a static scene, the rigidity of the scene provides in general two constraints on the cameras' internal parameters from one camera displacement by using image information alone. Therefore, if images are taken by the same camera with fixed internal parameters, correspondences between three images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3-D structure up to a similarity.

In general three types of constraints are applied to perform self-calibration: scene constraints, camera motion constraint or constraints in camera intrinsic parameters.

- c) *Image Array Calibration*: it's possible to proceed with traditional camera calibration methods based on images of a known object point array. Common methods adopted are [Tsai, 1987], [J. Hekkila and O. Silven, 1997] and [Zhang, 2000]. These are based on the pinhole camera model and include terms for modelling radial distortion.

Using these methods we suppose that, after calibration on ground, intrinsic parameters of the camera will not change during the flight; this strategy was applied successfully to MAST camera for Mars Rover Missions [Won S. Kim et al., 2005].

4.2.3 Stereo Camera Error Modeling

Now it will be described how to model errors on a vision system which use a stereo camera to get the tridimensional information of a feature. The error modeling presented here is based on the pin-hole camera model presented above.

The geometry of stereo triangulation is shown in figure (4.4) for the case of 2D points projecting onto one dimensional image. Suppose point P projects onto the left image at x_l in the image plane and the right image at x_r . Because of errors in the measurement, the stereo system will determine x_l and x_r with some error which in turn causes error in the estimated location $P(X, Y, Z)$ of the feature.

From pin-hole model we have

$$x_l = f \frac{X}{Z} \quad (4.39)$$

and

$$x_r = f \frac{X - b}{Z} \quad (4.40)$$

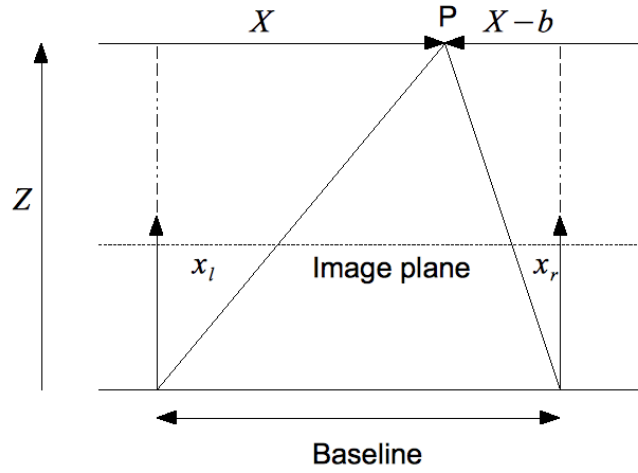


Figure 4.4: Stereo Camera Model

where f is the focal length and b is the baseline of the stereo system.

From the above equation we can find the expression for X and Z , we get

$$X = \frac{bx_l}{x_l - x_r} \quad (4.41)$$

$$Z = \frac{bf}{x_l - x_r} \quad (4.42)$$

Now remember that for a function of a random variable $\mathbf{y} = f(\mathbf{x})$ we have

$$\Sigma_{\mathbf{y}} = \nabla_{\mathbf{x}} \Sigma_{\mathbf{x}} \nabla_{\mathbf{x}}^T \quad (4.43)$$

for the covariace of $\mathbf{y}$, we can get the espression for covariance of X and Z

$$\Sigma_X = \begin{bmatrix} \partial_{x_l} X & \partial_{x_r} X \end{bmatrix} \begin{bmatrix} \Sigma_x & 0 \\ 0 & \Sigma_x \end{bmatrix} \begin{bmatrix} \partial_{x_l} X \\ \partial_{x_r} X \end{bmatrix} \quad (4.44)$$

$$\Sigma_Z = \begin{bmatrix} \partial_{x_l} Z & \partial_{x_r} Z \end{bmatrix} \begin{bmatrix} \Sigma_x & 0 \\ 0 & \Sigma_x \end{bmatrix} \begin{bmatrix} \partial_{x_l} Z \\ \partial_{x_r} Z \end{bmatrix} \quad (4.45)$$

$$(4.46)$$

where $\sigma_{x_l}^2 = \sigma_{x_r}^2 = \Sigma_x$. The expression for covariances are

$$\Sigma_X = \sigma_x^2 b^2 \frac{x_l^2 + x_r^2}{(x_l - x_r)^4} \quad (4.47)$$

$$\Sigma_Z = 2\sigma_x^2 b^2 f^2 \frac{1}{(x_l - x_r)^4} \quad (4.48)$$

$$(4.49)$$

To compute the covariance is convenient to consider the image at bounds on FOV, so

$$x_l = \frac{1}{2} \epsilon_x [m] \quad (4.50)$$

$$X_l = \frac{dx_l}{f} [m] \quad (4.51)$$

$$X_r = X_l - b \quad (4.52)$$

$$x_r = \frac{fX_r}{d} \quad (4.53)$$

where ϵ_x is the CCD size along x , d the distance of the system from the target.

4.2.4 Correspondence Problem

When a landmark is observed we have the necessity to determine the correspondence of adjacent pixels in a window (matching points is carried out by matching windows) in two different images in order to track a specific *feature*.

Problem is not simple because both the window shape and the image values undergo transformations as a consequence of the change in viewpoint, and the image intensity is corrupted by additive noise.

It's important to underline what happens when the value of the image at each pixel is constant across different regions, it's not possible to tell exactly which one is the corresponding region. This is the well known *aperture problem*, which occurs when the brightness profile within a selected region is not rich enough to allow us to recover the chosen transformation uniquely.

It will be wise then to restrict our attention only to those regions for which the correspondence problem can be solved.

Those regions will be called *features* and they establish the link between photometric measurement and the geometric primitives.

Suppose that I_1 and I_2 are images of the same scene, and they satisfy the irradiance equation

$$I_2(\mathbf{x}_2) = I_1(\mathbf{x}_1) \mathcal{R}(p)$$

where we consider the radiance distribution $\mathcal{R} : \mathbb{R}^3 \rightarrow \mathbb{R}_+$ of a visible surface in its value in the point p .

Under this restriction the correspondence (or matching) problem consists in establishing the relationship between $\mathbf{x}_1$ and $\mathbf{x}_2$, verifying that the two points are indeed images of the same 3-D point.

Supposing that the displacement between the two camera view points is a rigid body motion, there will be correspondence if

$$\mathbf{x}_2 \lambda_2({}^c\mathbf{P}) = [{}^c\mathbf{R}_w \ {}^c\mathbf{P}_w] \begin{bmatrix} \lambda_1({}^c\mathbf{P})\mathbf{x}_1 \\ 1 \end{bmatrix}$$

Therefore a model for deformation between two images of the same scene is given by an image matching constraint

$$I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_2)) \quad \forall \mathbf{x}_1 \in \Omega \cap h^{-1}(\Omega) \subset \mathbb{R}^2$$

This equation is called the *brightness constancy constraint*, since it expresses the fact that given a point on an image, there exists a different (transformed) point in another image that has the same brightness, the function h describes the transformation of the domain.

The constraint that all the scene is Lambertian is extremely unreal, so we concentrate on choosing a class of simple transformations and then restrict them to an image's regions. Such transformations occur in the domain window $W(\mathbf{x})$ around $\mathbf{x}$. We indicate $I(\tilde{\mathbf{x}})$ the intensity value, $\tilde{\mathbf{x}} \in W(\mathbf{x})$.

4.2.5 Basic Matching Strategies

The unrealistic assumption that each point in the 3-D space results in two different images with the same measured irradiance must be revisited.

As a first approximation one could lump all sources of uncertainty into an additive noise term ν . So we can write the correspondence equation as

$$I_1(\mathbf{x}_1) = I_2(h(\mathbf{x}_2)) + \nu(h(\mathbf{x}_1)) \quad (4.54)$$

therefore we formulate correspondence as the solution of an optimization problem. We chose a class of transformations and we look for the particular one, say $\hat{h}$ that minimize the effect of noise

$$\hat{h} = \arg \min_h \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} \|n(\tilde{\mathbf{x}})\|^2$$

or

$$\hat{h} = \arg \min_h \sum_{\tilde{\mathbf{x}} \in W(\mathbf{x})} \|I_1(\tilde{\mathbf{x}}_1) - I_2(h(\tilde{\mathbf{x}}_2))\|^2 \quad (4.55)$$

We define a *point feature* if there exists a neighborhood $W(\mathbf{x})$ such that the equations

$$I_1(\tilde{\mathbf{x}}) = I_2(h(\tilde{\mathbf{x}}, \alpha)) \quad \forall \tilde{\mathbf{x}} \in W(\mathbf{x}) \quad (4.56)$$

uniquely determine the parameters α which is the type of transformation in the image domain (e.g translations, affine motion model, projective motion model). The aperture problem applied to this formulation is that any h would solve the minimization.

Consider now a simple translational motion in (4.56)

$$h(\mathbf{x}) = \mathbf{x} + \Delta \mathbf{x}$$

if we have two images taken from infinitesimally close vantage point, defining $\Delta \mathbf{x} = \mathbf{u} dt$ applying a Taylor series expansion to the right hand side of (4.56) and some algebra we get

$$\nabla I^T \mathbf{u} + I_t = 0 \quad (4.57)$$

where

$$\nabla I^T = \begin{bmatrix} I_x(\mathbf{x}, t) \\ I_y(\mathbf{x}, t) \end{bmatrix} \in \mathbb{R}^2 \quad I_t = \frac{\partial I}{\partial t}(\mathbf{x}, t) \in \mathbb{R}$$

If we fix our attention at a particular image location $\hat{\mathbf{x}}$ and use (4.57) to compute the velocity $\mathbf{u}(\hat{\mathbf{x}}, t)$ of particle flowing through that pixel we talk about *optical flow*; when attention is on a particular particle instead and compute (4.57) at the location as it moves through the image domain we refer as *feature tracking*.

The equation (4.57) if applied in each single point has infinitely solutions $\mathbf{u}$, but if we apply it to each point $\tilde{\mathbf{x}}$ in a region $W(\mathbf{x})$ that contains sufficient texture and the motion $\mathbf{u}$ is assumed to be constant in this region, equation provide enough constraints on $\mathbf{u}$. This constancy assumption enables us to integrate the constraints from all point in the region and seek the best image velocity consistent with all the point constraints.

In order to account for the effect of noise in the model we minimize the following quadratic error function based on constraint equation (4.57).

$$E_b(\mathbf{u}) = \sum_{W(\mathbf{x})} [\nabla I^T \mathbf{u} + I_t]^2 \quad (4.58)$$

$\nabla E(\mathbf{u}) = 0$ when

$$G\mathbf{u} + \mathbf{b} = 0$$

the subscript "b" is for baseline, where

$$G = \begin{bmatrix} \sum I_x^2 & \sum I_x \sum I_y \\ \sum I_x \sum I_y & \sum I_y^2 \end{bmatrix} \quad b = \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

if G is invertible we could solve (4.58), if not (intensity variation in a local image window varies only along one direction or vanishes) we have the previously mentioned *aperture problem*.

So a basic tracking algorithm (small baseline) is, given an image at time t :

- set a window W
- compute G in every pixel of W
- select a number of point features by choosing $\mathbf{x}_1, \dots, \mathbf{x}_k$ such that $G(\mathbf{x}_k)$ is invertible
- compute $\mathbf{b}$
- compute $\mathbf{u}$
- at time $t + 1$ repeat the operation at point $\mathbf{x} + \mathbf{u}$

The previous algorithm results in very efficient and fast implementation, however when features are tracked over an extended time period (this could happen for many architecture and hardware solution) the estimation error resulting from matching templates between two adjacent frames accumulates in time.

The deformation of the image regions between the first frame and the current can no longer be modelled by a simple translational model. Instead an affine model deformation is used

$$h(\tilde{\mathbf{x}}) = A\tilde{\mathbf{x}} + \mathbf{d}$$

and the brightness constraint become

$$I_1(\tilde{\mathbf{x}}) = I_2(A\tilde{\mathbf{x}} + \mathbf{d})$$

We can estimate now the unknown affine parameters $A, \mathbf{d}$ by integrating the above constraint for all the points in the region $W(\mathbf{x})$

$$E_a(A, \mathbf{d}) = \sum_{W(\mathbf{x})} [I_2(A\tilde{\mathbf{x}} + \mathbf{d}) - I_1(\tilde{\mathbf{x}})]^2 \quad (4.59)$$

The above minimization problem can be resolved using linear least-squares directly from measurement of spatial and temporal gradients

Another matching strategy is the *Template Matching* in which we use the *sum of squared differences* (SSD) criterion to measure the similarity of a window $W(\mathbf{x})$ taken at time t and $t + dt$ is . The SSD approach consider an image window $W(\mathbf{x})$ centered at the location (x, y) at time t and another candidate location $(x + dx, y + dy)$ at time $t + dt$; the goal is to find a displacement $\Delta x = (dx, dy)$ at a location in the image (x, y) that minimizes the SSD criterion

$$E_t(dx, dy) = \sum_{W(\mathbf{x})} [I(x + dx, y + dy, t + dt) - I(x, y, t)]^2 \quad (4.60)$$

The main advantage of SSD is that it does not require computation of image gradients.

On the other hand SSD is not invariant to scalings and shift in image intensity, often caused by changing lighting conditions over time. For the purpose of template matching a better choice is

the *normalized cross-correlation*. Given two non uniform image regions $I_1(\tilde{\mathbf{x}})$ and $I_2(h(\tilde{\mathbf{x}}))$ with $\tilde{\mathbf{x}} \in W(\mathbf{x})$ and $N = |W(\mathbf{x})|$ the number of pixel in region we have[Y.Ma et al., 2006]

$$NCC(h) = \frac{\sum_{W(\mathbf{x})} (I_1(\tilde{\mathbf{x}}) - \hat{I}_1)(I_2(h(\tilde{\mathbf{x}})) - \hat{I}_2)}{(\sum_{W(\mathbf{x})} (I_1(\tilde{\mathbf{x}}) - \hat{I}_1)^2 \sum_{W(\mathbf{x})} (I_2(h(\tilde{\mathbf{x}})) - \hat{I}_2)^{1/2})} \quad (4.61)$$

where

$$\hat{I}_1 = \frac{1}{N} \sum_{W(\mathbf{x})} I_1(\tilde{\mathbf{x}})$$

$$\hat{I}_2 = \frac{1}{N} \sum_{W(\mathbf{x})} I_2(h(\tilde{\mathbf{x}}))$$

The NCC value ranges between $-1, 1$. When $NCC=1$ the two regions match perfectly. For point features the *nearest Neighbour association* could be used.

4.2.6 SLAM Detection and Matching Hypotesis

Even if in this work we don't include any detection and matching strategy in the navigation software, some strategy has been studied for future implementations.

The solution to the tracking or correspondance problem for the case of pure translation relied on inverting the G matrix, and it means that the image has no trivial x,y-gradients in the region considered; there is a corner structure so we can implement a *Harris corner detection* for initial feature selection; on the other hand when one of the singular values (SVD transformation of the image, so they are eigenvalue of G) is large and the other is close to zero the brightness varies mostly along a single direction (and edge feature).

In the SLAM problem the detection and successive identification of the features or *landmark* is crucial for the succesfull implementation of the algorithm.

Visual landmarks should be invariant to image translation, rotation, scaling, partial illumination changes and viewpoint changes. *Interest points*, such as detected by Harris detector has proven to have a good stability properties when the scale change is not greater that 1.5 times.

If there is a prior knowledge on the scale change, even approximate, a scale adaptive version of the base detector could be applied. When no information on scale change is available, scale adaptation is not possible. In such cases, scale invariant feature detection algorithms have recently been proposed. However, these methods generate much less features than the standard or scale adaptive detectors. Matching features in such contexts is quite time consuming, scale being an additional dimension to search through.

This work will concern itself primarily with SLAM using interpolated points as landmarks from a given DTM.

4.2.7 SLAM VNS Set Up

When dealing with vision systems for navigation, different configurations are allowed and a trade is necessary to match operations requirement.

A monocular system provides essentially a bearing measurement (angle between one reference direction and the target feature in the image plane) while range measurement could be obtained with a stereo-camera system, solving the distance from the vehicle and the feature by triangulation. In the navigation system developed both mono and stereo camera configuration have been implemented for SLAM.

In the case of loop-closure requirement we have to select the proper cameras resolution and optic

Z	SzX	ResX	SzY	ResY	ResSte	D	S_X	S_Y	S_Z	ΔT
50.00	58.9375	0.0288	70.4375	0.0288	68.0473	69.6	0.719	0.507	0.425	5.894
60.00	70.7250	0.0345	84.5250	0.0345	82.1348	58.0	1.035	0.730	0.611	7.072
70.00	82.5125	0.0403	98.6125	0.0403	96.2223	49.7	1.409	0.993	0.831	8.251
80.00	94.3000	0.0460	112.7000	0.0460	110.3098	43.5	1.840	1.297	1.085	9.430
90.00	106.0875	0.0517	126.7875	0.0517	124.3973	38.6	2.329	1.641	1.373	10.609
100.00	117.8750	0.0575	140.8750	0.0575	138.4848	34.8	2.875	2.026	1.695	11.787
110.00	129.6625	0.0633	154.9625	0.0633	152.5723	31.6	3.479	2.451	2.051	12.966
120.00	141.4500	0.0690	169.0500	0.0690	166.6598	29.0	4.140	2.917	2.441	14.145
130.00	153.2375	0.0747	183.1375	0.0747	180.7473	26.8	4.859	3.423	2.865	15.324
140.00	165.0250	0.0805	197.2250	0.0805	194.8348	24.8	5.635	3.970	3.322	16.503
150.00	176.8125	0.0862	211.3125	0.0862	208.9223	23.2	6.469	4.557	3.813	17.681

Table 4.2: Table of 2050×2450 stereo camera parameters and uncertainties.

in order to have the appropriate Fields of View (FOV) for images overlapping.

A basic application for an autonomous airship is to map an area to build a digital terrain model; if an area of [1 km×2 km] has to be mapped, a basic camera's requirement is that its footprint must permit the partial overlapping of two orizontal strips; in this way the feature could be re-observed at least one time. A possible trajecotry for this task is shown in figure 4.5.

We can suppose the airship to fly between 60 to 120 metres of altitude, using a camera with resolution of 2050 × 2450 pixel. The filter update should occur every 0.5 – 0.25 hz, an interval compatible with the image processing time by on board commerical FPGA processing unit.

A possible set-up could be:

	px	pixel size [10^{-6} m]	sensor size [mm]	FoV
sensor x	2050	3.45	7.07	61.03
sensor y	2450	3.45	8.45	70.32

- focal: 6.00 [mm]
- baseline: 2.00 [m]

Table 4.2 summarize the quantities that characterize the sensor and are indicative for SLAM appication

Z: is the altitude [m]

SzX: is the footprint on X-axis [m]

ResX: is the X-resolution on CCD [px/m]

SzY: is the footprint on Y-axis [m]

ResY: is the Y-resolution on CCD [px/m]

ResSte: is the stereo footprint along Y [m]

D: is the disparity ($x_l - x_r$)

Z	SzX	ResX	SzY	ResY	ResSte	D	S_X	S_Y	S_Z	ΔT
10.00	12.2808	0.0119	19.3256	0.0119	16.1783	168.1	0.059	0.059	0.038	1.228
20.00	24.5616	0.0238	38.6512	0.0238	35.5039	84.0	0.238	0.231	0.148	2.456
30.00	36.8424	0.0357	57.9768	0.0357	54.8295	56.0	0.535	0.519	0.331	3.684
40.00	49.1232	0.0476	77.3024	0.0476	74.1551	42.0	0.952	0.921	0.587	4.912
50.00	61.4040	0.0595	96.6280	0.0595	93.4807	33.6	1.488	1.439	0.915	6.140
60.00	73.6848	0.0714	115.9536	0.0714	112.8063	28.0	2.142	2.071	1.317	7.368
70.00	85.9656	0.0833	135.2792	0.0833	132.1319	24.0	2.916	2.818	1.792	8.597
80.00	98.2464	0.0952	154.6048	0.0952	151.4575	21.0	3.808	3.681	2.340	9.825
90.00	110.5272	0.1071	173.9304	0.1071	170.7831	18.7	4.820	4.658	2.961	11.053

Table 4.3: Table of 1032×1624 stereo camera parameters and uncertainties.

S_X, S_Y, S_Z : are the uncertainties along X,Y,Z [m]

Δ_T : is the minimum time interval to have 50% of overlapping along airship's longitudinal axis

The field of view along one direction could be easily computed by

$$FOV = \tan^{-1} \left[\frac{\epsilon_x}{2f} \right] \quad (4.62)$$

where ϵ_x is half of the sensor size in the x -direction. It could be seen from Table 4.2 that a range of different altitude are compatible with uncertainties and overlapping constraint (the footprint has to be larger than half of the distance between two consecutive paths, fig. 4.5). We can use two cameras with less resolution, with the same baseline, but obviously we have higher uncertainties in the same range of altitudes.

	px	pixel size [10^{-6} m]	sensor size [mm]	FoV [°]
sensor x	1032	7.14	7.4	63.1032
sensor y	1624	7.14	11.6	88.0350

- focal: 6.00 [mm]
- baseline: 2.00 [m]

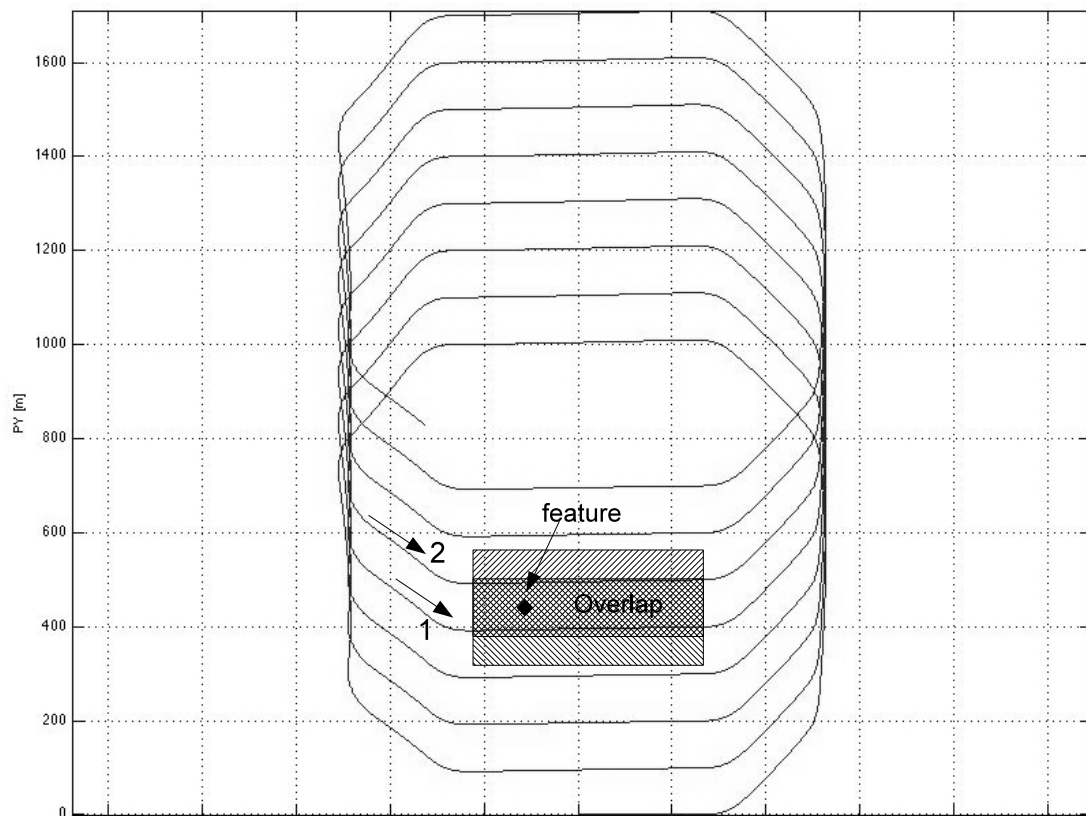


Figure 4.5: Overlapping area between two passages of the vehicle over the same strip. Trajectory 1 is the first passage, 2 the second.

Chapter 5

The SLAM Problem

The simultaneous localization and mapping (SLAM) problem asks if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map.

The “solution” of the SLAM problem has been one of the notable successes of the robotics community over the past decade. SLAM has been formulated and solved as a theoretical problem in a number of different forms. SLAM has also been implemented in a number of different domains from indoor robots to outdoor, underwater, and airborne systems.

This chapter presents the SLAM problem, in its mathematical formulation and its solution using Extended Kalman Filtering Theory. We also approach to the problem of the computational complexity of the standard algorithm giving an alternative solution, based on the *Compressed Filter*, in order to manage large map and reducing the process time. Also a SLAM algorithm for monocular camera based navigation is introduced to extend the domain of this technique beyond the knowledge, by range measurement, of the 3D relative location of a feature.

5.1 Formulation of the SLAM problem

The main advantage of SLAM is that it eliminates the need for artificial infrastructures or a priori topological knowledge of the environment. A solution to the SLAM problem would be of inestimable value in a range of applications where absolute position or precise map information is unobtainable, including, amongst others, autonomous planetary exploration, subsea autonomous vehicles, autonomous air-borne vehicles, and autonomous all-terrain vehicles in tasks such as mining and construction.

The result of years of discussion point out that the problem of SLAM had a probabilistic nature. A key element was to show that there must be a high degrees of correlation between estimates of the location of different landmarks in a map and that these correlations would grow with successive observation.

As a mobile robot moves through an unknown environment taking relative observations of landmarks, the estimates of these landmarks are all necessarily correlated with each other because of the common error in estimated vehicle location. The implication of this was profound: a consistent full solution to the combined localization and mapping problem would require a *joint state* composed of the vehicle pose and every landmark position, to be updated following each landmark observation. In turn, this would require the estimator to employ a huge state vector (on the order of the number of landmarks maintained in the map) with computation

scaling as the square of the number of landmarks.

The conceptual breakthrough came with the realization that the combined mapping and localization problem, once formulated as a single estimation problem, was actually convergent. Most importantly, it was recognized that the correlations between landmarks, which most researchers had tried to minimize, were actually the critical part of the problem and that, on the contrary, the more these correlations grew, the better the solution!

5.1.1 Probabilistic Nature of SLAM

Consider a vehicle moving through an environment taking relative observations of a number of unknown landmarks using a sensor located on the vehicle as a stereo camera.

At a time instant k , the following quantities are defined:

- $\mathbf{x}_k$: the state vector describing the location and orientation of the vehicle
- $\mathbf{u}_k$: the input vector, applied at time $k - 1$ to drive the vehicle to a state $\mathbf{x}_k$ at time k
- $\mathbf{m}_i$: a vector describing the location of the i -th landmark whose true location is assumed *time invariant*
- $\mathbf{z}_{ik}$: an observation taken from the vehicle of the location of the i th landmark at time k . When there are multiple landmark observations at any one time or when the specific landmark is not relevant to the discussion, the observation will be written simply as $\mathbf{z}_k$
- $\mathbf{X}_{0:k} = [\mathbf{x}_0, \dots, \mathbf{x}_k] = [\mathbf{X}_{0:k-1}, \mathbf{x}_k]$: history of vehicle locations
- $\mathbf{U}_{0:k} = [\mathbf{u}_0, \dots, \mathbf{u}_k] = [\mathbf{U}_{0:k-1}, \mathbf{u}_k]$: history of control input
- $\mathbf{m} = [m_1, \dots, m_k]$: set of all landmarks
- $\mathbf{Z}_{0:k} = [\mathbf{z}_0, \dots, \mathbf{z}_k] = [\mathbf{Z}_{0:k-1}, \mathbf{z}_k]$: set of all landmark observations

In the probabilistic form, the SLAM building problem requires that the probability distribution density

$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$$

be computed for all times k . This probability distribution density describes the joint posterior density of the landmark locations and vehicle state (at time k) given the recorded observations and control inputs up to and including time k together with the initial state of the vehicle.

In general, a *recursive* solution to the SLAM problem is desirable. Starting with an estimate for the distribution $p(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{x}_0)$ at time $k - 1$, the joint posterior, following a control $\mathbf{u}_k$ and observation $\mathbf{z}_k$, is computed using *Bayes theorem*. This computation requires that a *state transition model* and an *observation model* are defined describing the effect of the control input and output of the system respectively.

The *observation model* describes the probability of making an observation $\mathbf{z}_k$ when the vehicle location and landmark locations are known and is generally described in the form

$$p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})$$

It is reasonable to assume that once the vehicle location and map are defined, observations are conditionally independent given the map and the current vehicle state.

The *dynamic model* for the vehicle can be described in terms of a probability distribution on state transitions in the form

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$$

That is, the state transition is assumed to be a *Markov process* in which the next state $\mathbf{x}_k$ depends only on the immediately preceding state $\mathbf{x}_{k-1}$ and the applied control $\mathbf{u}_k$ and is independent of both the observations and the map. The propagation of the vehicle dynamics could be done by IMU data or by dynamical model.

The SLAM algorithm is implemented in a standard two-step recursive *prediction* (time-update) and *correction* (measurement-update):

Prediction

$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \cdot p(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) d\mathbf{x}_{k-1} \quad (5.1)$$

Update is made by Bayes theorem; set

$$p_{\mathbf{x}}(x) = (x_1, \dots, x_n) \quad (5.2)$$

as the prior probability density of $\mathbf{x}$.

We suppose known the join density of $\mathbf{y}$ given the *value* x of a stochastic variable $\mathbf{x}$, $f(y | \mathbf{x} = x) = f(y | x)$

$$p(y | x) = \frac{P(y \leq \mathbf{y} \leq y + dy | \mathbf{x} = x)}{dy} \quad (5.3)$$

the function $p(y | x)$ is the mathematic description of measure instrument.

The *Bayesian estimate problem* is estimate the stochastic vector $\mathbf{x}$ on the base of knowledge of the probabilistic model (5.2) and (5.3) and by observation vector $\mathbf{y} = y$; in other words it is the problem of computing the new probability density of $\mathbf{x}$ by observation of $\mathbf{y} = y$.

If $p_{\mathbf{x}}$ is known the problem has the solution

$$p(x | \mathbf{y} = y) = \frac{p(y | x)p_{\mathbf{x}}(x)}{\int_{\mathbb{R}^n} p(y | x)p_{\mathbf{x}}(x)dx} \quad (5.4)$$

$$= \frac{p(y | x)p_{\mathbf{x}}(x)}{p_{\mathbf{y}}(y)} \quad (5.5)$$

which is the description of how much the observation of $\mathbf{y} = y$ change our priori knowledge of $\mathbf{x}$, given by da $p_{\mathbf{x}}(x)$. Equation (5.4) is the *posterior probability density* of x .

So we have for the SLAM update

$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \frac{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \cdot p(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0)}{\int p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \cdot p(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) d\mathbf{x}_{k-1}} \quad (5.6)$$

where¹

$$\int p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \cdot p(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) d\mathbf{x}_{k-1} = p_{\mathbf{z}}(\mathbf{z}_k) \quad (5.7)$$

It is worth noting that the **map building** problem may be formulated as computing the conditional density

$$p(\mathbf{m} | \mathbf{X}_{0:k}, \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}) \quad (5.8)$$

¹With $p_{\alpha(\alpha_k)}$, we intend the multivariate probability function of the stochastic variable variable α evaluated in the vector α_k

This assumes that the location of the vehicle $\mathbf{x}_k$ is known (or at least deterministic) at all times, subject to knowledge of initial location. A map $\mathbf{m}$ is then constructed by fusing observations from different locations. Conversely, the **localization** problem may be formulated as computing the probability distribution

$$p(\mathbf{x}_k | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{m})$$

This assumes that the landmark locations are known with certainty, and the objective is to compute an estimate of vehicle location with respect to these landmarks.

It's important to underline that the observation model $p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})$ makes explicit the dependence of observations on both the vehicle and landmark locations. It follows that the joint posterior cannot be partitioned in the obvious manner

$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{z}_k) \neq P(\mathbf{x}_k | \mathbf{z}_k) P(\mathbf{m} | \mathbf{z}_k)$$

Referring to Figure ?? , it can be seen that much of the error between estimated and true landmark locations is common between landmarks and is in fact due to a single source; errors in knowledge of where the robot is when landmark observations are made. In turn, this implies that the errors in landmark location estimates are highly correlated.

5.1.2 SLAM Solution Requirements

The most important insight in SLAM was to realize that the correlations between landmark estimates increase monotonically as more and more observations are made.

Practically, this means that knowledge of the relative location of landmarks always improves and never diverges, regardless of vehicle motion. In probabilistic terms, this means that the joint probability density on all landmarks $p(\mathbf{m})$ becomes monotonically more peaked as more observations are made.

Consider the vehicle at location $\mathbf{x}_k$ observing the two landmarks $\mathbf{m}_i$ and $\mathbf{m}_j$. The relative location of observed landmarks is clearly independent of the coordinate frame of the vehicle, and successive observations from this fixed location would yield further independent measurements of the relative relationship between landmarks. As the vehicle moves to location $\mathbf{x}_{k+1}$, it again observes landmark $\mathbf{m}_j$; this allows the estimated location of the robot and landmark ($\mathbf{m}_i$) to be updated relative to the previous location $\mathbf{x}_k$ (fig. 5.1). In turn, this propagates back to update landmark $\mathbf{m}_i$ even though this landmark is not seen from the new location. This occurs because the two landmarks are highly correlated (their relative location is well known) from previous measurements. Further, the fact that the same measurement data is used to update these two landmarks makes them more correlated.

Note that in Fig.5.1 at location $\mathbf{x}_{k+1}$, the robot observes two new landmarks relative to $\mathbf{m}_j$. These new landmarks are thus immediately linked or correlated to the rest of the map. Later updates to these landmarks will also update landmark $\mathbf{m}_j$ and through this landmark $\mathbf{m}_i$ and so on. That is, all landmarks end up forming a network linked by relative location or correlations whose precision or value increases whenever an observation is made.

This process can be visualized (Fig 5.2) as a network of springs connecting all landmarks together or as a rubber sheet in which all landmarks are embedded. An observation in a neighborhood acts like a displacement to a spring system or rubber sheet such that its effect is great in the neighborhood and, dependent on local stiffness (correlation) properties, diminishes with distance to other landmarks.

On other words, the landmarks are connected by springs describing correlations between landmarks. As the vehicle moves back and forth through the environment, spring stiffness or corre-

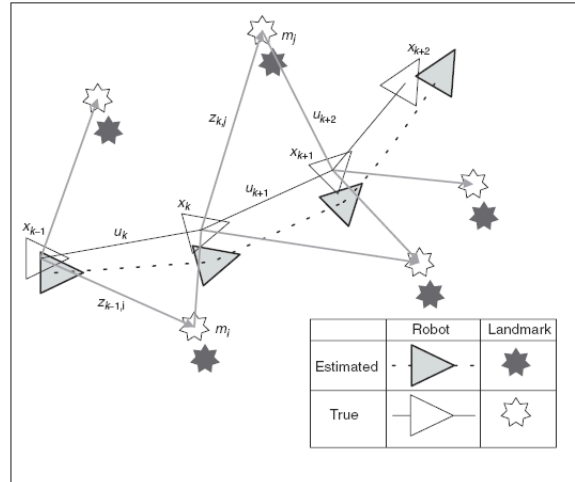


Figure 5.1: Essential SLAM problem. A simultaneous estimate of both robot and landmark locations is required [T. Bailey and H.F. Durant-Whyte, 2006].

lations increase (red links become thicker). As landmarks are observed and estimated locations are corrected, these changes are propagated through the spring network.

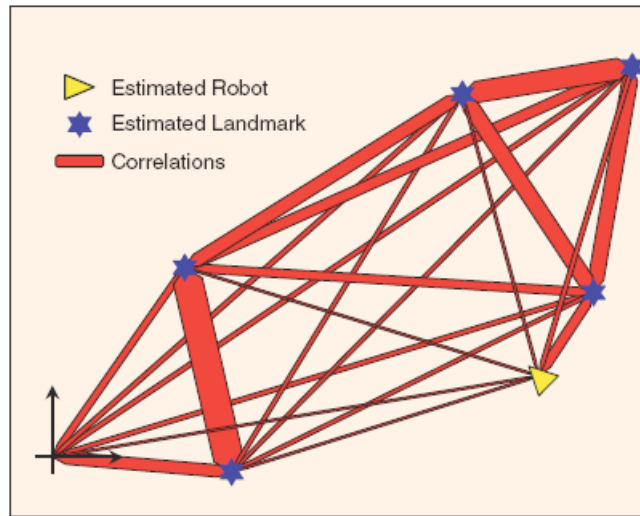


Figure 5.2: Spring network analogy [T. Bailey and H.F. Durant-Whyte, 2006].

The solution of the SLAM problem is based on the following remarkable results

- 1) The determinant of any submatrix of the map covariance matrix decreases monotonically as observations are successively made.
- 2) In the limit as the number of observations increases, the landmark estimates become fully correlated.
- 3) In the limit, the covariance associated with any single landmark location estimate is determined only by the initial covariance in the vehicle location estimate.

These three results describe, in full, the convergence properties of the map and its steady state behavior. In particular they show the following.

- The entire structure of the SLAM problem critically depends on maintaining complete knowledge of the cross correlation between landmark estimates. Minimizing or ignoring cross correlations is precisely contrary to the structure of the problem
- As the vehicle progresses through the environment the errors in the estimates of any pair of landmarks become more and more correlated, and indeed never become less correlated.
- In the limit, the errors in the estimates of any pair of landmarks becomes fully correlated. This means that given the exact location of any one landmark, the location of any other landmark in the map can also be determined.
- As the vehicle moves through the environment taking observations of individual landmarks, the error in the estimates of the relative location between different landmarks reduces monotonically to the point where the map of relative locations is known with absolute certainty.
- As the map converges in the above manner, the error in the absolute location of every landmark (and thus the whole map) reaches a lower bound determined only by the error that existed when the first observation was made.

Thus a solution [M.W.M.Dissanayake et al., 2001] to the general SLAM problem exists and it is indeed possible to construct a perfectly accurate map and simultaneously compute vehicle position estimates without any prior knowledge of vehicle or landmark locations.

5.2 Non-Linear Systems

Many dynamic systems and sensors are not absolutely linear, but they are not far from it. Following the considerable success enjoyed by linear estimation methods on linear problems, extension of these methods were applied to such nonlinear problems.

At the base of this extension is that even though measurement or state dynamics are non linear, they could be linear for small perturbations in the values of the state variables. So methods of linear estimation theory can be applied to such non linear problems by linear approximation of the effects of small in the state of the non linear system from a "nominal" value.

For some problems, the nominal values of the state variables are fairly well known beforehand. These include guidance and control applications for which operational performance depends on staying close to an optimal trajectory. For these applications, the estimation problem can often be effectively linearized about the nominal trajectory and the Kalman gains can be precomputed to relieve the real-time computational burden.

The nominal trajectory can also be defined "on the fly" as the current best estimate of the actual trajectory. This approach is called *extended Kalman filtering*. It has the advantage that the perturbations include only the state estimation errors, which are generally smaller than the perturbations from any predefined nominal trajectory and therefore better conditioned for linear approximation. The major disadvantage of extended Kalman filtering is the added real-time computational cost of linearization about an unpredictable trajectory, for which the Kalman gains cannot be computed beforehand.

Now suppose to have a non linear continuous-time non linear system

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \quad (5.9)$$

where $\mathbf{x}$ is the state vector and $\mathbf{u}$ is the inputs vector.

A way to transform this system in a discrete-time one is using *Euler step forward* method, from which we could write

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{x}(x) + \mathbf{f}(t, \mathbf{x}(k), \mathbf{u}(k))\Delta t + O(\Delta t^2) \\ &\simeq \mathbf{x}(x) + \mathbf{f}(t, \mathbf{x}(k), \mathbf{u}(k))\Delta t \\ &= \mathbf{F}(t, \mathbf{x}(k), \mathbf{u}(k)) = \mathbf{F}_k\end{aligned}\quad (5.10)$$

It's easy to compute the jacobian with respect to $\mathbf{x}$ and $\mathbf{u}$, we obtain

$$\nabla_{\mathbf{x}}\mathbf{F}_k = \mathbb{I} + \nabla_{\mathbf{x}}\mathbf{f}(t, \mathbf{x}(k), \mathbf{u}(k))\Delta t \quad (5.11)$$

$$\nabla_{\mathbf{u}}\mathbf{F}_k = \nabla_{\mathbf{u}}\mathbf{f}(t, \mathbf{x}(k), \mathbf{u}(k))\Delta t \quad (5.12)$$

We'll show now the important results that the state transition function $\Phi(k+1|k)$ of the linearized system is approximable with the jacobian of $\mathbf{F}(t, \mathbf{x}(k))$.

Expanding $\mathbf{F}_k$ in Taylor series we get around a nominal trajectory $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ in the variables space we get

$$\begin{aligned}\mathbf{F}_k(\mathbf{x}_k, \mathbf{u}_k) &= \mathbf{F}_k(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) + \nabla_{\mathbf{x}}\mathbf{F}_k\Big|_{\bar{\mathbf{x}}_k} (\mathbf{x}_k - \bar{\mathbf{x}}_k) + \nabla_{\mathbf{u}}\mathbf{F}_k\Big|_{\bar{\mathbf{u}}_k} (\mathbf{u}_k - \bar{\mathbf{u}}_k) + O(2) \\ &\approx \mathbf{F}_k(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) - \nabla_{\mathbf{x}}\mathbf{F}_k\Big|_{\bar{\mathbf{x}}_k} \bar{\mathbf{x}}_k - \nabla_{\mathbf{u}}\mathbf{F}_k\Big|_{\bar{\mathbf{u}}_k} \bar{\mathbf{u}}_k + \nabla_{\mathbf{x}}\mathbf{F}_k\Big|_{\bar{\mathbf{x}}_k} \mathbf{x}_k + \nabla_{\mathbf{u}}\mathbf{F}_k\Big|_{\bar{\mathbf{u}}_k} \mathbf{u}_k \\ &= \mathbf{L}_k + [\nabla_{\mathbf{x}}\mathbf{F}_k \ \nabla_{\mathbf{u}}\mathbf{F}_k] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \\ &\approx [\nabla_{\mathbf{x}}\mathbf{F}_k \ \nabla_{\mathbf{u}}\mathbf{F}_k] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}\end{aligned}\quad (5.13)$$

$$(5.14)$$

since

$$\mathbf{L}_k = \mathbf{F}_k(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) - \nabla_{\mathbf{x}}\mathbf{F}_k\Big|_{\bar{\mathbf{x}}_k} \bar{\mathbf{x}}_k - \nabla_{\mathbf{u}}\mathbf{F}_k\Big|_{\bar{\mathbf{u}}_k} \bar{\mathbf{u}}_k \approx 0$$

we obtain

$$\mathbf{x}(k+1|k) = [\Phi_{\mathbf{x}}(k+1|k) \ \Phi_{\mathbf{u}}(k+1|k)] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \quad (5.15)$$

From the theory of Kalman filter we have that the prediction of the state error covariance matrix is

$$\mathbf{P}(k|k-1) = \Phi(k|k-1)\mathbf{P}(k-1|k-1)\Phi(k|k-1)^{\top} \quad (5.16)$$

where

$$\mathbf{P}(k-1|k-1) = \begin{bmatrix} \mathbf{P}_{\mathbf{x}}(k-1|k-1) & \mathbf{O} \\ \mathbf{O} & \mathbf{U}(k-1) \end{bmatrix} \quad (5.17)$$

So from (5.16) we have the equation for propagating covariance matrix taking into account the input vector

$$\mathbf{P}(k|k-1) = \nabla_{\mathbf{x}}\mathbf{F}_k\mathbf{P}(k-1|k-1)\nabla_{\mathbf{x}}\mathbf{F}_k^{\top} + \nabla_{\mathbf{u}}\mathbf{F}_k\mathbf{U}_k\nabla_{\mathbf{u}}\mathbf{F}_k^{\top} + \mathbf{Q}(k) \quad (5.18)$$

5.3 EKF Solution

Solutions to the probabilistic SLAM problem involve finding an appropriate representation for both the observation model and motion model that allows efficient and consistent computation of the prior and posterior distributions. By far, the most common representation is in the form of a state-space model with additive Gaussian noise, leading to the use of the extended Kalman filter (EKF) [M.S. Grewal and A.P. Andrews, 2001] to solve the SLAM problem.

5.3.1 System State

The vehicle travels through the environment using its sensors to observe features. The state of the system at time k can therefore be represented by the *augmented* state vector, $\mathbf{x}_k$, consisting of the $\mathbf{x}_v$ state vector representing the vehicle and the state vector $\mathbf{m}$ describing the observed features

$$\mathbf{x}_k = [\mathbf{x}_v(k), \mathbf{m}(k)]^T = [\mathbf{x}_v(k), \mathbf{m}_1(k), \dots, \mathbf{m}_i(k)]^T \quad (5.19)$$

where vehicle state and landmarks state are represented in the same reference frame.

5.3.2 Vehicle and Landmark Model

In general the process model for a system describes how the system states change as a function of time and is usually written as a first order non-linear vector differential equation or state model of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{w}(t) \quad (5.20)$$

where $(\mathbf{x}(t) \in \mathbb{R}^n)$ is a vector of the states of interest at time t , $\mathbf{u}(t) \in \mathbb{R}^r$ is a known control input, $\mathbf{f}(\cdot, \cdot, \cdot)$ is a model of the rate of change of system state as a function of time and $\mathbf{w}(t)$ is a random vector describing uncertainties in the state model itself (*process noise*). In general, process models are continuous and must be discretised for implementation on a digital computer so we have²

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k)) + \mathbf{w}(k) \quad (5.21)$$

A vehicle model describes the fundamental relationship between the vehicle's past state and the current state, given a control input, so we get

$$\mathbf{x}_v(k) = \mathbf{f}_v(\mathbf{x}_v(k-1), \mathbf{u}(k)) + \mathbf{w}_v(k) \quad (5.22)$$

where $\mathbf{x}_v(k) \in \mathbb{R}^v$ is the vehicle state at time step k and $\mathbf{w}_v(k)$ is a random vector describing both uncertainties in the vehicle state model itself.

An accurate vehicle model is an essential component of most navigation schemes. Vehicle models can be of varying degrees of complexity. Depending on the reliability with which the motion of the vehicle can be modeled and the sensors available for predicting the change of state of the vehicle, highly accurate estimates of vehicle motion can be generated. Ideally, of course, a vehicle model would capture the motion of the vehicle without any uncertainty and for each state transition, the vehicle state would be precisely known. This is a practical impossibility, however, as the models used do not, in general, perfectly capture the vehicle motion and sensors and actuators are subject to noise that will slowly corrupt the accuracy of the state estimate.

In phase of testing, where there's a necessity to generate syntetic vehicle data (IMU measures), the dynamic model of the vehicle must be accurate as possible. In this scenario the noise process term include all is unknown in the model.

In the context of SLAM, a landmark is a feature of the environment that can be consistently and reliably observed using the vehicle's sensors. Landmarks must be described in parametric form to allow them to be incorporated into a state model. For the SLAM algorithm, the feature states are usually assumed to be stationary. The dynamic portion of the process model therefore consists only of a vehicle model.

This leads to the simple landmark model

$$\mathbf{m}_i(k) = \mathbf{m}_i(k-1) \quad (5.23)$$

²We indicate with $\mathbf{f}$ the same function $\mathbf{F}$ of eq.(5.10)

where $\mathbf{m}_i(k) \in \mathbb{R}^l$ is the landmark state at time k

5.3.3 Sensor model

The state observation process can also be modelled in state-space notation by a non-linear vector function in the form

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) + \boldsymbol{\nu}(t)$$

This description is analogous to what said for process model, here h is a model of the observation of system states as a function of time and $\boldsymbol{\nu}(t)$ is a random vector describing both measurement corruption noise and uncertainties in the measurement model itself. These models describe the true observation at time t given the true state of the system $\mathbf{x}(t)$. The observation process lends itself well to the discrete observation model

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \boldsymbol{\nu}(k) \quad (5.24)$$

5.3.4 Estimation Process

In the estimation-theoretic formulation of the SLAM problem, the Kalman filter [M.S. Grewal and A.P. Andrews, 2001] is used to provide estimates of vehicle and landmark location. We briefly summarize the notation and main stages of this process as a necessary prelude to the future developments.

The Kalman filter recursively computes estimates for a state $\mathbf{x}(k)$ which is evolving according to the process model in and which is being observed according to the observation model. The Kalman filter computes an estimate which is equivalent to the conditional mean

$$\hat{\mathbf{x}}(p|q) = E[\mathbf{x}(p)|\mathbf{Z}_{0:q}] \quad (p \geq q)$$

The error in the estimate is denoted by $\tilde{\mathbf{x}} = \mathbf{x}(p|q) - \hat{\mathbf{x}}(p|q)$. The Kalman filter also provides a recursive estimate of the covariance in the estimate

$$\mathbf{P}(p|q) = E[\tilde{\mathbf{x}}(p|q)\tilde{\mathbf{x}}(p|q)^T | \mathbf{Z}_{0:q}]$$

The posterior state estimate will be written

$$\hat{\mathbf{x}}(k|k)$$

and the priori state estimate

$$\hat{\mathbf{x}}(k|k-1)$$

The filter fuses a prior state estimate with an observation $\mathbf{z}(k)$ of the state $\mathbf{x}(k)$ at time k to produce the updated estimate $\hat{\mathbf{x}}(k|k)$. The Kalman Filter makes the simplifying assumption that the process noise, $\mathbf{w}(k)$, and observation noise, $\boldsymbol{\nu}(k)$, are temporally uncorrelated and zero mean

$$E[\mathbf{w}(k)] = E[\boldsymbol{\nu}(k)] = E[\mathbf{w}(k)\mathbf{w}(k)^T] = 0 \forall k$$

with covariances

$$E[\mathbf{w}(k)\mathbf{w}(k)^T] = \mathbf{Q}(k)$$

$$E[\boldsymbol{\nu}(k)\boldsymbol{\nu}(k)^T] = \mathbf{R}(k)$$

For control inputs we have

$$E[\mathbf{u}(k)] = 0 \forall k \quad E[\mathbf{u}(k)\mathbf{u}(k)^T] = \mathbf{U}(k)$$

The estimate of the augmented state vector include both vehicle pose estimate and landmarks' position.

$$\hat{\mathbf{x}} = [\hat{\mathbf{x}}_v(k), \mathbf{m}_1(k), \dots, \mathbf{m}_l(k)] \quad (5.25)$$

The covariance matrix for this state, which defines the mean squared error and error correlations in each of the state estimate is

$$\mathbf{P}(k|k) = E[(\mathbf{x}(k|k) - \hat{\mathbf{x}}(k|k))(\mathbf{x}(k|k) - \hat{\mathbf{x}}(k|k))^T | \mathbf{Z}_{0:k}] \quad (5.26)$$

in matrix form

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{vv}(k|k) & \mathbf{P}_{v1}(k|k) & \cdots & \mathbf{P}_{vl}(k|k) \\ \mathbf{P}_{v1}^T(k|k) & \mathbf{P}_{11}(k|k) & \cdots & \mathbf{P}_{1l}(k|k) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{vl}^T(k|k) & \mathbf{P}_{1l}^T(k|k) & \cdots & \mathbf{P}_{ll}(k|k) \end{bmatrix} \quad (5.27)$$

where

- $\mathbf{P}_{vv}(k|k)$ represents vehicle covariance
- $\mathbf{P}_{ii}(k|k)$ represents i-landmark covariance
- $\mathbf{P}_{vi}(k|k)$ represents the cross-covariance between the vehicle and i-landmark estimates
- $\mathbf{P}_{ij}(k|k)$, $1 < i, k < l$ represents the cross-covariance between landmarks

we can write the (5.27) in a more compact form as

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{vv}(k|k) & \mathbf{P}_{vm}(k|k) \\ \mathbf{P}_{vm}^T(k|k) & \mathbf{P}_{mm}(k|k) \end{bmatrix} \quad (5.28)$$

5.3.4.1 Prediction Stage

The prediction stage of the filter uses the model of the motion of the vehicle (or data from an instrument) to generate an estimate of the vehicle state parameters at instant k given the information available to instant $k-1$ and the landmark model for landmark position prediction. Together these two model result in the propagation of the augmented state matrix during the prediction cycle of the filter

$$\begin{bmatrix} \hat{\mathbf{x}}_v(k|k-1) \\ \hat{\mathbf{m}}(k|k-1) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\hat{\mathbf{x}}(k-1|k-1), \mathbf{u}(k)) \\ \hat{\mathbf{m}}(k-1|k-1) \end{bmatrix} \quad (5.29)$$

The covariance matrix must also be propagated through the vehicle model. The Extended Kalman Filter linearises the propagation of uncertainty about the current state estimate $\hat{\mathbf{x}}(k-1|k-1)$ using the the Jacobian $\nabla_{\mathbf{x}} \mathbf{f}(k)$ of $\mathbf{f}$ evaluated at $\hat{\mathbf{x}}(k-1|k-1)$, and we get

$$\mathbf{P}(k|k-1) = \nabla_{\mathbf{x}} \mathbf{f}(k) \mathbf{P}(k-1|k-1) \nabla_{\mathbf{x}} \mathbf{f}^T(k) + \mathbf{Q}(k) \quad (5.30)$$

Uncertainty in the control inputs, $\mathbf{u}(k)$ used to drive the prediction can be also be accounted

$$\mathbf{P}(k|k-1) = \nabla_{\mathbf{x}} \mathbf{f}(k) \mathbf{P}(k-1|k-1) \nabla_{\mathbf{x}} \mathbf{f}^T(k) + \nabla_{\mathbf{u}} \mathbf{f}(k) \mathbf{U}(k) \nabla_{\mathbf{u}} \mathbf{f}^T(k) + \mathbf{Q}(k) \quad (5.31)$$

For the SLAM algorithm, this step in the filter can be simplified because of the assumption that the feature states are stationary. This allows the complexity of computing the predicted covariance to be reduced by requiring that only the variances associated with the vehicle and

the cross covariance terms between the vehicle and the map are updated during the prediction step, in fact we have

$$\nabla_{\mathbf{x}} \mathbf{f}^\top(k) = \begin{bmatrix} \nabla_{\mathbf{v}} \mathbf{f}^\top(k) & 0 \\ 0 & \mathbb{I} \end{bmatrix} \quad (5.32)$$

and so

$$\begin{aligned} \mathbf{P}(k|k-1) &= \begin{bmatrix} \nabla_{\mathbf{v}} \mathbf{f}(k) & 0 \\ 0 & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{vv}(k-1|k-1) & \mathbf{P}_{vm}(k-1|k-1) \\ \mathbf{P}_{vm}^\top(k-1|k-1) & \mathbf{P}_{mm}(k-1) \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{v}} \mathbf{f}^\top(k) & 0 \\ 0 & \mathbb{I} \end{bmatrix} \\ &+ \nabla_{\mathbf{u}} \mathbf{f}(k) \mathbf{U}(k) \nabla_{\mathbf{u}} \mathbf{f}^\top(k) + \mathbf{Q}(k) \\ &= \begin{bmatrix} \nabla_{\mathbf{v}} \mathbf{f}(k) \mathbf{P}_{vv}(k-1|k-1) \nabla_{\mathbf{v}} \mathbf{f}^\top(k) & \nabla_{\mathbf{v}} \mathbf{f}(k) \mathbf{P}_{vm}(k-1|k-1) \\ \mathbf{P}_{vm}^\top(k-1|k-1) \nabla_{\mathbf{v}} \mathbf{f}^\top(k) & \mathbf{P}_{mm}(k-1) \end{bmatrix} \\ &+ \nabla_{\mathbf{u}} \mathbf{f}(k) \mathbf{U}(k) \nabla_{\mathbf{u}} \mathbf{f}^\top(k) + \mathbf{Q}(k) \end{aligned} \quad (5.33)$$

5.3.4.2 Observation

Using the observation model we could calculate the predicted observation

$$\hat{\mathbf{z}}(k|k-1) = \mathbf{h}(\hat{\mathbf{x}}(k|k-1))$$

When the observation are received from on board sensors they must be associated with a feature in the map (see below for feature initialization phase). The difference between the actual observation, $\mathbf{z}(k)$, received from the system's sensors and the predicted observation, is called the *innovation* $\gamma(k)$,

$$\gamma(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k|k-1) \quad (5.34)$$

Now we can compute the *innovation covariance*

$$\mathbf{S}(k) = \nabla_{\mathbf{x}} \mathbf{h}(k) \mathbf{P}(k|k-1) \nabla_{\mathbf{x}} \mathbf{h}^\top(k) + \mathbf{R}(k) \quad (5.35)$$

The calculation of the innovation covariance can be simplified by noting that each observation is only a function of the feature being observed so

$$\nabla_{\mathbf{x}} \mathbf{h}(k) = [\nabla_{\mathbf{v}} \mathbf{h}(k) \ 0 \ \dots \ 0 \ \nabla_i \mathbf{h}(k) \ 0 \ \dots] \quad (5.36)$$

5.3.4.3 Update

Once the observation has been associated with a particular feature in the map, the state estimate can be updated using the optimal gain matrix $\mathbf{W}(k)$. This gain matrix provides a weighted sum of the prediction and observation and is computed using the innovation covariance, $\mathbf{S}(k)$ and the predicted state covariance, $\mathbf{P}(k|k-1)$. Finally we can compute the state update $\hat{\mathbf{x}}(k|k)$ as well as the updated state covariance $\mathbf{P}(k|k)$

$$\mathbf{W}(k) = \mathbf{P}(k|k-1) \nabla_{\mathbf{x}} \mathbf{h}^\top(k) \mathbf{S}^{-1}(k) \quad (5.37)$$

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{W}(k) \gamma(k) \quad (5.38)$$

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{W}(k) \mathbf{S}(k) \mathbf{W}^\top(k) \quad (5.39)$$

For simplicity of notation we'll refer to the estimated state omitting the hat.

5.4 Managing Features

5.4.1 Feature Initialization and State Augmentation

When a new feature is observed (no data association occurs) its estimate must be properly initialised and added to the state vector. A feature initialization model $\mathbf{g}(\cdot, \cdot)$ is needed to map the current state estimate and a new observation $\mathbf{z}(k)$ to a new feature estimation $\mathbf{m}_i(k)$

$$\mathbf{m}_i(k) = \mathbf{g}_i(\mathbf{x}_v(k|k-1), \mathbf{z}(k)) \quad (5.40)$$

This *new* state estimate is then *appended* to the state vector as new map feature element. The covariances of the new feature estimates must also be properly initialised since the initial feature estimate depends on the current vehicle estimate and is therefore correlated with the rest of the vehicle and other map state estimates. Ignoring the correlation between the new state estimates and the remainder of the map can lead to inconsistency in the filtering process.

The general idea of state augmentation can be applied whenever new states are functions of a subset of existing states

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1(k|k-1) \\ \mathbf{x}_2(k|k-1) \\ \mathbf{g}(\mathbf{x}_2(k|k-1), \mathbf{q}(k)) \end{bmatrix}$$

so the Jacobian matrix will be

$$\nabla_{\mathbf{x}} \mathbf{g}(k) = \begin{bmatrix} \mathbb{I} & 0 & 0 \\ 0 & \mathbb{I} & 0 \\ 0 & \nabla_{\mathbf{x}_2} \mathbf{g}(k) & \nabla_{\mathbf{q}} \mathbf{g}(k) \end{bmatrix} \quad (5.41)$$

The covariance matrix is augmented with the observation covariance $\mathbf{R}(k)$

$$\mathbf{P}^*(k|k-1) = \begin{bmatrix} \mathbf{P}_{11}(k|k-1) & \mathbf{P}_{12}(k|k-1) & 0 \\ \mathbf{P}_{12}^T(k|k-1) & \mathbf{P}_{22}(k|k-1) & 0 \\ 0 & 0 & \mathbf{R}(k) \end{bmatrix} \quad (5.42)$$

The final covariance is

$$\begin{aligned} \mathbf{P}(k|k) &= \\ &= \begin{bmatrix} \mathbb{I} & 0 & 0 \\ 0 & \mathbb{I} & 0 \\ 0 & \nabla_{\mathbf{x}_2} \mathbf{g}(k) & \nabla_{\mathbf{q}} \mathbf{g}(k) \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11}(k|k-1) & \mathbf{P}_{12}(k|k-1) & 0 \\ \mathbf{P}_{12}^T(k|k-1) & \mathbf{P}_{22}(k|k-1) & 0 \\ 0 & 0 & \mathbf{R}(k) \end{bmatrix} \begin{bmatrix} \mathbb{I} & 0 & 0 \\ 0 & \mathbb{I} & \nabla_{\mathbf{x}_2} \mathbf{g}^T(k) \\ 0 & 0 & \nabla_{\mathbf{q}} \mathbf{g}^T(k) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{P}_{11}(k|k-1) & \mathbf{P}_{12}(k|k-1) & \mathbf{P}_{12}(k|k-1) \nabla_{\mathbf{x}_2} \mathbf{g}^T \\ \mathbf{P}_{12}^T(k|k-1) & \mathbf{P}_{22}(k|k-1) & \mathbf{P}_{22}(k|k-1) \nabla_{\mathbf{x}_2} \mathbf{g}^T \\ \nabla_{\mathbf{x}_2} \mathbf{g} \mathbf{P}_{12}^T(k|k-1) & \nabla_{\mathbf{x}_2} \mathbf{g} \mathbf{P}_{22}(k|k-1) & \nabla_{\mathbf{x}_2} \mathbf{g} \mathbf{P}_{22}(k|k-1) \nabla_{\mathbf{x}_2} \mathbf{g}^T + \nabla_{\mathbf{q}} \mathbf{g} \mathbf{R}(k) \nabla_{\mathbf{q}} \mathbf{g}^T \end{bmatrix} \end{aligned}$$

Applying this results on the augmented state

$$\begin{bmatrix} \mathbf{x}_v(k|k-1) \\ \mathbf{m}(k|k-1) \\ \mathbf{g}(\mathbf{x}_v(k|k-1), \mathbf{z}(k)) \end{bmatrix} \quad (5.43)$$

the augmented covariance become

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{P}_{vv}(k|k-1) & \mathbf{P}_{vm}(k|k-1) & \mathbf{P}_{vv}^T(k|k-1) \nabla_{\mathbf{v}} \mathbf{g}^T(k) \\ \mathbf{P}_{vm}^T(k|k-1) & \mathbf{P}_{mm}(k|k-1) & \mathbf{P}_{vm}^T(k|k-1) \nabla_{\mathbf{v}} \mathbf{g}^T(k) \\ \nabla_{\mathbf{v}} \mathbf{g}(k) \mathbf{P}_{vv}^T(k|k-1) & \nabla_{\mathbf{v}} \mathbf{g}(k) \mathbf{P}_{vm}^T(k|k-1) & \nabla_{\mathbf{v}} \mathbf{g}(k) \mathbf{P}_{vv}^T(k|k-1) \nabla_{\mathbf{v}} \mathbf{g}^T(k) + \nabla_{\mathbf{z}} \mathbf{g}(k) \mathbf{R}(k) \nabla_{\mathbf{z}} \mathbf{g}^T(k) \end{bmatrix}$$

Essential steps of the filtering process is showed in figure 5.3.

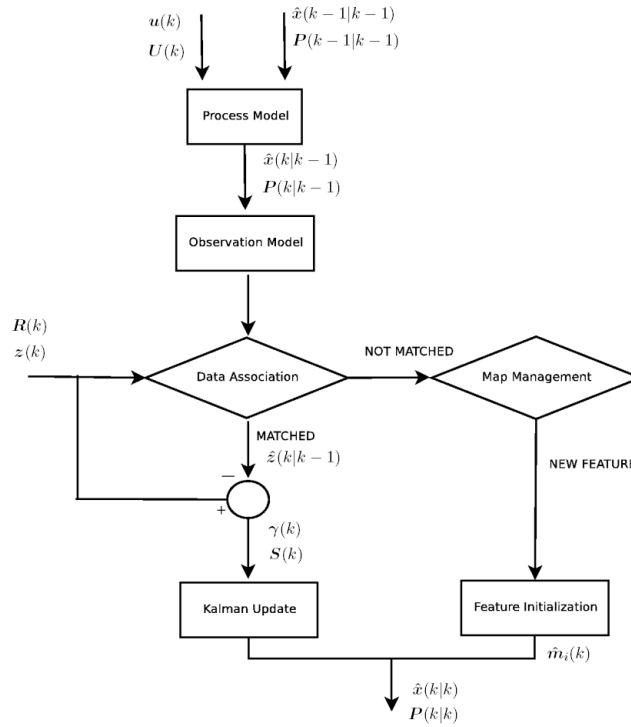


Figure 5.3: The essential steps of the filtering process

5.4.2 Features Extraction

The development of feature based navigation relies on the ability of a sensor system to extract appropriate and reliable features with which to build maps.

The feature extraction process is highly application dependent and depends on the anticipated environment in which the vehicle will operate, and the sensors used to observe this environment. Some environments, such as those found in typical offices, lend themselves to the extraction of corner and line features using such sensors as lasers and vision systems. In unstructured environments as in case of surface monitoring, simple corner or line features are not commonly observed. Additionally outdoor sensors very often can not provide the same accuracy or detail as can be obtained from indoor navigation sensors. Should the vehicle be required to operate in more unstructured environments, these features may prove to be insufficient for modelling the vehicle's surroundings. Alternative methods for terrain modelling have recently appeared in the literature in subsea robotic applications, it would be interesting to study their application on aerovehicles case.

5.4.3 Data Association

Data association is essential to the operation of the Simultaneous Localisation and Mapping algorithm. The estimated location of landmark positions rely on the accuracy of the vehicle location estimate. An incorrect association of an observation to the map can cause the filter

to diverge from a consistent estimate, effectively rendering all future predicted observations incorrect.

A common statistical discriminator is based on the normalised innovation vector squared between two estimates, called *nearest neighbour* association. Given the observation vector at instant k $\mathbf{z}(k)$, comprising a range and bearing to the observed landmark, the innovation, $\boldsymbol{\gamma}(k)$, and innovation covariance, $\mathbf{S}(k)$ could be used for a validation rule.

$$d_{m_i} = \boldsymbol{\gamma}^T \mathbf{S}^{-1} \boldsymbol{\gamma} < \tau \quad (5.44)$$

Unfortunately, it is quite difficult to detect and recover from an incorrect association by relying exclusively on nearest neighbour techniques as associations depend on calculating a predicted observation based on the current estimate of vehicle location. If the vehicle location estimate is in error, then an observation to a known landmark will be estimated to have occurred from a different map position. In this case, there is a risk that the filter may incorrectly associate the observation with another landmark in the map or initialise a new feature at this position. Recent work examined the possibility to improve the performance of data association compared with the simple data association approach.

5.5 Computational Complexity

One significant obstacle on the road to the implementation and deployment of large scale SLAM algorithms is the computational effort required to maintain the correlation information between features in the map and between the features and the vehicle. Performing the update of the covariance matrix is of $O(n^3)$ for a straightforward implementation of the Kalman Filter. In the case of the SLAM algorithm, this complexity can be reduced to $O(n^2)$ given the sparse nature of typical observations as shown in equations (5.33), (5.36), (5.44). Even so, this implies that the computational effort will grow with the square of the number of features maintained in the map. For maps containing more than a few tens of features, this computational burden will quickly make the update intractable - especially if the observation rates are high. An effective map-management technique is therefore required in order to help manage this complexity.

One simple solution could be reduce the number of feature in the map, using an active sensing feature in with a sensor is controlled to look at a only limited part in the map. Alternatively, features can be deleted from the map after the vehicle has left an area; as might be expected, this loss of information will lead to a sub-optimal estimate.

In [J. Guivant et al., 2000] it's proposed a simplification based on the fact that update of a large proportion of features in the map is insignificant. The updates are applied to small local region and update the global map done only at a much lower frequency (*partitioned updates*). It's shown that the computational complexity reduces to $O(n)$ [T. Bailey and H.F. Durant-Whyte, 2006]. Submaps methods are means of addressing the issue of computation scaling quadratically with the number of landmarks during measurement update. The *local submap* estimates are obtained using the standard optimal SLAM algorithm using only the locally referenced landmarks. The resulting submap structures are then arranged in a hierarchy leading to computational efficiency but sometimes with lack of optimality [C. Estrada et al., 2005].

5.6 Compressed Extended Kalman Filter (CEKF)

In this section we demonstrate that it is not necessary to perform a full SLAM update when working in a local area. This is a fundamental result because it reduces the computational

requirement of the SLAM algorithm to the order of the number of features in the vicinity of the vehicle; independent of the size of the global map. A common scenario is to have the airship moving in an area and observing features within this area.

This approach will also present significant advantages when the vehicle navigates for long periods of time in a local area or when the external information is available at high rate. Although high frequency external sensors are desirable to reduce position error growth, they also introduce a high computational cost in the SLAM algorithm.

Here we show that while working in a local area observing local landmarks we can preserve all the information processing a SLAM algorithm of the order of the number of landmarks in the local area. When the vehicle departs from this area, the information acquired can be propagated to the global landmarks without loss of information. This will also allow incorporating high frequency external information with very low computational cost. Another important implication is that the global map will not be required to update sequentially at the same rate of the local map.

With CEKF the full estimation problem is temporarily converted to a reduced estimation process that finally gives a solution identical to the full EKF estimation. The CEKF can easily manage high frequency sequences of predictions and updates avoiding the expensive cost of the full EKF estimation approach.

Consider the vehicle moving in a local area A and that the rest of the map is a part of the global area B . The state vector of the system be divided in two groups of states:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{bmatrix} \quad (5.45)$$

the states $\mathbf{x}_A$ can be initially selected as all states representing vehicle pose and landmarks in the area A . Suppose that for a time period $\Omega = \{k : k_1 \leq k \leq k_2\}$ the model and the observations have the following characteristics

$$\begin{bmatrix} \mathbf{x}_A(k) \\ \mathbf{x}_B(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_A(\mathbf{x}_A(k-1), \mathbf{u}(k)) + \mathbf{w}(k) \\ \mathbf{x}_A(k-1) + \mathbf{w}(k) \end{bmatrix} \quad (5.46)$$

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}_A(k)) + \mathbf{v}(k) \quad (5.47)$$

where $k \in \Omega$. So in the period Ω the observation obtained are only related with the states $\mathbf{x}_A(k)$ and do not involve states of $\mathbf{x}_B(k)$.

The covariance matrix could be represented according to this states division as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{AA} & \mathbf{P}_{AB} \\ \mathbf{P}_{AB}^\top & \mathbf{P}_{BB} \end{bmatrix} \quad (5.48)$$

It's evident that for the observation model holds

$$\nabla_{\mathbf{x}} \mathbf{h} = [\nabla_{\mathbf{x}_A} \mathbf{h} \quad \mathbf{0}] \quad (5.49)$$

and it's easy to prove that the kalman gain is

$$\mathbf{W}(k) = \mathbf{P}(k|k-1) \nabla_{\mathbf{x}} \mathbf{h}^\top \mathbf{S}^{-1}(k) = \begin{bmatrix} \mathbf{P}_{AA}(k|k-1) \nabla_{\mathbf{x}_A} \mathbf{h}^\top \mathbf{S}^{-1}(k) \\ \mathbf{P}_{AB}^\top(k|k-1) \nabla_{\mathbf{x}_A} \mathbf{h}^\top \mathbf{S}^{-1}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{W}_A(k) \\ \mathbf{W}_B(k) \end{bmatrix} \quad (5.50)$$

From these equation it is possible to see that

1. the Jacobian matrix $\nabla_{\mathbf{x}_A} \mathbf{h}$ has no dependence on the state $\mathbf{x}_B$

2. the innovation covariance matrix $\mathbf{S}$ and Kalman gain $\mathbf{W}_A$ are function of $\mathbf{P}_{AA}$ and $\nabla_{\mathbf{x}_A} \mathbf{h}$ only.

During long term navigation missions, the number of states in $\mathbf{x}_A$ will be in general much smaller than the total number of states in the global map, $\mathbf{x}_B, \mathbf{P}_{AB}, \mathbf{P}_{BB}$, only required when the vehicle enters in a new region; their evaluation can be done in one iteration with full SLAM computational cost using the compressed expression. The structure of the algorithm it's shown below.

At the beginning of the time period Ω a set of auxiliary matrices, Φ, Ψ, Θ that are never bigger than $\mathbf{P}_{AA}$, is created

$$\Phi, \Psi \in \mathbb{R}^{n_A \cdot n_A} \quad (5.51)$$

$$\Theta \in \mathbb{R}^{n_A} \quad (5.52)$$

having as initial conditions at $k = k_1$

$$\Phi(k_1) = \mathbf{1}, \Psi(k_1) = \mathbf{0}, \Theta(k_1) = \mathbf{0} \quad (5.53)$$

In every prediction or update stage, in the Ω interval, a normal EKF is run over the subsystem $\mathbf{x}_A, \mathbf{P}_{AA}$ and additional operations are done over the auxiliary matrices. Then in any prediction stage in the time period Ω , a standard EKF prediction step is done for $\mathbf{x}_A, \mathbf{P}_{AA}$ and the auxiliary matrices are updated according to

$$\Phi(k) = (\nabla_{\mathbf{x}_A} \mathbf{f}_A) \cdot \Phi(k-1) \quad (5.54)$$

$$\Psi(k) = \Psi(k-1) \quad (5.55)$$

$$\Theta(k|k-1) = \Theta(k-1) \quad (5.56)$$

At any local update a standard EKF update step is done for the subsystem $\mathbf{x}_A, \mathbf{P}_{AA}$ and the auxiliary matrices are updated according to

$$\Phi(k) = (\mathbf{1} - \mu(k))\Phi(k-1) \quad (5.57)$$

$$\Psi(k) = \Psi(k-1) + \Phi^\top(k-1) \cdot \beta(k)\Phi(k-1) \quad (5.58)$$

$$\Theta(k) = \Theta(k-1) + \Phi^\top(k-1) \cdot \nabla_{\mathbf{x}_A} \mathbf{h}^\top(k-1) \cdot \mathbf{S}^{-1}(k-1) \cdot \gamma(k-1) \quad (5.59)$$

where

$$\beta(k) = \nabla_{\mathbf{x}_A} \mathbf{h}(k) \quad (5.60)$$

$$\mu(k) = \mathbf{P}_{AA}(k|k-1)\beta(k) \quad (5.61)$$

$$\mathbf{S}(k) = (\nabla_{\mathbf{x}_A} \mathbf{h}(k))\mathbf{P}_{AA}(k|k-1)(\nabla_{\mathbf{x}_A} \mathbf{h}(k))^\top + \mathbf{R}(k) \quad (5.62)$$

$$\gamma(k) = \mathbf{z}(k) - \mathbf{h}(\mathbf{x}_A(k)) \quad (5.63)$$

At the end of the interval Ω , at k_2 , the update of all the states and the complete covariance matrix is done³:

$$\mathbf{P}_{AB}(k_2) = \Phi(k_2)\mathbf{P}_{AB}(k_1) \quad (5.64)$$

$$\mathbf{P}_{BB}(k_2) = \mathbf{P}_{BB}(k_1) - \mathbf{P}_{AB}^\top(k_1)\Psi(k_2)\mathbf{P}_{AB}(k_1) \quad (5.65)$$

$$\mathbf{x}_B(k_2) = \mathbf{x}_B(k_1) - \mathbf{P}_{AB}(k_1) \cdot \Theta(k_2) \quad (5.66)$$

It can be seen that the knowledge of $\mathbf{x}_B, \mathbf{P}_{AB}, \mathbf{P}_{BB}$ is only explicit in times k_1, k_2 .

No explicit information about this family of states and their related covariance and cross-covariance is known in $k_1 < k < k_2$. All this information is implicitly contained (at any instant k) in the auxiliary matrixes. The matrixes All the information (covariance, cross-covariance, value) of the set of states $\mathbf{x}_B$ is compressed in these 3 auxiliary matrices.

³The states $\mathbf{x}_B$ is never propagated but only updated

5.7 Monocular SLAM

A stereo camera system provide, bearing/elevation and range measurement, so the 3D coordinate of the observed landmarks are, to a certain extend, known.

However the 3D range of a stereo camera is limited and the immanent uncertainty of a landmarks's estimated 3D coordinated gets larger with its distance from the camera. While this rising uncertainty can be taken into account, depending on the camera parameters (base length, focal length pixel size), the range measurement bears no more viable information for landmarks from certain distance. This maximum observable range may be sufficiently large for ground operating robots with acceptable baseline lengths. Flying robots however, especially airship that could operate often in altitudes of 100 meters and beyond, needs larger baseline. After calibration, the slightest change in camera orientations leads to massive errors in depth estimation; stereo vision is very dependent on a good and enduring calibration. Large baseline lead either to more fragile and instable constructions. Due to these flaws of stereo approaches several authors addressed the problem with single or mono camera methods. As a single camera can not provide range measurement, the resulting algorithms are often called bearing only SLAM or simply monocular SLAM.

To implement the monocular SLAM we need a subfiltering process. When a feature is selected it's not possible to simply insert it into the state of the model since the initial condition is unknown, being available only measures of bearing.

One can address this problem by running a separate filter in parallel for each point using the current estimate of the motion from the main EKF in order to reconstruct the initial conditions. The basic idea is based on the fact that when we observe a feature at τ the subfilter follows the dynamical and observation model respectively

$$\mathbf{x}_\tau^i(t) = \mathbf{x}_\tau^i(t-1) + \mathbf{w}_x \quad (5.67)$$

$$\lambda_\tau^i(t) = \lambda_\tau^i(t-1) + \mathbf{w}_\lambda \quad (5.68)$$

$$\mathbf{x}^i(t) = {}^i\mathbf{H}_s(t) {}^i\mathbf{H}_s(\tau) {}^s\mathbf{x}^i(t) \lambda^i(t) = \mathbf{h}({}^s\mathbf{x}^i(t), \lambda^i(t), \tau) \quad (5.69)$$

where

- $\mathbf{x}^i$ is the bearing measurement vector of the i -feature;
- λ^i is the range measurement;
- $\mathbf{w}_x$ and $\mathbf{w}_\lambda$ are gaussian noise vectors;
- $\mathbf{H}$ is the homogeneous transformation

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} \quad (5.70)$$

Here $\mathbf{H}(t)$ and $\mathbf{H}(\tau)$ are the current best estimate and the best estimate at time τ of $\mathbf{H}$ respectively. When a new feature is observed at τ we store the information about state estimation and start the subfiltering process. When the feature is re-observed at time $t > \tau$, the subfilter relates the current pose of the vehicle with measure at time τ "predicting" the current measure by the model equation

$$\mathbf{x}^i(t) = {}^i\mathbf{H}_s(t) {}^i\mathbf{H}_s(\tau) {}^s\mathbf{x}^i(t) \lambda^i(t) = \mathbf{h}({}^s\mathbf{x}^i(t), \lambda^i(t), \tau) \quad (5.71)$$

The structure of the subfilter is the follow:

a) Initialization

$$\begin{aligned}
\mathbf{x}_\tau^i(\tau|\tau) &= \mathbf{x}^i(\tau) \\
\lambda_\tau^i(\tau|\tau) &= \lambda_0 \\
\mathbf{P}_\tau^i(\tau|\tau) &= \begin{bmatrix} \sigma_\alpha^2 & 0 & 0 \\ 0 & \sigma_\beta^2 & 0 \\ 0 & 0 & \sigma_{\lambda_0}^2 \end{bmatrix}
\end{aligned} \tag{5.72}$$

b) Prediction

$$\begin{aligned}
\mathbf{x}_\tau^i(k|k-1) &= \mathbf{x}^i(k|k) \\
\lambda_\tau^i(k|k-1) &= \lambda_\tau^i(k|k-1) \\
\mathbf{P}_\tau^i(k|k-1) &= \mathbf{P}_\tau^i(k|k) + \Sigma_w
\end{aligned} \tag{5.73}$$

c) Update

$$\begin{aligned}
\gamma(k) &= \mathbf{x}_\tau^i(k) - \mathbf{h}(\mathbf{x}^i(k), \lambda^i(k), \tau) \\
\mathbf{S}(k) &= \nabla_{\mathbf{x}} \mathbf{h} \mathbf{P}_\tau^i(k|k-1) \nabla_{\mathbf{x}} \mathbf{h}^\top + \mathbf{R}(k) \\
\mathbf{W}(k) &= \mathbf{P}_\tau^i(k|k-1) \nabla_{\mathbf{x}} \mathbf{h}^\top \mathbf{S}^{-1}(k) \\
\mathbf{x}_\tau^i(k|k) &= \mathbf{x}_\tau^i(k|k-1) + \mathbf{W}(k) \gamma(k) \\
\mathbf{P}_\tau^i(k|k) &= \mathbf{P}_\tau^i(k|k-1) + \mathbf{W}(k) \mathbf{S}(k) \mathbf{W}^\top(k)
\end{aligned} \tag{5.74}$$

Several heuristic can be employed in order to decide when the state estimate from the subfilter is good enough for it to be inserted into the main Kalman filter. A simple criterion is the covariance of the estimation error in the sub filter being comparable to the covariance in the main filter; another criterion is that covariance of the estimation error in the subfilter being less than a threshold given by a tuning phase.

When the feature is moved from the subfilter to the main filter the covariance matrix has to be properly initialized.

Chapter 6

Airship SLAM Implementation

6.1 Basics Remarks

Before developing a complex 3D SLAM implementation it has been usefull to simulate the behaviour of EKF based SLAM in some simple 1D and 2D, to approach to the main goal smoothly. This has been usefull to develop the proper mathematical toolbox, but also a certain comprehension of the problem, necessary for a more complex application.

It has been very instructive to understand the basis of SLAM, that could be quite simple in theory but frustrating during its implementation, working on simple physical model first.

6.1.1 1-DOF EKF Slam

In our first numerical approach we study the SLAM problem in the case of an accelerated unidimensional motion of a particle; no environment perturbations are present. The features are located along the line of motion and the vehicle has a fixed Field of View along it.

The dynamic model is described simply by the linear model¹

$$\dot{\mathbf{x}} = \frac{d}{dt}[\mathbf{v} \ \mathbf{p}]^T = \mathbf{A}\mathbf{x} + \mathbf{u} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \quad (6.1)$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad (6.2)$$

and the input vector

$$\mathbf{u} = [a(t) \ 0]^T = [1 \ 0]^T a(t) \quad (6.3)$$

where $a(t)$ is the linear acceleration. From this model we compute the Jacobians of the model function respect the vehicle statevector and input vector

$$\nabla_{\mathbf{x}} \mathbf{f} = \mathbf{A} \quad (6.4)$$

$$\nabla_{\mathbf{u}} \mathbf{f} = [1 \ 0]^T = \mathbf{b} \quad (6.5)$$

Integrating the equations of motion of the particle under the action of a specific force the mass trajecotry has been derived like simulated IMU measurement over the time, simulated by adding a gaussian noise to specific force profile.

¹In this case velocity and position are scalar

The continuous differential system is transformend in a dicrete-time one by standard first order *Euler step forward* method

$$\mathbf{x}(t_{n+1}) = \mathbf{x}_{n+1} = \mathbf{x}(t_n + \Delta t) = \mathbf{x}(t_n) + \dot{\mathbf{x}}\Delta t = \mathbf{x}(t_n) + \mathbf{f}(t, \mathbf{x}, \mathbf{u})\Delta t \quad (6.6)$$

so we have

$$\mathbf{x}_{n+1} = \mathbf{F}(t_n, \mathbf{x}_n, \mathbf{u}) \quad (6.7)$$

The Jacobians of $\mathbf{F}(t_n, \mathbf{x}_n, \mathbf{u})$ respect $\mathbf{x}$ and $\mathbf{u}$ from (5.11) are

$$\begin{cases} \nabla_{\mathbf{x}}\mathbf{F} = \mathbb{I} + \nabla_{\mathbf{x}}\mathbf{f}\Delta t = \mathbb{I} + \mathbf{A}\Delta t \\ \nabla_{\mathbf{u}}\mathbf{F} = \nabla_{\mathbf{u}}\mathbf{f}\Delta t = \mathbf{b}\Delta t \end{cases} \quad (6.8)$$

The augmented state which include features (monodimensional) locations, espressed in a reference frame with the origin in the vehicle frame at $t = 0$, is

$$\mathbf{x}_a = [\mathbf{x} \ \mathbf{m}]^T \quad (6.9)$$

Since landmarks are static in the map the dynamic of the system is

$$\dot{\mathbf{x}}_a = [\mathbf{x} \ 0]^T = \begin{bmatrix} \mathbf{f}(t_n, \mathbf{x}_n, \mathbf{u}) \\ 0 \end{bmatrix} \quad (6.10)$$

If we define the augmented discrete-time model similarly as (6.6)

$$\mathbf{x}_a(t+1) = \mathbf{x}_a(t) + \dot{\mathbf{x}}_a\Delta t \quad (6.11)$$

and so

$$\mathbf{F}_a = \begin{bmatrix} \mathbf{x} \\ \mathbf{m} \end{bmatrix} + \begin{bmatrix} \mathbf{f}(t, \mathbf{x}, \mathbf{u})\Delta t \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{x} + \mathbf{f}(t, \mathbf{x}, \mathbf{u})\Delta t \\ \mathbf{m} \end{bmatrix} \quad (6.12)$$

Now we can compute the Jacobian of $\mathbf{F}_a$ with respect to augmented state vector $\mathbf{x}_a$

$$\nabla_{\mathbf{x}_a}\mathbf{F}_a = \begin{bmatrix} \mathbb{I}_{(2 \times 2)} + \mathbf{A}\Delta t & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_{(m \times m)} \end{bmatrix} \quad (6.13)$$

where

$$\mathbb{I}_{(2 \times 2)} + \mathbf{A}\Delta t = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} \quad (6.14)$$

Range measures, y_i , of the i th feature is obtained from difference between features absolute position m_i and vehicle position p , when compatible with the FOV, in addition with a fixed range gaussian measurement noise.

$$y_i = m_i - p + \epsilon$$

The measurement depends of the position of the vehicle so features and vehicle dynamics are correlated; the Jacobian of measurement model function (5.36) follow from (5.36)²

$$\nabla_{\mathbf{x}_a}\mathbf{h} = [0, -1, \underbrace{0, \dots, 0}_i, \dots, 0] \quad (6.15)$$

where the number of rows depend from how observation are made at time t . So when a measurement or more are available and the observed feature is already present in the augmented state, we can compute the innovation vector from measurement prediction (known from previous step) from (5.34) the innovation covariance from (5.35) the Kalman gain from (5.37) and so the new estimate of the state (5.38) and estimate error covariance matrix (5.39).

If a feature in a new one the state has to be augmented and the covariance properly initialized. If no observation is available then only the prediction stage from IMU measurement is performed³.

²Features are indipendent each other so $\nabla_{\mathbf{m}}\mathbf{h} = \delta_{ij}$

³In this implementation every 25 time step a IMU measure and so a prediction is performed and every 250 (4 hz) time step a range measured is performed too.

6.1.1.1 Simulation Results

In this section we report the results of a simulation run where a mass point ($m = 50$ kg) moves subjected a certain force profile. A camera is placed on the point and grabs images of features with a frequency of 4 hz with a range extension of 15 m with an error of 0.7 m. The IMU sampling frequency is fixed at 40 hz. Features are positioned respect to the origins of motion at 5, 25, 30, 40, 70, 90, 100 meters.

As we could see from simulations results (Fig. 6.1), the estimate of vehicle state (position and velocity) is better if observations are available since vehicle and fetures are correlated entities. When measures are not available the filter do not update the state but only predict it from dynamical model and accumulates errors due to IMU noise. Features estimation error keeps within 0.5 m.

On order to set the filter initial parameters a *tuning* phase is necessary.

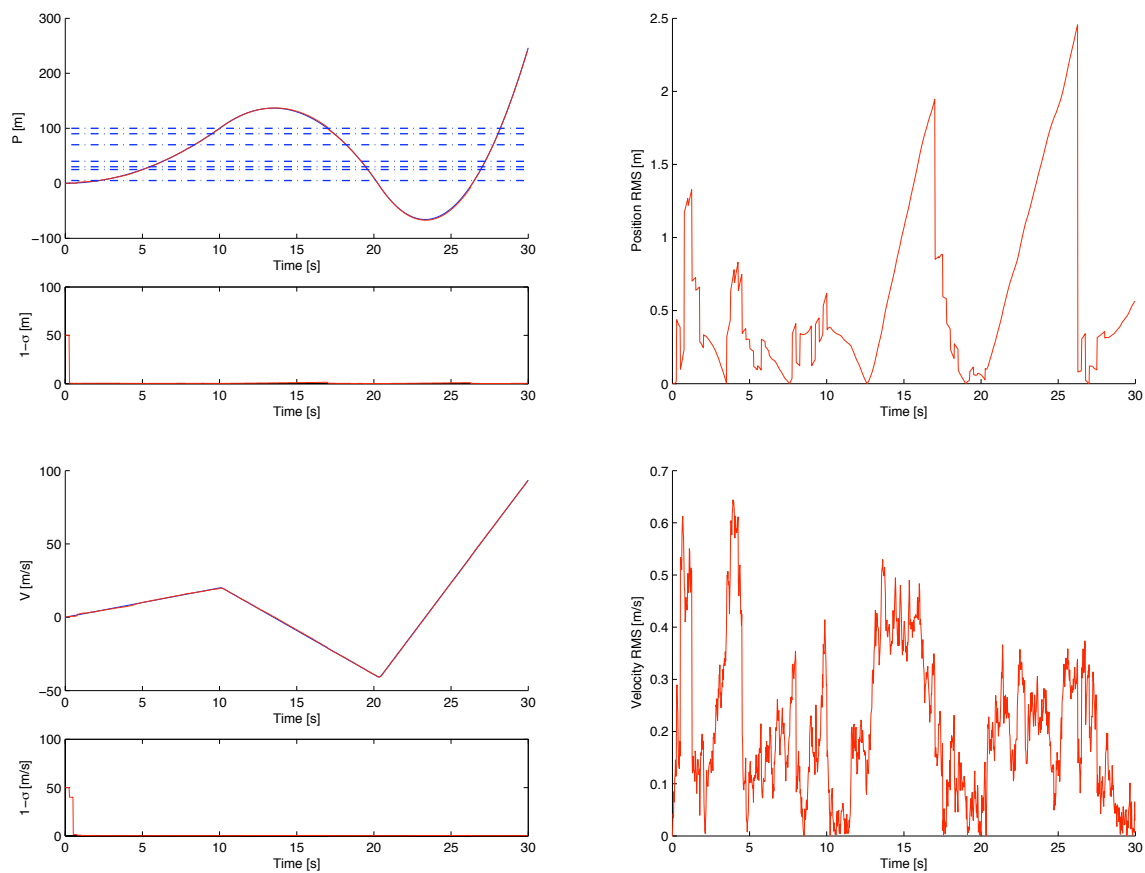


Figure 6.1: Real (blue) and estimated (red) vehicle position and velocity, below RMS error. Dashed line indicates feature position.

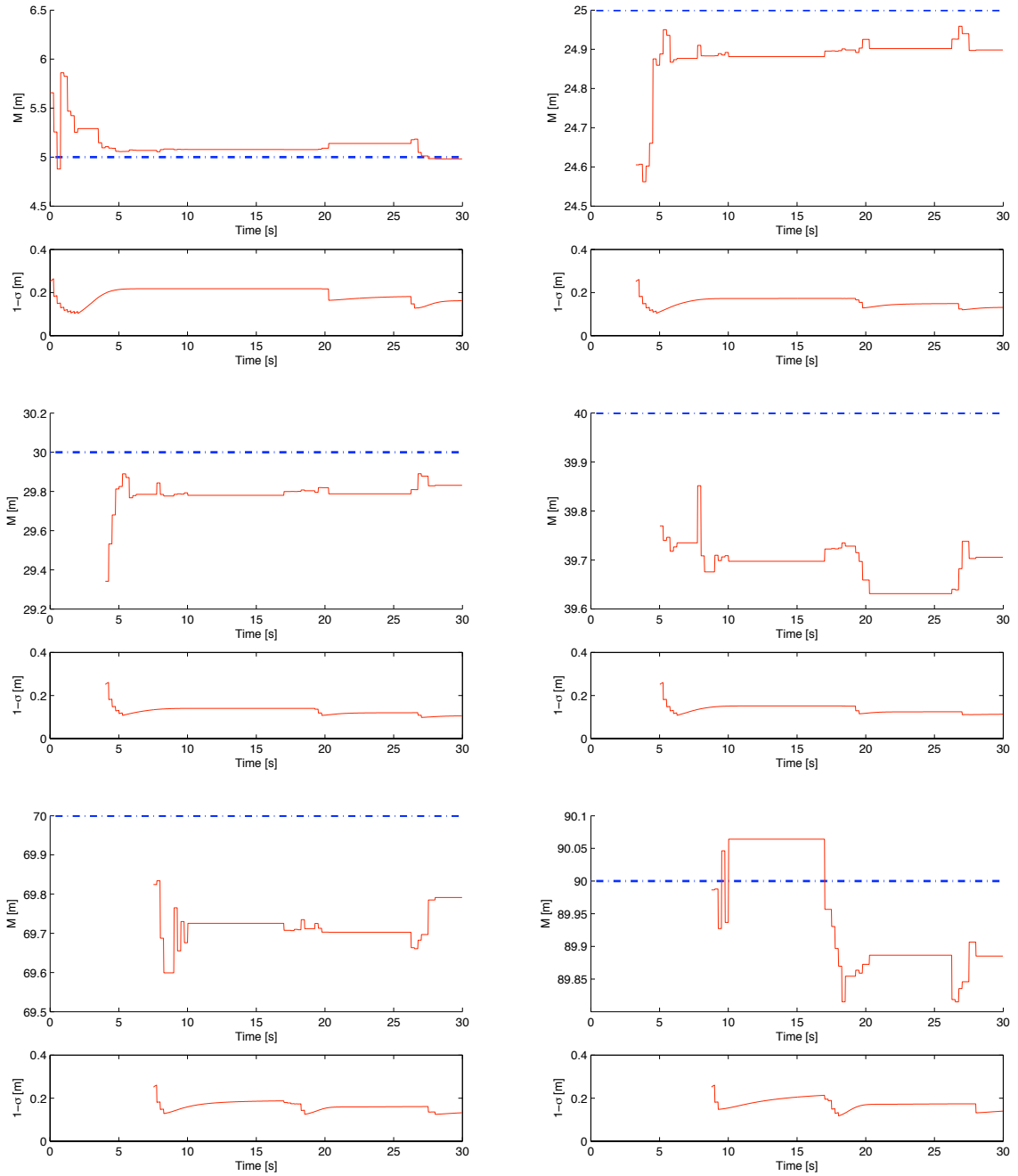


Figure 6.2: Features (1 to 6) estimation and stadard deviation

6.1.2 2-DOF EKF Slam: a rover with encoders

Assume a wheel vehicle equipped with dead reckoning capabilities and an external sensor capable of measuring relative distance between vehicle and the environment. The steering control γ , and the speed $\mathbf{v}_c$ are used with the kinematic model to predict the position of the vehicle. In this case the external sensor returns range and bearing information about features $\mathbf{m}_i$ ($i = 1 \dots n$)⁴. Suppose the vehicle controlled through a demanded velocity $\mathbf{v}_c$ and a steering angle γ , the process

⁴Now every feature is a point in a plane so $\mathbf{m}_i = (X, Y)$.

model that predicts the trajectory of the system in the inertial frame is given by

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) = \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} \begin{bmatrix} \frac{v_c}{L} \tan \gamma \\ v_c \cos \phi \\ v_c \sin \phi \end{bmatrix} \quad (6.16)$$

where L is the distance from the back fixed wheel axis and the front moving wheels axis.

A grid of landmark are placed in the inertial plane and the vehicle trajectory is obtained by integration of the equations of motion, defining a priori steering profile (a spiral) over the time. As in the monodimensional case we define some timing parameters like the total simulation time, encoders reading frequency (40 hz) and range-bearing measurements frequency (5 hz) The vehicle state vector is described by

$${}^i\mathbf{x} = [\phi, {}^i\mathbf{r}_v]^\top$$

and the input vector by

$${}^i\mathbf{u} = [\gamma, v_c]^\top$$

We are dealing with three different reference frame, the inertial reference frame where the vehicle motion and feature position are expressed, the vehicle reference frame and the sensor reference frame, where the measures are taken (see Figure 6.3,6.4).

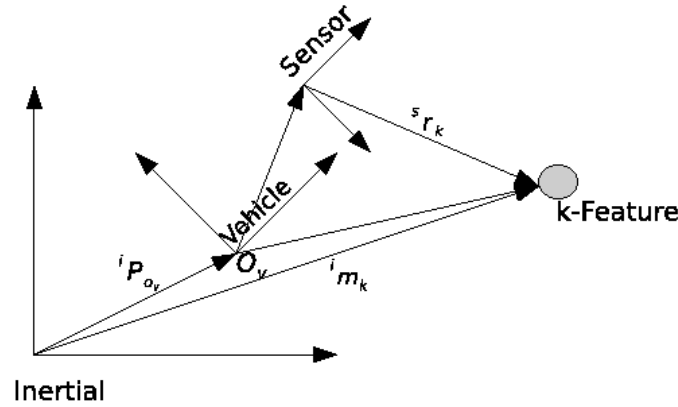


Figure 6.3: System reference frames

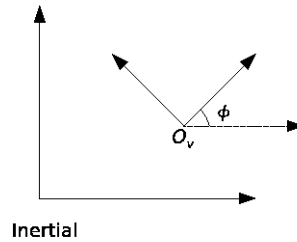


Figure 6.4: Vehicle bank angle

The sensor reference frame is located on the vehicle by the homogeneous transformation

$${}^v\mathbf{H}_s = \begin{bmatrix} {}^v\mathbf{R}_s & {}^v\mathbf{P}_{O_s} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^v\mathbf{R}_s & {}^v\mathbf{T}_s \\ 0 & 1 \end{bmatrix} \quad (6.17)$$

Now from (6.16) we could compute the jacobians $\nabla_{\mathbf{x}}\mathbf{f}$, $\nabla_{\mathbf{u}}\mathbf{f}$ obtaining

$$\nabla_{\mathbf{x}}\mathbf{f} = \begin{bmatrix} 0 & 0 & 0 \\ -v_c \sin \phi & 0 & 0 \\ v_c \cos \phi & 0 & 0 \end{bmatrix} \quad (6.18)$$

and

$$\nabla_{\mathbf{u}}\mathbf{f} = \begin{bmatrix} v_c/(r \cos^2 \gamma) & \tan \gamma/r \\ \cos \phi & 0 \\ \sin \phi & 0 \end{bmatrix} \quad (6.19)$$

As in the monodimensional case the augmented state is given by the vehicle state and the position of the observed features in the inertial frame.

$$\mathbf{x}_a = [\mathbf{x} \ \mathbf{m}]^T \quad (6.20)$$

where $\mathbf{m}$ is the vector contains all observed features position vector

$$\mathbf{m} = [\mathbf{m}_k]^T \quad k \in 1, \dots, n \quad (6.21)$$

The discrete-time system is obtained from continuous by (5.10) and from (6.18) and (6.19) we can compute the jacobians of the linearized discrete time system $\mathbf{F}_k$

$$\nabla_{\mathbf{x}}\mathbf{F}_k = \mathbb{I} + \nabla_{\mathbf{x}}\mathbf{f}(t, \mathbf{x}(k), \mathbf{u}(k))\Delta t \quad (6.22)$$

$$\nabla_{\mathbf{u}}\mathbf{F}_k = \nabla_{\mathbf{u}}\mathbf{f}(t, \mathbf{x}(k), \mathbf{u}(k))\Delta t. \quad (6.23)$$

If no observation are made by the system, the prediction stage of the filter is computed following equations (5.29), (5.31).

6.1.3 Measurement Process

A 2D grid of points is generated in the inertial frame, every point is a feature in the map. We suppose to have a stereo camera system in order to have range and bearing measurement of a feature.

Instead of taking range and bearing measure as $\mathbf{z}(k) = (r, \beta)$ where r is the distance from the beacon to the range sensor and β is the sensor bearing angle measured with respect to the vehicle coordinate frame, we formulate the problem from the pin hole camera model; first we chose the sensor frame to be the *canonical retinal frame*, where the optic axes is along y axis and x axis ortogonal to y passing through the optical center of the system. Then from pihhole model eq.(4.30) we have .

$$x = \frac{X}{Y}f \quad (6.24)$$

where (X, Y) is the world coorinate of the feature in inertial reference frame, and $f = 1$ without loss of generalization.

As said above the features are initialized in the inertial frame, but we have to get the measures in the sensor frame. First we have to map features visible by sensor (compatible with FOV and range capabilities of the cameras system) in the sensor reference frame by

$${}^s\mathbf{m} = {}^s\mathbf{H}_i {}^i\mathbf{m} \quad (6.25)$$

where ${}^i\mathbf{m}$ is known by the simulator. So we obtain

$$\begin{aligned} {}^s\mathbf{m} &= {}^s\mathbf{H}_v {}^v\mathbf{H}_i {}^i\mathbf{m} = [{}^i\mathbf{H}_v {}^v\mathbf{H}_s]^T {}^i\mathbf{m} \\ &= {}^i\mathbf{H}_s^T {}^i\mathbf{m} = (x_s, y_s) \end{aligned} \quad (6.26)$$

where

$${}^i\mathbf{H}_s = \begin{bmatrix} {}^i\mathbf{R}_v & {}^iT_v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^v\mathbf{R}_s & {}^v\mathbf{T}_s \\ 0 & 1 \end{bmatrix} \quad (6.27)$$

We could generate the range-bearing measurement vector (x, λ) in the sensor frame from 2D point copordinate simply⁵ (Fig 6.5)

$${}^s\boldsymbol{\alpha} = [x, \lambda] + \boldsymbol{\sigma}_m = [x_s/y_s, y_s]^T + \boldsymbol{\sigma}_m \quad (6.28)$$

where $\boldsymbol{\sigma}_m$ is a gaussian white measurement noise.

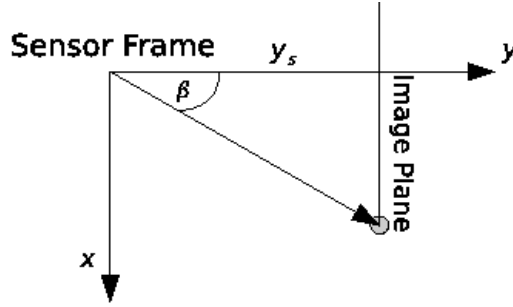


Figure 6.5: Camera Frame

When a feature is observed it must be checked if it's a new one or a previous observed by some kind of data association. The kalman filter requires the calculation of the measurement model to determine the innovation vector, and the measurement function jacobian for the state update. The measurement model is given in the inertial frame by the relation

$${}^i\hat{\mathbf{r}} = {}^i\mathbf{m} - {}^iT_v = \mathbf{h}(\mathbf{x}_a) \quad (6.29)$$

Now we explicitate the measurements as a function of vehicle position respect to the inertial frame in which the equations are written. First we refer all respect to the vehicle frame

$${}^v\mathbf{r} = {}^v\mathbf{R}_i({}^i\mathbf{m} - {}^iT_v) \quad (6.30)$$

where

$${}^iT_v = {}^i\mathbf{P}_{o_v} = x_{o_v} {}^i\hat{\mathbf{e}}_x + y_{o_v} {}^i\hat{\mathbf{e}}_y$$

the jacobian of the measured equation is therefore

$$\nabla_{\mathbf{x}_a} \mathbf{h} = \frac{\partial {}^v\mathbf{r}}{\partial [\phi, x, y, {}^i\mathbf{m}]} \quad (6.31)$$

where

$$\frac{\partial {}^v\mathbf{r}}{\partial \phi} = \frac{\partial {}^v\mathbf{r}}{\partial {}^v\mathbf{R}_i} \frac{\partial {}^v\mathbf{R}_i}{\partial \phi} \quad (6.32)$$

$$\frac{\partial {}^v\mathbf{r}}{\partial (x, y)} = -{}^v\mathbf{R}_i \quad (6.33)$$

$$\frac{\partial {}^v\mathbf{r}}{\partial {}^i\mathbf{m}} = {}^v\mathbf{R}_i \quad (6.34)$$

⁵Is more convenient to work with the tangent of the bearing angle

So we obtain when the k -th feature is observed

$$\nabla_{\mathbf{x}_a} {}^v\mathbf{h} = \begin{bmatrix} \frac{\partial {}^v\mathbf{r}}{\partial {}^v\mathbf{R}_i} \frac{\partial {}^v\mathbf{R}_i}{\partial \phi} & - {}^v\mathbf{R}_i & \underbrace{\mathbf{0}_{2 \times 2} \cdots {}^v\mathbf{R}_i \cdots}_{k} & \mathbf{0}_{2 \times 2} \end{bmatrix} \quad (6.35)$$

Since the measurement is made in the sensor frame we transform (6.30) simply by

$${}^s\mathbf{r} = {}^s\mathbf{H}_v {}^v\mathbf{r} \quad (6.36)$$

and also the jacobian ⁶

$$\nabla_{\mathbf{x}_a} {}^s\mathbf{h} = {}^s\mathbf{R}_v (\nabla_{\mathbf{x}_a} {}^v\mathbf{h}) \quad (6.37)$$

From ${}^s\mathbf{r}$ we derive the range-bearing measure by the function

$$(x_s, y_s) \xrightarrow[\Phi]{} (\tan \beta, y_s) = (x, \lambda) \quad (6.38)$$

The complete measurement process is described by

$${}^v\mathbf{x}_a \xrightarrow[h]{} {}^s\mathbf{r} \xrightarrow[\Phi]{} (\tan \beta, y_s) \quad (6.39)$$

where

$$\nabla_{\mathbf{x}_a} (\Phi(\mathbf{h})) = \nabla_{{}^s\mathbf{r}} (\Phi) \cdot \nabla_{\mathbf{x}_a} (\mathbf{h}) \quad (6.40)$$

When a new feature is observed we must initialize it initialize it by equation (5.40), in this case

$$\mathbf{g}({}^i\mathbf{x}_v, {}^i\mathbf{h}) = {}^i\mathbf{m} = {}^v\mathbf{m} + {}^i\mathbf{T}_v \quad (6.41)$$

so we augment the state vector including it and initialize the new covariance matrix by equation (5.44).

6.1.4 Simulation Results

In this section we report the results of the rover EKF Slam by simulations; feaures are positioned in a square grid with an equal distance to each other. Similary as 1-DOF case, the estimate of vehicle state (position and steering) is better if observations are available since vehicle and fetures are correlated entities. When measures are not available the filter do not update the state but only predict it from dynamical model and accumulating errors. In Table 6.1 are listed main simulation parameters.

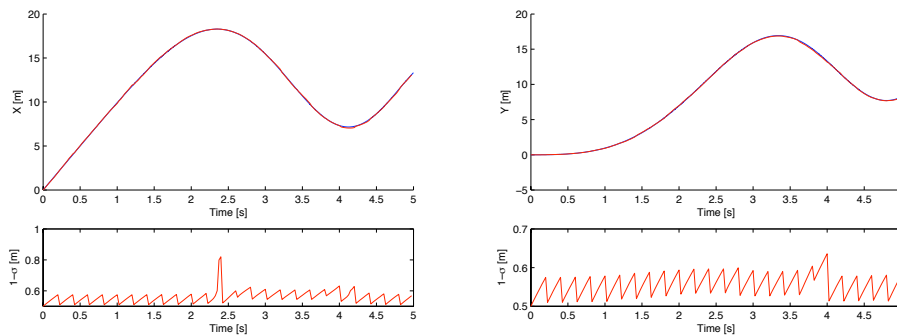


Figure 6.6: Real (blue) and estimated (red) vehicle X,Y-position component, and std deviation

⁶In general if ${}^s\mathbf{r} = {}^s\mathbf{R}_v {}^v\mathbf{r} + {}^s\mathbf{T}_v$ then applying the linear differential operator $\nabla_{\mathbf{x}}$ we have $\nabla_{\mathbf{x}} {}^s\mathbf{r} = {}^s\mathbf{R}_v \nabla_{\mathbf{x}} {}^v\mathbf{r}$

Wheel Axes Distance (L)	2 m
Encoder Frequency	40 Hz
Encoder Noise Variance, σ_{enc}^2	$1(m/s)^2$
Camera Frequency	5 Hz
Camera Range	$[0.5, 10.5]$
Camera FOV	20°
Measurement Noise Variance, σ_m^2	$[0.001, 0.01]^2$
Vehicle Velocity	10 m/s
Grid Resolution	2.5 m

Table 6.1: Simulation parameters

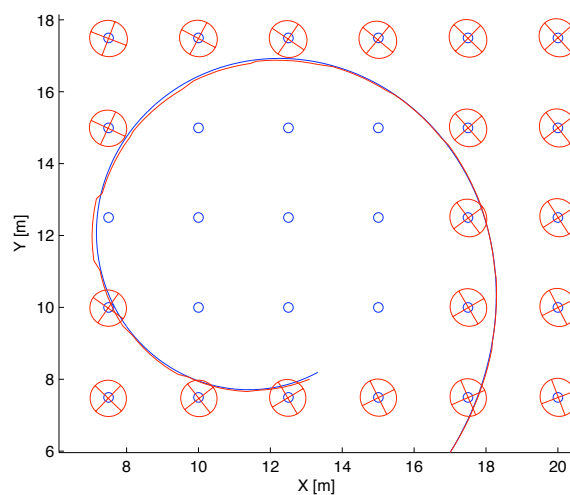


Figure 6.7: Estimated trajectory (red) and mapped points with uncertainty ellipse

It's interesting to analyze the situation in which the vehicle moves where no features are present. In this case no update is done but only the prediction stage is computed; figure 6.9 shows how this behavior affects the feature position estimate error.

If the update stage is not computed the filter accumulates error so when a new feature is observed its error is high due to the grown vehicle's position error. Only when previous features are re-observed, the filter is able to reduce errors due to their correlations.

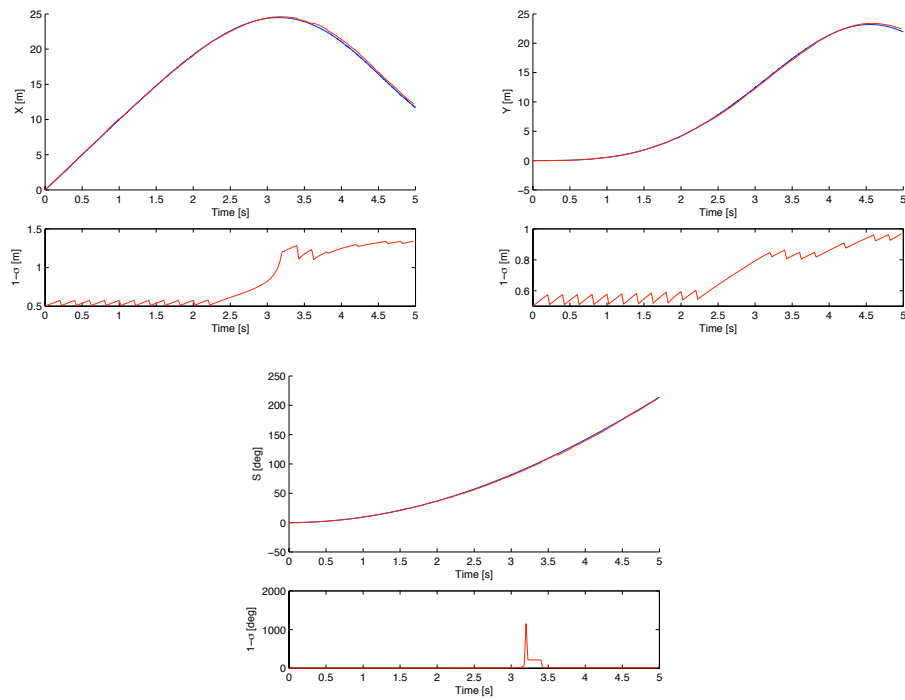


Figure 6.8: Real (blue) and estimated (red) vehicle X,Y-position component and steering angle, and std deviation

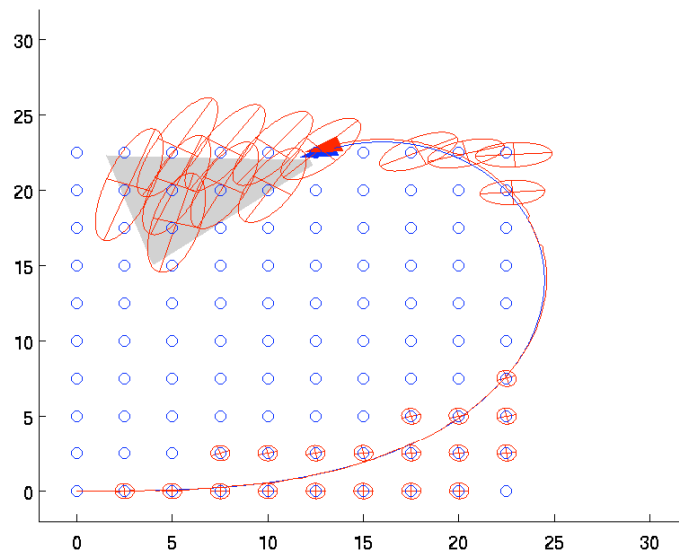


Figure 6.9: Estimated trajectory (red) and mapped points with uncertainty ellipse

6.2 Airship Slam

In this section are presented the implementation of the SLAM technique applied to an airship flight.

The purpose is to place the airship in an unknown environment and get terrain features measurement by a camera; by measures of vision and strapdown IMU systems we can estimate both vehicle kinematic state (attitude, velocity and position) and features positions with respect to the inertial frame whose origin is located on an arbitrary point of the surface⁷. Of course is possible to place this local map in an inertial geo-referenced frame if one point of the relative map is known in such frame.

Depending on flying altitude, a stereo or mono camera configuration shall be considered and therefore different estimation algorithms have to be implemented. If the landscape is too large, the computational complexity makes the estimation time too long and memory requirements to store data too expensive; should be considered to apply the compressed variant of the filter to manage submaps.

The only requirements for SLAM navigation are:

- on board inertial measurement;
- on board ground features measurement by vision by cameras;
- on board computer to process measurement data and perform state estimation;

6.2.1 State Estimation Module (SEM)

The aim of the Estimation module of the MASS software is the implementation of SLAM algorithms. The software is divided in two fundamental tasks that must communicate each another. The first requirement for developing a navigation system is to have a simulation of the physical behaviour of the vehicle under consideration; then it's possible to simulate the measurements process by IMUs and cameras. Finally measurements and dynamics are inputs for the estimation process, the core of the SLAM.

Basically the software is composed by two modules; the vehicle dynamic and control module, and the measurements and estimation module (navigation).

The Estimation Process Module simulate the basic requirements for SLAM:

- the acquisition of IMU, by loading dynamical data (airship state vector)
- camera range measurements
- filtering process

These measurements are used by the prediction and update stages of the Extended Kalman Filter which estimate vehicle kinematics and the location of features in a local referenced map. All simulation data are loaded into the SEM workspace by including the proper dynamics data file.

The navigation module implements the data flow reported in the illustration below and deal with the following tasks:

⁷The airship, in a hovering flight at low altitudes, could be considered solidal to atmosphere. This means that any local geodetical frame could be considered as inertial because vehicle motion could be considered not influenced by the planet rotation.

- map management;
- range, bearing measurement;
- IMU measurement;
- feature management;
- range measurement prediction and its Jacobians computing;
- EKF update stage;
- EKF prediction stage and Jacobian computing;

6.2.2 Map Management

In §5.6 we have shown that is not necessary to perform a full SLAM update when working in a local area, and this assumption reduces the computational cost of the SLAM algorithms, so while the vehicle operates in a local area all the information gathered can be maintained with a cost complexity proportional to the number of landmarks in this area.

The question is now the selection of local areas. One convenient approach consists of dividing the global map into rectangular regions with size at least equal to the range of the external sensor. The map management method is presented in Fig.6.10.

When the vehicle navigates in the region R the compressed filter includes in the group $\mathbf{x}_A$ the vehicle states and all the states related to landmarks that belong to region R and its *eight neighboring regions*. This implies that the local states belong to 9 regions, each of size greater than the range of the external sensor. The vehicle will be able to navigate inside this region using the compressed filter; a full up date will only be required when the vehicle leaves the central region R . Every time the vehicle moves to a new region, the active state group $\mathbf{x}_A$, changes to those states that belong to the new region R and its adjacent regions. The active group always includes the vehicle states. In addition to the swapping of the $\mathbf{x}_A$ states, a global update (see §5.6) is also required at full SLAM algorithm cost. This means that every time we need a full update the states and covariance matrix includes all features in the global map and the computational cost is the same of the classical EKF implemetation.

If the airship has to fly in an completely unknown enviroment, we have to plan the global area size and divide it into subregion of known size. The vehicle position is known in the local area during the flight by SLAM, so it's possible to determine if a landmark belong or not to a the local area, with a level of uncertainty, by range measurement informations. It is not critical if the landmark belongs to this region or a close connected region. When the vehicle is located in the boundary of a region it could associate only featured inside that region and any new feature is rejected until it enter in the new region. This approach is good since the side length of the regions are bigger that the range of the external sensor; observation could be discharged if they cannot be associated with any local landmarks.

When the vehicle is navigating inside the central region R the filter will maintain its position and build the local map with a SLAM cost of relatively to the number of features in the local area formed by R and the eight neighbour regions. A full SLAM update is only required when the vehicle leaves the region.

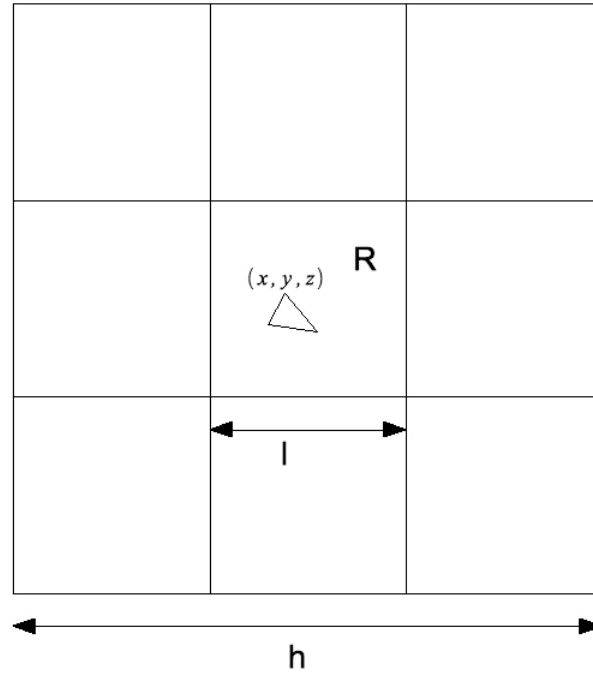


Figure 6.10: Submap management. When the Airship navigate in the region R the compressed filter include in the state $\mathbf{x}_A$ the vehicle states and all states related to landmarks that belong to R and its eight neighboring regions. A full update will only be required when the vehicle leaves the region R and the active group state $\mathbf{x}_A$ changes to those states that belong to the new region and its neighboring.

6.2.3 Computational Cost

The total computational requirement for this algorithm is of $O(N_A^2)$ and the cost of the update when the vehicle leaves the local area is of $O(N_A^2 N_B^2)$. Provided that the vehicle remains for a period of time in a given area, the computational saving will be considerable. This has important implications since in many applications it will allow the exact implementation of SLAM in very large areas. This will be possible with the appropriate selection of local areas. The system evaluates the location of the vehicle and the landmark of the local map continuously at the cost of a local SLAM. Although a full update is required at a transition, this update can be implemented as a parallel task. The only states that need to be fully updated are the new states in the new local area. A selective update could then be done only to those states while the full update for the rest of the map runs as a background task with lower priority. These results are important since it demonstrates that even in very large areas the computational limitation of SLAM can be overcome with the compression algorithm and appropriate selection of local areas.

6.2.4 IMU Measurement

IMU measurement are obtained by the Dynamics Module of the software (DCM). The module provides a set of dynamical and kinematic quantities among which those necessary to the navigation equation (4.13).

For simplicity if IMU is placed on the CV of the vehicle, ${}^b\mathbf{r}_S = 0$, the model equation is simply,

$${}^b\dot{\mathbf{v}} = {}^b\mathbf{S} + {}^b\mathbf{R}_i {}^i\mathbf{g} - \boldsymbol{\omega} \times {}^b\mathbf{v} + \boldsymbol{\sigma}_{imu} \quad (6.42)$$

where σ_{imu} is the measurement gaussian noise and the specific force ${}^b\mathbf{S}$ is known from dynamics. The gravity vector is known in the inertial frame and then transformed into the body frame. Finally the output of the IMU system are the vector quantities ${}^b\dot{\mathbf{v}}$ and $\boldsymbol{\omega}$ with added noise. From IMU specification in Table 4.1 we can determine the variance of the gaussian noise σ_{imu}

$$\sigma_{IMU} = [\sigma_{Acc} \ \sigma_{Gyro}]^T = [1.5 \cdot 10^{-3} \text{ m/s}^2 \ 1.5 \cdot 10^{-4} \text{ }^\circ/\text{s}]^T \quad (6.43)$$

It's interesting to observe that the effect of this noise afflicts the integration process, to determine velocity position and attitude, leading to large errors; in Fig. 6.11 it's shown the situation in which the state of the system it's only propagate by IMU inputs without any feature measurement and estimation process.

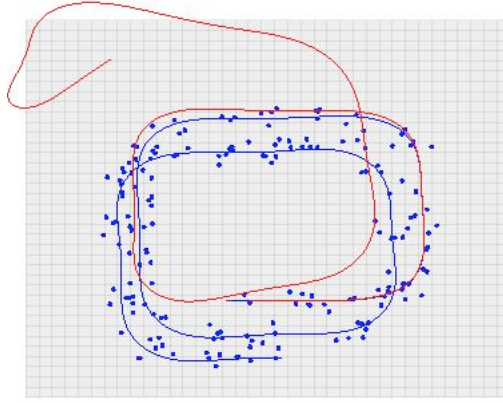


Figure 6.11: Without filtering process IMU measurement noise leads to large errors

6.2.5 EKF Prediction Stage

The 3D problem is quite different from 2D, and in generally is more complex in Jacobians calculations. We start here from computation of Jacobians for the prediction stage.

In continuous time the system, vehicle and features, is described by

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \quad (6.44)$$

$$\dot{\mathbf{m}} = \mathbf{0} \quad (6.45)$$

where $\mathbf{x}$ is the state vector of the vehicle and $\mathbf{m}$ the vector containing the features position.

$$\mathbf{x} = [{}^b\boldsymbol{\theta}, {}^b\mathbf{v}, {}^i\mathbf{p}]^T \quad (6.46)$$

using the *Euler step forwarding* we can propagate the vehicle state vector from t_0 to t_1

$$\mathbf{x}(t_1) = \mathbf{x}(t_0) + \mathbf{f}(t_0, \mathbf{x}(t_0) + \mathbf{u}(t_0))\Delta t \quad (6.47)$$

where

$$\mathbf{f}(t_0, \mathbf{x}(t_0) + \mathbf{u}(t_0)) = \begin{bmatrix} {}^b\dot{\boldsymbol{\theta}}(t_0) \\ {}^b\dot{\mathbf{v}}(t_0) \\ {}^i\dot{\mathbf{p}}(t_0) \end{bmatrix} = \begin{bmatrix} \mathbf{H} {}^b\boldsymbol{\omega} \\ {}^b\mathbf{F} + {}^b\mathbf{R}_i {}^i\mathbf{g} \\ {}^b\mathbf{R}_i {}^i\mathbf{p} \end{bmatrix} \quad (6.48)$$

there $\mathbf{p}$ is the vehicle position vector, and $\boldsymbol{\theta} = (\phi, \theta, \psi)$ is the attitude and $\boldsymbol{\omega}$ the vehicle angular velocity.

It's necessary to compute the jacobians in continuous time with respect $\mathbf{x}$ and $\mathbf{u}$ to get them in

the discrete time by equations (5.11).
Jacobian of the state function is

$$\nabla_{\mathbf{x}} \mathbf{f} = \nabla_{[\theta, \mathbf{v}, \mathbf{p}]} \begin{bmatrix} {}^b \dot{\boldsymbol{\theta}}(t_0) \\ {}^b \dot{\mathbf{v}}(t_0) \\ {}^i \dot{\mathbf{p}}(t_0) \end{bmatrix} \quad (6.49)$$

Expliciting the single parts of the derivative process we have⁸

$$\nabla_{[\theta, \mathbf{v}, \mathbf{p}]} \dot{\boldsymbol{\theta}} = \nabla_{[\theta, \mathbf{v}, \mathbf{p}]} (\mathbf{H} \boldsymbol{\omega}) \quad (6.50)$$

Now

$$\frac{\partial(\mathbf{H} \boldsymbol{\omega})}{\partial \mathbf{v}} = \frac{\partial(\mathbf{H} \boldsymbol{\omega})}{\partial \mathbf{p}} = \mathbf{0} \quad (6.51)$$

since

$$\begin{aligned} \frac{\partial(\mathbf{H} \boldsymbol{\omega})}{\partial \boldsymbol{\theta}} &= \frac{\partial(\mathbf{H} \boldsymbol{\omega})}{\partial(\mathbf{H}, \boldsymbol{\omega})} \cdot \frac{\partial(\mathbf{H}, \boldsymbol{\omega})}{\partial \boldsymbol{\theta}} \\ &= \begin{bmatrix} \frac{\partial(\mathbf{H}, \boldsymbol{\omega})}{\mathbf{H}} & \frac{\partial(\mathbf{H}, \boldsymbol{\omega})}{\boldsymbol{\omega}} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{H}}{\partial \boldsymbol{\theta}} \\ \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\theta}} \end{bmatrix} \end{aligned} \quad (6.52)$$

and being $\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{\theta}} = \mathbf{0}$ we get

$$\nabla_{\mathbf{x}} \dot{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial(\mathbf{H} \boldsymbol{\omega})}{\partial \mathbf{H}} \frac{\partial(\mathbf{H})}{\partial \boldsymbol{\theta}} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (6.53)$$

The two terms of the last product are easy to compute (see Appendix B).
For the velocity we have

$$\nabla_{[\theta, \mathbf{v}, \mathbf{p}]} \dot{\mathbf{v}} = \nabla_{[\theta, \mathbf{v}, \mathbf{p}]} ({}^b \mathbf{R}_i {}^i \mathbf{g} - \mathbf{S}_{\boldsymbol{\omega}} \mathbf{v}) \quad (6.54)$$

where $\mathbf{S}_{\boldsymbol{\omega}} = \mathbf{S}(\boldsymbol{\omega})$ is the skew operator applied to $\boldsymbol{\omega}$ ⁹
Now,

$$\frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{\partial \mathbf{v}} = \frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{\partial \mathbf{p}} = \mathbf{0} \quad (6.55)$$

and

$$\begin{aligned} \frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{\partial \boldsymbol{\theta}} &= \frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{\partial({}^b \mathbf{R}_i, {}^i \mathbf{g})} \cdot \frac{\partial({}^b \mathbf{R}_i, {}^i \mathbf{g})}{\partial \boldsymbol{\theta}} \\ &= \begin{bmatrix} \frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{{}^b \mathbf{R}_i} & \frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{{}^i \mathbf{g}} \end{bmatrix} \begin{bmatrix} \frac{\partial {}^b \mathbf{R}_i}{\partial \boldsymbol{\theta}} \\ \frac{\partial {}^i \mathbf{g}}{\partial \boldsymbol{\theta}} \end{bmatrix} \end{aligned} \quad (6.56)$$

and being $\frac{\partial {}^b \mathbf{R}_i}{\partial {}^i \mathbf{g}} = \mathbf{0}$ we get

$$\frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{\partial \boldsymbol{\theta}} = \frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{\partial {}^b \mathbf{R}_i} \frac{\partial {}^b \mathbf{R}_i}{\partial \boldsymbol{\theta}} \quad (6.57)$$

⁸If any upper script is omitted we refer to body frame vector representation

⁹Remember that $\boldsymbol{\omega} \times \mathbf{v} = \mathbf{S}_{\boldsymbol{\omega}} \mathbf{v}$.

in the same way is easy to prove that

$$\frac{\partial \mathbf{S}_\omega \mathbf{v}}{\partial \boldsymbol{\theta}} = \mathbf{0} \quad (6.58)$$

$$\frac{\partial \mathbf{S}_\omega \mathbf{v}}{\partial {}^i \mathbf{p}} = \mathbf{0} \quad (6.59)$$

$$\frac{\partial \mathbf{S}_\omega \mathbf{v}}{\partial \mathbf{v}} = \frac{\partial \mathbf{S}_\omega \mathbf{v}}{\partial \mathbf{v}} \cdot \mathbf{1} \quad (6.60)$$

so

$$\nabla_{\mathbf{x}} \dot{\mathbf{v}} = \begin{bmatrix} \frac{\partial({}^b \mathbf{R}_i {}^i \mathbf{g})}{\partial {}^b \mathbf{R}_i} \frac{\partial {}^b \mathbf{R}_i}{\partial \boldsymbol{\theta}} & - \frac{\partial \mathbf{S}_\omega \mathbf{v}}{\partial \mathbf{v}} \cdot \mathbf{1} & \mathbf{0} \end{bmatrix} \quad (6.61)$$

Similarly for the position vector we have

$$\nabla_{[\boldsymbol{\theta}, \mathbf{v}, \mathbf{p}]}({}^i \dot{\mathbf{p}}) = \nabla_{[\boldsymbol{\theta}, \mathbf{v}, \mathbf{p}]}({}^i \mathbf{R}_b {}^b \mathbf{v}) \quad (6.62)$$

$$\begin{aligned} \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial \boldsymbol{\theta}} &= \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial({}^i \mathbf{R}_b, {}^b \mathbf{v})} \cdot \frac{\partial({}^i \mathbf{R}_b, {}^b \mathbf{v})}{\partial \boldsymbol{\theta}} \\ &= \begin{bmatrix} \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^i \mathbf{R}_b} & \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^b \mathbf{v}} \end{bmatrix} \begin{bmatrix} \frac{\partial {}^i \mathbf{R}_b}{\partial \boldsymbol{\theta}} \\ \frac{\partial {}^b \mathbf{v}}{\partial \boldsymbol{\theta}} \end{bmatrix} \end{aligned} \quad (6.63)$$

$$= \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^i \mathbf{R}_b} \frac{\partial {}^i \mathbf{R}_b}{\partial \boldsymbol{\theta}} \quad (6.64)$$

and

$$\begin{aligned} \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^b \mathbf{v}} &= \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial({}^i \mathbf{R}_b, {}^b \mathbf{v})} \cdot \frac{\partial({}^i \mathbf{R}_b, {}^b \mathbf{v})}{\partial {}^b \mathbf{v}} \\ &= \begin{bmatrix} \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^i \mathbf{R}_b} & \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^b \mathbf{v}} \end{bmatrix} \begin{bmatrix} \frac{\partial {}^i \mathbf{R}_b}{\partial {}^b \mathbf{v}} \\ \frac{\partial {}^b \mathbf{v}}{\partial {}^b \mathbf{v}} \end{bmatrix} \end{aligned} \quad (6.65)$$

$$= \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^b \mathbf{v}} \quad (6.66)$$

and finally we get

$$\frac{\partial {}^i \dot{\mathbf{p}}}{\partial {}^b \mathbf{v}} = \begin{bmatrix} \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^i \mathbf{R}_b} \frac{\partial {}^i \mathbf{R}_b}{\partial \boldsymbol{\theta}} & \frac{\partial({}^i \mathbf{R}_b {}^b \mathbf{v})}{\partial {}^b \mathbf{v}} & \mathbf{0} \end{bmatrix} \quad (6.67)$$

Now we compute the continuous time Jacobian of the state function with respect to the input vector $\mathbf{u} = [\boldsymbol{\omega}, \dot{\mathbf{v}}]$, where $\dot{\mathbf{v}}$ is the specific acceleration measured by IMU, it's easy to prove that

$$\nabla_{[\boldsymbol{\omega}, \mathbf{f}]}(\mathbf{H}\boldsymbol{\omega}) = \begin{bmatrix} \frac{\partial(\mathbf{H}\boldsymbol{\omega})}{\partial \boldsymbol{\omega}} & \mathbf{0} \end{bmatrix} \quad (6.68)$$

$$\nabla_{[\boldsymbol{\omega}, \mathbf{f}]}(\dot{\mathbf{v}}) = \begin{bmatrix} -\frac{\partial \mathbf{S}_\omega \mathbf{v}}{\partial \boldsymbol{\omega}} \frac{\partial \mathbf{S}_\omega}{\partial \boldsymbol{\omega}} & \mathbf{1} \end{bmatrix} \quad (6.69)$$

$$\nabla_{[\boldsymbol{\omega}, \mathbf{f}]}({}^i \dot{\mathbf{p}}) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (6.70)$$

With the knowledge of continuous time jacobian matrix we could derive the discrete time ones by (5.11) and so the state transition function to propagate the vehicle state vector from eq.(5.15). Moreover, the jacobian matrix with respect the entire state of the system could be computed using (5.32) and finally the prediction for the covariannce matrix by eq.(5.33). In this stage also CEKF auxiliary matrices have to be computed by (5.54).

6.2.6 Stereo Camera Measurement

Camera measurement are crucial for the updating stage of the filter, here we examine how measures are built and managed.

We use the camera proposed in §4.2.7

	px	pixel size [10^{-6} m]	sensor size [mm]	FoV
sensor x	2050	3.45	7.07	61.03
sensor y	2450	3.45	8.45	70.32

- focal: 6.00 [mm]
- baseline: 2.00 [m]

where the parameters to use in the navigation software are reported in Table 4.2. It's reasonable to stay in a range of altitude from 30-100 m in order to have limited uncertainties.

In the 3D case, we extend what said for a 2D system, the measured quantities are:

- range measurement z_s from camera triangulation along the sensor optical axis ${}^s\hat{\mathbf{k}}$;
- bearing measurement, $\tan^{-1}(x_s/z_s)$;
- elevation measurement, $\tan^{-1}(y_s/z_s)$;

so the range-bearing-elevation measurement vector is chosen as

$$\mathbf{y} = [x_s/z_s, y_s/z_s, z_s] + \boldsymbol{\sigma}_s \quad (6.71)$$

and can be computed when a point coordinates $[x_s, y_s, z_s]$ are known on the sensor frame.

The measurement model is usually expressed in the inertial frame by

$${}^i\hat{\mathbf{r}} = {}^i\mathbf{m} - {}^iT_v = \mathbf{h}(\mathbf{x}_a) \quad (6.72)$$

and so in the vehicle frame

$${}^b\mathbf{r} = {}^b\mathbf{R}_i({}^i\mathbf{m} - {}^iT_b) \quad (6.73)$$

For filtering process we need the Jacobian of the measurement model with respect to the system state

$$\nabla_{\mathbf{x}_a} \mathbf{h}(\mathbf{x}_a) \quad (6.74)$$

It's easy to show that hold the following equations

$$\frac{\partial({}^b\mathbf{R}_i{}^i\mathbf{r})}{\partial\boldsymbol{\theta}} = \left[\frac{\partial({}^b\mathbf{R}_i{}^i\mathbf{r})}{\partial{}^b\mathbf{R}_i} \frac{\partial{}^b\mathbf{R}_i}{\partial\boldsymbol{\theta}} \right] \quad (6.75)$$

$$\frac{\partial({}^b\mathbf{R}_i{}^i\mathbf{r})}{\partial\mathbf{v}} = \mathbf{0} \quad (6.76)$$

$$\frac{\partial({}^b\mathbf{R}_i{}^i\mathbf{r})}{\partial{}^i\mathbf{p}} = -{}^b\mathbf{R}_i \quad (6.77)$$

$$\frac{\partial({}^b\mathbf{R}_i{}^i\mathbf{r})}{\partial{}^i\mathbf{m}} = {}^b\mathbf{R}_i \quad (6.78)$$

so we have for the k -th features observed the matrix

$$\nabla_{\mathbf{x}_a} {}^v\mathbf{r} = \begin{bmatrix} \frac{\partial({}^b\mathbf{R}_i {}^i\mathbf{r})}{\partial {}^b\mathbf{R}_i} \frac{\partial {}^b\mathbf{R}_i}{\partial \boldsymbol{\theta}} & \mathbf{0}_{3 \times 3} & - {}^b\mathbf{R}_i & \underbrace{\mathbf{0}_{3 \times 3} \cdots {}^b\mathbf{R}_i \cdots \mathbf{0}_{3 \times 3}}_{k\text{-feature}} \end{bmatrix} \quad (6.79)$$

$$(6.80)$$

Measures are taken in the sensor frame so it's necessary to transform the jacobian matrix

$${}^s\mathbf{R}_b(\nabla_{\mathbf{x}_a} {}^b\mathbf{r}) \quad (6.81)$$

and if ϕ is the function that maps 3D points in the sensor frame in a range-bearing-elevation vector the final expression for the measurement jacobian matrix for each feature

$$\nabla_{{}^s\mathbf{r}} \phi \cdot {}^s\mathbf{R}_b(\nabla_{\mathbf{x}_a} {}^b\mathbf{r}) \quad (6.82)$$

6.2.7 Data association and State Augmentation

How can we verify if a feature is new one or a previous observed one? We need a strategy to implement the association between observations and the map in the system state vector.

The traditional data association metric in target tracking is the normalized innovation squared, which is the error between predicted and actual observation normalized by error covariance. A threshold on this quantity is known as the innovation gate τ . The innovation gate is also used for data association in stochastic SLAM, and permits landmarks to be distinguished by their position. However, landmarks that are too close in position give rise to ambiguous associations. Batch association methods, where multiple measurements are considered simultaneously, greatly reduce this uncertainty but ambiguous associations cannot be completely eliminated; in that case the measure is ignored.

If $\mathbf{y}_j$ is a new measure, we compute the innovation vector

$$\boldsymbol{\gamma}_{i,j} = \mathbf{y}_j - \mathbf{h}(\mathbf{x}_v, \mathbf{m}_i) \quad (6.83)$$

where $\mathbf{h}(\mathbf{x}_v, \mathbf{m}_i)$ is the observation prediction for all features present in the local map. Therefore we compute the innovation covariance

$$\mathbf{S}_i(k) = \nabla_{\mathbf{x}_a} \mathbf{h}(\mathbf{x}_v, \mathbf{m}_i) P(k|k-1) \nabla_{\mathbf{x}_a} \mathbf{h}(\mathbf{x}_v, \mathbf{m}_i)^\top + \mathbf{R}(k) \quad (6.84)$$

We define the *normalized innovation square*, $n_{k,j}$

$$n_{i,j} = \boldsymbol{\gamma}_{i,j}^\top \mathbf{S}_i(k)^{-1} \boldsymbol{\gamma}_{i,j} \quad (6.85)$$

and the *normalized distance*

$$d_{i,j} = n_{i,j} + \log(\det \mathbf{S}_i(k)) \quad (6.86)$$

If

$$\min_i d_{i,j} < \tau \quad (6.87)$$

the measure is associated and if

$$\min_i n_{i,j} > \tau \quad (6.88)$$

the measure represent a new feature. The τ threshold have to be determinate by a tuning phase.

All associated features are used to compute the Kalman gain and so the updated system state, new features instead have to be initialized, added to the state vector and used to compute the augmented covariance matrix.

A feature initialization model function is needed to give a first estimation $\mathbf{m}_i(k)$ to include in the state vector in the prediction stage. The state covariance matrix must be *augmented* since the initial feature estimates depend on the current vehicle state and so correlated with the rest of the estimated features as described in §5.4.1.

From a range-bearing-elevation measure we can calculate the feature position in the map (inertial) frame with the transformation

$${}^i\mathbf{m}_j = {}^i\mathbf{R}_b {}^b\mathbf{R}_s \phi^{-1}(\mathbf{y}_j) \quad (6.89)$$

So we can relate the new measure with the state of the vehicle by

$${}^i\mathbf{m}_j = {}^i\mathbf{R}_b {}^b\mathbf{m}_j = \mathbf{g}_j(\mathbf{x}_v(k+1|k), \mathbf{y}_j) \quad (6.90)$$

where we have defined the initialization function $\mathbf{g}(\cdot)$.

To compute the augmented state covariance from eq. (5.44) we need to know the jacobian of the initialization function with respect to the vehicle state

$$\nabla_{\mathbf{v}} \mathbf{g}_j$$

and with respect to the measure

$$\nabla_{\mathbf{y}_j} \mathbf{g}_j$$

It's easy to prove that

$$\frac{\partial({}^i\mathbf{R}_b {}^b\mathbf{m}_j)}{\partial \boldsymbol{\theta}} = \frac{\partial({}^i\mathbf{R}_b {}^b\mathbf{m}_j)}{\partial {}^i\mathbf{R}_b} \frac{\partial {}^i\mathbf{R}_b}{\partial \boldsymbol{\theta}} \quad (6.91)$$

$$\frac{\partial({}^i\mathbf{R}_b {}^b\mathbf{m}_j)}{\partial {}^b\mathbf{v}} = \mathbf{0}_{3 \times 3} \quad (6.92)$$

$$\frac{\partial({}^i\mathbf{R}_b {}^b\mathbf{m}_j)}{\partial {}^i\mathbf{p}} = \mathbf{1}_{3 \times 3} \quad (6.93)$$

and

$$\nabla_{\mathbf{y}_j} \mathbf{g} = {}^i\mathbf{R}_b {}^b\mathbf{R}_s \nabla_{\mathbf{y}_j} \phi^{-1} \quad (6.94)$$

6.2.8 Update Stage

Both prediction and measurements quantities are now computable. From prediction stage we can compute the prediction of the covariance matrix, and by landmark's measures in the local area we get the innovation vector and Jacobian of the measurement equation. It's important to remember that if measurement are available and associated the local state could be updated, otherwise the new features have to be initialized.

The innovation covariance and the Kalman gain can be calculated also CEKF auxiliary matrix have to be updated updated by eq.(5.57); the local update stage gives the estimation of the local state vector $\mathbf{x}_A$.

When the vehicle exits from the local region a full SLAM update is performed: computation of the global covariance matrix and the estimate of global state $\mathbf{x}_B$. Essential steps of the classical EKF process (in a local map) could be summarized in fig. 5.3 while the flow of the MASS estimation module is shown in fig. 6.12.

State Estimation Module (SEM) (Global Data Flow)

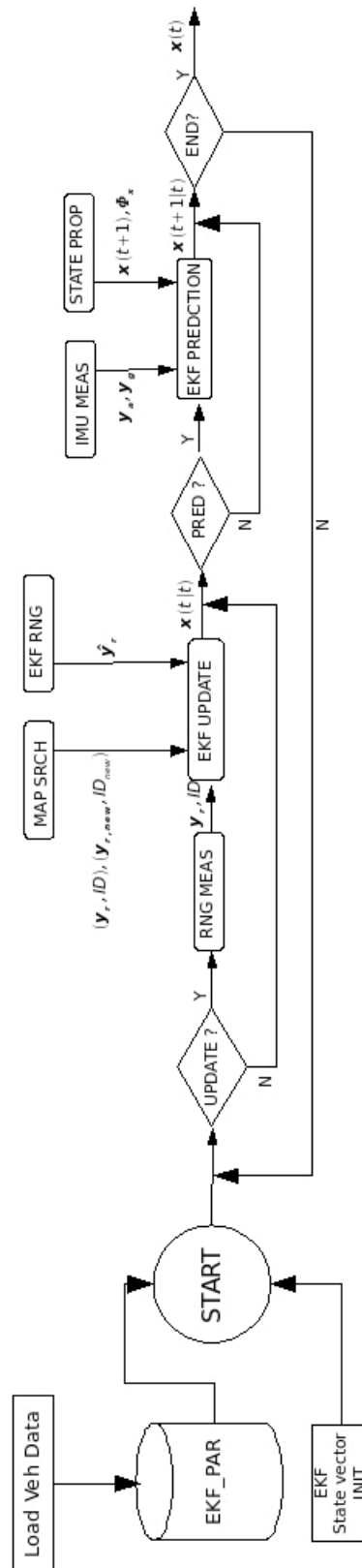


Figure 6.12: Flow of the MASS software estimation module which implements SLAM navigation

6.3 Simulations results

In this section are reported some numerical results obtained with MASS software.

SLAM navigation has been simulated on Titan satellite since it is one of the only other bodies in the solar system with a dense atmosphere (as Venus). The Cassini-Huygens mission has revealed a unique environment, showing methane lakes, river channels and drainage basins, sands dunes, cryovolcanoes and terraces. Furthermore Titan observation both from ground and space have shown the presence of complex weather phenomena, including clouds and storms. All these different features have increased the scientific interest in a possible follow-up mission to Titan using a mongolfier or airship. In particular airships are very interesting vehicles because of the reduced required power, the extended duration and the extremely long distances they can travel. Furthermore they are able to execute regional surveys, transport and deploy scientific instrumentation and deploy insitu facilities and perform sampling of particular interesting sites. Titan is the largest moon of Saturn, with a radius of 2575km; the atmospheric density at surface is around 5.54kg/m^3 , and a composition of 95% nitrogen, 3% methane and 2% argon. The surface pressure is approximately 1.35m/s^2 . The surface temperature is around 93K . The winds measured by the Huygens probe at the surface are mainly zonal with speeds around 0.5m/s and less than 1m/s at altitudes below 10km . but even with 1m/s velocity the airship is able to cover $\approx 80\text{km}$ in 24hrs ; exceeding far the mobility of any foreseeable rover. The airship will carry on all the instruments needed to sample Titan's atmosphere, surface, possible methane lakes-rivers, use multi-spectral imagers for surface reconnaissance; to take close up surface images, take core samples and deploy seismometers during landing phase. Both active and passive broadband remote sensing techniques can be used for surface topography, winds and composition measurements. For all these science needs there are several capabilities that an airship for exploration of Titan must have:

- vehicle safing;
- accurate and robust autonomous flight control;
- deployment, long traverses, hovering and touch-and-go surface sampling;
- spatial mapping;
- self localisation;
- advanced perception of hazards and target recognition;
- tracking and visual servoing

The airship is characterised by having several flight modes: take-off/landing, hovering/station-keeping, loitering, ascent/descent, high-speed cruise, low-speed cruise. Due to the presence of the cloud layer surface imaging is limited to low-resolution radar and infrared maps. The airship can therefore be considered as a relatively stable aerial platform from where execute a high resolution imaging over large surface areas by flying below the clouds for long duration missions. Furthermore surface sampling from the airship requires both hovering capability and low ground relative speed. Another interesting point that must be outlined is the fact that a long range airship does not require at all precision landing on Titan because it is able to fly to any point of interest on the surface. The need for autonomous operation is motivated by the time transmission needed by a command signal to reach Titan from Earth and the echo from the airship indicating that the telecommand has been correctly executed to reach back the Earth. Such large latencies do not allow the airship to be operated directly from Earth and to have a

fast response for obstacle avoidance and sample collection. These are mainly the reasons why the airship must have very robust autonomous control functions.

The general goal of the aerial flight segment from an operational perspective is to cover as much of the surface of Titan as possible (global survey) while stopping to concentrate on interesting areas. The airship will proceed along an assigned path, moving with the wind but employing its propellers to keep the path somewhat independent of the wind. This path can be uploaded to the airship on a periodic basis more or less frequently depending on the current activity and its requirement for ground operations input. The assigned trajectory will be corrected in real-time by the on-board navigation system using the propellers. Altitude profile along the path will also be part of the assigned trajectory and will be controlled. There will be two methods for determining sites of scientific interest which will require the airship to make a dedicated survey either by hovering over the area (if possible, given the wind speed), circling about it, or initiating a ground interaction. Prior to the uploading of the assigned airship path, sites along the path can be designated as areas of interest that will require concentrated effort by the airship.

In this study we are going to focus mainly on the following scenarios:

- transfer cruise, in which the airship has simply to navigate from one reference point to another. In this case the primary purpose is to navigate toward a specific site to perform the planned mission operations; this phase can also be seen as a recon phase needed to navigate safely around Titan and identify prime sites for surface sampling and analysis. In this phase it is not necessary to map features with high accuracy not reachable with the mono camera SLAM system used by the navigation system at high altitudes ($\geq 200m$) .
- flying at low altitudes ($< 100m$) which will enable the mapping of the underneath landscape with a stereo vision system. Purpose of this phase is the mapping of the terrain with an accuracy sufficient to recognise and localise landmarks on the surface and to know the position and attitude of the vehicle itself within the map because correlation of position with the science measurements is essential to increase the validity and fidelity of the data. Hazard detection and consequently landing site selection will be possible outcomes of this scenario.

The proposed vision system is the same in both scenarios (see Table 4.2 for camera's reference.) The images "grab" process occurs at a frequency of 0.5 hz and so the update stage of the filter, it's a reasonable time considering the performances of a FPGA based image processing hardware on board. The IMU frequency is 100 hz.

Camera errors in range-bearing-elevation measurements are calculated from Table 4.2 considering an altitude of 80 m:

	Rng [m]	Brg [deg]	Elev [deg]
σ	1.02	0.5	0.7

Table 6.2: Uncertainties in range-bearing-elevation measurement

6.3.1 Operations Flight

If the primary task is mapping an unknown area to build a digital terrain map we present here the results of two simulations. The landscape with dimension of 800x800 m and the airship trajectory is shown in fig. 6.13; IMU input data are provided by airship Titan's dynamical simulations.

The airship altitude in the first simulation is 80 m; it's quite a limit altitude to use the stereo system and so it's interesting to test the estimation process, results are very encouraging. In the second simulation the airship altitude is 50 m, a more comfortable altitude for a stereo vision system.

6.3.1.1 80 m altitude flight

Estimation of vehicle parameters are reported: attitude (Fig. 6.15, 6.16, 6.17), velocity (Fig. 6.18, 6.19, 6.20) and position in the inertial frame (Fig. 6.21, 6.22, 6.23). Each figure is splitted; the first show the parameter estimation where dashed lines reports the parameter trend from dynamics; the second figure show the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

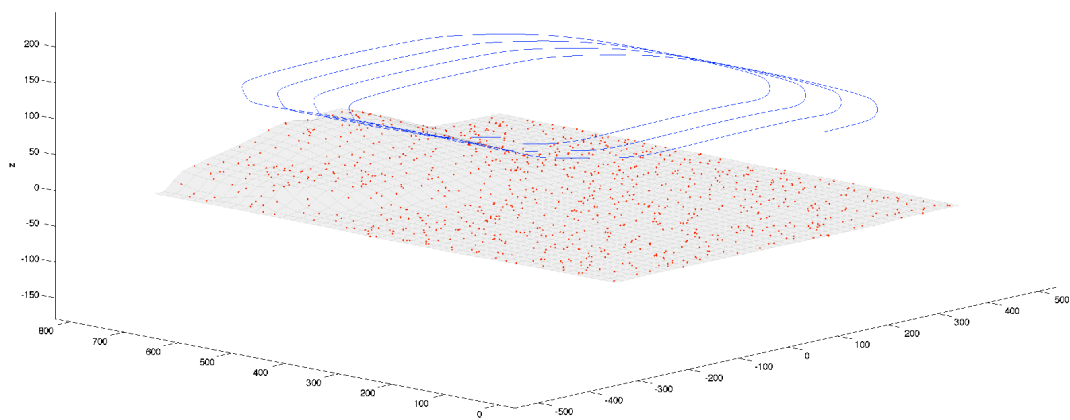


Figure 6.13: Landscape and trajectory to terrain mapping tasks

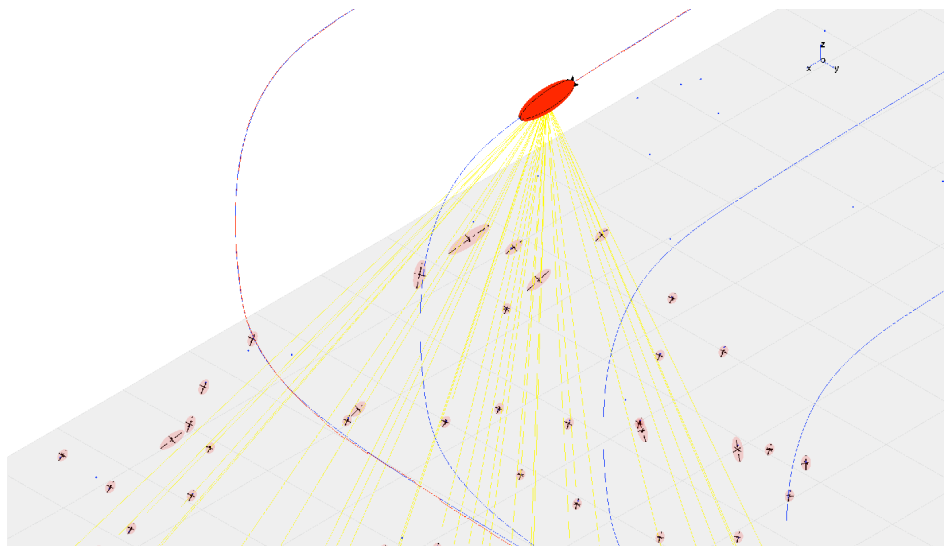


Figure 6.14: A phase of the simulation, the software camera follow the airship so the user could check the estimation process. A view like this is useful to tune filter parameters

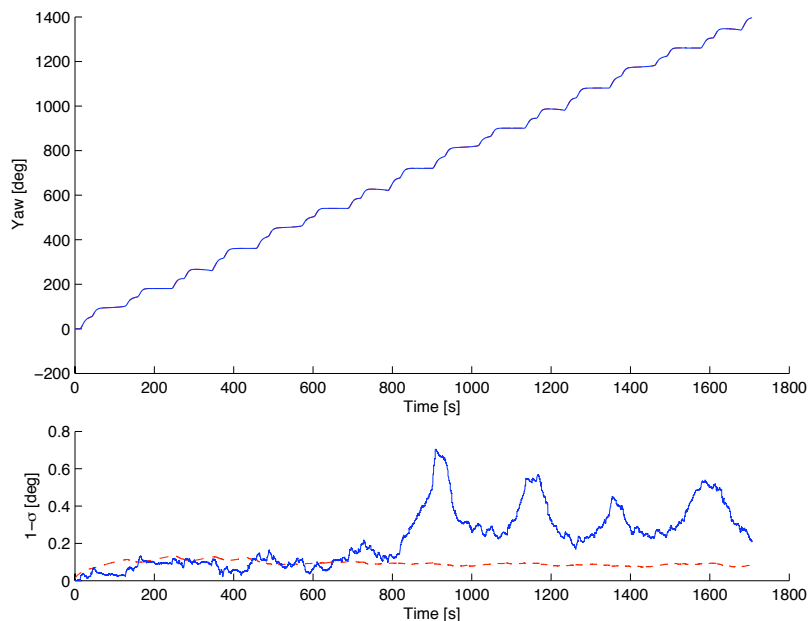


Figure 6.15: Yaw angle estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

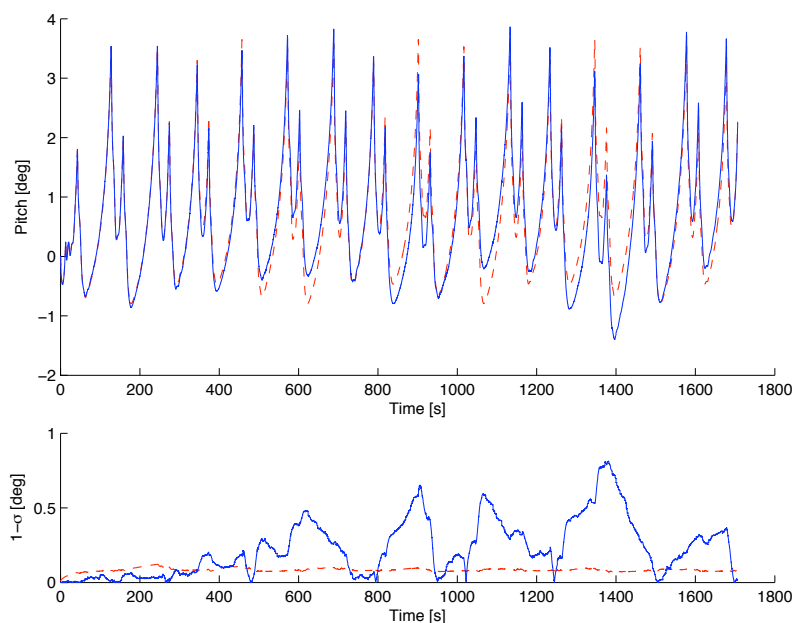


Figure 6.16: Pitch angle estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

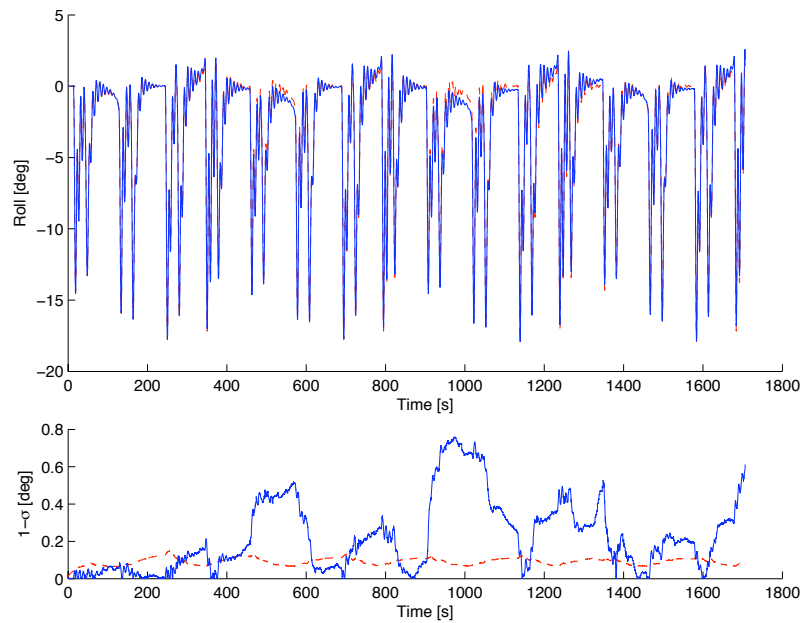


Figure 6.17: Roll angle estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

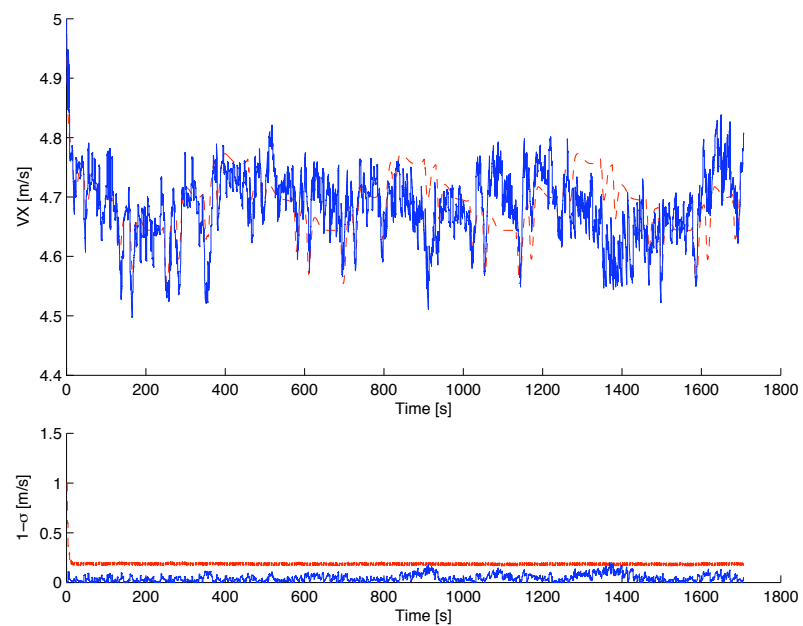


Figure 6.18: Longitudinal velocity estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

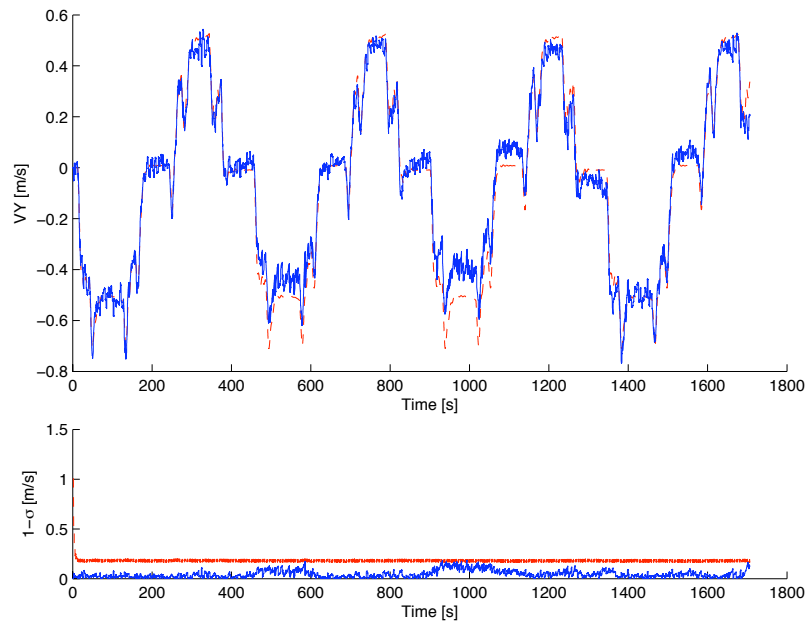


Figure 6.19: Transversal velocity estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

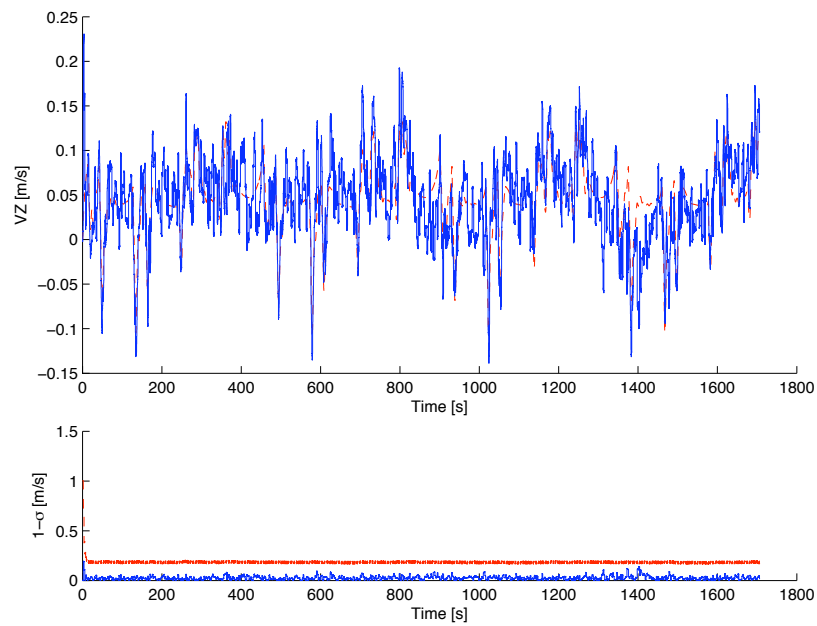


Figure 6.20: Vertical velocity estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

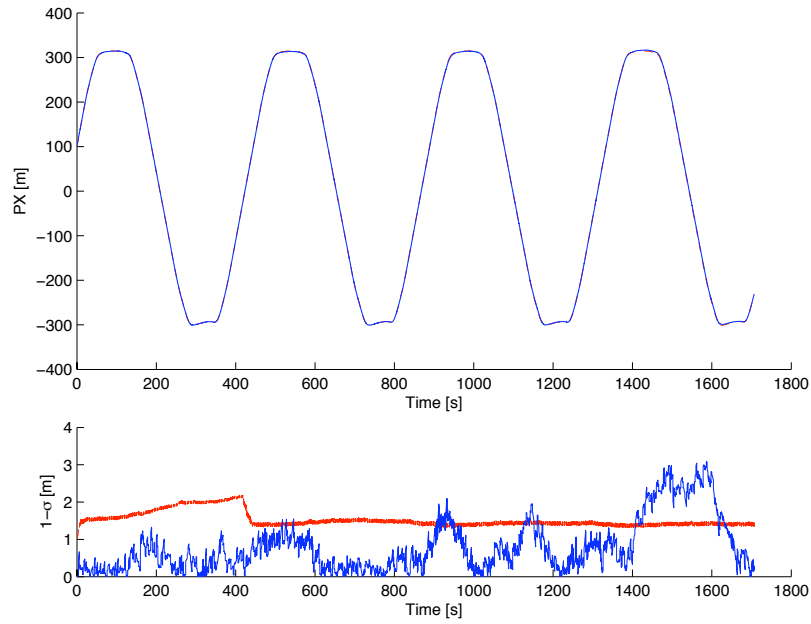


Figure 6.21: X-position estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

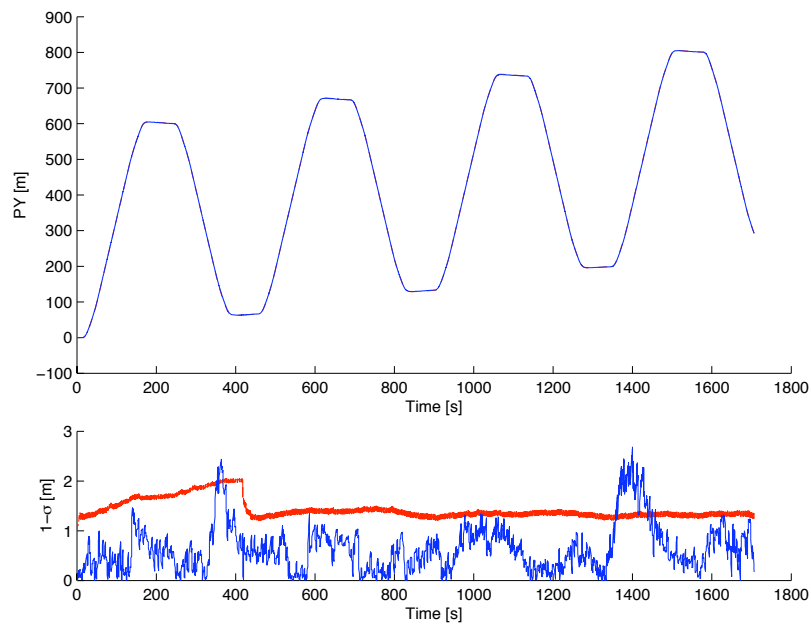


Figure 6.22: Y-position estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

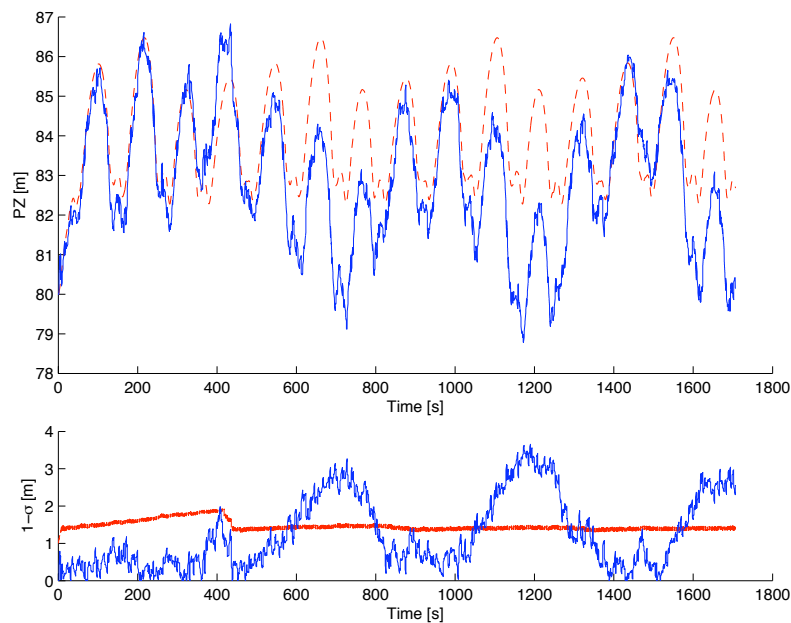


Figure 6.23: Z-position estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

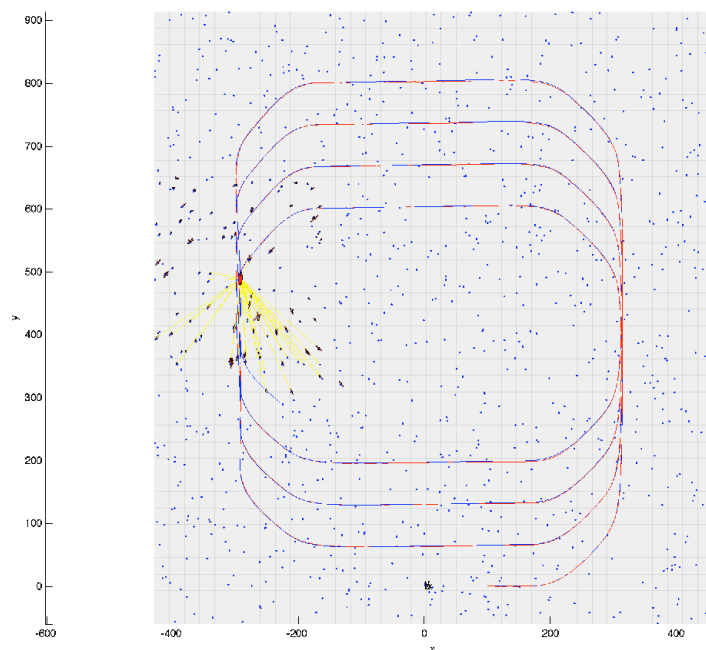


Figure 6.24: Estimated (red) and true (blue) trajectories; is visible (yellow) the camera FOV and the features measured

Error on attitude are below 0.8° , error on velocity are under 0.1 m/s and errors on the inertial position of the airship are under 3.5 m.

In Fig. 6.24 the two trajectories estimated (red), and by dynamics (blue) are shown. Features are correlated with the airship state and with other features in the map, so the estimation of a

feature improves by loop-closure effect. This is evident in Fig 6.25 where it's shown the effect of filter update in the feature position estimate and on its variance.

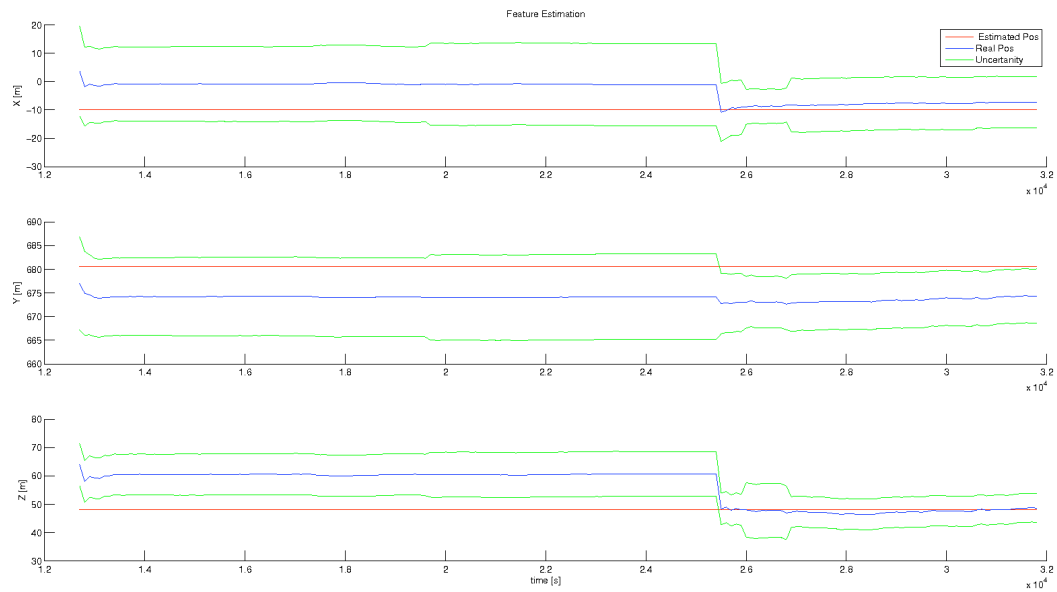


Figure 6.25: Effect of the filter update on a feature estimation

In Fig. 6.26 a density plot show the features estimates error. The minimum error is 0.21 m , and the maximum is 13.93 [m]; the mean error is 2.82 m, 94% of the features are estimated with an error less than 5 m (Fig.6.27)

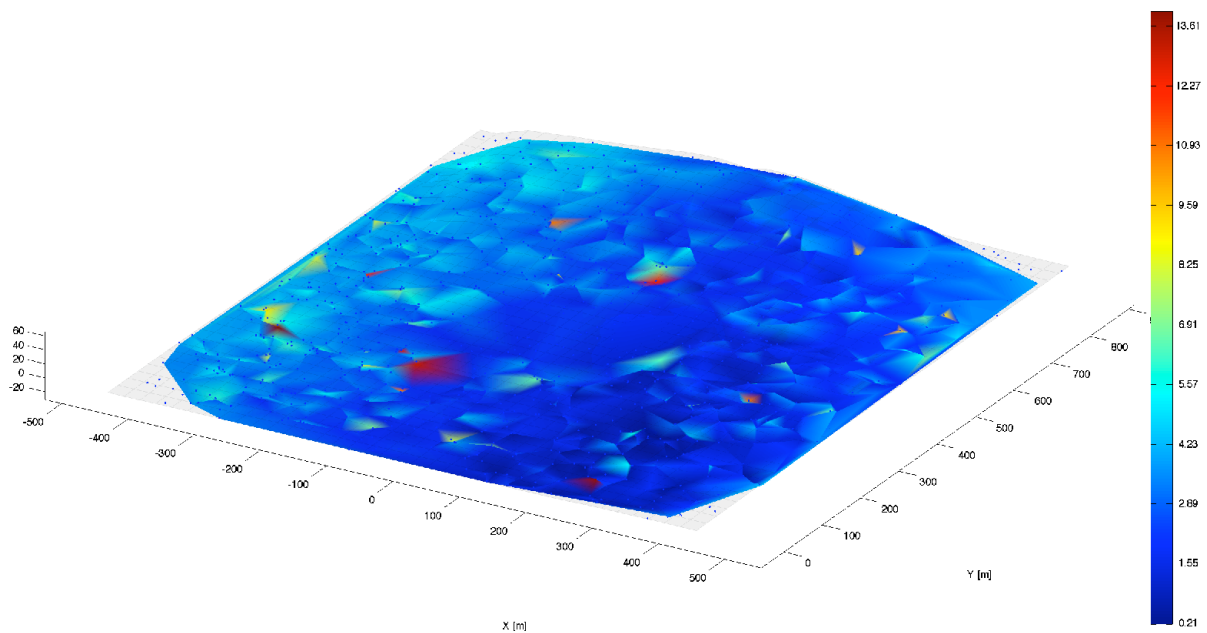


Figure 6.26: Feature Estimation Error represented by a gradient plot

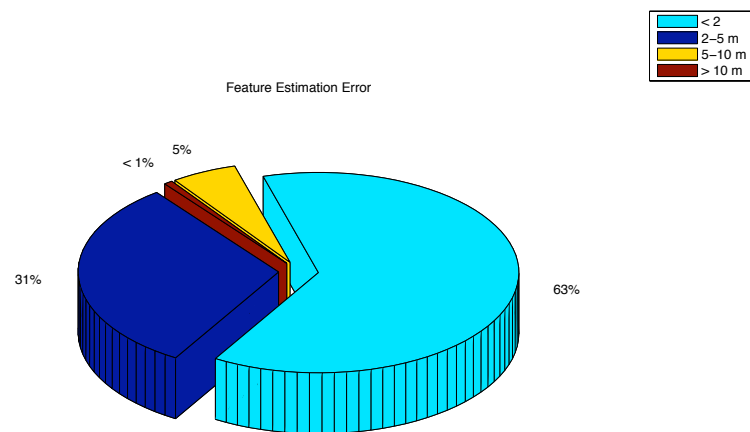


Figure 6.27: Error distribution in feature estimation

6.3.1.2 50 m altitude flight

In this case the ariship flies at 50 m of altitude; the stereo vision uncertainties are lower, see Table 4.2 in §4.2.7 for reference. The mean error is 1.45 m and the 99% of the features are estimated with an error less than 5 m.

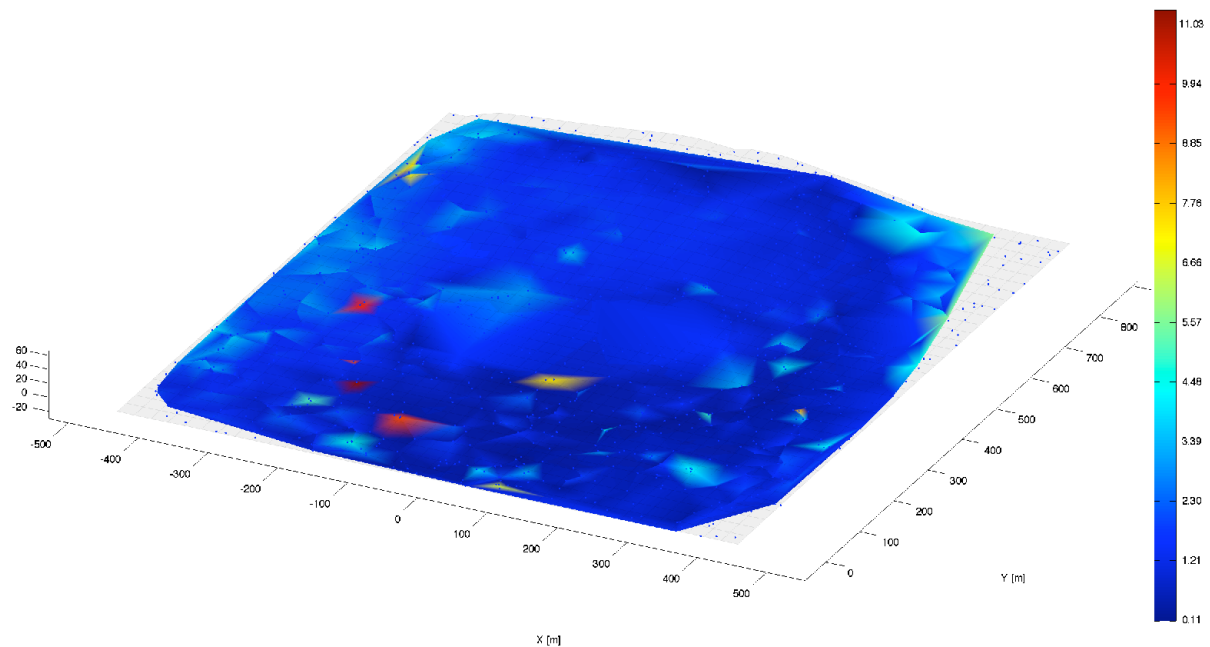


Figure 6.28: Feature Estimation Error at 50 m represented by a gradient plot

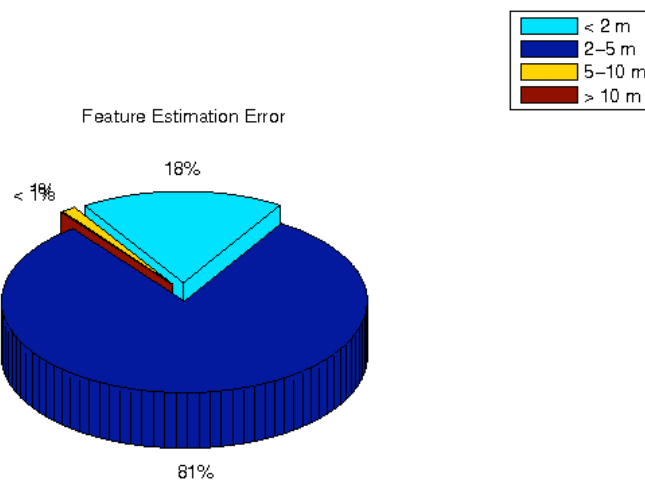


Figure 6.29: Error distribution in feature estimation at 50 m

6.3.2 Monocamera Slam

Until now, we have supposed to have two cameras mounted in a known calibrated configuration. We have seen how two cameras provide, bearing/elevation and range measurement, so the 3D coordinate of the observed landmarks are, to a certain extend, known.

However the 3D range of a stereo camera is limited and the immanent uncertainty of a landmarks's estimated 3D coordinated gets larger with its distance from the camera.

Airship navigation is not limited only by terrain mapping tasks, as seen above this kind of operation requires a certain degrees and precision in the map estimation; for example it could be necessary for the vehicle to move toward another site to perform the same tasks after uploading the data to an orbiter.

It's desirable that the airship could navigate autonomously also in the transfer cruise, at higher altitudes, where the positioning requirements are lower.

We ask if it's possible, at a certain altitude, for the airship to navigate only by using one camera. In §5.7 the structure of the monocamera SLAM is shown. Parallel to the main filter a subfiltering process has to be invoked for any new feature observed in order to estimate from the vehicle motion the range measurement. When this estimate is available the point in the map is included in the main filter.

The subsystem model is

$$\mathbf{x}_\tau^i(t) = \mathbf{x}_\tau^i(t-1) + \mathbf{w}_x \quad (6.95)$$

$$\lambda_\tau^i(t) = \lambda_\tau^i(t-1) + \mathbf{w}_\lambda \quad (6.96)$$

$$\mathbf{x}^i(t) = {}^i\mathbf{H}_s(t) {}^i\mathbf{H}_s(\tau) {}^s\mathbf{x}^i(t) \lambda^i(t) = \mathbf{h}({}^s\mathbf{x}^i(t), \lambda^i(t), \tau) \quad (6.97)$$

where the first two equations represent the state vector model and the third the measurement model.

Some implementation issues are necessary to be clarified.

Bearing measurement are simulated by getting 3D points from a DTM map loaded in the software; these points expressed in the body frame, ${}^s\mathbf{r}$, are processed them by the function ξ to get the bearing simulated measure.

$$\xi : {}^s\mathbf{r}^i \longrightarrow \mathbf{x}^i \quad (6.98)$$

From measure we need to compute the Jacobian of the model, which depends on the transformation

$$[\mathbf{x}^i, \lambda^i]_\tau \xrightarrow{\eta^{-1}} {}^s\mathbf{r}(\tau) \xrightarrow{{}^s\mathbf{H}_s(t, \tau)} {}^s\mathbf{r}(t) \xrightarrow{\xi} \mathbf{x}^i(t) \quad (6.99)$$

so the measure model equation become

$$\mathbf{z}(\mathbf{x}^i, \lambda^i, t, \tau) = \xi({}^i\mathbf{H}_s(t) {}^i\mathbf{H}_s(\tau) \eta^{-1}([\mathbf{x}^i, \lambda^i]_\tau)) \quad (6.100)$$

where

$$\begin{aligned} {}^i\mathbf{H}_s(t) {}^i\mathbf{H}_s(\tau) &= \\ &= \begin{bmatrix} {}^s\mathbf{R}_v(t) & {}^s\mathbf{T}_v(t) \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} {}^v\mathbf{R}_i(t) & {}^v\mathbf{T}_i(t) \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} {}^i\mathbf{R}_v(\tau) & {}^i\mathbf{T}_v(\tau) \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} {}^v\mathbf{R}_s(\tau) & {}^v\mathbf{T}_s(\tau) \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \\ &= {}^s\mathbf{R}_v(t) {}^v\mathbf{R}_i(t) {}^i\mathbf{R}_v(\tau) {}^v\mathbf{R}_s(\tau) \\ &+ {}^s\mathbf{R}_v(t) [{}^v\mathbf{R}_i(t) ({}^i\mathbf{R}_v(\tau) {}^v\mathbf{T}_s(\tau) + {}^i\mathbf{T}_v(\tau)) + {}^v\mathbf{T}_i(t)] + {}^s\mathbf{T}_v(t) \end{aligned} \quad (6.101)$$

The composed Jacobian is then

$$\nabla_{[\mathbf{x}^i, \lambda^i]} \mathbf{z}(k) = \nabla \xi \cdot {}^i\mathbf{H}_s(t) {}^i\mathbf{H}_s(\tau) \cdot \nabla(\eta^{-1}) \quad (6.102)$$

When the feature is moved from the subfilter to the main filter the covariance matrix has to be properly initialized and added into the main filter covariance matrix. The feature (in the inertial frame) is added as it was be observed at τ by

$$\mathbf{m}^i(\tau, \mathbf{x}^i, \lambda^i) = {}^i\mathbf{R}_v(\tau) {}^v\mathbf{R}_s \boldsymbol{\eta}^{-1}([\mathbf{x}^i, \lambda^i]) = \mathbf{g}(\mathbf{x}) \quad (6.103)$$

where $\mathbf{x} = [\mathbf{x}^i, \lambda^i]$ is the subilter state vector. The previous equation is a function of the subfilter state and produce a 3D point to be added to state of the main filter as a new point in the map. In order to compute the appropriate covariance we use that fact that if $\mathbf{y} = \mathbf{f}(\mathbf{x})$ yields

$$\mathbf{P}_y = \nabla_{\mathbf{x}} \mathbf{f} \mathbf{P}_x \nabla_{\mathbf{x}}^T \mathbf{f} \quad (6.104)$$

so

$$\mathbf{P}_{mm}(\tau|\tau) = \nabla_{\mathbf{x}} \mathbf{g} \cdot \mathbf{P}_x(t|t) \cdot \nabla_{\mathbf{x}}^T \mathbf{g} \quad (6.105)$$

where

$$\nabla_{\mathbf{x}} \mathbf{g} = {}^i\mathbf{R}_v(\tau) {}^v\mathbf{R}_s \cdot \nabla_{\mathbf{x}s} \boldsymbol{\eta}^{-1}([\mathbf{x}^i, \lambda^i]) \quad (6.106)$$

The simulation of the mono-camera SLAM doesn't use the CEKF as main filter, because the number of features is lower respect mapping operations.

The flight area is a strip with lenght of 2 km and the airship altitude is 200 m. Results are very encouraging because the maximum estimation error of a feature is $\simeq 8$ m (Fig. 6.40, 6.41). For a very long flight the uncertainty could grow slowly because loop-closure lack; for a transfer trajectory it's not so dramatic.

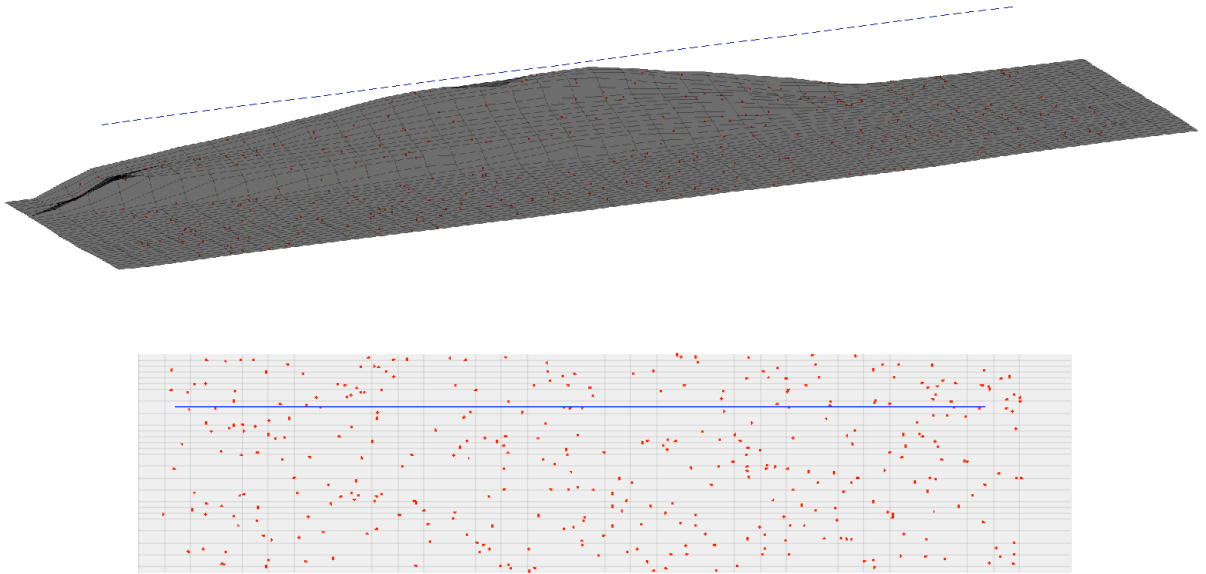


Figure 6.30: Trajectory and feature position over a landscape for monocular SLAM

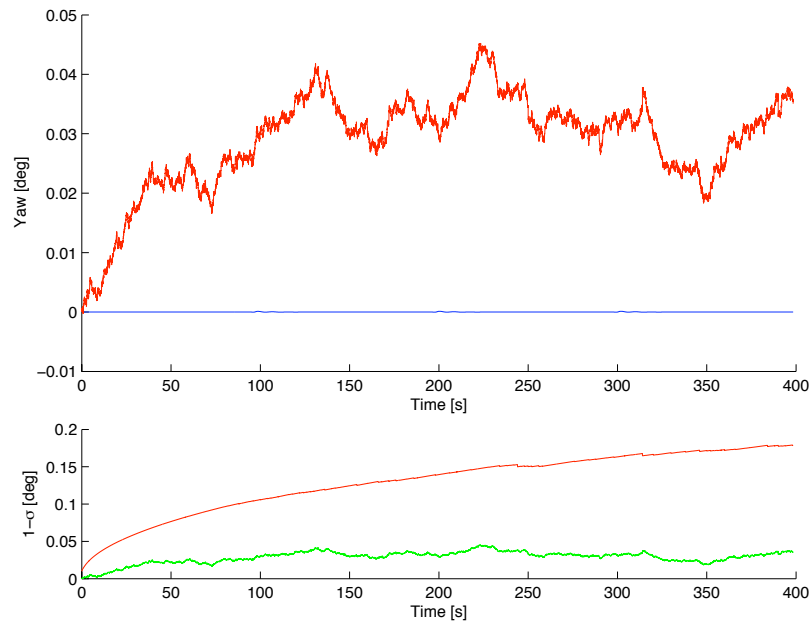


Figure 6.31: Yaw angle estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

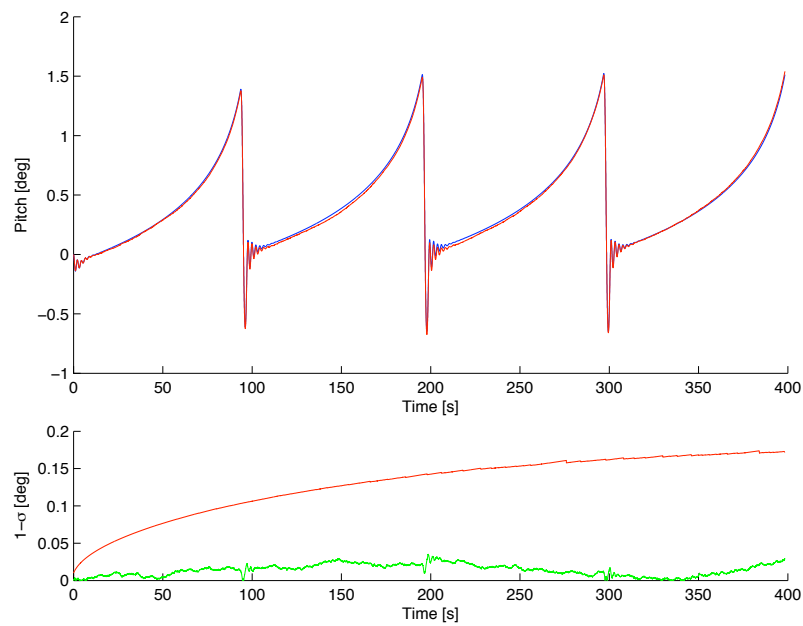


Figure 6.32: Pitch angle estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

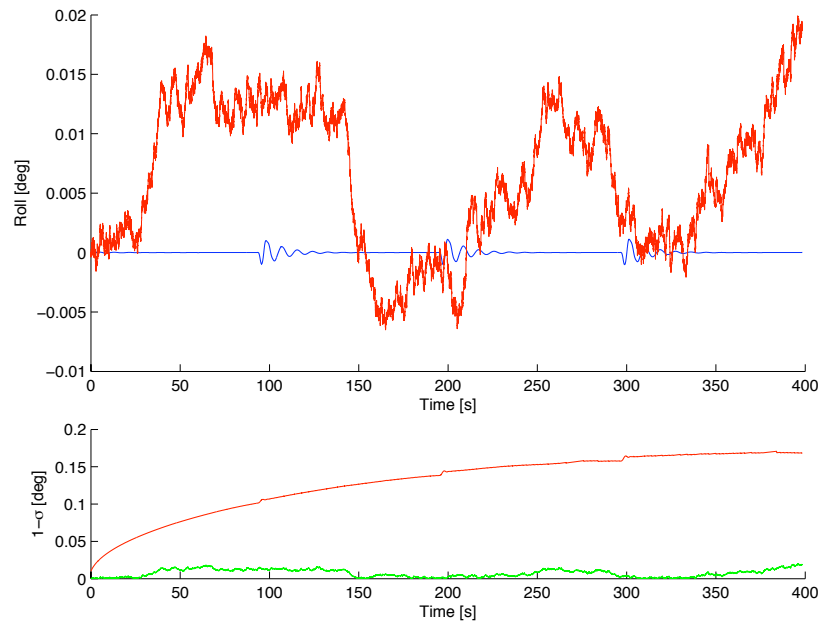


Figure 6.33: Roll angle estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

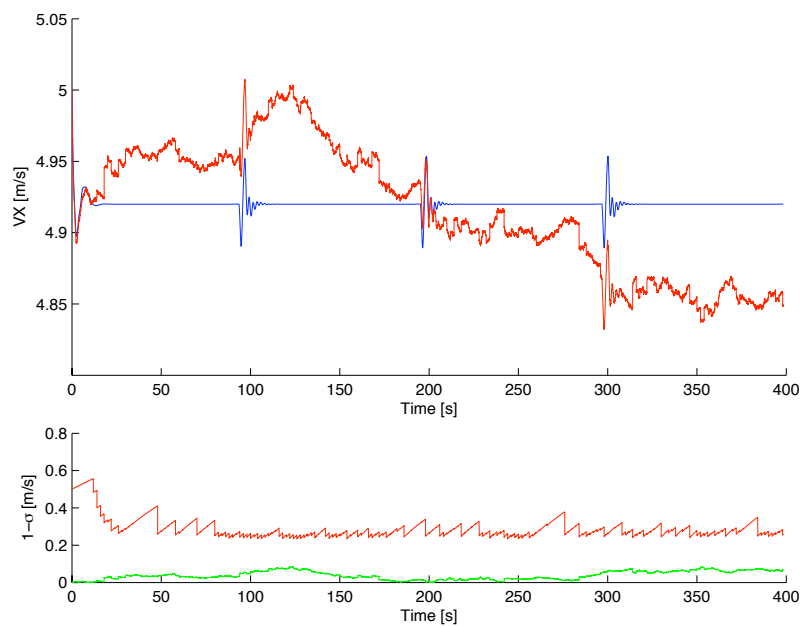


Figure 6.34: Longitudinal velocity estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

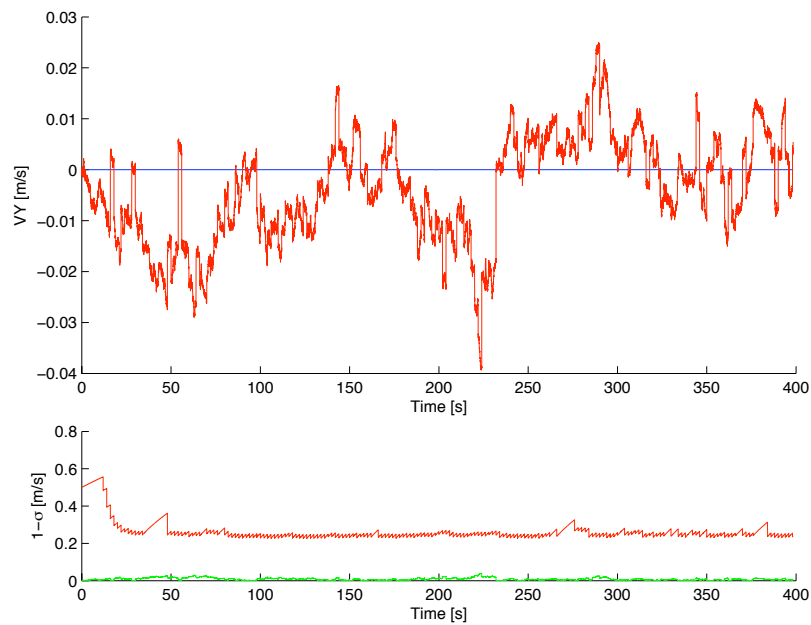


Figure 6.35: Transversal velocity estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

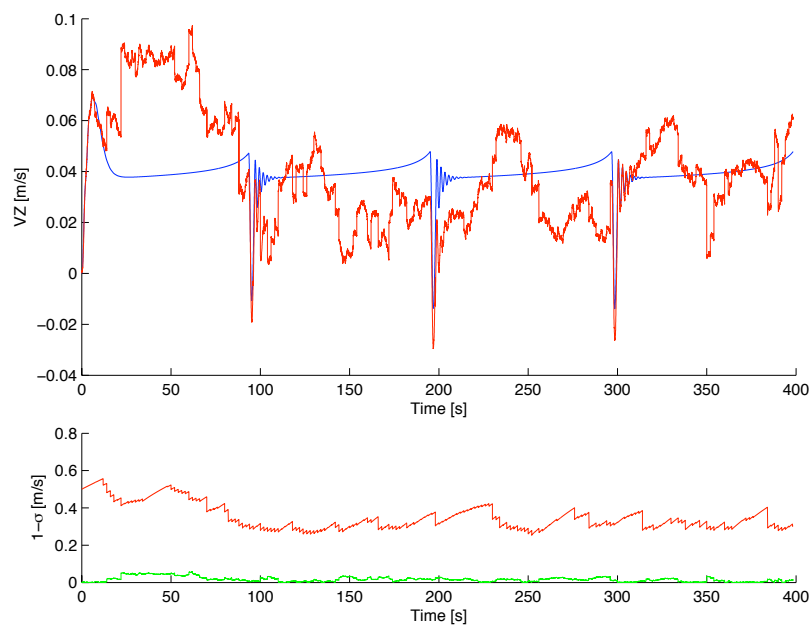


Figure 6.36: Vertical velocity estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

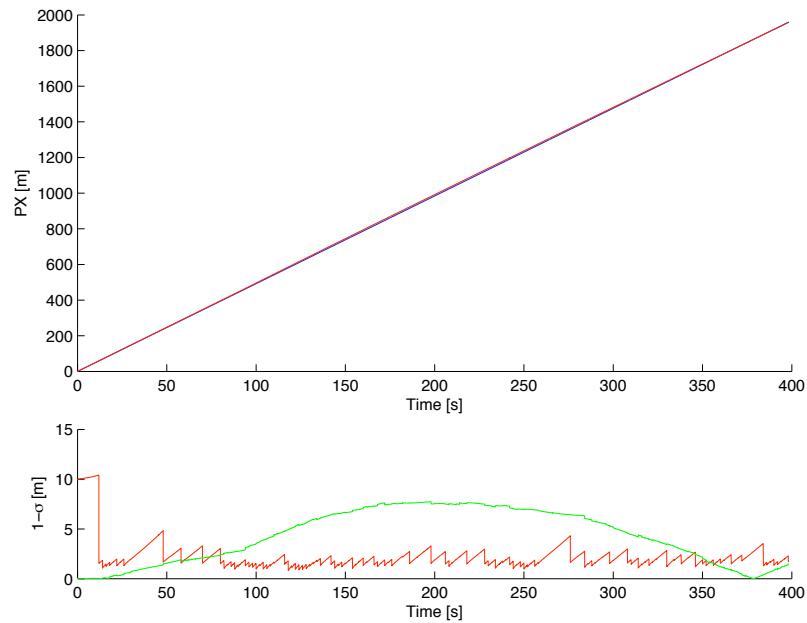


Figure 6.37: X-position estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

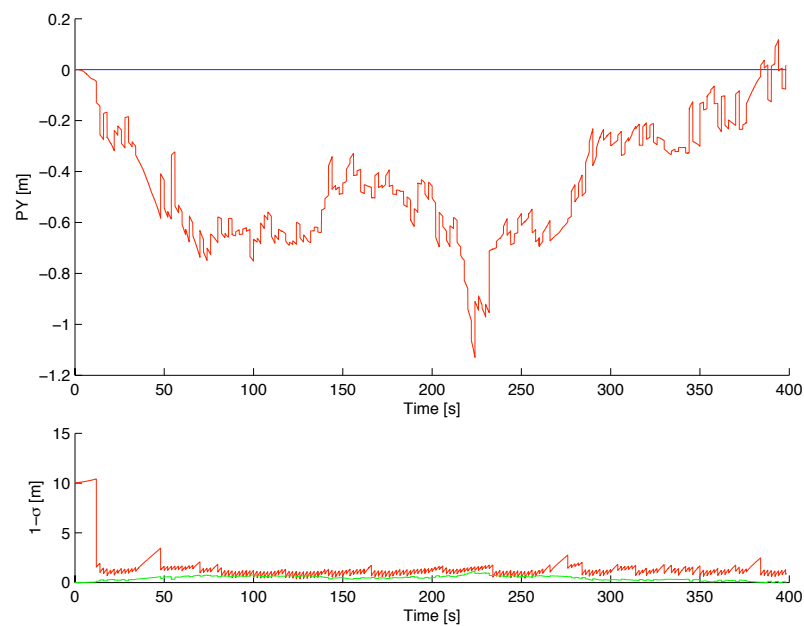


Figure 6.38: y-position estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

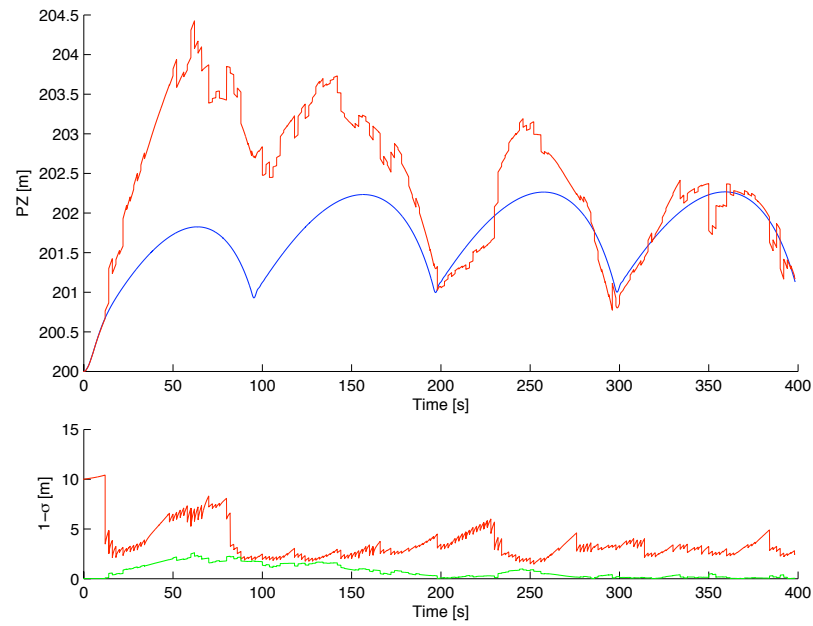


Figure 6.39: Z-position estimation; dashed lines represent the trend by dynamics. Below the estimation error (continuous line) and the parameter's variance updated by the filter (dash red line)

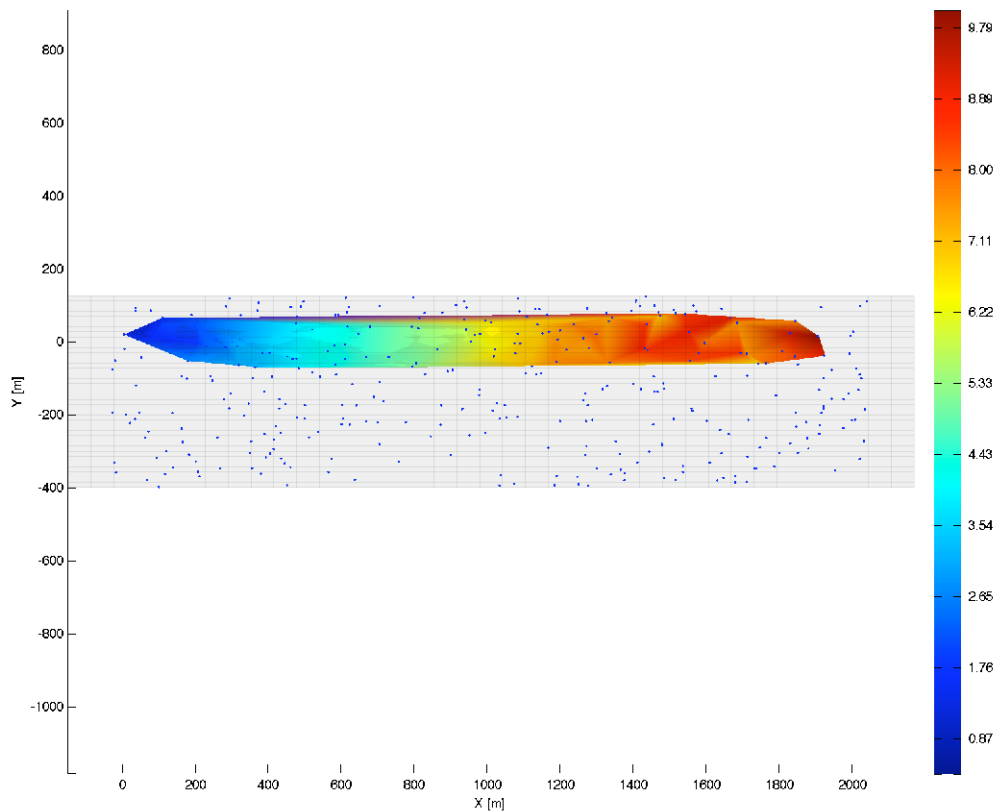


Figure 6.40: Feature Estimation Error represented by a gradient plot

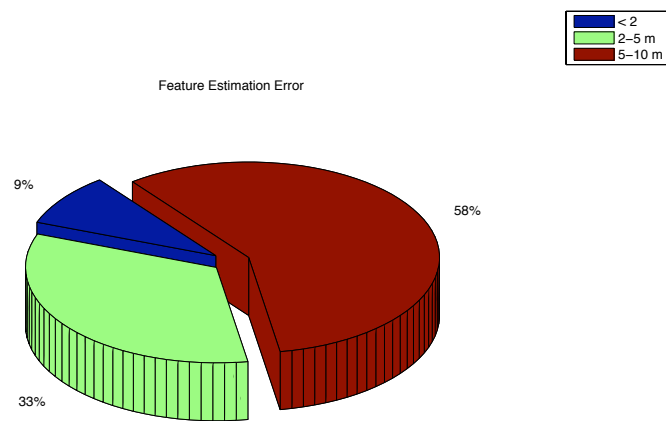


Figure 6.41: Error distribution in feature estimation

Conclusions

Exploration of the planets and moons of the solar system has so far been done through the remote sensing from Earth, fly-by probes, orbiter, landers and rovers. Remote sensing systems, probes and orbiters can only provide non-contact, low to medium resolution imagery across various spectral bands; landers provide high-resolution imagery and in situ data collection and analysis capabilities but only for a single site; while rovers allow imagery collection and in situ science across their path. The crucial drawback of ground based system is their limited coverage.

Airships combine the long-term airborne capability of balloons with the maneuverability of airplanes and helicopters. Their controllability allows precise flight path execution for surveying purposes, long-range as well as close-up ground observations, station keeping for long term monitoring of high-value science sites, deployment and transportation of scientific instruments across vast distance to key science sites and opportunistic flight path replanning in response to the detection of relevant sensor signatures or orbiter messages.

In this work we have handled the study of airships dynamics, control and navigation by IMU-vision sensor fusion.

Airship dynamics and control is a crucial point to model a realistic navigation system which takes into account the characteristics of the motions of this kind of vehicle in a selected environment. Literature in this field is rich and poor at the same time. Many studies are dated from the beginning of the 20th century where first applications of hydrodynamics take place. Authors as Max Munk provided simplified methods to describe the dynamical behaviour of forces and moments acting on airship hull under certain simplifications in the physical conditions. Most of efforts of the various authors were addressed to determine the aerodynamic forces and moments due to pressure and viscosity effects. While an estimate of pressure induced forces could be derived and modelled from hydrodynamic theory, the determination of viscous forces required to deal with experimental data. In this sense a lot of works dealt with force and pressure distribution measurements on number of different fuselage forms of varying slenderness ratio, varying rearward position of maximum thickness and varying nose ratio. With these profiles a lot of semi-empirical relations were derived to establish the limit of applicability of the potential theory and where a model for viscous forces has to be applied instead, to compute normal and axial forces.

The concept of added mass has played an important role in all theoretical approach; it has to be considered when a body moves through a fluid otherwise at rest. There's a certain amount of kinetic energy of the fluid caused by the motion of the body, when it's not constant (accelerated motion) the body makes a positive work on the fluid. Added mass is the principal cause of pressure induced forces on airships.

By numerical integrations it's possible to compute the pressure forces and moments without an explicit analytical expression for them, simply including in the equation of motions the added mass contribution like a further mass and inertia property of the system.

The characteristics of added mass matrix could be determined by geometric property of the

rigid body. It's quite simple for shapes as ellipsoid but calculation could be more complex in the case of generic solids. Numerical techniques are applied in this sense.

An explicit form for aerodynamic forces are required when we want to compute viscous effects on the hull because we have to remove the potential contribution from one airship station (function of hull length, shape and empirical coefficients) to the end of the hull and replace it by a viscous contribution. For slender ellipsoid hull adopted in our model, the interested station is near the tail. Our recent studies are oriented to determine added mass and viscous force contribution for different kind of drop-shaped hulls.

An airship simulator has been developed and two different kind of vehicles have been modelled, for terrestrial and planetary applications. Also a control strategy, for path planning, has been designed with a particular attention to its robustness in case of wind disturbances. Terrestrial and Titan atmosphere, gravity and dynamical viscosity were considered.

The question of the autonomous navigation of the airship in an unknown environment has been solved with SLAM technique. The main difficulty to build a digital terrain map is to precisely determine the sensor position and orientation as airship moves. Dead reckoning techniques that integrate over the time the data provided by motion estimation sensors are not sufficient because they're intrinsically prone to generate position estimates with unbounded error growth. Precise visual motion estimation techniques that use stereo or mono vision also accumulate error over the time, since they do not memorize any environment feature. The only solution to diminish the errors is to memorize detected features as the sensor moves. The problem of mapping such features and to estimate vehicle location are intimately tied together and they must be solved in a unified manner. This is the core of SLAM in which IMU and vision measurement are joined to estimate vehicle and feature position together. The solution proposed in this work is based on Extended Kalman filter where the system state merges vehicle kinematic parameters and feature location.

The state-based formulation of the SLAM involves the estimation of a joint state and its computational complexity scales quadratically with the number of landmarks.

First implementations adopted the classic EKF to build a consistent map of a unknown landscape by IMU measurement as inputs for the system and stereo camera to provide bearing and range measurements. As the number of landmarks increased less performant became the estimation process, so in order to manage large maps a variation of the EKF has been implemented: the Compressed Extended Kalman Filter.

With CEKF it is not necessary to perform a full SLAM update when working in a local area; so it was possible to divide the landscape into submaps and reduces the computational requirement of the SLAM algorithm to the order of the number of features in the vicinity of the vehicle.

Both EKF and CEKF solutions have been tested virtually in a Titan simulated environment, using the dynamical data of the Titan Explorer airship produced by the dynamics and control simulator; the results are very encouraging. Using a stereo vision setup at two different altitudes, 80 m and 50 m, the mean estimation error was 2.8 m and 1.45 m respectively.

If the airship has to move from one site to another it would be desirable to apply SLAM technique at altitudes greater than the limit of a stereo vision system. To provide a feedback to this kind of requirement it has been developed a SLAM technique based on a single camera measurement. Here for every new landmark measured, a subfiltering process is invoked to estimate the depth of the landmark by relating the current pose of the vehicle to that when the new feature was observed for the first time.

Simulation results are positive: the airship flies along a 2 km rectilinear trajectory over a stripe of surface at an altitude of 200 m, the mean estimation error is 5 m with a maximum error of 10 m. It's important to underline that in this situation no loop-closure is possible and so the error

will grow with the time; this is not dramatic being the localization requirements less stringent. This work provide a good baseline to start a series of validation tests with a real airship and to port the algorithms to a navigation's embedded computer platform.

Appendix A

Rotations Kinematics

A.1 Kinematic form of Rotation Matrix

Consider a vector $\mathbf{r}$ rotating with a constant vector angular velocity $\boldsymbol{\omega}$ so that

$$\frac{d\mathbf{r}}{dt} = \boldsymbol{\omega} \wedge \mathbf{r}, \quad (\text{A.1})$$

Let l, m, n be the direction cosine of $\boldsymbol{\omega}$ and $d\psi/dt$ the constant angular speed, so

$$\boldsymbol{\omega} = \frac{d\psi}{dt}(l\hat{\mathbf{i}} + m\hat{\mathbf{j}} + n\hat{\mathbf{k}}) = \frac{d\psi}{dt}\hat{\boldsymbol{\omega}} \quad (\text{A.2})$$

where $\hat{\boldsymbol{\omega}} = (l, m, n)^\top$. So we have

$$\frac{d\mathbf{r}}{dt} = \frac{d\mathbf{r}}{d\psi} \frac{d\psi}{dt} = \boldsymbol{\omega} \wedge \mathbf{r} = \frac{d\psi}{dt}\hat{\boldsymbol{\omega}} \wedge \mathbf{r} \quad (\text{A.3})$$

and

$$\frac{d\mathbf{r}}{d\psi} = \hat{\boldsymbol{\omega}} \wedge \mathbf{r} = S\mathbf{r} \quad (\text{A.4})$$

where

$$S = \begin{pmatrix} 0 & -n & m \\ n & 0 & -l \\ -m & l & 0 \end{pmatrix}. \quad (\text{A.5})$$

So we have a linear vector differential equation for $\mathbf{r}$; the solution as a function of ψ for which $\mathbf{r}(0) = \mathbf{r}_0$ may be written as

$$\mathbf{r}(\psi) = \left(\mathbf{I} + \psi S + \frac{1}{2!}\psi^2 S^2 + \frac{1}{3!}\psi^3 S^3 + \dots \right) \mathbf{r}_0. \quad (\text{A.6})$$

The matrix coefficient of $\mathbf{r}_0$ is the rotation matrix $\mathbf{R}$ expressed as an infinite matrix series. To obtain a more useful expression, we recall that

$$S\mathbf{r}_0 = \hat{\boldsymbol{\omega}} \wedge \mathbf{r}_0$$

so

$$S^2\mathbf{r}_0 = \hat{\boldsymbol{\omega}} \wedge (\hat{\boldsymbol{\omega}} \wedge \mathbf{r}_0) = (\hat{\boldsymbol{\omega}} \cdot \mathbf{r}_0)\hat{\boldsymbol{\omega}} - \mathbf{r}_0$$

and

$$S^3\mathbf{r}_0 = \hat{\boldsymbol{\omega}} \wedge [\hat{\boldsymbol{\omega}} \wedge (\hat{\boldsymbol{\omega}} \wedge \mathbf{r}_0)] = -\hat{\boldsymbol{\omega}} \wedge \mathbf{r}_0 = -S\mathbf{r}_0$$

from which we conclude that

$$S^3 = -S.$$

Thus, all powers of the matrix S are either $\pm S$ or $\pm S^2$ and we have

$$R = I + \left(\psi - \frac{1}{3!}\psi^3 + \dots \right) S + \left(\frac{1}{2!}\psi^2 - \frac{1}{4!}\psi^4 + \dots \right) S^2$$

or

$$R = I + \sin \psi S + (1 - \cos \psi) S^2. \quad (\text{A.7})$$

With the notation $\hat{\omega}\hat{\omega}^\top = \hat{\omega} \otimes \hat{\omega}$ we refer to matrix

$$\begin{pmatrix} \hat{l}\hat{i} \wedge \hat{l}\hat{i} & m\hat{j} \wedge \hat{l}\hat{i} & n\hat{k} \wedge \hat{l}\hat{i} \\ \vdots & \ddots & \end{pmatrix} = \begin{pmatrix} 0 & -n & m \\ n & 0 & -l \\ -m & l & 0 \end{pmatrix} = S.$$

It could be seen that $S^2 = -I + \hat{\omega}\hat{\omega}^\top$, from which $S^2 + I = \hat{\omega}\hat{\omega}^\top$. So we get

$$\begin{aligned} R &= I + \sin \psi S + (1 - \cos \psi) S^2 \\ &= I + \sin \psi S + (1 - \cos \psi)(\hat{\omega}\hat{\omega}^\top - I) \\ &= \cos \psi I + \sin \psi S + (1 - \cos \psi) \hat{\omega}\hat{\omega}^\top. \end{aligned} \quad (\text{A.8})$$

A.2 Time dependence of Rotation Matrix

Time dependence of the rotation matrix could be solved as follows. We could write $R(t + \Delta t) = R'R(t)$ where $R' = R'(\alpha)$ is the totation matrix respect an arbitrary axis as shown in (A.8) :

$$R(\alpha) = \cos \alpha I + (1 - \cos \alpha) \hat{\omega}\hat{\omega}^\top + \sin \alpha S. \quad (\text{A.9})$$

For small angles $\cos \alpha \simeq 1, \sin \alpha \simeq \alpha$, from which $R' = I + \alpha S$. If we take into account that $\alpha = |\omega|\Delta t$, we could write

$$R' = I + \tilde{\omega}\Delta t \quad \text{where} \quad \tilde{\omega} = \begin{pmatrix} 0 & -\omega_w & \omega_v \\ \omega_w & 0 & -\omega_u \\ -\omega_v & \omega_u & 0 \end{pmatrix} \quad (\text{A.10})$$

so

$$\frac{dR(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{R(t + \Delta t) - R(t)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{R'R(t) - R(t)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{(R' - I)R(t)}{\Delta t} = \tilde{\omega}R(t). \quad (\text{A.11})$$

Appendix B

MATHCORE Library Reference

A very efficient implementation has been done for Jacobian computation of linear functions. Most of this property has been used in `§airshipslam` to compute the derivative of the various linear functions with respect to state variables.

So a special linear algebra manipulation library was created especially for derivative issues. This library has been implemented in both MATLAB and C languages; here we report the C functions' description without any loss of generality and clarity.

B.1 Functions

- void **scm_zyx2rot** (const gsl_vector *a, gsl_matrix *R, gsl_matrix *dR)
RPY angles to rotation matrix and its derivative Returns Rotation Matrix R from (z,y,x) rotations angles.
- void **scm_zyx_kin** (const gsl_vector *a, gsl_matrix *H, gsl_matrix *dH)
Euler ZYX angles kinematic matrix.
- void **scm_skew** (const gsl_vector *w, gsl_matrix *S, gsl_matrix *dS)
Skew operator and its derivative from a input vector.
- void **scm_matmul** (gsl_matrix *A, gsl_matrix *B, gsl_matrix *C, gsl_matrix *dC, gsl_matrix *dCdA, gsl_matrix *dCdB)
Matrix multiplication and its derivative.
- void **scm_set_index** (gsl_vector_int *v)
Vector of indices from 0 to size(idx) with step 1.
- gsl_vector_int * **scm_set_gen_index** (int start, int end, int step)
Generic vector of indices.
- void **scm_set_matrix_block** (gsl_matrix *D, gsl_vector_int *idx_r, gsl_vector_int *idx_c, gsl_matrix *S)
Set matrix block of a destination matrix to a source matrix.

- void **scm_get_matrix_block** (gsl_matrix *D, gsl_vector_int *idx_r, gsl_vector_int *idx_c, gsl_matrix *S)
Get a sub matrix from a source matrix.
- void **scm_transpose_matrix** (gsl_matrix *D, gsl_matrix *S, gsl_matrix *dD, gsl_matrix *dS)
Matrix transpose and its derivative.
- void **scm_h2rt** (gsl_matrix *H, gsl_matrix *R, gsl_vector *T)
Get rotation matrix and translation vector T from homogeneous transformation matrix H.
- void **scm_rt2h** (gsl_matrix *H, gsl_matrix *R, gsl_vector *T)
Build homogeneous transformation matrix H from rotation matrix and translation vector T.
- void **scm_htra** (gsl_matrix *H, gsl_vector *v_b, gsl_vector *v_i)
Homogeneous point transformation from body frame to an inertial frame.
- void **scm_mat2row** (gsl_matrix *M, gsl_vector *v)
Vector build from a matrix putting rows in sequence.
- void **scm_matdiff** (gsl_matrix *A, gsl_matrix *B, gsl_matrix *dC)
Matrix difference and its derivative.
- void **scm_matdiv** (gsl_matrix *A, gsl_matrix *B, gsl_matrix *C, gsl_matrix *dC)
Matrix element wise division and its derivative.
- void **scm_matsum** (gsl_matrix *A, gsl_matrix *B, gsl_matrix *dC)
Matrix element wise sum and its derivative.
- void **scm_rot2zyx** (gsl_matrix *R, gsl_vector *v)
Rotation matrix to (ZYX) YPR angles.
- void **scm_hinv** (gsl_matrix *H, gsl_matrix *H_inv)
Inverse Homogeneous transformation matrix.
- void **scm_pt2rng** (gsl_vector *p, gsl_vector *r, gsl_matrix *D_r_p)
3D point to range-bearing measurement as modelled by frontal pinhole camera with the image plane parallel to z-axis at a distance f
- void **scm_rng2pt** (gsl_vector *p, gsl_vector *r, gsl_matrix *D_p_r)
Range measurement s modelled by frontal pinhole camera with the image plane parallel to z-axis at a distance f to 3D point.
- void **scm_invert_matrix** (gsl_matrix *S, gsl_matrix *D)
Invert a matrix.
- void **scm_cout_matrix** (gsl_matrix *M)
- void **scm_cout_vector** (gsl_vector *v)
- void **scm_cout_vector_int** (gsl_vector_int *v)

B.1.1 Function Documentation

B.1.1.1 `void scm_get_matrix_block (gsl_matrix * D, gsl_vector_int * idx_r,
gsl_vector_int * idx_c, gsl_matrix * S)`

Get a sub matrix from a source matrix.

Parameters:

D destination sub matrix

S source matrix

idx_r vector of S row indices

idx_c vector of S column indices

See also:

`scm_set_matrix_block` (p. 160)

B.1.1.2 `void scm_h2rt (gsl_matrix * H, gsl_matrix * R, gsl_vector * T)`

Get rotation matrix and translation vector T from homogeneous transformation matrix H.

Parameters:

H homogeneous transformation matrix

R rotation matrix (3,3)

T translation vector (3)

$$H = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

See also:

`scm_rt2h` (p. 159), `scm_hinv` (p. 153), `scm_htra` (p. 154)

Todo

derivatives

B.1.1.3 `void scm_hinv (gsl_matrix * H, gsl_matrix * H_inv)`

Inverse Homogeneous transformation matrix.

Parameters:

H homogeneous transformation matrix

H_inv vector in the body frame

$$H = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

See also:

`scm_rt2h` (p. 159), `scm_h2rt` (p. 153), `scm_htra` (p. 154)

B.1.1.4 `void scm_htra (gsl_matrix * H, gsl_vector * v_b, gsl_vector * v_i)`

Homogeneous point transformation from body frame to an inertial frame.

Parameters:

H homogeneous transformation matrix

v_b vector in the body frame

v_i vector in the inertial frame

See also:

`scm_h2rt` (p. 153), `scm_hinv` (p. 153), `scm_rt2h` (p. 159)

B.1.1.5 `void scm_invert_matrix (gsl_matrix * S, gsl_matrix * D)`

Invert a matrix.

Parameters:

S source matrix

D inverted matrix

B.1.1.6 `void scm_mat2row (gsl_matrix * M, gsl_vector * v)`

Vector build from a matrix putting rows in sequence.

Parameters:

M matrix (m,n)

v vector (m*n)

B.1.1.7 `void scm_matdiff (gsl_matrix * A, gsl_matrix * B, gsl_matrix * dC)`

Matrix difference and its derivative.

Parameters:

A matrix (m,n)

B matrix (m,n)

C matrix (m,n), $C=A-B$

dC matrix of derivatives of C elements wrt A and B elements ($M*N, 2*M*N$)

Note:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

then

$$dC = \begin{bmatrix} dc_{11}/a_{11} & dc_{11}/a_{12} & dc_{11}/a_{21} & dc_{11}/a_{22} & dc_{11}/b_{11} & dc_{11}/b_{12} & dc_{11}/b_{21} & dc_{11}/b_{22} \\ dc_{12}/a_{11} & dc_{12}/a_{12} & dc_{12}/a_{21} & dc_{12}/a_{22} & dc_{12}/b_{11} & dc_{12}/b_{12} & dc_{12}/b_{21} & dc_{12}/b_{22} \\ dc_{21}/a_{11} & dc_{21}/a_{12} & dc_{21}/a_{21} & dc_{21}/a_{22} & dc_{21}/b_{11} & dc_{21}/b_{12} & dc_{21}/b_{21} & dc_{21}/b_{22} \\ dc_{22}/a_{11} & dc_{22}/a_{12} & dc_{22}/a_{21} & dc_{22}/a_{22} & dc_{22}/b_{11} & dc_{22}/b_{12} & dc_{22}/b_{21} & dc_{22}/b_{22} \end{bmatrix}$$

$$dC = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

See also:

`scm_matsum` (p. 157), `scm_matmul` (p. 156), `scm_matdiv` (p. 155)

B.1.1.8 `void scm_matdiv (gsl_matrix * A, gsl_matrix * B, gsl_matrix * C, gsl_matrix * dC)`

Matrix element wise division and its derivative.

Parameters:

A matrix (m,n)

B matrix (m,n)

C matrix (m,n), $C=A/B$ element by element

dC matrix of derivatives of C elements wrt A and B elements ($M*N, 2*M*N$)

Note:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

then

$$dC = \begin{bmatrix} dc_{11}/a_{11} & dc_{11}/a_{12} & dc_{11}/a_{21} & dc_{11}/a_{22} & dc_{11}/b_{11} & dc_{11}/b_{12} & dc_{11}/b_{21} & dc_{11}/b_{22} \\ dc_{12}/a_{11} & dc_{12}/a_{12} & dc_{12}/a_{21} & dc_{12}/a_{22} & dc_{12}/b_{11} & dc_{12}/b_{12} & dc_{12}/b_{21} & dc_{12}/b_{22} \\ dc_{21}/a_{11} & dc_{21}/a_{12} & dc_{21}/a_{21} & dc_{21}/a_{22} & dc_{21}/b_{11} & dc_{21}/b_{12} & dc_{21}/b_{21} & dc_{21}/b_{22} \\ dc_{22}/a_{11} & dc_{22}/a_{12} & dc_{22}/a_{21} & dc_{22}/a_{22} & dc_{22}/b_{11} & dc_{22}/b_{12} & dc_{22}/b_{21} & dc_{22}/b_{22} \end{bmatrix}$$

$$dC = \begin{bmatrix} 1/b_{11} & 0 & 0 & 0 & -a_{11}/b_{11}^2 & 0 & 0 & 0 \\ 0 & 1/b_{12} & 0 & 0 & 0 & -a_{12}/b_{12}^2 & 0 & 0 \\ 0 & 0 & 1/b_{21} & 0 & 0 & 0 & -a_{21}/b_{21}^2 & 0 \\ 0 & 0 & 0 & 1/b_{22} & 0 & 0 & 0 & -a_{22}/b_{22}^2 \end{bmatrix}$$

See also:

`scm_matsum` (p. 157), `scm_matmul` (p. 156), `scm_matdiff` (p. 154)

B.1.1.9 `void scm_matmul (gsl_matrix * A, gsl_matrix * B, gsl_matrix * C, gsl_matrix * dC, gsl_matrix * dCdA, gsl_matrix * dCdB)`

Matrix multiplication and its derivative.

Parameters:

A matrix (m,n)

B matrix (n,l)

C matrix (m,l), C=A*B

dC matrix (m*l,m*n+n*l) of derivatives of C elements wrt A and B elements

dCdA matrix of derivatives of C elements wrt A elements (m*l,m*n)

dCdB matrix of derivatives of C elements wrt B elements (m*l,n*l)

Note:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

then

$$dC = \begin{bmatrix} dc_{11}/a_{11} & dc_{11}/a_{12} & dc_{11}/a_{21} & dc_{11}/a_{22} & dc_{11}/b_{11} & dc_{11}/b_{12} & dc_{11}/b_{21} & dc_{11}/b_{22} \\ dc_{12}/a_{11} & dc_{12}/a_{12} & dc_{12}/a_{21} & dc_{12}/a_{22} & dc_{12}/b_{11} & dc_{12}/b_{12} & dc_{12}/b_{21} & dc_{12}/b_{22} \\ dc_{21}/a_{11} & dc_{21}/a_{12} & dc_{21}/a_{21} & dc_{21}/a_{22} & dc_{21}/b_{11} & dc_{21}/b_{12} & dc_{21}/b_{21} & dc_{21}/b_{22} \\ dc_{22}/a_{11} & dc_{22}/a_{12} & dc_{22}/a_{21} & dc_{22}/a_{22} & dc_{22}/b_{11} & dc_{22}/b_{12} & dc_{22}/b_{21} & dc_{22}/b_{22} \end{bmatrix}$$

$$\frac{dC}{dA} = \begin{bmatrix} dc_{11}/a_{11} & dc_{11}/a_{12} & dc_{11}/a_{21} & dc_{11}/a_{22} \\ dc_{12}/a_{11} & dc_{12}/a_{12} & dc_{12}/a_{21} & dc_{12}/a_{22} \\ dc_{21}/a_{11} & dc_{21}/a_{12} & dc_{21}/a_{21} & dc_{21}/a_{22} \\ dc_{22}/a_{11} & dc_{22}/a_{12} & dc_{22}/a_{21} & dc_{22}/a_{22} \end{bmatrix}$$

$$\frac{dC}{dB} = \begin{bmatrix} dc_{11}/b_{11} & dc_{11}/b_{12} & dc_{11}/b_{21} & dc_{11}/b_{22} \\ dc_{12}/b_{11} & dc_{12}/b_{12} & dc_{12}/b_{21} & dc_{12}/b_{22} \\ dc_{21}/b_{11} & dc_{21}/b_{12} & dc_{21}/b_{21} & dc_{21}/b_{22} \\ dc_{22}/b_{11} & dc_{22}/b_{12} & dc_{22}/b_{21} & dc_{22}/b_{22} \end{bmatrix}$$

See also:

`scm_matsum` (p. 157), `scm_matdiff` (p. 154), `scm_matdiv` (p. 155)

B.1.1.10 `void scm_matsum (gsl_matrix * A, gsl_matrix * B, gsl_matrix * dC)`

Matrix element wise sum and its derivative.

Parameters:

A matrix (m,n)

B matrix (m,n)

C matrix (m,n), C=A+B element by element

dC matrix of derivatives of C elements wrt A and B elements (M*N,2*M*N)

Note:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

then

$$dC = \begin{bmatrix} dc_{11}/a_{11} & dc_{11}/a_{12} & dc_{11}/a_{21} & dc_{11}/a_{22} & dc_{11}/b_{11} & dc_{11}/b_{12} & dc_{11}/b_{21} & dc_{11}/b_{22} \\ dc_{12}/a_{11} & dc_{12}/a_{12} & dc_{12}/a_{21} & dc_{12}/a_{22} & dc_{12}/b_{11} & dc_{12}/b_{12} & dc_{12}/b_{21} & dc_{12}/b_{22} \\ dc_{21}/a_{11} & dc_{21}/a_{12} & dc_{21}/a_{21} & dc_{21}/a_{22} & dc_{21}/b_{11} & dc_{21}/b_{12} & dc_{21}/b_{21} & dc_{21}/b_{22} \\ dc_{22}/a_{11} & dc_{22}/a_{12} & dc_{22}/a_{21} & dc_{22}/a_{22} & dc_{22}/b_{11} & dc_{22}/b_{12} & dc_{22}/b_{21} & dc_{22}/b_{22} \end{bmatrix}$$

$$dC = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

See also:

`scm_matdiv` (p. 155), `scm_matmul` (p. 156), `scm_matdiff` (p. 154)

B.1.1.11 void scm_pt2rng (gsl_vector * *p*, gsl_vector * *r*, gsl_matrix * *D_r_p*)

3D point to range-bearing measurement as modelled by frontal pinhole camera with the image plane parallel to z-axis at a distance *f*

$$x = f \frac{x}{z}; y = f \frac{y}{z}$$

Parameters:

p 3D point vector

r range measurement [px/z; py/z; pz]

D_r_p matrix of derivatives of *r* wrt to *p*

$$\partial r_p = \begin{bmatrix} 1/pz & 0 & -px/pz^2 \\ 0 & 1/pz & -py/pz^2 \\ 0 & 0 & 1 \end{bmatrix}$$

B.1.1.12 void scm_rng2pt (gsl_vector * *p*, gsl_vector * *r*, gsl_matrix * *D_p_r*)

Range measurement *s* modelled by frontal pinhole camera with the image plane parallel to z-axis at a distance *f* to 3D point.

$$x = f \frac{x}{z}; y = f \frac{y}{z}$$

Parameters:

p 3D point vector

r range measurement [px/z; py/z; pz]

D_p_r matrix of derivatives of *p* wrt to *r*

$$\partial p_r = \begin{bmatrix} rz & 0 & rx \\ 0 & rz & ry \\ 0 & 0 & 1 \end{bmatrix}$$

B.1.1.13 void scm_rot2zyx (gsl_matrix * *R*, gsl_vector * *v*)

Rotation matrix to (ZYX) YPR angles.

z = yaw

y = pitch

x = roll

Rotation sequence X-> Y-> Z From body to Inertial $R = Z*Y*X$

Parameters:

R is the pointer to rotation Matrix

v vector of rotation sequence

See also:

scm_zyx_kin (p. 161), **scm_zyx2rot** (p. 161)

B.1.1.14 void scm_rt2h (gsl_matrix * *H*, gsl_matrix * *R*, gsl_vector * *T*)

Build homogeneous transformation matrix *H* from rotation matrix and translation vector *T*.

Parameters:

H homogeneous transformation matrix

R rotation matrix (3,3)

T translation vector (3)

$$H = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

See also:

scm_h2rt (p. 153), scm_hinv (p. 153), scm_htra (p. 154)

Todo

derivatives

B.1.1.15 gsl_vector_int* scm_set_gen_index (int *start*, int *end*, int *step*)

Generic vector of indices.

Parameters:

start start index

end end index

step

Returns:

gsl int vector idx

See also:

scm_set_index (p. 159)

B.1.1.16 void scm_set_index (gsl_vector_int * *v*)

Vector of indices from 0 to size(idx) with step 1.

Parameters:

v int vector pointer

See also:

scm_set_gen_index (p. 159)

B.1.1.17 `void scm_set_matrix_block (gsl_matrix * D, gsl_vector_int * idx_r,
gsl_vector_int * idx_c, gsl_matrix * S)`

Set matrix block of a destination matrix to a source matrix.

Parameters:

D destination matrix

S source matrix

idx_r vector of D row indices

idx_c vector of D column indices

Note:

the dimension of S is [S]=[idx_r -> size, idx_c -> size]

See also:

`scm_get_matrix_block` (p. 153)

B.1.1.18 `void scm_skew (const gsl_vector * w, gsl_matrix * S, gsl_matrix * dS)`

Skew operator and its derivative from a input vector.

Parameters:

w vector pointer (3)

S skew operator pointer (3,3)

dS derivatives of S elements wrt w elements (9,3)

B.1.1.19 `void scm_transpose_matrix (gsl_matrix * D, gsl_matrix * S,
gsl_matrix * dD, gsl_matrix * dS)`

Matrix transpose and its derivative.

Parameters:

S source matrix (m,n)

D transpose matrix (n,m)

dS matrix of derivatives (M*N,L) of a elements wrt to certain parameters vector, $\mathbf{p} = [p_1, \dots, p_l]$

dD matrix (M*N,L) of derivatives wrt to the same parameter vector

Note:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$p = [p_1, p_2, p_3]$$

$$dA = \begin{bmatrix} da_{11}/p_1 & da_{11}/p_2 & da_{11}/p_3 \\ da_{12}/p_1 & da_{12}/p_2 & da_{12}/p_3 \\ da_{21}/p_1 & da_{21}/p_2 & da_{21}/p_3 \\ da_{22}/p_1 & da_{22}/p_2 & da_{22}/p_3 \end{bmatrix}$$

then

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$dB = \begin{bmatrix} db_{11}/p_1 & db_{11}/p_2 & db_{11}/p_3 \\ db_{12}/p_1 & db_{12}/p_2 & db_{12}/p_3 \\ db_{21}/p_1 & db_{21}/p_2 & db_{21}/p_3 \\ db_{22}/p_1 & db_{22}/p_2 & db_{22}/p_3 \end{bmatrix}$$

See also:

B.1.1.20 `void scm_zyx2rot (const gsl_vector * a, gsl_matrix * R, gsl_matrix * dR)`

RPY angles to rotation matrix and its derivative Returns Rotation Matrix *R* from (z,y,x) rotations angles.

z = yaw

y = pitch

x = roll

Rotation sequence X-> Y-> Z From body to Inertial $R = Z*Y*X$

Parameters:

a representing euler angle sequence

R is the pointer to rotation Matrix

dR is the pointer to rotation Matrix derivative wrt *a*

See also:

`scm_zyx_kin` (p. 161), `scm_rot2zyx` (p. 158)

B.1.1.21 `void scm_zyx_kin (const gsl_vector * a, gsl_matrix * H, gsl_matrix * dH)`

Euler ZYX angles kinematic matrix.

z = yaw

y = pitch

$x = \text{roll}$

Rotation sequence X-> Y-> Z From body to Inertial $R = Z*Y*X$

Parameters:

$\mathbf{a}$ rapresenting euler angle sequence

$\mathbf{H}$ kinematic matrix i.e. $da/dt = H * w$ (3,3)

$d\mathbf{H}$ derivative of H elements wrt Euler angles (9,3)

See also:

`scm_zyx2rot` (p. 161), `scm_rot2zyx` (p. 158)

Bibliography

- [A. Colozza , 2004] A. Colozza (2004). Airships for planetary exploration. *NASA Publication*. CR-2004-213345.
- [A. Coustenis and R. D. Lorenz, 1998] A. Coustenis and R. D. Lorenz (1998). *Titan*, in *Academic Press Encyclopaedia of the Solar System*. eds.
- [A. Elfes et al., 2003] A. Elfes, S. Bueno, M. Bergmann, E. de Pavia, and J. Ramos and J. Azinheira (2003). Robotic airships for exploration of planetary body with an atmosphere. *Autonomy Challenges, "Autonomous Robots"*. Vol 14,2003 pp 147,164.
- [A. Alfes et al., 2004] A. Alfes, J.L.Hall, J.F.Montgomery, C.F.Bergh, and B.A.Dudik (2004). Towards a substantially autonomous aerobot for exploration of titan. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*.
- [A.B.Chatfield, 1997] A.B.Chatfield (1997). *Fundamental of High Inertial Navigation*. Progress in Astronautics and Aeronautics, vol. 174.
- [Alexey A. Pankine et al., 2005] Alexey A. Pankine, Kim M. Aaron, Nathan C. Barne, and Kerry T. Nock (2005). Sailing the planets: Exploring mars from guided balloons. *AIAA 5th Aviation, Technology, Integration, and Operations Conference*.
- [Allen, 1959] Allen, J. (1959). Estimation of the forces and moments acting on inclined bodies of revolution of high fineness ratio. *NACA RM A9126*.
- [C. Estrada et al., 2005] C. Estrada, J. Neira, and J.D. Tardos (2005). Hierarchical slam: Real-time accurate mapping of large environments. *IEEE Transactions on Robotics*. vol.21, N.4.
- [D. A. Fairbrother, 2006] D. A. Fairbrother (2006). Development of planetary balloons. *NASA Pub.*
- [D. Barnes et al., 1995] D. Barnes, A. Shaw, P. Summers, R. Ward, M. Woods, M. Evans, G.Paar, and M. Sims (1995). Imagins and localization software demonstrator for planetary aerobots. *1995 IEEE International Conference on Robotics and Automation*.
- [D.A. Vallado, 2001] D.A. Vallado (2001). *FUndamental of Astrodynamics and Applications*. Microcosm Press.
- [D.D.Diel et al., 2005] D.D.Diel, P.DeBitetto, and S.Teller (2005). Epipolar constraint for vision-aided inertial navigation. *IEEE Proc. 7th Workshop on Application of Computer Vision*.
- [Ed. J.R. Werz, 2001] Ed. J.R. Werz (2001). *Spacecraft Attitude Determination and Control*. Microcosm Press.
- [Edward J. Hopkins, 1951] Edward J. Hopkins (1951). A semiempirical method for calculating the pitching moment of body of revolution. *NACA Research Memorandum*.

- [E.Hygounenc et al., 1998] E.Hygounenc, I.Jung, P.Soueres, and S.Lacroix (1998). The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping. *Paper supported By COMETS IST project*.
- [E.Laan et al., 2004] E.Laan, A.Wielders, A.Eggers, and J.Burke (2004). Advances in planetary aerobots. *IAF*. 04-Q.4.03.
- [E.Nebot and H.Durrant-Whyte, 1997] E.Nebot and H.Durrant-Whyte (1997). Initial calibration and alignment of an inertial navigation. *IEEE*.
- [Fossen, 1994] Fossen, T. I. (1994). *Guidance and Control of Ocean Vehicles*. Wiley.
- [G.A. Khoury and J.D. Gillett, 1999] G.A. Khoury and J.D. Gillett (1999). *Airship Technology*. Cambrige.
- [G.H. Fichtl et al., 1971] G.H. Fichtl, R.E. DeMandel, and S.J Krivo (1971). Aerodynamic properties of sphericsl balloon wind sensors. *Journal of Applied Meteorology*. vol.11.
- [J. Guivant et al., 2000] J. Guivant, E.M Nebot, and H.F. Durant-Whyte (2000). Simultaneous localization and map building using natural features in outdoor enviroments. *Proc 6th Int. Conference on intelligent autonomous systems*. volume 1, pages 581-588.
- [J. Hekkila and O. Silven, 1997] J. Hekkila and O. Silven (1997). A four-step camera calibration procedure with implicit image correction. *CVPR97*.
- [J.A Cutts et al., 1995] J.A Cutts, G. Rodriguez, S.P Synnott, K.T Neck, J. Balaram, J.A Jones, and G.E. Powell (1995). Aerovehicles for planetary exploration. *1995 IEEE International Conference on Robotics and Automation*.
- [J.B. Mueller and M.A Paluszek,] J.B. Mueller and M.A Paluszek. Develompent of an aerodynamic model and control law design for a high altitude airship. *American Institute of Aeronautics and Astronautics*.
- [J.H. Allen and E.W. Perkins, 1952] J.H. Allen and E.W. Perkins (1952). A study of effects of viscosity on flow over slender inclued bodies of revolution. *NACA Report 1048*.
- [J.Heikkilä and O.Silvèn,] J.Heikkilä and O.Silvèn. A four-step camera calibration procedure with implicit image correction. *Paper*.
- [J.Kim and S. Sukkarieh, 2004] J.Kim and S. Sukkarieh (2004). Autonomous airborne navigation in unknown terrain enviroments. *IEEE TRansactions on Aerospace and Electronic Systems*. vol.40,N.3.
- [J.Kim and S. Sukkarieh, 2007] J.Kim and S. Sukkarieh (2007). Real-time implementation of airborne inertial-slam. *Robotics and Automation Systems*. pag 62-71.
- [Joseph A. Ames, 1954] Joseph A. Ames (1954). A résumé of the advances in theoretical aeronautics made by max munk. *Report No.213, National Advisory Committee For Aeronautics*.
- [J.R.Azinheira et al., 2002] J.R.Azinheira, E.de Paiva, and S.Bueno (2002). Influence of a wind speed on airship dynamics. *Journal of Guidance, Control and Dynamics*. Vol.25,N.6.
- [J.S. Levine and H.S. Wright, 2005] J.S. Levine and H.S. Wright (2005). Titan explorer. *NASA Report for NRA-03-OSS-01*.
- [K.J.Walchko and P.A.C.Manson, 2002] K.J.Walchko and P.A.C.Manson (2002). Inertial navigation. *2002 Florida COnference on Recent Advances in Robotics*.

- [Lamb, 1945] Lamb, S. H. (1945). *Hydrodynamics*. Dover.
- [Max M. Munk, 1924] Max M. Munk (1924). The aerodynamic forces on airship hulls. *Report No.184, National Advisory Committee For Aeronautics*.
- [M.Bryson and S.Sukkarieh,] M.Bryson and S.Sukkarieh. Vehicle model aided inertial navigation for uav using low-cost sensors.
- [M.S. Grewal and A.P. Andrews, 2001] M.S. Grewal and A.P. Andrews (2001). *Kalman Filtering: Theory and Practice Using Matlab*. Wiley Interscience Publication.
- [M.S.Grewal et al., 1990] M.S.Grewal, V.D. Henderson, and R.S.Miyasako (1990). Application of kalman filtering to the calibration and alignment of inertial navigation systems. *Proceedings of the 29th conference on Decision and Control, 1990*.
- [M.W.M.Dissanayake et al., 2001] M.W.M.Dissanayake, P. Newman, S.Clarke, H.F. Durant-Whyte, and M. Csorba (2001). A solution to the simultaneous localization map building problem. *IEEE Transactions on Robotics and Automation*. vol.17, N.3.
- [Nielsen, 1988] Nielsen, J. N. (1988). *Missile Aerodynamics*. AIAA.
- [P.G.Thomasson, 2000] P.G.Thomasson (2000). Equation of motions of a vehicle in a moving fluid. *Journal of Aircraft*. Vol.37,N.4.
- [R. D. Lorenz, 2000] R. D. Lorenz (2000). Optimizing science return from titan aerial explorers. *IEEE*.
- [R. D. Lorenz and T.Tokano, 2006] R. D. Lorenz and T.Tokano (2006). Gcm simulation of balloon trajectories on titan. *Planetary and Space Science*.
- [R.Da forno, 2001] R.Da forno (2001). *Dal Corpo Rigido al Robot con MATLAB*. McGraw-Hill.
- [R.D.Lorenz, 2001] R.D.Lorenz (2001). Scaling laws for flight power of aeroplanes, airships and helicopters: Application to planetary exploration. *Journal of Aircraft*.
- [R.H. Upson and W.A Klikoff, 1931] R.H. Upson and W.A Klikoff (1931). Application of practical hydrodynamics to airship design. *NACA Report 405*.
- [S. Joshi et al., 2004] S. Joshi, G. Nagai, and C. Sullivan (2004). Planetary navigation architecture for cooperative autonomous ground-air-space robots. *AIAA*. 2004-6256.
- [S. van der Zwaan et al., 2001] S. van der Zwaan, A. Bernardino, and J. Santos-Victor (2001). Vision based station keeping and docking for floatin vehicle. *European Control Conference*.
- [S.B.V. Gomes and J.Ramos, 1998] S.B.V. Gomes and J.Ramos (1998). Airship dynamic modeling for autonomous operation. *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*.
- [S.I.Roumeliotis et al.,] S.I.Roumeliotis, A.E. Johnson, and J.F.Montgomery. Augmented inertial navigation with image-based motion estimation. *Paper*.
- [S.P. Jones and J.D De Lournier, 1981] S.P. Jones and J.D De Lournier (1981). Aerodynamic estimation techniques for aerostats and airships. *Journal of Aircraft*. VOL. 20, NO. 2.
- [T. Bailey and H.F. Durant-Whyte, 2006] T. Bailey and H.F. Durant-Whyte (2006). Simultaneous localization and mapping part ii. *IEEE Robotics and Automation Magazine*. pages 109-110.

- [Tsai, 1987] Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the shelf tv camera and lenses. *IEEE Int. Journal of Robotics and Automation*. Vol 3, pp.323-344.
- [Tuckerman, 1925] Tuckerman, L. (1925). Inertia factors of ellipsoid for use in airship design. *National Advisory Committee for Aeronautics*. Report No.210.
- [Won S. Kim et al., 2005] Won S. Kim, Adnan I. Ansar, and Robert D. Steele (2005). Rover mast calibration, exact camera pointing, and camera handoff for visual target tracking. *IEEE*.
- [Y.Ma et al., 2006] Y.Ma, S.Soatto, J.Košecká, and S.Shankar Sastry (2006). *Invitation to 3-D Vision*. Springer.
- [Yuwen Li and Meyer Nahon, 2007] Yuwen Li and Meyer Nahon (2007). Modeling and simulation of airship dynamics. *Journal of Guidance Control and Dynamics*. Vol 30, No.6, November-December 2007.
- [Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Trans. on PAMI*. Vol 22(11), pp 1330-1334.