UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA MECÂNICA

Apolo Silva Marton

Control Architecture for the Navigation System of Robotic Airship using Incremental Controllers

Arquitetura de Controle para Sistema de Navegação de um Dirigível Robótico utilizando Controladores Incrementais

CAMPINAS

2021

Apolo Silva Marton

Control Architecture for the Navigation System of Robotic Airship using Incremental Controllers

Arquitetura de Controle para Sistema de Navegação de um Dirigível Robótico utilizando Controladores Incrementais

Thesis presented to the School of Mechanical Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Mechanical Engineering, in the area of Machatronics.

Tese apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Doutor em Engenharia Mecânica, na Área de Mecatrônica.

Orientador: Prof. Dr. André Ricardo Fioravanti

Coorientadores: Prof. Dr. José Raul Carreira Azinheira e Prof. Dr. Ely Carneiro de Paiva

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELO ALUNO APOLO SILVA MARTON, E ORIENTADA PELO PROF. DR. ANDRÉ RICARDO FIORAVANTI.

CAMPINAS
2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

M367c

Marton, Apolo Silva, 1990-
Control arquitecture for the navigation system of robotic airship using incremental controllers / Apolo Silva Marton. – Campinas, SP : [s.n.], 2021.

Orientador: André Ricardo Fioravanti.
Coorientadores: Ely Carneiro de Paiva e José Raul Carreira Azinheira.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Robôs - Sistemas de controle. 2. Voo - Controle. 3. Sistemas de veículos auto-guiados. 4. Filtragem de Kalman. 5. Redes Neurais. I. Fioravanti, André Ricardo, 1982-. II. Paiva, Ely Carneiro de, 1965-. III. Azinheira, José Raul Carreira. IV. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. V. Título.

Informações para Biblioteca Digital

Título em outro idioma: Arquitetura de controle para sistema de navegação de um dirigível robótico utilizando controladores incrementais
Palavras-chave em inglês:
Robots - Control systems
Flight - Control
Auto-guided vehicle systems
Kalman Filtering
Neural Networks
Área de concentração: Mecatrônica
Titulação: Doutor em Engenharia Mecânica
Banca examinadora:
André Ricardo Fioravanti [Orientador]
José Reginaldo Hughes de Carvalho
Henrique de Souza Vieira
Alim Pedro de Castro Gonçalves
Paulo Roberto Gardel Kurka
Data de defesa: 29-11-2021
Programa de Pós-Graduação: Engenharia Mecânica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/ 0000-0001-8387-049
- Currículo Lattes do autor: http://lattes.cnpq.br/8539232670666917

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

TESE DE DOUTORADO

Control Architecture for the Navigation System of Robotic Airship using Incremental Controllers

Arquitetura de Controle para Sistema de Navegação de um Dirigível Robótico utilizando Controladores Incrementais

Autor: Apolo Silva Marton
Orientador: Prof. Dr. André Ricardo Fioravanti
Coorientador: Prof. Dr. José Raul Carreira Azinheira e Prof. Dr. Ely Carneiro de Paiva

A banca examinadora composta pelos membros abaixo aprovou esta tese:

Prof. Dr. André Ricardo Fioravanti
UNICAMP

Prof. Dr. José Reginaldo Hughes de Carvalho
UFAM

Prof. Dr. Henrique de Souza Vieira
UNINORTE

Prof. Dr. Alim Pedro de Castro Gonçalves
UNICAMP

Prof. Dr. Paulo Roberto Gardel Kurka
UNICAMP

A Ata de Defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 29 de Novembro de 2021

Dedicatória

A minha esposa, familía e amigos que sempre estiveram me incentivando à vencer os desafios a cada dia.

Agradecimentos

Primeiramente a Deus que tem me dado força todos os dias para vencer as dificuldades que me aparecem como desafio;

Ao meu orientador, Prof. Dr. André Ricardo Fioravanti por ter me orientado em vários instantes da minha vida acadêmica e estar sempre disponível nos momentos de dificuldade cooperando para o bom progresso durante todo o trabalho;

Ao meu co-orientador, Prof. Dr. José Raul Azinheira por ter cooperado com o tema, ideias e soluções para o desenvolvimento do trabalho, além de sempre estar disponível para me auxiliar e ensinar sobre os tópicos de interesse deste trabalho;

Ao meu co-orientador, Prof. Dr. Ely Carneiro de Paiva por ter me ajudado em vários áreas da vida acadêmica o que tornou possível a conclusão deste trabalho;

Ao Centro de Tecnologia de Informação Renato Archer, por ter fornecido infraestrutura e ter me concedido a oportunidade de estar trabalhando no projeto DRONI;

Aos membros das bancas de qualificação e defesa por cooperarem para o melhor desenvolvimento da pesquisa no âmbito da UNICAMP e do Brasil;

A toda minha família por ter me apoiado durante todo o período de desenvolvimento deste trabalho;

Aos amigos e companheiros de laboratório por me receberem como parte do grupo e por terem incentivado meu crescimento e a conclusão deste trabalho;

A CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pelo indispensável apoio financeiro através do edital INCT - MCTI/CNPq/CAPES/FAPs n^{o} 16/2014, processo número 88887.313212/2019-00;

Ao CNPq pelo apoio financeiro através do Projeto DRONI, chamadas 402112/2013-0 e 381682/2019-6;

Ao programa Erasmus Mundus SMART^{2} que tornou possível minha mobilidade para Portugal através da chamada 552042-EM-1-2014-1-FR-ERA MUNDUS-EMA2;

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Abstract

This thesis is in the context of autonomous airships. Airships are advantageous platforms for long duration missions because the lifting gas sustain part of the vehicle weight, reducing energy consumption. Also, airships are platforms with low degree of intrusion in the environment, which is an important characteristic for environmental monitoring applications. This thesis is placed in the context of the project DRONI. This project aims to develop an autonomous airship for the enviromental monitoring in the Mamirauá reserve in Amazon rainforest. Airships are known to have a nonlinear dynamics with many uncertain parameters. The identification of such parameters require complex testing, such as wind tunnel tests. This thesis proposes Incremental Controllers (ICs) as a solution for the control loop in order to overcome the modeling lack. ICs reduce the sensitivity to model uncertainties by adding the dependency of derivative measuring, which increases the sensitivity to measurement noise and delay. Furthermore, actuator redundancy and saturation are issues when designing ICs for airships. Therefore, in this thesis, we discuss how to design ICs presenting solutions for the design issues mentioned before. Nevertheless, the control and guidance loops require access to linear and angular velocities. Meanwhile, the guidance must have access to positions, attitude, and incident wind in the airship body. These information can be retrieved by inertial sensors (such as gyroscope, accelerometer and magnetometer), GPS, and barometer. Thus, a comparison between different filtering strategies is presented in order to define which technique provides better feedback for the control and guidance loops. Also, the wind estimation problem is addressed. Traditional model-based estimators are presented, then a data-driven strategy is proposed. The results obtained led us to propose a novel estimation strategy that performs a fusion between both model-based and data-driven approaches resulting in a hybrid version of a wind estimator. Finally, the guidance problem is addressed. A modified version of the so-called Line-of-sight strategy is presented. This strategy receives adaptations for considering the kinematic restrictions of the airship and the varying behavior when transitioning between hovering and cruise flights. As a final result, we obtained a closed loop system addressing Guidance, Control and Estimation problems. As demonstrated in simulation results, this final control architecture is capable of covering all flight phases of a complete mission, namely: cruise, hovering, vertical take-off and landing.

Keywords: Robots - Control systems; Flight - Control; Auto-guided vehicle systems; Kalman Filtering; Neural Networks;

Resumo

Esta tese está contextualizada no âmbito de dirigíveis autônomos. Os dirigíveis são plataformas vantajosas para aplicações de longa duração devido sua capacidade de sustentar parte de seu próprio peso, reduzindo o consumo de energia. Além disso, os dirigíveis são plataformas com baixo grau de intrusão, o que é uma característica muito importante para aplicações de monitoramento ambiental. Esta tese está contextualizada dentro do projeto DRONI. Este projeto visa desenvolver um dirigível autônomo para monitoramento ambiental da reserva de Mamirauá na floresta Amazônica. Os dirigíveis são conhecidos por sua dinâmica não-linear com várias incertezas paramétricas. Esta tese propõe Controladores Incrementais (ICs) como uma solução para a malha de controle a fim de contornar os erros paramétricos de modelagem. Os controladores incrementais reduzem a sensibilidade do sistema à erros de modelagem adicionando a dependência de medição das derivadas dos estados. Consequentemente, ICs aumentam a sensibilidade do sistema aos ruídos e atrasos de medida. Adicionalmente, redundância entre atuadores e saturação devem ser levados em conta ao projetar ICs. Portanto, nesta tese, discutimos como projetar ICs apresentando soluções para estes problemas. No entanto, as malhas de controle e guiamento necessitam de informações tais como velocidade linear e angular, posição geográfica e orientação. Estas informações podem ser extraídas através de sensores inerciais (tais como girocópio, acelerômetro e magnetômetro), GPS e barômetro. Portanto, uma comparação entre diferentes técnicas de filtragem é apresentada com o objetivo de definir qual técnica melhor se encaixa às malhas de controle e guiamento. Além disso, o problema de estimação do vento é tratado. As técnicas tradionais baseadas em modelo são apresentadas e discutidas. Então, uma estratégia inovadora baseada em dados é proposta. Os resultados obtidos nos levaram a propor uma estratégia invadora resultado da fusão entre a técnica baseada em modelo e a técnica baseada em dados, resultando em um técnica híbrida de estimação de vento. Finalmente, o problema de guiamento é tratado. Uma versão modificada da estratégia Line-of-sight é apresentada. Esta estratégia recebeu adaptações para considerar as restrições cinemáticas do dirigível e seu comportamento variante no tempo quando em transição entre voo pairado e voo cruzeiro. Como resultado final, obtivemos um sistema em malha fechada composto por Guiamento, Controle e Estimação. Como demonstrado nos resultados de simulação, a arquitetura de controle proposta é capaz de cobrir todas as fases de voo do dirigível em uma missão completa, nomeadamente: voo cruzeiro, voo pairado, decolagem e aterrissagem vertical.

Palavras-chave: Robôs - Sistemas de controle; Voo - Controle; Sistemas de veículos autoguiados; Filtragem de Kalman; Redes Neurais;

List of Figures

List of Tables

Nomenclature

Variables

α the angle of attack

β the side slipe angle

Λ, λ Input scaling gain (matrix, scalar)

$\Omega = [p \;\; q \;\; r]^T$ the three-dimensional airship angular velocity in body frame (roll, pitch and yaw angular velocities)

Ω_w the angular velocity of the wind

$\Phi = [\phi \;\; \theta \;\; \psi]^T$ the three-dimensional airship angular position in the NED frame (roll, pitch and yaw angles)

ξ the airship vector of kinematic states

a_g the gravitational acceleration

M_v the virtual mass

S_a the rotation matrix from the wind frame to the body frame

δ_i the normalized voltage applied to the i-th thruster

$\mathbf{w}$, w State derivative (vector, scalar)

ω_i the angular velocity of rotation of the i-th propeller

ρ the air density

τ_m the mechanic time constant

τ_e the electric time constant

$\mathbf{F}_a$ the aerodynamic forces and moments

$\mathbf{F}_g$ the gravity forces and moments

$\mathbf{F}_k$ the kinematic forces and moments

$\mathbf{F}_p$	the wind-induced forces and moments
$\mathbf{F}_w$	the wind-induced forces and moments
$\mathbf{I}_3$	the identity matrix of third order
$\mathbf{J}$	the inertia matrix of the airship
$\mathbf{J}_B$	the inertia matrix of the buoyancy air
$\mathbf{J}_v$	the virtual inertia matrix
$\mathbf{M}_a$	the apparent mass of the airship
$\mathbf{N}_i$	the normalized vector of orientation in 3-axis of the i-th propeller
$\mathbf{O}_c$	the vector of distance between CG and CB
$\mathbf{O}_{c\times}$	the skew matrix which translates the Center of Gravity (CG) components to CB
$\mathbf{O}_{CT_i}$	the distance vector between propeller and CB
$\mathbf{P} = [P_N \;\; P_E \;\; P_D]^T$	the three-dimensional airship cartesian position in the NED frame
$\mathbf{R}$	the rotation matrix for transforming velocities from NED frame to body frame
$\mathbf{S}$	the rotation matrix from the body frame to the NED frame
$\mathbf{V} = [u \;\; v \;\; w]^T$	the three-dimensional airship linear velocity in body frame
$\mathbf{V}_a$	the airspeed
$\mathbf{V}_w$	the wind speed
$\mathbf{x}$	the airship vector of dynamic states in body frame
$\mathbf{y}$, y	System output (vector, scalar)
ζ, ω_n	Damping ratio, natural frequency
C_{P_i}	the propulsion coefficient of the i-th propeller
C_{T_i}	the thrust coefficient of the i-th propeller
$f(\mathbf{x}, \xi, \mathbf{d}, \mathbf{u})$	the dynamic equations of the airship model
F_i	the force generated by the i-th propeller
I_i	the current in the i-th thruster

J_m	the moment of inertia of the rotor
k_t	the coefficient of momentum
m	the scalar mass
m_B	the buoyancy mass
M_i	the momentum generated by the i-th propeller
r_{prop}	the propeller radius
V_t	the true airspeed
V_{max}	the maximum voltage of the battery

Subindexes

$(\cdot)_i$	i-th component of a vector

Acronyms

ALOS	Adaptive Line-of-sight
CB	Center of buoyancy
CG	Center of gravity
DRONI	Robotic Airship with Innovative Conception
EKF	Extended Kalman Filter
EKF	Unscented Kalman Filter
ENU	East-North-Upper
GPS	Global Positioning System
IBKS	Incremental Backstepping
ILOS	Integral Line-of-sight
INDI	Incremental Nonlinear Dynamic Inversion
ISDM	Mamirauá Institute of Sustainable Development
ISG	Input Scale Gain
ISM	Incremental Sliding Modes

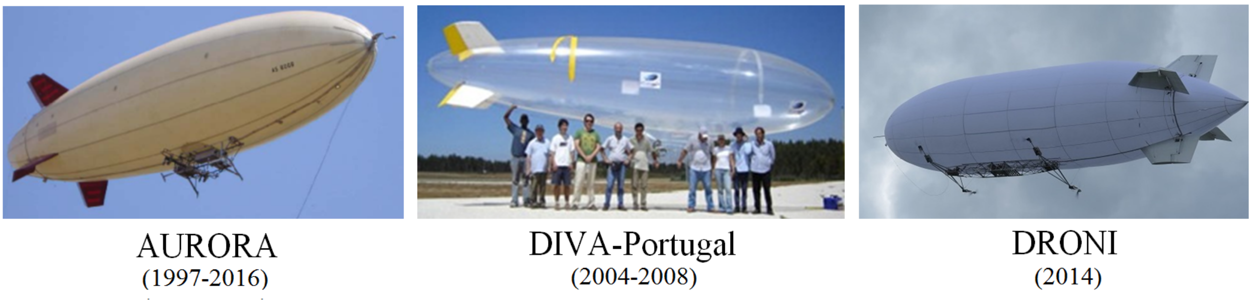
KF	Kalman Filter
LOS	Line-of-sight
LPF	Low-pass filter
MIMO	Multiple Inputs and Multiple Outputs
NAW	Natural Anti-Windup
NED	North-East-Down
NED	North-East-Down
RMS	Root Mean Square
ROS	Robotic Operating System
RS	Reference shapping
SB(A/I)LOS	Sensor Based (Adaptive/Integral) Line-of-sight
SOD	Second-order Differentiator
UAV	Unmanned Aerial Vehicle
VPC	Vertical Position Control
VTOL	Vertical Take-off and Landing
WOF	Washout Filter
ZOH	Zero-Order Hold

Contents

1 INTRODUCTION

Airships with autonomous operation capacity are the focus of a worldwide investigation. A less common aerial vehicle, airships are particularly suited to scenarios that demand long endurance, high payload, and low operational risks, such as surveillance and environmental monitoring [Liu et al., 2009]. Airships are advantageous platforms for long duration missions because they sustain part of its weight, reducing energy consumption [Elfes et al., 2003]. Also, airships are platforms with a low degree of intrusion in the environment, since they emit less sound noise than other UAVs, which is an important characteristic for environmental monitoring applications. Relevant works have shown many useful tasks for airships, such as: flight data acquisition [Kungl et al., 2004], terrain mapping [Hygounenc et al., 2004], vision surveillance [Moutinho et al., 2007] and deep space exploration [Elfes et al., 2004].

Unmanned Airships typically are equipped with two combustion rotors. However due to the increasing advances in battery, the combustion rotors are being switched by electrical rotors. At the same time, the multirotors (which are sustained by rotative wings) are being widely studied around the world. Although they present overactuation and high maneuverability, the multirotors are characterized by the short flight time capacity. Considering the multirotors advantages and the airships advantage, a novel configuration vehicle was proposed by the project DRONI - Robotic Airship with Innovative Conception. This project is conducted by “Centro de Tecnologia da Informação Renato Archer” (CTI) in partnership with several institutions – UNICAMP, UFAM, ITA, IST, USP-São Carlos, and IDSM. This partnership between Brazilian and Portuguese institutions (UNICAMP, CTI, and IST) was first established in 1997 with the Project AURORA [Elfes et al., 1998]. It was maintained during the project DIVA [Moutinho et al., 2007] and now it also includes other universities (UFAM, USP-São Carlos, and ITA) in project DRONI (see Figure 1.1).


AURORA
(1997-2016)

DIVA-Portugal
(2004-2008)

DRONI
(2014)

Figure 1.1. Projects AURORA, DIVA e DRONI

1.1 Background in Autonomous Airships

Aiming at the design of an autonomous airship, aerial platform positioning and path-tracking should be assured by a control and navigation system. In the field of robotics, autonomous navigation involves four basic topics: Path planning, Guidance, Control, and Data Acquisition. This separation is advantageous since it modularizes the platform allowing to switch the guidance strategy without losing performance and stability guarantees given by the control loop or estimation loop, for example. Also, the tuning is easier, once the tests can be done with a modular procedure.

The Path Planner is responsible for generating a trajectory for the Guidance loop. Normally, the Guidance aims to minimize the pose error (in other words, position and orientation). While, the Control level aims to correct velocity and/or acceleration errors. The Data Acquisition includes subjects of sensing, filtering, and estimation, which aims to extract the necessary information about the vehicle motion for closing the control and guidance loops.

In Figure 1.2 is shown a general block diagram in a cascaded structure which represents the overall navigation system of a robot, where $\mathbf{x}$ is the vector of linear and angular velocities in body frame, ξ is the vector of linear and angular positions in the global frame, $\mathbf{u}$ is the control input for the airship, and $\mathbf{d}$ is the disturbance vector and the subscript $._d$ is used for commanded signals.

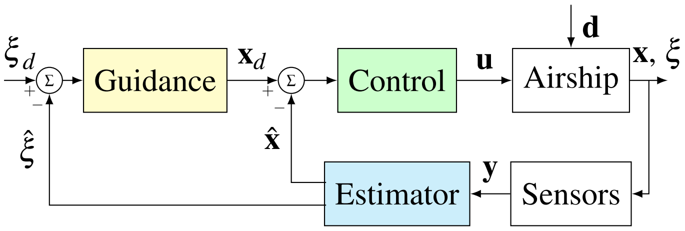


Figure 1.2. General block diagram of the navigation system.

For each of the blocks in Figure 1.2, a section is written presenting the studies and developed solutions in the context of autonomous airships.

1.1.1 Guidance Strategies

The Guidance is the intermediary level of control responsible to provide velocity and attitude references to the control level. In robotics, a path is defined as a sequence of poses (position and orientation) in the map frame. However in the context of Unmanned Aerial Vehicles (UAVs), the guidance is divided into two steps: position tracking and attitude control as shown in Figure

1.3. Normally, linear control strategies such as Linear Quadratic Regulator (LQR) [Moutinho et al., 2016] and Proportional-Derivative-Integral (PID) [Azinheira et al., 2000] are adopted for these steps.

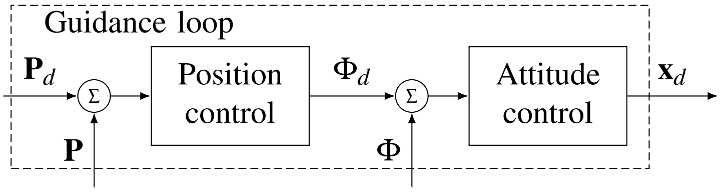


Figure 1.3. General block diagram of the guidance loop of an Autonomous Airship

There exist two basic approaches for the completion of a path, namely: *path tracking* and *path following* as defined by [Tsourdos et al., 2010]. In the *path tracking* approach, the path is parameterized by the time. In other words, each pose has a time compromise. It works as if there existed a virtual vehicle moving along the path which is the reference for the real vehicle. In *path following*, the vehicle must follow the trajectory, however without a temporal compromise of execution as shown in [Moutinho et al., 2016].

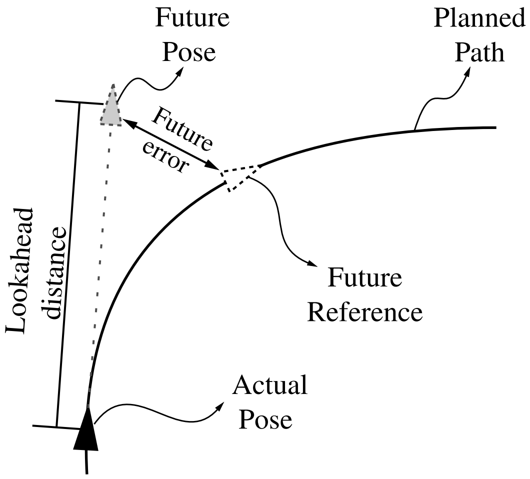


Figure 1.4. Line-of-sight strategy applied to an UAV

Almost 20 years ago Line-of-sight (LOS) was introduced by [Lekkas and Fossen, 2003, Rysdyk, 2003]. This strategy complements Path following/tracking tasks by considering future position errors, however, without the temporal compromise. The idea is that a future position error can be reduced by acting before it occurs. As an example, consider a vehicle laterally underactuated such as a missile. Considering that the target moves (e.g. by doing an accentuated curve), the projectile must initialize a correction immediately to achieve the final goal. This solution was used in [Fossen et al., 2015] by inserting weights for the actual pose error and the future pose error through the regulation of a lookahead parameter as illustrated by Figure 1.4. This strategy is interesting for laterally underactuated vehicles as is the case of airships.

Similar to LOS, [Park et al., 2004] proposed the L1 approach. This strategy introduces some kinetic terms to be compensated, such as centripetal acceleration. Thus, the guidance level generates smoother references to the control level and more compatible with the vehicle dynamics.

Another way of computing smooth errors is the Vector Fields technique proposed by [Goncalves et al., 2010]. This approach uses the gradient of the path to generating smooth velocity references for the control level. This strategy was proved to be stable and also it was tested in real vehicles such as: fixed-wing aircraft models, quadrotors and holonomic terrain vehicles [Gonçalves et al., 2010, Avellar et al., 2013]. While quadrotors are over-actuated and fixed wings are always flying with high airspeed, airships are slower and underactuated. Thus, in the context of airships, it is challenging due to its actuator restrictions. Also, wind drag and the uncertain lateral modeling are challenge issues when designing Vector Fields for airships.

In Chapter 7, a careful review in LOS guidance is presented. Then, an improvement is proposed in order to perform hovering flight. A comparison is established between the many variations of LOS theory through simulations in the nonlinear model of the NOAMAY airship.

1.1.2 Airship Control

A reliable navigation system and a robust controller are key factors for the success of monitoring and surveillance tasks. In this context, several control strategies have been proposed. Previous techniques used only the yaw rate to go-to-waypoint missions [Azinheira et al., 2000]. Then, nonlinear control techniques were proposed aiming at the accomplishment of path-following tasks, such as: Backstepping [Hygounenc and Soueres, 2002]; Sliding Modes [Paiva et al., 2009]; and Dynamic Inversion [Moutinho and Azinheira, 2006]. However, these strategies require the identification of aerodynamic and inertial parameters, which are usually uncertain. Additionally, control allocation is an issue for these strategies, because the theory behind them consider actuation directly through forces and moments [Vieira, 2019].

Thus, Adaptive Control Laws (ACL) have been proposed to handle model uncertainties. In [Acosta and Joshi, 2007], an adaptive version of the Nonlinear Dynamic Inversion was proposed followed by works such as [Wu et al., 2012], which propose ACL for trajectory tracking using models derived from the vehicle equations of motion.

Even though ACL increases the robustness of the controllers, they still depend on the identification of aerodynamic and inertial parameters, which requires expensive experimentation. For the airship example, the identification process uses wind tunnel tests with the entire envelope.

Also, Hovering and VTOL flight phases require specific control design. A few works address this problem. For instance, in [Gobiha and Sinha, 2018] the authors overcome the hovering problem assuming it is a cruise flight under a small corridor around the desired position. In

[Zheng et al., 2014], an observer was designed to estimate the wind attitude in order to compensate for the disturbance into the outer-loop of a cascade baseline controller. In [Azinheira et al., 2006], the hovering controller was designed using Backstepping considering nonlinearities such as the actuator saturation. In [Daskiran et al., 2017], a Reinforcement learning technique was introduced to overcome the problems of underactuated and uncertain dynamics during hovering flight.

Another common issue is an abrupt, although continuous, transition between the two flight phases, namely: cruise and hovering. Each phase requires a different use of actuators. The most common solution is to switch between two controllers, one for each phase.

Incremental controllers are promising techniques to cover a complete mission, composed by: cruise, hovering, Vertical Take-Off, and Landing (VTOL) flight phases. They use solely the modeling of input-dependent parameters of the system. The dynamics of the system states are approximated by the value of the feedback states and their derivatives at each sample time. As shown by [Wang et al., 2018], the Incremental Nonlinear Dynamic Inversion (INDI) is tolerant of external disturbances and model uncertainties. As a sensor-based approach (in contrast to model-based ones), the method assumes that all needed states and derivatives are measured at a sufficiently high sampling rate. Also, the control action needs to be fast when compared to the system dynamics. Related works on aerial vehicles have shown promising results using INDI to control different platforms, such as: helicopters [Simplício et al., 2013], unmanned tiltrotors [Francesco and Mattei, 2016], and quadrotors [Smeur et al., 2016]. In 2015, INDI was introduced in the scenario of airships [Azinheira et al., 2015], addressing only the lateral control in cruise flight, where the underactuation and the longitudinal coupling in the lateral motion are negligible.

In Chapter 3, the incremental controllers are introduced presenting the theory behind Incremental Dynamics and generic formulations of INDI. Furthermore, solutions for mitigating INDI design issues are proposed. Numerical examples and closed-loop analysis highlight the benefits and drawbacks of the proposed solutions. Finally, simulations in the airship model take place showing the performance and robustness of the control loop.

1.1.3 Filtering and Estimation

Airships are known to be nonlinear and underactuated systems. Thus, autonomous navigation is a difficult task to accomplish. State estimation techniques are crucial to the development of guidance and control techniques. Non-linear control strategies such as Sliding-modes as proposed by [Vieira et al., 2017] and INDI proposed by [Azinheira et al., 2015] require the knowledge of velocities and accelerations. Meanwhile, guidance requires information about attitude, position, and airspeed as stated by [Moutinho et al., 2016].

The solution for this problem relies on filtering, estimation, and sensor fusion methods. Several algorithms have been developed to address this issue. A traditional algorithm is the Kalman Filter, which can be applied to linear systems. While, its extensions such as the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are switched to nonlinear systems.

In [Kim et al., 2006], it is shown that such approaches are also useful for treating information redundancies that are common in sensor fusion problems. Nevertheless, [Oh, 2010] compensates sensor dynamics (such as bias) by adding some modifications to the algorithm.

The choice of an estimation technique plays a crucial role in any autonomous navigation system. Aerospace systems are critical and usually do not tolerate failure. As a consequence, they demand comparative studies, in simulation environment, before embedding the chosen technique(s) on the platform. As an example, in [Giannitrapani et al., 2011], the authors presented a performance comparison between EKF and UKF addressing the spacecraft localization problem in a simulation environment.

This topic will be also addressed in this thesis. In Chapter 5, we present a comparison between EKF, UKF, and Low-pass Filter (LPF) for estimating pose and velocity of NOAMAY. We consider IMU, GPS, Pitot Tube, Barometer, and Thermometer as available sensors.

1.1.4 The Wind Estimation Problem

Outdoor airships commonly have a guidance control to track a trajectory. The first idea is that the attitude reference shall be coincident with the reference trajectory attitude. However, there are two situations when this is not desirable: in the presence of wind disturbances (an almost certainty when flying outdoors) and if the objective is ground-hover (since the desired attitude is arbitrarily defined).

An aircraft of conventional shape must fly against the apparent wind to have low drag. This is also true for airships, especially due to of the lateral underactuation [Moutinho et al., 2016]. Therefore, in the presence of wind, the airship must try to align itself with the relative airspeed, thus reducing the sideslip angle. This implies that guidance control also depends on information about wind velocity and attitude. However, measuring such elements is not a trivial task.

The most common solution is to estimate the wind velocity to extract the necessary information about the vehicle motion. Model-based techniques are the most used strategies in the literature. As an example, in [Perry et al., 2008], it is proposed an approach for estimating the angle of attack and sideslip angle by the kinematic equations of motion of an aerobatic UAV. With the same kinematic equations, in [Cho et al., 2011] an Extended Kalman Filter (EKF) is proposed for estimating the wind heading and velocity using an aircraft with a single GPS and Pitot tube. In [Johansen et al., 2015] a wind velocity observer also based in the kinematics is proposed for small UAVs with experimental results. Similarly, in [Shen et al., 2015] it is also proposed an EKF for wind velocity estimation, however applied to a Stationary stratospheric airship in a simulation environment. Then in [Rhudy et al., 2017] four Model-based solutions are presented considering an aircraft with four different possible configurations of sensors.

Machine Learning has become popular in the field of robotics. The impressive growth of computational resources and increasing acquired data over the years have increased the potential of these Data-driven techniques. These strategies were already introduced in applications such as control of aircraft [Chaturvedi et al., 2002] and air data estimation for a Micro-UAV [Samy et al., 2010]. The wind estimation problem is addressed in [Allison et al., 2019] using a quadrotor with a Machine-Learning approach. However, a Data-driven online estimation of wind velocity for robotic airships is still a challenge.

In Chapter 6, this problem is addressed. As a result, three alternative versions of wind velocity estimation using Model-based and Data-driven techniques are presented.

1.2 Motivation

The main objective of the project DRONI is to develop an autonomous airship capable of perform monitoring tasks in the Mamirauá Sustainable Development Reserve (www.mamiraua.org.br) in the Amazon Rainforest. The Mamirauá Institute of Sustainable Development (ISDM), is located in the center of the Amazon Rainforest at a distance of 500 km from Manaus. It contains several research spots to provide support to the researchers who conduct research works related to the understanding of the Amazon biome. Such researches depend on a systematic data collecting method. Thus, they demands a platform with high maneuverability, long-duration flight capability, and easy usability.

The airship developed during project DRONI was named as Noamay [de Oliveira, 2018]. It is instrumented with an inertial sensor known as XSens 700 MTi-G which includes: gyroscope, accelerometer, magnetometer, GPS, Barometer, and Thermometer [Rueda et al., 2017]. Also, it is equipped with a Camera located at the gondola, and a Pitot Tube located at the airship nose. The airship is composed by a hull with 11m length, and 2.48m diameter equipped with: 4 vectored propellers with independent thrusters, and tail surfaces (rudder, elevator, and aileron). In March 2018, Noamay performed its first remote controlled flight (see Figure 1.5).

Nowadays the DRONI project is in the phase of research and development of automatic controllers to cover all flight phases, namely: cruise and hovering flight; Vertical Take-off and Landing (VTOL). Hovering flight is the condition where the airship performs a stationary flight in relation to a fixed reference in the ground. Meanwhile, cruise flight is the flight phase where the airship performs a moving flight.

Figure 1.5. Robotic airship performing its first flight

A common issue is the abrupt although continuous transition between the flight phases. Each phase requires different controllers as shown in [Azinheira et al., 2006]. The lateral underactuation in hovering flight was the main cause for proposing this strategy. In the hovering scenario, the lateral and longitudinal motion are coupled. Meanwhile in cruise flight both motions can be decoupled as shown in [Khoury, 2012].

The Noamay is a particular case of an airship with overactuation in the longitudinal (and vertical) motion due to its propellers redundancy. There are many possible choices of inputs to achieve a target velocity in the body frame. However, such inputs must be carefully chosen by considering parameters such as: energy efficiency and flight performance.

Airships are known to have a nonlinear dynamics with many uncertain parameters. The identification of such parameters requires complex testing, such as wind tunnel tests. Also, tunnel tests are not sufficient to identify aerodynamic parameters of the lateral motion of airships. Additionally climate conditions and altitude are issues in the modeling task. Therefore, these tests are uncertain, expensive and insufficient for modeling all dynamic behavior of an airship. Thus, it is difficult to design controllers with techniques that depend of the knowledge of the model. This thesis proposes Incremental Controllers (ICs) as a solution for the control loop in order to overcome this modeling lack.

1.3 Objectives

As a general objective, this thesis aims to develop a control architecture for the navigation system of a robotic airship using incremental controllers capable of covering all flight phases (VTOL, cruise and hovering flight) and transitioning between them without a complex switching scheme.

1.3.1 Specific Objectives and Contributions

This thesis has four main specific objectives:

- Understand the INDI theory and how it can be applied to the airship;
- Provide solutions for INDI design issues;
- Design INDI for decoupled lateral and logitudinal motions of the Noamay airship;
- Explore solutions for estimation and filtering of the data provided by the available sensors, in order to feedback the guidance and control loops;
- Estimate wind velocity and direction in order to provide wind velocity and direction for the guidance loop;
- Investigate and choose a guidance technique in the literature;
- Propose improvements for the chosen guidance technique in order to compensate wind drag when transitioning between cruise and hovering flight.

1.4 Thesis Organization

ICs reduce the sensitivity to model uncertainties by adding the dependency of derivative measuring, which increases the sensitivity to measurement noise and delay. Such characteristics could be introduce instability to the systems. Also, Noamay presents actuator redundancy which may lead to excessive energy consumption. Additionally, all actuators includes saturations which may cause a loss of performance in the control loop. Therefore, Chapter 3 presents a survey on ICs as well as a tutorial of how to design ICs presenting solutions for the design issues. Finally, in Chapter 4, the INDI control is designed for the Noamay airship lateral and longitudinal motions separately, presenting preliminary simulation results.

Much information about the airship motion is necessary for feedback on the control and guidance loops. The control must have access mainly about the airship linear and angular velocities. Meanwhile, the guidance must have access to positions, attitude, and incident wind in the airship body. Thus, in Chapter 5 a comparison between different filtering strategies is presented in order to define which technique provides better feedback for the control and guidance loops.

Also, the wind estimation problem is addressed in Chapter 6. Traditional model-based estimators are presented, then a data-driven strategy is proposed. The results obtained led us to propose a novel estimation strategy that performs a fusion between both model-based and data-driven approaches resulting in a hybrid version of a wind estimator.

Finally, the guidance problem is addressed in Chapter 7, which presents a modified version of the so-called Line-of-sight strategy. This strategy receives adaptations for considering the kinematic restrictions of the airship and the varying behavior when transitioning between hovering and cruise flights.

1.5 Publications

During the doctoral program, the Ph.D. candidate has published two Conference papers and two Journal papers.

The first conference paper was published as a co-author in "Simpósio Brasileiro de Automação Inteligente" (SBAI) with the title "Controle de um Dirigível Robótico Autônomo de Propulsão Quádrupla utilizando Modos Deslizantes" [Vieira et al., 2017]. The paper addresses the problem of path tracking control for a robotic airship with quadruple propulsion in cruise flight. The paper presents the design of a sliding mode MIMO controller. To evaluate the controllers performance, a real scenario was simulated in which there is a path to be tracked in wind presence.

The second was published in the "Simpósio Brasileiro de Automação Inteligente - 2019" (SBAI - 2019) with the title "Comparative study for Robotic Airship State estimation with LPF, EKF and UKF" [Marton et al., 2019]. The paper presents a comparative study between three approaches of pose and velocity estimation for a robotic airship. The first approach is composed of multiples second-order Low-pass filters applied to GPS and IMU measured data. The second approach consists of the sensor fusion between GPS, IMU, Barometer, and Thermometer with an Extended Kalman Filter (EKF) based in the kinematic equations of motion for a six degrees of freedom (6-DOF) vehicle. Finally, the same 6-DOF equations of motion and sensors are used for the design of an Unscented Kalman Filter (UKF) as a third estimation approach. As result, we obtain greater precision with EKF with a minor advantage over the UKF.

Then in February 2020, a regular paper was published in "Journal of the Brazilian Society of Mechanical Sciences and Engineering" with the title: "Hybrid Model-Based and Data-Driven Wind Velocity Estimator for an Autonomous Robotic Airship" [Marton et al., 2020]. The paper presents three alternative versions for the estimation of wind velocity. Firstly, an Extended Kalman Filter is designed as a model-based approach. Then a Neural Network is designed as a data-driven approach. Finally, a hybrid estimator is proposed by performing a fusion between the previously designed estimators: model-based and data-driven. All approaches consider only Global Positioning System (GPS), Inertial Measurement Unit (IMU) and a one dimensional

Pitot tube as available sensors. Simulations in a realistic nonlinear model of the airship suggest that the cooperation between these two techniques increases the estimation performance.

Also, one other paper was accepted for publication [Cordeiro et al., 2021], which address the Lateral model of Noamay using INDI for the control loop. It provides a solution for mitigating noise and delay in the measurements using Input Scale Gain (ISG) and Second-Differentiator (SOD) with INDI. Firstly, a simple Single-input Single-output (SISO) model is used to evaluate the influence of the design parameters in the closed-loop response and stability. Then, the combined SOD and ISG approach is illustrated in a more complex case application where INDI is used to control the lateral motion of an autonomous airship, which results corroborate the ISG as an asset to increase the maximum allowed delay in the feedback loop.

2 AIRSHIP DESIGN AND MODELING

Airship dynamics have been extensively studied in the last two decades. Most of the research works consider the Newton-Euler equations to derive the dynamic model. The dynamic equations derive from the gravitational, kinematic, propulsion, wind, and aerodynamic forces. There are many works focused in airship modeling that we can cite here, such as [Elfes et al., 2002, Mueller et al., 2004, Moutinho et al., 2016, Li, 2008].

Dynamic models of airships are mostly nonlinear and become even more complex when analyzing outdoor airships because of aerodynamics. Aerodynamic parameters are normally estimated through tests of wind tunnel [Jones and Delaurier, 1983] or in experimental flights [Patino et al., 2005].

Even though this thesis does not address the modeling problem, in order to present control solutions for Noamay, some concepts about airship design and modeling shall be clarified.

2.1 Fundamentals of Airship Design

Airships are unique aerial vehicles which are quite different from conventional fixed-wing aircrafts. Usually, most of the airship lift is generated by a lifting gas, such as hydrogen or helium. As a consequence, airships are generally designed with an ellipsoid shape which is an inefficient producer of aerodynamic lift.

The airship lifting force is generated by a density difference between the volume of lifting gas (inside the envelope) and ambient air. Such vertical force is a result of the Archimedes principle. About 2300 years ago, Archimedes postulated and later proved that buoyant force depends only on the body volume and the density difference between the submerged body and its surrounding fluid (liquid or gas) [Carichner and Nicolai, 2013]. This force is located at the Center of Buoyancy (CB), which is typically approximately equal to the Center of Volume (CV). In aircraft design, the efforts are concentrated on maintaining a proper CG location that results in good performance and handling. For airship design, the CB is an additional design parameter that must be positioned carefully. Generally, the CB is placed slightly aft of the CG so that there is a slight nose-down moment from the lifting gas as shown in Figure 2.1.

In history, there are two main concepts of airship structural design, namely: *rigid* and *non-rigid*. Although these are the most commons, there is an intermediate concept named as *semi-*

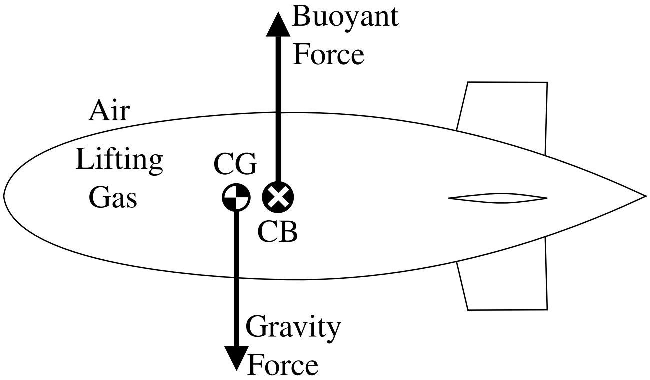


Figure 2.1. Buoyant and Gravity forces acting on an airship

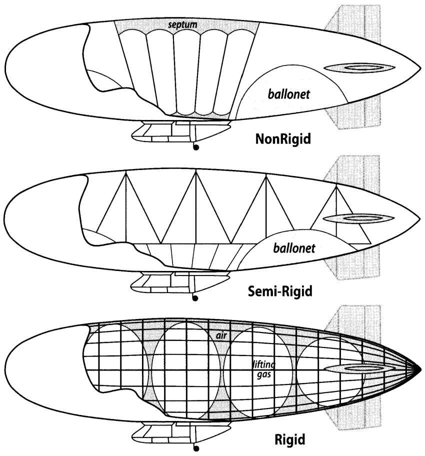


Figure 2.2. Comparison of structural concepts for body of revolution airship [Carichner and Nicolai, 2013].

rigid as shown by [Carichner and Nicolai, 2013], however, it still has not proven to be commercially viable. As a result, there are very few successful semi-rigid designs. Unlike the rigid design the non-rigid and semi-rigid designs use a flexible bag called ballonet inside the main envelope as shown in Figure 2.2. The main objective of the ballonet is to prevent any variation of the differential pressure. In nonrigid designs, the interior gas may change its density in response from a temperature variation (e.g. due to climate or altitude changes). As a consequence, the differential pressure across the envelope is also changed.

In rigid designs, this variation is negligible since the lifting gas is allocated in bags attached to the rigid structure and it has sufficient volume to operate at its maximum altitude. The envelope of a rigid structure has only aerodynamic purposes.

In the last 20 years, the technological advancements have improved the design of airships by introducing the concept of hybrid airships. These airships are extremely different from conventional ships and airships of 80 years ago. Most notably in ships, the buoyancy principle, postulated by Archimedes, is vastly used for the transportation of goods around the world. The main idea is that, the static or buoyant lift (due to gas) is always “on”. Meanwhile, dynamic lift (due to aerodynamics and thrust vectored) can be turned “on” and “off” when it is convenient. Thus, hybrid airships are designed to use both lifting sources such as the Airlander from Figure 2.3 and the Hybrid Airship from Figure 2.4 projected by the Lockheed Martin company.

This combination of buoyancy and aerodynamic lift makes hybrid designs superior when transporting goods. A conventional airship is near “neutrally buoyant” and it becomes dramatically light when offloading. A hybrid airship is designed to be partially buoyant, in other words, i.e. somewhat “heavier than air”. Such characteristic allied to its shape causes an increase of the aerodynamic lift. Such ability of lift modulation allows a significant increase operational flexibility. Moreover, it permits the offloading of larger payloads without losing its controllabil-

Figure 2.3. Photograph of HAV 304 Airlander produced by Hybrid Air Vehicles company

Figure 2.4. Conceptual image of the Hybrid Airship project from Lockheed Martin company

ity. In contrast, surveillance missions where the vehicle is preferable to stay in static positions are more difficult for the hybrid design, since it has to compensate its heaviness with thrusters.

The tail surfaces of airships are another major contributor to airship behavior. They are surfaces similar to wings that generate aerodynamic forces through their flexible edge control surfaces. Because the airship needs to move both up and down through aerodynamic forces generated by those surfaces, the tails will normally be symmetric airfoil sections. These aerodynamic forces are responsible for producing stabilizing moments in order to move along a desired path and compensate for the drag provoked by external disturbances.

Optimizing aerodynamic performance for an airship is a complicated task. It involves a trade-off between several variables with a multi-objective function including endurance, range, speed, altitude, etc. Usually, the optimization processes are similar to the fixed-wing aircrafts with some additional variables such as the size of ballonets, amount of buoyancy ratio, body cross-section, shape, envelope material properties, and volume.

Another parameter of concern is known as virtual mass (or added mass). Generally, the virtual mass phenomenon is poorly understood because it is not significant enough to impact airplane performance. Consequently, it is not included in the optimizing task when designing fixed-wing aircrafts. However, the weighting mass is always present for any object which is moving through a fluid. Basically, a body in a fluid behaves as though it has more mass than it actually does. Such behavior varies with the motion nature (accelerating, decelerating, or turning). Thus, a significant effect on the dynamics may be experienced by vehicles that have a mass similar to the displaced external fluid (such as airships).

2.2 Aspects about the Motion and Design of Noamay Airship

The Noamay airship has a nonrigid body filled with helium gas without a ballonet. Additionally, it is equipped with four vectoring propellers with independent thrusters and tail surfaces (see figure 2.5). It combines aerodynamics, buoyancy, and the traditional quadrotor maneuverability characteristics.

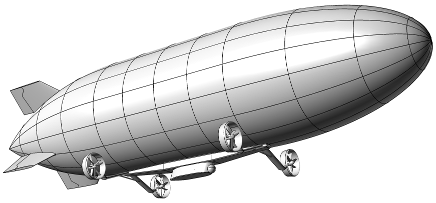


Figure 2.5. Conceptual view in perspective of the Noamay airship

Noamay is in cruise flight when the aerodynamic forces sustain mostly of its apparent mass. Normally, it occurs when flying in high longitudinal speed relative to ground (or ground speed), generally over 3m/s. In such condition, the vectoring capability of thrusters let the flying task much easier. Also, lateral underactuation is not an issue, since the tail surfaces have high authority over the dynamics.

In contrast to cruise, when in VTOL or hovering flight (less than 3m/s of ground speed), the thrusters are the main source of actuation. The tail surfaces may not have sufficient authority. Thus, the thrusters are responsible for sustaining the apparent mass and for doing maneuvers by applying differential rotation between propellers, similar to a conventional quadrotor.

However, considering external disturbances (such as wind) and the huge lateral area of the envelope, forces and momentums generated by thrusters may be insufficient for controlling the yaw rate. To perform hovering and VTOL tasks, the cooperation with wind is essential. Under wind, the airship may have a higher or lower relative speed to the air (or airspeed) which will virtually influence in the vehicle stability.

When the wind direction is against the airship nose, it has a higher airspeed. Consequently, the tail surfaces have higher authority. In such a condition, the drag can be compensated through the thrusters and tail surfaces letting the airship dynamics stable.

2.3 Airship Modeling and Simulation

The main tool to validate the proposed control and estimation approaches is a dynamical realistic non-linear model of an airship. This tool is a result of the research group efforts since the project AURORA [Elfes et al., 2002] which was improved during the project DIVA [Moutinho et al., 2016] and DRONI [Rueda et al., 2017]. Such a nonlinear model can be expressed in a state-space form as following:

$$\dot{\xi} = g(\xi, \mathbf{x}, \mathbf{d}), \tag{2.1a}$$

$$\dot{\mathbf{x}} = f(\xi, \mathbf{x}, \mathbf{d}, \mathbf{u}), \tag{2.1b}$$

where:

- the kinematic states $\xi = [\mathbf{P}^T \;\; \Phi^T]^T$ include the cartesian position $\mathbf{P} = [P_N \;\; P_E \;\; P_D]^T$ and angular position $\Phi = [\phi \;\; \theta \;\; \psi]^T$ in the North-East-Down (NED) frame (SI units);
- the dynamic states $\mathbf{x} = [\mathbf{V}^T \;\; \Omega^T]^T$ include the linear speed $\mathbf{V} = [u \;\; v \;\; w]^T$ and angular speed $\Omega = [p \;\; q \;\; r]^T$ in the body frame (SI units);
- the input vector $\mathbf{u} = [\delta_e \; \delta_a \; \delta_r \; \delta_0 \; \delta_q \; \mu_0]^T$ include: δ_e, δ_a and δ_r which are elevator, aileron and rudder deflection; δ_0 as the normalized thrusters voltage; δ_q as the differential voltage between the front-back thrusters; μ_0 as the common vectoring angle of the propellers (SI units);
- and, finally, the disturbance vector $\mathbf{d}$ that include wind velocities and gust parameters (see [McLean, 1990]).

Note that, although the propellers have independent vectoring, here we use the same angle for of all them through a single control input μ_0. Also, the thrusters are independent, however, we use two control inputs (δ_0 and δ_q), such that, when combined, produce input signals for each thruster. Further details will be given in the next subsections.

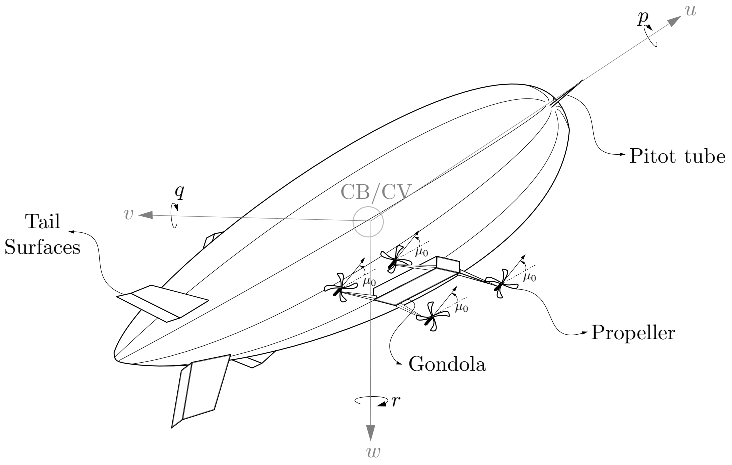


Figure 2.6. DRONI airship body diagram

The dynamics are based on the Newton-Euler equations including five components of forces and moments, namely:

- $\mathbf{F}_w$ – the wind-induced forces and moments;

- $\mathbf{F}_a$ – the aerodynamic forces and moments;
- $\mathbf{F}_p$ – the propulsion forces and moments;
- $\mathbf{F}_k$ – the kinematic forces and moments;
- $\mathbf{F}_g$ – the gravity forces and moments.

Therefore, the linear and angular accelerations are given by:

$$f(\mathbf{x},\xi,\mathbf{d},\mathbf{u}) = \mathbf{M}_a^{-1}\Big(\mathbf{F}_w + \mathbf{F}_a + \mathbf{F}_p + \mathbf{F}_k + \mathbf{F}_g\Big), \tag{2.2}$$

where $\mathbf{M}_a$ is the apparent mass matrix. These equations are referenced in the body frame centered in the Center of Buoyancy (CB) that is approximately equivalent to the Center of Volume (CV) as shown in Fig. 2.6. Moreover, some nonlinear effects are included in the simulation environment, such as: atmospheric pressure variation, temperature variation, lifting gas pressure variation, Coriolis effect and actuator nonlinearities (saturation, hysteresis, quantizer, delay and rate limiter).

When the displaced fluid mass is not negligible, as is the case for airships or balloons, the equations of motion are more usually derived from the Lagrangian approach (see [Lamb, 1918] and [Thomasson, 2000]). In the next subsections, a brief description of the apparent mass and each force component modeled in simulation is presented. More modeling details about the simulation environment can be found in works of [Moutinho, 2007, Arias, 2014, Moutinho et al., 2016, Vieira, 2019, Marton et al., 2019].

2.3.1 Apparent Mass

The apparent mass of airships is a combination between both, scalar (centered at the Center of Gravity) and buoyancy (centered at CB) mass and inertia elements. It can be mathematically modeled as:

$$\mathbf{M}_a = \begin{bmatrix} m\mathbf{I}_3 + \mathbf{M}_v & -m\mathbf{O}_{c\times} \\ m\mathbf{O}_{c\times} & \mathbf{J} + \mathbf{J}_v \end{bmatrix}, \tag{2.3}$$

where m is the airship scalar mass, $\mathbf{I}_3$ is the identity matrix of third order, $\mathbf{O}_{c\times}$ is a skew matrix which translates the Center of Gravity (CG) components to CB, $\mathbf{M}_v$ is the virtual mass, $\mathbf{J}$ is the inertia matrix of the airship and $\mathbf{J}_v$ is the virtual inertia matrix.

As discussed before, the virtual mass is a point of concern when modeling airships. It can be modeled as a combination of inertial coefficients as below:

$$M_v = \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 \\ 0 & 0 & Z_{\dot{w}} \end{bmatrix}, \qquad J_v = \begin{bmatrix} L_{\dot{p}} & 0 & 0 \\ 0 & M_{\dot{q}} & 0 \\ 0 & 0 & N_{\dot{r}} \end{bmatrix}, \tag{2.4}$$

where $X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}$ and $L_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}$ are virtual mass and inertia terms [Lamb, 1918].

2.3.2 Gravitational forces

The gravitational forces are given by a relation between the airship scalar mass, buoyancy mass and gravitational acceleration as below:

$$\mathbf{F}_g = \begin{bmatrix} (m - m_B)\mathbf{S}a_g \\ m\mathbf{O}_{c\times}\mathbf{S}a_g \end{bmatrix}, \tag{2.5}$$

where m_B is the buoyancy mass, $a_g = [0\ 0\ a_g]^T$ is the gravitational acceleration and $\mathbf{S}$ is a rotation matrix from NED frame to the body frame, given by:

$$\mathbf{S} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix}. \tag{2.6}$$

2.3.3 Kinematic forces

The kinematic forces appear as a consequence when deriving the momentum equations. The resultant equation is composed of inertial and centrifugal components, as shown below:

$$\mathbf{F}_k = \begin{bmatrix} -m\Omega \times (\Omega \times \mathbf{O}_c) - \Omega \times (m\mathbf{I}_3 + \mathbf{M}_v)\mathbf{V} \\ -\Omega \times (\mathbf{J} + \mathbf{J}_v)\Omega - \Omega \times (\mathbf{O}_{c\times} m\mathbf{V}) \end{bmatrix}, \tag{2.7}$$

where $\mathbf{O}_c$ is the vector between CG and CB.

2.3.4 Wind-induced Forces

The wind-induced forces appear as an external disturbance in the system caused by the introduction of a moving fluid in momentum equations [Azinheira et al., 2008]. They can be determined as follows:

$$\mathbf{F}_w = \begin{bmatrix} (m_B\mathbf{I}_3 + \mathbf{M}_v)\dot{\mathbf{V}}_w + \Omega \times (m_B\mathbf{I}_3 + \mathbf{M}_v)\mathbf{V}_w \\ (\mathbf{J}_B + \mathbf{J}_v)\dot{\Omega}_w + \Omega \times (\mathbf{J}_B + \mathbf{J}_v)\Omega_w \end{bmatrix} \tag{2.8}$$

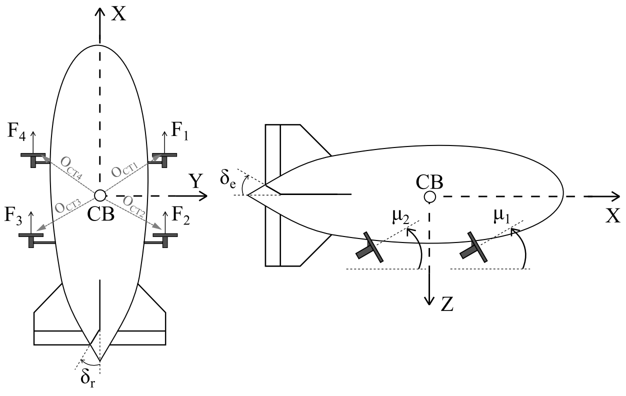
where $\mathbf{J}_B$ is the inertia matrix of the buoyancy air and Ω_w is the angular velocity of the wind.

2.3.5 Propulsion forces

The propulsion forces are mainly based on propellers characteristics and brushless motors dynamics. In [Moutinho, 2007], propulsion forces are modeled for a two-rotor airship and in [Arias, 2014] a quadrotor airship was modeled. The resultant propulsion force for Noamay can be described as below:

$$\mathbf{F}_p = \sum_{i=1}^{4} \begin{bmatrix} F_i\mathbf{N}_i \\ M_i\mathbf{N}_i + \mathbf{O}_{CT_i} \times \left(F_i\mathbf{N}_i\right) \end{bmatrix}, \tag{2.9}$$

where F_i is the force generated by the i-th propeller, $\mathbf{O}_{CT_i}$ is the distance vector between propeller and CB, M_i is the momentum generated by the i-th propeller and $\mathbf{N}_i$ is the normalized vector of orientation in 3-axis of the i-th propeller. In Figure 2.7 are shown the propellers numbering adopted here.



(a) (b)

Figure 2.7. Diagram of Noamay with: (a) upper view showing the propellers numbering and (b) side view showing the vectoring

The force and momentum generated by each propeller are given by a nonlinear functions of its rotation and the propeller characteristics as shown in [Arias, 2014]. They can be described as:

$$F_i = \frac{4}{\pi^2}\rho r_{prop}^4 \omega_i^2 C_{T_i}, \tag{2.10}$$

and

$$M_i = (-1)^{i+1}\frac{4}{\pi^3}\rho r_{prop}^5 \omega_i^2 C_{P_i}, \tag{2.11}$$

where ρ is the air density, r_{prop} is the propeller radius, C_{P_i} is the propulsion coefficient of the i-th propeller and C_{T_i} is the thrust coefficient of the i-th propeller. The coefficients C_{P_i} and C_{T_i} were experimentally estimated as shown in [Arias, 2014].

The rotation of the propellers are given by the dynamics of a brushless motor as follows:

$$\begin{bmatrix} \dot{I}_i \\ \dot{\omega}_i \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau_e R_m}\left(\delta_i V_{max} - R_m I_i - k_e \omega_i\right) \\ \frac{1}{J_m}\left(k_t I_i - |M_i|\right) - \frac{1}{\tau_m}\omega_i \end{bmatrix} \tag{2.12}$$

where I_i is current in the i-th thruster, ω_i is the velocity of rotation of the i-th propeller, τ_e is the electric time constant, R_m is the electric resistance, k_e is the electromotive force coefficient, J_m is the moment of inertia of the rotor, τ_m is the mechanical time constant, k_t is the torque coefficient, δ_i is the normalized voltage applied to the i-th thruster and V_{max} is the voltage of the battery.

The input signal for the thrusters is established through δ_i. This control input is a result of combining the two previously mentioned control inputs δ_0 and δ_q. The common thrust δ_0 increases all δ_i while δ_q increases front thrusters voltage (δ_1 and δ_4) and decreases the back thrusters voltage (δ_2 and δ_3), as follows:

$$\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_0 \\ \delta_q \end{bmatrix}. \tag{2.13}$$

2.3.6 Aerodynamic forces

Aerodynamic forces appears as a result of the combination of relative to air velocity (airspeed) with coefficients extracted from wind tunnel tests [Cortés et al., 2003]. The linear airspeed ($\mathbf{V}_a$) is given by the difference between ground speed ($\mathbf{V}$) and wind velocity incident ($\mathbf{V}_w$) in the body, as described below:

$$\mathbf{V}_a = \mathbf{V} - \mathbf{V}_w, \tag{2.14}$$

where $\mathbf{V}_a = [u_a \; v_a \; w_a]^T$ is the airspeed and $\mathbf{V}_w = [u_w \; v_w \; w_w]^T$ is the wind speed incident in the body frame.

Another important definition is the **true airspeed** (V_t). The true airspeed corresponds to the euclidean norm of the airspeed, in other words:

$$V_t = \sqrt{u_a^2 + v_a^2 + w_a^2} \tag{2.15}$$

Along with this definition, the concept of sideslip (β) and attack (α) angles can be introduced. The angles β and α are given by the difference between the orientation of the airspeed $\mathbf{V}_a$ and the airship X-Y-Z axis as shown in Figure 2.8.

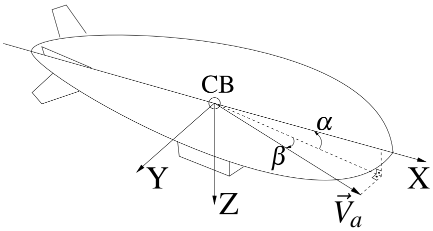


Figure 2.8. Sideslip angle (β) and angle of attack (α).

Therefore, we can define β and α by the following statements:

$$\beta = \sin^{-1}\frac{v_a}{V_t}, \tag{2.16a}$$

$$\alpha = \tan^{-1}\frac{w_a}{u_a}. \tag{2.16b}$$

Thus, the rotation matrix which transforms from the wind frame to the body frame is given by:

$$S_a = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.17}$$

Now we can define the aerodynamic forces and moments as below:

$$F_a = \begin{bmatrix} F_{ar}S_a[-C_d \;\; C_Y \;\; -C_l]^T \\ M_{ar}[-C_L \;\; C_M \;\; -C_N]^T \end{bmatrix} + D_{ar} \begin{bmatrix} [0 \;\; -\frac{C_{nr}}{l_t}r \;\; -\frac{C_{mq}}{l_t}q]^T \\ [C_{lp}p \;\; C_{mq}q \;\; C_{nr}r]^T \end{bmatrix}, \tag{2.18}$$

where: C_d, C_Y, C_l, C_L, C_M and C_N are aerodynamic coefficients of the body, which are composed by characteristics of the airship body and deflection of the tail surfaces (see [Fossen, 2013]); C_{nr}, C_{mr}, C_{lp}, and C_{mq} are damping coefficients; l_t is the tail fin momentum; and F_{ar}, M_{ar}, and D_{ar} are aerodynamic force, moment and damping references, respectively, which are nonlinear equations of true airspeed, atmospheric pressure, air density and dimensions of the airship including its tail surfaces. A more detailed description can be verified in [Cortés et al., 2003, Moutinho, 2007, Azinheira et al., 2008, Arias, 2014].

2.3.7 Kinematic equations of motion

In the simulation environment, the kinematics modeling considers quaternions for the modeling of angular positions as shown by [Azinheira et al., 2009]. Modeling in quaternions is more advantageous since it avoids the gimbal lock issue. However, for control purposes, we use Euler angles. Finally, the kinematic states (position and orientation) are updated through (2.1a), which is given by:

$$\begin{bmatrix} \dot{\mathbf{P}} \\ \dot{\Phi} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^T & 0_{3\times 3} \\ 0_{3\times 3} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \Omega \end{bmatrix}, \tag{2.19}$$

where,

$$\mathbf{R} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi / \cos\theta & \cos\phi / \cos\theta \end{bmatrix}. \tag{2.20}$$

2.4 Conclusion

In this chapter we presented insights about airship design and modeling. The aspects presented were essential for the design of an innovative four-propelled airship named Noamay. The airship nonlinear model was presented, based on both dynamics and kinematics analysis. The model introduced considers all forces that act upon it, namely, aerodynamics, gravity, propulsion, kinematics and wind.

The airship is controlled by the action of four vectored propellers and control surfaces. It was seen, however, that these actuators authority or influence is not constant. In fact, it varies as function of the airspeed. This indicates that the action of the different actuators shall depend not only on the goal mission which may include, for instance, groundspeed tracking, but also on the wind disturbances present since they have influence in the resulting airspeed. All this information and knowledge is essential to the next chapters, where the control of the airship will be addressed.

3 INCREMENTAL CONTROLLERS

Researchers studied Non-linear Dynamics Inversion (NDI) as a promising approach to unify the control scheme of an Unmanned Aerial Vehicles (UAV) during the different phases of a standard flight plan [Lane and Stengel, 1988, Benallegue et al., 2006, Voos, 2009].

To cancel model nonlinearites, NDI controllers require a complete and precise model of the system. However, in real-world systems, realistic dynamic models with accurate parameters are almost impossible to be obtained. Firstly presented in [Smith, 1998], the ID was used for designing a simplified version of Nonlinear Dynamic Inversion. Later this strategy was named as Incremental Nonlinear Dynamic Inversion (INDI) [Sieberling et al., 2010]. Since then, several works use Incremental Dynamics (ID) for designing nonlinear control laws. As some examples, in [Acquatella et al., 2013], the authors use ID for designing Incremental Backstepping (IBKS). In [Wang et al., 2019a], ID is used for design the Incremental Sliding-Mode (ISM) control law.

3.1 Incremental dynamics

This section summarizes the mathematical tools that support the incremental dynamics general formulation presented by [Sieberling et al., 2010]. Consider a control affine nonlinear system in state space representation:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \tag{3.1a}$$

$$\mathbf{y} = h(\mathbf{x}). \tag{3.1b}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of state variables, $\mathbf{u} \in \mathbb{R}^m$ is the vector of control inputs, $\mathbf{y} \in \mathbb{R}^l$ is the output vector, and f, g, h are real analytic Lipschitz continuous functions.

The system dynamics (3.1a) can be approximated by its Taylor series expansion around $\mathbf{x} = \mathbf{x}_0$ and $\mathbf{u} = \mathbf{u}_0$:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \mathbf{B}_0(\mathbf{u} - \mathbf{u}_0) + \mathcal{O}((t - t_0)^2), \tag{3.2}$$

where $(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u})$ and $(\mathbf{x}_0, \dot{\mathbf{x}}_0, \mathbf{u}_0)$ are respectively the state, the state derivative, and input at current time t and some previous time $t_0 < t$, $\mathcal{O}((t - t_0)^2)$ includes the higher order terms of the Taylor expansion, and

$$\mathbf{A}_0 = \frac{\partial}{\partial \mathbf{x}}[f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}_0]\bigg|_{\mathbf{x}_0}, \quad \mathbf{B}_0 = g(\mathbf{x}_0), \tag{3.3}$$

are state-dependent matrices that capture the linear system dynamics relationship with the state and input variables, respectively.

Let us consider the following assumptions:

Assumption 3.1 The time interval $t_s = t - t_0$ elapsed between $\mathbf{x}_0$ and $\mathbf{x}$ is sufficiently small, such that we can assume $\mathbf{x}_0 \approx \mathbf{x}$, considering that state changes are due to integration and, therefore, slower.

Then the system dynamics (3.2) can be approximated by the so-called incremental dynamics formulation:

$$\dot{\mathbf{x}} \approx \dot{\mathbf{x}}_0 + \mathbf{B}_0(\mathbf{u} - \mathbf{u}_0), \tag{3.4}$$

yielding the current state derivative from the knowledge of its value at the previous time step and of the input increment $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}_0$.

Incremental controllers such as IBKS [Acquatella et al., 2013], ISM [Wang et al., 2019a] and INDI [Azinheira et al., 2015] are sensor-based controllers, taking advantage of the simplified dynamics (3.4), where the use of state dependent dynamics is replaced by the measurement of the previous time derivative $\dot{\mathbf{x}}_0$.

3.2 Incremental Nonlinear Dynamic Inversion

INDI is the equivalent of the well known NDI control applied to the incremental dynamics (3.4).

Let us impose a desired dynamics $\nu \in \mathbb{R}^n$. Then, the incremental dynamic inversion results in the following control law:

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{B}_0^+(\nu - \dot{\mathbf{x}}_0), \tag{3.5}$$

where $\mathbf{B}_0\mathbf{B}_0^+ = \mathbf{I}_n$ is the identity matrix of order n.

Note that if the inversion is perfect then, replacing the control law in the incremental dynamic equation gives

$$\dot{\mathbf{x}} \approx \dot{\mathbf{x}}_0 + \mathbf{B}_0\Delta\mathbf{u} = \dot{\mathbf{x}}_0 + \mathbf{B}_0\mathbf{B}_0^+(\nu - \dot{\mathbf{x}}_0) = \nu \tag{3.6}$$

which shows that:

- the system modes are decoupled, which allows the design of independent linear controllers for each of them.

- the state derivative tracks the dynamics imposed by ν;
- the previous state derivatives, and consequently their nonlinearities, are canceled (NDI is also called feedback linearization);

Taking advantage of the above, we can define ν as a pseudo-control signal, which is usually taken as a linear state feedback. Imposing a linear dynamics, the feedback gain places the closed loop poles, designing the desired response. As it is common sense in cascade control, a Time Scale Separation Principle (TSSP) must be respected, and the INDI loop must converge faster than the linear control loop. In addition, the implementation of incremental controller considers the following assumptions:

Assumption 3.2 The (3.4) system is output controllable, and any internal dynamics are intrinsically stable in closed-loop.

Assumption 3.3 States are sampled at a sufficiently high frequency when compared with system dynamics.

Assumption 3.4 Fast control action in comparison to the system modes.

Assumption 3.5 The control signals and state references are measurable, continuous and bounded. Additionally, accurate information on the state derivatives and actuator variables is available.

Assumption 3.6 The input matrix B_0 has known coefficient signals, and it is non-singular around the region of interest.

Note that *Assumption 3.3* and *Assumption 3.4* are consequence of *Assumption 3.1* presented before.

Since all assumptions of Incremental controllers are satisfied we can define the INDI control law (3.5) for the system in (3.1a). The equivalent block diagram of a sensor based INDI control approach is shown in Figure 3.1, where z^{-1} represents a delay of one sample time.

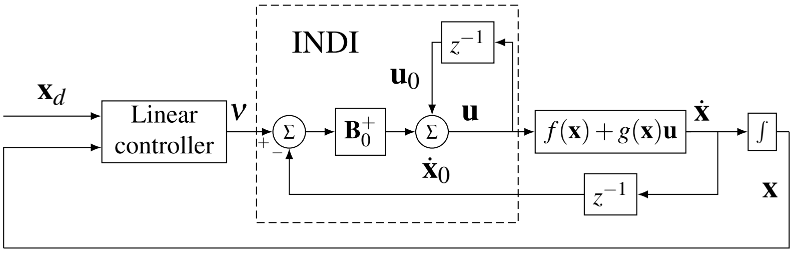


Figure 3.1. INDI closed loop block diagram

3.3 Advantages on Applying Incremental Controllers

The most common control approaches are said to be model based, for their design being strongly dependent on the system dynamics model. This aspect requires the definition of an accurate mathematical representation, as well as a careful system identification process. Nevertheless, the resulting model will still be subject to noise, disturbances and remaining model uncertainties. Moreover the inverse dynamics assumes affine relationship with inputs.

The approach based on the incremental dynamics on its hand requires only the identification of the input matrix, neglecting the parameters that depend exclusively on the internal states and it is expected to be robust to model uncertainties. Furthermore, the incremental formulation is very simple and intuitive, supported on well established mathematical fundamentals, with only a few parameters to set-up.

On the other hand, restrictions on the use of incremental controllers arise out of *Assumption 3.1* to *Assumption 3.6*. Initially, as a consequence of the model simplifications, all the needed information about the system states is obtained from measurements. Thus, to ensure quality in the measurements, the feedback states and state derivatives are to be updated at a sufficiently high sampling rate, with good quality sensors. Furthermore, *Assumption 3.4* implies that input control signal must have dominance on the system dynamics, demanding fast actuation when compared with system modes. These assumptions are satisfied in the most of UAVs since the control actuation has the greater influence over the dynamics.

3.4 Mitigating Incremental Controllers Design Issues

As stated in the previous section, despite its clear advantages, INDI controllers assumptions impose the measurement, at each sample time, of the states derivatives and a fast actuation to ensure the dominance of the control inputs over the system states. Solutions to mitigate the impact of these two restrictive aspects of INDI are presented in the sequence while preserving the afore mentioned advantages of the approach, namely its robustness to model uncertainty and design simplicity.

3.4.1 Actuator Saturation

Actuator saturation is a relevant design issue since the conception of control systems. This problem is a focus of study in many works from several different areas of control systems. In the context of airships, some works address this issue for the control design such as in [Zheng et al., 2016] and [Zheng and Xie, 2017].

In fact, actuator saturation is an issue of concern for incremental controllers. When the input saturation is achieved, the integral terms accumulates a significant error during the rise causing an effect known as integral windup.

The solution for this problem is well known in the literature as anti-windup. It is a solution addressed to controllers with integrative components. There are plenty of works presenting such solution as well as its advantages, drawbacks and analysis under measurement noise such as in [da Silva et al., 2018]. Thus, it can be considered as a standard solution.

In this work, a Natural Anti-Windup (NAW) strategy was introduced. Consider that: $\bar{\mathbf{u}}$ is saturated input; $\mathbf{u}_{max}$ is the upper limit of the actuator; and $\mathbf{u}_{min}$ is the lower limit of the actuator. After compute the new input $\mathbf{u}$ by the INDI control law (3.5), the input to be applied is corrected as shown below:

$$\bar{\mathbf{u}} = \begin{cases} \text{if } \mathbf{u} > \mathbf{u}_{max} & \text{then } \mathbf{u}_{max} \\ \text{if } \mathbf{u} < \mathbf{u}_{min} & \text{then } \mathbf{u}_{min} \\ \text{if } \mathbf{u}_{max} \geq \mathbf{u} \geq \mathbf{u}_{min} & \text{then } \mathbf{u} \end{cases} \tag{3.7}$$

Note that $\bar{\mathbf{u}}$ will always respect the actuator saturations. As a result, the controller will stop the incrementing behavior when the saturation is achieved. Consequently, it recovers faster after achieving actuator saturation. In further block diagrams this component is represented as a saturation node as shown in Figure 3.2.

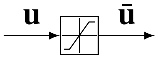


Figure 3.2. Saturation node

Numerical Example

As an example consider the following linear system with one input and one state:

$$\dot{x} = x + 8u_1 \tag{3.8}$$

where x is the state, $-0.5 \leq u_1 \leq 0.5$ is input.

By applying INDI, we obtain the following control law:

$$u_1 = \frac{1}{b_0}(\nu - \dot{x}_0) \tag{3.9}$$

where $\nu = k(x_d - x)$ is a desired dynamics of first order and $b_0 = 8$. The block diagram from Figure 3.3 illustrates the closed loop system, where z^{-1} represents a delay of one sample time $t_s = 0.01$ seconds, u_1 is the input signal generated by the INDI control law, $\bar{u}_1$ is the saturated control signal and $k = 20s^{-1}$.

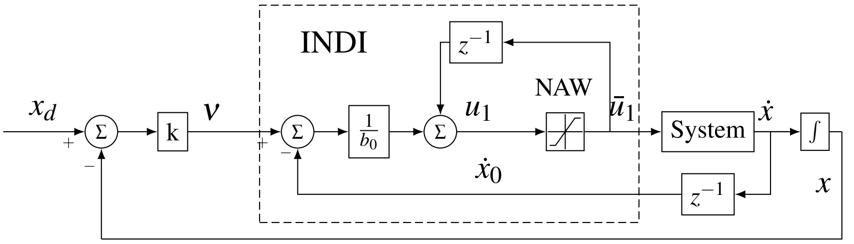


Figure 3.3. Numerical Example: Block diagram of the closed loop system of INDI+NAW

Figures 3.4 and 3.5 show the results for a step in x_d using NAW denominated case (a) (or “INDI+NAW”) and without NAW denominated case (b) (or “INDI”). For sake of comparison consider the following cost function:

$$J_u = \int_0^{t_f} u_1^2(t)dt \tag{3.10}$$

where $t_f = 10$ seconds is the final time of simulation.

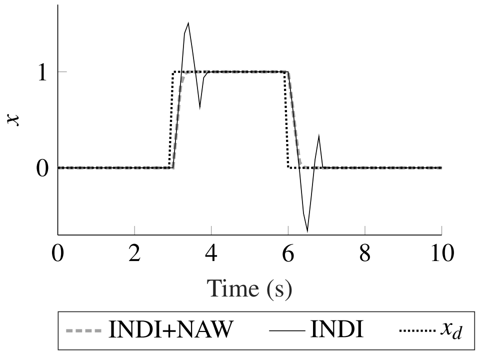


Figure 3.4. Numerical example: comparison of tracking performance between INDI and INDI+NAW

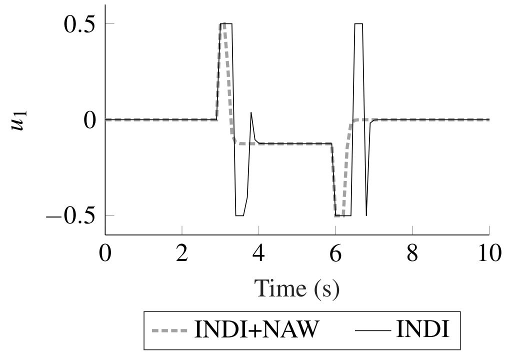


Figure 3.5. Numerical example: comparison of input signals between INDI and INDI+NAW

The tracking performance is similar for both cases, however the control signals are different with higher energy consumption for the case (b) with $J_u = 4.4876$, while case (a) spent $J_u = 1.7466$ in total energy. Moreover, note that the tracking performance is better in case (a) due to the fact that the input signal recovers faster from a saturated condition in instants $t = 3s$ and $t = 6s$ which impacts in the tracking performance by avoiding overshoots.

3.4.2 Input Redundancy Treatment

Once the input matrix $\mathbf{B}_0$ from (3.5) can be not square but full row rank, thus input redundancy may occur. Therefore, there are various solutions of inputs $\mathbf{u}_e$ that achieve a given equilibrium point $\mathbf{x}_e$. A common issue is that the redundant inputs can cancel each other when the system achieves a stationary state condition which results in more energy consumption.

As an example consider a simplified vehicle dynamics with a single state variable x given by the longitudinal velocity and two inputs u_1 and u_2. This system dynamics can be represented by the following mathematical modeling:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2. \tag{3.11}$$

Both control inputs u_1 and u_2 have influence over the vehicle velocity dynamics. Now consider an equilibrium point $x_e > 0$, thus $\dot{x}_e = 0$ and the following can be stated:

$$\begin{bmatrix} b_1 & -b_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -f(x_e), \tag{3.12}$$

where $b_1 = g_1(x_e)$ and $b_2 = -g_2(x_e)$ are positive constants.

In the problem above, there are many choices for u_1 and u_2 that achieve a given constant longitudinal velocity x_e. However, this choice will impact in the energy consumption. The ideal solution is to use the minimal control effort in order to save energy.

One solution for this problem is to perform a filtering in the commanded redundant input which is less important for maintaining in the equilibrium point. As a result we obtain a system dynamics described in the following form:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)\hat{u}_2, \tag{3.13a}$$

$$\dot{\hat{u}}_2 = \frac{-1}{\tau}\hat{u}_2 + u_2. \tag{3.13b}$$

where $\hat{u}_2$ is the filtered commanded input for the brake. By imposing this dynamics, u_2 will naturally converge to zero and u_1 will also reduce, once u_2 is no longer canceling it. Note that u_2 is still useful for the transient state condition when the vehicle needs to slowdown fast. This strategy is commonly referenced in the literature as Washout Filter (WOF). This control tool was a focus of study of [Hassouneh et al., 2004] showing its benefits and drawbacks.

The solution can be extended for systems with Multiple Inputs and Multiple Outputs (MIMO). Consider the following generalized MIMO system dynamics:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u}, \tag{3.14}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$ and $m \geq n$.

Thus, applying the WOF to the redundant inputs we obtain the following extended dynamics:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\hat{\mathbf{u}}}_s \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & g_s(\mathbf{x}) \\ \mathbf{0}_{l\times n} & \mathbf{T}_s \end{bmatrix} \begin{bmatrix} f(\mathbf{x}) \\ \hat{\mathbf{u}}_s \end{bmatrix} + \begin{bmatrix} g_m(\mathbf{x}) & \mathbf{0}_{n\times l} \\ \mathbf{0}_{l\times n} & \mathbf{I}_l \end{bmatrix} \begin{bmatrix} \mathbf{u}_m \\ \mathbf{u}_s \end{bmatrix} \tag{3.15}$$

where: $l = m - n$, $\mathbf{x} \in \mathbb{R}^n$ is the vector of states; $\mathbf{u}_m \in \mathbb{R}^n$ is the vector of main actuators; $\mathbf{u}_s \in \mathbb{R}^l$ is the vector of secondary (or redundant) actuators; $\hat{\mathbf{u}}_s \in \mathbb{R}^l$ is the vector of filtered input signals, $f(\mathbf{x})$ is the function of state dynamics; $g_m(\mathbf{x})$ is a function which describes the influence of $\mathbf{u}_m$ in the state dynamics; $g_s(\mathbf{x})$ is a function which describes the influence of $\mathbf{u}_s$ in the state dynamics; $\mathbf{T}_r = \mathrm{diag}([-1/\tau_1; -1/\tau_2; -1/\tau_3; \cdots; -1/\tau_l])$ is a diagonal matrix with positive constants $\tau_1, \cdots, \tau_l$ chosen by the designer; $\mathbf{I}_n$ is the identity matrix of order n; and $\mathbf{0}_{n\times l}$ is a matrix full of zeros with n lines and l columns.

The vector of redundant actuators can be chosen by analyzing the input function $g(\mathbf{x})$ through a systematic procedure. First, the designer must linearize $g(\mathbf{x})$ in a chosen point $\mathbf{x}_0$, obtaining the following:

$$\mathbf{B}_0 = \frac{\partial}{\partial \mathbf{x}} g(\mathbf{x})\bigg|_{x_0}. \tag{3.16}$$

By analyzing the matrix $\mathbf{B}_0 \in \mathbb{R}^{n\times m}$, the designer must identify the linearly dependent columns, which indicates the redundant inputs. After identifying the redundant actuators, the designer must choose and separate between main ($\mathbf{u}_m$) and secondary ($\mathbf{u}_s$) actuators, by also defining the functions $g_m(\mathbf{x})$ and $g_s(\mathbf{x})$. Then the extended dynamics (3.15) can be applied. This solution can be represented by a block diagram in a cascaded form as shown in Figure 3.6, where $\hat{\mathbf{u}} = [\mathbf{u}_m\ \hat{\mathbf{u}}_s]^T$ and $g(\mathbf{x}) = [g_m(\mathbf{x})\ g_s(\mathbf{x})]$.

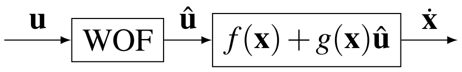


Figure 3.6. Washout Filter Block Diagram

Numerical Example

As an example consider the following linear system with two redundant inputs:

$$\dot{x} = x + 0.2u_1 - 0.8u_2 \tag{3.17}$$

where x is the state, u_1 is the first input and u_2 is the second input.

By applying INDI, we obtain the following control law:

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{B}_0^+(\nu - \dot{x}_0) \tag{3.18}$$

where $\mathbf{u} = [u_1 \;\; u_2]^T$, $\mathbf{B}_0 = [0.2 \;\; -0.8]$, $\mathbf{u}_0$ is the previous input and $\nu = k(x_d - x)$ is a desired dynamics of first order. The block diagram in Figure 3.7 illustrates the closed loop system, where z^{-1} represents a delay of one sample time $t_s = 0.01$ seconds, $\hat{\mathbf{u}} = [u_1 \;\; \hat{u}_2]^T$ is the control signal filtered by the WOF, $k = 20s^{-1}$ is a linear gain and $\tau = 5$ seconds is the time constant of the WOF.

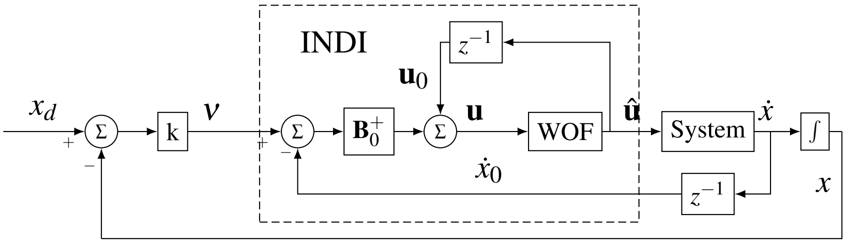


Figure 3.7. Numerical Example: Block diagram of the closed loop system of INDI+WOF

Figures 3.8 and 3.9 show the results for a step in x_d using WOF denominated case (a) (or "INDI+WOF") and without WOF denominated case (b) (or "INDI"). For sake of comparison consider the following cost function:

$$J_u = \int_0^{t_f} u_1^2(t) + u_2^2(t)dt \tag{3.19}$$

where $t_f = 15$ seconds is the final time of simulation.

The tracking performance is similar for both cases, however the control signals are different with higher energy consumption for the case (a) with $J_u = 17.82$, while case (b) spent $J_u = 12.93$ in total energy. Moreover, note that in both cases the control inputs returns to zero, thus the WOF is not advantageous for this specific system which do not contains actuator saturation.

However, vehicle dynamics commonly have actuator saturation. Thus, including the saturation limits in the system inputs given by $-0.5 \leq u_1 \leq 0.5$ and $-0.5 \leq u_2 \leq 0.5$, we obtain the simulation results shown in Figures 3.11 and 3.12. Two simulation results are depicted in these figures: (a) INDI is applied using NAW and WOF as depicted in the block diagram from Figure 3.10, where $\hat{\mathbf{u}}$ is the resultant saturated and filtered input signal; and (b) only INDI with NAW is used, as depicted in block diagram from Figure 3.3.

In Figure 3.12, the saturation is achieved in both control inputs for both cases (a) and (b). In case (b) the commanded signals u_1 and u_2 converge to values different from zero after the transient state ($t > 6$ seconds). Consequently the system has higher energy consumption in case (b) with $J_u = 17.85$. In case (a) the energy consumption is reduced in comparison to the

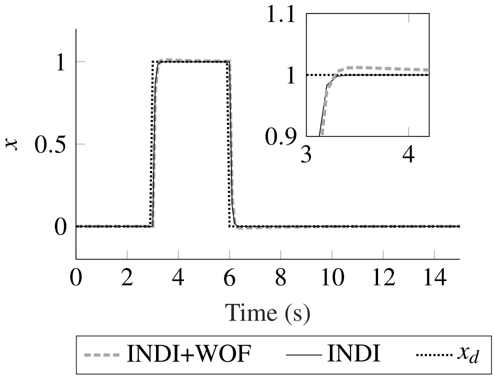


Figure 3.8. Numerical example: comparison of tracking performance between INDI+WOF and INDI

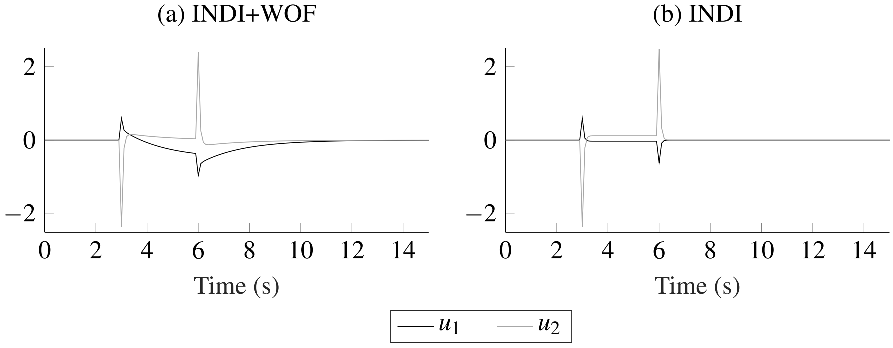


Figure 3.9. Numerical example: comparison of input signals between INDI+WOF and INDI

previous simulation, achieving $J_u = 5.74$. The tracking performance shown in Figure 3.11 is nearly the same presenting insignificant differences in comparison to the previous simulation. Therefore, the WOF appears as a simple and advantageous solution for systems with redundant actuation which have actuator saturation, such as aerial vehicles i.e. multirotor drones, aircrafts and airships.

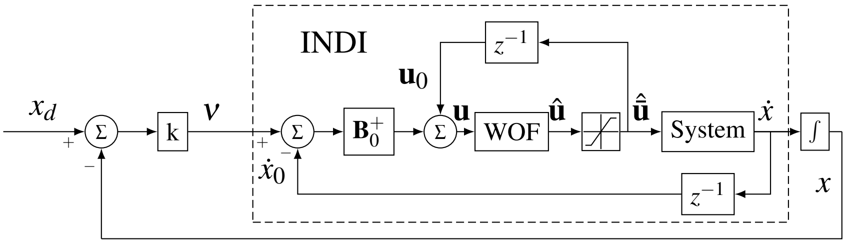


Figure 3.10. Numerical Example: Block diagram of the closed loop system of INDI+NAW+WOF

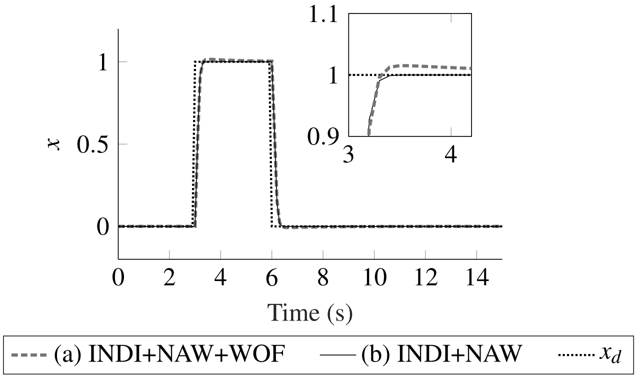


Figure 3.11. Numerical example: comparison of tracking performance between INDI+NAW+WOF and INDI+NAW

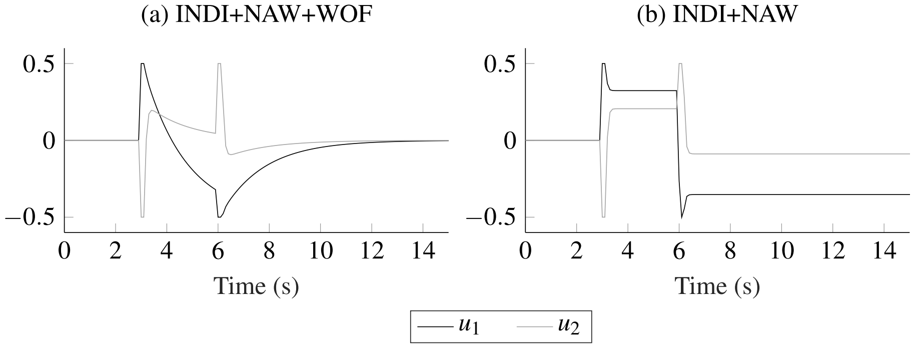


Figure 3.12. Numerical example: comparison of input signals between INDI+NAW+WOF and INDI+NAW

3.4.3 The Second-order Differentiator and the Estimation of Derivatives

Standard control techniques use state feedback for control strategies. In the past years, significant effort was applied to create sensors to measure usual states, such as positions and velocities. However, incremental controllers would require measurements of the state derivative, demanding new sensors which often did not exist yet (such as derivative of angular rates). As alternative, the state derivative may be estimated by a second order differentiator.

Assuming that the states are measured, we may derive them to obtain $\dot{\mathbf{x}}$. However, directly applying a discrete derivative is unadvised since this technique is strongly sensitive to measurement noise. To overcome this, a Second-Order Differentiator is proposed here.

Let us denote the state derivative as $\mathbf{w} = \dot{\mathbf{x}}$. The SOD is designed as a band-pass filter in which the state derivative estimation $\hat{\mathbf{w}}$ is obtained by filtering the state measurement $\mathbf{y}$ as follows:

$$\hat{\mathbf{w}}(s) = \left[\frac{\omega_n^2 s}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right]\mathbf{y}(s)\,. \tag{3.20}$$

The SOD can be seen as a combination of a typical second-order low-pass filter with unitary dc-gain, and one derivative operator. This approach has two main objectives: a) the second-order low-pass filter attenuates high-frequency noise in the measurement $\mathbf{y}$, and b) the derivative operator provides the estimated state derivative $\hat{\mathbf{w}}$ for lower frequencies.

The filter can be designed for each required state derivative: the natural frequency ω_n defines the passband mid-frequency, while the damping ratio ζ is use to set the passband size.

3.4.4 Input Scaling Gain

A disadvantage of using the SOD is the lag produced by the second-order filter, creating a delayed estimation. Thus, the controller has to be robust to delays, which is a known issue for incremental controllers [Koschorke et al., 2013, van 't Veld et al., 2018, Wang et al., 2019b]. We propose the use of the Input Scaling Gain as a solution to increase the controller robustness to delays in the feedback loop.

The ISG was first presented by [Azinheira et al., 2015] as a scalar factor. Now we generalized it to a diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$. The ISG is used to scale down the difference $(\nu - \dot{\mathbf{x}}_0)$, reducing the bandwidth of the closed-loop system. Each diagonal component λ_{ii} is in the interval $]0,\, 1]$. The modified INDI control law is given as:

$$\mathbf{u} = \mathbf{u}_0 + \mathbf{B}_0^{+}\Lambda(\nu - \dot{\mathbf{x}}_0)\,, \tag{3.21}$$

Note that with $\Lambda = \mathbf{I}_n$ we have the traditional INDI control law (3.5).

Substituting (3.21) in the closed loop Incremental Dynamics (3.4) results in:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \Lambda(\nu - \dot{\mathbf{x}}_0)\,. \tag{3.22}$$

Considering the discrete implementation, where the time interval between t_0 and t is $\Delta t = T_s$, we can rewrite (3.22) as:

$$\frac{\Delta\dot{\mathbf{x}}}{\Delta t} = \frac{\Delta\dot{\mathbf{x}}}{T_s} = \frac{1}{T_s}\Lambda(\nu - \dot{\mathbf{x}}_0)\,, \tag{3.23}$$

Assuming a fast sampling rate (small Δt), and denoting $\dot{\mathbf{x}}$ as $\mathbf{w}$, (3.23) approximates to:

$$\dot{\mathbf{w}} \simeq \frac{1}{T_s}\Lambda(\nu - \mathbf{w}_0)\,. \tag{3.24}$$

Considering $t_0 = t - T_s$, (3.24) can be rewritten as:

$$\dot{\mathbf{w}}(t) = \frac{1}{T_s}\Lambda\left[\nu(t) - \mathbf{w}(t - T_s)\right]\,. \tag{3.25}$$

Applying Laplace transform to (3.25), provides the transfer function between the state derivative and the pseudo-control as:

$$\mathscr{L}\left\{\dot{\mathbf{w}}(t)+\frac{1}{T_s}\Lambda\mathbf{w}(t-T_s)\right\}=\frac{1}{T_s}\Lambda\mathscr{L}\{\nu(t)\}\Leftrightarrow$$
$$\Leftrightarrow T_s\Lambda^{-1}s\mathbf{w}(s)+e^{-sT_s}\mathbf{w}(s)=\nu(s)\Leftrightarrow$$
$$\Leftrightarrow \mathbf{w}(s)=\left(T_s\Lambda^{-1}s+e^{-sT_s}\mathbf{I}\right)^{-1}\nu(s)=\operatorname{diag}\left\{\frac{1}{\frac{T_s}{\lambda_{ii}}s+e^{-sT_s}}\right\}\nu(s)\,. \tag{3.26}$$

Note that:

- for small sampling time, the approximation $e^{-sT_s}\simeq 1$ is accurate and (3.26) results in a first order low-pass filter: the ISG will filter high-frequency disturbances in the pseudo-control ν, specially any noise appearing in the outer-loop controller;
- as a trade-off, the ISG reduces the bandwidth of the dynamics imposed by the outer-loop controller, which may lead to a loss of performance;
- since Λ is a diagonal matrix, the ISG will define an independent filter for each component of ν where the filter applied to the component ν_i has a time constant $\frac{T_s}{\lambda_{ii}}$.

3.4.5 Combining ISG and SOD: Closed-Loop Analysis

To better understand the impact of the SOD+ISG solution in the closed-loop system, a classical continuous-time analysis is considered, providing us well-known tools to assess stability properties and delays in the system. The analysis is done using the structure illustrated by the block diagram of Fig. 3.13.

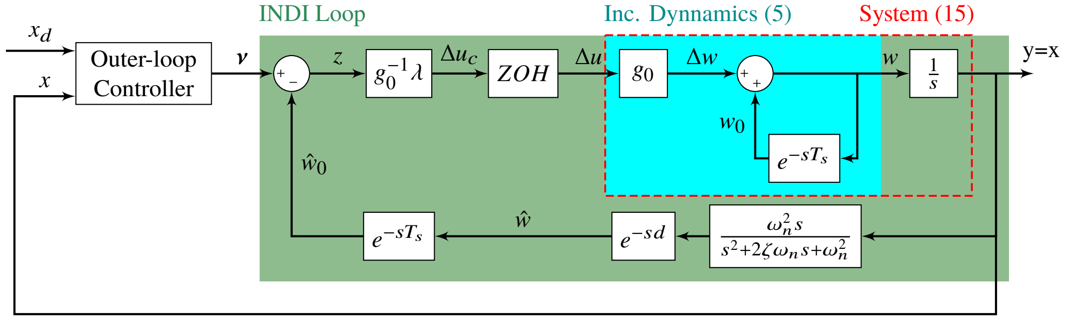


Figure 3.13. Continuous representation of the INDI controller applied to a first-order SISO system.

To analyze INDI using a continuous approach, the discrete sample delays z^{-1} (see Fig. 3.1) are replaced by time delays e^{-sT_s}. Additionally, to emulate the discrete behavior of the INDI formulation, a Zero-Order Hold (ZOH) is included in the control loop. An extra delay d is included in order to assess the effects of other delays (such as communication delays) in the INDI feedback loop.

For the sake of simplicity, the INDI controller is applied to a first-order single-input-single-output (SISO) system, but this analysis can be extended to high-order multiple-input-multiple-output (MIMO) systems since the ISG is independent for each component z_i of the INDI-loop error $\mathbf{z} = \nu - \mathbf{w}_0$, and the SOD is also independent for each state x_i.

Analytical formulation

The system to be controlled is defined as:

$$\begin{cases} \dot{x} = \dot{x}_0 + g(x_0)\Delta u \\ y = x \end{cases}, \tag{3.27}$$

where its dynamics is described using ID formulation (3.4), and its output is the state itself.

In this section we only analyze the local behavior of the system, where the control effectiveness function $g(x_0)$ is approximated by a constant g_0.

The transfer function $H_1(s)$ relating the state derivative w with the input increment Δu is obtained through the incremental dynamics formulation (3.4) as:

$$H_1(s) = \frac{w(s)}{\Delta u(s)} = g_0\left(\frac{1}{1-e^{-sT_s}}\right), \tag{3.28}$$

and, therefore, the transfer function $H_2(s)$ of the system (3.27) is:

$$H_2(s) = \frac{y(s)}{\Delta u(s)} = \frac{1}{s}H_1(s). \tag{3.29}$$

The main goal of the INDI controller, presented in result (3.6), is that the state derivative w should track a desired pseudo-control ν. Thus, the transfer function from ν to w, denoted here as $H_3(s)$, should be computed. According to the block diagram of Fig. 3.13, $H_3(s)$ is given by:

$$\begin{aligned} H_3(s) = \frac{w(s)}{\nu(s)} &= \frac{g_0^{-1}\lambda Z(s)H_1(s)}{1+g_0^{-1}\lambda Z(s)H_2(s)\frac{\omega_n^2 s}{s^2+2\zeta\omega_n s+\omega_n^2}e^{-s(T_s+d)}} \\ &= \frac{g_0^{-1}\lambda Z(s)H_1(s)\left(s^2+2\zeta\omega_n s+\omega_n^2\right)}{s^2+2\zeta\omega_n s+\omega_n^2+g_0^{-1}\lambda\omega_n^2 e^{-s(T_s+d)}Z(s)H_2(s)s}, \end{aligned} \tag{3.30}$$

where $Z(s) = \frac{1-e^{-sT_s}}{sT_s}$ is the ZOH transfer function.

Substituting (3.28), (3.29), and $Z(s)$ into (3.30) provides the transfer function of w in respect to ν as:

$$H_3(s) = \frac{\frac{\lambda}{T_s}\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)}{s^3 + 2\zeta\omega_n s^2 + \omega_n^2 s + \frac{\lambda}{T_s}\omega_n^2 e^{-s(T_s+d)}}. \tag{3.31}$$

Note that (3.31) is a linear time-invariant delayed system whose stability can be assessed using methods such as presented in [Walton and Marshall, 1987] and [Olgac and Sipahi, 2002]. However, since (3.31) is clearly a retarded dynamics system and the polynomial associated with the delay in the characteristic equation has zero order, a simpler Padé approximation provides an accurate stability analysis of the system.

Given the fast sampling required by incremental controllers, delays in the system are expected to be small. Thus, a first-order Padé approximation is proposed, in which the exponential delay terms can be approximated as:

$$e^{-sT} \approx \frac{1 - sT/2}{1 + sT/2}. \tag{3.32}$$

The Padé approximation (3.32) can be used in (3.30), providing $H_3(s)$ as:

$$H_3(s) = \frac{\frac{\lambda}{T_s}\left(1 + s\frac{(T_s+d)}{2}\right)\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)}{\frac{(T_s+d)}{2}s^4 + \left(\zeta\omega_n(T_s+d)+1\right)s^3 + \left(2\zeta\omega_n + \frac{(T_s+d)}{2}\omega_n^2\right)s^2 + \left(\frac{(2-\lambda)T_s - \lambda d}{2T_s}\omega_n^2\right)s + \frac{\lambda}{T_s}\omega_n^2}. \tag{3.33}$$

We may conclude from (3.33) that:

- the transfer function has a unitary dc-gain. Therefore, as desired by the INDI strategy, w perfectly tracks a constant pseudo control ν;

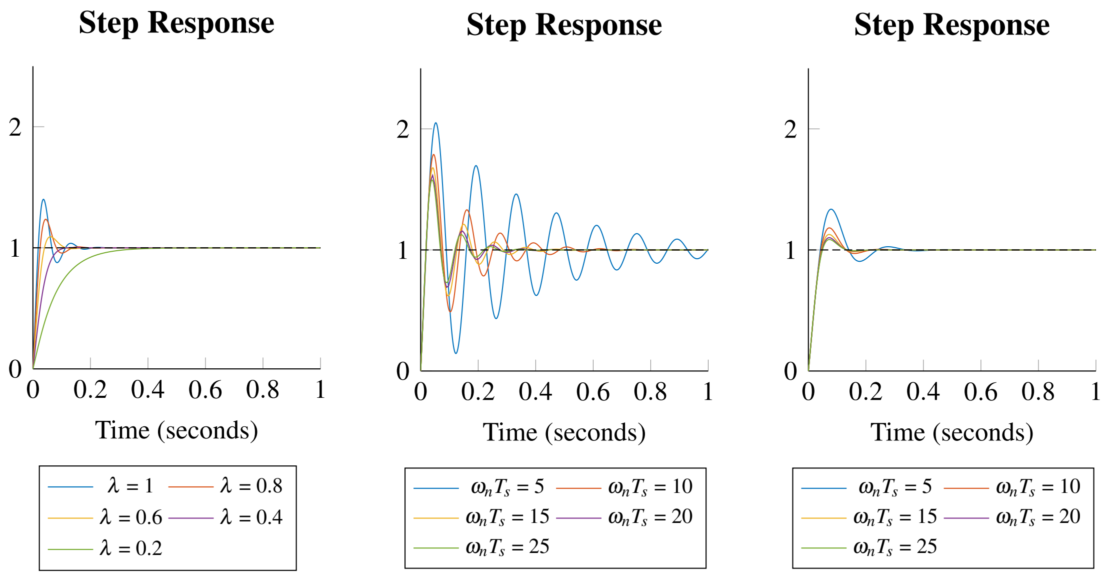
- $H_3(s)$ will only be stable if all terms of its denominator are greater than zero, which will only be achieved if $(2-\lambda)T_s - \lambda d > 0$;

- although the INDI approach requires fast sampling frequencies, reducing the sampling time also reduces the maximum allowed delay in the feedback loop;

- The ISG is not only useful to attenuate noise in the pseudo control ν, as discussed in Section 3.4.4, but it is also important to minimize the effects of additional delays in the INDI loop. By instance, without the ISG ($\lambda = 1$), the extra delay cannot exceed T_s, and an extra one-sample delay in the feedback already makes the INDI loop unstable.

Numerical analysis

A numerical example is used to analyze the impact of the ISG and the differentiator solutions in the INDI loop. A sampling frequency of 50 Hertz is assumed. In a first moment, no extra delay in the feedback loop is considered ($d = 0$). The differentiator damping ratio is set to 2. Figs. 3.14 and 3.15 show, respectively, the step response and the Bode plots of $H_3(s)$, comparing three scenarios: a) Different values of ISG with no differentiator (ideal state derivative measurement); b) different natural frequencies of the SOD without ISG; and c) different differentiators with an ISG of 0.5. Since the results emulate the discrete system, the Bode plots are presented until the Nyquist frequency ($f_s/2$).



(a) (b) (c)

Figure 3.14. Step response of the INDI loop with no extra delay in the state derivative feedback, comparing: (a) ISG values with ideal state feedback; (b) SOD frequencies without ISG; (b) SOD frequencies with ISG of 0.5.

Note from Fig. 3.14 that if no differentiator is required, the step response is similar to a second-order system. Actually, as shown in (3.26), the INDI loop is described by a first-order system with one-sample time delay, being the later responsible for the second-order behavior. The ISG can be used to shape the second-order closed-loop behavior: reducing λ will reduce the natural frequency and increase the damping ratio of the system (see Fig. 3.15a). Consequently, reducing the ISG, we do reduce the overshoot of the step response, but also increase the settling time, as shown in Fig. 3.14a.

When the SOD is required, the filter introduces an extra delay in the feedback loop, inversely proportional to its natural frequency, creating even more overshoot and oscillation in the step response (see Fig. 3.14b). Applying an ISG of 0.5, we can see in Fig. 3.14c that the overshoot and the oscillation are reduced. As a drawback, the rising time increases since the low-pass behavior slows down the system response. Note from the Bode plot in Fig. 3.15c that the ISG is able to increase the system damping, reducing resonance peaks.

Both Figs. 3.14 and 3.15 corroborate that the delay has a critical impact in the INDI loop response, and that the ISG is an interesting tool to mitigate its effect. To further investigate the

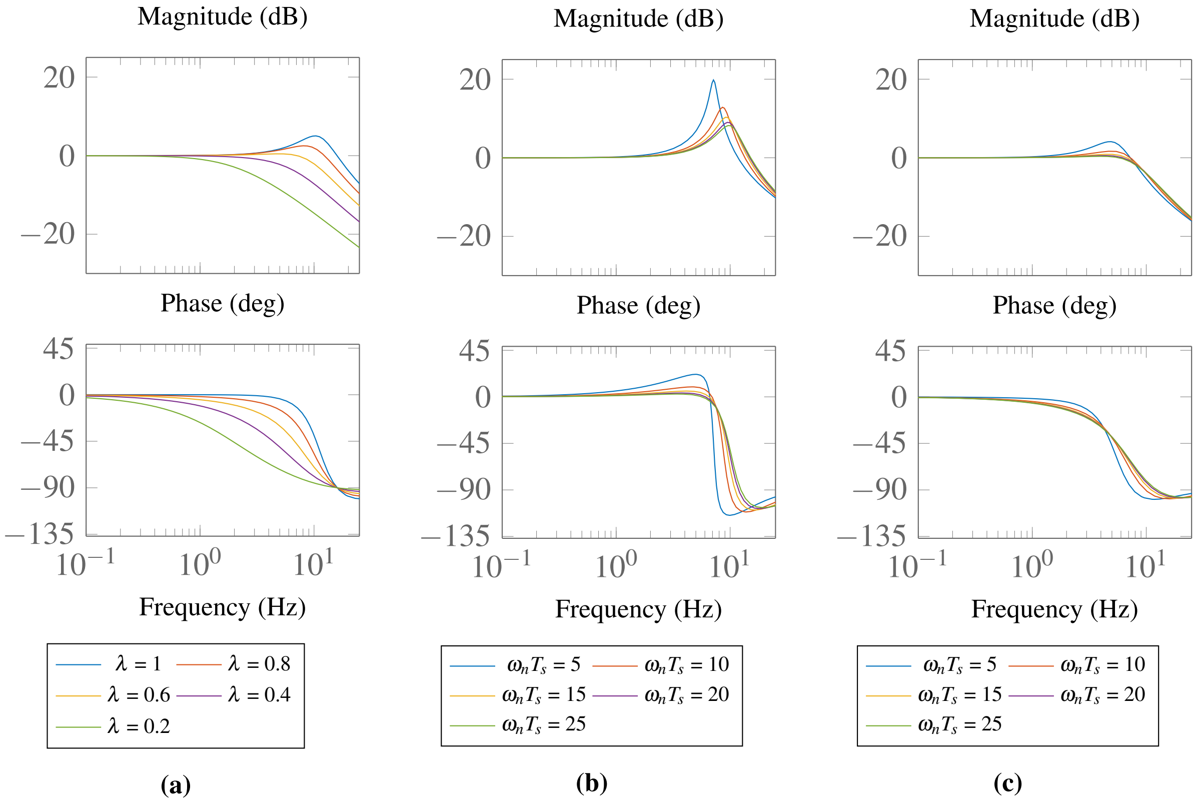


Figure 3.15. Bode plots of the INDI loop with no extra delay in the state derivative feedback, comparing: (a) ISG values with ideal state feedback; (b) SOD frequencies without ISG; (b) SOD frequencies with ISG of 0.5.

role of the delay in the INDI loop, let us analyze the scenario in which there is an extra delay in the INDI feedback loop. Figs. 3.16 and 3.17 show step results and Bode plots when d is larger than T_s. In these figures, the scenarios considered are: a) Different values of ISG considering measured state derivative; b) Different values of ISG for a differentiator with natural frequency equal to $15/T_s$; and c) differentiators with different frequencies for an ISG of 0.5.

The step response shows that, without the ISG, extra delays larger than T_s produce an unstable response, and the closed-loop system is unstable even with ideal measurement of the state derivative (see Fig. 3.16a).

When using a differentiator designed with $\omega_n T_s = 15$, the extra delay added by the filter makes the system unstable even for an ISG of 0.8, while an ISG of 0.5 is enough to stabilize the system for different frequencies of the differentiator.

From these results, the solution using the ISG and the differentiator appears to be an interesting solution for cases in which there is no measurement of the state derivatives since the ISG can mitigate the effects of the delay generated by the differentiator. Additionally, it can also be used to mitigate other delays in the measurements.

If the additional delay is known, we may evaluate the required ISG and differentiator to guarantee the local stability of the INDI controller using the Rooth-Hurwitz criteria. Fig. 3.18

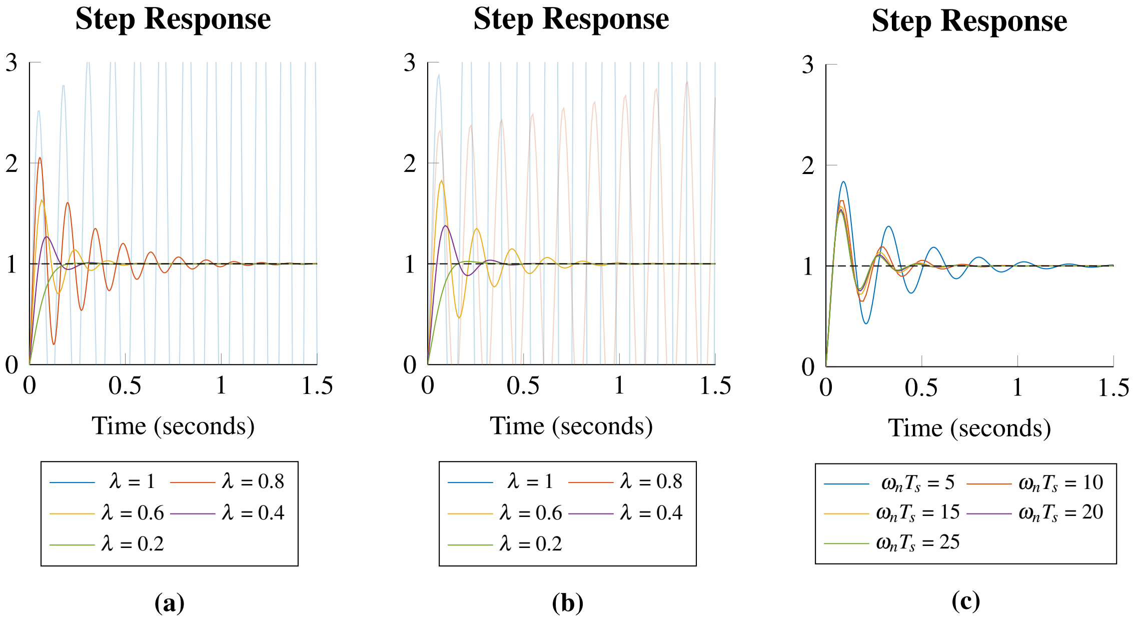


Figure 3.16. Step response of the INDI loop with $d = 1.1T_s$ (22 ms), comparing: (a) ISG values with ideal state feedback; (b) ISG values with $\omega_n T_s = 15$ in the SOD; (c) SOD frequencies with ISG of 0.5.

shows the maximum delay allowed for different values of ω_n and λ. The figure illustrates the two important contributions of this work:

1. *The ISG might be used to make the system stable when using the SOD*: without extra delay and ISG ($d = 0$, $\lambda = 1$), the system will only be stable if $\omega_n T_s$ is greater than, approximately, two. If the ISG is included ($\lambda < 1$), the minimun frequency required by the SOD can be reduced if necessary. However, low values of λ and ω_n may lead to a feeble closed-loop performance.

2. *The ISG enhances the tolerance to delays in the feedback loop*: Fig. 3.18 shows that reducing the ISG increases significantly the maximum delay allowed in the feedback loop.

Note that the maximum allowed delay and the minimum required frequency of the SOD are functions of the sampling time. Therefore, although the INDI requires fast sampling rate, the selection of T_s has to take into account the maximum absolute delay in the feedback loop, especially if a differentiator is required.

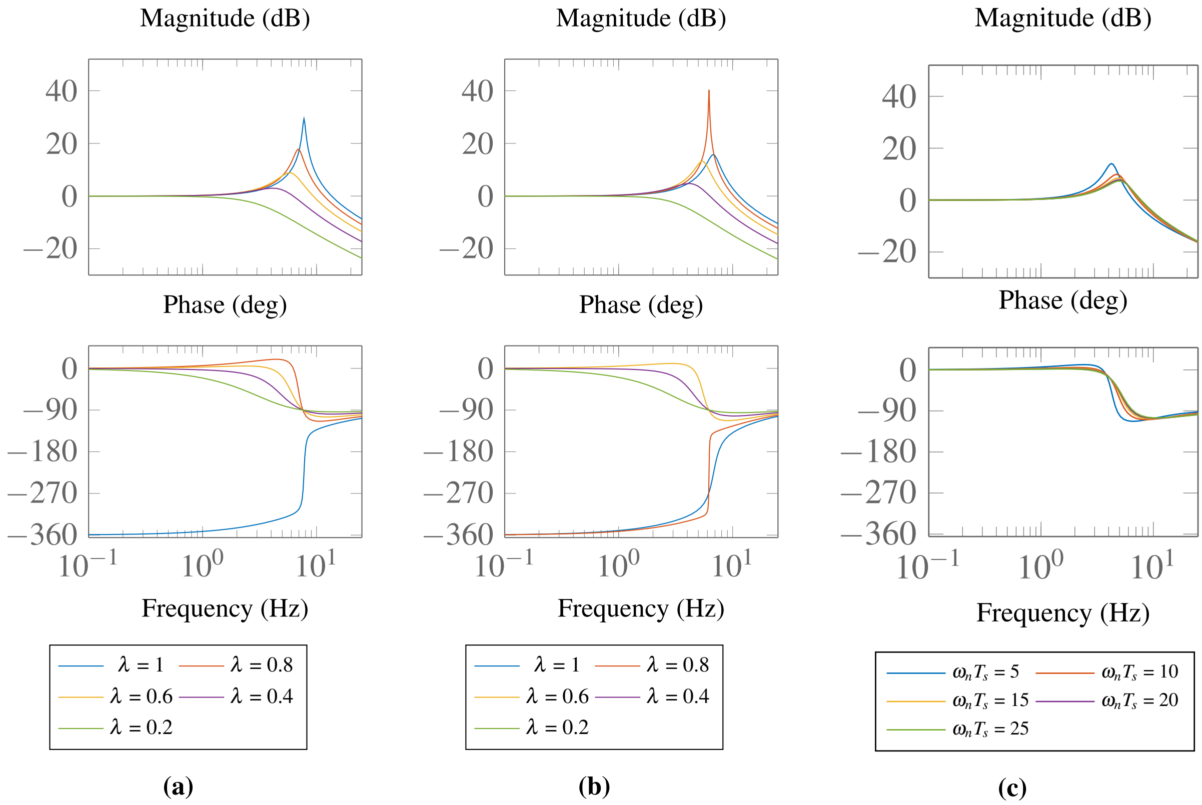


Figure 3.17. Bode plot of the INDI loop with $d = 1.1T_s$ (22 ms), comparing: (a) ISG values with ideal state feedback; (b) ISG values with $\omega_n T_s = 15$ in the SOD; (c) SOD frequencies with ISG of 0.5.

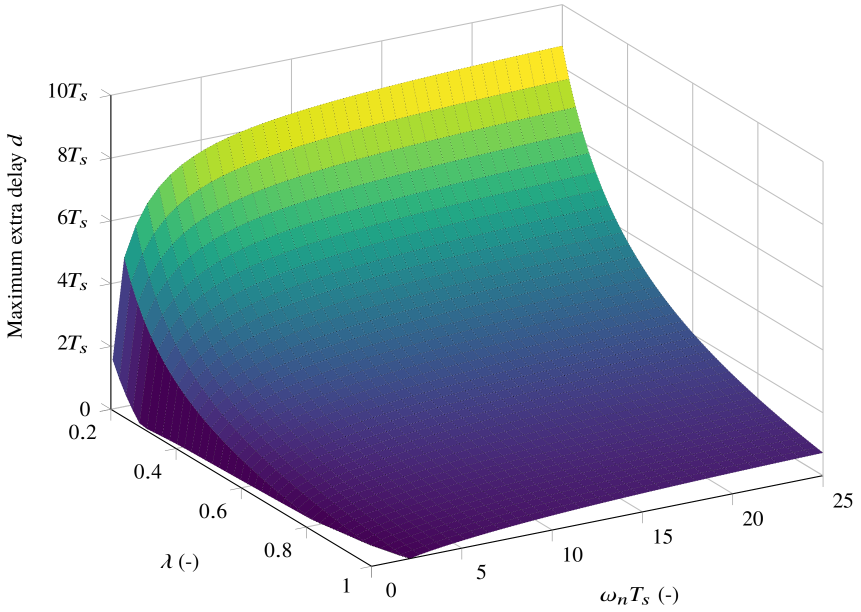


Figure 3.18. Maximum allowed extra delay as a function of the SOD+ISG parameters.

3.5 Conclusion

In this chapter, the Incremental Dynamics (ID) was presented as a promising solution for airship control. ID has a intrinsically robustness to model uncertainties which is an important characteristic, mainly because airships are nonlinear systems with many aerodynamic uncertainties. However, some design issues are highlighted, such as: (1) actuator redundancy and saturation; (2) measurement noise and delay sensitivity. In order to mitigate (1), we propose the combination of Washout Filter (WOF) and Natural Anti Wind-up (NAW). To mitigate (2), the combination of Input Scale Gain and Second-order Differentiator (SOD) is presented. Through numerical examples, and closed loop analysis, we showed that, the proposed solutions increase incremental controllers performance and robustness to measurement errors.

4 DESIGN OF INDI CONTROLLER FOR ROBOTIC AIRSHIP

In this chapter, we propose the design of INDI for the tracking of the lateral and longitudinal motions in a decoupled form for the Noamay airship.

A less common UAV platform, airships are particularly suited to scenarios that demands long endurance, high payload and low operational risk, such as surveillance and environmental monitoring [Liu et al., 2009].

The complex aerodynamics due to the voluminous envelope (helium container) and its lighter-than-air characteristics makes airship models highly uncertain, mainly because of the lateral dynamics which is difficult to be modeled in wind tunnel tests. Thus, incremental approaches appear as particularly suited for such systems since most of the aerodynamic parameters disappear in the ID formulation.

When in aerodynamic flight (forward speed, with non-vanishing airspeed), the longitudinal and lateral dynamics of the airship may be considered as decoupled. In the first section, we derive the control for the lateral motion, assuming that longitudinal and vertical states, such as forward speed and altitude, are properly regulated. Then, in the following section, we assume that the lateral motion is properly controlled to design a controller for regulating the longitudinal and vertical motions.

4.1 Lateral control

As shown by [Moutinho et al., 2016], it is usual in aeronautics to separate airship dynamics into two motions: the motion in the vertical plane (or longitudinal motion); and motion in horizontal plane (or lateral/directional motion). The airship lateral model has the state $\mathbf{x}_h = [\phi\ \psi\ \beta\ p\ r]^T$ and the input $\mathbf{u}_h = [\delta_a\ \delta_r]^T$, where: ϕ and ψ are, respectively, the roll and yaw Euler attitude angles; β is the sideslip angle; p and r are the roll and yaw rates respectively; and δ_a and δ_r are, respectively, the aileron and rudder deflection angles.

During aerodynamic flight, the aileron and rudder surfaces normally have small deflections, and the airship lateral motion can be modeled as an affine nonlinear model like (3.1a): $\dot{\mathbf{x}}_h = \mathbf{f}(\mathbf{x}_h) + \mathbf{G}(\mathbf{x}_h)\mathbf{u}_h$.

Let us separate the state $\mathbf{x}_h$ into two state vectors $\xi_h = \mathbf{C}_{\xi_h}\mathbf{x}_h = [\phi\ \psi\ \beta]^T$ and $\bar{\mathbf{x}}_h = \mathbf{C}_{x_h}\mathbf{x}_h = [p\ q]^T$, being $\mathbf{C}_{\xi_h}$ and $\mathbf{C}_{x_h}$ matrices to select the correspondent states. Thus, we can split the model dynamics into:

$$\begin{cases} \dot{\xi}_h = \mathbf{C}_{\xi_h}\dot{\mathbf{x}}_h = \mathbf{C}_{\xi_h}\mathbf{f}(\mathbf{x}_h) + \mathbf{C}_{\xi_h}\mathbf{G}(\mathbf{x}_h)\mathbf{u}_h = \mathbf{f}_{\xi_h}(\xi_h,\bar{\mathbf{x}}_h) + \mathbf{G}_{\xi_h}(\xi_h,\bar{\mathbf{x}}_h)\mathbf{u}_h \\ \dot{\bar{\mathbf{x}}}_h = \mathbf{C}_{x_h}\dot{\mathbf{x}}_h = \mathbf{C}_{x_h}\mathbf{f}(\mathbf{x}_h) + \mathbf{C}_{x_h}\mathbf{G}(\mathbf{x}_h)\mathbf{u}_h = \mathbf{f}_{x_h}(\xi_h,\bar{\mathbf{x}}_h) + \mathbf{G}_{x_h}(\xi_h,\bar{\mathbf{x}}_h)\mathbf{u}_h \end{cases} . \tag{4.1}$$

For a simpler notation, function variables will be omitted in the remainder of this section.

The dynamics of the attitude angles ϕ and ψ do not vary instantaneously with the surfaces deflection (kinematic states). Assuming that the influence of the rudder in the sideslip angle dynamics is negligible regarding the influence of the yaw rate, the control effectiveness matrix $\mathbf{G}_{\xi_h}$ is equal to zero. Then, the airship dynamics model (4.1) resumes to:

$$\begin{cases} \dot{\xi}_h = \mathbf{f}_{\xi_h} \\ \dot{\bar{\mathbf{x}}}_h = \mathbf{f}_{x_h} + \mathbf{G}_{x_h}\mathbf{u}_h \end{cases} , \tag{4.2}$$

where $\bar{\mathbf{x}}_h$ and ξ_h are defined, respectively, as the internal and external dynamics states of the system.

Since the internal dynamics are Input-to-State Stable [Khalil, 2000], the INDI strategy can be applied to control only the external dynamics of system (4.2), making the entire system stable [Wang et al., 2019b].

To apply the INDI strategy to the external dynamics of (4.2), $\mathbf{G}^{+}_{x_h}$ is required (*Assumption 3.6*). The selection matrix $\mathbf{C}_{x_h} = [\mathbf{0}_{2\times3}\ \mathbf{I}_2]$ has dimension 2×5 and the airship control effectiveness function $\mathbf{G}_h$ results in a matrix with dimension 5×2, thus, $\mathbf{G}_{x_h}$ is a 2×2 matrix. Given the dynamics properties of the vehicle, $\mathbf{G}_{x_h}$ has full rank and the $\mathbf{G}^{+}_{x_h}$ required by the INDI approach is unique and it is equal to $\mathbf{G}^{-1}_{x_h}$.

Therefore, (3.21) is used to apply the INDI strategy with the ISG+SOD and the following control law is achieved:

$$\mathbf{u}_h = \mathbf{u}_{h0} + \mathbf{G}^{-1}_{x_h}\Lambda_h(\nu_h - \hat{\mathbf{w}}_{x_{h0}}) , \tag{4.3}$$

where $\hat{\mathbf{w}}_{\mathbf{x}_h0}$ is the estimated derivative of $\bar{\mathbf{x}}_h$ obtained with the SOD. Fig. 4.1 presents the block diagram of the applied control.

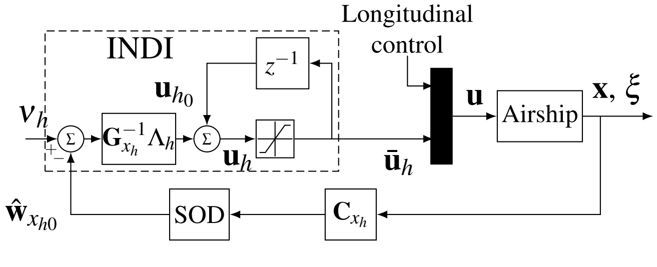


Figure 4.1. INDI strategy for tracking the lateral motion of an airship.

4.1.1 Outer-loop Controller

Once the INDI control loop acts to cancel the nonlinearities of the airship, the outer-loop can be used to impose a desired linear dynamics for the airship lateral attitude state $\chi_h = [\phi \ \psi]^T$.

The outer-loop controller must track the desired attitude χ_{h_d}. The main goal of the controller is to impose dynamics constrains for the closed-loop attitude of the airship through a pseudo-control law $\nu_h = \mathbf{w}_{x_{h_d}} = [\dot{p}_d \ \dot{r}_d]^T$.

Let us consider the desired closed-loop response as a second-order system. Therefore, the transfer functions of the lateral attitude angles of the airship in respect to their desired values are designed as:

$$\frac{\phi(s)}{\phi_d(s)} = \frac{\omega_p^2}{s^2 + 2\omega_p s + \omega_p^2}, \tag{4.4a}$$

$$\frac{\psi(s)}{\psi_d(s)} = \frac{\omega_r^2}{s^2 + 2\omega_r s + \omega_r^2}. \tag{4.4b}$$

where ω_p and ω_r are, respectively, the desired roll and yaw closed-loop bandwidths.

Considering $\dot{\phi} \approx p$ and $\dot{\psi} \approx r$, the pseudo-control can be obtained from (4.4) as:

$$\begin{aligned}
\nu_h &= -\mathbf{K}_1 \mathbf{x}_h + \mathbf{K}_2 \chi_{h_d} \\
&= -\begin{bmatrix} \omega_p^2 & 0 & 0 & 2\omega_p & 0 \\ 0 & \omega_r^2 & 0 & 0 & 2\omega_r \end{bmatrix} \mathbf{x}_h + \begin{bmatrix} \omega_p^2 & 0 \\ 0 & \omega_r^2 \end{bmatrix} \chi_d \\
&= \begin{bmatrix} -\omega_p^2(\phi - \phi_d) - 2\omega_p p \\ -\omega_r^2(\psi - \psi_d) - 2\omega_r r \end{bmatrix}
\end{aligned} \tag{4.5}$$

We can apply the pseudo-control provided in (4.5) into the INDI control law (4.3) in order to track the desired attitude χ_{h_d} respecting the dynamics constrains designed as (4.4).

4.1.2 Preliminary results

In this section, simulations in the airship nonlinear model are performed in order to evaluate the INDI loop performance in different scenarios. Also the advantages of using ISG and SOD are highlighted. Three scenarios are considered:

1. Ideal feedback or “Nominal case” – supposes that the full state vector $\mathbf{x}_h$ and its derivatives are measurable without noise or delay;

2. Noisy feedback or “Noisy case” – supposes that only velocities and attitude are measurable with noise;

3. Delayed and noisy feedback or “Real case” – supposes that only velocities and attitude are measurable with noise and delay;

In all scenarios, a “Reference” signal is applied to the yaw angle. This signal is composed by a step of 30° applied at instant $t = 3s$. In addition, the Outer-loop control gains are the same for all scenarios and given by $\omega_p = 2.3rad/s$ and $\omega_r = 2.3rad/s$.

Nominal case

The Fig. 4.2 shows results for the Nominal case. These results shows that INDI successfully tracks the yaw reference with approximately 3 seconds of rise time and without overshoot.

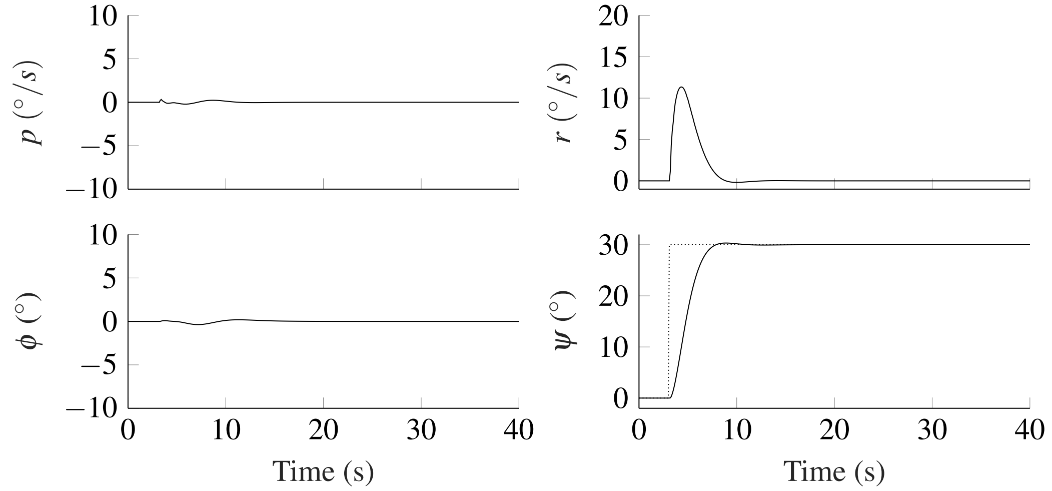


Figure 4.2. Simulations in the nonlinear model with ideal feedback using INDI.

Noisy case

Considering the Noisy case, we use SOD for feedback the INDI loop as proposed before. Also, it was introduced noise with standard deviation (σ_{RMS}[1]) given by Tab. 4.1, as specified by the datasheet of the sensor Xsens Mti-G 700. The simulation results for this scenario are shown in Fig. 4.2.

By the results in Figure 4.3, we can verify that with $\omega_n = 50$rad/s we obtain the best tracking results. With a higher natural frequency $\omega_n = 150$rad/s , the noise is not efficiently attenuated, as a result high frequency oscillations appears. However, with a lower natural frequency $\omega_n = 25$rad/s , the roll angle becomes oscillatory after the reference step. It occurs due to the introduction of measurement delay caused by SOD. In this case, ISG can be used in order to solve this specific issue.

[1] The noise power is given by $(\sigma_{RMS})^2/$(sample frequency)

Table 4.1. Sensor noise specification

State	σ_{RMS}
ϕ (rad)	$5.2\cdot 10^{-3}$
ψ (rad)	$1.75\cdot 10^{-2}$
p, r (rad/s)	$3.49\cdot 10^{-4}$

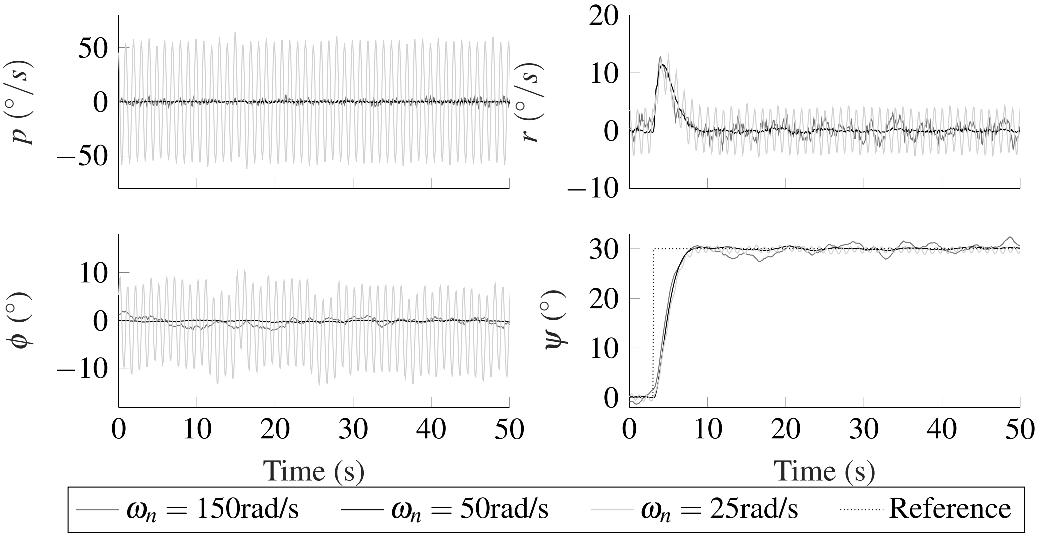


Figure 4.3. Simulation in the nonlinear model with measurement noise using INDI+SOD.

Real case

Now consider the third scenario, where it is introduced measurement delay with $T_d = 5T_s$. In this scenario, we are able to check the importance of ISG in the INDI-loop while tracking a desired attitude with noisy and delayed measurements. Simulation results can be checked in Fig. 4.4, where $\Lambda = diag([\lambda,\ \lambda])$, $\lambda = 1,\ 0.3,\ 0.1$ and $\omega_n = 100$rad/s . Note that $\lambda = 1$ is equivalent to removing ISG of the INDI-loop.

Considering $\lambda = 1$, high frequency oscillations appears in all states. However, such oscillations are attenuated as λ is decreased.

In the simulation with $\lambda = 0.1$, it is obtained low oscillation in all states and performance in an acceptable margin. Although, the high frequency oscillations in all states behavior disappear, it also increases the overshoot in ψ. Thus the designer shall be aware of the trade-off between robustness to measurement uncertainties (noise and delay) and tracking performance.

By previous simulations in Noisy and Real cases, we can note that the roll mode is the most sensitive mode to measurement errors. Thus, in order to increase its robustness a lower ISG must be set. However, since yaw and roll modes are decoupled by the Outer-loop controller, a higher ISG can be set to the yaw mode, in order to increase yaw tracking performance.

As a final simulation, different values are given for each diagonal component of the ISG matrix. The simulation results are depicted in Fig. 4.5 by setting $\Lambda = diag([0.1,\ 0.4])$. Also in this figure, the Real and Nominal cases are compared. As a result, the Real and Nominal cases

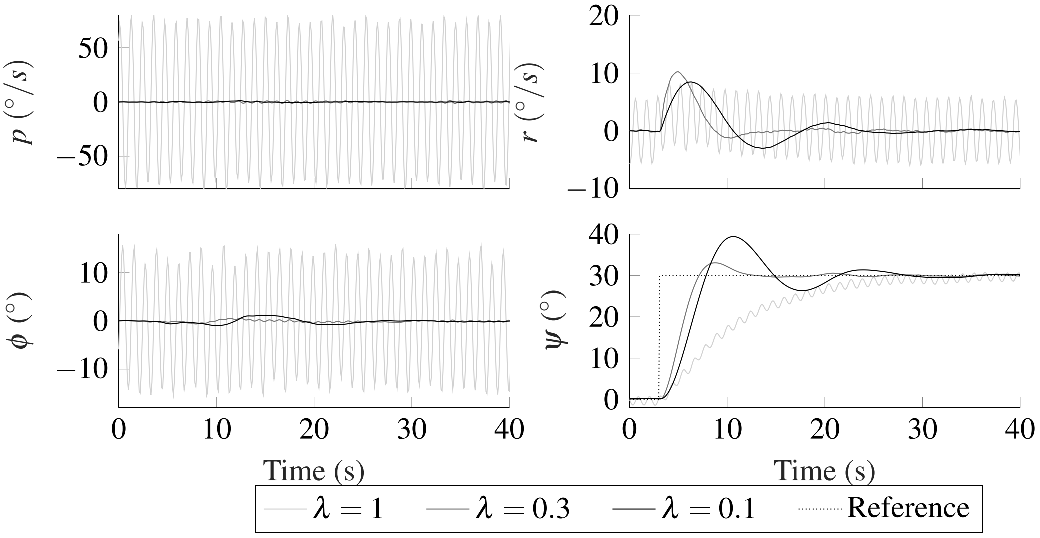


Figure 4.4. Simulation in the nonlinear model with measurement noise and delay, using INDI+SOD+ISG.

have similar performance with minimal differences due to some measurement noise. These results shows that the proposed control loop combined with SOD and ISG may stabilize the system dynamics and maintain performance even under measurement noise and uncertain model parameters.

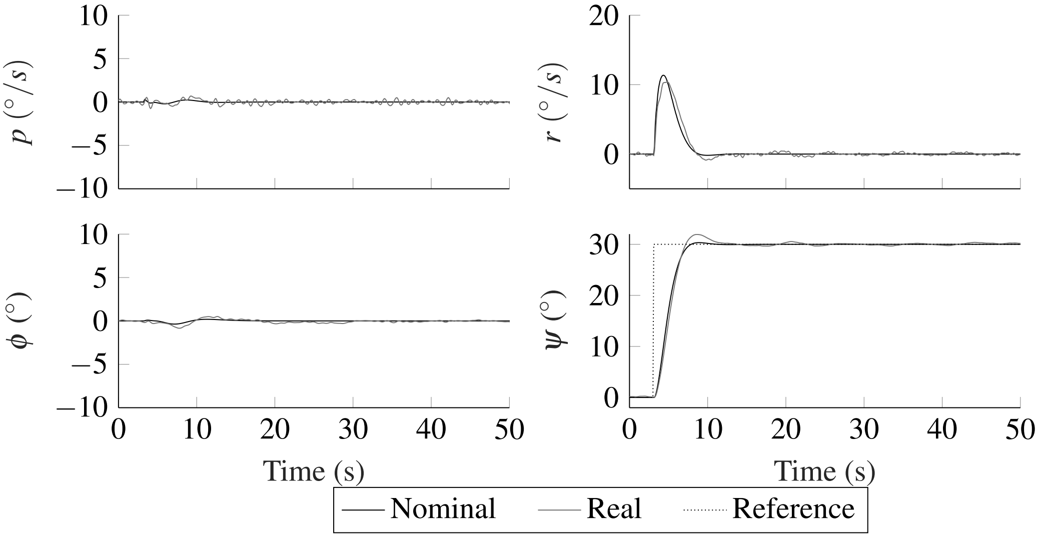


Figure 4.5. Simulation in the nonlinear model. Comparison between Nominal and Real (with measurement noise and delay using INDI+SOD+ISG) cases.

4.2 Longitudinal control

For the longitudinal model, we also decouple the longitudinal motion from the lateral motion. The states with most significant influence over the longitudinal motion (vertical plane) are given by $\mathbf{x}_v = [u\ w\ q\ \theta]^T$ and P_D. Therefore, the decoupled airship longitudinal dynamics is given by:

$$\dot{P}_D = g_v(P_D, \mathbf{x}_v, \mathbf{d}), \tag{4.6a}$$

$$\dot{\mathbf{x}}_v = f_v(\mathbf{x}_v, P_D, \mathbf{u}_v, \mathbf{d}), \tag{4.6b}$$

where $\mathbf{u}_v = [\delta_e\ \delta_0\ \delta_q\ \mu_0]$ is the input vector. Note that the vertical position P_D has influence over the dynamic states in $\mathbf{x}_v$. It occurs due to the temperature variation of the atmosphere which is a function of altitude. This will impact in the internal gas volume and pressure, as consequence certain parameters of the vehicle dynamics may change during the flight.

For a hovering objective with respect to the ground or a ground target, the wind disturbance appears both: as a positive factor, which will help to control the airship due to the increased tail authority; and a drawback, producing a mostly horizontal force that needs to be balanced by airship actuators. In such conditions, aligning the airship with wind is a common strategy to reduce the lateral drag and increase the tail surfaces authority. Although the tail surfaces authority is quite limited, their contribution is important to establish the transition between different flight phases.

When an airship is flying with low airspeed, the thrusters develop an important task by stabilizing the vertical position. In contrast, with high airspeed, they are important for controlling airship longitudinal speed, since the vertical position can be controlled efficiently by the pitch angle and elevator due to aerodynamic forces. Such characteristics shall be wisely used when designing a longitudinal controller.

In the literature, there are a few strategies which address transition between different flight phases, such as: by switching between controllers specifically designed for each flight phase [Azinheira et al., 2006]; assuming that the airship dynamic model is well known and designing a coupled controller [Moutinho et al., 2016]; or performing an optimization in control inputs also using the dynamical model [Zhu et al., 2015]. Here we design a single decoupled longitudinal controller for all flight phases. It is commanded by a higher level controller which performs a quadratic optimization in the commanded signals based in airship kinematics only.

The control architecture adopted in this work is composed by a higher level which is trajectory planning (pilot application) and three control levels: (1) a low-level controller (INDI), which generates the input signals for the actuators in order to track a given pseudo-control signal (ν_v); (2) an Outer-loop controller which tracks desired velocities and pitch angle and commands the pseudo-control signal ν_v; and (3) a Vertical Position Control (VPC) which commands a desired pitch (θ_d) and vertical velocity (w_d) based in the vertical position (P_D) and given longitudinal velocity command (u_d). This structure can be represented by the block diagram shown in Figure 4.6.

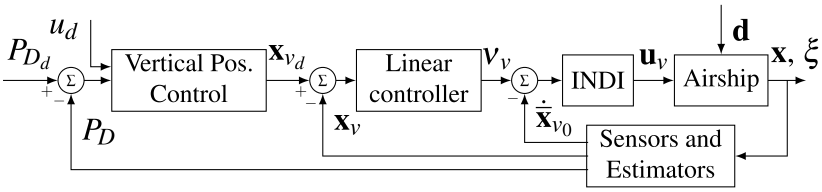


Figure 4.6. Closed loop system

4.2.1 INDI controller design

Considering that the proposed INDI control solution requires the velocity states only, lets consider the reduced state vector $\bar{\mathbf{x}}_v = [u\ w\ q]^T = \mathbf{C}_v\mathbf{x}_v$ instead of full state $\mathbf{x}_v$.

Using the Incremental dynamics approach from Chapter 3 in the dynamic equation 4.6b, the following incremental model is obtained:

$$\dot{\bar{\mathbf{x}}}_v = \dot{\bar{\mathbf{x}}}_{v_0} + \bar{\mathbf{B}}_v(\mathbf{u}_v - \mathbf{u}_{v_0}). \tag{4.7}$$

However, to design a INDI control law (3.5) the matrix $\bar{\mathbf{B}}_v$ should be inverted and here it is a 3x4 matrix: we have redundant actuation and then a control allocation must be used [Oppenheimer et al., 2006]. Normally, pseudo-inverse is used as solution for control allocation purposes. However, since the airship has actuators with different ranges and units, we use the weighted pseudo-inverse presented by [Bacon et al., 2001] as below:

$$\bar{\mathbf{B}}_v^+ = \mathbf{W}^{-1}\bar{\mathbf{B}}_v^T(\bar{\mathbf{B}}_v\mathbf{W}^{-1}\bar{\mathbf{B}}_v^T)^{-1} \tag{4.8}$$

where $\mathbf{W} \in \mathbb{R}^{m\times m}$ is a diagonal weighting matrix. This solution reduces some nonlinear behaviors of the actuators, such as saturations and rate limits. As a result, the INDI control law will be given by:

$$\mathbf{u}_v = \mathbf{u}_{v_0} + \bar{\mathbf{B}}_v^+\Lambda(\nu_v - \dot{\bar{\mathbf{x}}}_{v_0}) \tag{4.9}$$

where $\Lambda = diag([\lambda_u,\ \lambda_w,\ \lambda_q])$ is the Input Scale Gain (ISG) presented previously in Chapter 3, $\mathbf{u}_{v_0}$ is the previous input applied to the system and $\dot{\bar{\mathbf{x}}}_{v_0}$ is the vector of state derivatives which shall be measured or estimated.

This control law, requires the tuning of $\mathbf{W}$ and determination of $\bar{\mathbf{B}}_v$ to compute $\bar{\mathbf{B}}_v^+$. There is a simple procedure for tuning the weighting matrix $\mathbf{W}$. It is based in the actuators range. As an example consider the elevator control signal δ_e. This specific actuator saturates at -0.44rad and $+0.44$rad ($\pm 25^\circ$). Thus, its corresponding weight is chosen as $\frac{1}{0.88}$, since the range of actuation is 0.88rad. This procedure was followed to each control input resulting in the diagonal matrix $\mathbf{W} = \text{diag}([1/0.88;\ 1/0.9;\ 1/0.9;\ 1/2.6])$ with appropriate units.

In addition, the input matrix $\overline{\mathbf{B}}_v$ of the linearized system must be computed. In real applications, wind tunnel tests are required for identifying some input dependent coefficients. The advantage of using INDI also is seen in that procedure, since only the input dynamics is required to be identified. It reduces significantly the cost and time of parametric identification tasks.

In this work a set of input matrices is computed by a preprocessing algorithm based in the Newton-Rapson linearization for constant altitude $h = -P_D$, pitch angle θ and a given true airspeed V_t. Trim conditions were computed for fixed altitude $h = 50$m, $\theta = 0$ and varying true airspeed between $0m/s \leq V_t \leq 10m/s$ with steps of $2m/s$. Then, the matrix $\overline{\mathbf{B}}_v$ is scheduled during the flight accordingly to the measured true airspeed (see Appendix A, for the set of matrices and schedule algorithm). Although, the input matrix varies with altitude and pitch angle, INDI has sufficient tolerance for such parametric variation. In this sense, Sec. 4.2.4 presents simulation results in order to verify the parametric robustness of INDI to variations in the aerodynamic coefficients.

The resultant input for the airship is given by $\hat{\mathbf{u}}_v = [\bar{\delta}_e\ \bar{\delta}_0\ \hat{\bar{\delta}}_q\ \bar{\mu}_0]^T$ as shown in the block diagram from Figure 4.7, while the lateral control is designed as shown in [Azinheira et al., 2015].

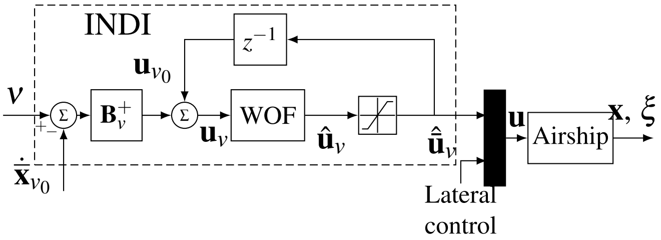


Figure 4.7. Resultant INDI loop

4.2.2 Outer-loop Control

Once the INDI control loop acts to cancel the nonlinearities of the airship, the outer-loop can be used to impose a desired linear dynamics for the airship longitudinal states $\mathbf{x}_v$. The outer-loop controller must track the desired attitude and velocities $\mathbf{x}_{v_d}$.

For this control level, any technique can be used. In our application we use a pole placement. Thus, the pseudo-control signal is composed by a regulation term plus a feed-forward term $\dot{\mathbf{x}}_{v_d}$, which corresponds to the derivatives of the reference signal $\mathbf{x}_{v_d}$. Therefore, the following linear control law can be stated:

$$\nu_v = \mathbf{K}_v(\mathbf{x}_{v_d} - \mathbf{x}_v) + \dot{\mathbf{x}}_{v_d}, \text{ where } \mathbf{K}_v = \begin{bmatrix} k_u & 0 & 0 & 0 \\ 0 & k_w & 0 & 0 \\ 0 & 0 & 2k_q & k_q^2 \end{bmatrix}, \tag{4.10}$$

with constants $k_u, k_w, k_q > 0$ chosen by the designer.

The command derivative $\dot{\mathbf{x}}_{v_d}$ can be obtained through a Second-order Differentiator (similarly to the approach shown by [Farrell et al., 2009]) or analytically based in the reference path. In further simulations it will be given through derivation of the commanded signal $\mathbf{x}_{v_d}$. The resultant Linear Control loop block diagram is shown in Figure 4.8.

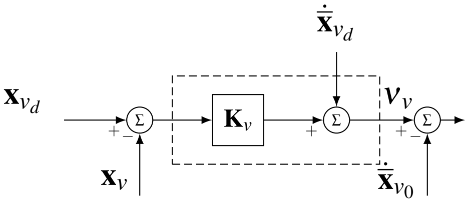


Figure 4.8. Linear Control loop

4.2.3 Vertical Position Control

The vertical Position dynamics (4.6a) can be approximated to the following kinematic equation:

$$\dot{P}_D = w\cos(\theta) - u\sin(\theta), \tag{4.11}$$

Through (4.11), it can be seen that there are many possible choices for pitch angle (θ) and vertical velocity in body-frame (w) that accomplish a vertical velocity in Earth-fixed frame ($\dot{P}_D$). In contrast, this choice will impact directly in the performance VPC loop. The aerodynamic forces can be used through the pitch regulation to control altitude and reduce the energy consumption of the actuators.

Considering that the longitudinal velocity u, vertical velocity w and pitch angle θ are well controlled by the inner loop, define a control input $\mathbf{u}_{vpc} = [w_d \ \theta_d]^T$ such that:

$$\dot{P}_D = [1 \ -u_d]\,\mathbf{u}_{vpc}. \tag{4.12}$$

Also consider the vertical position error given by $\tilde{P}_D = P_D - P_{D_d}$, thus the following control law tracks the reference signal P_{D_d}:

$$\mathbf{u}_{vpc} = [1 \;\; -u_d]^{+} \left(\dot{P}_{D_d} - k_{P_D}\tilde{P}_D\right) \tag{4.13}$$

where $^{+}$ denotes the pseudo-inverse and $k_{P_D} > 0$ is a scalar constant chosen by the designer.

The proof follows from the choice of a Lyapunov function. Consider the following Lypunov candidate $V_{vpc}(\tilde{P}_D) = \frac{1}{2}\tilde{P}_D^2$ which is positive definite. The derivative $\dot{V}_{vpc}$ is given by:

$$\dot{V}_{vpc} = \tilde{P}_D\dot{\tilde{P}}_D \tag{4.14}$$

which can be expanded to:

$$\dot{V}_{vpc} = \tilde{P}_D\big([1 \;\; -u_d]\,\mathbf{u}_{vpc} - \dot{P}_{D_d}\big) \tag{4.15}$$

Finally, by substituting (4.13) in (4.15):

$$\dot{V}_{vpc} = -k_{P_D}\tilde{P}_D^2 \tag{4.16}$$

which is clearly negative definite for all $\tilde{P}_D \in \mathbb{R}$.

The resultant control law can be considered as a quadratic optimization procedure, since the pseudo-inverse in 4.13 has more than one solution. Thus the following optimization algorithm can be defined:

$$\begin{array}{rl} \min\, _{\theta_d, w_d} & \frac{1}{2}\mathbf{u}_{vpc}^T\mathbf{W}_{rs}\mathbf{u}_{vpc} \\ \text{subject to} & \mathbf{K}_{rs}\mathbf{u}_{vpc} = V_{D_d} \end{array}$$

where $V_{D_d} = \dot{P}_{D_d} - k_{P_D}\tilde{P}_D$, $\mathbf{K}_{rs} = [1 \;\; -u_d]$ and $\mathbf{W}_{rs} \in \mathbb{R}^{m\times m}$ is a diagonal positive definite weighting matrix. The matrix $\mathbf{W}_{rs}$ will regulate the resultant value of θ_d and w_d. Normally, a higher weight shall be given for θ_d in comparison to w_d due to scale differences.

In addition, we need to consider that NOAMAY can not achieve an arbitrary pitch angle, thus some constraints must be added. Define θ_{max} and θ_{min} as the upper and lower limits for commanded pitch angle, respectively, thus the following quadratic optimization can be defined:

$$\begin{array}{rl} \min\, _{\theta_d, w_d} & \frac{1}{2}\mathbf{u}_{vpc}^T\mathbf{W}_{rs}\mathbf{u}_{vpc} \\ \text{subject to} & \mathbf{K}_{rs}\mathbf{u}_{vpc} = V_{D_d} \\ & \theta_d \leq \theta_{max} \\ & \theta_d \geq \theta_{min} \end{array}$$

This optimization is performed in real time along the mission to determine the inputs for the Linear control loop. Note that, the commanded pitch angle will be $\theta_d = 0$ as long $u_d = 0$. This choice is convenient since, at low longitudinal speed also the airspeed will be low, thus aerodynamic forces in vertical plane will be negligible in comparison to the resultant propulsion forces.

In this thesis, this optimization algorithm will be referenced as Reference Shaping (RS). There are some works in the literature that also use RS to solve similar optimization issues, i.e. [Ann et al., 2017]. The resulting block diagram of VPC loop is shown in Figure 4.9, where $\mathbf{x}_{v_d} = [u_d \; w_d \; q_d \; \theta_d]^T$.

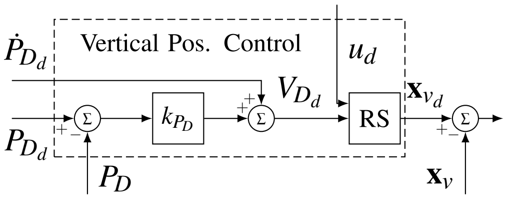


Figure 4.9. Vertical Position Control loop

4.2.4 Preliminary Results

This section will verify whether the proposed airship controller is tolerant to uncertainties in the aerodynamic parameters of the model. We want to evaluate the performance and robustness of the proposed control loop in the three flight phases, namely: VTOL, hovering and cruise flight. Firstly, we evaluate the overall performance in a nominal condition, then we introduce uncertainties in the aerodynamic parameters in order to establish a comparison with the nominal behavior.

The simulations performed are composed by a single scenario with three predefined flight phases: (1) vertical take-off, (2) cruise flight and (3) vertical landing. Also, hovering flight is performed during the transition of these phases. A description of the flight phases is given below:

1. Vertical take-off:

   The airship starts at the altitude of 5 meters from ground, then a slope rate of $\ddot{P}_{D_d} = -0.1m/s^2$ in the desired vertical speed $\dot{P}_{D_d}$ is given until achieves $-1m/s$, meanwhile the longitudinal desired speed stays as $u_d = 0m/s$. As a result, the airship performs a vertical take-off during 20 seconds, then $\dot{P}_{D_d}$ returns to zero after another slope rate with the opposite signal.

2. Cruise flight:

   For the second phase, references of vertical and longitudinal velocities are given to the RS module. The desired speeds are signals with slope rate of $\dot{u}_d = 0.5m/s^2$ and $\ddot{P}_{D_d} =$

$-0.1m/s^2$, until they achieve $\dot{P}_{D_d} = -0.5m/s$ and $u_d = 5m/s$. Then they remain constant during about 10 seconds, finally a slope rate with opposite signal is applied, until the commanded velocities achieves zero ($\dot{P}_{D_d} = 0m/s$ and $u_d = 0m/s$).

3. Vertical landing:

   For this phase, a vertical landing is performed, setting a slope rate of $0.1m/s^2$ in the commanded vertical velocity $\dot{P}_{D_d}$ until it achieves $\dot{P}_{D_d} = 1m/s$. Then, it remains constant during about 15 seconds. Finally, a slope rate of $-0.1m/s^2$ is applied to that signal until $\dot{P}_{D_d}$ achieves zero.

Also, wind with speed of $|\mathbf{V}_w| = 3m/s$ was introduced in this scenario against the airship nose during all mission. The trajectory shown in Figure 4.10 illustrates the described scenario. The constants and gains used in further simulations are depicted in Table 4.2.

The simulation without uncertainties in the model will be referred as "Nominal". The Nominal simulation performed is available for visualization as a video at https://youtu.be/ZJNylNzZuGY. The selected parameters have major influence in the longitudinal motion. Also, most of them are highly uncertain in real applications. The weighting mass or heaviness, which is the difference between the weight and buoyancy forces, was also considered, since it affects the flight envelope. The parameters are written below:

- C_{δ_0}, C_{l_0} - drag and lift coefficient
- C_{d_i}, C_{L_α} - aerodynamic force derivative coefficient;
- C_{M_0} - pitch moment coefficient;
- C_{M_α}, $C_{M_{\beta\alpha}}$ - aerodynamic torque derivative coefficient;
- $C_{M_{\delta_e}}$, $C_{L_{\delta_e}}$ - aerodynamics input coefficient derivatives;

Table 4.2. Constants and Gains

Name	Value	Name	Value	Name	Value
k_u	1.3 s^{-1}	k_q	2.6 s^{-1}	λ_u	0.2
k_w	1.3 s^{-1}	k_{P_D}	0.1 s^{-1}	λ_w	0.2
τ_q	4.46 s	t_s	0.0625 s	λ_q	0.2

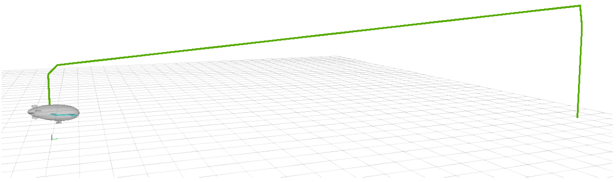


Figure 4.10. Planned trajectory

Table 4.3. Tolerance tests on model parameters (RMS values of the errors).

		$\tilde{u}$ (m/s)	$\tilde{w}$ (m/s)	$\tilde{q}$ (°/s)	$\tilde{\theta}$ (°)			$\tilde{u}$ (m/s)	$\tilde{w}$ (m/s)	$\tilde{q}$ (°/s)	$\tilde{\theta}$ (°)
Nominal		0.03	0.04	0.26	0.45	Nominal		0.03	0.04	0.26	0.45
C_{δ_0}	−80%	**0.09**	0.04	**0.63**	**1.42**	$C_{l_{\delta_e}}$	−80%	0.03	0.04	0.29	0.47
	+80%	0.03	0.04	0.28	0.41		+80%	**0.08**	0.04	**0.60**	**1.19**
C_{l_0}	−80%	0.04	0.04	0.52	0.84	$C_{M_{\delta_e}}$	−80%	**0.13**	0.05	**0.64**	**1.80**
	+80%	0.03	0.04	0.51	0.80		+80%	0.03	0.04	0.34	0.48
C_{δ_i}	−80%	0.04	0.04	0.52	0.84	$C_{M_{\beta\alpha}}$	−80%	0.04	0.04	0.51	0.82
	+80%	0.03	0.04	0.51	0.81		+80%	0.04	0.04	0.51	0.82
C_{l_α}	−80%	0.03	0.04	0.35	0.58	c_{mq}	−80%	0.02	0.04	0.38	0.39
	+80%	0.06	0.04	0.56	1.03		+80%	0.05	0.04	0.46	0.91
C_{M_0}	−80%	0.03	0.04	0.51	0.81	m_w	−80%	**0.28**	**0.07**	**0.64**	**1.65**
	+80%	0.04	0.04	0.52	0.84		+80%	**0.15**	**0.08**	0.41	1.02
C_{M_α}	−80%	0.03	0.04	0.45	0.71						
	+80%	0.04	0.04	0.59	1.00						

- c_{mq} - aerodynamic damping coefficient of the pitch rate;
- m_w - weighting mass which is the difference between the weight and buoyancy forces.

In simulations with ±50% uncertainties in the absolute value of those parameters, the airship achieved similar performance to the Nominal with negligible RMS errors. Therefore, it was increased until ±80% and the RMS values can be compared in Table 4.3. The first row of Table 4.3 shows the RMS error in the Nominal conditions of the selected variables given by: longitudinal velocity $\tilde{u} = u - u_d$, vertical velocity $\tilde{w} = w - w_d$, pitch rate $\tilde{q} = q - q_d$ and pitch angle $\tilde{\theta} = \theta - \theta_d$. In **bold** letter we can see the most significant values in comparison to the Nominal.

Through the obtained results, it is possible to verify that the most sensitive model parameters are m_w, C_{δ_0}, $C_{l_{\delta_e}}$ and $C_{M_{\delta_e}}$. The parameters m_w and C_{δ_0} have influence over the longitudinal and vertical performance. Therefore, lower values on those parameters will change the dynamics to have faster response once the damping is lower. Thus, in that case, the difference $\mathbf{x}_v - \mathbf{x}_{v_0}$ is not negligible as considered by assumption *Assumption 3.1* in Chapter 3. Meanwhile $C_{l_{\delta_e}}$ and $C_{M_{\delta_e}}$ have influence over the elevator efficiency. Thus, reducing its efficiency also goes against assumption *Assumption 3.1*, which assume that the state dynamics is slower than the input dynamics.

In addition, it was performed a simulation considering the worst case of each parameter with ±40% and ±60% inaccuracy. As shown in Figure 4.12 the vectoring angle (μ_0) and the total thrust (δ_0) achieve saturation in the transition between phases (2) and (3) when the controller is trying to reduce the longitudinal and vertical velocities at the same time. It results in a worst performance in the pitch angle tracking as depicted in Figure 4.11, since the elevator has less authority than expected.

Despite the poor identification, the proposed control strategy achieved quiet efficiently performance in terms of RMS error values. In any case, the control design approach may be considered as promising, and, among the list selected, these four parameters are in fact the model

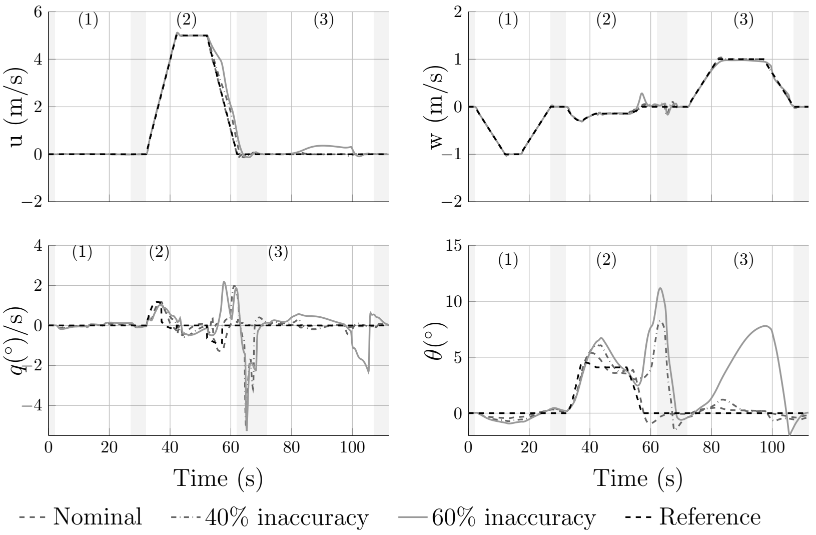


Figure 4.11. Simulation of the Airship nonlinear model with inaccuracy in the identification: commanded (Reference) and actual values for both cases (40% and %60 of model inaccuracy)

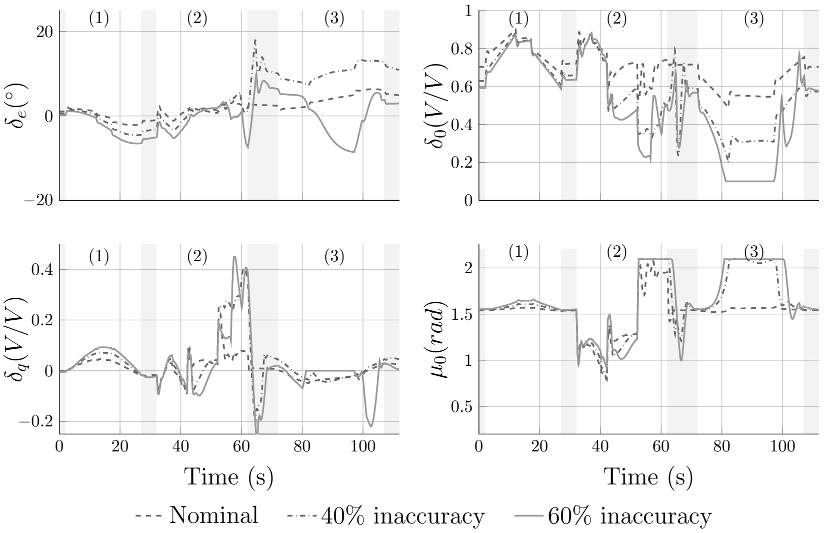


Figure 4.12. Simulation of the Airship nonlinear model with inaccuracy in the identification: control inputs for both cases (40% and %60 of model inaccuracy)

parameters for which a more careful identification or determination should take place, though the required precision should merely remain inside a say 50% margin.

4.3 Conclusion

In this Chapter, we presented how to design INDI for lateral and longitudinal motions of the airship Noamay. For the lateral motion it was used ISG and SOD as solutions for mitigating the high frequency perturbations introduced by the outer-loop and measurement errors. Also NAW was used in order to avoid over shoot in the lateral modes. For the longitudinal motion it was added WOF due to the redundancy between two elevator and rotors.

Additionally, we presented the Reference Shaping (RS) methodology in order to provide smooth transitioning between cruise and hovering flight. Through this quadratic optimization it is possible to provide a dynamical filtering of the commands such that the demanded signals are within the capabilities of the vehicle kinematics.

The proposed control loop was tested by performing VTOL and cruise flight. The results obtained demonstrates its efficiency to stabilize the system and track the desired ground velocities with the performance characteristics imposed by the commanded signals.

Robustness tests showed that the proposed control approach is insensitive to model uncertainties within the expected range found in reality. This property is relevant since the accurate identification of airship models is costly and extremely difficult. Therefore, the proposed approach may be the basis for controlling an autonomous airship. A simplified model of the platform has shown to be sufficient to compute the necessary linearized input matrix and efficiently control the vehicle.

All tests were performed in a realistic simulator however, it still have its limitations, such as the modeling of lateral drag which is very uncertain. Thus those approaches developed here should be tested and validated in experimental flights of the Noamay airships.

5 FILTERING AND ESTIMATION

Airships are known to be extremely nonlinear and under-actuated systems. Thus, automatic control, guidance and navigation are difficult tasks to accomplish. Good state estimation techniques are crucial to the development of guidance and control techniques. Initial techniques used only the yaw rate to go-to-waypoint missions as described by [Azinheira et al., 2000]. However, more sophisticated control strategies such as Sliding-modes as proposed by [Vieira et al., 2017] and Dynamic Inversion proposed by [Azinheira et al., 2015] require the knowledge of velocities and accelerations. Meanwhile, guidance requires information about attitude, position and airspeed as stated by [Moutinho et al., 2016].

The solution for this problem relies in filtering, estimation and sensor fusion methods. Several algorithms have been developed addressing this issue. A traditional algorithm is the Kalman Filter, which can be applied to linear systems. For nonlinear systems, extensions such as the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) exist.

[Kim et al., 2006] showed that such approaches are also useful for treating information redundancies that are common in sensor fusion problems. Nevertheless, [Oh, 2010] compensates sensor dynamics (such as drifting) by adding some modifications to the algorithm. In [Zanoni and de Barros, 2014], the EKF is applied for the fusion of the sample data from different sensors, and in [Zaki et al., 2019], angular acceleration is estimated through a cascaded filter structure.

The choice of an estimation technique plays a crucial role in any autonomous navigation system. Aerospace systems are critical and, usually, do not tolerate failure. As a consequence, it demands comparative studies in simulation environment before embedding the chosen technique(s) in the platform. As an example, in [Giannitrapani et al., 2011], the authors presented a performance comparison between EKF and UKF addressing the spacecraft localization problem in simulation environment.

This chapter presents a comparative study evaluating three solutions for state estimation of NOAMAY. As mentioned before in Chapter 1, NOAMAY is instrumented with a pitot tube and a Xsens Mti-G 700 which contains: accelerometer, gyroscopes, GPS, barometer and thermometer.

In the next section a summary of the sensors modeling take place considering the manufacturer specifications of each device, such as bias and sample frequency. Finally, two classical approaches are compared to a baseline approach that consists of a second-order filter in discrete time. Results obtained in simulation are presented. Then, the performance difference between the three approaches is highlighted. The algorithms were developed in C/C++ using the Robotic

Operating System (ROS) developed by [Quigley et al., 2009], aiming future tests in the actual platform.

5.1 Sensors modeling

The robotic airship is instrumented with many sensors and an infrastructure of communication [Rueda et al., 2017]. We can highlight the following sensors: a Pitot Tube, responsible for measuring airspeed (V_{pitot}); and a Xsens Mti-G 700 which is composed by a set of sensors providing the following information:

- absolute position in the world $\mathbf{P}_{NED}$ (m) measured by GPS;
- absolute orientation in the world Φ (rad), as result of a sensor fusion between Gyroscope and Magnetometer;
- inertial linear velocity ($\mathbf{V}_g$) in three-axis (m/s), estimated by the GPS;
- inertial angular velocity (Ω) in three-axis (rad/s), estimated by the IMU;
- inertial acceleration in three-axis ($a = [a_x \, a_y \, a_z]^T$) (m/s^2), measured by the Accelerometer;
- atmospheric pressure P_h (hPa), measured by the barometer;
- and temperature T_h (K), measured by the thermometer.

Each sensor has sample frequency specified as in the Table 5.2. Also, each sensor has a generic modeling of a first order Gauss Markov error as shown by [Meyer et al., 2012]. The general sensor modeling is given by:

$$y = \hat{y} + b + w_y \tag{5.1a}$$

$$\dot{b} = -\frac{1}{\tau} b + w_b \tag{5.1b}$$

where b is the sensor bias, w_b is a Gaussian noise for updating bias, w_y is a Gaussian noise for the measure as specified by the manufacturer in Table 5.1, $\hat{y}$ is the true simulated value and y is the sensor output. The bias is not present in all sensors. For the sensors in which it is modeled, the σ_{RMS}[1] is specified in Table 5.3. Accelerometer, Barometer and Thermometer require specific modeling that is evaluated in the subsections below.

[1] The noise power is given by $(\sigma_{RMS})^2/(\text{sample frequency})$

5.1.1 Pose measurement

For the case of pose measurement a coordinate transformation is applied to the data (with translational and rotational components). The GPS output is modeled as following:

$$\mathbf{P}_{gps_k} = \mathbf{S}_{xsens}(\mathbf{P}_{NED_k} + \mathbf{P}_{xsens}) + \varphi_{pos_k}, \tag{5.2}$$

where: $\mathbf{S}_{xsens}$ is the rotational matrix from the CB angular pose to the Xsens angular pose; and $\mathbf{P}_{xsens}$ is the Xsens position in the body frame; φ_{pos_k} is a position Gaussian noise vector at instant k; and $\mathbf{P}_{gps_k}$ is the sampled position given by the GPS at instant k.

The orientation output is modeled as following:

$$\Phi_{imu_k} = \Phi_{xsens} + \Phi_k + \varphi_{ori_k}, \tag{5.3}$$

where: Φ_k is the orientation given in simulation at instant k; Φ_{xsens} is the Xsens angular pose in the body frame; φ_{ori_k} is an orientation Gaussian noise vector at instant k; and Φ_{imu_k} is the sampled angular position given by the IMU at instant k.

5.1.2 Velocity measurement

For the case of velocity measurement a coordinate transformation is also applied to the data (with rotational components).

The linear velocity output is modeled as following:

$$\mathbf{V}_{gps_k} = \mathbf{S}_{xsens}(\mathbf{V}_{g_k} + \Omega_k \times \vec{O}_{xsens}) + \varphi_{vel_k}, \tag{5.4}$$

where: Ω_k is the angular velocity given in simulation at instant k; $\vec{O}_{xsens}$ is the position vector from the CB to the Xsens position; $\mathbf{S}_{xsens}$ is the rotational matrix from the CB angular pose to

Table 5.1. Sensor noise specification (w_y)

Data	σ_{RMS}	Data	σ_{RMS}
ϕ, θ (rad)	$5.2{\cdot}10^{-3}$	a (m/s^2)	$4{\cdot}10^{-3}$
ψ (rad)	0.1	P_N, P_E (m)	2.5
P_D (m)	5	V_{pitot} (m/s)	0.002
Ω (rad/s)	$3.49{\cdot}10^{-4}$	$\mathbf{V}_g$ (m/s)	0.4
P_h (hPa)	0.01	T_h (K)	1.0

Table 5.2. Sample frequency specification

Sensor	Frequency	Sampled data
IMU	100 Hz	Φ, Ω, a
GPS	4 Hz	$\mathbf{V}_g$, $\mathbf{P}_{NED}$
Barometer	50 Hz	P_h
Thermometer	1 Hz	T_h

Table 5.3. Sensor bias specification (w_b)

Sensor	σ_{RMS}
Gyroscope ($^\circ/s$)	$1.3{\cdot}10^{-4}$
Accelerometer (m/s^2)	$2.3{\cdot}10^{-5}$
Barometer (hPa)	$1.4{\cdot}10^{-5}$

the Xsens angular pose; and $\mathbf{P}_{xsens}$ is the Xsens position in the body frame; φ_{vel_k} is a position Gaussian noise vector at instant k; and $\mathbf{P}_{gps_k}$ is the sampled position given by the GPS at instant k.

The angular velocity output is modeled as following:

$$\Omega_{imu_k} = \mathbf{S}_{xsens}\Omega_k + \varphi_{rate_k}, \tag{5.5}$$

where: φ_{rate_k} is an angular velocity Gaussian noise vector at instant k; and Ω_{imu_k} is the sampled angular velocity given by the IMU at instant k.

5.1.3 Acceleration measurement

Before using the generic modeling (5.1), the accelerometer has some extra components of acceleration, namely: centripetal acceleration (a_c) and gravitational acceleration (a_g), thus resulting in the following model:

$$a = \dot{\mathbf{V}}_g + \dot{\Omega} \times \vec{O}_{xsens} - \mathbf{S}_\Phi a_g + a_c, \tag{5.6}$$

where $\vec{O}_{xsens}$ is the vector from the CB to the sensor location, S_Φ is the rotation matrix from CB to the NED frame and $a_c = \Omega \times (\Omega \times \vec{O}_{xsens})$.

5.1.4 Static pressure measurement

For simulating the static pressure at the current altitude, we use the International Standard Atmosphere (ISA) model as defined by the International Civil Aviation Organization (ICAO), as shown below:

$$P_h = P_0 \left(\frac{T_h}{T_h - 0.0065 P_D} \right)^{5.257} \tag{5.7}$$

where P_h is the static pressure at the given altitude, T_h is temperature in Kelvin, P_D is the altitude (in meters) in NED frame and P_0 is the pressure at zero altitude (at the sea level $P_0 = 101.325$ hPa). Then, the barometer is modeled with the generic sensor model (5.1).

5.1.5 Temperature measurement

The temperature in simulation is modeled with a constant rate with the altitude as follows:

$$T_h = T_0 + 0.0065 P_D \tag{5.8}$$

where T_0 is the temperature at zero altitude. Note that P_D is negative since z-axis points down, thus the temperature tends to fall as the airship goes up. Then, the thermometer is modeled with the generic sensor model (5.1).

5.2 Filtering and Estimation

In this section, we present three different strategies for filtering sensor data. The first one is a simple second-order filter that serves as a baseline for comparison purposes. This is followed by an EKF and UKF solutions. From now on we use “$\hat{.}$” notation for estimated states and “$._0$” for previous measurements.

Table 5.4. Configuration for Low-pass filter

State	Sensor	ω_f (rad/s)	ζ
$\mathbf{P}_{NED}$, $\mathbf{V}_g$	GPS	3.33	0.96
Φ, Ω	IMU	10	0.96

5.2.1 Low-pass filter

The most simple approach is a second-order Low-pass filter (LPF). It can be defined as the following transfer function:

$$\frac{\chi(s)}{Y(s)} = \frac{\omega_f^2}{s^2 + 2\omega_f \zeta s + \omega_f^2} \tag{5.9}$$

where $\chi(s)$ is the estimated state at frequency domain, $Y(s)$ is the sensor measure at frequency domain, ω_f is the cut-off frequency given by the designer and ζ is the damping coefficient given by the designer.

For each estimated state, the chosen sensor, cut-off frequency and damping coefficients are given in the Table 5.4. The cut-off frequency was chosen considering that in GPS measures we have higher errors, thus a lower bandwidth is necessary. Meanwhile, the IMU has lower errors, thus the cut-off frequency can be higher. The choice was based in several simulations until acceptable RMS error was obtained.

5.2.2 Extended Kalman Filter

The EKF algorithm is well known in literature and can be found in many works. In this thesis we use a package[2] for ROS developed by [Moore and Stouch, 2014]. This package (known as `robot_localization`) provides the EKF algorithm implementation in C/C++ for a generic robot with 6-DOF (omnidirectional). In the configuration parameters, it is possible to set the states of interest, such as position, attitude, velocity, angular rate and linear acceleration.

The filter implementation is able to receive input from different sensors. Each sensor provides information about a subset of state variables. In this work, we are interested in estimating the whole state vector $(\mathbf{P}_{NED}, \Phi, \mathbf{V}_g, \Omega, a)$. Thus, the sensors are used in measurement update stage as described in Table 5.5 , where 1 is true and 0 is false. From this table, it is possible to observe that only one redundancy is present, between the GPS and Barometer measured vertical speed (w). Also, note that Barometer does not measure altitude, it measures the static pressure. Thus, with the measured pressure and measured temperature, it was computed a mea-

[2] Available in https://wiki.ros.org/robot_localization

Table 5.5. Configuration for measurement update stage

Data\Sensor	GPS	IMU	Barometer
P_{NED}	1	0	0
Φ	0	1	0
u, v	1	0	0
w	1	0	1
Ω	0	1	0
a	0	1	0

Table 5.6. Process covariance matrix

State	Weight	State	Weight
Φ (rad)	0.01	a (m/s^2)	0.1
Ω (rad/s)	0.1	u, v (m/s)	0.05
P_N (m)	0.001	P_E (m)	0.001
P_D (m)	0.01	w (m/s)	0.001

sured altitude P_D through (5.7). Then the resulting altitude is used as input for the estimators.

Another requirement imposed by the `robot_localization` package, is to set all sensor information in the East-North-Upper (ENU) frame. However, our sensors are modeled to retrieve data in NED frame. Therefore, this data is converted from NED to ENU frame. For the specific case of acceleration it was used a cascaded scheme where the acceleration data is corrected by removing the centripetal acceleration as in (5.10) based in the previous filtered angular velocity resulting in the block diagram shown in Figure 5.1.

$$\hat{\mathbf{V}}_g = a - \hat{\Omega}_0 \times (\hat{\Omega}_0 \times \vec{O}_{xsens}) \tag{5.10}$$

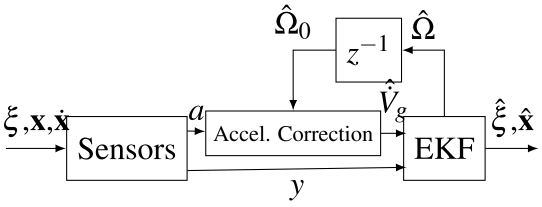


Figure 5.1. Cascaded acceleration correction

The covariance matrix of the process is a 15 × 15 matrix and it is chosen as a diagonal matrix with the weights described in Table 5.6. These values were chosen based in several simulations performed. It is known that the airship does not have fast response in the states, therefore the process covariance is presumed to be small.

5.2.3 Unscented Kalman Filter

The Unscented Kalman Filter has the same measurement update stage as in the EKF. Nevertheless, UKF addresses the approximation issues of the EKF. Therefore, in the time update stage, it is elaborated an Unscented transformation in order to calculate the statistics of a random variable which undergoes a nonlinear transformation as shown by [Wan and Merwe].

Our UKF implementation uses the same ROS package (`robot_localization`). The package also contains the UKF algorithm for a robot with 6-DOF based in the kinematic equations. Therefore, the same data conversions and configuration parameters used for EKF were applied. Other configuration and implementation details are available in a GitHub repository[3].

5.3 Estimation Results

In order to establish a comparison between the three approaches, a simulation was performed using ideal state feedback for a controller developed by [Azinheira et al., 2015]. All three estimators are working at approximately 32Hz. Also, wind flowing from the North to South with 0° of incidence is considered along the path. The simulation is performed in Simulink and then visualized in RViz as illustrated in Figure 5.3. The estimation results are available for visualization as video in `youtu.be/VL5dvCyOZwY` (LPF), `youtu.be/jaATwV0rG30` (EKF) and `youtu.be/B26xaKtAyWo` (UKF).

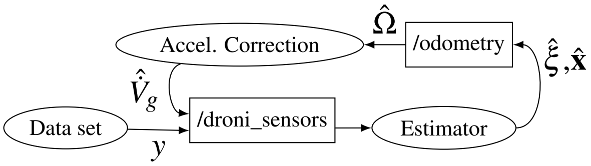


Figure 5.2. ROS communication flow chart

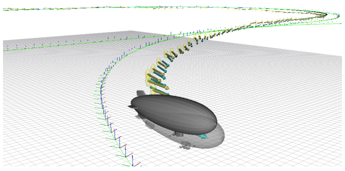


Figure 5.3. Simulation visualization example

The resulting data set (state values, derivatives and sensors outputs) is stored in a file[4] in ROS message format (`.bag`). Such data set includes sensor messages (published in the topic “`/droni_sensors`”) and the ground truth data (published in the topic “`/droni_states`”). Then the estimators are evaluated in open loop as shown in the block diagram from Figure 5.2. In this Figure, the estimator is a ROS node running independently from other estimators,

[3] https://github.com/leve-fem/airship_estimator.git
[4] Available in https://bit.ly/2VRNCUk

receiving the measured data through the ROS topic "`/droni_sensors`" and publishing the resulting estimation in the ROS topic "`/odometry`". Also, there is a node for each estimator responsible for the Acceleration correction presented previously in (5.10).

An important implementation detail is that we first regulated the covariance matrices for the UKF and then just applied the same configuration for the EKF. The estimated trajectories are shown in Figure 5.4. In order to establish a comparison, the Figure 5.6 shows two intervals of the estimated trajectory along the ground truth, obtained from the simulation (GT).

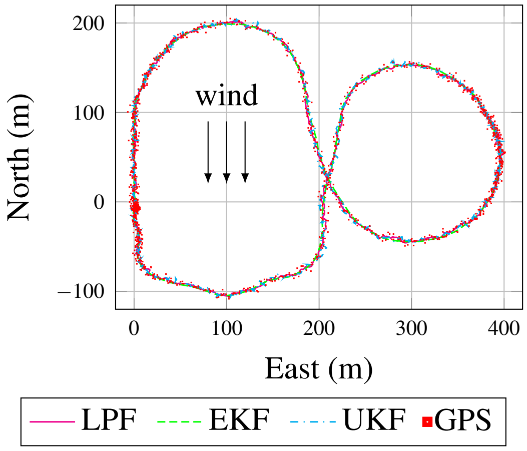


Figure 5.4. Estimated trajectory

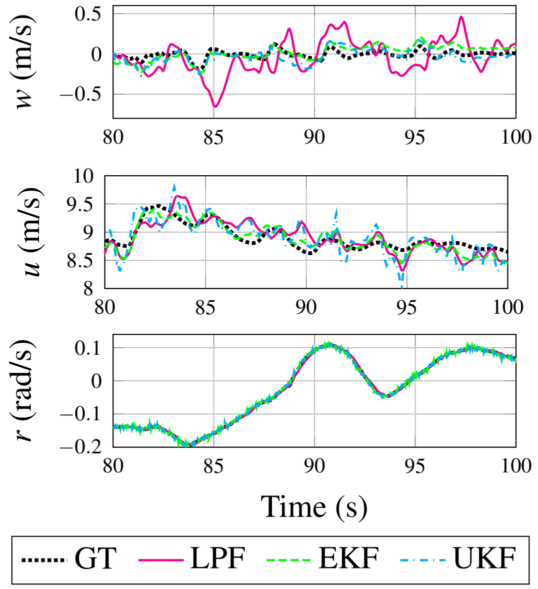


Figure 5.5. Estimated vertical velocity (w), longitudinal velocity (u) and angular velocity (r)

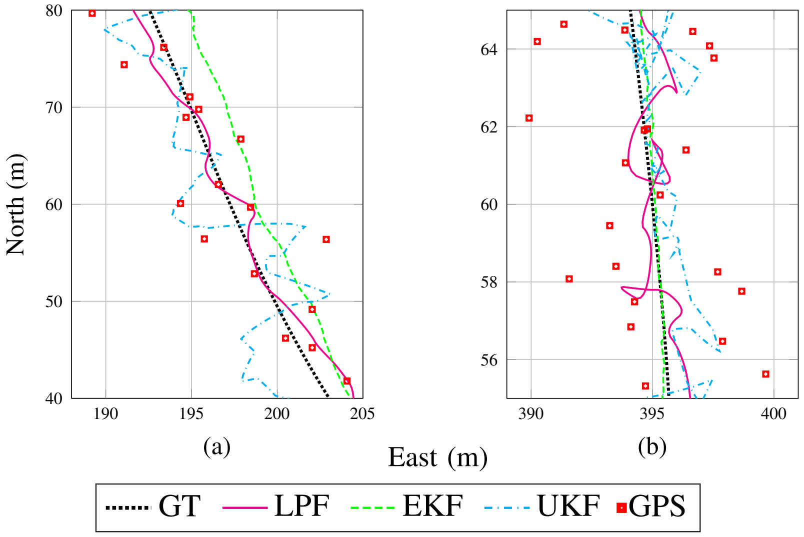


Figure 5.6. Estimated position (a) at high ground speed and (b) low ground speed

In the interval (a), EKF and LPF position estimation is smoother than the UKF, whilst the

LPF has lower absolute error. However, in the interval (b) EKF is smoother and also has a lower absolute error. Thus, EKF appears to be smoother and also has less error when considering the whole mission. It can be confirmed by the RMS of position errors shown in Table 5.7 (in **bold** the lowest RMS values). The euler angles are not shown in this table, because Xsens already provides a good estimation resulting from a sensor fusion between gyroscope and magnetometer, thus the graphical difference as the RMS error are negligible.

Table 5.7. RMS of estimation error

State	LPF	EKF	UKF
P_N (m)	3.1329	**2.4118**	3.3654
P_E (m)	3.3633	**2.6114**	3.3945
P_D (m)	3.8960	**2.0282**	2.1521
u (m/s)	0.4215	**0.3186**	0.3724
v (m/s)	0.4713	**0.2948**	0.4267
w (m/s)	0.3071	**0.2871**	0.2915
p (rad/s)	**0.0644**	0.0685	0.0664
q (rad/s)	**0.0355**	0.0359	0.0366
r (rad/s)	0.0373	0.0305	**0.0304**

The most relevant states for the control loop are longitudinal (u), vertical (w) and angular velocities (Ω). For the first interval shown before, the estimation of vertical velocity, longitudinal velocity, and angular in z-axis (r) in Figure 5.5. By these figures, we can conclude that LPF provides good filtering. However, it inserts a significant delay. On the other hand, UKF provides a faster response with a higher error. Meanwhile, EKF has a time response on par with the UKF while maintaining lower error levels. Thus, for the airship model, the EKF shows a better performance in comparison to the two other estimation approaches.

5.4 Conclusion

The presented approaches have shown an acceptable filtering and estimation of the whole state vector, with sufficient precision to facilitate the control techniques developed by the group. The EKF and UKF have shown similar estimation performance. However, the EKF presented a slightly better performance.

It is important to highlight that (as we explained before) we have chosen the covariance parameters for the UKF searching for the best results possible. Then the same parameters were applied to EKF, without any additional optimization. Consequently, we initially expected to achieve a better performance with UKF. However, from the results presented in the previous sections, one can observe that the EKF clearly had a better performance. It suggests that, for this specific platform and under the same conditions, the EKF may be a more suitable choice. These are interesting results, since in [Giannitrapani et al., 2011] the UKF presented a slightly better performance in terms of average localization accuracy for their specific case of a spacecraft model.

The algorithms were developed in C/C++ using ROS in order to embed the estimator in the platform in future works. Also, error tolerance analyses and closed-loop simulations will take place in future works.

6 WIND VELOCITY ESTIMATORS

This chapter addresses the problem of wind velocity estimation. Firstly, it is presented an alternative version of a Model-based wind velocity estimator using EKF. This technique is similar to the solution presented in [Cho et al., 2011], however taking the kinematic equations of an airship. Then, a Data-driven approach of estimation using Neural Network (NN) is proposed. Finally, a hybrid version that uses both Model-based and Data driven techniques is proposed. The main tool to validate the proposed solutions is the realistic nonlinear model in Simulink/MATLAB presented in Chapter 2.

6.1 Kinematic Equations of motion

In this section we find three equations which are correlated with the Pitot probe measurement, wind speed ($\mathbf{V}_w$), airship groundspeed ($\mathbf{V}_g$) and orientation (Φ).

Let the airship motion be represented by its inertial velocity $\mathbf{V}_g$. Similarly, the wind is described by an inertial velocity $\mathbf{V}_w$. The airship relative air velocity is called airspeed ($\mathbf{V}_a$) and it is given by:

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w, \tag{6.1}$$

where $\mathbf{V}_w = [u_w \, v_w \, w_w]^T$ and $\mathbf{V}_a = [u_a \, v_a \, w_a]^T$.

The Euclidean norm of the airspeed is called *true airspeed* (V_t) and it is given by:

$$V_t = ||\mathbf{V}_a||_2 = \sqrt{u_a^2 + v_a^2 + w_a^2}. \tag{6.2}$$

Other important definitions are the sideslip angle β and angle of attack α. The sideslip angle is a relative orientation between the vertical plane of the vehicle and the vector $\mathbf{V}_a$. Moreover, the angle of attack α is given by the angle between the vector $\mathbf{V}_a$ and the horizontal plane of the vehicle, as shown in Figure 6.1.

Threfore, we can define β and α by the following statements:

$$\beta = \sin^{-1} \frac{v_a}{V_t}, \tag{6.3}$$

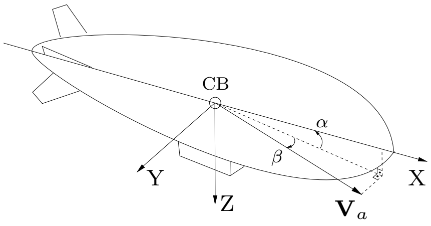


Figure 6.1. Sideslip angle (β) and angle of attack (α).

$$\alpha = \tan^{-1}\frac{w_a}{u_a}. \tag{6.4}$$

Thus, an equivalent formula is given by:

$$w_a = u_a\frac{\sin\alpha}{\cos\alpha} \quad \text{and} \quad v_a = V_t\sin\beta. \tag{6.5}$$

Finally, we obtain:

$$V_t = \frac{u_a}{\cos\alpha\cos\beta}. \tag{6.6}$$

The Pitot tube is located in the airship nose, thus it will be measuring the longitudinal dynamic pressure ΔP in the body frame. Also, it has a correlation with the airspeed as shown below:

$$\Delta P = \eta\left(u_a\right)^2, \tag{6.7}$$

where η is the calibrating factor that is correlated with the air density and pitot efficiency. Now consider the following variable transformation:

$$V_{pitot} = \sqrt{\Delta P}, \tag{6.8}$$

Thus, V_{pitot} is correlated with the true airspeed through the following statement:

$$V_t = \frac{V_{pitot}}{\sqrt{\eta}\cos\alpha\cos\beta}. \tag{6.9}$$

Because there are uncertainties in η and the angles α and β are unmeasurable with the actual available sensors, those values will be estimated together as a scale factor c_f given by:

$$c_f = \sqrt{\eta}\cos\alpha\cos\beta, \tag{6.10}$$

therefore (6.9) becomes:

$$V_t = \frac{1}{c_f}V_{pitot}. \tag{6.11}$$

Now, consider that the rotation of vector $\mathbf{V}_g$ from the body frame to the NED frame is given by $\mathbf{V}_{NED} = [V_N \; V_E \; V_D]^T$. Also, consider that such rotation applied to vector $\mathbf{V}_w$ is given by $\mathbf{V}_{NEDw}$. Thus, the airspeed in NED frame is given by:

$$\mathbf{V}_{NEDa} = \mathbf{V}_{NED} - \mathbf{V}_{NEDw} = S_{\Phi}^{T}\mathbf{V}_a. \tag{6.12}$$

It is known that a rotational operation does not change the vector modulus, thus the following statement is valid, considering the wind strictly horizontal:

$$V_{pitot}^2 = c_f^2\big((V_N - V_{N_w})^2 + (V_E - V_{E_w})^2 + (V_D)^2\big). \tag{6.13}$$

Supposing that, the airship starts from an initial condition where α and β are negligible ($\alpha \approx \beta \approx 0$) we have $u_a = V_t$ and $v_a = w_a = 0$. Therefore in the global frame we have:

$$V_{N_a} = u_a \cos\psi \cos\theta, \tag{6.14}$$

$$V_{E_a} = u_a \sin\psi \cos\theta, \tag{6.15}$$

where ψ and θ are the yaw and pitch angles, respectively. Since we have (6.1), then:

$$V_N = \frac{V_{pitot}}{c_f}\cos\psi\cos\theta + V_{N_w}, \tag{6.16}$$

$$V_E = \frac{V_{pitot}}{c_f}\sin\psi\cos\theta + V_{E_w}. \tag{6.17}$$

The values of V_{pitot}, V_N and V_E are measurable by the Pitot tube and GPS, therefore (6.13), (6.16) and (6.17) can be used as observation equations, while c_f, V_{N_w} and V_{E_w} are estimated states in the EKF. Note that, the Euler angles (ϕ, θ and ψ) can be measured by the IMU. Although these equations are linear dependent, we are introducing redundant information of different sensors by adding IMU combined with GPS measurements, which may lead to faster convergence and better filtering of noise.

6.2 Extended Kalman Filter

In this section we propose an EKF in order to estimate the incident wind in the airship body. Assuming that the wind is strictly horizontal the main goal is to obtain an estimation of the wind velocity in the horizontal plane (North-East) and the Pitot probe scale factor. One main benefit is expected from this method. The estimator presented in [Cho et al., 2011], uses only one measurement update equation which is similar to (6.13). By introducing (6.16) and (6.17) in the measurement update stage we expect to have a better performance than using only (6.13).

There is no given model to determine the wind behavior. Thus, we assume that the wind is constant with a Gaussian input with significant covariance. In addition, as long as we do not have a sideslip sensor, thus the uncertain and time varying factor c_f will be estimated, which leads us to the following reduced system model:

$$\chi_{k+1} = \mathbf{F}\chi_k + \nu_k, \tag{6.18a}$$

$$\mathbf{z}_k = \mathbf{h}(\chi_k) + \upsilon_k \tag{6.18b}$$

where: $\chi_k = [V_{N_{wk}}\ V_{E_{wk}}\ c_{fk}]^T$ is the state vector in the instant $t = kt_s$; $\mathbf{z}_k = [V^2_{pitot_k}\ V_{N_k}\ V_{E_k}]^T$ is the system output in the instant $t = kt_s$; t_s is the sample time in seconds; $\mathbf{h}(\chi_k)$ is the output function, which can be computed through (6.13), (6.16) and (6.17);

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$\nu_k \sim \mathbf{N}(0, \mathbf{Q})$ is the process noise with Gaussian distribution and covariance $\mathbf{Q}$; and, finally, $\upsilon_k \sim \mathbf{N}(0, \mathbf{R})$ is the measurement noise also with Gaussian distribution and covariance $\mathbf{R}$.

Given the model described in (6.18a), we can update the state and covariance matrix ($\mathbf{P}$) as follows:

$$\chi_{k|k-1} = \mathbf{F}\chi_{k-1}, \tag{6.19}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}. \tag{6.20}$$

For the accomplishment of the EKF final stage we obtain the Jacobian matrix of $\mathbf{h}(\chi_k)$ evaluated in the measured values of V_{pitot}, $\mathbf{V}_{NED}$, Φ and $\chi_{k|k-1}$ which is given by:

$$\mathbf{H}_k = \left[\frac{\partial \mathbf{h}(\chi)}{\partial V_{N_w}}, \frac{\partial \mathbf{h}(\chi)}{\partial V_{E_w}}, \frac{\partial \mathbf{h}(\chi)}{\partial c_f}\right]\Bigg|_{\chi_{k|k-1}},$$

where:

$$\frac{\partial \mathbf{h}(\chi)}{\partial V_{N_w}} = [-2\hat{c}_f^2 \hat{V}_{N_w}(V_N - \hat{V}_{N_w})\ 1\ 0]^T,$$

$$\frac{\partial \mathbf{h}(\chi)}{\partial V_{E_w}} = [-2\hat{c}_f^2 \hat{V}_{E_w}(V_E - \hat{V}_{E_w})\ 0\ 1]^T \text{ and}$$

$$\frac{\partial \mathbf{h}(\chi)}{\partial c_f} = \begin{bmatrix} 2\hat{c}_f\left((V_N - \hat{V}_{N_w})^2 + (V_E - \hat{V}_{E_w})^2 + (V_D)^2\right) \\ -\frac{V_{pitot}}{\hat{c}_f^2}\cos\psi\cos\theta \\ -\frac{V_{pitot}}{\hat{c}_f^2}\sin\psi\cos\theta \end{bmatrix}.$$

Finally, the standard algorithm of EKF can be applied as follows:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{h}(\chi_{k|k-1}),$$

$$\mathbf{C}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R},$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{C}_k^{-1},$$

$$\chi_k = \chi_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k,$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1},$$

where **P** is the covariance matrix, **C** is the covariance error, $\tilde{\mathbf{y}}$ is the measurement error, **K** is the Kalman gain and **I** is the identity matrix with appropriate dimensions.

6.3 Neural Network

This section proposes a Neural Network (NN) to estimate the wind speed in NED frame and the Pitot probe scale factor c_f. The main benefit expected from this method is the NN intrinsic property of creating a mapping from input parameters to the output. Because the NN is trained by measured data, it is able to detect abrupt variation in the wind velocity. However, it is also expected to be very sensitive to measurement errors.

As the analytical model (6.18a)–(6.18b) shows, the model output is composed by (6.13), (6.16) and (6.17), which are nonlinear equations in the model states, vehicle velocity, orientation and Pitot pressure. In order to avoid complex nonlinearities, the measured data is remapped into 8 inputs given by:

$$\mathbf{z}_{nn} = \begin{bmatrix} V_{pitot}^2 \\ V_D^2 \\ V_N \\ V_E \\ V_E^2 \\ V_N^2 \\ V_{pitot}\cos\psi\cos\theta \\ V_{pitot}\sin\psi\cos\theta \end{bmatrix}.$$

Meanwhile the output vector χ_{nn} of the NN contains the estimated wind velocities in the horizontal plane and the scale factor c_f as shown below:

$$\chi_{nn} = \begin{bmatrix} V_{N_w} \\ V_{E_w} \\ c_f \end{bmatrix}.$$

The NN was designed in the MATLAB Neural Network Toolbox^{TM}. It is a three-layer fitting NN, which has three nonlinear hidden layers containing 24 neurons each and three linear outputs. The activation function of the nonlinear neurons is sigmoidal. The resulting flow chart is shown in Figure 6.2.

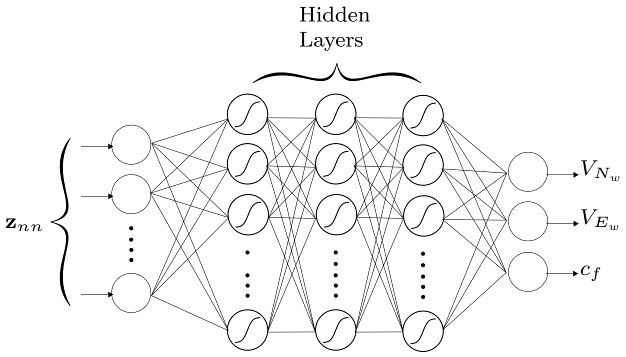


Figure 6.2. Neural Network flow chart.

The training dataset is composed by simulations in variations of the two trajectories shown in Figure 6.3. The trajectories (a) and (b) are rotated of {0, 45, 90, 135, 180, 225, 270, 315} degrees around origin generating 16 scenarios (see Figure C.1) in which the airship performs curves and straight lines in different directions. For each scenario simulations were performed with wind speed at $|\vec{V}_w| = \{0, 1, 2, 3, 4, 5\}$m/s and heading ϕ_w ={0, 22.5, 45, 67.5, 90, 112.5, 135, 157.5, 180, 202.5, 225, 247.5, 270, 292.5, 315, 337.5} degrees, where $\phi_w = \tan^{-1}\left(\frac{V_{E_W}}{V_{N_W}}\right)$. Hence, a total of 1281 simulations were performed. In all simulations the airship performs a typical cruise flight at 7m/s airspeed and constant altitude of 50 meters.

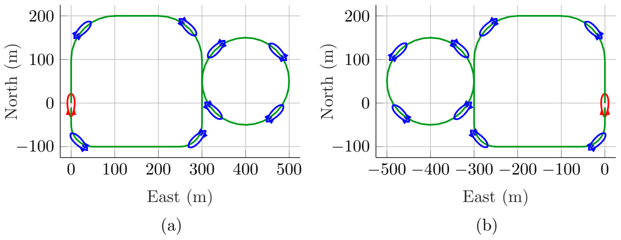


Figure 6.3. Training missions: (a) first path and (b) second path.

The stored dataset has approximately 375 MB. Training phase and further evaluations were carried out on a desktop architecture which features a four-core 4.00 GHz Intel Core i7-6700K

Processor, NVIDIA GeForce GTX 950, 32GB of RAM and Ubuntu 16 LTSOS. The system took 22 minutes and 17 seconds to train the afore-mentioned dataset.

Training was achieved using Matlab Neural Network Toolbox. The Scaled Conjugate Gradient (SCG) algorithm is used with 5000 epochs of training iterations using 70% of the collected data randomly taken as the training set, 15% used for validation set and 15% as the test set. The performance evaluation is made by Mean Squared Error (MSE).

The residual error of the trained NN is shown in Figure 6.4. Note that, about 80% of the total data is distributed around zero, in which 60% was taken as training data, 10% as validation data and about 10% as test data.

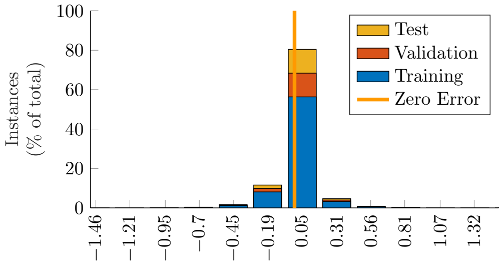


Figure 6.4. Error histogram of the NN dataset

The correlation coefficient (R-value), which is a linear regression between the NN predicted values and the targets, and best MSE are shown in Table 6.1. Values of R closer to 1 indicate better agreement between targets and predicted values. Note that both R-value and MSE are close for Training, Validation and Test dataset, which indicates that the NN is not overfitting the data.

Table 6.1. Correlation coefficient R-value and MSE.

Dataset	R-value	MSE
Training	0.99731	0.0206
Validation	0.99732	0.0205
Test	0.99720	0.0205
Total	0.99729	0.0205

In order to avoid high frequency oscillations in the estimation, a low-pass filter was introduced for filtering the input values $\mathbf{V}_g$, Φ and V_{pitot}. The low-pass filter can be expressed by the following Laplace transfer function:

$$\frac{Y(s)}{Z(s)} = \frac{1}{\tau s + 1} \tag{6.21}$$

where τ is the time constant (here we use $\tau = 1.5$ seconds), $Z(s)$ is the Laplace transform of the input and $Y(s)$ is the Laplace transform of the filter output. After filtering the measured sensor data, the input vector $\mathbf{z}_{nn}$ is computed and passed by the trained NN as shown in the block diagram from Figure 6.5. Finally, the estimated wind velocity χ_{nn} is given to the airship navigation system.

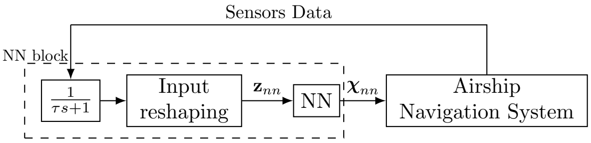


Figure 6.5. Resulting block diagram of the NN approach

6.4 Hybrid estimator

Here we propose a hybrid estimator that performs a fusion between both estimators, namely: from the EKF designed in Sect. 6.2 and NN designed in Sect. 6.3. This fusion is performed by changing the measure update stage of the EKF approach. The NN output χ_{nn} is added to the measurement vector of the EKF as a redundant measure. Thus, resulting in the new measurement vector $\mathbf{z}_{h_k}$, updating function $\mathbf{h}_{h_k}(\chi_k)$ and its respective Jacobian $\mathbf{H}_{h_k}$ shown below:

$$\mathbf{z}_{h_k} = \begin{bmatrix} \mathbf{z}_k \\ \chi_{nn} \end{bmatrix}, \mathbf{h}_{h_k}(\chi_k) = \begin{bmatrix} \mathbf{h}(\chi_k) \\ \chi_k \end{bmatrix} \text{ and } \mathbf{H}_{h_k} = \begin{bmatrix} \mathbf{H} \\ \mathbf{I}_3 \end{bmatrix},$$

where $\mathbf{I}_3$ is the identity matrix of third order. Then the EKF standard algorithm is used by updating the dimensions of the matrices $\mathbf{C}_k$, $\mathbf{K}_k$ and $\mathbf{R}$. The resulting estimator has a cascaded form as illustrated in Figure 6.6.

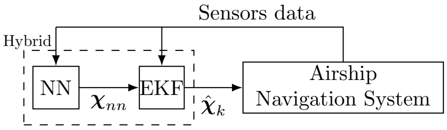


Figure 6.6. Hybrid estimator with cascaded form.

It is important to highlight that the NN used here is the same NN previously designed and trained in Sect. 6.3.

6.5 Simulation results

In this section two simulations are performed in order to evaluate all the three approaches presented before and establish a comparison with the traditional model-based approach proposed by [Cho et al., 2011].

During simulation the airship is well controlled with ideal feedback, meanwhile the estimators evaluated are receiving noisy data from the modeled sensors as shown in the block diagram from Figure 6.7. The sensors are modeled in the simulation environment with sample frequency as specified in Table 6.3. Also, each sensor has a generic modeling including a Gaussian noise as specified by the manufacturer in Table 6.2. The generic sensor modeling is shown in Figure 6.8, where w_z is a Gaussian noise, $\hat{z}$ is the true simulated value and z is the sensor output. All estimators use the same sample frequency of 16Hz ($t_s = 62.5$ ms).

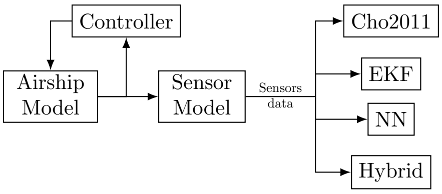


Figure 6.7. Block diagram of simulation

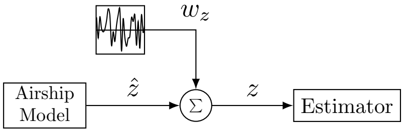


Figure 6.8. Block diagram of a generic sensor modeling.

Table 6.2. Sensor noise standard deviation

Data	$\sigma(w_y)$
ϕ, θ (rad)	$5.2{\cdot}10^{-3}$
ψ (rad)	0.1
$\mathbf{V}_g$ (m/s)	0.4
V_{pitot} (m/s)	$6.04{\cdot}10^{-4}$

Table 6.3. Sample frequency specification

Sensor	Frequency	Sampled data
IMU	100 Hz	$\mathbf{\Phi}$
GPS	4 Hz	$\mathbf{V}_g$
Pitot tube	18 Hz	V_{pitot}

An online repository[1] is available containing all approaches presented here. The algorithms are implemented in C/C++ and Python inside the Robot Operating System (ROS) [Quigley et al., 2009]. Also, in this same repository a link to the dataset used for the NN training task and the simulations performed are available for future researches.

[1] https://github.com/leve-fem/airship_estimator

6.5.1 First scenario

The first scenario considers wind with absolute value $|\vec{V}_w| = 2m/s$ and heading $\psi_w = \frac{\pi}{2}$ rad (blowing from East to West). Then in the instant $t = 160s$ the wind is intensified to $|\vec{V}_w| = 3m/s$ and its heading is changed to $\psi_w = \pi$ (blowing from South to North).

The airship is controlled to follow the trajectory shown in Figure 6.9. Five instants are highlighted with gray background in order to establish further comparisons with the results in Figure 6.10. Moreover, results using the estimator proposed by [Cho et al., 2011] were introduced as "Cho2011" in order to establish a comparison. The covariance matrices used in the Model-based approaches can be found in Appendix.

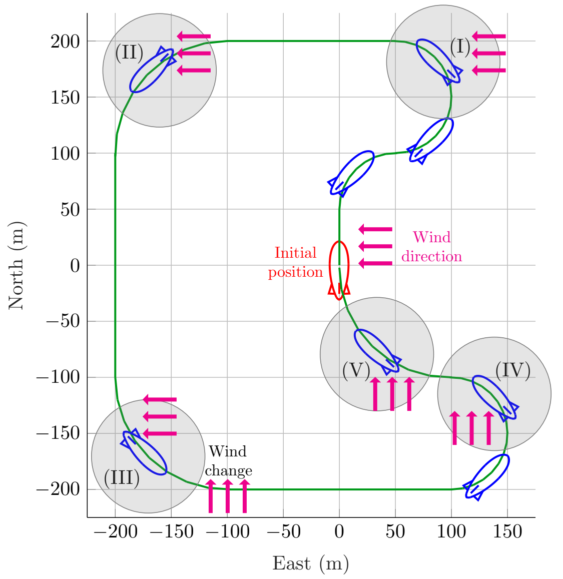


Figure 6.9. Simulation trajectory.

In Figure 6.10 is possible to note that, the "Cho2011" estimator has to acquire information about the motion in all directions before it converges to the correct values. After the instant (I), all estimators converges to values within of an acceptable error. It is important to note that, the NN has some high frequency oscillations at the trajectory curves, which deteriorate the performance. When the airship is following a straight line the designed NN has a good estimation, although sometimes with a bias from the real value.

Also in Figure 6.10, we can note that when the wind velocity has a significant variation in the instant (III), the two Model-based approaches ("EKF" and "Cho2011") do not converge immediately because both depend on information (given by the Pitot tube) about the other directions to converge to the correct wind velocity. Meanwhile the NN clearly has an instantly reaction to these variations. Even though the NN converges for a biased value, such information was sufficient to correct the estimation of the Hybrid approach before the instants (IV) and (V).

In these final instants, the Model-based estimators finally converges for values within a range of acceptable error.

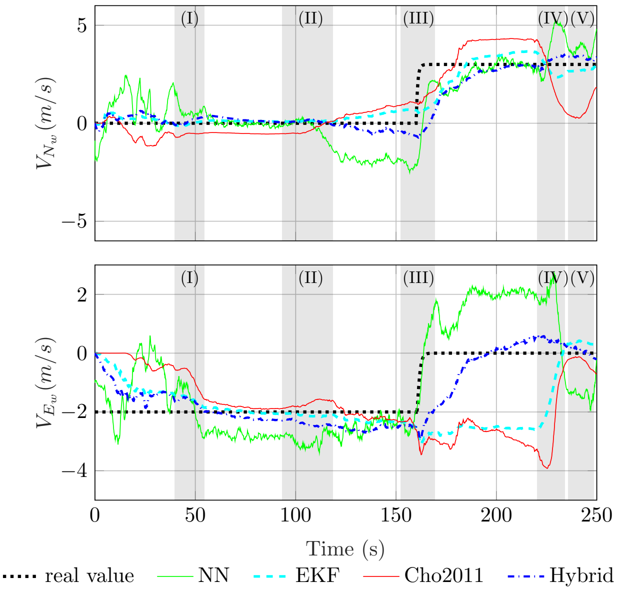


Figure 6.10. First scenario of simulation in the airship nonlinear model with realistic sensor noise: wind velocity estimation in North-East frame.

In Table 6.4 are shown the RMS values of the estimation errors $\tilde{V}_{N_w} = \hat{V}_{N_w} - V_{N_w}$ and $\tilde{V}_{E_w} = \hat{V}_{E_w} - V_{E_w}$ and the percentage in relation to the approach presented by [Cho et al., 2011]. By the RMS values we can observe that the NN had a better estimation of V_{E_w} in comparison to the Model-based approaches. However, for the component V_{N_w}, the Model-based approaches presented a better performance. Meanwhile the "Hybrid" which has the information of both approaches had the best performance in the estimation of V_{E_w} and acceptable estimation in the V_{N_w} component.

Table 6.4. RMS value of the estimation error.

	$\tilde{V}_{N_w}$ (m/s)	$\tilde{V}_{E_w}$ (m/s)
Cho2011	1.01	1.74
EKF	0.58 (-42.6%)	1.42 (-18.3%)
NN	1.19 (+17.8%)	1.25 (-27.0%)
Hybrid	0.74 (-26.7%)	0.71 (-59.2%)

Figure 6.11 shows the histograms of computational time necessary for each method. Although NN and Hybrid approaches have higher computational time (100 ∼ 160 microseconds), they are able of running at almost 1000Hz. Meanwhile, both Model-based approaches (EKF and Cho2011) are able of running at more than 1000Hz with a computational time between 25 and 35 microseconds. Note that the proposed approach by [Cho et al., 2011] has lowest computational cost, since the EKF proposed here has two additional measurement update equations.

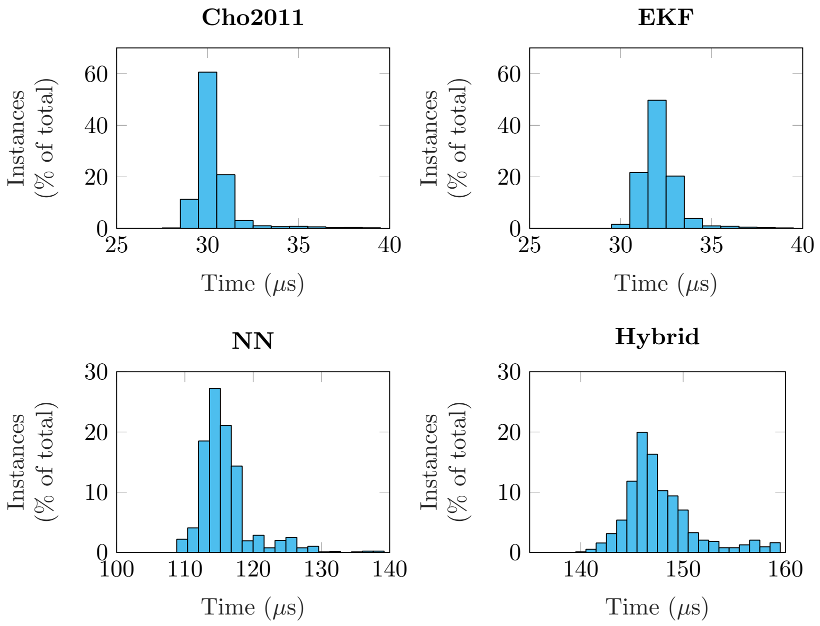


Figure 6.11. Computational time histogram

6.5.2 Second scenario

In a second simulation in the same trajectory of the previous simulation, it is considered a different variation in the wind velocity. Initially the wind has absolute value $|\vec{V}_w| = 2m/s$ and heading $\psi_w = 0$ (blowing from North to South), then in the instant $t = 160s$ the wind is intensified to $|\vec{V}_w| = 3m/s$ and its heading is changed to $\psi_w = \frac{\pi}{2}$rad (blowing from East to West). The results are shown in Figure 6.12 and resultant RMS error is shown in Tab. 6.5.

Table 6.5. RMS value of the estimation error in second simulation.

	$\tilde{V}_{N_w}$ (m/s)	$\tilde{V}_{E_w}$ (m/s)
Cho2011	0.71	0.52
EKF	0.52 (-26.5%)	0.38 (-26.9%)
NN	1.01 (+42.2%)	1.21 (+132.7%)
Hybrid	0.46 (-35.2%)	0.52 (0%)

In this second simulation, the airship starts with the nose against the wind, thus a few seconds of simulation are sufficient to the Model-based approaches estimate correctly the wind velocity components. Note that as the airship do curves in the scenario in instants (I) and (II), NN and Cho2011 approaches present some estimation errors, while the EKF and Hybrid maintain the estimation near the true value. As expected, adding 6.16 and 6.17 as additional measurement update equations to the EKF, results in better filtering and convergence.

In the overall simulation, Model-based approaches had better performance. It occurs be-

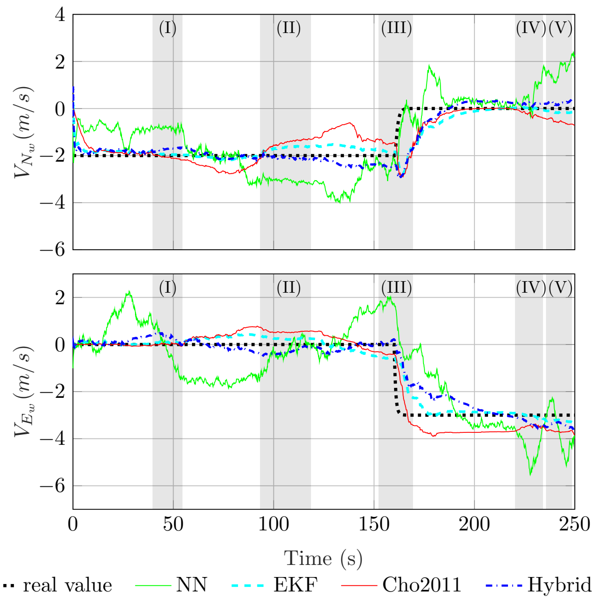


Figure 6.12. Second scenario of simulation in the airship nonlinear model with realistic sensor noise: wind velocity estimation in North-East frame.

cause the observability is not degraded during the wind transition at instant $t = 160$ seconds. After the instant (III), the airship is traveling to the East, thus the wind is against its nose. Therefore, the Pitot tube is able to capture the relative speed perfectly. By Table 6.5 we can also note that the EKF presented similar estimation performance for both wind components (V_{N_w} and V_{E_w}), differently from the previous simulation. Clearly, in this scenario, the NN is not too advantageous. It presented the higher RMS errors.

6.6 Conclusion

In this chapter we presented Model-based and Data-driven approaches for estimation of wind velocity for a robotic airship. The Model-based approach uses only kinematic equations of motion of the airship for the design of an EKF. The Data-driven proposed approach is composed by a NN trained with a dataset containing 1281 simulations in different conditions. Also, a novel Hybrid approach is proposed, by performing a fusion between the designed Model-based and Data-driven approaches with a cascaded structure.

A high-fidelity and realistic airship dynamic model for Matlab/Simulink platform was used in the simulations. Two simulation scenarios were presented. In the first, the wind variation is not detected properly by the model-based approaches, thus the Data-driven is advantageous in this scenario. In the second, the Model-based are more precise in the estimation, and the Data-driven performance was the same or worse.

The performance results obtained showed that the proposed EKF has a slightly better performance in comparison to the strategy found in the literature. It occurs because of the two additional measurement update equations that we introduced. Although, the performance was improved, EKF still has the same drawbacks when in situations of low observability. On the other hand, the NN presents higher sensitivity to wind variations in such situations, however with biased estimations and high frequency oscillations due to sensor noise and non-trained situations. The NN approach is independent of any mathematical model, however, requires a sufficient and representative database.

The Hybrid approach maintains satisfactory performance in the two simulation scenarios presented. This approach combines the advantages of both estimators (model-based and data-driven) and mitigates the drawbacks, once it has the information about the model and about previous missions (training dataset). However, it requires a representative dataset as well as a mathematical modeling.

As a general result, we can conclude that the cooperation between both approaches (Model-based and Data-driven) can be highly effective for solving estimation problems with observation deficiency. For the specific problem of wind estimation we obtained satisfactory results with the Hybrid approach. However, since the proposed methods are only analyzed theoretically and validated via simulation, an actual benchmark or field test is needed in the subsequent work to verify the proposed approaches. Future efforts will be made to validate these results outside of a simulation environment.

In Chapter 7 more simulations will be presented however closing this estimation loop with the Guidance and Control loops in order to verify the overall performance in closed-loop.

7 GUIDANCE

As mentioned before, the problem of trajectory tracking for autonomous aerial vehicles is normally separated into two main loops: the Guidance (outer loop) and Control (inner loop). The Guidance is the loop responsible for generating velocity commands based in the reference trajectory, while the inner loop (or Control) is responsible for stabilizing the vehicle motion and tracking the velocity commanded by the Guidance.

In the literature, we can find strategies which combine both loops such as receding horizon [Keviczky and Balas, 2003], differential flatness [Murray, 1996, Rathinam and Murray, 2002] and neural network based adaptive controls [Johnson et al., 2002]. In the context of autonomous airships, it is even more popular with the growing use of Backstepping [Zheng and Xie, 2017] and Sliding-modes [Vieira et al., 2017]. However these complex control techniques require much model information which makes difficult to use them in practice.

Separating the Guidance loop from the control loop is advantageous, since it allows the designer to choose a more suitable inner loop controller for the target vehicle. Also, it allows the reuse of designed guidance or control techniques in further projects requiring only to tune some parameters.

In this chapter we present the Line-of-Sight (LOS) strategy as a possible solution for the Guidance loop of an autonomous airship. This technique was initially designed for heavier-than-air aerial vehicles [Rysdyk, 2003] and marine vehicles [Lekkas and Fossen, 2003]. Mainly because of the non measured disturbances it received several improvements. In order to compensate wind forces, [Fossen et al., 2015] introduced an adaptive term with an integrative dynamics to the LOS guidance law, named Adaptive Line-of-Sight (ALOS). For the same problem, [Caharija et al., 2016] presented a similar approach also with this adaptive term. This strategy is now known as Integral Line-of-Sight (ILOS).

In this chapter, we present the formulations of traditional LOS, ALOS and ILOS. Then we analyze it for the case of an autonomous robotic airship. As a final result we obtain a Sensor-Based Integral Line-of-Sight (SBILOS) guidance formulation capable of covering a complete flight mission.

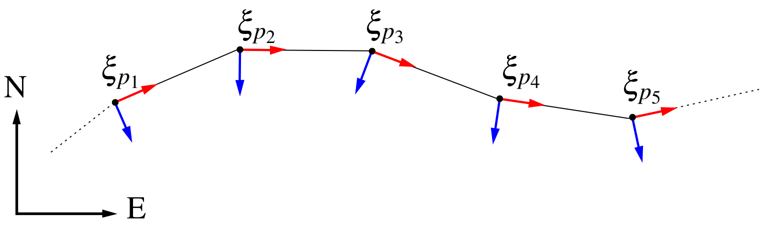


Figure 7.1. Reference trajectory as a sequence of poses.

7.1 Trajectory tracking formulation

Firstly, define pose as a vector composed by a position and orientation vectors in NED frame $\xi = [P^T, \Phi^T]^T$. Now, consider a path $\mathscr{P}$ as a set of $n \in \mathbb{Z}^+$ poses $\mathscr{P} := \{\xi_{p_1}, \xi_{p_2}, \cdots, \xi_{p_n}\}$ linked by straight lines as illustrated in Figure 7.1. During a flight mission, the airship reference pose $\xi_d = \xi_{p_k} \in \mathscr{P}$ given to the Guidance loop is switched sequentially ($k = 1, 2, ..., n$) according to a predefined rule. There are two traditional concepts for trajectory tracking:

- Path tracking – when each reference pose ξ_{p_k} has a temporal compromise for accomplishment, such that it is associated to a time instant $t = kt_p$, where t_p is a constant time period in SI units. In other words, the reference pose is switched sequentially for the next pose after a period of time t_p;
- Path following – when there is no temporal compromise for achieving the reference pose ξ_{p_k}, thus the switch between reference is due to an error tolerance.

Therefore, considering a given path $\mathscr{P}$, we will design a single guidance law in order to minimize the pose error $\tilde{\xi} = \xi - \xi_d$, where $\xi_d \in \mathscr{P}$, for the path following problem in cruise and hovering flight under wind disturbances.

7.2 Line-of-Sight formulation

In 2003, [Rysdyk, 2003] presented the LOS guidance law as a promising solution for laterally underactuated UAVs. The LOS is based on minimizing the future position error through the commanded heading. Basically, it define a virtual reference pose $\xi_d = [\mathbf{P}_d^T \, \Phi_d^T]^T$ which is located at a distance Δ_{los} from the true reference $\xi_p = [\mathbf{P}_p^T \, \Phi_p^T]^T$, as illustrated in Figure 7.2. The commanded heading is then defined by the guidance law given by:

$$\psi_d = \psi_p - \psi_{los} \tag{7.1a}$$

$$\psi_{los} = \tan^{-1}\left(\frac{e_y}{\Delta_{los}}\right) \tag{7.1b}$$

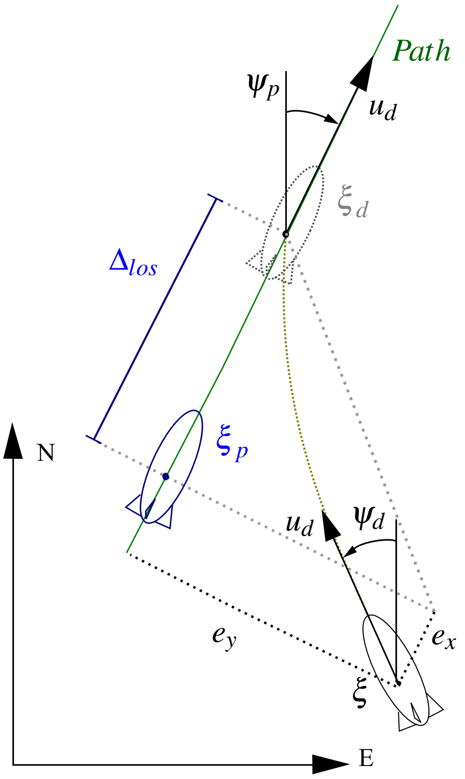


Figure 7.2. Line-of-sight formulation theory in North-East frame.

The position errors $\mathbf{e} = [e_x\ e_y\ e_z]^T$ are computed as below:

$$\mathbf{e} = \mathbf{S}_{\Phi_p}(\mathbf{P} - \mathbf{P}_p) \tag{7.2}$$

where $\mathbf{S}_{\Phi_p}$ is the rotation matrix from the NED frame to the ξ_p pose frame given the path orientation $\Phi_p = [0\ 0\ \psi_p]^T$.

Now, consider the following kinematic model:

$$\dot{\mathbf{P}} = \mathbf{S}_\Phi^T \mathbf{V} \tag{7.3}$$

where $\mathbf{V} = [u\ v\ w]^T$ is the vehicle groundspeed in the body frame and $\mathbf{S}_\Phi$ is the rotation matrix from NED to body frame.

The LOS algorithms for path following are usually employed at a kinematic level where the goal is to prescribe a desired value for the heading angle ψ to the vehicle. Consequently, the following assumptions are employed:

Assumption 5.1 The airship autopilot tracks a constant longitudinal velocity $u = u_d > 0$.

Assumption 5.2 The heading autopilot tracks the desired yaw angle perfectly such that $\psi = \psi_d$.

Assumption 5.3 The reference poses are linked by straight lines, and the adjacent poses are sufficiently near of each other such that the path can be considered continuous.

In the path following approach, the longitudinal error e_x is not defined, in other words, $e_x = 0$. Considering the position error defined in (7.2), the following simplified dynamics can be derived:

$$\dot{e}_y = u_d \sin(\psi_d - \psi_p), \tag{7.4}$$

whereψ_d is a commanded yaw angle.

Consider the following Lyapunov candidate:

$$V_{los} = \frac{1}{2} e_y^2, \tag{7.5}$$

thus its derivative is given by:

$$\dot{V}_{los} = e_y \Big(u_d \sin(\psi_d - \psi_p) \Big), \tag{7.6}$$

Applying the LOS control law 7.1, we obtain:

$$\dot{V}_{los} = -u_d \frac{e_y^2}{\sqrt{\Delta_{los}^2 + e_y^2}}. \tag{7.7}$$

We conclude that, as long as $u_d > 0$, the above derivative is negative definite for all $e_y \in \mathbb{R}$. Therefore, the origin $e_y = 0$ is a globally asymptotically stable equilibrium point of the closed-loop system. In other words, the lateral position error will remain close enough but also eventually converge to the equilibrium.

7.2.1 Adaptive Line-of-Sight

The traditional LOS guidance law presented is valid only for the specific scenario where the vehicle longitudinal velocity is much higher than external disturbances (e.g. missiles and supersonic aircrafts). When that is not true (e.g. submarines and airships), the kinematic model is changed. In this case, Adaptive Line-of-Sight (ALOS) was presented by [Fossen et al., 2015]. The ALOS guidance law is defined as:

$$\psi_d = \psi_p - \psi_{alos} \tag{7.8a}$$

$$\psi_{alos} = \tan^{-1}\left(\frac{e_y}{\Delta_{los}} + \varpi_\beta\right) \tag{7.8b}$$

$$\dot{\varpi}_\beta = \gamma_{los}\frac{u_d \Delta_{los} e_y}{\sqrt{(e_y + \Delta_{los}\varpi_\beta)^2 + \Delta_{los}^2}} \tag{7.8c}$$

where $\gamma_{los} > 0$ is a constant gain chosen by the designer. Supposing that the vehicle is moving forward with constant longitudinal speed, this guidance law was proved to be globally exponentially stable.

For the proof, consider the groundspeed as the sum of relative airspeed and the wind speed, as shown below:

$$\mathbf{V} = \mathbf{V}_a + \mathbf{V}_w, \tag{7.9}$$

where $\mathbf{V}_a = [u_a \; v_a \; w_a]^T$ is the airspeed and $\mathbf{V}_w = [u_w \; v_w \; w_w]^T$ is the wind speed in the airship body frame.

Considering (7.3), (7.9) and *Assumption 5.1* to *Assumption 5.3*, we can state the following:

$$\dot{\mathbf{P}} = \mathbf{S}_\Phi^T(\mathbf{V}_a + \mathbf{V}_w) \tag{7.10}$$

Considering that the vehicle is at constant altitude without angle of attack ($\alpha = 0$), under strictly horizontal wind, we may use the definition of true airspeed V_t given in (2.15) and sideslip angle β in (2.16a) to obtain the following:

$$\dot{\mathbf{P}} = \mathbf{S}_\Phi^T\left(\begin{bmatrix} u_a \\ v_a \\ w_a \end{bmatrix} + \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix}\right) = \mathbf{S}_\Phi^T\begin{bmatrix} V_t\cos(\beta) \\ V_t\sin(\beta) \\ 0 \end{bmatrix} + \begin{bmatrix} V_{N_w} \\ V_{E_W} \\ 0 \end{bmatrix}. \tag{7.11}$$

where V_{N_w} and V_{E_w} are the wind velocity components in North-East frame.

Considering that $\mathbf{P}_p$ and $\mathbf{S}_{\Phi_p}$ are constant and deriving (7.2), we obtain:

$$\dot{\mathbf{e}} = \mathbf{S}_{\Phi_p}\left(\mathbf{S}_\Phi^T\begin{bmatrix} V_t\cos(\beta) \\ V_t\sin(\beta) \\ 0 \end{bmatrix} + \begin{bmatrix} V_{N_w} \\ V_{E_w} \\ 0 \end{bmatrix}\right). \tag{7.12}$$

Considering that *Assumption 5.1*, *Assumption 5.2* and *Assumption 5.3* are valid, we obtain the following simplified error dynamics:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \end{bmatrix} = \begin{bmatrix} V_t\cos(\beta + \psi_d - \psi_p) - V_w\cos(\psi_p - \psi_w) \\ V_t\sin(\beta + \psi_d - \psi_p) + V_w\sin(\psi_p - \psi_w) \end{bmatrix}, \tag{7.13}$$

where $V_w = ||\mathbf{V}_w||_2$ and $\psi_w = \tan^{-1}\frac{V_{E_W}}{V_{N_W}}$.

For the path following case, we can ignore the longitudinal error e_x assuming that there is a longitudinal controller maintaining constant true airspeed V_t. Also, we ignored the error e_z, assuming there is vertical position controller which maintain a constant altitude.

Finally, we can simplify the model (7.13) to the following:

$$\dot{e}_y = V_t \sin(\beta + \psi_d - \psi_p) + V_w \sin(\psi_p - \psi_w) \tag{7.14}$$

Consider the following assumption:

Assumption 5.4 The sideslip angle β is small and constant during path following such that $\dot{\beta} = 0$.

Thus, the lateral error dynamics becomes:

$$\dot{e}_y = V_t \beta \cos(\psi_d - \psi_p) + V_t \sin(\psi_d - \psi_p) + V_w \sin(\psi_p - \psi_w) \tag{7.15}$$

Thus we may design a control law for ψ_d, such that, the error e_y remain close enough of the origin:

$$\psi_d = \psi_p - \tan^{-1}\left(\frac{e_y}{\Delta_{los}} + \varpi_\beta\right) \tag{7.16}$$

where ϖ_β is a control law to be designed later. The resulting dynamics in closed-loop is now given by:

$$\dot{e}_y = V_t \frac{\Delta_{los}(\beta - \varpi_\beta) - ey}{\sqrt{(e_y + \Delta_{los}\varpi_\beta)^2 + \Delta_{los}^2}} + V_w e_y \sin(\psi_w - \psi_p) \tag{7.17}$$

Therefore consider the previous presented Lyapunov candidate V_{los} given by (7.5). Considering the simplified dynamics (7.17), the resultant derivative is given by:

$$\dot{V}_{los} = -W(e_y, \varpi_\beta) + \frac{V_t e_y \Delta_{los} \tilde{\beta}}{\sqrt{(e_y + \Delta_{los}\varpi_\beta)^2 + \Delta_{los}^2}} + e_y V_w \sin(\psi_p - \psi_w) \tag{7.18}$$

where $W(e_y, \varpi_\beta) = \frac{V_t e_y^2}{\sqrt{(e_y + \Delta_{los}\varpi_\beta)^2 + \Delta_{los}^2}} > 0 \; \forall \, e_y, \, \varpi \in \mathbb{R}$ and $\tilde{\beta} = \beta - \varpi_\beta$ is the sideslip estimation error.

We choose the following dynamic equation for $\dot{\varpi}_\beta$:

$$\dot{\varpi}_\beta = \gamma \frac{V_t \Delta_{los} e_y}{\sqrt{\Delta_{los}^2 + (e_y + \Delta \varpi_\beta)^2}} \tag{7.19}$$

where $\gamma > 0$ is constant chosen by the designer.

Define an augmented Lyapunov function given by:

$$V_{alos} = \dot{V}_{los} + \frac{1}{2\gamma}\tilde{\beta}^2 \tag{7.20}$$

Note that $\dot{\tilde{\beta}} = -\dot{\varpi}_\beta$, therefore the derivative $\dot{V}_{alos}$ is given by:

$$\dot{V}_{alos} = -W(e_y, \varpi_\beta) + \frac{V_t e_y \Delta_{los}\tilde{\beta}}{\sqrt{(e_y + \Delta_{los}\varpi_\beta)^2 + \Delta_{los}^2}} + e_y V_w \sin(\psi_p - \psi_w) - \frac{1}{\gamma}\tilde{\beta}\dot{\tilde{\beta}} \tag{7.21}$$

which results in:

$$\dot{V}_{alos} = -W(e_y, \varpi_\beta) + V_{w_y} \tag{7.22}$$

where $V_{w_y} = V_w e_y \sin(\psi_p - \psi_w)$. At this point, we can note two main concerns of using line-of-sight for airships:

Assumption 5.5 The airship must move forward with airspeed $u_a > V_{max}$ where $V_{max} = ||\mathbf{V}_w||_2$. The airship is naturally lateral underactuated, thus in order to maintain the stability and control effectiveness it is preferable to have airspeed greater than the incident wind in the airship body;

Assumption 5.6 When objective is ground hover the reference orientation shall be $\psi_p = \psi_w + \pi$ in order to have zero influence from wind velocity ($\sin(\pi) = 0$) in the lateral error. Hovering flight is performed at $u_d = 0$. Therefore, the airship nose must be aligned against to the wind to accomplish previous mentioned remark $u_a \geq V_{max}$. In other words, performing hovering flight under lateral wind is a complicated task which may cause instability;

Also, three main advantages are raised:

Remark 5.1 As long as, the above restrictions are fully respected, the lateral error is asymptotically stable;

Remark 5.2 The control law may cover a complete flight mission (i.e. ground hover, cruise flight, vertical take-off and landing).

Remark 5.3 NOAMAY airship is equipped with a Pitot tube which provides the longitudinal dynamical pressure which has a quadratic relation with u_a. Thus it is possible to design a longitudinal controller to maintain $u_a > V_{max}$ as required by *Assumption 5.5*

Therefore, considering the Lyapunov candidate (7.20), the ALOS is globally exponentially stable if the *Assumptions 5.1* to *5.6* are satisfied.

7.2.2 Integral Line-of-Sight

The Integral Line-of-Sight (ILOS) was firstly proposed by [Caharija et al., 2016]. In his work, Caharija considers both two-dimensional model of a marine surface vessel and a three-dimensional model of a underwater autonomous vehicle. The ILOS guidance law for two dimensional model is given by:

$$\psi_d = \psi_p - \psi_{ilos} \tag{7.23a}$$

$$\psi_{ilos} = \tan^{-1}\left(\frac{e_y + \sigma_{los} e_{y_{int}}}{\Delta_{los}}\right) \tag{7.23b}$$

$$\dot{e}_{y_{int}} = \frac{\Delta_{los} e_y}{(e_y + \sigma_{los} e_{y_{int}})^2 + \Delta_{los}^2} \tag{7.23c}$$

where $\sigma_{los} > 0$ is a constant gain chosen by the designer.

[Caharija et al., 2016], demonstrated that ILOS guidance law was globally asymptotically stable for marine vehicles if the *Assumptions 5.1* to *5.6* are satisfied. In his formulation, the following model was considered:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{v}_a \end{bmatrix} = \begin{bmatrix} u_{a_d}\cos(\psi_d - \psi_p) - v_a \sin(\psi_d - \psi_p) + u_w \\ u_{a_d}\sin(\psi_d - \psi_p) + v_a \cos(\psi_d - \psi_p) + v_w \\ X(u_{a_d})\dot{\psi}_d + Y(u_{a_d}) v_a + w_v \end{bmatrix} \tag{7.24}$$

where v_a is the sway velocity of the vehicle, $w_v = \kappa_v(\gamma_w)\sin(\psi_w - \psi_d + \psi_p)$ represents the external disturbance and $\gamma_w = \psi_d - \psi_p - \psi_w - \pi$. The detailed surge and yaw dynamics are not considered, and hence u_{a_d} and ψ_d are the control inputs of (7.24). The functions $X(u_{a_d})$ and $Y(u_{a_d})$ satisfy the following assumptions:

Assumption 5.7 The functions $X(u_{a_d})$ and $Y(u_{a_d})$ are continuous and bounded for bounded arguments.

Assumption 5.8 $Y(u_{a_d})$ is such that $|Y(u_{a_d})|$ is strictly increasing for $u_{a_d} > 0$ and satisfies $Y(u_{a_d}) \leq -Y^{min} < 0, \forall u_{a_d} \in [-V_{max}, V_t]$, where Y^{min} is a positive constant.

[Caharija et al., 2016] provides a Lyapunov closed loop analysis which yields explicit bounds on the guidance law gains to guarantee uniform global asymptotic stability (UGAS) and uniform local exponential stability (ULES). The demonstration considers same assumptions as ALOS and does not add any different details from the previous one. Therefore, it was not included in this work. We recommend the reader to see the entire proof in [Caharija et al., 2016].

7.2.3 Sensor-based Line-of-sight

Taking advantage of the wind estimator designed in Chapter 6, we will provide a Sensor-based solution for Line-of-sight. This solution cancels the drag caused by the term V_{w_y} present in the lateral error dynamics (7.13).

The afore mentioned Line-of-sight strategies (i.e. ALOS and ILOS), were proposed using slightly different models for the same kind of marine vehicle. Both, [Fossen et al., 2015] and [Caharija et al., 2016] obtained an adaptive term to cancel the external disturbances however considering strong assumptions.

In this new solution we will use solely the traditional LOS control law (7.1). Then, we apply Reference Shaping in ψ_p considering the estimation provided by the wind estimator designed in Chapter 6.

Wind Based Reference Shaping

Considering that $\mathbf{V}_p = [u_d \; 0 \; 0]^T$ is the reference velocity vector in the path frame, thus we can obtain the yaw reference ψ_{a_p} as below:

1. Obtain the wind velocity $\mathbf{V}_{NED_w}$ in North-East-Down (NED) frame through an estimator (see Chapter 6):

$$\mathbf{V}_{NED_w} = \mathbf{S}^T_{\Phi_w} \begin{bmatrix} V_w \\ 0 \\ 0 \end{bmatrix}, \tag{7.25}$$

   where $\Phi_w = [0 \; 0 \; \psi_w]^T$ is the wind heading and V_w is the magnitude.

2. Rotate the vector $\mathbf{V}_p$ to the NED frame:

$$\mathbf{V}_{NED_p} = \mathbf{S}^T_{\Phi_p} \mathbf{V}_p. \tag{7.26}$$

3. Obtain the reference airspeed in NED frame:

$$\mathbf{V}_{NED_{ap}} = \mathbf{V}_{NED_p} - \mathbf{V}_{NED_w}. \tag{7.27}$$

4. Compute the new heading reference:

$$\psi_{a_p} = \tan^{-1}\left(\frac{V_{E_{ap}}}{V_{N_{ap}}}\right). \tag{7.28}$$

Considering that *Assumption 5.1*, *Assumption 5.2* and *Assumption 5.3* are valid, define the Sensor Based LOS guidance law (or SBLOS) given by:

$$\psi_d = \psi_{a_p} - \psi_{los}, \tag{7.29}$$

where ψ_{los} is given by (7.1). The guidance approaches ALOS and ILOS already consider the external disturbances in their formulations. However, since this strategy does not interfere in the stability proofs, we can also add it to ALOS (7.8) and ILOS (7.23). Therefore, in further simulations we will introduce this solution in order to verify the improvement of guidance performance.

Remark 5.4 Note that, if the objective is ground hover (or hovering flight), the desired ground speed shall be $u_d = 0$. As a consequence we obtain $\psi_{a_p} = \psi_w + \pi$, which is the ideal heading as discussed earlier in *Assumption 5.6*.

7.3 Simulation Results

In this section we establish a comparison between six approaches of guidance derived from the traditional Line-of-sight control law. Simulations are performed in a scenario where the airship must follow a path with constant airspeed of $u_a = 5$m/s and altitude $h = 50$m. Also wind is considered during all the mission with $V_w = 3$m/s, $\psi_w = 150°$ and turbulence with $\sigma = 1m/s$ modeled as demonstrated by [McLean, 1990].

The final control architecture including the Wind Based Reference Shaping (WRSB) for the commanded heading is exemplified in the block diagram of Figure 7.3. In this block diagram, we included the lateral and longitudinal controllers designed in Sec. 4.1 and 4.2, respectively. The INDI controller guarantees the tracking of attitude, linear velocities and angular rates. In addition, the VPC designed in Sec. 4.2.3 assures the tracking of altitude. Also, we added two Proportional-Derivative (PD) controllers for the tracking of longitudinal and vertical positions given by:

$$\begin{bmatrix} u_d \\ V_{D_d} \end{bmatrix} = \begin{bmatrix} u_p - k_{p_{lon}}(e_x) - k_{d_{lon}}(u_r - u_p) \\ w_p - k_{p_{vert}}(e_z) - k_{d_{vert}}(w_r - w_p) \end{bmatrix} \tag{7.30}$$

where:

$$\mathbf{V}_r = \mathbf{S}_{\Phi_p}\mathbf{S}_{\Phi}^T\mathbf{V}, \ \ \mathbf{V} = [u \ \ v \ \ w]^T, \ \ \mathbf{V}_r = [u_r \ \ v_r \ \ w_r]^T;$$

$k_{p_{lon}}$, $k_{d_{lon}}$, $k_{p_{vert}}$ and $k_{d_{vert}}$ are constant gains; u_p and w_p are the feedforward term for longitudinal and vertical ground speed, respectively; u_d and V_{D_d} are the commanded longitudinal and vertical ground speeds. The constants and gains are depicted in Tables 7.1 and 4.2.In addition, we designed Second-order differentiators (or SOD proposed in Sec. 3.4.3) in order to obtain the state derivatives for the INDI based controllers. The natural frequency and damping ratio is the same for all SODs and given by $\omega_n = 100 rad/s$ and $\zeta = 2.0 s^{-1}$. Also, the Input Scale Gain (proposed in 3.4.4) is applied for both Longitudinal and Lateral controllers. For the lateral motion, we use ISG given by $\Lambda_h = diag([0.1,\ 0.4])$, and for the longitudinal motion we use $\Lambda_v = diag([0.2,\ 0.2,\ 0.2])$.

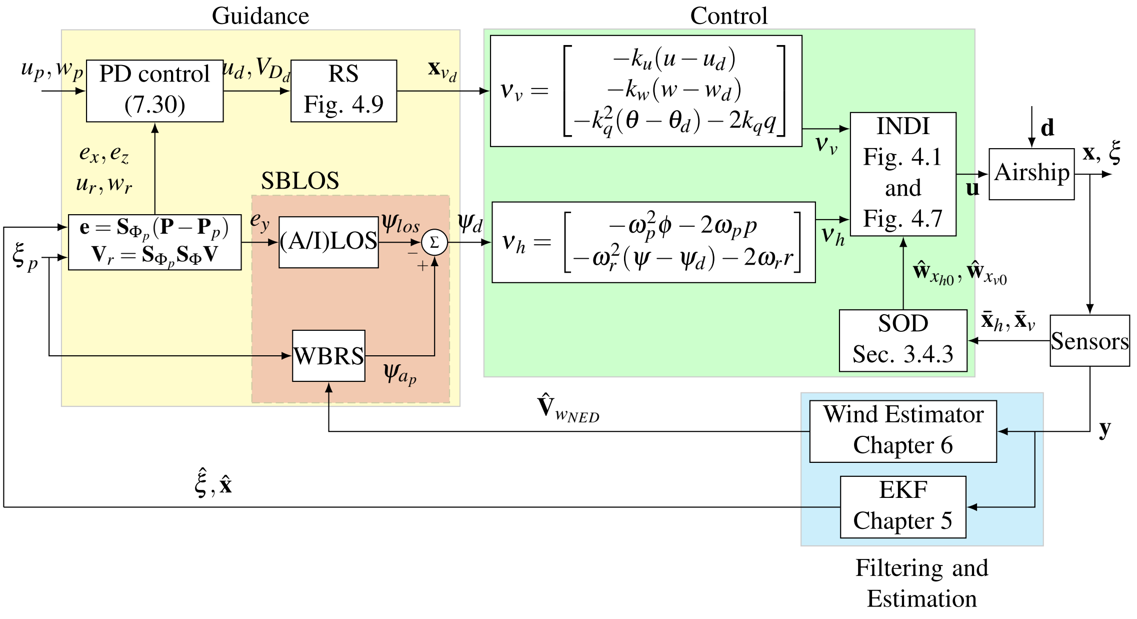

Guidance
Control
u_p, w_p
PD control (7.30)
u_d, V_{D_d}
RS Fig. 4.9
$\mathbf{x}_{v_d}$
$v_v = \begin{bmatrix} -k_u(u-u_d) \\ -k_w(w-w_d) \\ -k_q^2(\theta-\theta_d)-2k_q q \end{bmatrix}$
v_v
INDI Fig. 4.1 and Fig. 4.7
$\mathbf{d}$
Airship
$\mathbf{x}, \xi$
$\mathbf{u}$
e_x, e_z
u_r, w_r
SBLOS
$\mathbf{e} = \mathbf{S}_{\Phi_p}(\mathbf{P}-\mathbf{P}_p)$
$\mathbf{V}_r = \mathbf{S}_{\Phi_p}\mathbf{S}_{\Phi}\mathbf{V}$
ξ_p
e_y
(A/I)LOS
ψ_{los}
Σ
−
+
ψ_d
$v_h = \begin{bmatrix} -\omega_p^2\phi - 2\omega_p p \\ -\omega_r^2(\psi-\psi_d) - 2\omega_r r \end{bmatrix}$
v_h
$\hat{\mathbf{w}}_{x_{h0}}, \hat{\mathbf{w}}_{x_{v0}}$
SOD Sec. 3.4.3
$\bar{\mathbf{x}}_h, \bar{\mathbf{x}}_v$
Sensors
WBRS
ψ_{a_p}
$\hat{\mathbf{V}}_{WNED}$
Wind Estimator Chapter 6
$\mathbf{y}$
$\hat{\xi}, \hat{\mathbf{x}}$
EKF Chapter 5
Filtering and Estimation

Figure 7.3. Block diagram of the overall closed loop system for simulation.

A total of six simulations were performed. For each simulation a different yaw command ψ_d is given, switching between: LOS given by (7.1), ALOS given by (7.8), ILOS given by (7.23) and the sensor-based versions of each of then given by (7.29) (nominated SBLOS, SBALOS and SBILOS, respectively). The resulting trajectory of each guidance approach is shown in Figure 7.4. The tracking errors are shown in Figure 7.5.

In order to establish a performance comparison, we use $\sigma_{RMS}^{(a/i)los}$ to denote the RMS of lateral error calculated for each guidance approach. During cruise flight (see Figure 7.5), the best performance was obtained by SBILOS with $\sigma_{RMS}^{sbilos} = 1.51$m, with minor differences to SBLOS ($\sigma_{RMS}^{sblos} = 1.76$m) and SBALOS ($\sigma_{RMS}^{sbalos} = 1.98$m). In another hand, ILOS obtained the worst performance, presenting $\sigma_{RMS}^{ilos} = 5.4$m followed by LOS with $\sigma_{RMS}^{los} = 5.28$m and ALOS with $\sigma_{RMS}^{alos} = 3.25$m.

Table 7.1. Constants and Gains for (A/I)LOS

Name	Value	Name	Value	Name	Value
Δ_{los}	10 m	$k_{p_{lon}}$	0.01 s^{-1}	$k_{p_{vert}}$	0.2 s^{-1}
γ_{los}	0.001	$k_{d_{lon}}$	0.1	$k_{d_{vert}}$	0.2
σ_{los}	6.0 m/s	ω_p	$2.3 rad/s$	ω_r	$2.3 rad/s$

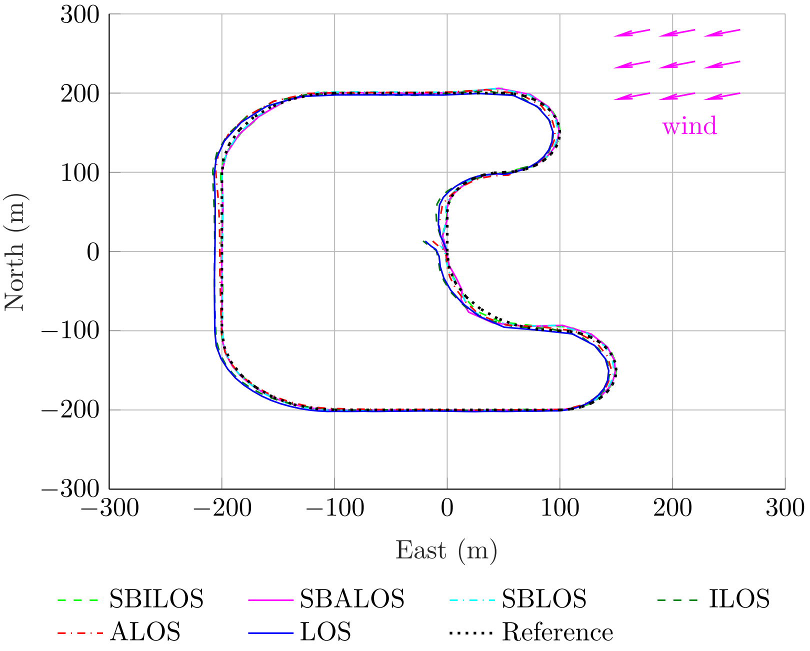


Figure 7.4. Resulting trajectory of simulation of each approach.

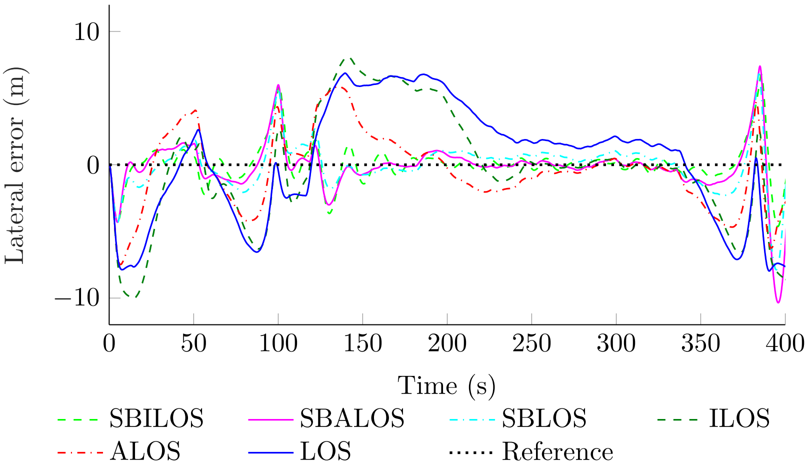


Figure 7.5. Resulting tracking error of each approach.

In hovering flight (see Figure 7.6), we note that the traditional approaches from literature are not able to maintain zero lateral error as shown in Figure 7.7. This occurs, because the adaptive dynamics are too slow to cancel the heading difference between airship and wind. Even though, the sideslip angle is maintained near null (as shown in Figure 7.9), ALOS and ILOS do not cancels the lateral velocity caused by the wind drag. Meanwhile, the approaches SB(A/I)LOS changes the commanded heading (see Figure 7.8) reducing the drag effect and thus maintaining the airship stable at the reference position.

Table 7.2 presents a performance comparison of RMS of the lateral error obtained by each guidance approach. By this table we can note that, the information about the wind heading can significantly increase the guidance performance. The most advantageous strategy is SBILOS,

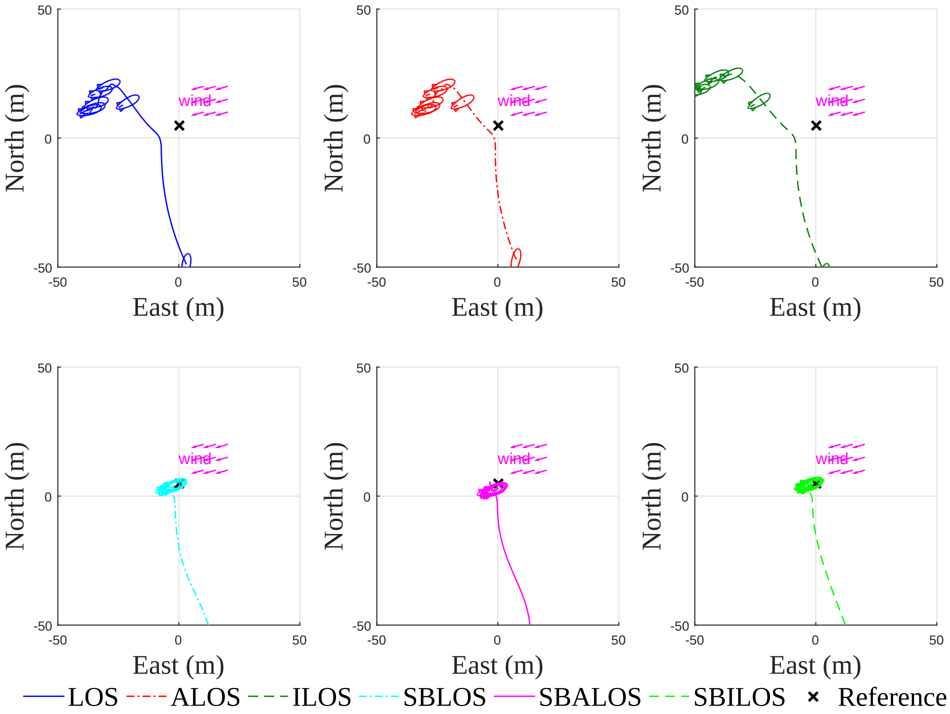


Figure 7.6. Resulting trajectory of simulation of each approach in Hovering flight.

in contrast, ILOS obtained the higher RMS errors.

Table 7.2. RMS of the lateral error when in hovering flight

Guidance law	σ_{RMS} (m)	Guidance	σ_{RMS} (m)
LOS	30.09	SBLOS	0.9554
ALOS	23.89	SBALOS	2.24
ILOS	39.90	SBILOS	0.84

7.4 Conclusion

We presented a review of three guidance solutions commonly used in the literature for laterally underactuated vehicles. The first one is the so called Line-of-sight developed by [Rysdyk, 2003]. Taking advantage of the theory behind this approach, [Fossen et al., 2015] and [Caharija et al., 2016] proposed new guidance control laws called Adaptive Line-of-sight and Integral Line-of-sight. Although they obtained similar control laws using Lyapunov proofs, the performance obtained in simulation was slightly different. ALOS presented better performance than ILOS, for the NOAMAY airship case. However, we demonstrate by both Lyapunov analysis and simulation results, that (A/I)LOS can be unstable when dealing with the hovering flight

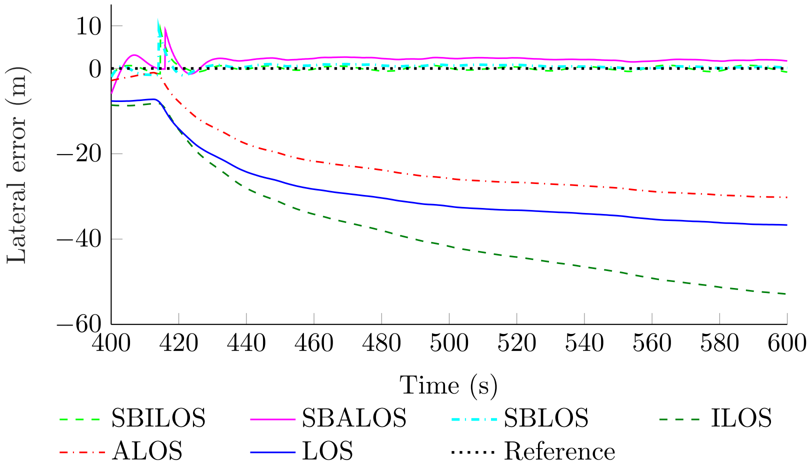


Figure 7.7. Resulting tracking error of each approach in hovering flight.

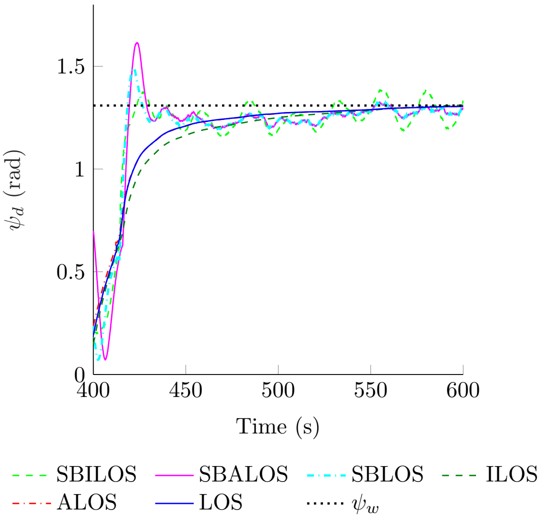


Figure 7.8. Airship commanded heading.

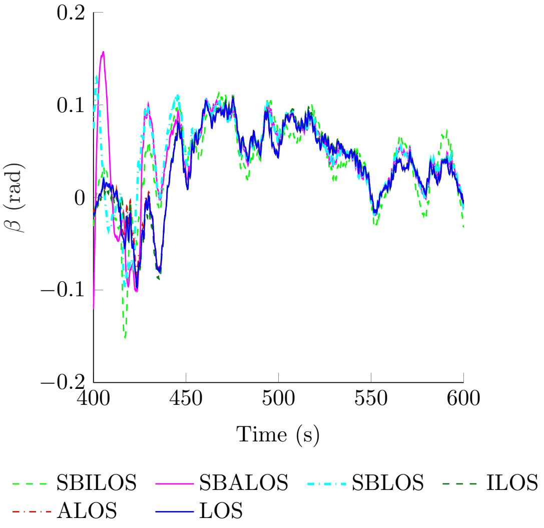


Figure 7.9. Airship sideslip angle.

problem.

Considering this scenario, we proposed the Sensor-based Line-of-Sight (SBLOS) strategy for dealing with both problems of hovering and cruise flight without the need of a complex switching scheme. We propose to add the information provided by the Hybrid Wind Estimator designed in Chapter 6, in order to reshape the reference heading. As a result, the drag caused by the wind is canceled. Closed-loop simulations demonstrated the efficacy of SB(A/I)LOS for the NOAMAY airship model.

Therefore, we conclude that the information about incident wind in airship body (provided by the Hybrid Wind Estimator from Chapter 6) can significantly increase guidance performance. In addition, the proposed solution removes the need of switching controllers when changing the objective between cruise and hovering flight.

8 FINAL CONSIDERATIONS

In this thesis we aimed to develop a control architecture for the navigation system of a robotic airship using incremental controllers capable of covering all flight phases (VTOL, cruise and hovering flight) and transitioning between them without a complex switching scheme. As stated before, the accomplishment of trajectory tracking/following tasks depends on three main topics: Guidance, Control and Estimation.

In Chapter 3, the Incremental Dynamics (ID) was presented as a promising solution for airship control. ID has an intrinsic robustness to model uncertainties which is an important characteristic, mainly because airships are nonlinear systems with many aerodynamic uncertainties. However, some design issues are highlighted, such as: (1) actuator redundancy and saturation; (2) measurement noise and delay sensitivity. In order to mitigate (1), we propose the combination of Washout Filter (WOF) and Natural Anti Wind-up (NAW). To mitigate (2), the combination of Input Scale Gain and Second-order Differentiator (SOD) is presented. Through numerical examples, and closed loop analysis, we showed that, the proposed solutions increase incremental controllers performance and robustness to measurement errors.

The control design of INDI presented in Chapter 4, for tracking desired linear velocities and angular rates (as proposed in this thesis) is also an important contribution to the development of autonomous airships. It considers nonlinearities in the model while providing robustness in the presence of parameter uncertainties. The Longitudinal guidance loop is mainly based on Incremental Dynamics and employs a RS methodology to provide a dynamical filtering of the commands such that the demanded signals are within the capabilities of the vehicle.

The proposed control loop was tested by performing simulations with different flight phases. The obtained results demonstrate their efficiency to stabilize the system and track the desired ground velocities with the performance characteristics imposed by the commanded signals.

Robustness tests showed that the proposed control approach is insensitive to model uncertainties within the expected range found in reality. This property is relevant since the accurate identification of airship models is costly and extremely difficult. Therefore, the proposed approach may be the basis for controlling an autonomous airship. A simplified model of the platform has shown to be a sufficiently good approximation to compute the necessary linearized input matrix and efficiently control the vehicle.

In Chapter 5, we provide a comparison between estimation and filtering techniques. Three approaches were addressed: Low-pass Filter (LPF), Extended Kalman Filter (EKF), and Unscented Kalman Filter (UKF). Also, we present the modeling of sensors based in the specification of the actual sensors equipped in NOAMAY airship. For the comparison, we have chosen the covariance parameters for the UKF searching for the best results possible. Then the same parameters were applied to EKF, without any additional optimization. Consequently, we initially expected to achieve a better performance with UKF. However, from the results presented in Chapter 5, one can observe that the EKF clearly had a better performance. It suggests that, for this specific platform and under the same conditions, the EKF may be a more suitable choice. These are interesting results, since in [Giannitrapani et al., 2011] the UKF presented a slightly better performance in terms of average localization accuracy for their specific case of a spacecraft model.

In Chapter 6, the wind estimation problem was addressed. We presented Model-based and Data-driven approaches for estimation of wind velocity for a robotic airship. The Model-based approach uses only kinematic equations of motion for the design of an EKF. The Data-driven proposed approach is composed by a Neural Network (NN) trained with a large dataset with several simulations in different conditions. Also, a novel Hybrid approach is proposed, by performing a fusion between the designed Model-based and Data-driven approaches through a cascaded structure.

Simulation results obtained showed that the proposed EKF has a slightly better performance in comparison to strategies found in the literature [Cho et al., 2011]. It occurs because of the two additional measurement update equations that we included. Meanwhile, the NN presented better sensitivity to wind variations, however with biased estimations. As a consequence the Hybrid approach had the better performance, once it had the information of both approaches. These results show that the cooperation between Model-based and Data-driven can be highly effective for solving estimation problems.

Finally, the Guidance problem was addressed in Chapter 7. The chapter presents a careful review on Line-of-sight (LOS). LOS is usually applied to laterally underactuated vehicles. The works found in literature, mainly address the situation where the vehicle is at constant longitudinal movement. Then, we addressed the problem of hovering flight and proposed a solution. As a result we obtained the Sensor-based Line-of-sight. This novel solution uses the information about wind velocity (provided by a wind estimator) to compute a new heading reference. Simulation results showed that with SBLOS the airship is capable of switching between cruise and hovering flight without the need of a complex switching strategy.

As a final result, we obtained a closed loop system addressing Guidance, Control and Estimation problems which was depicted in Figure 7.3. As demonstrated in simulation results, this final control architecture is capable of covering all flight phases of a complete mission, namely: cruise, hovering, vertical take-off and landing.

8.1 Future Works

As future works, the wind estimator could be improved to a three dimensional (3D) estimator. Such estimative could be used for the choice of pitch angles in the VPC loop. Also, the optimization proposed for the VPC loop could be also improved by considering the vehicle airspeed instead of the pure kinematics of ground speed. These two improvements may help to save battery energy when in cruise flight, since the aerodynamics could sustain part of the vehicle weight by increasing the angle of attack.

Another topic of interest of this thesis, is the development of INDI for the coupled model of the airship (namely lateral and longitudinal). In this thesis we only address the design of INDI controllers for the decoupled model, in other words, we design two INDI loops: one for the lateral motion; and another for the longitudinal motion. Thu, the design of a coupled strategy should be investigated, and consequently a comparison between these two strategies could be established. Additionally, Incremental Backstepping and Incremental Sliding modes should be investigated and compared with the INDI proposed approach.

Also, LOS is not the most recent technique in the literature. Although it is an advantageous strategy for guidance of laterally underactuted vehicles, it could be improved, by considering the 3D space. Additionally, this technique could be compared with other solutions such as Vector Fields and L1 Adaptive Control.

Bibliography

D. Acosta and S. Joshi. Adaptive nonlinear dynamic inversion control of an autonomous airship for the exploration of titan. In *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, aug 2007.

P. B. Acquatella, E.-J. van Kampen, and Q. P. Chu. Incremental backstepping for robust nonlinear flight control. In *EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation & Control*, April 2013.

S. Allison, H. Bai, and B. Jayaraman. Estimating wind velocity with a neural network using quadcopter trajectories. In *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2019.

S. Ann, S. Lee, and Y. Kim. Reference shaping for impact angle and time control under field-of-view limit. *IFAC-PapersOnLine*, 50(1):15191–15196, July 2017.

R. R. M. Arias. Modelagem de um dirigível robótico com propulsão elétrica de quatro motores. Master's thesis, Universidade Estadual de Campinas, 2014.

G. S. C. Avellar, G. D. Thums, R. R. Lima, P. Iscold, L. A. B. Torres, and G. A. S. Pereira. On the development of a small hand-held multi-UAV platform for surveillance and monitoring. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. Institute of Electrical & Electronics Engineers (IEEE), may 2013.

J. Azinheira, E. C. de Paiva, J. Ramos, and S. Bueno. Mission path following for an autonomous unmanned airship. In *Proceedings IEEE International Conference on Robotics and Automation. Symposia Proceedings*. Institute of Electrical & Electronics Engineers (IEEE), 2000.

J. Azinheira, A. Moutinho, and J. Carvalho. Lateral control of airship with uncertain dynamics using incremental nonlinear dynamics inversion. *IFAC-PapersOnLine*, 48(19):69–74, 2015.

J. R. Azinheira, A. Moutinho, and E. C. D. Paiva. Airship hover stabilization using a backstepping control approach. *Journal of Guidance, Control, and Dynamics*, 29(4):903–914, jul 2006.

J. R. Azinheira, E. C. Paiva, and A. Moutinho. Erratum on "influence of wind speed on airship dynamics". *Journal of Guidance, Control, and Dynamics*, 31(2):pp. 443–444, Mar. 2008.

J. R. Azinheira, A. Moutinho, and E. C. de Paiva. A backstepping controller for path-tracking of an underactuated autonomous airship. *International Journal of Robust and Nonlinear Control*, 19(4):pp. 418–441, Mar. 2009.

B. Bacon, A. Ostroff, and S. Joshi. Reconfigurable NDI controller using inertial sensor failure detection & isolation. *IEEE Transactions on Aerospace and Electronic Systems*, 37(4):1373–1383, 2001.

A. Benallegue, A. Mokhtari, and L. Fridman. Feedback linearization and high order sliding mode observer for a quadrotor UAV. In *International Workshop on Variable Structure Systems, 2006. (VSS06)*. Institute of Electrical and Electronics Engineers (IEEE), jul 2006.

W. Caharija, K. Y. Pettersen, M. Bibuli, P. Calado, E. Zereik, J. Braga, J. T. Gravdahl, A. J. Sorensen, M. Milovanovic, and G. Bruzzone. Integral line-of-sight guidance and control of underactuated marine vehicles: Theory, simulations, and experiments. *IEEE Transactions on Control Systems Technology*, 24(5):1623–1642, sep 2016.

G. E. Carichner and L. M. Nicolai. *Fundamentals of Aircraft and Airship Design Volume 2 - Airship Design and Case Studies*. AIAA, 2013.

D. K. Chaturvedi, R. Chauhan, and P. K. Kalra. Application of generalised neural network for aircraft landing control system. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 6(6):441–448, Sept. 2002.

A. Cho, J. Kim, S. Lee, and C. Kee. Wind estimation and airspeed calibration using a UAV with a single-antenna GPS receiver and pitot tube. *IEEE Transactions on Aerospace and Electronic Systems*, 47(1):109–117, jan 2011.

R. A. Cordeiro, A. S. Marton, J. R. Azinheira, J. R. H. Carvalho, and A. Moutinho. Increased robustness to delay in incremental controllers using input scaling gain. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–1, 2021. doi: 10.1109/taes.2021.3123215. URL https://doi.org/10.1109/taes.2021.3123215.

V. Cortés, J. Azinheira, and E. Paiva. Parameter identification of nonlinear dynamic model of aurora airship. *IFAC Proceedings Volumes*, 36(17):49–54, Sept. 2003. doi: 10.1016/s1474-6670(17)33368-2.

L. R. da Silva, R. C. Flesch, and J. E. Normey-Rico. Analysis of anti-windup techniques in PID control of processes with measurement noise. *IFAC-PapersOnLine*, 51(4):948–953, 2018. doi: 10.1016/j.ifacol.2018.06.100.

O. Daskiran, B. Huff, and A. Dogan. Low speed airship control using reinforcement learning and expert demonstrations. In *AIAA Atmospheric Flight Mechanics Conference*. American Institute of Aeronautics and Astronautics, Jan. 2017.

M. de Oliveira. Dirigível sobre a floresta. http://revistapesquisa.fapesp.br/2018/05/21/dirigivel-sobre-a-floresta, may 2018. [Online; accessed in 03-august-2020].

A. Elfes, S. S. Bueno, M. Bergerman, J. J. G. Ramos, and S. B. V. Gomes. Project aurora: Development of an autonomous unmanned remote monitoring robotic airship. *Journal of the Brazilian Computer Society*, 4, 04 1998.

A. Elfes, S. S. Bueno, J. J. G. Ramos, E. C. de Paiva, M. Bergerman, J. R. H. Carvalho, S. M. Maeta, L. G. B. Mirisola, B. G. Faria, and J. R. Azinheira. Modelling, control and perception for an autonomous robotic airship. In *Sensor Based Intelligent Robots*, pages 216–244. Springer Science Business Media, 2002.

A. Elfes, S. S. Bueno, M. Bergerman, E. C. D. Paiva, and J. G. Ramos. Robotic airships for exploration of planetary bodies with an atmosphere: Autonomy challenges. *Autonomous Robots, SpringerLink*, pages pp. 147–164, 2003.

A. Elfes, J. Hall, J. Montgomery, C. Bergh, and B. Dudik. Towards a substantially autonomous aerobot for exploration of titan. In *IEEE International Conference on Robotics and Automation ICRA 2004*. Institute of Electrical and Electronics Engineers (IEEE), 2004.

J. Farrell, M. Polycarpou, M. Sharma, and W. Dong. Command filtered backstepping. *IEEE Transactions on Automatic Control*, 54(6):1391–1395, June 2009.

T. I. Fossen. *Mathematical Models for Control of Aircraft and Satellites*, volume 3th ed. Department of Engineering Cybernetics, NTNU, apr 2013.

T. I. Fossen, K. Y. Pettersen, and R. Galeazzi. Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, 23(2):820–827, mar 2015.

G. D. Francesco and M. Mattei. Modeling and incremental nonlinear dynamic inversion control of a novel unmanned tiltrotor. *Journal of Aircraft*, 53(1):73–86, jan 2016.

A. Giannitrapani, N. Ceccarelli, F. Scortecci, and A. Garulli. Comparison of EKF and UKF for spacecraft localization via angle measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 47(1):75–84, Jan. 2011.

D. Gobiha and N. K. Sinha. Hover corridor for a stratospheric airship. *IFAC-PapersOnLine*, 51 (1):371–376, 2018.

V. M. Goncalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra, and G. A. S. Pereira. Vector fields for robot navigation along time-varying curves in n-dimensions. *IEEE Transactions on Robotics*, 26(4):647–659, aug 2010.

V. M. Gonçalves, L. C. A. Pimenta, C. A. Maia, G. A. S. Pereira, B. C. O. Dutra, N. Michael, J. Fink, and V. Kumar. Circulation of curves using vector fields: Actual robot experiments in 2d and 3d workspaces. In *2010 IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers (IEEE), may 2010.

M. Hassouneh, H.-C. Lee, and E. Abed. Washout filters in feedback control: benefits, limitations and extensions. In *Proceedings of the 2004 American Control Conference*. IEEE, 2004.

E. Hygounenc and P. Soueres. Automatic airship control involving backstepping techniques. In *IEEE International Conference on Systems, Man and Cybernetics*. Institute of Electrical and Electronics Engineers (IEEE), oct 2002.

E. Hygounenc, I.-K. Jung, P. Souères, and S. Lacroix. The autonomous blimp project of LAAS-CNRS: Achievements in flight control and terrain mapping. *The International Journal of Robotics Research*, 23(4):473–511, apr 2004.

T. A. Johansen, A. Cristofaro, K. Sorensen, J. M. Hansen, and T. I. Fossen. On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. Institute of Electrical & Electronics Engineers (IEEE), jun 2015.

E. Johnson, A. Calise, and J. Corban. A six degree-of-freedom adaptive flight control architecture for trajectory following. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Aug. 2002. doi: 10.2514/6.2002-4776.

S. P. Jones and J. D. Delaurier. Aerodynamic estimation techniques for aerostats and airships. *Journal of Aircraft*, 20(2):120–126, feb 1983.

T. Keviczky and G. Balas. Software enabled flight control using receding horizon techniques. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Aug. 2003. doi: 10.2514/6.2003-5671.

H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, 3 edition, 2000.

G. A. Khoury, editor. *Airship Technology (Cambridge Aerospace Series)*. Cambridge University Press, 2012.

J.-H. Kim, S. Sukkarieh, and S. Wishart. Real-time navigation, guidance, and control of a UAV using low-cost sensors. In *Springer Tracts in Advanced Robotics*, pages 299–309. Springer Science Business Media, 2006.

J. Koschorke, W. Falkena, E.-J. V. Kampen, and Q. P. Chu. Time delayed incremental nonlinear control. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, August 2013. doi: 10.2514/6.2013-4929.

P. Kungl, M. Schlenker, D.-A. Wimmer, and B. H. Kröplin. Instrumentation of remote controlled airship lotte for in-flight measurements. *Aerospace Science and Technology*, 8(7): 599–610, oct 2004.

H. Lamb. The inertia coefficients of an ellipsoid moving in fuid, oct 1918. URL http://resolver.tudelft.nl/uuid:03f8585f-f6b0-4423-b0ad-784d778107e0.

S. H. Lane and R. F. Stengel. Flight control design using non-linear inverse dynamics. *Automatica*, 24(4):471–483, jul 1988.

A. M. Lekkas and T. I. Fossen. Line-of-sight path following of underactuated marine craft. In *PROCEEDINGS OF THE 6th IFAC*. Institute of Electrical and Electronics Engineers (IEEE), jan 2003.

Y. Li. *Dynamics Modeling and Simulation of Flexible Airships*. PhD thesis, McGill University, 2008.

Y. Liu, Z. Pan, D. Stirling, and F. Naghdy. Control of autonomous airship. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Institute of Electrical & Electronics Engineers (IEEE), dec 2009.

A. S. Marton, L. A. C. d. O. Nogueira, A. Fioravanti, and E. C. de Paiva. Comparative study for robotic airship state estimation with LPF, EKF and UKF. In *2019 SBAI - XIV Simpósio Brasileiro de Automação Inteligente*, oct 2019.

A. S. Marton, A. R. Fioravanti, J. R. Azinheira, and E. C. de Paiva. Hybrid model-based and data-driven wind velocity estimator for an autonomous robotic airship. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 42(3), Feb. 2020. doi: 10.1007/s40430-020-2215-8.

D. McLean. *Automatic flight control systems*. Prentice-Hall, 1990.

J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk. Comprehensive simulation of quadrotor uavs using ros and gazebo. In *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, 2012.

T. Moore and D. Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

A. Moutinho and J. R. Azinheira. Hover Stabilization of an Airship using Dynamic Inversion. *IFAC Proceedings Volumes*, 39(15):193–198, 2006.

A. Moutinho, L. Mirisola, J. Azinheira, and J. Dias. *Projetc DIVA: Guidance and Vision Surveillance Techniques for an Autonomous Airship*, chapter 2. Nova Science Publishers, Inc, 2007.

A. Moutinho, J. R. Azinheira, E. C. de Paiva, and S. S. Bueno. Airship robust path-tracking: A tutorial on airship modelling and gain-scheduling control design. *Control Engineering Practice*, 50:22–36, May 2016.

A. B. Moutinho. *Modeling and nonlinear control for airship autonomous flight*. PhD thesis, Instituto Superior Técnico, 2007.

J. Mueller, M. Paluszek, and Y. Zhao. Development of an aerodynamic model and control law design for a high altitude airship. In *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*. American Institute of Aeronautics and Astronautics (AIAA), sep 2004. doi: 10.2514/6.2004-6479. URL http://dx.doi.org/10.2514/6.2004-6479.

R. M. Murray. Trajectory generation for a towed cable system using differential flatness. *IFAC Proceedings Volumes*, 29(1):2792–2797, June 1996. doi: 10.1016/s1474-6670(17)58099-4.

S.-M. Oh. Multisensor fusion for autonomous UAV navigation based on the unscented kalman filter with sequential measurement updates. In *2010 IEEE Conference on Multisensor Fusion and Integration*. IEEE, sep 2010.

N. Olgac and R. Sipahi. An exact method for the stability analysis of time-delayed linear time-invariant (lti) systems. *IEEE Transactions on Automatic Control*, 47(5):793–797, May 2002. doi: 10.1109/TAC.2002.1000275.

M. W. Oppenheimer, D. B. Doman, and M. A. Bolender. Control allocation for over-actuated systems. In *2006 14th Mediterranean Conference on Control and Automation*. IEEE, jun 2006.

E. Paiva, F. Benjovengo, S. Bueno, and P. Ferreira. Sliding mode control approaches for an autonomous unmanned airship. In *18th AIAA Lighter-Than-Air Systems Technology Conference*. American Institute of Aeronautics and Astronautics (AIAA), May 2009.

S. Park, J. Deyst, and J. How. A new nonlinear guidance logic for trajectory tracking. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, Aug. 2004. doi: 10.2514/6.2004-4900.

D. Patino, L. Solaque, S. Lacroix, and A. Gauthier. Estimation of the aerodynamical parameters of an experimental airship. In *IEEE IROS 2005 Workshop on Robot Vision for Space Applications*, pages 22–28, aug 2005.

J. Perry, A. Mohamed, B. Johnson, and R. Lind. Estimating angle of attack and sideslip under high dynamics on small uavs. In *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, pages 1165–1173. Institute of Electrical and Electronics Engineers (IEEE), sep 2008.

M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

M. Rathinam and R. Murray. Configuration flatness of lagrangian systems underactuated by one control. In *Proceedings of 35th IEEE Conference on Decision and Control*. IEEE, 2002. doi: 10.1109/cdc.1996.572795.

M. B. Rhudy, Y. Gu, J. N. Gross, and H. Chao. Onboard wind velocity estimation comparison for unmanned aircraft systems. *IEEE Transactions on Aerospace and Electronic Systems*, 53 (1):55–66, Feb. 2017.

M. Rueda, L. Mirisola, L. Nogueira, G. Fonseca, J. Ramos, M. Koyama, J. Azinheira, R. Carvalho, S. Bueno, and E. de Paiva. Uma infraestrutura, de hardware, software e comunicação para a robotização de plataformas rádio - controladas: Aplicação a um dirigível robótico. In *2017 SBAI - XIII Simpósio Brasileiro de Automação Inteligente*, oct 2017.

R. Rysdyk. UAV path following for constant line-of-sight. In *2nd AIAA "Unmanned Unlimited" Conf. and Workshop & Exhibit*. American Institute of Aeronautics and Astronautics (AIAA), sep 2003.

I. Samy, I. Postlethwaite, D.-W. Gu, and J. Green. Neural-network-based flush air data sensing system demonstrated on a mini air vehicle. *Journal of Aircraft*, 47(1):18–31, jan 2010.

S. Shen, L. Liu, B. Huang, X. Lin, W. Lan, and H. Jin. Wind speed estimation and station-keeping control for stratospheric airships with extended kalman filter. In *Proceedings of the 2015 Chinese Intelligent Automation Conference*, pages 145–157. Springer Science Business Media, 2015.

S. Sieberling, Q. P. Chu, and J. A. Mulder. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. *Journal of Guidance, Control, and Dynamics*, 33(6):1732–1742, nov 2010.

P. Simplício, M. Pavel, E. van Kampen, and Q. Chu. An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion. *Control Engineering Practice*, 21(8):1065–1077, aug 2013.

E. J. J. Smeur, Q. Chu, and G. C. H. E. de Croon. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, mar 2016.

P. Smith. A simplified approach to nonlinear dynamic inversion based flight control. In *23rd Atmospheric Flight Mechanics Conference*. American Institute of Aeronautics and Astronautics (AIAA), aug 1998.

P. G. Thomasson. Equations of motion of a vehicle in a moving fluid. *Journal of Aircraft*, 37 (4):630–639, July 2000. doi: 10.2514/2.2645.

A. Tsourdos, B. White, and M. Shanmugavel. *Cooperative Path Planning of Unmanned Aerial Vehicles*. Wiley, 2010.

R. van 't Veld, E.-J. V. Kampen, and Q. P. Chu. Stability and robustness analysis and improvements for incremental nonlinear dynamic inversion control. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, January 2018. doi: 10.2514/6.2018-1127.

H. Vieira, A. Marton, E. C. de Paiva, A. Fioravanti, and S. S. Bueno. Controle de um dirigível robótico autônomo de propulsão quádrupla utilizando modos deslizantes. In *2017 SBAI - XIII Simpósio Brasileiro de Automação Inteligente*, oct 2017.

H. d. S. Vieira. *Unified design of nonlinear controllers for a robotic electric propulsion airship*. PhD thesis, University of Campinas, Aug. 2019.

H. Voos. Nonlinear control of a quadrotor micro-UAV using feedback-linearization. In *2009 IEEE International Conference on Mechatronics*. Institute of Electrical & Electronics Engineers (IEEE), 2009.

K. Walton and J. E. Marshall. Direct method for tds stability analysis. *IEE Proceedings D - Control Theory and Applications*, 134(2):101–107, March 1987. doi: 10.1049/ip-d.1987.0018.

E. Wan and R. V. D. Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*. IEEE.

X. Wang, E.-J. V. Kampen, Q. P. Chu, and P. Lu. Stability analysis for incremental nonlinear dynamic inversion control. In *2018 AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, jan 2018.

X. Wang, E.-J. v. Kampen, Q. Chu, and P. Lu. Incremental sliding-mode fault-tolerant flight control. *Journal of Guidance, Control, and Dynamics*, 42(2):244–259, 2019a. doi: 10.2514/1.G003497.

X. Wang, E.-J. van Kampen, Q. Chu, and P. Lu. Stability analysis for incremental nonlinear dynamic inversion control. *Journal of Guidance, Control, and Dynamics*, 42(5):1116–1129, 2019b. doi: 10.2514/1.G003791.

Y. Wu, M. Zhu, Z. Zuo, and Z. Zheng. Adaptive trajectory tracking control of a high altitude unmanned airship. *Journal of Computers*, 7(11), nov 2012.

H. Zaki, G. Alcan, and M. Unel. Robust trajectory control of an unmanned aerial vehicle using acceleration feedback. *International Journal of Mechatronics and Manufacturing Systems*, 12(3/4):298, 2019.

F. D. Zanoni and E. A. de Barros. A real-time navigation system for autonomous underwater vehicle. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 37(4): 1111–1127, Aug. 2014.

Z. Zheng and L. Xie. Finite-time path following control for a stratospheric airship with input saturation and error constraint. *International Journal of Control*, 92(2):368–393, Aug. 2017. doi: 10.1080/00207179.2017.1357839.

Z. Zheng, M. Zhu, D. Shi, and Z. Wu. Hovering control for a stratospheric airship in unknown wind. In *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics (AIAA), jan 2014.

Z. Zheng, K. Yan, S. Yu, B. Zhu, and M. Zhu. Path following control for a stratospheric airship with actuator saturation. *Transactions of the Institute of Measurement and Control*, 39(7): 987–999, Jan. 2016.

M. Zhu, S. Yu, and Z. Zheng. Model predictive control for path following of stratospheric airship with magnitude and rate constrains of rudder. In *The 27th Chinese Control and Decision Conference (2015 CCDC)*. Institute of Electrical & Electronics Engineers (IEEE), may 2015.

A INPUT MATRICES FOR INCREMENTAL CONTROLLERS

The set of input matrix $\mathbf{B}_{v_j}$ used during simulation are given below, where $j \in \mathbf{J} = \{0, 2, 4, 6, 8, 10\}$m/s is the corresponding true airspeed:

$$\mathbf{B}_{v_0} = \begin{bmatrix} 0.0484 & 0.0145 & -0.9883 & -1.0439 \\ -0.0499 & -1.9149 & 0.0021 & -0.0115 \\ -0.0762 & 0.0050 & 1.5685 & -0.2571 \end{bmatrix}, \ \mathbf{B}_{v_2} = \begin{bmatrix} 0.5061 & 0.2220 & -0.9935 & -1.0187 \\ -0.4785 & -1.9250 & 0.0021 & -0.0760 \\ -0.7377 & 0.0561 & 1.5767 & -0.2508 \end{bmatrix}$$

$$\mathbf{B}_{v_4} = \begin{bmatrix} 1.4817 & 0.6578 & -0.8613 & -0.7123 \\ -1.2855 & -1.6690 & 0.0018 & -0.1587 \\ -2.0089 & 0.1632 & 1.3669 & -0.1753 \end{bmatrix}, \ \mathbf{B}_{v_6} = \begin{bmatrix} 2.5861 & 0.7457 & -0.8289 & -0.6671 \\ -2.5126 & -1.6066 & 0.0017 & -0.1838 \\ -3.8591 & 0.1848 & 1.3155 & -0.1641 \end{bmatrix}$$

$$\mathbf{B}_{v_8} = \begin{bmatrix} 4.4470 & 2.1247 & -0.1571 & 0.0292 \\ -4.1120 & -0.3046 & 0.0003 & -0.2412 \\ -6.3595 & 0.5234 & 0.2494 & 0.0074 \end{bmatrix}, \ \mathbf{B}_{v_{10}} = \begin{bmatrix} 6.3148 & 2.5174 & -0.1118 & 0.0517 \\ -6.1387 & -0.2165 & 0.0002 & -0.3112 \\ -9.4277 & 0.6201 & 0.1774 & 0.0130 \end{bmatrix}$$

Considering this given set of input matrices, the index of the actual chosen matrix $i = \{1, 2, 3, 4, 5, 6\}$, the vector of available airspeeds $\mathbf{J}$ and the measured airspeed by the pitot tube V_{pitot}, the input matrix $\mathbf{B}_v$ is computed by the Algorithm 1.

Algorithm 1: Algorithm for updating input matrix used in the INDI control loop

```
Result: Chosen matrix $\mathbf{B}_v$ and corresponding index in the set $i$
Data: Measured airspeed $V_{pitot}$, last index $i$, vector $\mathbf{J}$ and the set of matrices $\mathbf{B}_{v_j}$
$i_{previous} \leftarrow \max(i-1,1)$
$i_{next} \leftarrow \min(i+1,6)$
$\Delta_{actual} \leftarrow |V_{pitot} - \mathbf{J}[i])|$
$\Delta_{previous} \leftarrow |V_{pitot} - \mathbf{J}[i_{previous}]|$
$\Delta_{next} \leftarrow |V_{pitot} - \mathbf{J}[i_{next}]|$
if $\Delta_{previous} < \Delta_{next}$ then
    $\Delta_{closest} \leftarrow \Delta_{previous}$
    $i_{closest} \leftarrow i_{previous}$
else
    $\Delta_{closest} \leftarrow \Delta_{next}$
    $i_{closest} \leftarrow i_{next}$
end
$k \leftarrow \mathbf{J}[i_{closest}]$
$j \leftarrow \mathbf{J}[i]$
$c_{actual} \leftarrow \Delta_{actual}/(\Delta_{actual} + \Delta_{closest})$
$c_{closest} \leftarrow \Delta_{closest}/(\Delta_{actual} + \Delta_{closest})$
$\mathbf{B}_v \leftarrow c_{actual} \cdot \mathbf{B}_{v_j} + c_{closest} \cdot \mathbf{B}_{v_k}$
if $\Delta_{actual} > \Delta_{previous}$ then
    $i \leftarrow i_{previous}$
end
if $\Delta_{actual} > \Delta_{next}$ then
    $i \leftarrow i_{next}$
end
```

B COVARIANCE MATRICES FOR MODEL BASED WIND ESTIMATORS

The covariance matrices used for each Model-based approach are shown below:

$$\begin{aligned}
&\text{Cho2011}:\\
&\mathbf{R} = \begin{bmatrix}163.84\end{bmatrix},\\
&\mathbf{Q} = \text{diag}\left(\begin{bmatrix}10^{-3} & 10^{-4} & 5\cdot 10^{-6}\end{bmatrix}\right).
\end{aligned}$$

$$\begin{aligned}
&\text{EKF}:\\
&\mathbf{R} = \text{diag}\left(\begin{bmatrix}40.96 & 40.96 & 40.96\end{bmatrix}\right),\\
&\mathbf{Q} = \text{diag}\left(\begin{bmatrix}10^{-4} & 10^{-4} & 5\cdot 10^{-7}\end{bmatrix}\right).
\end{aligned}$$

$$\begin{aligned}
&\text{Hybrid}:\\
&\mathbf{R} = \text{diag}\left(\begin{bmatrix}10.24 & 10.24 & 10.24 & 10.24 & 10.24 & 10.24\end{bmatrix}\right),\\
&\mathbf{Q} = \text{diag}\left(\begin{bmatrix}10^{-4} & 10^{-4} & 5\cdot 10^{-7}\end{bmatrix}\right).
\end{aligned}$$

C TRAINING TRAJECTORIES FOR WIND ESTIMATORS

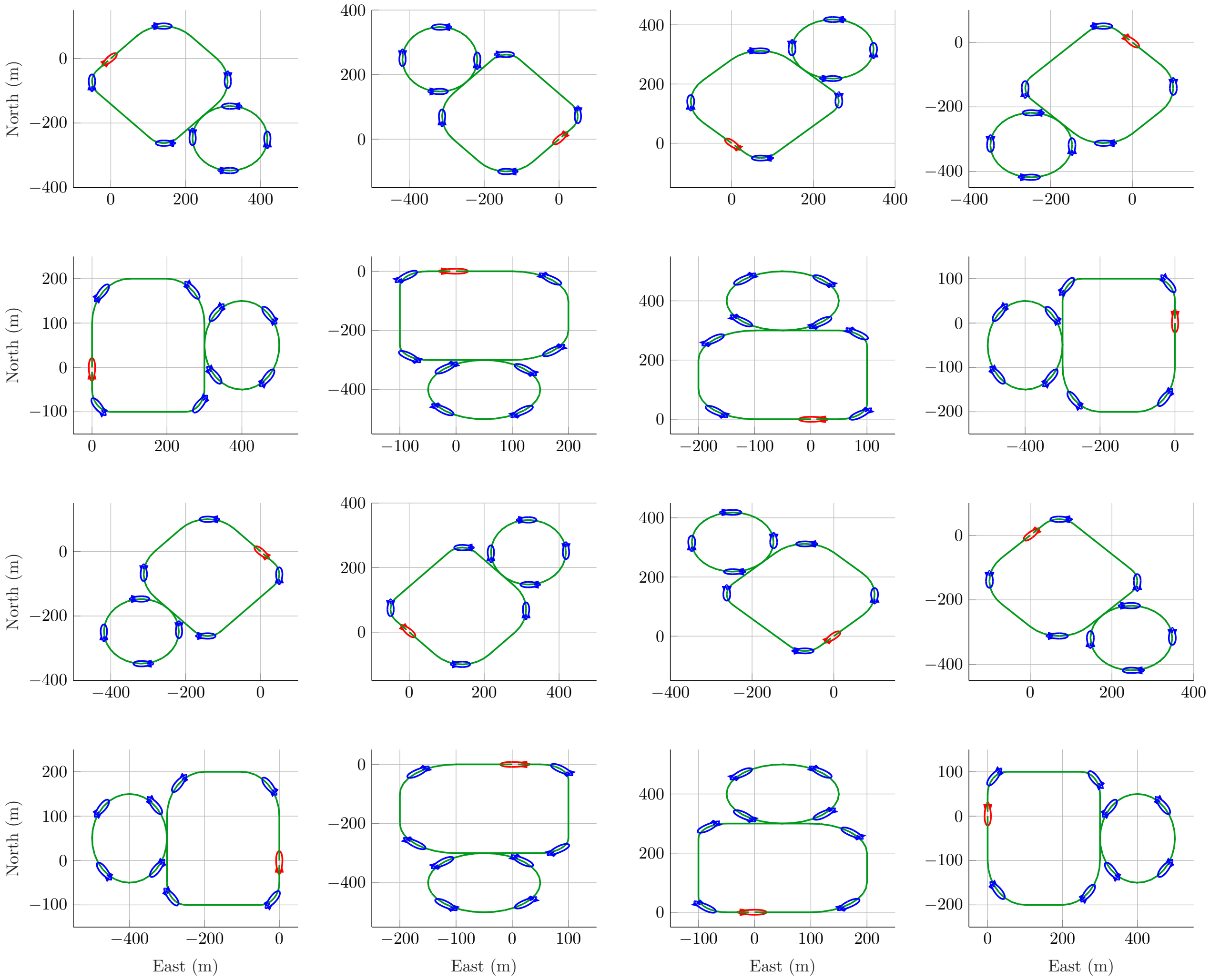


Figure C.1. Simulation trajectories used for the NN training task