UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Development of an autonomous unmanned aerial vehicle – specification of a fixed-wing vertical takeoff and landing aircraft

Natássya Barlate Floro da Silva

Tese de Doutorado do Programa de Pós-Graduação em Ciências de
Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Natássya Barlate Floro da Silva

Development of an autonomous unmanned aerial vehicle – specification of a fixed-wing vertical takeoff and landing aircraft

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco

Co-advisor: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
May 2018

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

S586d Silva, Natássya Barlate Floro
Development of an autonomous unmanned aerial
vehicle - specification of a fixed-wing vertical
takeoff and landing aircraft / Natássya Barlate
Floro Silva; orientadora Kalinka Regina Lucas
Jaquie Castelo Branco; coorientador Denis Fernando
Wolf. -- São Carlos, 2018.
169 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2018.

1. AVALON. 2. Tailsitter. 3. Sensor fusion. 4.
UAV Control. 5. Path-following algorithm. I.
Branco, Kalinka Regina Lucas Jacquie Castelo,
orient. II. Wolf, Denis Fernando, coorient. III.
Título.

Natássya Barlate Floro da Silva

**Desenvolvimento de um veículo aéreo não tripulado
autônomo – especificação de uma aeronave asa-fixa capaz
de decolar e aterrissar verticalmente**

Tese apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Doutora em Ciências – Ciências de Computação e
Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientadora: Profa. Dra. Kalinka Regina Lucas Jaquie
Castelo Branco

Coorientador: Prof. Dr. Denis Fernando Wolf

**USP – São Carlos
Maio de 2018**

ACKNOWLEDGEMENTS

First, I would like to thank my thesis advisor Prof. Kalinka Branco for supporting me throughout all challenges faced during this doctorate and for offering all the opportunities that assisted me in becoming a researcher.

I would also like to thank my laboratory colleagues, especially Mariana Rodrigues, Rayner Pires and Rogers Cristo. A very especial gratitude goes out to Daniel Pigatto and João Fontes, whose encouragements and efforts were essential for the conclusion of this thesis.

I would like to thank all undergraduate students who collaborated with my project, particularly Guilherme Pelizer, Luiz Henrique Patire and Alex Sartin.

I would like to thank Daniel Wilson and Kaushi Madugalle for their assistance during the research internship abroad.

I would like to thank my parents, Adalberto and Vania, for their approval and support during all these years. I would also like to thank my siblings Anna and Alexey for their companionship during leisure time and all my family for their optimism. Moreover, I would like to thank Eduardo Andrade for his support.

I would like to thank the EESC USP Aerodesign for their assistance in building and testing AVALON prototype, especially Rodrigo Marin, Vinicius Pereira and Lucas Soares.

I would also like to thank and acknowledge ICMC and USP for the infrastructure, and FAPESP for the financial support from process 2012/13641-1 and BEPE process 2013/18562-5.

“The more we learn about the world, and the deeper our learning, the more conscious, specific, and articulate will be our knowledge of what we do not know, our knowledge of our ignorance.”
(Karl Popper)

ABSTRACT

SILVA, N. B. F. **Development of an autonomous unmanned aerial vehicle – specification of a fixed-wing vertical takeoff and landing aircraft.** 2018. 169 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Several configurations of Unmanned Aerial Vehicles (UAVs) were proposed to support different applications. One of them is the tailsitter, a fixed-wing aircraft that takes off and lands on its own tail, with the high endurance advantage from fixed-wing aircraft and, as helicopters and multicopters, not requiring a runway during takeoff and landing. However, a tailsitter has a complex operation with multiple flight stages, each one with its own particularities and requirements, which emphasises the necessity of a reliable autopilot for its use as a UAV. The literature already introduces tailsitter UAVs with complex mechanisms or with multiple counter-rotating propellers, but not one with only one propeller and without auxiliary structures to assist in the takeoff and landing. This thesis presents a tailsitter UAV, named AVALON (Autonomous VerticAl takeOff and laNding), and its autopilot, composed of 3 main units: Sensor Unit, Navigation Unit and Control Unit. In order to choose the most appropriate techniques for the autopilot, different solutions are evaluated. For Sensor Unit, Extended Kalman Filter and Unscented Kalman Filter estimate spatial information from multiple sensors data. Lookahead, Pure Pursuit and Line-of-Sight, Nonlinear Guidance Law and Vector Field path-following algorithms are extended to incorporate altitude information for Navigation Unit. In addition, a structure based on classical methods with decoupled Proportional-Integral-Derivative controllers is compared to a new control structure based on dynamic inversion. Together, all these techniques show the efficacy of AVALON's autopilot. Therefore, AVALON results in a small electric tailsitter UAV with a simple design, with only one propeller and without auxiliary structures to assist in the takeoff and landing, capable of executing all flight stages.

Keywords: AVALON, Tailsitter, Sensor fusion, UAV Control, Path-following algorithm.

RESUMO

SILVA, N. B. F. **Desenvolvimento de um veículo aéreo não tripulado autônomo – especificação de uma aeronave asa-fixa capaz de decolar e aterrissar verticalmente.** 2018. 169 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

Diversas configurações de Veículos Aéreos Não Tripulados (VANTs) foram propostas para serem utilizadas em diferentes aplicações. Uma delas é o *tailsitter*, uma aeronave de asa fixa capaz de decolar e pousar sobre a própria cauda. Esse tipo de aeronave apresenta a vantagem de aeronaves de asa fixa de voar sobre grandes áreas com pouco tempo e bateria e, como helicópteros e multicópteros, não necessita de pista para decolar e pousar. Porém, um *tailsitter* possui uma operação complexa, com múltiplos estágios de voo, cada um com suas peculiaridades e requisitos, o que enfatiza a necessidade de um piloto automático confiável para seu uso como um VANT. A literatura já introduz VANTs *tailsitters* com mecanismos complexos ou múltiplos motores contra-rotativos, mas não com apenas um motor e sem estruturas para auxiliar no pouso e na decolagem. Essa tese apresenta um VANT *tailsitter*, chamado AVALON (*Autonomous VerticAl takeOff and laNding*), e seu piloto automático, composto por 3 unidades principais: Unidade Sensorial, Unidade de Navegação e Unidade de Controle. Diferentes soluções são avaliadas para a escolha das técnicas mais apropriadas para o piloto automático. Para a Unidade Sensorial, *Extended Kalman Filter* e *Unscented Kalman Filter* estimam a informação espacial de múltiplos dados de diversos sensores. Os algoritmos de seguimento de trajetória *Lookahead*, *Pure Pursuit* and *Line-of-Sight*, *Nonlinear Guidance Law* e *Vector Field* são estendidos para considerar a informação da altitude para a Unidade de Navegação. Além do mais, uma estrutura baseada em métodos clássicos com controladores Proporcional-Integral-Derivativo desacoplados é comparada a uma nova estrutura de controle baseada em dinâmica inversa. Juntas, todas essas técnicas demonstram a eficácia do piloto automático do AVALON. Portanto, AVALON resulta em um VANT *tailsitter* pequeno e elétrico, com um design simples, apenas um motor e sem estruturas para auxiliar o pouso e a decolagem, capaz de executar todos os estágios de voo.

Palavras-chave: AVALON, Tailsitter, Fusão de sensores, Controle para VANTs, Algoritmos de seguimento de trajetória.

LIST OF FIGURES

Figure 1 – Fixed-wing UAV example with indicated control surfaces.	37
Figure 2 – Multirotor UAV example, also known as quadrotor.	39
Figure 3 – AVALON operation divided in five flight stages: takeoff, TVH, horizontal flight, THV and landing.	48
Figure 4 – AVALON design in different perspectives: isometric view in the top right, top view in the top left, front view in the bottom left and right view in the bottom right.	49
Figure 5 – Inertial frame and body frame with the angles roll ϕ , pitch θ , and yaw ψ	50
Figure 6 – Geometric model of AVALON in Tornado.	55
Figure 7 – Different perspective views of the geometric model of AVALON in Tornado.	55
Figure 8 – C_L coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.	56
Figure 9 – C_L coefficient for a fixed sideslip of 0° and intervals of α and $\ \mathbf{v}_{rel}\ $ resulted from Tornado's Vortex Lattice Method.	57
Figure 10 – AVALON's autopilot structure divided according to its main functions.	57
Figure 11 – Thesis structure based on the autopilot units.	59
Figure 12 – Positions estimates from EKF and UKF in the inertial frame from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The path is defined by 6 waypoints (red points), whose sequence is represented by the red numbers.	71
Figure 13 – EKF and UKF north position errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	72
Figure 14 – EKF and UKF east position errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	73

Figure 15 – EKF and UKF up position errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	73
Figure 16 – EKF and UKF north velocity errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	74
Figure 17 – EKF and UKF east velocity errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	74
Figure 18 – EKF and UKF up velocity errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	75
Figure 19 – EKF and UKF roll errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	75
Figure 20 – EKF and UKF pitch errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	76
Figure 21 – EKF and UKF yaw errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	76
Figure 22 – Angle of attack errors obtained from the estimates of EKF and UKF from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage. . . .	77
Figure 23 – Angle of sideslip errors obtained from the estimates of EKF and UKF from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage. . . .	78

Figure 24 – Velocity magnitude errors obtained from the estimates of EKF and UKF from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.	78
Figure 25 – Positions errors mean and confidence intervals for EKF and UKF and 30 replications of the entire AVALON’s operation with different configurations of sensor noise.	79
Figure 26 – Velocities errors mean and confidence interval for EKF and UKF and 30 replications of the entire AVALON’s operation with different configurations of sensor noise.	80
Figure 27 – Attitude angles errors mean and confidence interval for EKF and UKF and 30 replications of the entire AVALON’s operation with different configurations of sensor noise.	80
Figure 28 – Sensor metric mean and confidence interval for EKF and UKF and 30 replications of the entire AVALON’s operation with different configurations of sensor noise.	81
Figure 29 – Inertial, intermediate and path frames with their rotations along the Z_i of ζ_ψ and along the Y' of ζ_θ	87
Figure 30 – The error vector $\mathbf{e}$ in the path frame and its respective components e_x , e_y and e_z	88
Figure 31 – Positions in the inertial frame resulted from all path-following algorithms for the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit. The path is defined by 6 waypoints (red points), whose sequence is represented by the red numbers.	96
Figure 32 – North positions of all path-following algorithms resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	97
Figure 33 – East positions of all path-following algorithms resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	97
Figure 34 – Up positions of all path-following algorithms resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	98
Figure 35 – Error metric mean and confidence interval from all path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.	99

Figure 36 – Mean and confidence interval of the error metric difference among NLGL and the other path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.	100
Figure 37 – Effort metric mean and confidence interval from all path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.	101
Figure 38 – Metric mean and confidence interval from all path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.	101
Figure 39 – Mean and confidence interval of the metric difference among NLGL and the other path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.	102
Figure 40 – Reference and resulted pitch angle from trapezoidal velocity profile during TVH with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	103
Figure 41 – Reference and resulted pitch velocity from trapezoidal velocity profile during TVH with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	104
Figure 42 – Reference and resulted pitch acceleration from trapezoidal velocity profile during TVH with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	104
Figure 43 – Reference and resulted pitch angle from trapezoidal velocity profile during THV with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	105
Figure 44 – Reference and resulted pitch velocity from trapezoidal velocity profile during THV with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	105
Figure 45 – Reference and resulted pitch acceleration from trapezoidal velocity profile during THV with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	106
Figure 46 – Reference and resulted velocity up from PID controller during landing with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.	107
Figure 47 – Structure of vertical flight and transition control of the Decoupled PID.	111
Figure 48 – Horizontal flight control structure of the Decoupled PID.	112
Figure 49 – DI with PID structure for attitude control.	117
Figure 50 – Structure of the takeoff and landing feedback controllers for DI with PID.	118
Figure 51 – Structure of the transitions feedback controllers for DI with PID.	118

Figure 52 – Structure of the horizontal feedback controllers for DI with PID.	119
Figure 53 – Positions in the inertial frame resulted from Decoupled PID and DI with PID structures for the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The path is defined by 6 waypoints (red points), whose sequence is represented by the red numbers.	123
Figure 54 – Roll angle for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	124
Figure 55 – Pitch angle for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	124
Figure 56 – Yaw angle for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	125
Figure 57 – Angle of attack for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	125
Figure 58 – Sideslip angle for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	126
Figure 59 – Velocity magnitude for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	126

Figure 60 – Throttle command for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	127
Figure 61 – Ailerons deflection for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	128
Figure 62 – Elevators deflection for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	128
Figure 63 – Rudders deflection command for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.	129
Figure 64 – Number of successful and failed replications in each flight stage for Decoupled PID from the simulation results with different configuration of sensor noise.	129
Figure 65 – Number of successful and failed replications in each flight stage for DI with PID from the simulation results with different configuration of sensor noise.	130
Figure 66 – Error metric mean and confidence interval for Decoupled PID and DI with PID from replications with different configurations of sensor noise.	131
Figure 67 – Effort metric mean and confidence interval for Decoupled PID and DI with PID from replications with different configurations of sensor noise.	131
Figure 68 – Metric mean and confidence interval for Decoupled PID and DI with PID from replications with different configurations of sensor noise. . . .	132
Figure 69 – Number of successful and failed replications in each flight stage for Decoupled PID from the simulation results with different intensities of wind disturbance.	133
Figure 70 – Number of successful and failed replications in each flight stage for DI with PID from the simulation results with different intensities of wind disturbance.	133

Figure 71 – Error metric mean and confidence interval for Decoupled PID and DI with PID from replications with different intensities of wind disturbance.	134
Figure 72 – Effort metric mean and confidence interval for Decoupled PID and DI with PID from replications with different intensities of wind disturbance.	134
Figure 73 – Metric mean and confidence interval for Decoupled PID and DI with PID from replications with different intensities of wind disturbance. . .	135
Figure 74 – AVALON prototype.	138
Figure 75 – Interfaces and communication protocols among the main processing board Odroid XU4, the auxiliary board Arduino Leonardo, the radio receiver AR8000, the ESC EFLA 1080B, the servomotors Hobbico CS-12MG, the IMU ADIS16407, the GPS LS20031, the sonar MB1242 and the airspeed sensor MS4525DO.	139
Figure 76 – Autopilot container.	141
Figure 77 – Sequence of images from two flight experiments with a human pilot controlling AVALON prototype.	142
Figure 78 – C_{Lift} coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.	159
Figure 79 – C_{Lift} coefficient for a fixed sideslip of 0° and intervals of α and $\ \mathbf{v}_{rel}\ $ resulted from Tornado's Vortex Lattice Method.	160
Figure 80 – C_{Drag} coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.	160
Figure 81 – C_{Drag} coefficient for a fixed sideslip of 0° and intervals of α and $\ \mathbf{v}_{rel}\ $ resulted from Tornado's Vortex Lattice Method.	161
Figure 82 – C_Y coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.	161
Figure 83 – C_Y coefficient for a fixed sideslip of 0° and intervals of α and $\ \mathbf{v}_{rel}\ $ resulted from Tornado's Vortex Lattice Method.	162
Figure 84 – C_L coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.	162
Figure 85 – C_L coefficient for a fixed sideslip of 0° and intervals of α and $\ \mathbf{v}_{rel}\ $ resulted from Tornado's Vortex Lattice Method.	163
Figure 86 – C_M coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.	163
Figure 87 – C_M coefficient for a fixed sideslip of 0° and intervals of α and $\ \mathbf{v}_{rel}\ $ resulted from Tornado's Vortex Lattice Method.	164
Figure 88 – C_N coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.	164
Figure 89 – C_N coefficient for a fixed sideslip of 0° and intervals of α and $\ \mathbf{v}_{rel}\ $ resulted from Tornado's Vortex Lattice Method.	165

LIST OF ALGORITHMS

Algorithm 1 – Lookahead algorithm.	89
Algorithm 2 – Nonlinear Guidance Law algorithm.	91
Algorithm 3 – Pure Pursuit and Line-of-Sight algorithm.	92
Algorithm 4 – Vector Field algorithm.	93

LIST OF TABLES

Table 1 – Main characteristics from related work.	45
Table 2 – AVALON’s physical parameters.	49
Table 3 – Extended Kalman Filter Algorithm.	65
Table 4 – Unscented Kalman Filter Algorithm	68
Table 5 – Standard deviation of accelerometers, gyroscopes, magnetometers, barometer and GPS used in the simulation environment to generate sensor noise.	70
Table 6 – Parameters of all extended path-following algorithms resulted from GA and SQP optimisations with the objective function (5.22).	95
Table 7 – Expected errors and efforts for each flight stage used in the metrics $M_{error,fs}$ defined by (6.21) and $M_{effort,fs}$ defined by (6.22).	120
Table 8 – Controller gains of Decoupled PID and DI with PID structures resulted from GA and SQP optimisations with the objective function (6.20).	122

LIST OF ABBREVIATIONS AND ACRONYMS

ANAC	Agência Nacional de Aviação
ANATEL	Agência Nacional de Telecomunicações
ASCII	American Standard Code for Information Interchange
AVALON	Autonomous VerticAl takeOff and laNding
BLOS	Beyond-Line-Of-Sight
CG	Centre of Gravity
DECEA	Departamento de Controle do Espaço Aéreo
DI	Dynamic Inversion
EKF	Extended Kalman Filter
ESC	Electric Speed Controller
GA	Genetic Algorithm
GCS	Ground Control Station
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
INDI	Incremental Nonlinear Dynamic Inversion
LOS	Line-Of-Sight
LQR	Linear Quadratic Regulator
MAC	Mean Aerodynamic Chord
NDI	Nonlinear Dynamic Inversion
NLGL	NonLinear Guidance Law
NMEA	National Marine Electronics Association
P	Proportional
PD	Proportional-Derivative
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PLOS	Pure Pursuit and Line-Of-Sight
PWM	Pulse-Width Modulation
RPA	Remotely Piloted Aircraft
RPAS	Remotely Piloted Aircraft System
RPM	Revolutions Per Minute

SPI	Serial Peripheral Interface
SQP	Sequential Quadratic Programming
THV	Transition from Horizontal to Vertical position
TTL	Transistor-Transistor Logic
TVH	Transition from Vertical to Horizontal position
UART	Universal Asynchronous Receiver-Transmitter
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
USB	Universal Serial Bus
VLOS	Visual Line-Of-Sight
VTOL	Vertical TakeOff and Landing
WGS	World Geodetic System
XPS	Extruded polystyrene

CONTENTS

1	INTRODUCTION	31
1.1	Motivation	32
1.2	Research question and hypothesis	33
1.3	Objectives	33
1.4	Contributions	34
1.5	Thesis overview	34
2	UNMANNED AERIAL VEHICLES	35
2.1	Chapter overview	35
2.2	Unmanned aircraft system	35
2.2.1	<i>Fixed-wing aircraft</i>	37
2.2.2	<i>Vertical takeoff and landing aircraft</i>	38
2.3	Related work	39
2.4	Final remarks	46
3	AVALON	47
3.1	Chapter overview	47
3.2	Physical design	47
3.3	Kinematic and dynamic models	48
3.3.1	<i>Aerodynamic derivatives from Tornado</i>	54
3.4	Autopilot	57
3.5	Final remarks	58
4	SENSOR FUSION	61
4.1	Chapter overview	61
4.2	State estimation from sensors	61
4.3	Extended Kalman filter	64
4.4	Unscented Kalman filter	66
4.5	Simulation and comparison results	69
4.6	Final remarks	81
5	NAVIGATION	83
5.1	Chapter overview	83
5.2	Proportional-integral-derivative controller	84

5.3	Horizontal flight	85
5.3.1	<i>Path-following problem</i>	86
5.3.2	<i>Path-following algorithms</i>	86
5.3.2.1	<i>Lookahead</i>	88
5.3.2.2	<i>Nonlinear guidance law</i>	89
5.3.2.3	<i>Pure pursuit and line-of-sight</i>	90
5.3.2.4	<i>Vector field</i>	90
5.3.3	<i>Simulation and comparison results</i>	93
5.4	Transitions	102
5.5	Landing	106
5.6	Final remarks	107
6	CONTROL	109
6.1	Chapter overview	109
6.2	Velocity controller	110
6.3	Decoupled PID structure	110
6.4	DI with PID structure	112
6.4.1	<i>Computed torque</i>	112
6.4.2	<i>Incremental Nonlinear Dynamic Inversion</i>	114
6.4.3	<i>Attitude control</i>	116
6.5	Simulation and comparison results	119
6.6	Final remarks	135
7	AVALON PROTOTYPE	137
7.1	Chapter overview	137
7.2	Physical structure	137
7.3	Autopilot hardware and software	139
7.4	Experimental flights	141
7.5	Final remarks	141
8	CONCLUSION	143
8.1	Contributions	144
8.2	Limitations	145
8.3	Future research	145
8.4	Declaration of authorship	146
8.5	Publications list	146
8.5.1	<i>Published papers</i>	146
	BIBLIOGRAPHY	149

APPENDIX A	AVALON AERODYNAMIC COEFFICIENTS	159
APPENDIX B	EKF JACOBIANS	167

INTRODUCTION

Unmanned Aerial Vehicle (UAV) is an aircraft that can fly autonomously or be remotely piloted without a human aboard. It is a topic of increasing interest in academic and commercial fields due to its flexibility and efficiency in executing functions with reduced human risk. The advantages are additionally prominent when dealing with small UAVs, which are more versatile, portable, easier to maintain and cheaper than the larger ones, besides of being able to perform the same tasks with repeatable flights ([VALAVANIS, 2007](#)).

Several configurations of UAVs were proposed in the literature, each one fulfilling different requirements, such as fixed-wing, Vertical TakeOff and Landing (VTOL), flapping wings and aerostats ([NONAMI, 2010](#)). Flapping wings are aircraft with mechanisms based on the movement of birds and insects and aerostats are vehicles lighter than the air. Fixed-wing aircraft require a runway to take off and land and have a high endurance, while VTOL vehicles can take off and land vertically in small areas, but usually operate at low velocities.

Recent developments in the field of small UAVs have led to a renewed interest in tailsitter aircraft that combine the advantages of fixed-wing and VTOL. Some of these designs rely in complex mechanical structures, like tilting rotors ([CETINSOY, 2015](#); [WANG *et al.*, 2017](#)), ducted-fan motors with control vanes ([NALDI; MARCONI, 2013](#); [ARGYLE; BEARD; MORRIS, 2013](#)) and vectored-thrust motors ([MIYAZAKI; TSUBAKINO, 2017](#); [WANG; ZHU; KUANG, 2017](#)). Others consider the use of multiple motors with counter-rotating propellers ([RITZ; D'ANDREA, 2017](#); [ZHOU *et al.*, 2017](#)).

The disadvantage of these hybrid aircraft is their complex operation with multiple stages of flight in the vertical and horizontal positions and in the transitions between them ([SAEED *et al.*, 2015](#)). But regarding UAVs, this operation can be facilitated by using an autopilot to provide flight autonomy.

In the UAV development, the autopilot is the most critical part with the responsibilities of keeping the aircraft stable during the flight manoeuvre by commanding its actuators. It can also execute other tasks, e.g. following a predefined path, monitoring the aircraft state, acquiring information from the environment and controlling the vehicle peripherals (VALAVANIS; VACHTSEVANOS, 2015).

Considering the abovementioned advantages and disadvantages, this thesis specifies a new tailsitter UAV, named AVALON (Autonomous VerticAl takeOff and laNding), which is a small electric aircraft with fixed wings and only one propeller. Along with the physical design, it is presented an autopilot structure and an evaluation of the appropriate techniques for its operation. The autopilot collects and processes information from multiple sensors, controls the aircraft throughout its entire operation and performs path-following tasks.

1.1 Motivation

As already stated, a tailsitter UAV combines the benefits of fixed-wing, by having a higher endurance and being able to achieve higher velocities, with the benefit of VTOL, by not requiring a runway to take off and land. These abilities favour the use of the aircraft in both fixed-wing and VTOL applications.

UAVs are already employed in aerial photography, monitoring of coastlines, power-lines inspection, search and rescue operations, monitoring of pollution, detection and assistance in fires and atmospheric sampling for meteorological services (AUSTIN, 2010). Specific applications to the precision agriculture are also reported, like estimating degree and extent of shrub utilisation, mapping vegetation, measuring crop biomass, detecting diseases, assessment of irrigation systems, among others (ZHANG; KOVACS, 2012). All these are suitable for tailsitter UAVs, with the advantage of offering more conducive takeoff and landing in small areas.

Due to its complex operation, a tailsitter UAV can only be employed in these situations if it has a reliable autopilot. The flight control of this type of vehicles is challenging because of its highly nonlinear behaviour, making the autopilot development itself a unique and arduous challenge. The process must be validated considering the entire aircraft operation.

Previous studies have proposed small tailsitter UAVs, but with complex mechanical structures or with multiple motors, which increases their weight and cost, and hinders their maintenance. Matsumoto *et al.* (2010) designed a modified radio-controlled model aircraft to work as a tailsitter with only one propeller. However, it required a fixed structure on the ground to assist the takeoff and landing and a hook on the aircraft. It remains unknown if a tailsitter with only one propeller and without auxiliary structures can fly autonomously.

1.2 Research question and hypothesis

The research question of this thesis is: can a tailsitter UAV with only one propeller and without auxiliary structures perform autonomously the vertical takeoff and landing, the transitions between vertical and horizontal positions, and the horizontal flight?

In order to answer this question, the following hypothesis was defined: a tailsitter UAV with only one propeller and without auxiliary structures can perform autonomously the vertical takeoff and landing, the transitions between vertical and horizontal positions, and the horizontal flight with an autopilot capable of estimating the position, velocity and attitude of the aircraft by using sensor fusion techniques and applying these information in control laws that provide trajectories and stabilise each flight stage with feedback and feedforward controllers.

However, such hypothesis raises more questions: i) which sensor fusion techniques should be used to estimate position, velocity and attitude?; ii) which methods can provide the desired trajectories?; and iii) which controllers can stabilise all flight stages?

From the literature, it is possible to list the Extended Kalman Filter (EKF) (LEWIS, 1986) and Unscented Kalman Filter (UKF) (JULIER; UHLMANN, 1997) as sensor fusion techniques, the trapezoidal velocity profile (SICILIANO; SCIavicco; VILLANI, 2009) to define smooth trajectories, the path-following algorithms Lookahead (BREIVIK; FOSSEN, 2005), Pure Pursuit and Line-of-Sight (PLOS) (VENKATRAMAN *et al.*, 2010), Nonlinear Guidance Law (NLGL) (PARK; DEYST; HOW, 2007) and Vector Field (NELSON *et al.*, 2007) to provide control laws that guarantee path tracking, and Proportional-Integral-Derivative (PID) controllers (ÅSTRÖM; HÄGGLUND, 2006) and Dynamic Inversion (DI) (LEWIS; DAWSON; ABDALLAH, 2003; SIEBERLING; CHU; MULDER, 2010) as control methods to stabilise each flight stage.

1.3 Objectives

The primary aim of this thesis is to specify a tailsitter UAV with only one propeller and without auxiliary structures and its autopilot, which has to provide estimations about the aircraft spatial information, perform guidance tasks and execute stable flight manoeuvres through vertical takeoff and landing, transitions between vertical and horizontal positions and horizontal flight.

Since different methods exist to accomplish these requirements, it can also be listed as secondary objectives to evaluate and select the most appropriate techniques for sensor fusion, path-following and controlling the flight stages.

1.4 Contributions

The thesis' most important contribution is the definition of a new tailsitter UAV, called AVALON, that can take off and land in the vertical position in small areas, change its position from vertical to horizontal and fly horizontally as a fixed-wing following a reference path built from straight lines segments, all without assistance of a human pilot. Moreover, to accomplish all these requirements, other contributions were derived:

- Kinematic and dynamic models of AVALON, implemented in a simulation environment with sensor noise and wind disturbances to validate the autopilot techniques (SILVA; MARCONATO; BRANCO, 2015; SILVA *et al.*, 2017).
- Development and comparison of Extended Kalman Filter and Unscented Kalman Filter as a GPS (Global Positioning System) aided inertial navigation with inclusion of magnetometer and barometer data under different configurations of sensor noise. UKF shows smaller errors in the attitude estimation than EKF (SILVA; WILSON; BRANCO, 2015).
- Extension of Lookahead, Pure Pursuit and Line-of-Sight, Nonlinear Guidance Law and Vector Field to contemplate changes in the altitude and comparison of these extended path-following algorithms under different intensities of wind disturbance. Lookahead, PLOS and Vector Field have a similar performance, but Vector Field is more robust to wind disturbances (PELIZER; SILVA; BRANCO, 2017b; PELIZER; SILVA; BRANCO, 2017a).
- A new control structure using computed torque and Incremental Nonlinear Dynamic Inversion, which considers the system nonlinearities, and its comparison with a classical method based only in PID controllers. The new control strategy presents better accuracy than the classical method (SILVA *et al.*, 2017; SILVA *et al.*, 2018).

1.5 Thesis overview

The remaining part of the thesis proceeds as follows: Chapter 2 details the UAV concepts and the different types of tailsitter with a literature review; AVALON specification, its kinematic and dynamic models and autopilot structure are presented in Chapter 3; Chapter 4 compares the Extended Kalman Filter and Unscented Kalman Filter; the extended path-following algorithms and the other solutions to define trajectories are shown in Chapter 5; the control structures responsible to stabilise the aircraft through all flight stages are described in Chapter 6; Chapter 7 introduces AVALON prototype; and, finally, Chapter 8 reports the conclusions.

UNMANNED AERIAL VEHICLES

2.1 Chapter overview

UAV is generally any vehicle capable of flying without a human aboard ([VALANIS, 2007](#)). This description is broad and there are several terms used to refer to unmanned aircraft, so this chapter presents the definitions and nomenclatures adopted in this text. It also introduces the different types of vehicles to contextualise AVALON and the related work regarding the development of a tailsitter.

2.2 Unmanned aircraft system

Although the designation UAV is applied to the pilotless aircraft that can be controlled remotely or from an onboard system, there is an entire system which supports the aircraft operation, called Unmanned Aircraft System (UAS). This name is adopted by the Federal Aviation Administration and by the international academic community not only to describe the physical vehicle, but also the elements associated with its functions, such as payload, control station, communication links and support equipment ([U.S. GOVERNMENT ACCOUNTABILITY OFFICE \(U.S. GAO\), 2008](#); [AUSTIN, 2010](#)).

The aircraft concept includes fuselage, wings, tail and other physical structures, propulsion system, flight control and navigation system, communication equipment and electrical power supplies. The main requirements for its design are endurance, range and velocity, which determine its configuration.

The payload is composed by sensors related to the UAV mission or application, which could be small cameras or even complex high-power radar. Sometimes the information from these sensors can be extracted from multiple data and are processed in onboard modules.

A control station is usually based on the ground, when is called Ground Control Station (GCS), and concentrates the operation and management of the mission. It receives and sends information from and to the aircraft, like telemetry data from sensors and commands to the actuators. It can also be the location where the mission is planned, before and during the flight, and where the support equipment is controlled.

Communication links are very important since they provide means to take control of the aircraft during the mission and allow the monitoring of its condition. The most common medium is wireless communication, which offers an uplink from the GCS to the vehicle and a downlink from the aircraft to the GCS. It is also possible to have communication links among different vehicles such as in a UAV swarming.

Depending on the mission and aircraft complexity, support equipment can also be needed for the correct operation, e.g. launch, recovery and transportation equipment. This equipment usually stays within the GCS to facilitate its control.

Despite the importance of all elements for a UAV project, this thesis focuses in the aircraft development, which already presents relevant challenges. However, it must be stated that all relevant elements of the UAS must be considered for AVALON to operate in a complete mission, with the exception of the support equipment.

Remotely Piloted Aircraft System (RPAS) or Remotely Piloted Aircraft (RPA) are other terms employed for an unmanned aircraft when it includes an off-board human pilot in the loop to control the flight (COOKE *et al.*, 2016). In this case, the pilot can have a Visual Line-Of-Sight (VLOS) with the aircraft, which means that he can visualise it without auxiliary accessories for all procedures, or can operate the vehicle Beyond-Line-Of-Sight (BLOS) with the assistance of specialised equipment, like onboard cameras.

The Brazilian legislation regarding UAVs is established by the Agência Nacional de Aviação Civil (ANAC), responsible for regulating and supervising the civil aviation, by the Departamento de Controle do Espaço Aéreo (DECEA), which regulates the integration into the national airspace, and by the Agência Nacional de Telecomunicações (ANATEL), which regulates the telecommunications. ANAC and DECEA adopt the definition of RPA as an aircraft remotely piloted by a human with a purpose different from recreation. According to these institutes, it must be possible for the pilot to take control of the aircraft during the entire flight, with the autonomous flight being allowed only during a remote control malfunction (ANAC, 2017; DEPARTAMENTO DE CONTROLE DO ESPAÇO AÉREO (DECEA), 2016). For this reason, AVALON can also be controlled by a pilot, but its main concept is to operate autonomously, so the preferred term chosen and applied in this text is UAV.

The RPAs are classified by ANAC and DECEA according to their maximum take-off weight: class 1 as aircraft with a maximum takeoff weight greater than 150 kg, class 2

as aircraft with a maximum takeoff weight greater than 25 kg and less than or equal to 150 kg and class 3 as aircraft with a maximum takeoff weight less than or equal to 25 kg. Thus, AVALON is a class 3 aircraft, which is considered a small UAV.

Another possible classification is regarding the physical configuration, like fixed-wing aircraft, Vertical TakeOff and Landing (VTOL) vehicles, aerostats and flapping-wing aircraft. Most of the small UAVs developed recently are fixed-wing or VTOLs ([VALAVANIS; VACHTSEVANOS, 2015](#)), whose main characteristics are described in the following subsections.

2.2.1 Fixed-wing aircraft

A fixed-wing aircraft fly due to the lift force generated by their forward velocity in relation to the air and by the shape of their main wings. Gliders and tethered kites are examples of fixed-wing vehicles that gain altitude from the air movement, but even they require thrust from propellers to generate the forward velocity, such as in traditional fixed-wing vehicles.

UAVs with fixed wings operate similarly to traditional fixed-wing aircraft, being controlled by their propellers and by primary control surfaces in their wings and tails. These control surfaces, indicated in Figure 1, are articulated devices used to control the vehicle attitude, which is the aircraft orientation defined by three angles: roll, pitch and yaw. Ailerons provide roll control, while elevators provide pitch control and rudder provides yaw control ([MOIR; SEABRIDGE, 2008](#)).

Figure 1 – Fixed-wing UAV example with indicated control surfaces.

Source: Elaborated by the author.

Fuselage, wings and tail are the physical parts that compose the fixed-wing aircraft. The fuselage is the protective structure along the body of the aircraft and where the payload, batteries and flight system are stored.

The wings provide most of the lift to sustain the aircraft in the air. This force is generated due to the difference in pressure caused by the air in the lower and upper part

of the wings and is affected by the wing shape.

The tail has a horizontal stabiliser, where elevators are located, and a vertical stabiliser with rudder. In addition to sustaining these control surfaces, the tail assists in the stabilisation of the flight, keeping the vehicle in a straight path.

These parts can appear in different configurations or can even be omitted on fixed-wing UAVs, e.g. in the flying wings that are tailless. Each configuration has specific characteristics, but they all have similar advantages and disadvantages when compared to rotary wing aircraft.

Fixed-wing UAVs operate at higher speeds and have more efficient aerodynamics, which provides a longer flight duration. These two characteristics jointly guarantee that the aircraft can cover wider areas in each flight. Also, fixed-wing vehicles can carry larger payloads for longer distances, which is interesting for applications that require heavy sensors. As disadvantages, fixed-wing UAVs require a runway or a human/equipment launcher for takeoff and landing and they cannot stay stationary at one specific position.

2.2.2 Vertical takeoff and landing aircraft

Most of the UAVs that are able to takeoff and land in the vertical position are rotary-wing aircraft, which use the thrust generated by their propellers to overcome gravity. Their main archetype is the multirotor, but there are also helicopters UAVs.

Helicopters have one main rotor responsible for keeping the vehicle in the air and another rotor at the tail responsible for the attitude control. They have greater mechanical complexity and difficult operation when compared to multirotors, which can result in increased costs.

Multirotors have multiple rotors fixed in a symmetric structure, as seen in Figure 2. They can be classified according to their number of rotors as trirotor with 3 rotors, quadrotor with 4 rotors, hexarotor with 6 rotors or octarotor with 8 rotors.

The operation of multirotors is based on their rotors velocity, which spins in opposite directions to cancel each others torque. The altitude change is based on the different velocities of all rotors: a velocity increase results in an ascend, while a decrease results in a descent. The attitude control depends on the change of a set of rotors velocities, for example, an increase on the velocities of the two rotors at the back results in a positive pitch.

The main advantages of rotary wings are the takeoff and landing in small areas without the need of a runway and the ability to hover, which facilitates the observation of fixed points. However, they operate with small velocities and with shorter endurance, which limits their capacity to flight over significant areas.

Figure 2 – Multirotor UAV example, also known as quadrotor.

Source: Elaborated by the author.

Therefore, the existence of hybrid aircraft that aggregate the advantages of fixed-wing and VTOL UAVs benefits their insertion in civil applications. Different designs were developed to achieve this goal, as presented in the following section.

2.3 Related work

In the literature, several designs were proposed for hybrid UAVs able to take off and land vertically and fly horizontally with high velocity as tailsitters. These aircraft usually contain complex mechanism, like ducted-fan rotors with control vanes (NALDI; MARCONI, 2013; ARGYLE; BEARD; MORRIS, 2013; JUNG; SHIM, 2012), tilting rotors (GUERRERO *et al.*, 2009; GARCIA-SALAZAR *et al.*, 2008; PAPACHRISTOS; ALEXIS; TZES, 2013; ROSAS *et al.*, 2012; TA; FANTONI; LOZANO, 2012; CETINSOY, 2015; WANG *et al.*, 2017), counter-rotating propellers (GARCIA *et al.*, 2012), cyclic/collective rotor (SONG; UNDERWOOD, 2007) and vectored-thrust motors (XILI; YONG; JIHONG, 2008; HANG *et al.*, 2013; ZHANG *et al.*, 2013; MIYAZAKI; TSUBAKINO, 2017; WANG; ZHU; KUANG, 2017).

Other types of tailsitters are aircraft with more than one propeller that rotate in opposite directions, with two motors in the main wings (STONE *et al.*, 2008; KUBO; SUZUKI, 2008; BAPST *et al.*, 2015; RITZ; D'ANDREA, 2017; VERLING *et al.*, 2016), with two motors aligned with the lateral or longitudinal axes of the aircraft body frame (ESCARENO *et al.*, 2008; TA; FANTONI; LOZANO, 2011) or with four motors symmetrically aligned in the main wings (SINHA *et al.*, 2012; OOSEDO *et al.*, 2013; ZHOU *et al.*, 2017). Fewer works propose tailsitter UAVs with only one propeller (BILODEAU; WONG, 2010; KITA; KONNO; UCHIYAMA, 2010).

Saeed *et al.* (2015) also report a different categorisation of small hybrid UAVs as convertiplanes or tailsitters. The authors consider tilt-rotors, tilt-wings, rotor-wing and aircraft with multiple rotors as convertiplanes and tailsitters are ducted-fan aircraft, vehicles with control surfaces to perform the transition, aircraft with multiple motors that use differential thrust to control their attitude and reconfigurable wings aircraft, which can extend or retract wings during vertical flight. Related to control, it lists decoupled PID as the most used technique, followed by LQR (Linear Quadratic Regulator), gain-scheduling, backstepping and Nonlinear Dynamic Inversion (NDI).

All these types of vehicles were found with a systematic review whose objective was to identify tailsitter UAVs and the methods applied to their autopilots. Among these works, the ones discussing a small aircraft without complex mechanisms and controlled by traditional control surfaces and throttle were selected to constitute this thesis' related work. As traditional control surfaces, it is considered aileron, elevator, rudder and elevon. The elevon is not very common on conventional aircraft, but is frequently applied to vehicles without tail by combining the functions of ailerons and elevators.

Stone *et al.* (2008) present a tailsitter UAV dubbed T-Wing with two combustion engines and counter-rotating propellers, one in each wing. The vehicle prototype weighs 29.48 kg, has a wing span of 2.177 m and a height of 1.524 m. Its autopilot hardware is a 400 MHz Celeron with an IMU (Inertial Measurement Unit), GPS and static pressure sensor. Also, a Kalman Filter in the commercial module estimates the position, velocity and attitude from the IMU and GPS data.

The T-Wing's autopilot allows the operation in four modes: vertical flight, horizontal flight and transitions from vertical to horizontal and from horizontal to vertical positions. During the vertical flight, the autopilot low-level controllers are LQR controllers obtained from the linearised equations of motion for a nonclimbing vertical flight with an Euler angle representation of the attitude. Its guidance is based on PID waypoint navigation, which provides translational velocities, belly point angle and a climb rate to the low-level controllers.

In the transitions, the guidance is given by a quaternion control, which generates angular velocity to the low-level classical controllers. These low-level controllers are also used in the horizontal flight, but the guidance is a proportional navigation bank-to-turn algorithm. The change between each stage is based on a flag that determines if the next waypoint is horizontal or vertical.

The validation was performed with simulations and flight testing of the prototype. The simulation results considered wind disturbances and covered all four modes with a plus flight pattern during hover test. The vertical flight test was also conducted with a plus flight pattern and with the aircraft linked to a tether rig. Moreover, it is reported flight data for autonomous takeoff, transitions and horizontal flight.

[Kubo and Suzuki \(2008\)](#) propose a tailsitter UAV with two counter-rotating propellers fixed in their main wings, such as T-Wing, but with electric motors. Its autopilot has LQR controllers for the longitudinal dynamics and a neural network as the transition reference command generator. The longitudinal controllers are a fixed-gain controller for velocity and a gain-scheduled controller for pitch, whose gains were obtained from linearised models of the vehicle for trimmed points.

The reference values of pitch for the gain-scheduled controller is set by a neural network with 3 layers and trained offline for the transition flight. The neural network is optimised by a genetic algorithm approach with a variation of the model uncertainty and wind disturbances after some generations.

The overall control strategy for the tailsitter transition was evaluated in a simulated environment with interpolation among three linear models with different flight velocities. The results revealed an improvement of the genetic algorithm approach to the transition from the horizontal to vertical position.

The tailsitter designed by [Bapst et al. \(2015\)](#) also has the twin-propellers configuration with electric motors, but their aircraft is tailless with the main wings constituting the entire airframe. The aircraft weights 0.9 kg and the autopilot hardware is a Pixhawk autopilot with inertial sensors, digital airspeed sensor and GPS. The autonomous flight was achieved with a single controller for hover and horizontal flight considering the propellers thrusts and elevons deflections as actuations. The controller is divided into two parts: the VTOL controller, which receives a reference acceleration and outputs references of attitude and thrust, and an attitude and body-rate controller, which receives these references and defines the actuators commands. The reference thrust is calculated from the projection of the reference acceleration in the thrust direction. The attitude reference is a combination of a predefined yaw rotation with pitch and roll rotations that aligns the thrust direction to the reference acceleration vector. Afterwards, a reference angular velocity, obtained from the quaternion error, is used to calculate a reference torque, which gives the actuation commands from the aircraft model.

The controller strategy was validated with simulations and flight experiments, which included hovering, transitions and horizontal flight with the velocity component controlled by a pilot.

A similar prototype was used by [Ritz and D'Andrea \(2017\)](#), who also propose a single controller structure for hover and horizontal flight with an estimation of unknown aerodynamic parameters. Their vehicle weights 0.15 kg and its control algorithm runs on a 168 MHz microcontroller. The estimation of positions, velocities, attitude and angular velocity is performed by a commercial system.

The controller strategy is composed by four parts: a position control that computes

a reference force based on position and velocity errors, a coordinated flight that defines a reference attitude and reference thrust, an attitude control and a body rate control. In the body rate control, the reference torques are converted to actuators commands by using parametrised functions of aerodynamic effects. An Extended Kalman Filter is applied to estimate the parameters of pitching moment and lift and drag forces from IMU measurements.

Experimental results of transition manoeuvre from hover to horizontal position and horizontal flight were performed to validate the controller. During the horizontal flight, the aircraft follows an eight trajectory with constant height.

Verling *et al.* (2016) also use a tailless twin-propellers tailsitter, like Bapst *et al.* (2015) and Ritz and D'Andrea (2017), to develop a single controller to handle its full flight operation. The vehicle uses the Pixhawk autopilot platform with IMU, GPS, magnetometer, barometer and airspeed sensor, weights 2.5 kg and has elevons and motors as actuators.

The aircraft model combines rigid-body motion principles with aerodynamics obtained from wind-tunnel data. The controller structure has two main parts: one that gives a reference torque based on errors of pitch and roll and angular velocities, and another that maps the reference torque, velocity and reference thrust to actuator commands.

Experimental results are presented for hover, horizontal flight and transitions, showing the correct tracking of the attitude. A NLGL guidance loop was also applied to the horizontal flight with the system flying to a waypoint and executing a loiter trajectory.

In the work of Escareno *et al.* (2008), it is defined a tailsitter with two counter-rotating propellers aligned in the longitudinal axis of the aircraft body frame. The dynamics of the vehicle are represented by a Newton-Euler formulation with the forces of lift and drag as aerodynamic effects. The authors focus in the hover flight, presenting a control strategy based on bounding the pitch via a tracking error after the stabilisation of the positions by a nonlinear control law. This control structure is validated through simulation and experimental flight, with an IMU as sensor and a Rabbit microcontroller implementing the control laws.

Another tailsitter with two counter-rotating propellers is presented by Ta, Fantoni and Lozano (2011), but aligned in the lateral axis of the body frame. The dynamics of the vehicle are defined from Newton-Euler formulation and the control forces and torques from the motors are the same for vertical flight and horizontal flight, but the aerodynamic forces and torques are different for each flight mode. During the vertical flight, the aerodynamic torques are given by aileron and rudder deflections, while, during the horizontal flight, the aerodynamic forces and torques are established by drag and lift forces from the wing

and aileron and rudder deflections.

For autonomous flight, the tailsitter has an altitude control for vertical flight and an attitude control for the entire flight. The vertical flight altitude controller is a PID controller of the altitude error that outputs the total thrust and the attitude control laws are saturated functions of the quaternion error with parameters obtained from pole placement method. As outer loop, the values of altitude and pitch angle are defined by trapezoidal velocity profiles during takeoff, landing and transition.

The controllers were validated with simulations and experimental flight. The simulation presented results from takeoff, transition, horizontal flight and landing, while the experimental flight included vertical flight with takeoff and landing. The vehicle prototype used in the experimental flight weights 0.9 kg, has a microcontroller dsPIC33 to implement the control strategies, an IMU to measure the attitude and a sonar in the tail to calculate the altitude.

[Sinha *et al.* \(2012\)](#) propose a tailsitter that can work as a quadrotor during vertical flight due to its four fixed rotors at a flying wing airframe. The prototype, called Quadshot, weights 0.7 kg, uses elevons and differential torque from the four propellers to control the attitude. As autopilot hardware, it is used Lisa/M or Lisa/L with an ARM Cortex-M3 microcontroller from the Paparazzi Project and an Aspirin IMU.

Quadshot has three operation modes: hover, forward flight and acrobatic flight. Each mode has a different algorithm as high-level controller to define a reference attitude, but all of them use the same low-level controllers with different gains for attitude control. The low-level controllers are decoupled PID loops of the quaternion error whose output is converted to body torques with differential thrust for the alignment of the vertical and lateral axes and elevons are used to align the longitudinal axis. There was also an addition of feedforward gains to the inner loops to improve performance.

During hover, the aircraft works as a quadrotor, controlling the attitude with differential thrust. The attitude reference is specified by the pilot as Euler angles in the hover mode and in forward flight. However, during the forward flight, the pilot can also perform a transition between forward flight and hover by setting a pitch step, which results in summed values of pitch until it reaches the final position. Differently, in the acrobatic mode, the pilot sets the angular velocity instead of Euler angles, which is converted to bounded references of quaternion.

The control strategy of Quadshot was validated with flight experiments. The first experiment had manual disturbances during the hover in roll, pitch and yaw and the second was a takeoff and transition from hover to forward flight.

The work of [Bilodeau and Wong \(2010\)](#) compares cascade controllers for a tailsitter vehicle during hover. Their tailsitter has one electric motor, weights 0.465 kg and contains

two elevons and one rudder. The attitude is estimated with accelerometers and gyroscopes by the autopilot hardware, which has a 50 Hz refresh rate.

The controllers are designed according to a linearised longitudinal model of the aircraft equations of motion, whose unknown parameters were identified by the prediction error method. The control structure has an inner loop with the angular velocity error and an outer loop with the pitch error obtained from the quaternion error. The first configuration has a PI (Proportional-Integral) controller in the outer loop and a P (Proportional) controller in the inner loop, while the second one has a P controller in the outer loop and a PI controller with anti-windup in the inner loop. Frequency analysis and flight test showed that the second configuration has a reduced overshoot when compared to the first one.

[Kita, Konno and Uchiyama \(2010\)](#) developed a tailsitter with only one propeller based in a radio-controlled model aircraft that weights 0.7 kg. As autopilot hardware, it uses a microcomputer board Alpha Project STK-7125 and a sensor collection with an attitude sensor module, GPS, pressure sensor and ultrasonic sensors.

The authors propose an offline optimisation to minimise the transition time with restriction in the altitude change. The optimisation process is described as finding values of thrust and pitch for each velocity that maximise an accelerating force by searching for all ranges of each parameter.

The pitch trajectories are sent to an attitude controller composed by gain scheduling PID controllers with quaternion feedback. The altitude is also controlled by a feedforward controller that outputs the propeller RPM (Revolutions Per Minute), kept by a PI controller.

Flight tests and simulations showed the correct following of the pitch trajectory during the transitions. It is also reported flight tests of takeoff and landing with a stand and hook structure.

Table 1 summarises the main characteristics from the related work. It is clear that the focus of these works is the control for the tailsitter flight manoeuvres. Also, it is notable the lack of works dealing with sensor fusion algorithms and navigation solutions for the correct operation of these UAVs.

In this thesis, AVALON is presented as a small electric fixed-wing tailsitter with only one motor. As seen in the literature, some authors choose counter-rotating propellers to avoid the gyroscopic effect of the propeller that produces parasitic torque. However, these disturbances are only completely cancelled if the motors and propellers are identical, which does not happen in practice. Therefore, the decision of using only one motor and propeller in AVALON reduces its mechanical complexity, but requires a better control and effort from the control surfaces to compensate for the parasitic torques. Also, differently

Table 1 – Main characteristics from related work.

Related work	Physical configuration	Control techniques	Guidance algorithms	Sensor estimation
(STONE <i>et al.</i> , 2008)	Two motors in the main wings	LQR and gain scheduling	PID waypoint navigation	Commercial
(KUBO; SUZUKI, 2008)	Two motors in the main wings	LQR and gain scheduling	-	-
(BAPST <i>et al.</i> , 2015)	Two motors in the main wings	NDI	-	-
(VERLING <i>et al.</i> , 2016)	Two motors in the main wings	NDI	NLGL path-following	Commercial
(RITZ; D'ANDREA, 2017)	Two motors in the main wings	NDI and adaptive control	-	Commercial
(ESCARENO <i>et al.</i> , 2008)	Two motors aligned with the lateral or longitudinal axes of the aircraft body frame	Nonlinear control based on Lyapunov	-	-
(TA; FANTONI; LOZANO, 2011)	Two motors aligned with the lateral or longitudinal axes of the aircraft body frame	NDI and Nonlinear control based on Lyapunov	-	-
(SINHA <i>et al.</i> , 2012)	Four motors	PID and gain scheduling	-	Commercial
(BILODEAU; WONG, 2010)	One motor	Decoupled PID	-	-
(KITA; KONNO; UCHIYAMA, 2010)	One motor	Decoupled PID and gain scheduling	-	-

Source: Elaborated by the author.

from the AVALON's autopilot, [Bilodeau and Wong \(2010\)](#) only present results for the vertical flight and [Kita, Konno and Uchiyama \(2010\)](#) tailsitter uses a hook and a fixed structure on the ground to land their aircraft.

In order to allow the correct operation of AVALON, two different techniques of Sensor Fusion are presented in this thesis, along with four different guidance algorithms. Moreover, two solutions referred in the literature were evaluated for AVALON's control, decoupled PID and NDI. Therefore, these comparisons add to the state-of-art for tailsitter UAV development and assist in the choosing of the right techniques for these UAVs.

2.4 Final remarks

Several configurations of small UAVs exist in the literature, but hybrid vehicles with VTOL and fixed-wing capabilities can outperform traditional fixed-wing aircraft and VTOL like multirotors. The associated advantage is the possibility to takeoff and land in small areas without the need of a runway with higher endurance.

Among these hybrid aircraft, called tailsitters, there is also different configurations, which sometimes depend on complex mechanisms or rely on counter-rotating propellers. To keep a simple design and a low cost, which is desirable for small UAVs, AVALON is based on a single motor and its autopilot has a structure that handles estimation of sensor data and guidance tasks, together with the control of the tailsitter flight.

Next chapter presents AVALON, a new tailsitter UAV, with the definitions of the physical design, kinematic and dynamic models and the main structure of the autopilot.

AVALON

3.1 Chapter overview

This chapter describes the AVALON design, its kinematic and dynamic models and its autopilot structure. It is the basis for the following chapters, presenting the main composition of AVALON operation and functions of the autopilot units. The mathematical model is also fundamental for all comparisons presented lately, since it defines the simulation environment where they are performed, comprising characteristics from reality like sensor noise and wind disturbance.

3.2 Physical design

During the project stage, the requirements for AVALON are to be a small electric tailsitter with fixed-wing, simple to construct and maintain. Also, it should encompass all flight stages from this type of vehicle, divided into five stages illustrated in Figure 3: vertical takeoff, Transition from Vertical to Horizontal position (TVH), horizontal flight, Transition from Horizontal to Vertical position (THV) and vertical landing.

Its operation starts with the aircraft in the vertical position for takeoff. The TVH begins when the desired altitude is reached and then pitch varies from approximately 90° to 0° . The following stage is the horizontal flight, in which the aircraft is able to follow a desired path with coordinated turn manoeuvres. At the end of the path, the aircraft performs the THV that finishes when the pitch is approximately 90° . The final flight stage is the landing that ends with the propellers turning off.

Respecting these specifications, a preliminary design represented in Figure 4 was produced from discussions with aeronautical engineers and validated in a flight simulator (SILVA; MARCONATO; BRANCO, 2015). The aircraft is symmetric in the horizontal

Figure 3 – AVALON operation divided in five flight stages: takeoff, TVH, horizontal flight, THV and landing.

Source: Elaborated by the author.

plane, uses flat plate as aerofoil and has a rounded structure to keep it in the vertical position during the takeoff and landing stages.

However, during the construction of the first prototype, the structure to keep it in the vertical position was changed to structures at the end of the tail, which has the same effect and reduced weight. Other changes in the preliminary design were the increase of another rudder to give more control action during the flight and the addition of carbon fibre on the wings to reduce their vibration.

Table 2 presents the estimated physical parameters based on the final prototype. These parameters are used in the kinematic and dynamic models that represents AVALON's flight during its entire operation.

3.3 Kinematic and dynamic models

AVALON's motion depends on the aircraft's inertia characteristic, its aerodynamic forces, the gravitational field from Earth and propulsive forces (PERKINS; HAGE, 1949). This motion can be described with two different frames, one body-fixed frame with an origin that coincides with the aircraft Centre of Gravity (CG) and another inertial frame Earth-fixed in local-level and tangent to the gravity. The body frame axes X (longitudinal axis) points to the aircraft propeller, Y (lateral axis) points to the right wing and Z (vertical axis) points to the Earth, as seen in Figure 5. Figure 5 also presents the inertial frame, with X_N axis pointing to the north, Y_E axis pointing to the east and Z_D axis

Figure 4 – AVALON design in different perspectives: isometric view in the top right, top view in the top left, front view in the bottom left and right view in the bottom right.

Source: Elaborated by the author.

Table 2 – AVALON's physical parameters.

Parameter	Value
Mass	2.5 kg
Height	1.7 m
Wingspan	1.74 m
Wing chord	0.22 m
Wing area	0.3828 m ²
Maximum engine power	600 W
I_{xx} moment of inertia element	0.86 kg.m ²
I_{yy} moment of inertia element	0.92 kg.m ²
I_{zz} moment of inertia element	0.13 kg.m ²
I_{xy} , I_{yx} , I_{xz} , I_{zx} , I_{yz} and I_{zy} moment of inertia elements	0 kg.m ²

Source: Research data.

pointing to the Earth centre.

Figure 5 – Inertial frame and body frame with the angles roll ϕ , pitch θ , and yaw ψ .

Source: Elaborated by the author.

Considering the aircraft as a rigid body with six degrees of freedom, its motion can be described with (3.1)-(3.4), where $\mathbf{p}_i = [P_N, P_E, P_D]^T$ is the position in the inertial frame, $\mathbf{v}_b = [U, V, W]^T$ is the velocity in the body frame, $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ is the quaternion representation of attitude, $\mathbf{R}_b^i$, defined by (3.5), is the direction cosine matrix that express the rotation from the body to the inertial frame, $\mathbf{v}_w$ is the wind velocity in the inertial frame, $\boldsymbol{\omega} = [P, Q, R]^T$ is the angular velocity, $\boldsymbol{\Omega}_\omega$ is the skew symmetric form of $\boldsymbol{\omega}$, $\mathbf{f}$ is the total applied force, m is the aircraft mass, $\mathbf{I}$ is the aircraft moment of inertia and $\boldsymbol{\tau} = [L, M, N]^T$ is the total applied torque (ROGERS, 2003).

$$\dot{\mathbf{p}}_i = \mathbf{R}_b^i \mathbf{v}_b + \mathbf{v}_w \quad (3.1)$$

$$\dot{\mathbf{v}}_b = -\boldsymbol{\omega} \times \mathbf{v}_b + \frac{1}{m} \mathbf{f} \quad (3.2)$$

$$\dot{\mathbf{q}} = \frac{1}{2} \boldsymbol{\Omega}_\omega \mathbf{q} \quad (3.3)$$

$$\dot{\boldsymbol{\omega}} = -\mathbf{I}^{-1} (-\boldsymbol{\omega} \times \mathbf{I} \boldsymbol{\omega}) + \mathbf{I}^{-1} \boldsymbol{\tau} \quad (3.4)$$

$$\mathbf{R}_b^i = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \quad (3.5)$$

The angular acceleration from (3.4) can be expanded into (3.6)-(3.8) by introducing the constants C_i defined with (3.9)-(3.18), where $i = 0, \dots, 9$ (STEVENS; LEWIS; JOHNSON, 2016).

$$\dot{P} = (C_1 R + C_2 P) Q + C_3 L + C_4 N \quad (3.6)$$

$$\dot{Q} = C_5 P R - C_6 (P^2 - R^2) + C_7 M \quad (3.7)$$

$$\dot{R} = (C_8 P + C_2 R) Q + C_4 L + C_9 N \quad (3.8)$$

$$C_0 = I_{xx} I_{zz} - I_{xz}^2 \quad (3.9)$$

$$C_1 = \frac{(I_{yy} - I_{zz}) I_{zz} - I_{xz}^2}{C_0} \quad (3.10)$$

$$C_2 = \frac{(I_{xx} - I_{yy} + I_{zz}) I_{xz}}{C_0} \quad (3.11)$$

$$C_3 = \frac{I_{zz}}{C_0} \quad (3.12)$$

$$C_4 = \frac{I_{xz}}{C_0} \quad (3.13)$$

$$C_5 = \frac{I_{zz} - I_{xx}}{I_{yy}} \quad (3.14)$$

$$C_6 = \frac{I_{xz}}{I_{yy}} \quad (3.15)$$

$$C_7 = \frac{1}{I_{yy}} \quad (3.16)$$

$$C_8 = \frac{(I_{xx} - I_{yy}) I_{xx} + I_{xz}^2}{C_0} \quad (3.17)$$

$$C_9 = \frac{I_{xx}}{C_0} \quad (3.18)$$

Equations (3.19)-(3.21) are the extended form of the translational acceleration from (3.2), where $\mathbf{g}_b = [G_x, G_y, G_z]^T$ is the gravitational acceleration in the body frame and $\mathbf{f}_b = [F_x, F_y, F_z]^T$ is the sum of the thrust and aerodynamic forces in the body frame.

$$\dot{U} = RV - QW + G_x + \frac{F_x}{m} \quad (3.19)$$

$$\dot{V} = -RU + PW + G_y + \frac{F_y}{m} \quad (3.20)$$

$$\dot{W} = QU - PV + G_z + \frac{F_z}{m} \quad (3.21)$$

The forces and moments acting on the aircraft can be divided into two parts. The first part is due to the propeller thrust and the second one is due to the aerodynamic structures. Equations (3.22)-(3.27) show this relation, where T_x , T_y and T_z are

the thrust components in the body frame, T_L , T_M and T_N are the moment components caused by the propeller, C_x , C_y and C_z are the aerodynamic coefficients for the forces in the body frame, C_L , C_M and C_N are the aerodynamic coefficients for the moments, $\mathbf{v}_{\text{rel}} = [V_{\text{rel},x}, V_{\text{rel},y}, V_{\text{rel},z}]^T$ is the velocity vector relative to the surrounding air, calculated according to (3.28), ρ is the atmospheric density, S is the wing area, B is the wing span and C_{Chord} is the wing chord.

$$F_x = T_x + \frac{1}{2}C_x\|\mathbf{v}_{\text{rel}}\|^2\rho S \quad (3.22)$$

$$F_y = T_y + \frac{1}{2}C_y\|\mathbf{v}_{\text{rel}}\|^2\rho S \quad (3.23)$$

$$F_z = T_z + \frac{1}{2}C_z\|\mathbf{v}_{\text{rel}}\|^2\rho S \quad (3.24)$$

$$L = T_L + \frac{1}{2}C_L\|\mathbf{v}_{\text{rel}}\|^2\rho SB \quad (3.25)$$

$$M = T_M + \frac{1}{2}C_M\|\mathbf{v}_{\text{rel}}\|^2\rho SC_{\text{Chord}} \quad (3.26)$$

$$N = T_N + \frac{1}{2}C_N\|\mathbf{v}_{\text{rel}}\|^2\rho SB \quad (3.27)$$

$$\mathbf{v}_{\text{rel}} = \mathbf{v}_{\text{b}} - \mathbf{R}_{\text{b}}^i \mathbf{v}_{\text{w}} \quad (3.28)$$

The forces and moments originated by the propeller depend on the alignment of the propeller regarding the aircraft CG, as shown in (3.29)-(3.34), where T is the thrust intensity approximated by (3.35), $\Delta\tau_{xy}$ and $\Delta\tau_{xz}$ are the angle offset in the xy and xz body frame and $\Delta T_{CG,x}$, $\Delta T_{CG,y}$ and $\Delta T_{CG,z}$ are the offset between the aircraft CG and the engine CG in the body frame. In (3.35), P_{max} is the maximum engine power, η is the propeller efficiency and δ_{thr} is the throttle control input.

$$T_x = T \cos(\Delta\tau_{xz}) \cos(\Delta\tau_{xy}) \quad (3.29)$$

$$T_y = T \sin(\Delta\tau_{xy}) \quad (3.30)$$

$$T_z = -T \sin(\Delta\tau_{xz}) \cos(\Delta\tau_{xy}) \quad (3.31)$$

$$T_L = -T_y \Delta T_{CG,z} + T_z \Delta T_{CG,y} \quad (3.32)$$

$$T_M = T_x \Delta T_{CG,z} - T_z \Delta T_{CG,x} \quad (3.33)$$

$$T_N = -T_x \Delta T_{CG,y} + T_y \Delta T_{CG,x} \quad (3.34)$$

$$T = \frac{\eta P_{\text{max}} \delta_{\text{thr}}}{\|\mathbf{v}_{\text{rel}}\|} \quad (3.35)$$

The aerodynamic coefficients are calculated with (3.36)-(3.43), where α is the angle of attack calculated by (3.44), β is the angle of sideslip calculated as (3.45), δ_{ail} , δ_{elev} and δ_{rud} are the deflection angles of ailerons, elevators and rudders respectively and $C_{\cdot,\cdot}$ are aerodynamic derivatives (ETKIN; REID, 1996).

$$C_{Lift} = C_{Lift,0} + C_{Lift,\alpha}\alpha + \frac{C_{Lift,Q}C_{Chord}Q}{2\|\mathbf{v}_{rel}\|} + C_{Lift,elev}\delta_{elev} \quad (3.36)$$

$$C_{Drag} = C_{Drag,0} + C_{Drag,\alpha}\alpha + \frac{C_{Drag,Q}C_{Chord}Q}{2\|\mathbf{v}_{rel}\|} + C_{Drag,elev}\delta_{elev} + C_{Drag,parasite} \quad (3.37)$$

$$C_X = -C_{Drag}\cos(\alpha) + C_{Lift}\sin(\alpha) \quad (3.38)$$

$$C_Y = -C_{Y,\beta}\beta - \frac{C_{Y,P}BP}{2\|\mathbf{v}_{rel}\|} + \frac{C_{Y,R}BR}{2\|\mathbf{v}_{rel}\|} + C_{Y,ail}\delta_{ail} + C_{Y,rud}\delta_{rud} \quad (3.39)$$

$$C_Z = -C_{Drag}\sin(\alpha) - C_{Lift}\cos(\alpha) \quad (3.40)$$

$$C_L = -C_{L,\beta}\beta - \frac{C_{L,P}BP}{2\|\mathbf{v}_{rel}\|} + \frac{C_{L,R}BR}{2\|\mathbf{v}_{rel}\|} + C_{L,ail}\delta_{ail} + C_{L,rud}\delta_{rud} \quad (3.41)$$

$$C_M = C_{M,\alpha}\alpha + \frac{C_{M,Q}C_{Chord}Q}{2\|\mathbf{v}_{rel}\|} + C_{M,elev}\delta_{elev} \quad (3.42)$$

$$C_N = -C_{N,\beta}\beta - \frac{C_{N,P}BP}{2\|\mathbf{v}_{rel}\|} + \frac{C_{N,R}BR}{2\|\mathbf{v}_{rel}\|} + C_{N,ail}\delta_{ail} + C_{N,rud}\delta_{rud} \quad (3.43)$$

$$\alpha = \frac{V_{rel,z}}{V_{rel,x}} \quad (3.44)$$

$$\beta = \frac{V_{rel,y}}{\|\mathbf{v}_{rel}\|} \quad (3.45)$$

One assumption of the kinematic and dynamic models is that the angles of attack and sideslip are equal to zero when the intensity velocity is smaller than 2 m/s to avoid their extrapolation at low velocities.

The actuators dynamics of δ_{ail} , δ_{elev} and δ_{rud} are also included in the model as (3.46), where the index s is *ail*, *elev* or *rud*, $\dot{\delta}_{max}$ is their maximum velocity of 3.49 rad/s and δ_s^c is the commanded deflections of the control surfaces.

$$\delta_s = \begin{cases} \int \dot{\delta}_s^c, & \text{for } \dot{\delta}_s^c \leq \dot{\delta}_{max} \\ \int \dot{\delta}_{max}, & \text{for } \dot{\delta}_s^c > \dot{\delta}_{max} \end{cases} \quad (3.46)$$

All aerodynamic derivatives were calculated with Tornado Vortex Lattice method for different values of α , β and $\|\mathbf{v}_{\text{rel}}\|$ (MELIN, 2000). Subsection 3.3.1 details the process to obtain all these derivatives for AVALON.

Since Tornado does not consider the parasite drag, it was necessary to add the coefficient $C_{\text{Drag,parasite}}$ to the model. This coefficient represents the drag caused by the wings and fuselage due to skin friction and pressure and is calculated according to (3.47), where k_{form} accounts for pressure drag, $C_{F,\text{wing}}$ and $C_{F,\text{fuselage}}$ are the skin friction coefficient for flat plates for the wing and fuselage respectively and S_{fuselage} is the wet area of the fuselage. The skin friction coefficient is calculated with the empirical log-fit by von Karman's in (3.48)-(3.49), where Re_{wing} is the Reynolds Number that considers the characteristic length as the wing chord, represented by (3.50), and Re_{fuselage} is the Reynolds Number with characteristic length as its full length (l_{fuselage}), represented by (3.51) (μ is the dynamic viscosity of the air) (PERKINS; HAGE, 1949).

$$C_{\text{Drag,parasite}} = \frac{k_{\text{form}}C_{F,\text{wing}}S + k_{\text{form}}C_{F,\text{fuselage}}S_{\text{fuselage}}}{S} \quad (3.47)$$

$$C_{F,\text{wing}} = \frac{0.455}{(\log_{10}Re_{\text{wing}})^{2.58}} \quad (3.48)$$

$$C_{F,\text{fuselage}} = \frac{0.455}{(\log_{10}Re_{\text{fuselage}})^{2.58}} \quad (3.49)$$

$$Re_{\text{wing}} = \frac{\rho\|\mathbf{v}_{\text{rel}}\|c}{\mu} \quad (3.50)$$

$$Re_{\text{fuselage}} = \frac{\rho\|\mathbf{v}_{\text{rel}}\|l_{\text{fuselage}}}{\mu} \quad (3.51)$$

3.3.1 Aerodynamic derivatives from Tornado

Tornado is a Vortex Lattice Method implementation for MATLAB that assists in the conceptual aircraft design. It can solve most aerodynamic derivatives for a wide range of wing geometries by calculating the forces and moments in different flight conditions (MELIN, 2000). Because this software is open source, easy to use and yields accurate results when compared to experimental data, it was chosen as a tool to calculate AVALON's derivatives.

In order to obtain the aerodynamic derivatives, first, the geometric model of the wings and the CG were defined in the Tornado, as seen in Figures 6 and 7. The CG was determined as 0.2 m from the aircraft nose to guarantee longitudinal static stability.

Afterwards, the aircraft flight condition was specified with fixed values of $\|\mathbf{v}_{\text{rel}}\|$, α and β . Since AVALON's flight envelope contains different flight conditions, an automatic script was built to collect data. In this script, all derivatives are saved in a matrix of

Figure 6 – Geometric model of AVALON in Tornado.

Source: Elaborated by the author.

Figure 7 – Different perspective views of the geometric model of AVALON in Tornado.

Source: Elaborated by the author.

4 dimensions for an airspeed range of $[1, 30]$ m/s with an 1 m/s step, an α range of $[-20^\circ, 20^\circ]$ with a 0.5° step and a β range of $[-20^\circ, 20^\circ]$ with a 1° step.

In each step of the script, Tornado generates a flow field of the air and calculates the forces and moments in the wings, which are used to compute the aerodynamic coefficients. These coefficients are then separated according to the contribution of each variable and normalised, resulting in the derivatives $C_{.,}$ from (3.36)-(3.43). Therefore, to be applied in the dynamic and kinematic modelling, these derivatives are saved in the 4-dimension matrix.

Examples of the resulting coefficients are shown in Figures 8 and 9, with values of C_L for intervals of α and β and for intervals of α and $\|\mathbf{v}_{\text{rel}}\|$ respectively. It is possible to observe that the coefficient significantly changes with these variables, which also reflects in the derivatives and justifies their computation for different flight conditions. The remaining coefficients can be found in similar Figures in Appendix A.

Figure 8 – C_L coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Finally, the script saves the final matrix to a *.mat* file, which is used in the simulation environment. This environment is an implementation in MATLAB of the dynamic and kinematic models and was constructed to validate AVALON's autopilot. An example of an experiment conducted in the simulation environment can be seen in a video published at http://bit.ly/AVALON_Simulation.

Figure 9 – C_L coefficient for a fixed sideslip of 0° and intervals of α and $\|\mathbf{v}_{rel}\|$ resulted from Tornado's Vortex Lattice Method.

Source: Research data.

3.4 Autopilot

AVALON's autopilot is one of the main contributions of this thesis and can be defined as the system that provides a certain level of autonomy to the aircraft. In this thesis, the autopilot supplies estimation about the aircraft state, performs stabilised manoeuvres during the flight and executes path-following tasks. In a high-level perspective, it has a structure divided according to its major functions, as represented in Figure 10.

Figure 10 – AVALON's autopilot structure divided according to its main functions.

Source: Elaborated by the author.

The Sensor Unit provides information about the aircraft state to the other Units by dealing with data from multiple sensors. In order to accomplish this function, two algorithms were analysed and compared: the Extended Kalman Filter and the Unscented Kalman Filter. Chapter 4 describes these techniques, their implementation and the results of the comparison.

The Navigation Unit manages the flight stages and supplies a reference state,

composed by the attitude as a quaternion, velocity in the Down coordinate, and velocity magnitude, for Control Unit. Moreover, it guarantees that the aircraft follows a desired path based on waypoints during the horizontal flight with a path-following algorithm. In order to choose the adequate path-following algorithm, a comparison was conducted among Lookahead, Non-Linear Guidance Law (NLGL), Pure Pursuit and Line-of-Sight (PLOS) and Vector Field, as presented in Chapter 5.

Finally, the Control Unit is responsible for stabilising the aircraft during the flight by sending commands to the actuators. It receives a reference state from Navigation Unit and computes the commands based on different control laws for each flight stage. Two strategies were employed in the definition of these controllers: the use of independent PID controllers that compute the actuation from the attitude, called Decoupled PID; and a dynamic inversion with PID controllers that calculates a commanded torque and then derives the actuation by considering the dynamic effects, named DI with PID. A comparison of these strategies and their determination can be found in Chapter 6.

The entire autopilot was implemented and validated under a simulation environment constructed with the kinematic and dynamic models in MATLAB, presented in the following chapters. This simulation environment runs with a time step of 1 ms, while the autopilot units run with a time step of 3 ms. The units period is defined according to the throughput of the Sensor Unit, which is also the period of the simulated IMU data, while simulated GPS data is delivered with a period of 20 ms. The comparison among all techniques considered replications, with sensor noise and wind disturbance generated for each one, increasing the physical fidelity of the experiments.

As a platform, the autopilot has a dedicated hardware equipped with interfaces to communicate with a collection of sensors. It is also able to send signals to the actuators. The Sensor Unit has access to data from all the sensors: an Inertial Measurement Unit (IMU) - with accelerometers, gyroscopes, magnetometers and barometer - and a GPS. The Navigation Unit receives a pre-defined path as a list of waypoints, which are followed during the horizontal flight, and the Control Unit determines the commands sent to the actuators. Chapter 7 specifies the autopilot hardware, its sensors and the aircraft actuators. Figure 11 shows the thesis structure related to the autopilot units.

3.5 Final remarks

AVALON has a nonlinear and coupled model and an operation with multiple stages, each one with unique characteristics, which brings challenges in terms of flight stabilisation and control. In order to suffice the proposal of enabling a level of autonomy to the aircraft, an autopilot structure was defined by dividing its main functions as Sensor Unit, Navigation Unit and Control Unit. All Units are essential for an appropriate flight and

Figure 11 – Thesis structure based on the autopilot units.

Source: Elaborated by the author.

were implemented with different algorithms and strategies detailed in the next chapters. All these techniques were evaluated in the simulation environment constructed from the aircraft dynamic and kinematic models. In the next chapter, the Sensor Unit is discussed.

SENSOR FUSION

4.1 Chapter overview

In order to stabilise and navigate the UAV during its flight, the autopilot requires information about the aircraft spatial state with a certain accuracy. This information is usually collected from several sensors for autonomous vehicles and, to deal with multiple data, sensor fusion techniques are used. One of these techniques is the optimal Kalman filter, but the equations that rule the UAV behaviour and its sensors are nonlinear, so it cannot be applied at all, since equations must be linear. Instead, Extended and Unscented Kalman filters have been developed and are extensively used to overcome this challenge (SADHANA; KUMARI; RAJU, 2012; GIANNITRAPANI *et al.*, 2011; ST-PIERRE; GINGRAS, 2004).

Consequently, EKF and UKF were selected as possible solutions to provide estimations about AVALON position, velocity and attitude from its sensor collection: accelerometers, gyroscopes, magnetometers, barometer and GPS. The estimation of each state obtained from the sensors data are defined in this chapter, along with the techniques descriptions. Also, to determine the more appropriate algorithm, a comparison is reported from simulation results.

4.2 State estimation from sensors

Each sensor retrieves different information from the environment, for example, accelerometers provide measures called specific forces, which are directly related to the body acceleration, while gyroscopes provide angular velocities. For the sensor fusion algorithms, each sensor measurement should be translated into knowledge for the autopilot, which is defined as a state. The state is composed by the position and velocity in the inertial frame and attitude, represented by quaternion.

An IMU is generally composed by accelerometers and gyroscopes, which measure respectively specific forces and angular velocities at a high frequency. The attitude, calculated from the angular velocities, allows the specific forces to be transformed into a known inertial frame and after compensating it with gravity, two integrations provide position estimates. The IMU operation does not require communication with other devices, making it immune to jamming. However, since accelerometers and gyroscopes are affected by noise, the integration of their data results in unbounded errors (TITTERTON; WESTON; ENGINEERS, 2004).

The calculation of the position, velocity and attitude from IMU measurements is called inertial mechanisation system and is represented by (4.1)-(4.3), where $\mathbf{x} = [\tilde{\mathbf{p}}_i, \tilde{\mathbf{v}}_i, \tilde{\mathbf{q}}]^T$ are the estimation of position and velocity in the inertial frame and attitude as quaternion, $\tilde{\mathbf{a}}$ is the measured specific force in the body frame from the accelerometers, $\tilde{\boldsymbol{\omega}}$ is the measured angular velocity in the body frame from the gyroscopes and $\mathbf{g}_i$ is the gravity acceleration in the inertial frame (ROGERS, 2003).

$$\dot{\tilde{\mathbf{p}}}_i = \tilde{\mathbf{v}}_i \quad (4.1)$$

$$\dot{\tilde{\mathbf{v}}}_i = \mathbf{R}_b^i \tilde{\mathbf{a}} + \mathbf{g}_i \quad (4.2)$$

$$\dot{\tilde{\mathbf{q}}} = \frac{1}{2} \boldsymbol{\Omega}_{\tilde{\boldsymbol{\omega}}} \tilde{\mathbf{q}} \quad (4.3)$$

Nowadays, IMUs are also complemented by other sensors, like magnetometers that measure the Earth magnetic field, and a barometer that measures the atmospheric pressure. From the magnetometer measurement, it is possible to estimate the yaw angle by rotating the magnetic field $\tilde{\mathbf{H}} = [\tilde{H}_x, \tilde{H}_y, \tilde{H}_z]^T$ in the body frame to an horizontal plane and extracting it from the resulting vector $[\tilde{M}_x, \tilde{M}_y]^T$, as can be seen in (4.4)-(4.6), where ψ_{dec} is a magnetic declination correction that accounts for the local difference between magnetic and true north (ROGERS, 2003). However, one limitation of the magnetometer solution is the requirement of information about the other angles that constitute the attitude.

$$\tilde{M}_x = \frac{\tilde{H}_x \cos \theta}{\|\tilde{\mathbf{H}}\|} + \frac{\tilde{H}_y \sin \phi \sin \theta}{\|\tilde{\mathbf{H}}\|} + \frac{\tilde{H}_z \cos \phi \sin \theta}{\|\tilde{\mathbf{H}}\|} \quad (4.4)$$

$$\tilde{M}_y = \frac{\tilde{H}_y \cos \phi}{\|\tilde{\mathbf{H}}\|} - \frac{\tilde{H}_z \sin \phi}{\|\tilde{\mathbf{H}}\|} \quad (4.5)$$

$$\tilde{\psi} = \tan^{-1} \left(\frac{-\tilde{M}_y}{\tilde{M}_x} \right) + \psi_{dec} \quad (4.6)$$

The atmospheric pressure $\tilde{p}$ is used to calculate the altitude with (4.7)-(4.8), where $\rho_{PD,0}$ is the atmospheric pressure at the initial altitude relative to the mean sea level

$-P_{D,0}$, considered here as the altitude at the takeoff measured by the GPS, ρ_0 is the sea level atmospheric pressure, L is the lapse rate defined as the rate of temperature increases in the atmosphere with increasing altitude, T_0 is the sea level temperature, R is the universal gas constant, g is the gravity acceleration intensity and M is the sea level mean molar mass. All the values of the constants relative to the atmosphere were retrieved from the International Standard Atmosphere ([INTERNATIONAL CIVIL AVIATION ORGANIZATION \(ICAO\), 1964](#)).

$$\rho_{P_{D,0}} = \rho_0 \left(1 + \frac{P_{D,0}L}{T_0} \right)^{\frac{RL}{gM}} \quad (4.7)$$

$$\tilde{P}_D = -\frac{T_0}{L} \left(1 - \left(\frac{\tilde{\rho}}{\rho_{P_{D,0}}} \right)^{\frac{RL}{gM}} \right) \quad (4.8)$$

GPS provides a deterministic solution for position using the distance between satellites orbiting the Earth. It generally has a low frequency and is susceptible to jamming. Its position measurements also have biases due to satellite ephemeris and clock errors, however, they are bounded ([TITTERTON; WESTON; ENGINEERS, 2004](#)).

The GPS position in the geodetic frame, represented as latitude λ , longitude φ and altitude h , are converted to the inertial frame by (4.9), where $\mathbf{R}_g^i$ is the direction cosine matrix that expresses the rotation from geodetic to inertial frame defined by (4.10), λ_0 , φ_0 and h_0 are, respectively, the latitude, longitude and altitude at the initial position measured by the GPS at the takeoff, $N_e(\varphi)$ is the prime vertical radius of curvature that varies with φ , according to (4.11), R_a is the semi-major axis and e_f is the first eccentricity of the world geodetic system 84. The world geodetic system 84, that defines the parameters e_f , R_a and $N_e(\varphi)$, is an approximation of the Earth to an ellipsoid and is also known as WGS 84 ([NATIONAL IMAGERY AND MAPPING AGENCY, 2000](#)).

$$\begin{bmatrix} \tilde{P}_N \\ \tilde{P}_E \\ \tilde{P}_D \end{bmatrix} = \mathbf{R}_g^i \begin{bmatrix} (N_e(\varphi) + h)(\cos \varphi \cos \lambda) - (N_e(\varphi_0) + h_0)(\cos \varphi_0 \cos \lambda_0) \\ (N_e(\varphi) + h)(\cos \varphi \sin \lambda) - (N_e(\varphi_0) + h_0)(\cos \varphi_0 \sin \lambda_0) \\ (N_e(\varphi)(1 - e_f^2) + h)(\sin \varphi) - (N_e(\varphi_0)(1 - e_f^2) + h_0)(\sin \varphi_0) \end{bmatrix} \quad (4.9)$$

$$\mathbf{R}_g^i = \begin{bmatrix} -\sin \varphi_0 \cos \lambda_0 & -\sin \varphi_0 \sin \lambda_0 & \cos \varphi_0 \\ -\sin \lambda_0 & \cos \lambda_0 & 0 \\ -\cos \varphi_0 \cos \lambda_0 & -\cos \varphi_0 \sin \lambda_0 & -\sin \varphi_0 \end{bmatrix} \quad (4.10)$$

$$N_e(\varphi) = \frac{R_a}{\sqrt{1 - e_f^2 \sin^2 \varphi}} \quad (4.11)$$

Since all these sensors provide different measurements with different characteristics (especially GPS and IMU, which are complementary), the fused measurements are usually better and more robust than using a standalone mode. The data fusion methods based on Kalman Filter, such as EKF and UKF, are composed by two stages, prediction and update. A common implementation of these methods is the GPS aided inertial navigation, that uses inertial mechanisation equations in the prediction stage and observations from the GPS and other sensors in the update stage.

4.3 Extended Kalman filter

The EKF is the most widely used estimator for nonlinear systems and is considered a nonlinear version of the Kalman filter that linearises about an estimated mean and covariance (LEWIS, 1986).

Assuming that a process model is described as a nonlinear stochastic difference equation $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$ with a similar observation model $\mathbf{h}(\mathbf{x}_k, \mathbf{v}_k)$, the system specifying the sensor state formulation can be defined by (4.12)-(4.13), where $\mathbf{x}_k$ and $\mathbf{z}_k$ are the actual state and measurement vectors, $\mathbf{u}_k$ is the control vector and $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero-mean Gaussian additive noises of the process and observation respectively, $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q})$ and $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \quad (4.12)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \quad (4.13)$$

The EKF algorithm is divided in two stages, prediction and update, as shown in Table 3, where $\mathbf{x}_{k+1}^-$ and $\mathbf{P}_{k+1}^-$ are the a priori estimation of the state vector and its covariance matrix, $\mathbf{x}_{k+1}$ and $\mathbf{P}_{k+1}$ are the a posteriori estimation of the state vector and its covariance matrix, $\mathbf{z}_{k+1}$ is the observed measurement vector, $\mathbf{A}_k$ is the Jacobian of the process model with respect to $\mathbf{x}$ evaluated at $\mathbf{x}_k$, $\mathbf{W}_k$ is the Jacobian of the process model with respect to $\mathbf{w}$ evaluated at $\mathbf{x}_k$ and $\mathbf{H}_k$ is the Jacobian of the observation model with respect to $\mathbf{x}$ evaluated at $\mathbf{x}_k$ (CRASSIDIS; JUNKINS, 2011).

For AVALON's Sensor Unit, the state vector is defined as $\mathbf{x} = [\mathbf{p}_i, \mathbf{v}_i, \mathbf{q}]^T$, i.e., position and velocity in the inertial frame and attitude quaternion. During the prediction stage, the a priori estimate is obtained with the accelerometer and gyroscope measurements by integrating the translational acceleration and velocity with Euler method and by forward integration for quaternion. The resulting propagation model is (4.14)-(4.16), where $\otimes$ represents a quaternion multiplication, Δt_s is the sensor time step and $\Delta \mathbf{q}_{\tilde{\omega}}$ is defined as (4.17) (SOLA, 2017).

Table 3 – Extended Kalman Filter Algorithm.

Prediction
$\mathbf{x}_{k+1}^- = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ $\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q} \mathbf{W}_k^T$
Update
$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k+1}^- \mathbf{H}_k^T + \mathbf{R})^{-1}$ $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \mathbf{h}(\mathbf{x}_{k+1}^-))$ $\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_k) \mathbf{P}_{k+1}^-$

Source: Elaborated by the author.

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \Delta t_s \quad (4.14)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k + (\mathbf{R}_b^i \tilde{\mathbf{a}} - \mathbf{g}_i) \Delta t_s \quad (4.15)$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k \otimes \Delta \mathbf{q}_{\tilde{\omega}} \quad (4.16)$$

$$\Delta \mathbf{q}_{\tilde{\omega}} = \begin{bmatrix} \cos(0.5 \Delta t_s \|\tilde{\omega}\|) \\ \sin(0.5 \Delta t_s \|\tilde{\omega}\|) \frac{\tilde{\omega}}{\|\tilde{\omega}\|} \end{bmatrix} \quad (4.17)$$

All measurements are considered to be affected by an additive Gaussian noise, $\tilde{\mathbf{a}}$ is the specific force summed with a Gaussian noise $\mathbf{w}_{\tilde{\mathbf{a}}} \sim \mathcal{N}(0, \mathbf{Q}_{\tilde{\mathbf{a}}})$ defined by (4.18) and $\tilde{\omega}$ is the angular velocity with an additive Gaussian noise $\mathbf{w}_{\tilde{\omega}} \sim \mathcal{N}(0, \mathbf{Q}_{\tilde{\omega}})$ according to (4.19).

$$\tilde{\mathbf{a}} = \mathbf{a} + \mathbf{w}_{\tilde{\mathbf{a}}} \quad (4.18)$$

$$\tilde{\omega} = \omega + \mathbf{w}_{\tilde{\omega}} \quad (4.19)$$

Afterwards, the GPS or magnetometer and barometric measurements are used in the update stage, with priority to GPS that has a slower frequency. During this stage, the GPS observation model is defined as the position in the inertial frame, since the conversion from the geodetic frame is performed previously by (4.9). The magnetometer observation model is the yaw angle obtained from the quaternion as (4.20) and the barometric is the

altitude obtained as $-P_D$. All these measurements are also under the influence of additive Gaussian noises, represented respectively as $\mathbf{w}_{\text{GPS}} \sim \mathcal{N}(0, \mathbf{R}_{\text{GPS}})$, $w_{\text{Mag}} \sim \mathcal{N}(0, R_{\text{Mag}})$ and $w_{\text{Bar}} \sim \mathcal{N}(0, R_{\text{Bar}})$.

$$\psi_{\text{Mag}} = \tan^{-1} \left(\frac{2(q_1 q_4 + q_2 q_3)}{1 - 2(q_3^2 + q_4^2)} \right) \quad (4.20)$$

Following the definitions of these models, the Jacobian matrices were analytically established, as can be found in Appendix B. For every step k of the filter, both stages are executed with the calculation of these matrices.

The EKF provides good results when the estimate maintained by the filter is close to the true state. It means that the second and higher-order terms in the Taylor Series expansion are small, however, this cannot be true for every system. Another disadvantage of this method is the difficulty to evaluate the Jacobian matrices of the process and observation models. Also, since the update stage does not maintain the proprieties of the quaternion, so it has to be normalised after each update.

4.4 Unscented Kalman filter

An alternative method as estimator for nonlinear systems is the UKF. The UKF uses a set of chosen weighted samples, called sigma points, to parametrise the means and covariances of probability distributions. The algorithm is based on the Unscented Transformation, a nonlinear transformation that statistically calculates the mean of a random variable.

Consider a n -dimensional random variable $\mathbf{x}$, with mean $\bar{\mathbf{x}}_{\mathbf{k}}$ and covariance $\mathbf{P}_{\mathbf{k}}$, that is propagated through a nonlinear system $\mathbf{x}_{\mathbf{k}+1} = \mathbf{f}(\mathbf{x}_{\mathbf{k}}, \mathbf{u}_{\mathbf{k}})$. The statistics of $\mathbf{x}_{\mathbf{k}+1}$ can be obtained with the Unscented Transformation by calculating $2n + 1$ sigma points and weights, as respectively detailed in (4.21)-(4.23) and (4.24)-(4.26), where $i = 1, \dots, n$, $\kappa \in \mathbb{R}$ is a scaling parameter, $\left(\sqrt{(n + \kappa) \mathbf{P}_{\mathbf{k}}}\right)_i$ is the i^{th} column of the matrix square root of $\sqrt{(n + \kappa) \mathbf{P}_{\mathbf{k}}}$ and W_i is the weight associated with i^{th} point (JULIER; UHLMANN, 1997).

$$\boldsymbol{\chi}_{0, \mathbf{k}+1} = \bar{\mathbf{x}}_{\mathbf{k}} \quad (4.21)$$

$$\boldsymbol{\chi}_{i, \mathbf{k}+1} = \bar{\mathbf{x}}_{\mathbf{k}} + \left(\sqrt{(n + \kappa) \mathbf{P}_{\mathbf{k}}}\right)_i \quad (4.22)$$

$$\boldsymbol{\chi}_{i+n, \mathbf{k}+1} = \bar{\mathbf{x}}_{\mathbf{k}} - \left(\sqrt{(n + \kappa) \mathbf{P}_{\mathbf{k}}}\right)_i \quad (4.23)$$

$$W_0 = \frac{\kappa}{n + \kappa} \quad (4.24)$$

$$W_i = \frac{1}{2(n + \kappa)} \quad (4.25)$$

$$W_{i+n} = \frac{1}{2(n + \kappa)} \quad (4.26)$$

In the UKF, the sigma points are propagated through the nonlinear function described by (4.27) and the mean and covariance of the state vector $\mathbf{x}_{\mathbf{k}+1}$ are approximated using a weighted mean and covariance, as shown in (4.28)-(4.29).

$$\gamma_{i,\mathbf{k}+1} = \mathbf{f}(\chi_{i,\mathbf{k}+1}, \mathbf{u}_{\mathbf{k}}), \quad \text{for } i = 1, \dots, n \quad (4.27)$$

$$\bar{\mathbf{x}}_{\mathbf{k}+1} \approx \sum_{i=1}^{2n+1} W_i \gamma_{i,\mathbf{k}+1} \quad (4.28)$$

$$\mathbf{P}, \mathbf{k} + 1 \approx \sum_{i=1}^{2n+1} W_i (\gamma_{i,\mathbf{k}+1} - \bar{\mathbf{x}}_{\mathbf{k}+1}) (\gamma_{i,\mathbf{k}+1} - \bar{\mathbf{x}}_{\mathbf{k}+1})^T \quad (4.29)$$

The same process is applied to the update stage to compute the measurement mean $\hat{\mathbf{z}}_{\mathbf{k}+1}$, from the $\Upsilon_{i,\mathbf{k}+1}$ points, as can be seen in the complete UKF algorithm in Table 4.

Similar to the EKF implementation, AVALON's UKF has the same state vector $\mathbf{x} = [\mathbf{p}_i, \mathbf{v}_i, \mathbf{q}]^T$, with the propagation model (4.14)-(4.16) and updated with GPS, magnetometer and barometric data. However, the Unscented Transformation cannot be directly applied to a quaternion because the quaternion is a member of a homogeneous Riemannian manifold and not of a vector space, thus, its mean is not a sum over all elements of a set divided by the size of the set (KRAFT, 2003).

Such problem can be solved with the intrinsic gradient descent algorithm (KRAFT, 2003; PENNEC, 1998), which computes the mean and covariance based on error vectors of quaternions. For the UKF, the set of error vectors $\mathbf{e}_i$ are iteratively found with (4.30), where $\gamma_{q,i,\mathbf{k}+1}$ are the quaternions of the propagated sigma points and $\bar{\mathbf{q}}$ is an initial estimated mean of this set. From one set of error vectors, its mean is calculated with the barycentric mean in (4.31), where $\mathbf{e}_{\mathbf{r},i}$ is the rotation vector from $\mathbf{e}_i = [e_{1,i}, e_{2,i}, e_{3,i}, e_{4,i}]^T$, defined in (4.32).

$$\mathbf{e}_i = \gamma_{q,i,\mathbf{k}+1} \otimes \bar{\mathbf{q}}^{-1} \quad \text{for } i = 1, \dots, n \quad (4.30)$$

Table 4 – Unscented Kalman Filter Algorithm

Prediction
$\chi_{k+1} = [\mathbf{x}_k \quad \mathbf{x}_k \pm \sqrt{(n + \kappa)(\mathbf{P}_k + \mathbf{Q})}]$ $\gamma_{i,k+1} = \mathbf{f}(\chi_{i,k+1}, \mathbf{u}_k) \quad \text{for } i = 1, \dots, n$ $\mathbf{x}_{k+1}^- = \sum_{i=1}^{2n+1} W_i \gamma_{i,k+1}$ $\mathbf{P}_{k+1}^- = \sum_{i=1}^{2n+1} W_i (\gamma_{i,k+1} - \mathbf{x}_{k+1}^-) (\gamma_{i,k+1} - \mathbf{x}_{k+1}^-)^T$
Update
$\Upsilon_{k+1} = \mathbf{h}(\gamma_{k+1}) \quad \text{for } i = 1, \dots, n$ $\bar{\mathbf{z}}_{k+1} = \sum_{i=1}^{2n+1} W_i \Upsilon_{i,k+1}$ $\mathbf{P}_{\mathbf{xz},k+1} = \sum_{i=1}^{2n+1} W_i (\gamma_{i,k+1} - \mathbf{x}_{k+1}^-) (\Upsilon_{i,k+1} - \bar{\mathbf{z}}_{k+1})^T$ $\mathbf{P}_{\mathbf{zz},k+1} = \mathbf{R} + \sum_{i=1}^{2n+1} W_i (\Upsilon_{i,k+1} - \bar{\mathbf{z}}_{k+1}) (\Upsilon_{i,k+1} - \bar{\mathbf{z}}_{k+1})^T$ $\mathbf{K}_{k+1} = \mathbf{P}_{\mathbf{xz},k+1} \mathbf{P}_{\mathbf{zz},k+1}^{-1}$ $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \bar{\mathbf{z}}_{k+1})$ $\mathbf{P}_{k+1} = \mathbf{P}_{k+1}^- - \mathbf{K}_{k+1} \mathbf{P}_{\mathbf{zz},k+1} \mathbf{K}_{k+1}^T$

Source: Elaborated by the author.

$$\bar{\mathbf{e}}_{\mathbf{r}} = \frac{1}{n} \sum_{i=1}^n \mathbf{e}_{\mathbf{r},i} \quad (4.31)$$

$$\mathbf{e}_{\mathbf{r},i} = 2 \tan^{-1} \left(\frac{\sqrt{e_{2,i}^2 + e_{3,i}^2 + e_{4,i}^2}}{e_{1,i}} \right) \begin{bmatrix} \frac{e_{2,i}}{\sqrt{e_{2,i}^2 + e_{3,i}^2 + e_{4,i}^2}} \\ \frac{e_{3,i}}{\sqrt{e_{2,i}^2 + e_{3,i}^2 + e_{4,i}^2}} \\ \frac{e_{4,i}}{\sqrt{e_{2,i}^2 + e_{3,i}^2 + e_{4,i}^2}} \end{bmatrix} \quad \text{for } i = 1, \dots, n \quad (4.32)$$

The error vector represents the rotation between the quaternion of the propagated sigma points and the estimated quaternion mean. Therefore, the error vector mean can be seen as the deviation between the estimated quaternion mean and the real mean. A better estimate quaternion mean can be computed with (4.33) from $\bar{\mathbf{e}}$, which is the quaternion representation of $\bar{\mathbf{e}}_{\mathbf{r}} = [\bar{e}_{1,r}, \bar{e}_{2,r}, \bar{e}_{3,r}]^T$ obtained with (4.34). Then, the new

estimate quaternion mean is fed back to create a new set of error vectors until $\bar{e}$ is close to 0, satisfying a defined precision.

$$\bar{\mathbf{q}}_{\mathbf{n}} = \bar{\mathbf{e}} \otimes \bar{\mathbf{q}} \quad (4.33)$$

$$\bar{\mathbf{e}} = \begin{bmatrix} \cos(0.5\|\bar{\mathbf{e}}_{\mathbf{r}}\|) \\ \frac{\sin(0.5\|\bar{\mathbf{e}}_{\mathbf{r}}\|)}{\|\bar{\mathbf{e}}_{\mathbf{r}}\|} \bar{e}_{1,r} \\ \frac{\sin(0.5\|\bar{\mathbf{e}}_{\mathbf{r}}\|)}{\|\bar{\mathbf{e}}_{\mathbf{r}}\|} \bar{e}_{2,r} \\ \frac{\sin(0.5\|\bar{\mathbf{e}}_{\mathbf{r}}\|)}{\|\bar{\mathbf{e}}_{\mathbf{r}}\|} \bar{e}_{3,r} \end{bmatrix} \quad (4.34)$$

The final set of error vectors can also be used to determine the quaternion part of the cross correlation matrix $\mathbf{P}_{\mathbf{xz},\mathbf{k}+1}$ as (4.35), because it represents the rotation that turns the orientation part of $\mathbf{x}_{\mathbf{k}+1}^-$ into $\gamma_{i,\mathbf{k}+1}$.

$$\mathbf{P}_{\mathbf{q},\mathbf{xz},\mathbf{k}+1} = \sum_{i=1}^{2n+1} (\mathbf{e}_{\mathbf{r},i}) (\boldsymbol{\gamma}_{i,\mathbf{k}+1} - \bar{\mathbf{z}}_{\mathbf{k}+1})^T \quad (4.35)$$

This result in a $\mathbf{P}_{\mathbf{k}+1}$ that depends on the error vector instead of quaternions. So in the next step of the filter, the part of the covariance that relates the quaternions are changed back to its quaternion form and multiplied by the quaternion of the estimated state vector to define its sigma points, as can be seen in (4.36).

$$\boldsymbol{\chi}_{q,\mathbf{k}+1} = \begin{bmatrix} \mathbf{x}_{\mathbf{q},\mathbf{k}} & \mathbf{x}_{\mathbf{k}} \otimes \pm \left(\sqrt{(n+\kappa)(\mathbf{P}_{\mathbf{k}} + \mathbf{Q})} \right)_{q,i} \end{bmatrix} \quad (4.36)$$

With the addition of these methods for the quaternion part, the prediction and update stages of the UKF are completely defined for AVALON's Sensor Unit. Its validation, along with EKF's, are presented in the next section.

4.5 Simulation and comparison results

The EKF and UKF implementations of AVALON's Sensor Unit are validated with simulations of the entire operation with a fixed wind and different configurations of sensor noise. The simulation environment was constructed with the kinematic and

dynamic models described in Section 3.3, with the same sensor noise generated offline and added to each measure model presented in Section 4.2.

Both methods consider sensor data from a GPS and an IMU that contains three accelerometers, three gyroscopes, three magnetometers and one barometer. The standard deviation used to generate each sensor noise were retrieved from data sheets of the collection of sensor chosen for AVALON (IMU ADIS16407 and GPS LS20031) and are presented in Table 5 (ANALOG DEVICES, 2011; LOCOSYS, 2009).

Table 5 – Standard deviation of accelerometers, gyroscopes, magnetometers, barometer and GPS used in the simulation environment to generate sensor noise.

Sensor	Standard deviation
Accelerometers	0.0882 m/s ²
Gyroscopes	0.8 °/s
Magnetometers	0.00006 G
Barometer	0.00027 N/m ²
GPS in north and east axes	2.5 m
GPS altitude	4 m

Source: Research data.

The noise covariances of the GPS, magnetometer and barometer used in the update stage are the same for both EKF and UKF, as seen in (4.37)-(4.38). The values for GPS and barometer are directly obtained from the data sheet. However, the value of magnetometer depends on the previous estimates, then it was adjusted by trial and error.

$$\mathbf{R}_{\text{GPS}} = \begin{bmatrix} 2.5^2 & 0 & 0 \\ 0 & 2.5^2 & 0 \\ 0 & 0 & 4^2 \end{bmatrix} \quad (4.37)$$

$$\mathbf{R}_{\text{Mag,Bar}} = \begin{bmatrix} 0.01024135^2 & 0 \\ 0 & 10^2 \end{bmatrix} \quad (4.38)$$

The process noise covariance is different for EKF and UKF because each algorithm has a specific function for it. For EKF, the values are directly related to the accelerometers and gyroscopes noise, as represented in (4.39). For UKF, the values that have given the smallest errors were insignificant, so the covariance is the 9×9 zero matrix, $\mathbf{O}_{9 \times 9}$.

$$\mathbf{Q} = \begin{bmatrix} 0.0882^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0882^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0882^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8^2 \end{bmatrix} \quad (4.39)$$

At the simulations, the aircraft starts at the position $\mathbf{p}_{i,initial} = [0, 0, 0]^T$, follows a path defined by 6 waypoints and starts the landing procedure after the last waypoint. The waypoints in the inertial frame that define the path are $[0, 0, -20]$, $[300, 0, -20]$, $[300, 300, -20]$, $[0, 300, -20]$, $[0, 0, -20]$, $[300, 0, -20]$. For all replications, the DI with PID structure from Control Unit stabilises the aircraft during all manoeuvres and Vector Field path-following is applied in the Navigation Unit.

The comparison was performed with 30 replications, each one with its own configuration of noise generated offline for all sensors and used for both EKF and UKF. The resulting paths of the first replication are illustrated in Figure 12, which shows the results for EKF and UKF for a fixed wind velocity of 0.4521 m/s in the north direction. It is possible to observe that the estimates provided by both algorithms allow the correct execution of all flight stages.

Figure 12 – Positions estimates from EKF and UKF in the inertial frame from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The path is defined by 6 waypoints (red points), whose sequence is represented by the red numbers.

Source: Research data.

Figures 13, 14 and 15 present the positions errors from EKF and UKF estimates for the first replication. The estimates from UKF vary more than the estimates from EKF, however, both errors are bounded between acceptable values that allowed the aircraft to follow the reference path. The increased error in the north position is caused by the wind, which is constant and whose influence is not captured by the accelerometers. The smallest errors in the up position happen because there are three sources of altitude information: accelerometers, GPS and barometer data.

Figure 13 – EKF and UKF north position errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

The velocities errors in the inertial frame are represented by Figures 16, 17 and 18. The variation of the UKF estimates are larger than the EKF estimates, as shown in the positions errors, but are also bounded.

Figures 19, 20 and 21 show the attitude error as Euler angles to facilitate its physical interpretation. In order to avoid the singularities for the error calculation in the plots, the pitch around $[\pm 87^\circ, \pm 90^\circ]$ is considered $\pm 90^\circ$ with a 0° roll and a $\pm 2 \tan^{-1}(q_2/q_1)$ yaw (BAKER, 2017). EKF errors of roll and yaw have a peak around 76 s due to the proximity to the singularity, which does not happen for UKF. During the takeoff, TVH and horizontal flight, the attitude errors are lower than 1° , but the error of roll and yaw increases during the THV and landing because of the proximity to the singularity.

In addition to the positions, velocities and attitude, the Control Unit requires estimates of the angle of attack and angle of sideslip, which are calculated from the state

Figure 14 – EKF and UKF east position errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 15 – EKF and UKF up position errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 16 – EKF and UKF north velocity errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 17 – EKF and UKF east velocity errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 18 – EKF and UKF up velocity errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 19 – EKF and UKF roll errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 20 – EKF and UKF pitch errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 21 – EKF and UKF yaw errors from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

vector estimates as (4.40) and (4.41), respectively. The resulting errors of the angle of attack, angle of sideslip and velocity magnitude obtained from the state vector of EKF and UKF are shown in Figures 22, 23 and 24.

$$\alpha = \frac{W}{U} \quad (4.40)$$

$$\beta = \frac{V}{\|\mathbf{v}_i\|} \quad (4.41)$$

Figure 22 – Angle of attack errors obtained from the estimates of EKF and UKF from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

The errors of angle of attack and angle of sideslip are higher at the initial time, close to the landing and around 76 s (it is the moment during the landing that the aircraft reaches the maximum altitude and starts to descend) due to the low velocities. However, these errors are corrected after some time due to the increase of the velocity in the X axis during the takeoff and landing. The errors of the velocity magnitude are consequences of the sum of each element error that composes the velocity vector in the inertial frame, but are still bounded between ± 1 m/s.

For the comparison between EKF and UKF estimates, it was used the mean and confidence intervals with a 95% confidence level of the errors of each element of the state

Figure 23 – Angle of sideslip errors obtained from the estimates of EKF and UKF from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 24 – Velocity magnitude errors obtained from the estimates of EKF and UKF from the first replication with a fixed wind of 0.4521 m/s in the north direction, DI with PID structure and Vector Field as path-following algorithm. The filled circles represent a change in the flight stage.

Source: Research data.

vector from the 30 replications of the entire AVALON's operation for each sensor fusion technique. The replications guarantee the statistical viability of the data and allow a correct analysis under different configurations of sensor noise (JAIN, 1991). Figures 25, 26 and 27 show the mean and confidence intervals for the positions errors, velocities errors and attitude angles errors respectively.

Figure 25 – Positions errors mean and confidence intervals for EKF and UKF and 30 replications of the entire AVALON's operation with different configurations of sensor noise.

Source: Research data.

The positions errors in the north and up directions are smaller for UKF, while the position error in the east direction is smaller for EKF. In the case of velocities, EKF errors are all smaller than UKF errors. However, the angles errors for UKF are significantly smaller than EKF errors, which also present considerably larger confidence intervals, showing a great variation among the replications.

For AVALON's operation, the importance of the angles errors prevails over the other states, since the manoeuvres depend completely on the estimation of the attitude, which is directly controlled by the Control Unit. In order to emphasise this importance and to allow a better comparison between the sensor fusion techniques, a metric was defined as (4.42), where E_{P_N} , E_{P_E} and E_{P_U} are the mean of the absolute errors of the positions in the north, east and up directions, E_{V_N} , E_{V_E} and E_{V_U} are the mean of the absolute errors of the velocities in the north, east and up directions and E_ϕ , E_θ and E_ψ are the mean of the absolute errors of the attitude as Euler angles. Each element that composes the metric is divided by its expected values as a normalisation procedure. This metric was calculated for each replication and the mean and confidence interval for all

Figure 26 – Velocities errors mean and confidence interval for EKF and UKF and 30 replications of the entire AVALON's operation with different configurations of sensor noise.

Source: Research data.

Figure 27 – Attitude angles errors mean and confidence interval for EKF and UKF and 30 replications of the entire AVALON's operation with different configurations of sensor noise.

Source: Research data.

replications are represented by Figure 28.

$$M_{\text{sensor}} = \frac{E_{P_N}}{12} + \frac{E_{P_E}}{1.5} + \frac{E_{P_U}}{1} + \frac{E_{V_N}}{0.5} + \frac{E_{V_E}}{0.5} + \frac{E_{V_U}}{0.1} + \frac{E_{\phi}}{0.15} + \frac{E_{\theta}}{0.15} + \frac{E_{\psi}}{0.15} \quad (4.42)$$

Figure 28 – Sensor metric mean and confidence interval for EKF and UKF and 30 replications of the entire AVALON’s operation with different configurations of sensor noise.

Source: Research data.

As Figure 28 shows, EKF metric is notably larger than UKF metric, which means that UKF provides better estimates when considering the importance of each element in the state vector. Also, the difference between the confidence interval of the two metrics shows a greater variance of the EKF results, therefore, UKF results are more reliable.

4.6 Final remarks

The observation models for each sensor of AVALON were presented, along with EKF and UKF implementations that provide estimates about the position, velocities and attitude of the aircraft. A comparison procedure was established using a simulation environment of AVALON’s entire operation. The comparison was also comprised of replications based on different configurations of sensor noise generated offline and applied equally for both methods. Its results show that EKF provides better results for velocities estimates, however, UKF produced smaller errors for the attitude and north and up positions. Since the attitude information is more critical to AVALON control, a metric was created using a normalisation of each error and its analysis emphasised the better performance of UKF.

In accordance with the comparison results, the UKF implementation was chosen to provide estimates to the Navigation Unit and Control Unit, detailed in the next chapters.

NAVIGATION

5.1 Chapter overview

One of the most important tasks regarding the navigation of a vehicle is the path-following, which allows the aircraft to follow a predefined path. Several approaches were created to solve this problem, usually based on geometric and control techniques (SUJIT; SARIPALLI; SOUSA, 2014). Aiming at the most adequate choice for AVALON scenario, four of them were implemented and compared: Lookahead (BREIVIK; FOSSEN, 2005), Pure Pursuit and Line-of-Sight (PLOS) (VENKATRAMAN *et al.*, 2010), Nonlinear Guidance Law (NLGL) (PARK; DEYST; HOW, 2007) and Vector Field (NELSON *et al.*, 2007). These implementations followed an extension process to consider not only the 2D scenario in which these algorithms were defined, but also the 3D scenario, more appropriate to an aircraft movement.

For AVALON, the path-following method is applied during the horizontal flight, with paths specified as straight lines that connect a list of waypoints with desired positions in the inertial frame. Thus, the path-following algorithms must provide reference attitude that result in manoeuvres that AVALON can perform to follow the path.

The Navigation Unit is also responsible to manage the flight stages and provides the reference attitude and velocity during all flight stages. The change of each flight stage is detected by continuously checking the measured position and attitude from the Sensor Unit. During the takeoff, the reference attitude and velocity have predefined values, but during the other flight stages they are determined by different techniques.

Since PID controllers are also used in the path-following algorithms, this chapter introduces their concepts and implementation used in the entire AVALON's autopilot. Afterwards, the main challenge of the Navigation Unit is treated, the path-following task. The path-following problem is established, each path-following algorithm resulted from

an extension process is presented and it is shown their simulation results and comparison. Finally, the formulations of the reference attitude and velocity to the other flight stages are presented.

5.2 Proportional-integral-derivative controller

The Proportional-Integral-Derivative (PID) controller is a feedback control method widely employed in several applications due to its flexibility. It allows systems to follow commands and reduce effects of disturbances from its feedback action, it can eliminate steady-state offsets through its integration action and anticipate the future through its derivative action ([ÅSTRÖM; HÄGGLUND, 2006](#)).

All actions from the PID controller are represented by (5.1), where $u(t)$ is the control variable, $e(t) = x_{sp} - x(t)$ is the error between the setpoint x_{sp} and the system state $x(t)$, K_P is the proportional gain, K_I is the integral gain and K_D is the derivative gain.

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (5.1)$$

The first part, called proportional, is a simple action proportional to the error. The second part, called integral, increases the action if the sum of the past errors is positive or decreases the action if it is negative, correcting possible steady-state errors. The last part is the derivative, which improves the stability and performance of the system response by predicting and anticipating the error behaviour. The higher the gains, more action is applied to the system, however, they can cause oscillations or make the system unstable, so balanced values should be used according to the characteristics of the system.

One of the problems of a PID controller is the integral windup, which occurs when a physical actuator reaches its limit and the controller actions do not represent the real actuation. While this actuator remains saturated, the integral term becomes very large and the error must change sign for a very long period before the integrator winds down. Therefore, to avoid this problem, the integral term must be limited to the actuators physical limits.

Another problem that should be considered is the high influence of measurement noise in the derivative term. This can be corrected by adding a first order filter to the derivative, resulting in (5.2) represented by the Laplace transformation of the PID, where T_f is the filter time constant. This change results in a derivative transfer function $K_D s$ for low frequencies and $\frac{K_D}{T_f}$ for high frequencies. The filter time constant is chosen as $N_{filter} K_D / K_P$, with N_{filter} usually in the range of 2 to 20 ([ÅSTRÖM; HÄGGLUND, 2006](#)).

$$u(s) = K_P e(s) + K_I \frac{e(s)}{s} + \frac{K_D e(s)s}{1 + sT_f} \quad (5.2)$$

The controller can be approximated to its discrete form to be implemented in digital systems, which results in (5.3)-(5.6), where $e(t_k)$ is the error at time t_k , $e(t_{k-1})$ is the error at the previous iteration and Δt_s is the time step.

$$u(t_k) = P_{action}(t_k) + I_{action}(t_k) + D_{action}(t_k) \quad (5.3)$$

$$P_{action}(t_k) = K_P e(t_k) \quad (5.4)$$

$$I_{action}(t_k) = I_{action}(t_{k-1}) + K_I e(t_k) \Delta t_s \quad (5.5)$$

$$D_{action}(t_k) = \frac{T_f D_{action}(t_{k-1})}{T_f + \Delta t_s} + \frac{K_D (e(t_k) - e(t_{k-1}))}{T_f + \Delta t_s} \quad (5.6)$$

Another specific problem of PID controller in the scenario of AVALON's autopilot occurs when dealing with Euler angles, whose error is not directly calculated as the difference between the reference angle and the measured angle due to its discontinuities at $\pm 2\pi$. Therefore, a different PID step is considered when the error is related to Euler angles, which is called *PID_Euler_step* in the path-following algorithms. This step calculates the errors by keeping the values in the interval $[-\pi, \pi]$.

Therefore, the discrete PID controllers comprised in AVALON's autopilot have anti-windup protection and derivative filters when the derivative gain is different from zero. They are used in the control structures of Control Unit, which is discussed in Chapter 6, in the controller of the reference velocity during the landing and in the path-following algorithms presented in the following sections.

5.3 Horizontal flight

The horizontal flight has the most complex path because it is longer and has a wider variation compared to the whole trajectory. The path is described as straight lines that connect the waypoints, but a fixed-wing aircraft cannot follow only straight lines. This happens as a result of its movement, that is always ahead, so the change in the velocity direction is accomplished by changes in the attitude. Consequently, the path-following algorithms must define the reference attitude that results in the change of the position that would direct the vehicle to the reference path.

This section presents the path-following problem and the resulting algorithms from the extension process. The results of the simulation are also shown as a comparison among all algorithms.

5.3.1 Path-following problem

Considering an aircraft with six degrees of freedom and given a reference path and the position of the vehicle in an inertial frame, it is possible to define the path error as the distance between the vehicle position and the reference path. A change in the aircraft attitude produces a change in the velocity vector, correcting its position. Therefore, the path-following problem is specified as the obtainment of a reference yaw and pitch that minimises the path error.

In the AVALON horizontal flight, the reference path is defined as straight lines connecting a list of waypoints W_i . The track of this path is accomplished with coordinated turn manoeuvres, in which yaw changes are obtained indirectly through changes of roll. The roll necessary to perform a curve, ϕ_{ref} , is specified by (5.7), where R_{ref} is the reference velocity of yaw, $\|\mathbf{v}_i\|$ is the velocity magnitude and g is the gravity acceleration.

$$\phi_{ref} = \tan^{-1} \left(\frac{R_{ref} \|\mathbf{v}_i\|}{g} \right) \quad (5.7)$$

Therefore, the path-following problem for AVALON can be specified as determining the reference yaw velocity R_{ref} and the reference pitch θ_{ref} to track a path defined by the straight line that connects the waypoints $\mathbf{W}_i$ and $\mathbf{W}_{i+1}$ with the aircraft position in the inertial frame $\mathbf{p}_i$ and its yaw ψ and pitch θ .

5.3.2 Path-following algorithms

New 3D path-following algorithms were designed with an extension of well-known 2D algorithms, Lookahead, NLGL, PLOS and Vector Field (PELIZER; SILVA; BRANCO, 2017a). The extension process involves the definition of two frames in addition to the inertial frame, a path frame and an intermediate frame. The path frame represents the frame whose X_p axis has the desired velocity direction that would keep the aircraft in the reference path (BREIVIK; FOSSEN, 2005).

The inertial frame, represented by axes X_N , Y_E and Z_D , is the same presented in Figure 5, an Earth-fixed tangential frame in local-level and tangent to the gravity. The intermediate frame, represented by axes X' , Y' and Z' , is the frame resulted from an ζ_ψ rotation in the Z_i axis of the inertial frame and the path frame, represented by axes X_p , Y_p and Z_p , is the result of a ζ_θ rotation in the Y' axis of the intermediate frame, as illustrated in Figure 29.

The error vector in the path frame is specified as (5.8), where $\mathbf{R}_Z(\zeta_\psi)$ is the rotation matrix from the inertial frame to the intermediate frame defined by (5.9) and $\mathbf{R}_{Y'}(\zeta_\theta)$ is the rotation matrix from the intermediate frame to the path frame defined by (5.10). The angles ζ_ψ and ζ_θ are calculated as (5.11) and (5.12) respectively, using

Figure 29 – Inertial, intermediate and path frames with their rotations along the Z_i of ζ_ψ and along the Y' of ζ_θ .

Source: Elaborated by the author.

the waypoints $\mathbf{W}_i = [x_i, y_i, z_i]^T$ and $\mathbf{W}_{i+1} = [x_{i+1}, y_{i+1}, z_{i+1}]^T$ that represent a straight line path. The waypoints can also be represented in the intermediate frame by $\mathbf{W}_i^R = \mathbf{R}_Z(\zeta_\psi) \mathbf{W}_i = [x_i^R, y_i^R, z_i^R]^T$ and $\mathbf{W}_{i+1}^R = \mathbf{R}_Z(\zeta_\psi) \mathbf{W}_{i+1} = [x_{i+1}^R, y_{i+1}^R, z_{i+1}^R]^T$.

$$\mathbf{e} = \mathbf{R}_{Y'}(\zeta_\theta) \mathbf{R}_Z(\zeta_\psi) (\mathbf{p} - \mathbf{W}_i) \quad (5.8)$$

$$\mathbf{R}_Z(\zeta_\psi) = \begin{pmatrix} \cos(\zeta_\psi) & -\sin(\zeta_\psi) & 0 \\ \sin(\zeta_\psi) & \cos(\zeta_\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.9)$$

$$\mathbf{R}_{Y'}(\zeta_\theta) = \begin{pmatrix} \cos(\zeta_\theta) & 0 & \sin(\zeta_\theta) \\ 0 & 1 & 0 \\ -\sin(\zeta_\theta) & 0 & \cos(\zeta_\theta) \end{pmatrix} \quad (5.10)$$

$$\zeta_\psi = \tan^{-1} \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right) \quad (5.11)$$

$$\zeta_\theta = \tan^{-1} \left(\frac{z_i^R - z_{i+1}^R}{\sqrt{(x_{i+1}^R - x_i^R)^2 + (y_{i+1}^R - y_i^R)^2}} \right) \quad (5.12)$$

The error vector components in the path frame from $\mathbf{e} = [e_x, e_y, e_z]^T$ are called along-track error (e_x), cross-track error (e_y) and vertical-track error (e_z) (BREIVIK; FOSSEN, 2005). All components can be seen in Figure 30.

The objectives of the new 3D path-following algorithms are to reduce the components cross-track error e_y and vertical-track error e_z . The reduction of these errors implies

Figure 30 – The error vector $\mathbf{e}$ in the path frame and its respective components e_x , e_y and e_z .

Source: Elaborated by the author.

in an approximation to the reference path defined by the straight lines. It is also important that the aircraft arrives at the reference path before it reaches the next waypoint, otherwise the convergence of the path-following algorithms cannot be guaranteed.

Lookahead, PLOS and Vector Field use PID controllers to compute their final references, but they all have different control laws and specific parameters. Each algorithm, which is the result of the extension process, are presented in the following subsections.

5.3.2.1 Lookahead

The Lookahead path-following algorithm uses the concept of virtual target to guarantee that the aircraft follows a reference path (BREIVIK; FOSSEN, 2005). The virtual target is set as a goal point to where the aircraft should be directed to. On the Lookahead approach, this point location is at a fixed distance, called lookahead distance. With the extension process, we can define two lookahead distances, one represented by δ_y relative to the $X_p Y_p$ plane and another one represented by δ_z relative to the $X_p Z_p$ plane.

The extended Lookahead algorithm is characterised by the control laws (5.13)-(5.14), which guarantee the convergence of the algorithm, where δ_y and δ_z respectively drive the position to the $X_p Y_p$ and $X_p Z_p$ planes. Then, the yaw velocity is obtained with a step of its PID controller. Algorithm 1 summarises all the steps of the Lookahead algorithm for AVALON.

$$\psi_{ref} = \zeta_\psi + \tan^{-1} \left(\frac{e_y}{\delta_y} \right) \quad (5.13)$$

$$\theta_{ref} = \zeta_\theta + \tan^{-1} \left(\frac{e_z}{\delta_z} \right) \quad (5.14)$$

Algorithm 1 – Lookahead algorithm.

```

1: function LOOKAHEAD( $\mathbf{W}_i = [x_i, y_i, z_i]^T$ ,  $\mathbf{W}_{i+1} = [x_{i+1}, y_{i+1}, z_{i+1}]^T$ ,  $\mathbf{p} = [x, y, z]^T$ ,  $\psi$ ,
    $\theta$ ,  $\delta_y$ ,  $\delta_z$ )
2:    $\zeta_\psi \leftarrow \tan^{-1} \left( \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right)$ 
3:    $\triangleright \mathbf{W}_i^R = [x_i^R, y_i^R, z_i^R]^T$ 
4:    $\mathbf{W}_i^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{W}_i$ 
5:    $\triangleright \mathbf{W}_{i+1}^R = [x_{i+1}^R, y_{i+1}^R, z_{i+1}^R]^T$ 
6:    $\mathbf{W}_{i+1}^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{W}_{i+1}$ 
7:    $\zeta_\theta \leftarrow \tan^{-1} \left( \frac{z_i^R - z_{i+1}^R}{\sqrt{(x_{i+1}^R - x_i^R)^2 + (y_{i+1}^R - y_i^R)^2}} \right)$ 
8:    $\triangleright \mathbf{e} = [e_x, e_y, e_z]^T$ 
9:    $\mathbf{e} \leftarrow \mathbf{R}_{Y'}(\zeta_\theta) \mathbf{R}_Z(\zeta_\psi) (\mathbf{p} - \mathbf{W}_i)$ 
10:  if ( $\zeta_\psi \geq 0$  and  $\zeta_\psi \leq \frac{\pi}{2}$ ) or ( $\zeta_\psi \geq -\pi$  and  $\zeta_\psi \leq \frac{-\pi}{2}$ ) or ( $\zeta_\psi == \pi$ ) then
11:     $e_y \leftarrow -e_y$ 
12:  end if
13:   $\psi_{ref} \leftarrow \zeta_\psi + \tan^{-1} \left( \frac{e_y}{\delta_y} \right)$ 
14:   $\theta_{ref} \leftarrow \zeta_\theta + \tan^{-1} \left( \frac{e_z}{\delta_z} \right)$ 
15:   $R_{ref} \leftarrow PID\_Euler\_step(\psi_{ref}, \psi)$ 
16:  return ( $R_{ref}, \theta_{ref}$ )
17: end function

```

5.3.2.2 Nonlinear guidance law

The NLGL also uses the concept of virtual target by calculating the interception between the path and a circle with a fixed radius from the aircraft position (PARK; DEYST; HOW, 2007; CURRY *et al.*, 2013). The control laws obtained from the extension process are (5.15)-(5.16), where L_ψ is the radius for the circle in the XY plane, L_θ is the radius for the circle in the $X'Z'$ plane, η_ψ is the angle that directs the aircraft to the path in the $X_p Y_p$ plane and η_θ is the angle that directs the aircraft to the path in the $X_p Z_p$ plane.

$$R_{ref} = 2 \frac{v^2}{L_\psi} \sin(\eta_\psi) \quad (5.15)$$

$$Q_{ref} = 2 \frac{v^2}{L_\theta} \sin(\eta_\theta) \quad (5.16)$$

In order to guarantee that the intersection between the path and the circles can always be found, it is included a countermeasure to the original method. When the cross-track error or vertical-track error are greater than their respective radius L_θ or L_ψ , the radius used in the intersection is the error plus 1. Also, to compute the reference pitch, an

approximation is used by propagating the computed velocity Q_{ref} with the Euler method. The complete algorithm is detailed in Algorithm 2.

5.3.2.3 Pure pursuit and line-of-sight

The Pure Pursuit and Line-of-Sight (PLOS) algorithm is composed of two strategies. The first, Line-of-Sight (LOS) guidance, is responsible for decreasing the position error with respect to the reference path, while the second one, pure pursuit, drives the aircraft to the next waypoint by adjusting the angle direction (VENKATRAMAN *et al.*, 2010).

In the extension process, both strategies are adapted, considering the cross-track error and vertical-track error for the LOS part and the desired angles as ζ_ψ and ζ_θ for the pure pursuit part. The resulting control structure uses four PID controllers, two that together give R_{ref} for the convergence in the $X_P Y_P$ plane and two others that together give θ_{ref} for the convergence in the $X_P Z_P$ plane. The errors for the R_{ref} PIDs are $\psi_d - \psi$ for the pure pursuit strategy and e_y for the LOS strategy, while the errors for the θ_{ref} PIDs are $\theta_d - \theta$ for the pure pursuit strategy and e_z for the LOS strategy. Algorithm 3 describes all calculations of the extended PLOS algorithm.

In the extended PLOS algorithm, one strategy can predominate over the other, according to the relation between the pure pursuit gain and the LOS gain. For this reason, besides finding values that give satisfactory responses, it is also important to find a balance between the gains.

5.3.2.4 Vector field

The Vector Field path-following algorithm uses the notion of vectors that are directed towards the reference path and indicate the desired direction of flight. The original Vector Field algorithm uses sliding mode approach to define the path desired course and its authors demonstrate the convergence of the control law in the presence of wind disturbances with Lyapunov stability arguments (NELSON *et al.*, 2007).

The resulting control laws from the extension process for the Vector Field algorithm are (5.17) and (5.18), where χ_ψ^∞ and χ_θ^∞ are the maximum angles of approach to the path, ϵ_ψ and ϵ_θ are the width of the transition region around the sliding surface that reduces the chatter in the control, s_ψ and s_θ are positive constants that influence the rate of transition of the desired angle from $\chi_{\psi,\theta}^\infty$ to 0, $\tilde{\chi}_\psi$ and $\tilde{\chi}_\theta$ are the desired course of the vector field defined, respectively, by (5.19) for the $X_P Y_P$ plane and by (5.20) for the $X_P Z_P$ plane, and $sat(x)$ is the saturation function defined in (5.21) that avoids adverse affects of chattering. The reference yaw velocity R_{ref} is calculated with a PID controller, as can be seen in Algorithm 4.

Algorithm 2 – Nonlinear Guidance Law algorithm.

```

1: function NLGL( $\mathbf{W}_i = [x_i, y_i, z_i]^T$ ,  $\mathbf{W}_{i+1} = [x_{i+1}, y_{i+1}, z_{i+1}]^T$ ,  $\mathbf{p} = [x, y, z]^T$ ,  $\psi$ ,  $\theta$ ,  $v$ ,
    $L_\psi$ ,  $L_\theta$ ,  $\Delta t_f$ )
2:    $\zeta_\psi \leftarrow \tan^{-1} \left( \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right)$ 
3:    $\triangleright \mathbf{W}_i^R = [x_i^R, y_i^R, z_i^R]^T$ 
4:    $\mathbf{W}_i^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{W}_i$ 
5:    $\triangleright \mathbf{W}_{i+1}^R = [x_{i+1}^R, y_{i+1}^R, z_{i+1}^R]^T$ 
6:    $\mathbf{W}_{i+1}^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{W}_{i+1}$ 
7:    $\triangleright \mathbf{p}^R = [x^R, y^R, z^R]^T$ 
8:    $\mathbf{p}^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{p}$ 
9:    $\zeta_\theta \leftarrow \tan^{-1} \left( \frac{z_i^R - z_{i+1}^R}{\sqrt{(x_{i+1}^R - x_i^R)^2 + (y_{i+1}^R - y_i^R)^2}} \right)$ 
10:   $\triangleright \mathbf{e} = [e_x, e_y, e_z]^T$ 
11:   $\mathbf{e} \leftarrow \mathbf{R}_{Y'}(\zeta_\theta) \mathbf{R}_Z(\zeta_\psi) (\mathbf{p} - \mathbf{W}_i)$ 
12:  if  $e_y \geq L_\psi$  then
13:     $\triangleright$  Calculates intersection between the circle of radius  $\|\mathbf{e}_y\| + 1$  and the reference
    path in the  $XY$  plane
14:     $\triangleright \mathbf{p}_{vp,\psi} = [x_{vp,\psi}, y_{vp,\psi}, z_{vp,\psi}]^T$ 
15:     $\mathbf{p}_{vp,\psi} \leftarrow \text{intersection}(\|\mathbf{e}_y\| + 1, [x_i, y_i], [x_{i+1}, y_{i+1}])$ 
16:  else
17:     $\triangleright$  Calculates intersection between circle of radius  $L_\psi$  and the reference path in
    the  $XY$  plane
18:     $\triangleright \mathbf{p}_{vp,\psi} = [x_{vp,\psi}, y_{vp,\psi}, z_{vp,\psi}]^T$ 
19:     $\mathbf{p}_{vp,\psi} \leftarrow \text{intersection}(L_\psi, [x_i, y_i], [x_{i+1}, y_{i+1}])$ 
20:  end if
21:  if  $e_z \geq L_\theta$  then
22:     $\triangleright$  Calculates intersection between the circle of radius  $\|\mathbf{e}_z\| + 1$  and the reference
    path in the  $X'Z'$  plane
23:     $\triangleright \mathbf{p}_{vp,\theta} = [x_{vp,\theta}, y_{vp,\theta}, z_{vp,\theta}]^T$ 
24:     $\mathbf{p}_{vp,\theta} \leftarrow \text{intersection}(\|\mathbf{e}_z\| + 1, [x_i^R, z_i^R], [x_{i+1}^R, z_{i+1}^R])$ 
25:  else
26:     $\triangleright$  Calculates intersection between the circle of radius  $L_\theta$  and the reference path
    in the  $X'Z'$  plane
27:     $\triangleright \mathbf{p}_{vp,\theta} = [x_{vp,\theta}, y_{vp,\theta}, z_{vp,\theta}]^T$ 
28:     $\mathbf{p}_{vp,\theta} \leftarrow \text{intersection}(L_\theta, [x_i^R, z_i^R], [x_{i+1}^R, z_{i+1}^R])$ 
29:  end if
30:   $\sigma_\psi \leftarrow \tan^{-1} \left( \frac{y_{vt,\psi} - y}{x_{vt,\psi} - x} \right)$ 
31:   $\sigma_\theta \leftarrow \tan^{-1} \left( \frac{z^R - z_{vt,\theta}}{x_{vt,\theta} - x^R} \right)$ 
32:   $\eta_\psi \leftarrow \psi - \sigma_\psi$ 
33:   $\eta_\theta \leftarrow \theta - \sigma_\theta$ 
34:   $R_{ref} \leftarrow 2 \frac{v^2}{L_\psi} \sin(\eta_\psi)$ 
35:   $Q_{ref} \leftarrow 2 \frac{v^2}{L_\theta} \sin(\eta_\theta)$ 
36:   $\theta_{ref} \leftarrow \theta + Q_{ref} \Delta t_f$ 
37:  return ( $R_{ref}, \theta_{ref}$ )
38: end function

```

Algorithm 3 – Pure Pursuit and Line-of-Sight algorithm.

```

1: function PLOS( $\mathbf{W}_i = [x_i, y_i, z_i]^T$ ,  $\mathbf{W}_{i+1} = [x_{i+1}, y_{i+1}, z_{i+1}]^T$ ,  $\mathbf{p} = [x, y, z]^T$ ,  $\psi$ ,  $\theta$ )
2:    $\zeta_\psi \leftarrow \tan^{-1} \left( \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right)$ 
3:    $\triangleright \mathbf{W}_i^R = [x_i^R, y_i^R, z_i^R]^T$ 
4:    $\mathbf{W}_i^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{W}_i$ 
5:    $\triangleright \mathbf{W}_{i+1}^R = [x_{i+1}^R, y_{i+1}^R, z_{i+1}^R]^T$ 
6:    $\mathbf{W}_{i+1}^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{W}_{i+1}$ 
7:    $\triangleright \mathbf{p}^R = [x^R, y^R, z^R]^T$ 
8:    $\mathbf{p}^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{p}$ 
9:    $\zeta_\theta \leftarrow \tan^{-1} \left( \frac{z_i^R - z_{i+1}^R}{\sqrt{(x_{i+1}^R - x_i^R)^2 + (y_{i+1}^R - y_i^R)^2}} \right)$ 
10:   $\triangleright \mathbf{e} = [e_x, e_y, e_z]^T$ 
11:   $\mathbf{e} \leftarrow \mathbf{R}_{Y'}(\zeta_\theta) \mathbf{R}_Z(\zeta_\psi) (\mathbf{p} - \mathbf{W}_i)$ 
12:  if  $(\zeta_\psi \geq 0 \text{ and } \zeta_\psi \leq \frac{\pi}{2})$  or  $(\zeta_\psi \geq -\pi \text{ and } \zeta_\psi \leq \frac{-\pi}{2})$  or  $(\zeta_\psi == \pi)$  then
13:     $e_y \leftarrow -e_y$ 
14:  end if
15:   $\psi_{ref} \leftarrow \tan^{-1} \left( \frac{y_{i+1} - y}{x_{i+1} - x} \right)$ 
16:   $\theta_{ref} \leftarrow \tan^{-1} \left( -\frac{z_{i+1}^R - z^R}{x_{i+1}^R - x^R} \right)$ 
17:   $R_{ref} \leftarrow PID\_Euler\_step(\psi_{ref}, \psi) + PID\_step(e_y, 0)$ 
18:   $\theta_{ref} \leftarrow -PID\_Euler\_step(\theta_{ref}, \theta) + PID\_step(e_z, 0)$ 
19:  return  $(R_{ref}, \theta_{ref})$ 
20: end function

```

$$\psi_{ref} = (\psi - \zeta_\psi) - \frac{\chi_\psi^\infty}{k_\psi} \frac{2}{\pi} \frac{s_\psi v \sin(\psi - \zeta_\psi)}{1 - (s_\psi e_y)^2} - \frac{\kappa_\psi}{k_\psi} \text{sat} \left(\frac{\tilde{\chi}_\psi}{\epsilon_\psi} \right) \quad (5.17)$$

$$\theta_{ref} = (\theta - \zeta_\theta) - \frac{\chi_\theta^\infty}{k_\theta} \frac{2}{\pi} \frac{s_\theta v \sin(\theta)}{1 - (s_\theta e_z)^2} - \frac{\kappa_\theta}{k_\theta} \text{sat} \left(\frac{\tilde{\chi}_\theta}{\epsilon_\theta} \right) \quad (5.18)$$

$$\tilde{\chi}_\psi = (\psi - \zeta_\psi) - \left(-\chi_\psi^\infty \frac{2}{\pi} \tan^{-1}(s_\psi e_y) \right) \quad (5.19)$$

$$\tilde{\chi}_\theta = (\theta - \zeta_\theta) - \left(-\chi_\theta^\infty \frac{2}{\pi} \tan^{-1}(s_\theta e_z) \right) \quad (5.20)$$

$$\text{sat}(x) = \begin{cases} x, & \text{if } |x| \leq 1 \\ 1, & \text{if } |x| > 1 \text{ and } x > 0 \\ 0, & \text{if } |x| > 1 \text{ and } x = 0 \\ -1, & \text{if } |x| > 1 \text{ and } x < 0 \end{cases} \quad (5.21)$$

Algorithm 4 – Vector Field algorithm.

```

1: function VECTOR_FIELD( $\mathbf{W}_i = [x_i, y_i, z_i]^T$ ,  $\mathbf{W}_{i+1} = [x_{i+1}, y_{i+1}, z_{i+1}]^T$ ,  $\mathbf{p} = [x, y, z]^T$ ,  $\psi$ ,  $\theta$ ,  $v$ ,  $\chi_\psi^\infty$ ,  $k_\psi$ ,  $s_\psi$ ,  $\kappa_\psi$ ,  $\epsilon_\psi$ ,  $\chi_\theta^\infty$ ,  $k_\theta$ ,  $s_\theta$ ,  $\kappa_\theta$ ,  $\epsilon_\theta$ )
2:    $\zeta_\psi \leftarrow \tan^{-1} \left( \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right)$ 
3:    $\triangleright \mathbf{W}_i^R = [x_i^R, y_i^R, z_i^R]^T$ 
4:    $\mathbf{W}_i^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{W}_i$ 
5:    $\triangleright \mathbf{W}_{i+1}^R = [x_{i+1}^R, y_{i+1}^R, z_{i+1}^R]^T$ 
6:    $\mathbf{W}_{i+1}^R \leftarrow \mathbf{R}_z(\zeta_\psi) \mathbf{W}_{i+1}$ 
7:    $\zeta_\theta \leftarrow \tan^{-1} \left( \frac{z_i^R - z_{i+1}^R}{\sqrt{(x_{i+1}^R - x_i^R)^2 + (y_{i+1}^R - y_i^R)^2}} \right)$ 
8:    $\triangleright \mathbf{e} = [e_x, e_y, e_z]^T$ 
9:    $\mathbf{e} \leftarrow \mathbf{R}_{Y'}(\zeta_\theta) \mathbf{R}_z(\zeta_\psi) (\mathbf{p} - \mathbf{W}_i)$ 
10:   $\psi_R \leftarrow \psi - \zeta_\psi$ 
11:   $\theta_R \leftarrow \theta - \zeta_\theta$ 
12:   $\tilde{\chi}_\psi \leftarrow \psi_R - \left( -\chi_\psi^\infty \frac{2}{\pi} \tan^{-1}(s_\psi e_y) \right)$ 
13:   $\tilde{\chi}_\theta \leftarrow \theta_R - \left( -\chi_\theta^\infty \frac{2}{\pi} \tan^{-1}(s_\theta e_z) \right)$ 
14:   $\psi_{ref} \leftarrow \psi_R - \frac{\chi_\psi^\infty}{k_\psi} \frac{2}{\pi} \frac{s_\psi v \sin(\psi)}{1 - (s_\psi e_y)^2} - \frac{\kappa_\psi}{k_\psi} \text{sat} \left( \frac{\tilde{\chi}_\psi}{\epsilon_\psi} \right)$ 
15:   $\theta_{ref} \leftarrow \theta_R - \frac{\chi_\theta^\infty}{k_\theta} \frac{2}{\pi} \frac{s_\theta v \sin(\theta)}{1 - (s_\theta e_z)^2} - \frac{\kappa_\theta}{k_\theta} \text{sat} \left( \frac{\tilde{\chi}_\theta}{\epsilon_\theta} \right)$ 
16:   $\psi_{ref} \leftarrow \psi_{ref} + \zeta_\psi$ 
17:   $\theta_{ref} \leftarrow \theta_{ref} + \zeta_\theta$ 
18:   $R_{ref} \leftarrow \text{PID\_Euler\_step}(\psi_{ref}, \psi)$ 
19:  return ( $R_{ref}, \theta_{ref}$ )
20: end function

```

The stability of the Vector Field algorithm is proved for $\chi_{\psi, \theta}^\infty \leq \pi/2$ and large values of it imply in faster convergence toward the path, so it can be set as $\pi/2$. Another parameter that can be set is $\epsilon_{\psi, \theta}$, which is only responsible for reducing the control chatter, so it can be set as 1. The other control parameters define the shape of the sliding surface and the resulting path and should be correctly chosen to avoid overshoot with large gains and slow convergence with small gains.

5.3.3 Simulation and comparison results

The validation and comparison of the extended path-following algorithms were performed with the kinematic and dynamic models of AVALON, like in the Sensor Fusion comparison. However, only the horizontal flight stage was considered with wind generated offline and applied to each path-following algorithm execution. All experiments use UKF implementation in the Sensor Unit to provide estimates for Navigation Unit and Control Unit with the sensor noises from the first replication from Section 4.5. Also, the PID with DI structure was used to stabilise the aircraft during the coordinated turn manoeuvres.

The wind is considered a constant disturbance that interferes directly in the air-

craft position, which justifies its variation in the path-following comparison. For each replication, a different intensity of wind in the north coordinate was generated offline and randomly with a variance of 1 m/s and kept constant during the entire trajectory.

As seen in the previous subsections, each path-following algorithm has its own characteristics and parameters. In order to improve the comparison fairness, all algorithms were optimised with two methods to find the most suitable parameters: Genetic Algorithm (GA) and Sequential Quadratic Programming (SQP).

The GA is an iterative process that uses the concepts of mutations, crossover and selection to find solutions for an optimisation problem. It evolves populations, which are possible solutions, by considering how fit they are in each iteration. The fitness is predetermined by an objective function that defines the problem itself. The advantages of GA as an optimisation algorithm is its ability to avoid multiple local optimum solutions by searching throughout a wide domain and the modularity of its implementation (MITCHELL, 1998).

The SQP is a nonlinear programming method that iteratively finds an optimised solution for a constrained problem. At each iteration, it models a nonlinear problem at a given approximated solution by a quadratic programming subproblem and computes its solutions. These solutions of the subproblem are then used to calculate a sequence of approximations that converge to a better solution (FLETCHER, 2013).

Both optimisation methods implementations are from MATLAB, through the functions *ga* and *fmincon*. The GA optimisation was run with a population size of 20 elements and a maximum of 20 generations and the SQP optimisation had a tolerance function of 10^{-5} as a stop criteria.

The objective function for all path-following algorithms was defined as (5.22), which has a part related to the error represented by (5.23) and a part related to the effort represented by (5.24). The error part contains the mean of the absolute error position for each coordinate of the inertial frame as E_{path,P_N} , E_{path,P_E} and E_{path,P_D} . The position error is characterised by the difference between the aircraft position and the projection of this position in the straight line represented as the reference path. The effort part contains the mean of the absolute actuator input as $\delta_{t,thr}$, $\delta_{t,ail}$, $\delta_{t,elev}$ and $\delta_{t,rud}$. Each element that composes the metrics is divided by its expected value as a normalisation and the error metric has a weight of 0.8, while the effort metric has a weight of 0.2. The higher weight of the error metric was determined because it was more important in providing a good convergence, however, the effort metric must also be considered to achieve feasible behaviours.

$$M_{path} = 0.8M_{path,error} + 0.2M_{path,effort} \quad (5.22)$$

$$M_{path,error} = \frac{E_{path,P_N}}{3} + \frac{E_{path,P_E}}{3} + \frac{E_{path,P_D}}{1} \quad (5.23)$$

$$M_{path,effort} = \frac{\delta_{t,thr}}{1} + \frac{\delta_{t,ail}}{10} + \frac{\delta_{t,elev}}{10} + \frac{\delta_{t,rud}}{10} \quad (5.24)$$

The main metric M_{path} was the objective function of GA to find good values of parameters for each algorithm. Afterwards, the GA optimised parameters were set as initial point to SQP with the same objective function, guaranteeing the minimisation of the metric. Table 6 presents the resulting parameters from these procedures and the controllers that resulted in the better performance were PD (Proportional-Derivative) and P controllers, where $K_{L,P}$, $K_{L,D}$ and N_L are the PD gains of Lookahead that has R_{ref} as output, $K_{PP,\psi,P}$, $K_{PP,\psi,D}$ and $N_{PP,\psi}$ are the PD gains of the Pure Pursuit part of PLOS for R_{ref} , $K_{LOS,\psi,P}$, $K_{LOS,\psi,D}$, $N_{LOS,\psi}$ are the PD gains of the LOS part of PLOS for R_{ref} , $K_{PP,\theta,P}$ is the P gain of the Pure Pursuit part of PLOS for θ_{ref} , $K_{LOS,\theta,P}$ is the P gain of the LOS part of PLOS for θ_{ref} and $K_{VF,P}$, $K_{VF,D}$, N_{VF} are the PD gains of Vector Field that has R_{ref} as output.

Table 6 – Parameters of all extended path-following algorithms resulted from GA and SQP optimisations with the objective function (5.22).

Algorithm	Parameters
Lookahead	$\delta_y = 19.32$, $\delta_z = 7.78$ $K_{L,P} = 2.13$, $K_{L,D} = 1.17$, $N_L = 2$
NLGL	$L_\psi = 121.73$, $L_\theta = 6.60$
PLOS	$K_{PP,\psi,P} = 0.86$, $K_{PP,\psi,D} = 5.54$, $N_{PP,\psi} = 2$ $K_{LOS,\psi,P} = 0.02$, $K_{LOS,\psi,D} = 6.90$, $N_{LOS,\psi} = 2$ $K_{PP,\theta,P} = 0.01$, $K_{LOS,\theta,P} = 0.16$
Vector Field	$k_\psi = 15.32$, $s_\psi = 0.07$, $\kappa_\psi = 0.90$ $k_\theta = 9.23$, $s_\theta = 0.10$, $\kappa_\theta = 9.94$ $K_{VF,P} = k_\psi$, $K_{VF,D} = 0.22$, $N_{VF} = 2$

Source: Research data.

Similar to the simulations used to compare the Sensor Fusion techniques, the simulations executed for the path-following algorithms also follow a path defined by 6 waypoints in the inertial frame: $[0, 0, -20]$, $[300, 0, -20]$, $[300, 300, -20]$, $[0, 300, -20]$, $[0, 0, -20]$, $[300, 0, -20]$. Therefore, the reference path is composed by the straight lines that connect each one of these waypoints.

Figure 31 shows the positions in the inertial-frame resulted from the first replication for all path-following algorithms. This first replication was executed with a fixed wind velocity of 0.4521 m/s in the north direction, UKF as Sensor Unit and the same configuration of sensor noise and DI with PID structure in Control Unit. Differently from the Sensor Unit, for the path-following validation, only the horizontal flight stage is considered.

Figure 31 – Positions in the inertial frame resulted from all path-following algorithms for the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit. The path is defined by 6 waypoints (red points), whose sequence is represented by the red numbers.

Source: Research data.

In order to elucidate the small differences among the results, the positions are individually represented in Figures 32, 33 and 34. It is expected that the north position increases until it reaches 300 m and stays at this value until the next waypoint. Then, it decreases until 0 m and keeps it again and finally increases to 300 m to the final waypoint. For the east position, it should be kept at 0 first, then it increases to 300 m and is kept until the fourth waypoint. Finally, it should decrease to 0 m. Also, the aircraft should lose altitude after the transition and maintain it at 20 meters during the entire path. Possible overshoots in north and east positions after achieving each waypoint are also expected because AVALON cannot change its direction instantaneously to correct the yaw, which is a physical constraint from fixed-wing aircraft.

Figure 32 – North positions of all path-following algorithms resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

Figure 33 – East positions of all path-following algorithms resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

Figure 34 – Up positions of all path-following algorithms resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

It is possible to observe that all algorithms executed the path correctly by directing the aircraft through all waypoints, however, NLGL results presented a considerable oscillation in the north and east positions. Regarding the time that each waypoint is achieved, all algorithms behaved similarly, with the exception of PLOS. The results from PLOS show that most waypoints are reached 1 second earlier, having a smaller rising time than the others, but it takes longer to stabilise with a higher settling time than Lookahead and Vector Field.

Moreover, in the up position, all algorithms have an oscillation after achieving the 20 m for the first time, but Lookahead overshoot is smaller. Although the loss of altitude during the curves is expected, Lookahead and Vector Field recovered faster and were able to keep the aircraft closer to the reference altitude during the path.

The complete comparison among the path-following algorithms was performed with the mean and confidence intervals with a 95% confidence level of the metrics $M_{path,error}$, $M_{path,effort}$ and M_{path} for 30 replications. Like in the Sensor Fusion techniques comparison, the replications guarantee the statistical viability of the data, and, for the path-following algorithms, allow a correct analysis of the results under different intensities of wind disturbances (JAIN, 1991).

During the execution of the replications, the autopilot was not able to perform the entire operation of the aircraft, resulting in unstable conditions. These failed executions

were not considered in the replications, but it is important to report that, in one of these failed executions, Vector Field succeeded while the others failed.

Figure 35 shows the results of the comparison for the error part of the metric, $M_{path,error}$. Since the means of Lookahead, PLOS and Vector Field are inside each other's confidence interval, they are statistically equal for this confidence level. However, these data does not provide enough information to compare them with NLGL because their confidence interval overlaps, requiring a new analysis with t-test for paired observations (JAIN, 1991).

Figure 35 – Error metric mean and confidence interval from all path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.

Source: Research data.

The T-test is executed by comparing the confidence interval of the error metric difference among NLGL and the other path-following algorithms for each replication. The mean and confidence intervals with a 95% confidence level of these differences are represented in Figure 36. The confidence interval of the difference between NLGL and PLOS contains zero, so they cannot be considered different with this confidence level. However, the same does not happen for Lookahead and Vector Field and the sign of their difference means show that their error metric are smaller than NLGL error metric.

Figure 37 presents the mean and confidence intervals when considering the effort metric $M_{path,effort}$ for each path-following algorithm. In this case, it is clear that NLGL requires more effort to execute the reference path, followed by Lookahead. But PLOS and Vector Field have overlapping confidence intervals, which requires a t-test analysis. The result of the difference between the effort metrics of PLOS and Vector Field

Figure 36 – Mean and confidence interval of the error metric difference among NLGL and the other path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.

Source: Research data.

is $(-0.01709453 \pm 0.0012661013)$, which shows that the effort metric for Vector Field is larger than for PLOS in average.

The comparison among all path-following according to the metric M_{path} is illustrated in Figure 38. The behaviour of the results is very similar to the one in the error metric due to its increased weight in the metric composition. Lookahead, PLOS and Vector Field have a statistically equal metric, while NLGL requires t-test analysis.

The results of the mean and confidence interval of the difference of the metric among NLGL and the other path-following algorithms are presented in Figure 39. Differently from the results obtained for the error metric, all confidence intervals do not contain zero, i.e., Lookahead, PLOS and Vector Field have distinct metrics from NLGL and, in average, their metrics are smaller.

Summarising, NLGL has the worse performance regarding both error and effort, which becomes clear in the final metric. PLOS has the smallest effort, followed by Vector Field and then Lookahead. But when evaluating the final metric, the three path-following algorithms have a similar efficiency, being potential candidates for AVALON's autopilot.

Figure 37 – Effort metric mean and confidence interval from all path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.

Source: Research data.

Figure 38 – Metric mean and confidence interval from all path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.

Source: Research data.

Figure 39 – Mean and confidence interval of the metric difference among NLGL and the other path-following algorithms for 30 replications of the horizontal flight with different intensities of wind disturbances.

Source: Research data.

5.4 Transitions

During the TVH and THV, the reference velocity and the angles of roll and yaw should be kept constant, but pitch should vary from approximately 0° to 90° . Since this trajectory is known at the beginning of the transition, it can be computed with a trapezoidal velocity profile, which results in a smooth trajectory.

A trapezoidal velocity profile imposes a constant acceleration at the start of the trajectory, a constant velocity during the middle and a constant acceleration at the end, resulting in a position history with a linear segment that connects two parabolic segments at the initial and final positions (SICILIANO; SCIAVICCO; VILLANI, 2009).

For a given acceleration $\ddot{\theta}_i$, deceleration $\ddot{\theta}_f$, constant velocity of $\dot{\theta}_c$, initial position θ_i and final position θ_f , the trajectory $\theta_{ref}(t)$ from the velocity trapezoidal profile can be computed with (5.25), where t_c is the time when the constant velocity is reached, defined as (5.26), and t_f is the time when the final position is achieved, restricted to (5.27) (SICILIANO; SCIAVICCO; VILLANI, 2009).

$$\theta_{ref}(t) = \begin{cases} \theta_i + \frac{\ddot{\theta}_i t^2}{2} & \text{for } 0 \leq t \leq t_c \\ \theta_i + \frac{\dot{\theta}_c^2}{2\ddot{\theta}_i} + \dot{\theta}_c t & \text{for } t_c \leq t \leq t_f - t_c \\ \theta_f + \frac{\dot{\theta}_f}{2} (t_f - t)^2 & \text{for } t_f - t \leq t \leq t_f \end{cases} \quad (5.25)$$

$$t_c = \frac{\dot{\theta}_c}{\ddot{\theta}_i} \quad (5.26)$$

$$t_f = \frac{\dot{\theta}_c}{2\ddot{\theta}_i} + \frac{\theta_f - \theta_i}{\dot{\theta}_c} - \frac{\dot{\theta}_c}{2\ddot{\theta}_f} \quad (5.27)$$

For AVALON, the trapezoidal velocity profile is generated with a constant acceleration/deceleration of $\pm 2 \text{ rad/s}^2$ and a middle velocity of $\pm 0.6 \text{ rad/s}$. Instead of using the derivative of the angle, the pitch velocity is represented in the body frame from the angular velocity vector $\boldsymbol{\omega}$ as Q . Accordingly, the references and results of pitch, pitch velocity and pitch acceleration are represented by Figures 40, 41 and 42 for TVH and by Figures 43, 44 and 45 for THV.

Figure 40 – Reference and resulted pitch angle from trapezoidal velocity profile during TVH with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

Figure 41 – Reference and resulted pitch velocity from trapezoidal velocity profile during TVH with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

Figure 42 – Reference and resulted pitch acceleration from trapezoidal velocity profile during TVH with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

Figure 43 – Reference and resulted pitch angle from trapezoidal velocity profile during THV with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

Figure 44 – Reference and resulted pitch velocity from trapezoidal velocity profile during THV with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

Figure 45 – Reference and resulted pitch acceleration from trapezoidal velocity profile during THV with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

The reference trajectory of the pitch angle has a smooth beginning and ending for both trajectories and the aircraft follows them correctly, however, the transitions end with a velocity greater than zero. This occurs due to the criteria to change the flight stage, which relies only on the values of the pitch angle. Nevertheless, the controllers of the next flight stage can handle the remaining velocity and acceleration, keeping the operation stable.

5.5 Landing

At takeoff, transitions and horizontal flight stages, the velocity magnitude is used as a reference velocity to the Control Unit and the values for each flight stage are predefined in the Navigation Unit. But during the landing stage, the direction of the velocity is more important than its intensity, so a different strategy is needed.

In the landing stage, AVALON has an initial velocity in the negative direction of Z_D , however, the weight acts as an opposite force and the aircraft is decelerated until it reaches a zero velocity and starts to descend. This descent must be executed with a low velocity when the aircraft is close to the ground, but it can be faster at a high altitude. Therefore, in order to guarantee an adequate landing velocity, a PID controller is used by the Navigation Unit to specify the reference velocity during the landing stage.

The PID controller receives the measured altitude from the Sensor Unit and uses zero as the reference, resulting in an error $0 - (-P_D) = P_D$. Its output is the reference velocity in the $-Z_D$ direction, i.e., $-V_{D,ref}$. The PID controller that gives the expected reference velocity is in fact just a Proportional controller with gain 0.8. The use of this controller results in a smooth deceleration during the landing stage, as can be seen in Figure 46.

Figure 46 – Reference and resulted velocity up from PID controller during landing with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and DI with PID structure in Control Unit.

Source: Research data.

The reference velocity tends to zero at the end of the landing and the Control Unit is able to follow it until the last two seconds of the operation. At these two seconds, the altitude achieves 0.1 m (the precision can be guaranteed by using a sonar to collect altitude data during the landing), the throttle is turned off and the velocity increases due to the gravity acceleration until the aircraft hits the ground.

5.6 Final remarks

All the tasks that Navigation Unit is responsible for were presented in this chapter, with a focus in the path-following. Lookahead, NLGL, PLOS and Vector Field algorithms were extended to the 3D scenario and the results of their performance regarding position errors and actuator efforts were compared. NLGL had the worse performance in replications with variation of the wind disturbances, while the other path-following algorithms had a similar efficiency.

The chosen path-following algorithm to compose the Navigation Unit of AVALON is Vector Field, because besides of having a good performance, it has proved to be more robust by prevailing in a situation where the other algorithms failed.

Moreover, a trapezoidal velocity profile was adopted to TVH and THV pitch trajectories and a Proportional controller was designed to decelerate the aircraft during the landing. Together, all these strategies from the Navigation Unit provide references of attitude and velocity to the Control Unit for a smoother operation. How the Control Unit is capable of tracking these references is answered in the next chapter.

CONTROL

6.1 Chapter overview

The Control Unit solves one of the most important challenges of AVALON operation, which is to keep a stable flight during all stages with different manoeuvres. This was accomplished with two structures for the attitude control, the Decoupled PID structure based on traditional methods and the DI with PID structure, based on dynamic inversion methods. Both structures use the same aircraft velocity controller as a PID controller that outputs the throttle command. All PID controllers presented in this chapter have the same functions of the PID controllers applied to the path-following algorithms, which is detailed in Section 5.2.

The Decoupled PID structure was constructed based on traditional methods that were already applied to the control of tailsitters. Such methods adopt the attitude error in PID controllers to obtain the control surfaces deflections (STONE *et al.*, 2008; MATSUMOTO *et al.*, 2010; BILODEAU; WONG, 2010). This implementation is a basis of comparison to the new structure based on dynamic inversion.

Dynamic inversion techniques use the knowledge from the system dynamics to define control laws that linearise the system using feedback structures (LEWIS; DAWSON; ABDALLAH, 2003; SICILIANO; SCIAVICCO; VILLANI, 2009). In this chapter, it is presented two of them, the computed-torque and the Incremental Nonlinear Dynamic Inversion (INDI). The computed-torque and feedback linearisation has already been applied to quadrotors control (LOTUFO *et al.*, 2015; BALASUBRAMANIAN; VASANTHARAJ, 2013), tri-rotors control (MOHAMED; LANZON, 2013) and fixed-wing control (CHEN; HE; QIAN, 2016; ZHANG *et al.*, 2013; HU *et al.*, 2012), but only preliminary results without convergence were applied to a tailsitter attitude controller (KOHNO; UCHIYAMA, 2014) and for the control of the flight velocity and path angle, which are more related

to the navigation of a tailsitter (JUNG; SHIM, 2012). INDI was also used for the control of fixed-wing (SIEBERLING; CHU; MULDER, 2010), a quadrotor (SMEUR; CHU; CROON, 2015) and a tiltrotor (FRANCESCO; MATTEI, 2015), but not for tailsitters, reinforcing the contribution of the DI with PID structure for AVALON.

The control techniques and structures Decoupled PID and DI with PID are presented in this chapter, along with their comparison from simulation results. The following section details the velocity controllers, which are common to both structures.

6.2 Velocity controller

Two different controllers are used as velocity controllers, one for the velocity magnitude during takeoff, transitions and horizontal stages and another for the velocity in the $-Z_D$ direction during the landing stage.

PID controllers with different gains for each flight stage control the velocity magnitude, but they all receive the error between the reference velocity magnitude from the Navigation Unit and the measured velocity magnitude and output the throttle command.

The PID controller for the landing stage uses the error between the reference velocity in the $-Z_D$ direction given from the Navigation Unit and the measured velocity in the $-Z_D$ direction and also outputs the throttle command.

These controllers guarantee the following of the reference velocities during all flight stages. However, the execution of the manoeuvres is also related to the following of the attitude, which are executed by the Decoupled PID structure and by the DI with PID structure, explained in the following sections.

6.3 Decoupled PID structure

The Decoupled PID structure is composed of PID controllers that have control surfaces deflections as outputs and the errors from the reference attitude given by the Navigation Unit and the measured attitude from Sensor Unit as inputs. Since different flight stages have different requirements, the control structure is divided into vertical flight and transition structure, and horizontal flight structure.

The vertical flight and transition control structure use the quaternion error $\mathbf{q}_e = [q_{e,1}, q_{e,2}, q_{e,3}, q_{e,4}]^T$, which is defined as (6.1), where $\otimes$ is the quaternion multiplication, $\mathbf{q}_{\text{ref}} = [q_{\text{ref},1}, q_{\text{ref},2}, q_{\text{ref},3}, q_{\text{ref},4}]^T$ is the reference attitude specified by the Navigation Unit and $\mathbf{q}^*$ is the conjugate of the measured quaternion (SOLA, 2017; ARGYLE *et al.*, 2014).

$$\mathbf{q}_e = \mathbf{q}_{\text{ref}} \otimes \mathbf{q}^* = \begin{bmatrix} q_{\text{ref},1} & -q_{\text{ref},2} & -q_{\text{ref},3} & -q_{\text{ref},4} \\ q_{\text{ref},2} & q_{\text{ref},1} & -q_{\text{ref},4} & q_{\text{ref},3} \\ q_{\text{ref},3} & q_{\text{ref},4} & q_{\text{ref},1} & -q_{\text{ref},2} \\ q_{\text{ref},4} & -q_{\text{ref},3} & q_{\text{ref},2} & q_{\text{ref},1} \end{bmatrix} \begin{bmatrix} q_1 \\ -q_2 \\ -q_3 \\ -q_4 \end{bmatrix} \quad (6.1)$$

The vector part of the quaternion error $[q_{e,2}, q_{e,3}, q_{e,4}]^T$ is directly related to the axes errors that direct the measurement quaternion to the reference quaternion due to the relation between the quaternion and a rotation vector representation defined by (6.2), where $\mathbf{u}_{\text{rot}}$ is the rotation axes and α_{rot} is the angle of rotation. However, when considering this relation, it is important to notice that when more than π radians are demanded, i.e. $q_{e,1} \leq 0$, the closest rotation is the conjugate of $\mathbf{q}_e$ (FRESK; NIKOLAKOPOULOS, 2013).

$$\mathbf{q} = \cos\left(\frac{\alpha_{\text{rot}}}{2}\right) + \mathbf{u}_{\text{rot}} \sin\left(\frac{\alpha_{\text{axis}}}{2}\right) \quad (6.2)$$

The angle of sideslip β should also be controlled by the rudders to improve the stability of the vertical manoeuvres. Consequently, the inputs of the PID controllers are the vector part of the quaternion error and the error of the sideslip, whose reference is also kept as zero, and their outputs are the control surfaces deflections commands δ_{ail}^c , δ_{elev}^c and δ_{rud}^c , as can be seen in Figure 47.

Figure 47 – Structure of vertical flight and transition control of the Decoupled PID.

Source: Elaborated by the author.

The quaternion error is a useful measure when the difference between the reference attitude and the measured attitude is small, however, when the reference attitude has a large change, the quaternion error is not representative. For example, when the aircraft

has to change its direction abruptly while performing a curve, the reference roll can also present a sudden change, invalidating the use of the quaternion error during the horizontal flight (MATSUMOTO *et al.*, 2010). Also, since during the horizontal flight the pitch is not close to 90° , there is no risk to occur the gimbal lock, enabling the use of Euler angles to compute the attitude error.

The coordinated turn is a curve manoeuvre with reduced sideslip, which can be achieved by using the change in the roll angle to obtain a change in the yaw without producing a high lateral velocity. This manoeuvre is performed with the horizontal flight control in Figure 48, which also has the elevator controlling the angle of attack α to maintain the lift during the flight.

Figure 48 – Horizontal flight control structure of the Decoupled PID.

Source: Elaborated by the author.

6.4 DI with PID structure

The DI with PID structure uses two techniques based on dynamic inversion, the computed torque and the Incremental Nonlinear Dynamic Inversion (INDI). The computed torque supplies reference torques to follow the reference attitude given by the Navigation Unit and the INDI calculates the actuators commands that tracks the reference torques. Both methods and the flight control structure applied to AVALON are presented in this section.

6.4.1 Computed torque

The computed torque control is an application of the feedback linearisation method to the nonlinear dynamics of robotic systems. It is also known as Nonlinear Dynamic

Inversion (NDI) control because it uses the modelled system dynamics to simplify the controller definition, which allows the trajectory tracking of a signal (LEWIS; DAWSON; ABDALLAH, 2003; SICILIANO; SCIAVICCO; VILLANI, 2009).

The computed torque formulation can be demonstrated by using (3.4) of AVALON's model and rewriting it as (6.3), where $\mathcal{C}(\omega)$ is defined by (6.4).

$$\mathbf{I}\dot{\omega} - \mathcal{C}(\omega) = \tau \quad (6.3)$$

$$\mathcal{C}(\omega) = \begin{bmatrix} QR(I_{zz} - I_{yy}) \\ PR(I_{xx} - I_{zz}) \\ PQ(I_{yy} - I_{xx}) \end{bmatrix} \quad (6.4)$$

An error of the angular velocity can be defined as (6.5), which is influenced by τ according to its first derivative in (6.6). By substituting (6.3) into (6.6), the error derivative results in (6.7). Consequently, with a control input function as (6.8), the error dynamics become a linear system, which can be controlled by feedback controllers, like PID.

$$\mathbf{e}_\omega = \omega_{ref} - \omega \quad (6.5)$$

$$\dot{\mathbf{e}}_\omega = \dot{\omega}_{ref} - \dot{\omega} \quad (6.6)$$

$$\dot{\mathbf{e}}_\omega = \dot{\omega}_{ref} - \mathbf{I}^{-1}(\mathcal{C}(\omega) + \tau) \quad (6.7)$$

$$\mathbf{u} = \dot{\omega}_{ref} - \mathbf{I}^{-1}(\mathcal{C}(\omega) + \tau) \quad (6.8)$$

The inverse of (6.8) is (6.9), which is called the computed-torque control law and is responsible to linearise AVALON's dynamics. This means that by choosing a control input $\mathbf{u}$ that stabilises $\mathbf{e}_\omega$, the nonlinear control input $\tau_{ref} = [L_{ref}, M_{ref}, N_{ref}]^T$ stabilises the entire nonlinear system. The control input $\mathbf{u}$ can be a Proportional controller with negative gain in this case.

$$\tau_{ref} = \mathbf{I}(\dot{\omega}_{ref} - \mathbf{u}) - \mathcal{C}(\omega) \quad (6.9)$$

Equations (3.22)-(3.43) show a linear relationship between the torques and the actuators deflections. However, these relationships are still coupled and subjected to uncertainties in the aerodynamic derivatives. In order to overcome these issues, another dynamic inversion technique is applied, as detailed in the next subsection.

6.4.2 Incremental Nonlinear Dynamic Inversion

The Incremental Nonlinear Dynamic Inversion (INDI) uses the incremental form of the dynamics to determine variations of the control input that stabilises the system. It is considered to be more robust than the classical NDI because it is less sensitive to model errors (SIEBERLING; CHU; MULDER, 2010; BACON; OSTROFF; JOSHI, 2001).

For AVALON, INDI formulation from the torques commands to actuators deflections can be obtained by using the moments derivatives in (6.10), which is a rewritten form of (3.41)-(3.43), where $\mathbf{M}_a$ is a matrix that generates the aerodynamic moments due to the airframe, defined by (6.11), and $\mathbf{M}_u$ is the matrix that generates the aerodynamic moments due to the control surfaces deflections, defined by (6.12).

$$\begin{bmatrix} C_L \\ C_M \\ C_N \end{bmatrix} = \mathbf{M}_a \begin{bmatrix} \alpha \\ \beta \\ P \\ Q \\ R \end{bmatrix} + \mathbf{M}_u \begin{bmatrix} \delta_{ail} \\ \delta_{elev} \\ \delta_{rud} \end{bmatrix} \quad (6.10)$$

$$\mathbf{M}_a = \begin{bmatrix} 0 & -C_{L,\beta} & -\frac{C_{L,P}B}{2\|\mathbf{v}_i\|} & 0 & \frac{C_{L,R}B}{2\|\mathbf{v}_{rel}\|} \\ C_{M,\alpha} & 0 & 0 & \frac{C_{M,Q}C_{Chord}}{2\|\mathbf{v}_{rel}\|} & 0 \\ 0 & -C_{N,\beta} & -\frac{C_{N,P}B}{2\|\mathbf{v}_i\|} & 0 & \frac{C_{N,R}B}{2\|\mathbf{v}_{rel}\|} \end{bmatrix} \quad (6.11)$$

$$\mathbf{M}_u = \begin{bmatrix} C_{L,ail} & 0 & C_{L,rud} \\ 0 & C_{M,elev} & 0 \\ C_{N,ail} & 0 & C_{N,rud} \end{bmatrix} \quad (6.12)$$

The first terms of the Taylor series expansion of (6.10) around known moment derivatives $[C_{L,0}, C_{M,0}, C_{N,0}]^T$ and control surfaces deflections $\boldsymbol{\delta}_{cs,0} = [\delta_{ail,0}, \delta_{elev,0}, \delta_{rud,0}]^T$ are represented by (6.13), where $\boldsymbol{\tau}_0$ is the known moments related to $[C_{L,0}, C_{M,0}, C_{N,0}]^T$ and $\boldsymbol{\delta}_{cs} = [\delta_{ail}, \delta_{elev}, \delta_{rud}]^T$ are the control surfaces deflections.

$$\begin{aligned}
\begin{bmatrix} C_L \\ C_M \\ C_N \end{bmatrix} &\approx \begin{bmatrix} C_{L,0} \\ C_{M,0} \\ C_{N,0} \end{bmatrix} + (\tau - \tau_0) \frac{\delta}{\delta \tau} \begin{bmatrix} \mathbf{M}_a \begin{bmatrix} \alpha \\ \beta \\ P \\ Q \\ R \end{bmatrix} + \mathbf{M}_u \begin{bmatrix} \delta_{ail} \\ \delta_{elev} \\ \delta_{rud} \end{bmatrix} \end{bmatrix}_{\tau_0, \delta_{cs,0}} \\
&+ (\delta_{cs} - \delta_{cs,0}) \frac{\delta}{\delta \delta_{cs}} \begin{bmatrix} \mathbf{M}_a \begin{bmatrix} \alpha \\ \beta \\ P \\ Q \\ R \end{bmatrix} + \mathbf{M}_u \begin{bmatrix} \delta_{ail} \\ \delta_{elev} \\ \delta_{rud} \end{bmatrix} \end{bmatrix}_{\tau_0, \delta_{cs,0}} \quad (6.13)
\end{aligned}$$

For a very small time increment, it is possible to assume that the second term is negligible, resulting in (6.14). This assumption is based on the fact that the changes in the aerodynamic moments are too small when compared to control moments (FRANCESCO; MATTEI, 2015).

$$\begin{bmatrix} C_L \\ C_M \\ C_N \end{bmatrix} \approx \begin{bmatrix} C_{L,0} \\ C_{M,0} \\ C_{N,0} \end{bmatrix} + \mathbf{M}_u (\delta_{cs} - \delta_{cs,0}) \quad (6.14)$$

Therefore, the control law (6.15) represents the incremental change in the surface deflections that produces the change in the moments derivatives, which result in the change of the moments. Also, $\mathbf{M}_u^{-1}$ can be solved analytically as (6.16), which reduces the computational cost of the control law.

$$\delta_{cs} = \mathbf{M}_u^{-1} \left(\begin{bmatrix} C_{L,ref} \\ C_{M,ref} \\ C_{N,ref} \end{bmatrix} - \begin{bmatrix} C_{L,0} \\ C_{M,0} \\ C_{N,0} \end{bmatrix} \right) + \delta_{cs,0} \quad (6.15)$$

$$\mathbf{M}_u^{-1} = \begin{bmatrix} 0 & \frac{1}{C_{M,elev}} & 0 \\ \frac{C_{N,rud}}{C_{L,ail}C_{N,rud} - C_{L,rud}C_{N,ail}} & 0 & -\frac{C_{L,rud}}{C_{L,ail}C_{N,rud} - C_{L,rud}C_{N,ail}} \\ -\frac{C_{N,ail}}{C_{L,ail}C_{N,rud} - C_{L,rud}C_{N,ail}} & 0 & \frac{C_{L,ail}}{C_{L,ail}C_{N,rud} - C_{L,rud}C_{N,ail}} \end{bmatrix} \quad (6.16)$$

In the AVALON attitude control, the reference torques from the computed-torque can be normalised with (6.17)-(6.19), resulting in the reference derivatives to the INDI

control law. Consequently, the combined use of computed-torque and INDI guarantee the tracking of the angular velocity by commanding the control surfaces deflections.

$$C_{L,ref} = \frac{2L_{ref}}{\|\mathbf{v}_i\|^2 \rho SB} \quad (6.17)$$

$$C_{M,ref} = \frac{2M_{ref}}{\|\mathbf{v}_i\|^2 \rho SC_{Chord}} \quad (6.18)$$

$$C_{N,ref} = \frac{2N_{ref}}{\|\mathbf{v}_i\|^2 \rho SB} \quad (6.19)$$

6.4.3 Attitude control

The computed-torque and INDI provide control laws that allow the tracking of the reference angular velocity $\boldsymbol{\omega}_{ref}$, however, the Navigation Unit specifies the reference attitude for the correct execution of the flight stages. Thus, it is necessary to establish control laws that can track the reference attitude as well as the reference angular velocity.

The reference attitude tracking can be solved with the time-scale separation technique, which divides a system into two subsystems, one with slow dynamics and another one with fast dynamics (NAIDU; CALISE, 2001). The time-scale separation can be applied to the rotational dynamics of an aircraft, whose angular velocity is considered to be ruled by fast dynamics, while the attitude has slow dynamics. Consequently, the control loops can also be separated, with an outer loop controlling the attitude and an inner loop controlling the angular velocity (SIMPLICIO, 2011; SADRAEY; COLGREN, 2005; SNELL *et al.*, 1992).

Figure 49 shows the resulting attitude control of the DI with PID structure with an inner loop that guarantees the angular velocity tracking and an outer loop responsible for the attitude tracking. The computed-torque and INDI is applied in the inner loop and is always the same for all flight stages, but the feedback controllers used in both loops are different for each flight stage due to their peculiarities.

The feedback controllers for the takeoff and landing are represented by Figure 50. This is the simplest structure with both attitude and computed-torque composed of PIDs that individually define the angular velocity reference as $\boldsymbol{\omega}_{u,ref} = [P_{u,ref}, Q_{u,ref}, R_{u,ref}]^T$ and angular acceleration reference as $\mathbf{u} = [\dot{P}_{u,ref}, \dot{Q}_{u,ref}, \dot{R}_{u,ref}]^T$.

During the transitions, the pitch trajectory is defined by the Navigation Unit with references of the angle, velocity and acceleration $(\theta_{ref}, \dot{\theta}_{ref}, \ddot{\theta}_{ref})$. For this reason, the feedback controllers have a different structure, represented by Figure 51, with a combined use of the attitude controller and the angular velocity controller for the pitch angle and velocity. The pitch reference is transformed to its respective quaternion together with the roll and yaw references before entering the attitude controller. Also, by summing the output of

Figure 49 – DI with PID structure for attitude control.

Source: Elaborated by the author.

Figure 50 – Structure of the takeoff and landing feedback controllers for DI with PID.

Source: Elaborated by the author.

both controllers, the angle and velocity are corrected by the feedback controllers through $\dot{Q}_{u,ref}$, which is added to the acceleration reference that defines the trajectory in the computed-torque. The other parts of the attitude controllers and computed-torque controllers are similar to the structure of the takeoff and landing controllers, which calculate the angular velocities and accelerations separately.

Figure 51 – Structure of the transitions feedback controllers for DI with PID.

Source: Elaborated by the author.

The horizontal controllers use the Euler angles as in the Decoupled PID, but their outputs are the velocities reference to the computed-torque controllers to keep the consis-

tency with the other structures. Also, the controllers that give the $Q_{ref,u}$ are composed by two PIDs, one for pitch and another for the angle of attack, which must be kept as α_{ref} to guarantee a certain lift during the path-following. All these characteristics, which guarantee the correct execution of the coordinated turn, can be seen in Figure 52, with the output of the feedback controllers being $\mathbf{u} = [\dot{P}_{u,ref}, \dot{Q}_{u,ref}, \dot{R}_{u,ref}]^T$.

Figure 52 – Structure of the horizontal feedback controllers for DI with PID.

Source: Elaborated by the author.

The results from simulation of the DI with PID structure, the Decoupled PID structure and the velocity controllers are reported in the next section, along with a comparison between the controllers for each flight stage.

6.5 Simulation and comparison results

The structures that compose the Control Unit are validated in a similar process to the Sensor Unit and Navigation Unit, using simulations based on the kinematic and dynamic models of AVALON defined in Section 3.3. The Control Unit is more susceptible to disturbances because of the fast dynamics of the angular velocities, nonlinearity of the equations of motion and unstable states during the manoeuvres. Consequently, to deeply cover the structures performance, the simulations of all flight stages were executed with variations of sensor noise and wind intensity, as described, respectively, in Section 4.5 and Subsection 5.3.3.

Also analogous to the path-following algorithms comparison, the adjustment of the controller gains of both structures was performed with GA and SQP optimisations to

improve the fairness of the comparison. In the case of the Control Unit, each flight mode controller was optimised separately.

Each flight mode has objective functions as metrics defined by (6.20) with a part related to the errors specified by (6.21) and with a part related to the efforts specified by (6.22). The subscript fs represents the flight stage and each flight stage has a set of expected values, shown in Table 7, that acts as a normalisation of each error/effort element.

$$M_{control,fs} = 0.8M_{error,fs} + 0.2M_{effort,fs} \quad (6.20)$$

$$M_{error,fs} = \frac{E_{\|v_i\|,fs}}{E_{exp,\|v_i\|,fs}} + \frac{E_{\phi,fs}}{E_{exp,\phi,fs}} + \frac{E_{\theta,fs}}{E_{exp,\theta,fs}} + \frac{E_{\psi,fs}}{E_{exp,\psi,fs}} + \frac{E_{\beta,fs}}{E_{exp,\beta,fs}} \quad (6.21)$$

$$M_{effort,fs} = \frac{\delta_{t,thr,fs}}{\delta_{exp,t,thr,fs}} + \frac{\delta_{t,ail,fs}}{\delta_{exp,t,ail,fs}} + \frac{\delta_{t,elev,fs}}{\delta_{exp,t,elev,fs}} + \frac{\delta_{t,rud,fs}}{\delta_{exp,t,rud,fs}} \quad (6.22)$$

Table 7 – Expected errors and efforts for each flight stage used in the metrics $M_{error,fs}$ defined by (6.21) and $M_{effort,fs}$ defined by (6.22).

Expected Error/Effort	Flight stages				
	Takeoff	TVH	Horizontal flight	THV	Landing
$E_{exp,\ v_i\ ,fs}$	1	1	1	1	1
$E_{exp,\phi,fs}$	1	1	5	1	1
$E_{exp,\theta,fs}$	1	1	1	1	1
$E_{exp,\psi,fs}$	1	1	30	1	1
$E_{exp,\beta,fs}$	0.1	0.1	0.1	1	0.1
$\delta_{exp,t,thr,fs}$	5	1	0.1	5	1
$\delta_{exp,t,ail,fs}$	0.1	1	1	1	0.1
$\delta_{exp,t,elev,fs}$	2	5	5	5	2
$\delta_{exp,t,rud,fs}$	0.1	1	2	1	0.1

Source: Elaborated by the author.

The error metric $M_{error,fs}$ is composed by the mean of the absolute errors of the velocity magnitude ($E_{\|v_i\|,fs}$), roll ($E_{\phi,fs}$), pitch ($E_{\theta,fs}$), yaw ($E_{\psi,fs}$) and sideslip angle ($E_{\beta,fs}$), whose reference is given by the Navigation Unit and which are directly controlled by the Control Unit structures. The effort metric $M_{effort,fs}$ is composed by the mean of the absolute actuator inputs of throttle ($\delta_{t,thr,fs}$) and ailerons ($\delta_{t,ail,fs}$), elevators ($\delta_{t,elev,fs}$) and rudders ($\delta_{t,rud,fs}$) deflections. The higher weight of the error metric was established to

emphasise the tracking function of the controllers, while the effort is part of the metric to provide a feasible behaviour.

Therefore, for each flight stage, the controller gains were optimised by MATLAB routines of GA, with a population size of 20 elements and a maximum of 20 generations, followed by SQP with a tolerance function of 10^{-5} . The results from the optimisation procedure for Decoupled PID controller gains and for DI with PID controller gains are respectively related in Table 8. The controllers that gave the best results for both structures were Proportional controllers, which justifies the absence of the other parameters.

All replications used to validate the control structures covered the entire operation of AVALON, going through the five flight stages. It started with the aircraft at the position $\mathbf{p}_{i,initial} = [0, 0, 0]^T$ with the takeoff, changed to TVH at the altitude of 20 m, followed the path defined by a list of 6 waypoints during the horizontal flight, changed to THV after the last waypoint was reached and finished with the landing. The reference path was defined by the waypoints $[0, 0, -20]$, $[300, 0, -20]$, $[300, 300, -20]$, $[0, 300, -20]$, $[0, 0, -20]$, $[300, 0, -20]$ in the inertial frame.

The first replication was executed with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit, the same configuration of sensor noise and Vector Field as path-following algorithm in the Navigation Unit. The resulting paths for Decoupled PID and DI with PID structures for the first replication are presented in Figure 53.

Both structures correctly execute the entire operation of AVALON under the conditions of the first replication. At the beginning of the horizontal flight, there is an increase in the altitude for the Decoupled PID, which is justified by the adjusting of the angle of attack before correcting the altitude with the pitch reference. However, the controller can still correct the altitude after some time, which indicates that the pitch reference is correctly followed. It can also be noted that DI with PID path has larger overshoots during the curves and during the landing.

Figures 54, 55, and 56 show the roll, pitch and yaw angles for each control structure for the first replication. The resulting angles follow the expected references, with a larger variation of roll during the horizontal flight to perform the coordinate turns and with a larger variation of pitch during the transitions. Yaw varies during the horizontal flight as a consequence of the roll variation, since yaw is not directly controlled during this stage.

By comparing the results of each structure, DI with PID roll and yaw has smaller overshoots than the ones from Decoupled PID, especially during THV and landing. This behaviour can be explained by the equations of motion nonlinearity that are handled by DI with PID, but not by Decoupled PID. The pitch angle also has a bigger overshoot for Decoupled PID at the beginning of the horizontal flight due to the adjustment of the angle of attack.

Table 8 – Controller gains of Decoupled PID and DI with PID structures resulted from GA and SQP optimisations with the objective function (6.20).

Flight stages	Controller gains	
	Decoupled PID	DI with PID
Takeoff	$K_{q_{e,2},P} = 0, K_{q_{e,3},P} = -2.01,$ $K_{q_{e,4},P} = 0, K_{\beta,P} = 0,$ $K_{\ v_i\ ,P} = 3$	$K_{q_{e,2},P} = 0, K_{q_{e,3},P} = 17.08,$ $K_{q_{e,4},P} = 70.76, K_{P,P} = 22.33,$ $K_{Q,P} = 34.73, K_{R,P} = 89.11$ $K_{\ v_i\ ,P} = 3$
TVH	$K_{q_{e,2},P} = 0.24, K_{q_{e,3},P} = -59.34,$ $K_{q_{e,4},P} = -83.36, K_{\beta,P} = 0,$ $K_{\ v_i\ ,P} = 10$	$K_{q_{e,2},P} = 0.08, K_{q_{e,3},P} = 135.86,$ $K_{q_{e,4},P} = 61.78, K_{P,P} = 50.29,$ $K_{Q,P} = 82.47, K_{R,P} = 122.28,$ $K_{\ v_i\ ,P} = 10$
Horizontal flight	$K_{\phi,P} = 1.57, K_{\theta,P} = -5.03,$ $K_{\alpha,P} = -13.55, K_{\beta,P} = 3.38,$ $K_{\ v_i\ ,P} = 1$	$K_{\phi,P} = 8.64, K_{\theta,P} = 89.86,$ $K_{\alpha,P} = 26.56, K_{\beta,P} = -55.15,$ $K_{P,P} = 31.25, K_{Q,P} = 45.04,$ $K_{R,P} = 132.19, K_{\ v_i\ ,P} = 1$
THV	$K_{q_{e,2},P} = 7.74, K_{q_{e,3},P} = -11.42,$ $K_{q_{e,4},P} = -7.12, K_{\beta,P} = 6.27,$ $K_{\ v_i\ ,P} = 10$	$K_{q_{e,2},P} = 22.93, K_{q_{e,3},P} = 463.06,$ $K_{q_{e,4},P} = 109.71, K_{P,P} = 142.58,$ $K_{Q,P} = 292.82, K_{R,P} = 500.54,$ $K_{\ v_i\ ,P} = 10$
Landing	$K_{q_{e,2},P} = 0, K_{q_{e,3},P} = -11.90,$ $K_{q_{e,4},P} = -17.21, K_{\beta,P} = 3.19,$ $K_{\ v_i\ ,P} = 2$	$K_{q_{e,2},P} = 1.25, K_{q_{e,3},P} = 15.69,$ $K_{q_{e,4},P} = 50.64, K_{P,P} = 19.17,$ $K_{Q,P} = 9.32, K_{R,P} = 300.00,$ $K_{\ v_i\ ,P} = 2$

Source: Research data.

The angles of attack and sideslip, which are controlled during the horizontal flight stage for both structures, are also taken into account when evaluating the Control Unit structures. Figures 57 and 58 present, respectively, their results for the first replication.

Both structures are able to follow the 2.5° angle of attack reference during the horizontal stage satisfactorily, with minor oscillations during its beginning caused by the

Figure 53 – Positions in the inertial frame resulted from Decoupled PID and DI with PID structures for the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The path is defined by 6 waypoints (red points), whose sequence is represented by the red numbers.

Source: Research data.

increased change in pitch to correct the altitude. During the other flight stages, the angle of attack increases when the aircraft reaches low velocities, as already expected. At very low velocities, at the end of the landing stage, the angle of attack is ignored, justifying its zero values.

The sideslip angle is also correctly controlled by both structures, keeping it at low values during all flight stages. However, the result from Decoupled PID has larger oscillations and overshoots than the one from DI with PID, which can be explained by its ability to consider the dynamics of the model and compensate for them.

Since both attitude control structures use the same controllers for the velocity, their velocity magnitude results are very similar, as demonstrated in Figure 59. The small variations, like the larger overshoot by Decoupled PID, occurs because of the attitude differences. Therefore, the velocity controller successfully follows the reference given by the Navigation Unit.

All the results presented for the first replication show the correct performance of both structures in following the reference attitude and velocity. But other characteristics

Figure 54 – Roll angle for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 55 – Pitch angle for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 56 – Yaw angle for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 57 – Angle of attack (alpha in degrees) for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 58 – Sideslip angle for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 59 – Velocity magnitude for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

that must be reported are related to the effort, i.e., the actuators inputs that commanded the flight. Figures 60, 61, 62 and 63 present the throttle command and the control surfaces deflections of the ailerons, elevators and rudders, respectively, for both structures in the first replication.

Figure 60 – Throttle command for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

First, it must be stated that the time duration of the entire operation is considerably larger than the delay of the actuators commands. For this reason, the changes in the commands seem to be instantaneous, however, they are not. Therefore, the results of the actuators inputs are physically possible.

In the horizontal flight, it is clear that the DI with PID effort is smaller than the Decoupled PID, especially by observing ailerons and rudders deflections. However, there is an increase in their oscillations during the landing for the DI with PID. These oscillations occur because the error of the sideslip angle is larger when the attitude is closer to the singularity and the velocity magnitude is smaller, which reflects in model uncertainty. This behaviour during the landing is a disadvantage of the DI with PID, which can only be overcome by better estimations of the aircraft state.

The comparison of Decoupled PID and DI with PID structures considered replications for different configurations of sensor noise and wind disturbances, which are analysed separately. For the sensor noise variation, the DI with PID structure was able to execute the entire operation in most of the replications, but Decoupled PID was not. Figures

Figure 61 – Ailerons deflection for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 62 – Elevators deflection for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

Figure 63 – Rudders deflection command for Decoupled PID and DI with PID structures resulted from the first replication with a fixed wind of 0.4521 m/s in the north direction, UKF as Sensor Unit and Vector Field as path-following algorithm in Navigation Unit. The filled circles represent a change in the flight stage.

Source: Research data.

64 and 65 show the number of successful and failed replications in each flight stage for Decoupled PID and DI with PID respectively.

Figure 64 – Number of successful and failed replications in each flight stage for Decoupled PID from the simulation results with different configuration of sensor noise.

Source: Research data.

Figure 65 – Number of successful and failed replications in each flight stage for DI with PID from the simulation results with different configuration of sensor noise.

Source: Research data.

According to these numbers, the takeoff and TVH were the most easily executed flight stages by both control structures, without any failures, while landing seems to be the hardest one with the larger number of failures. Moreover, DI with PID failed only in 7 replications and only during the landing stage, while Decoupled PID succeeded to execute the entire operation only in 9 replications. These results show the superiority of DI with PID when dealing with error estimates of the attitude, but the comparison regarding the metrics defined earlier takes into account only the succeeded flight stages.

Figure 66 shows the error metric mean and confidence interval with 95% confidence level for different configurations of sensor noise. For takeoff, TVH and THV, the error metrics are very similar for both control structures and these flight stages also present smaller errors. The largest differences are in the horizontal flight and landing, where DI with PID has a considerably smaller error metric than Decoupled PID.

However, when considering the effort metric mean and confidence interval, the same behaviour was not observed, as seen in Figure 67. Decoupled PID has a smaller effort metric for all flight stages with the exception of the horizontal flight. However, it should be taken into account that horizontal flight is the longest flight stage and that the difference between Decoupled PID effort metric and DI with PID effort metric is the largest among all flight stages.

Figure 68 provides the mean and confidence interval of the final metric, defined by (6.20), for the replications with different configurations of sensor noise. It can be noticed

Figure 66 – Error metric mean and confidence interval for Decoupled PID and DI with PID from replications with different configurations of sensor noise.

Source: Research data.

Figure 67 – Effort metric mean and confidence interval for Decoupled PID and DI with PID from replications with different configurations of sensor noise.

Source: Research data.

that the error metric has a strong influence in the final metric, since their behaviours are very similar. Furthermore, it is possible to infer that the resulting efforts were closer to their expected values than the errors, which emphasises the error metric importance. Overall, DI with PID has a better performance than Decoupled PID, especially when considering the horizontal flight and landing.

Figure 68 – Metric mean and confidence interval for Decoupled PID and DI with PID from replications with different configurations of sensor noise.

Source: Research data.

Figures 69 and 70 provide the numbers of successful and failed replications for different intensities of wind disturbance in each flight stage for Decoupled PID and DI with PID respectively. Again, Decoupled PID has more failures than DI with PID in general, however, it has succeeded during takeoff and TVH in one replication that DI with PID had not. DI with PID had only one failed replication for all flight stages. Another similar behaviour to both sets of replications is the larger number of failures for THV and landing in the Decoupled PID, followed by the failures in horizontal flight.

Only the successful replications in each flight stage for different intensities of wind disturbance were considered in the mean and confidence interval of the error metric, which is represented in Figure 71. Once again DI with PID had significantly smaller errors than Decoupled PID for horizontal flight and landing, but it is also better in the takeoff. During the transitions, both control structures provided comparable performance.

Figure 72 shows the mean and confidence interval for the effort metric under these same conditions. Decoupled PID has a smaller effort metric than DI with PID in most

Figure 69 – Number of successful and failed replications in each flight stage for Decoupled PID from the simulation results with different intensities of wind disturbance.

Source: Research data.

Figure 70 – Number of successful and failed replications in each flight stage for DI with PID from the simulation results with different intensities of wind disturbance.

Source: Research data.

flight stages, with the exception of the horizontal flight, which is the same behaviour found in the sensor noise replications.

Finally, the final metric mean and confidence interval for the replications with

Figure 71 – Error metric mean and confidence interval for Decoupled PID and DI with PID from replications with different intensities of wind disturbance.

Source: Research data.

Figure 72 – Effort metric mean and confidence interval for Decoupled PID and DI with PID from replications with different intensities of wind disturbance.

Source: Research data.

different wind disturbances is presented in Figure 73. The metric results also show a strong influence from the error metric, with DI with PID prevailing over Decoupled PID.

Figure 73 – Metric mean and confidence interval for Decoupled PID and DI with PID from replications with different intensities of wind disturbance.

Source: Research data.

Summarising, DI with PID has shown a better accuracy and robustness when compared to Decoupled PID by considering different sensor noise and wind disturbances. However, it is necessary to acknowledge that Decoupled PID requires less effort in its successful operations than DI with PID in most of the flight stages.

6.6 Final remarks

This chapter focused in the implementation and comparison of two control structures to stabilise the aircraft for all its flight manoeuvres. Both Decoupled PID and DI with PID structures were capable of executing all flight stages with sensor noise and wind disturbances, however, under several replications they presented different performances.

The replications comparison demonstrated that DI with PID is more robust and is able to follow the reference attitude with smaller errors than Decoupled PID. For this reason, DI with PID was chosen to compose the Control Unit of AVALON.

However, it is important to state that DI with PID also has its own disadvantage. As shown in the landing stage for the sensor noise replications, it is sensitive to the errors

of the sideslip angle, which results in model uncertainties. Therefore, the estimation error of this angle must be as small as possible to guarantee the DI with PID correctness.

All techniques presented in Chapters 4, 5 and 6 also belongs to the autopilot software, whose details are presented in the next chapter.

AVALON PROTOTYPE

7.1 Chapter overview

This chapter offers more details about the AVALON prototype by describing its physical structure and the autopilot hardware and software. The prototype follows the project design presented in Figure 4. However, modifications were done during its development to improve the mechanical stability.

7.2 Physical structure

The AVALON prototype airframe and wings were made of Extruded polystyrene (XPS) foam plates 5 mm thick, following the dimensions established in Figure 4. The XPS is a low density material that is already used in the construction of radio-controlled model aircraft due to its stiffness, easiness to be cut and glued and reduced cost. Also, no changes in shape were required for making the flat plate wing aerofoil. After the first flight experiment, it was necessary to add carbon fibre sticks in the main wings to avoid vibrations due to its large dimensions.

Another change from the preliminary design was the removal of the structure around the tail to reduce the prototype weight. This structure was replaced by small extensions in the tail to provide support during the vertical takeoff and landing.

The control surfaces were also built from the 5mm thick XPS plates, with a length of 40 cm and width of 11 cm for each aileron, elevator and rudder. The initial design had only one rudder, but flight experiments elucidated the need for another one that was added to the low wing of the vertical stabiliser at the tail. Therefore, the final configuration of surface controls are 2 ailerons at the main wings, 2 elevators at the horizontal stabilisers and 2 rudders at the vertical stabilisers.

The micro servomotors Hobbico CS-12MG (SERVODATABASE.COM, 2018) were linked to each control surface with T-style control horns, nylon links and metal arms, resulting in deflection angle ranges of $[-25^\circ, 25^\circ]$ for the ailerons, $[-40^\circ, 40^\circ]$ for the elevators and $[-40^\circ, 40^\circ]$ for the rudders.

The electric motor Align BL700MX (ALIGN CORPORATION LIMITE, 2018) is directly controlled by the Electric Speed Controller (ESC) EFLA 1080B (E-FLITE, 2013), which is powered by an exclusive battery LiPo 4S with 5000 mAh. Moreover, an APC Thin Electric Propeller 19×10 is fixed to the motor with a metal 6 mm spinner.

Another battery LiFe with 3000 mAh is used to power the servomotors, the radio receiver and the autopilot. The radio receiver is essential due to the fact that a human pilot must have the ability to take control of the aircraft at any moment, according to Brazil's legislation. Therefore, the transmitter DX8 Spektrum with the receiver AR8000 (SPEKTRUM, 2018) is also employed in the AVALON operation.

The resulting prototype is represented in Figure 74. The larger battery that powers the motor is positioned close to the main wings to correct the CG and the red box in the left side contains the autopilot, detailed in the next section.

Figure 74 – AVALON prototype.

Source: Elaborated by the author.

7.3 Autopilot hardware and software

The autopilot hardware is composed by one microprocessor board and an auxiliary microcontroller board. The processor board is an ODROID-XU4 with Samsung Exynos5422 Cortex-A15 and Cortex-A7 Octacore CPUs, 2 GB of DDR3 as memory RAM and Ubuntu Minimal as OS, suitable for running the software from Sensor Unit, Navigation Unit and Control Unit ([HARDKERNEL, 2015](#)). The ODROID-XU4 communicates with the microcontroller board, which is an Arduino Leonardo, through a USB (Universal Serial Bus) serial communication ([ARDUINO, 2018](#)). The microcontroller board is used to read the information from the PWM (Pulse-Width Modulation) channels of the radio receiver for monitoring reasons and to send PWM signals to the servomotors.

Figure 75 shows the interfaces and the communication protocols among the two processing boards, sensors and actuators.

Figure 75 – Interfaces and communication protocols among the main processing board Odroid XU4, the auxiliary board Arduino Leonardo, the radio receiver AR8000, the ESC EFLA 1080B, the servomotors Hobbico CS-12MG, the IMU ADIS16407, the GPS LS20031, the sonar MB1242 and the airspeed sensor MS4525DO.

Source: Elaborated by the author.

The IMU is the ADIS16407, which contains triaxial accelerometers, gyroscopes and magnetometers and a barometer ([ANALOG DEVICES, 2011](#)). The ODROID-XU4

accesses the information from each sensor through SPI (Serial Peripheral Interface) communication in mode 3 by writing the address of the register related to each sensor data with a 300 Hz update rate.

The GPS module is the LS20031 that provides information about the position in the geodetic frame, velocity magnitude, heading angle, time and number of satellites available (LOCOSYS, 2009). Its communication is a UART (Universal Asynchronous Receiver-Transmitter) with TTL (Transistor-Transistor Logic) level, with continuous messages sent at a 1 Hz frequency as ASCII (American Standard Code for Information Interchange) characters with the NMEA (National Marine Electronics Association) protocol.

The sonar MB1242 is attached to the aircraft tail to measure a more precise distance from the ground during the takeoff and landing. Its data is accessed by I2C (Inter-Integrated Circuit) communication with an update rate of 18 Hz (MAXBOTIX INC., 2012).

The airspeed sensor MS4525DO is connected to a Pitot tube, which can be used with the static pressure to provide airspeed information (MEASUREMENT SPECIALTIES, 2017). It can use I2C or SPI as communication protocols, but I2C was chosen to exchange data with the processing board because SPI pins were already used by the communication with the IMU.

Both the IMU ADIS16407 and ODROID-XU4 require a 5 V power supply. The IMU is more sensitive to voltage oscillations and therefore has a more strict interval requirement, within [4.75, 5.25] V. For this reason, it was necessary to adopt a DC/DC switching regulator to guarantee a stable power source, which is used to power all sensors, except the GPS module. The GPS module LS20031 only accepts 3.3 V, thus, another regulator is needed, but a simpler linear regulator LM338 was chosen for being lighter and cheaper. It is attached to the ODROID-XU4 Shifter Shield, which provides input and output pins and connects all sensors. The microcontroller board is powered by the ODROID-XU4 USB interface.

The autopilot hardware, the IMU and the GPS are stored in a plastic container built especially to protect these elements, as seen in Figure 76. The container was modelled in SolidWorks in three different parts and produced in a 3D printer. The first part is a basis attached to the aircraft, the second part is a support for the hardware and the third part is a lid with gaps that allow the air through the box to prevent electronics overheating.

The communication protocols among the ODROID-XU4 and all sensors were implemented with C code by producing libraries for each sensor. The C code for Sensor Unit, Navigation Unit and Control Unit was generated automatically from MATLAB as functions. Therefore, the final autopilot code is produced by calling each function in the

Figure 76 – Autopilot container.

Source: Elaborated by the author.

right order and using the sensors libraries to capture sensor data, which is passed to the Sensor Unit function as an argument.

The resulting autopilot software was completely accessible because it was entirely developed by the author, which provides more flexibility and control over all components.

7.4 Experimental flights

The prototype construction allowed the execution of experimental flights by a human pilot to improve the physical structure of the aircraft. It also confirmed the physical feasibility of its operation as a tailsitter, despite of the failure at landing.

Figure 77 presents a sequence of images from two flight experiments that can also be seen in videos published at http://bit.ly/AVALON_Flights. In both experiments, the pilot was able to take off vertically, change from the vertical position to the horizontal, fly horizontally and change the position back to the vertical. However, it was not possible to land due to lateral wind gusts that produced an unstable behaviour.

7.5 Final remarks

The AVALON prototype and the autopilot components were described to elucidate the overall physical system of a tailsitter UAV. The experimental flights also demonstrate the possibility to control this type of aircraft by a human pilot. Consequently, it is expected that the autopilot will control AVALON through all flight stages. Moreover, the decisions taken in respect to the autopilot hardware and software resulted in a flexible and low cost solution. Finally, the conclusions and future works are presented in the next chapter.

Figure 77 – Sequence of images from two flight experiments with a human pilot controlling AVALON prototype.

Source: Elaborated by the author.

CONCLUSION

The definition of a tailsitter UAV and its autopilot requires the overcoming of challenges related to sensing the environment, taking decisions with the available information and actuating in feasible ways. The correct execution of each step on this task can bring consequences to the others, emphasising the importance of all its functions. Therefore, to accomplish such demand, AVALON's autopilot was created with different units, but still responsible for the entire flight autonomy.

The AVALON's design followed simplicity as main characteristic and resulted in a small electric fixed-wing tailsitter not so complex to make and maintain. Its autopilot has a Sensor Unit responsible for providing estimation of position, velocity and attitude, a Navigation Unit responsible for managing the flight stages and executing path-following tasks, and a Control Unit responsible for stabilising the manoeuvres in each flight stage.

Both techniques of Sensor Unit, EKF and UKF, provided estimations of position, velocity and attitude with bounded errors under reasonable limits. However, UKF errors of attitude are smaller than EKF, resulting in a better performance.

In the path-following comparison, although Lookahead, PLOS and Vector Field presented similar performance, Vector Field was able to complete the path in a replication that the others were not. Consequently, it was chosen as the default path-following algorithm of the Navigation Unit.

The use of the dynamic inversion control methods as DI with PID structure produced a better accuracy than the classical PID controllers usually applied for tailsitters. This new control structure, composed by computed torque and INDI, is considered as a novelty because it was not evaluated for tailsitters before and can also be adapted to other UAVs.

As a result, the final autopilot of AVALON is comprised of UKF as Sensor Unit, Vector Field as path-following algorithm for Navigation Unit and DI with PID structure

for attitude control in Control Unit.

Therefore, from all evaluations of the autopilot units, it is possible to conclude that AVALON is a tailsitter with only one propeller and without auxiliary structures that can fly autonomously through the vertical takeoff and landing, the transitions between vertical and horizontal positions, and the horizontal flight with its final autopilot, validating the initial hypothesis. Furthermore, it is possible to state that AVALON goes ahead in the state of the art since it is the first time that a tailsitter UAV uses and presents these unique characteristics.

8.1 Contributions

The thesis' most important contribution is the definition of a new tailsitter UAV, called AVALON, that can take off and land in the vertical position in small areas, change its position from vertical to horizontal and fly horizontally as a fixed-wing following a reference path built from straight lines segments, all without assistance of a human pilot. Moreover, to accomplish all these requirements, other contributions were derived:

- Kinematic and dynamic models of AVALON, implemented in a simulation environment with sensor noise and wind disturbances to validate the autopilot techniques (SILVA; MARCONATO; BRANCO, 2015; SILVA *et al.*, 2017).
- Development and comparison of Extended Kalman Filter and Unscented Kalman Filter as a GPS (Global Positioning System) aided inertial navigation with inclusion of magnetometer and barometer data under different configurations of sensor noise. UKF shows smaller errors in the attitude estimation than EKF (SILVA; WILSON; BRANCO, 2015).
- Extension of Lookahead, Pure Pursuit and Line-of-Sight, Nonlinear Guidance Law and Vector Field to contemplate changes in the altitude and comparison of these extended path-following algorithms under different intensities of wind disturbance. Lookahead, PLOS and Vector Field have a similar performance, but Vector Field is more robust to wind disturbances (PELIZER; SILVA; BRANCO, 2017b; PELIZER; SILVA; BRANCO, 2017a).
- A new control structure using computed torque and Incremental Nonlinear Dynamic Inversion, which considers the system nonlinearities, and its comparison with a classical method based only in PID controllers. The new control strategy presents better accuracy than the classical method (SILVA *et al.*, 2017; SILVA *et al.*, 2018).

8.2 Limitations

The initial project proposed the design of a control structure based only in PID controllers, according to the classical methods used by other tailsitters. However, simulation results considering the wind disturbances have shown the inability of this solution, which was improved by DI with PID structure.

However, a clear limitation of DI with PID structure is its vulnerability to the estimation error of the sideslip angle, which results in failure during the landing. Moreover, the simulation environment did not include model uncertainties, which could also result in unstable conditions during the flights.

In addition, the metrics defined to optimise the gains for each flight stage controller considered only performance and not the rejection of disturbances, which could explain the large number of failures for Decoupled PID.

Another difficulty of the project that must be pointed out was the construction of the prototype, which required technical expertise in aircraft manufacture. Such problem was only solved by the assistance of the EESC USP Aerodesign, an extracurricular organisation composed by undergraduate students of aeronautical engineering, with the idealisation and conclusion of AVALON's prototype.

8.3 Future research

The challenges of developing a tailsitter UAV are vast and continuous, especially for a new one that was not extensively tested yet. Further evaluations of AVALON can be performed by considering model uncertainties, gusts of wind with larger intensities and other sensor problems, like scale factor and bias. Moreover, the inclusion of experimental flights would strengthen the validation of the autopilot.

The estimation of the angles of attack and sideslip can be improved by adding them to the state vector in the sensor fusion techniques, which, consequently, would improve the robustness of the DI with PID control.

Also, the optimisation procedure could incorporate the different configurations of sensor noise and the different intensities of wind in its processes. For example, the wind and noise could change for a set of generations in GA, producing parameters that would be evaluated in more than one condition. In this way, the optimised parameters could result in more robust controllers. Another solution to improve the robustness of the autopilot controllers would be to add other control methods to the comparison, like H-infinity or Model Predictive Control.

Finally, the overall system of AVALON would benefit with the addition of other

features necessary for a UAV operation, such as sense and avoid system, other communications architectures and a GCS to monitor AVALON during its operation.

8.4 Declaration of authorship

I confirm that this doctoral thesis has not been submitted in support of an application for another degree at this or any other teaching or research institution. It is the result of my own work and the use of all material from other sources has been properly and fully acknowledged. Research done in collaboration is also clearly indicated. Excerpts of this thesis have been either published or submitted for the appreciation of editorial boards of journals, conferences and workshops, according to the list of publications presented in the following section. My contributions to each publication are listed as well.

8.5 Publications list

8.5.1 Published papers

- **SILVA, N. B. F.; FONTES, J. V. C.; INOUE, R. S.; BRANCO, K. R. L. J. C. Dynamic inversion and gain-scheduling control for an autonomous aerial vehicle with multiple flight stages.** Journal of Control, Automation and Electrical Systems, v. 29, p. 328-339, 2018.

Contribution level: High – the author was the main investigator of this paper.

- **PELIZER, G. V.; SILVA, N. B. F.; BRANCO, K. R. L. J. C. 3D Path-Following Algorithms for Unmanned Aerial Vehicles Adjusted with Genetic Algorithm.** In: Branco K., Pinto A., Pigatto D. (eds) Communication in Critical Embedded Systems. WoCCES 2013, WoCCES 2014, WoCCES 2015, WoCCES 2016. Communications in Computer and Information Science, vol 702. Springer, Cham, 2017.

Contribution level: High – the author was one of the main investigators of this paper.

- **PIGATTO, D. F.; GONÇALVES, L.; ROBERTO, G. F.; RODRIGUES FILHO, J. F.; SILVA, N. B. F.; PINTO, A. R.; BRANCO, K. R. L. J. C. The HAMSTER Data Communication Architecture for Unmanned Aerial, Ground and Aquatic Systems.** Journal of Intelligent & Robotic Systems, v. 1, p. 1-19, 2016.

Contribution level: Medium – the author contributed with the case study development and analysis.

- PATIRE, L. H. G.; **SILVA, N. B. F.**; BRANCO, K. R. L. J. C. **Data Fusion Techniques Applied to Takeoff and Landing Procedures - A VTOL Case Study**. Revista IEEE América Latina, v. 14, p. 3962-3966, 2016.

Contribution level: Medium – the author contributed with the sensor fusion development and analysis.

- **SILVA, N. B. F.**; MARCONATO, E. A.; BRANCO, K. R. L. J. C. **AVALON: definition and modeling of a vertical takeoff and landing UAV**. Journal of Physics. Conference Series (Print), v. 633, p. 012125, 2015.

Contribution level: High – the author was the main investigator of this paper.

- **SILVA, N. B. F.**; PIGATTO, D. F.; MARTINS, P. S.; BRANCO, K. R. L. J. C. **Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer**. Journal of Network and Computer Applications, v. 60, p. 130-143, 2015.

Contribution level: High – the author was one of the main investigators of this paper.

- **SILVA, N. B. F.**; FONTES, J. V. C.; INOUE, R. S.; BRANCO, K. R. L. J. C. **Development of a fixed-wing vertical takeoff and landing aircraft as an autonomous vehicle**. 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR), Curitiba, 2017, pp. 1-6.

Contribution level: High – the author was the main investigator of this paper.

- PELIZER, G. V.; **SILVA, N. B. F.**; BRANCO, K. R. L. J. C. **Comparison of 3D path-following algorithms for unmanned aerial vehicles**. 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 2017, pp. 498-505.

Contribution level: High – the author was one of the main investigators of this paper.

- PELIZER, G. V.; **SILVA, N. B. F.**; BRANCO, K. R. L. J. C. **Um algoritmo de navegação não linear 3D para veículos aéreos não tripulados**. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2016), 2016, Salvador. IV Workshop of Communication in Critical Embedded Systems (WoC-CES), 2016. v. 1. p. 67-75.

Contribution level: High – the author was one of the main investigators of this paper.

- **SILVA, N. B. F.**; PIGATTO, D. F.; PIRES, R. M.; MARTINS, P. S.; BRANCO, K. R. L. J. C. **Case studies of performance evaluation of asymmetric and**

symmetric cryptographic algorithms for embedded systems. In: XII Simpósio Brasileiro de Automação Inteligente, 2015, Natal. XII Simpósio Brasileiro de Automação Inteligente, 2015.

Contribution level: High – the author was one of the main investigators of this paper.

- **SILVA, N. B. F.; WILSON, D. B.; BRANCO, K. R. L. J. C. Performance evaluation of the Extended Kalman Filter and Unscented Kalman Filter.** In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 2015, Denver. 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 2015. p. 733-741.

Contribution level: High – the author was the main investigator of this paper.

- **QUERINO FILHO, L. C.; RODRIGUES FILHO, J. F.; SILVA, N. B. F.; BRANCO, K. R. L. J. C. Attitude estimation of a simulated flight and GPS positioning with Kalman filter.** In: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 2015, Denver. 2015 International Conference on Unmanned Aircraft Systems (ICUAS), 2015. p. 742-750.

Contribution level: Low – the author assisted with the concepts and with the analysis of the paper.

- **SILVA, N. B. F.; BRANCO, K. R. L. J. C. A new concept of VTOL as fixed-wing.** In: 2013 International Conference on Unmanned Aircraft Systems (ICUAS), 2013, Atlanta. 2013 International Conference on Unmanned Aircraft Systems (ICUAS). p. 811-817.

Contribution level: High – the author was the main investigator of this paper.

- **SALLA, G. C.; SARTIN, A. M.; SILVA, N. B. F.; PIGATTO, D. F.; BRANCO, K. R. L. J. C. Performance Evaluation of Security Communication in Critical Embedded Systems.** In: CBSEC (Brazilian Conference on Critical Embedded Systems), 2012, Campinas. II Brazilian Conference on Critical Embedded Systems, 2012. v. 1. p. 54-57.

Contribution level: Low – the author participated on the research definition and results analysis.

BIBLIOGRAPHY

AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL (ANAC). **Requisitos gerais para aeronaves não tripuladas de uso civil**. 2017. RBAC-E no 94/2017. Citation on page 36.

ALIGN CORPORATION LIMITE. **RCM-BL700MX Brushless Motor User Manual**. 2018. Citation on page 138.

ANALOG DEVICES. **Datasheet ADIS16407**. 2011. Rev. C. Citations on pages 70 e 139.

ARDUINO. **Arduino Leonardo**. 2018. Accessed: 27/02/2018. Available: <<https://www.arduino.cc/en/Main/arduinoBoardLeonardo>>. Citation on page 139.

ARGYLE, M. E.; BEACH, J. M.; BEARD, R. W.; MCLAIN, T. W.; MORRIS, S. Quaternion based attitude error for a tailsitter in hover flight. In: **2014 American Control Conference**. Portland, USA, 2014. p. 1396–1401. ISSN 0743-1619. Citation on page 110.

ARGYLE, M. E.; BEARD, R. W.; MORRIS, S. The Vertical Bat tail-sitter: Dynamic model and control architecture. In: **American Control Conference (ACC)**. Washington, USA, 2013. p. 806–811. ISSN 0743-1619. Citations on pages 31 e 39.

ÅSTRÖM, K.; HÄGGLUND, T. **Advanced Pid Control**. USA: ISA-The Instrumentation, Systems and Automation Society, 2006. 460 p. ISBN 9781556179426. Citations on pages 33 e 84.

AUSTIN, R. **Unmanned Aircraft Systems: UAVS Design, Development and Deployment**. USA: Wiley, 2010. (Aerospace Series). ISBN 9780470664803. Citations on pages 32 e 35.

BACON, B. J.; OSTROFF, A. J.; JOSHI, S. M. Reconfigurable ndi controller using inertial sensor failure detection and isolation. **IEEE Transactions on Aerospace and Electronic Systems**, v. 37, n. 4, p. 1373–1383, Oct 2001. ISSN 0018-9251. Citation on page 114.

BAKER, M. J. **Maths - Conversion Quaternion to Euler**. 2017. Accessed: 10/01/2018. Available: <<http://www.euclideanspace.com/maths/geometry/rotations/conversions/quaternionToEuler/>>. Citation on page 72.

BALASUBRAMANIAN, E.; VASANTHARAJ, R. Dynamic modeling and control of quad rotor. **International Journal of Engineering and Technology (IJET)**, v. 5, p. 63–69, 2013. Citation on page 109.

BAPST, R.; RITZ, R.; MEIER, L.; POLLEFEYS, M. Design and implementation of an unmanned tail-sitter. In: **IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. Hamburg, Germany, 2015. p. 1885–1890. Citations on pages 39, 41, 42 e 45.

BILODEAU, P.-R.; WONG, F. Modeling and control of a hovering mini tail-sitter. **International Journal of Micro Air Vehicles**, v. 2, n. 4, p. 211–220, 2010. ISSN 17568293. Citations on pages [39](#), [43](#), [45](#), [46](#) e [109](#).

BREIVIK, M.; FOSSEN, T. I. Principles of guidance-based path following in 2d and 3d. In: **Proceedings of the 44th IEEE Conference on Decision and Control**. Seville, Spain, 2005. p. 627–634. ISSN 0191-2216. Citations on pages [33](#), [83](#), [86](#), [87](#) e [88](#).

CETINSOY, E. Design and control of a gas-electric hybrid quad tilt-rotor uav with morphing wing. In: **2015 International Conference on Unmanned Aircraft Systems (ICUAS)**. Denver, CO, USA, 2015. p. 82–91. Citations on pages [31](#) e [39](#).

CHEN, H.; HE, K.; QIAN, W. Attitude control of uav based on pi dynamic inversion. In: **2016 35th Chinese Control Conference (CCC)**. Chengdu, China, 2016. p. 10668–10672. Citation on page [109](#).

COOKE, N.; ROWE, L.; BENNETT, W.; JORALMON, D. **Remotely Piloted Aircraft Systems: A Human Systems Integration Perspective**. USA: Wiley, 2016. (Aerospace Series). ISBN 9781118965917. Citation on page [36](#).

CRASSIDIS, J. L.; JUNKINS, J. L. **Optimal Estimation of Dynamic Systems, Second Edition**. 2nd. ed. United Kingdom: Chapman & Hall/CRC, 2011. ISBN 1439839859, 9781439839850. Citation on page [64](#).

CURRY, R.; LIZARRAGA, M.; MAIRS, B.; ELKAIM, G. L+2, an improved line of sight guidance law for UAVs. In: **American Control Conference (ACC), 2013**. Washington, DC, USA, 2013. p. 1–6. ISSN 0743-1619. Citation on page [89](#).

DEPARTAMENTO DE CONTROLE DO ESPAÇO AÉREO (DECEA). **Sistemas de Aeronaves Remotamente Pilotadas e o Acesso ao Espaço Aéreo Brasileiro**. 2016. ICA 100-40. Citation on page [36](#).

E-FLITE. **80-Amp Pro Switch-Mode BEC Brushless ESC**. 2013. Citation on page [138](#).

ESCARENO, J.; SANCHEZ, A.; GARCIA, O.; LOZANO, R. Modeling and Global Control of the Longitudinal Dynamics of a Coaxial Convertible Mini-UAV in Hover Mode. **Journal of Intelligent and Robotic Systems**, v. 54, n. 1-3, p. 261–273, Jul. 2008. ISSN 0921-0296. Citations on pages [39](#), [42](#) e [45](#).

ETKIN, B.; REID, L. **Dynamics of flight: stability and control**. 3. ed. USA: Wiley, 1996. ISBN 0471034185. Citation on page [53](#).

FLETCHER, R. **Practical Methods of Optimization**. USA: Wiley, 2013. ISBN 9781118723180. Citation on page [94](#).

FRANCESCO, G. D.; MATTEI, M. Modeling and incremental nonlinear dynamic inversion control of a novel unmanned tiltrotor. **Journal of Aircraft**, v. 53, n. 1, p. 73–86, Apr 2015. ISSN 0021-8669. Citations on pages [110](#) e [115](#).

FRESK, E.; NIKOLAKOPOULOS, G. Full quaternion based attitude control for a quadrotor. In: **2013 European Control Conference (ECC)**. Zurich, Switzerland, 2013. p. 3864–3869. Citation on page [111](#).

GARCIA, O.; CASTILLO, P.; WONG, K.; LOZANO, R. Attitude stabilization with real-time experiments of a tail-sitter aircraft in horizontal flight. **Journal of Intelligent and Robotic Systems: Theory and Applications**, v. 65, n. 1-4, p. 123–136, 2012. ISSN 09210296. Citation on page [39](#).

GARCIA-SALAZAR, O.; SANCHEZ, A.; CASTRO, J. Escareño; LOZANO, R. Tail-sitter UAV having one tilting rotor: Modeling, control and real-time experiments. In: **IFAC Proceedings Volumes (IFAC-PapersOnline)**. Seoul, 2008. v. 17, n. 1 PART 1. ISBN 9783902661005. ISSN 14746670. Citation on page [39](#).

GIANNITRAPANI, A.; CECCARELLI, N.; SCORTECCI, F.; GARULLI, A. Comparison of EKF and UKF for spacecraft localization via angle measurements. **IEEE Transactions on Aerospace and Electronic Systems**, v. 47, n. 1, p. 75–84, 2011. Citation on page [61](#).

GUERRERO, J. A.; LOZANO, R.; ROMERO, G.; LARA-ALABAZARES, D.; WONG, K. C. Robust control design based on sliding mode control for hover flight of a mini tail-sitter Unmanned Aerial Vehicle. In: **35th Annual Conference of IEEE Industrial Electronics (IECON)**. Porto, Portugal, 2009. p. 2342–2347. ISSN 1553-572X. Citation on page [39](#).

HANG, Y.; JIHONG, Z.; XIAMING, Y.; CHAO, Z. Robust hover control of thrust-vectorized unmanned tail-sitter aircraft against gust load. In: **Control and Automation (ICCA), 2013 10th IEEE International Conference on**. Hangzhou, China, 2013. p. 735–738. ISSN 1948-3449. Citation on page [39](#).

HARDKERNEL. **User manual ODROID-XU4**. 2015. Citation on page [139](#).

HU, Q.; FEI, Q.; WU, Q.; GENG, Q. Flight control design for uav using hierarchical dynamic inversion. In: **International Conference on Automatic Control and Artificial Intelligence (ACAI 2012)**. Xiamen, China, 2012. p. 1069–1072. Citation on page [109](#).

INTERNATIONAL CIVIL AVIATION ORGANIZATION (ICAO). **Manual of ICAO Standard Atmosphere**. 1964. Doc 7488/2. Citation on page [63](#).

JAIN, R. **The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling**. USA: Wiley, 1991. I-XXVII, 1-685 p. (Wiley professional computing). ISBN 978-0-471-50336-1. Citations on pages [79](#), [98](#) e [99](#).

JULIER, S. J.; UHLMANN, J. K. A New Extension of the Kalman Filter to Nonlinear Systems. In: **Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls**. Orlando, FL, United States, 1997. p. 182–193. Citations on pages [33](#) e [66](#).

JUNG, Y.; SHIM, D. H. Development and application of controller for transition flight of tail-sitter UAV. **Journal of Intelligent & Robotic Systems**, v. 65, n. 1, p. 137–152, 2012. ISSN 1573-0409. Citations on pages [39](#) e [110](#).

KITA, K.; KONNO, A.; UCHIYAMA, M. Transition between level flight and hovering of a tail-sitter vertical takeoff and landing aerial robot. **Advanced Robotics**, v. 24, n. 5-6, p. 763–781, 2010. ISSN 01691864. Citations on pages [39](#), [44](#), [45](#) e [46](#).

KOHNO, S.; UCHIYAMA, K. Design of robust controller of fixed-wing uav for transition flight. In: **2014 International Conference on Unmanned Aircraft Systems (ICUAS)**. Orlando, FL, USA, 2014. p. 1111–1116. Citation on page [109](#).

KRAFT, E. A quaternion-based unscented kalman filter for orientation tracking. In: **Proceedings of the Sixth International Conference of Information Fusion**. Cairns, Queensland, Australia, 2003. v. 1, p. 47–54. Citation on page [67](#).

KUBO, D.; SUZUKI, S. Robust optimal autopilot design for a tail-sitter mini unmanned aerial vehicle. **Journal of Aerospace Computing, Information and Communication**, v. 5, n. 5, p. 135–154, 2008. ISSN 15429423. Citations on pages [39](#), [41](#) e [45](#).

LEWIS, F. **Optimal estimation: with an introduction to stochastic control theory**. USA: Wiley, 1986. (A Wiley-interscience publication). ISBN 9780471837411. Citations on pages [33](#) e [64](#).

LEWIS, F.; DAWSON, D.; ABDALLAH, C. **Robot Manipulator Control: Theory and Practice**. United Kingdom: CRC Press, 2003. (Automation and Control Engineering). ISBN 9780203026953. Citations on pages [33](#), [109](#) e [113](#).

LOCOSYS. **Datasheet of GPS smart antenna module, LS20030 3**. 2009. Rev. 1.2. Citations on pages [70](#) e [140](#).

LOTUFO, M. A.; COLANGELO, L.; PEREZ-MONTENEGRO, C.; CANUTO, E. The feedback linearisation method for embedded model control: The boreas project case-study. In: **2015 23rd Mediterranean Conference on Control and Automation (MED)**. Torremolinos, Spain, 2015. p. 501–507. Citation on page [109](#).

MATSUMOTO, T.; KITA, K.; SUZUKI, R.; OOSEDO, A.; GO, K.; HOSHINO, Y.; KONNO, A.; UCHIYAMA, M. A hovering control strategy for a tail-sitter VTOL UAV that increases stability against large disturbance. In: **2010 IEEE International Conference on Robotics and Automation (ICRA)**. Anchorage, AK, USA, 2010. p. 54–59. ISSN 1050-4729. Citations on pages [32](#), [109](#) e [112](#).

MAXBOTIX INC. **Datasheet I2CXL-MaxSonar- EZ™ Series**. 2012. Citation on page [140](#).

MEASUREMENT SPECIALTIES. **MS4525DO Specifications**. 2017. Citation on page [140](#).

MELIN, T. A vortex lattice MATLAB implementation for linear aerodynamic wing applications. **Master's Thesis, Department of Aeronautics, Royal Institute of Technology (KTH), Stockholm, Sweden**, 2000. Citation on page [54](#).

MITCHELL, M. **An Introduction to Genetic Algorithms**. Cambridge, MA, USA: MIT Press, 1998. ISBN 0262631857. Citation on page [94](#).

MIYAZAKI, K.; TSUBAKINO, D. Nonlinear hovering control laws for small vectored-thrust tail-sitter uavs. In: **2017 IEEE Conference on Control Technology and Applications (CCTA)**. Mauna Lani, HI, USA, 2017. p. 2102–2107. Citations on pages [31](#) e [39](#).

MOHAMED, M. K.; LANZON, A. Effect of unmodelled actuator dynamics on feedback linearised systems and a two stage feedback linearisation method. In: **52nd IEEE Conference on Decision and Control**. Florence, Italy, 2013. p. 841–846. ISSN 0191-2216. Citation on page 109.

MOIR, I.; SEABRIDGE, A. **Aircraft Systems: Mechanical, electrical and avionics subsystems integration**. 3. ed. Florence, Italy: John Wiley and Sons, 2008. (Aerospace Series). ISBN 978-0-470-05996-8. Citation on page 37.

NAIDU, D. S.; CALISE, A. J. Singular perturbations and time scales in guidance and control of aerospace systems: a survey. **Journal of Guidance Control and Dynamics**, AMERICAN INST OF AERONAUTICS AND ASTRONAUTICS, v. 24, n. 6, p. 1057–1078, 2001. Citation on page 116.

NALDI, R.; MARCONI, L. Robust control of transition maneuvers for a class of V/STOL aircraft. **Automatica**, v. 49, n. 6, p. 1693–1704, 2013. ISSN 00051098. Citations on pages 31 e 39.

NATIONAL IMAGERY AND MAPPING AGENCY. **Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems**. 3rd. ed. 2000. Accessed: 27/02/2018. Available: <http://earth-info.nga.mil/GandG/publications/tr8350.2/tr8350_2.html>. Citation on page 63.

NELSON, D.; BARBER, D.; MCLAIN, T.; BEARD, R. Vector field path following for miniature air vehicles. **IEEE Transactions on Robotics**, v. 23, n. 3, p. 519–529, June 2007. ISSN 1552-3098. Citations on pages 33, 83 e 90.

NONAMI, K. Classification of uav platforms. In: **Modeling and Control of Unmanned Small Scale Rotorcraft UAVs & MAVs**. Japan: Springer Japan, 2010. chap. 7, p. 10–12. ISBN 9784431538561. Citation on page 31.

OOSEDO, A.; ABIKO, S.; KONNO, A.; KOIZUMI, T.; FURUI, T.; UCHIYAMA, M. Development of a quad rotor tail-sitter VTOL UAV without control surfaces and experimental verification. In: **Robotics and Automation (ICRA), 2013 IEEE International Conference on**. Karlsruhe, Germany, 2013. p. 317–322. ISSN 1050-4729. Citation on page 39.

PAPACHRISTOS, C.; ALEXIS, K.; TZES, A. Model predictive hovering-translation control of an unmanned Tri-TiltRotor. In: **2013 IEEE International Conference on Robotics and Automation (ICRA)**. Karlsruhe, Germany, 2013. p. 5425–5432. ISSN 1050-4729. Citation on page 39.

PARK, S.; DEYST, J.; HOW, J. P. Performance and lyapunov stability of a nonlinear path following guidance method. **Journal of Guidance, Control, and Dynamics**, v. 30, n. 6, p. 1718–1728, November 2007. ISSN 0731-5090. Citations on pages 33, 83 e 89.

PELIZER, G. V.; SILVA, N. B. F.; BRANCO, K. R. L. J. C. 3D path-following algorithms for unmanned aerial vehicles adjusted with genetic algorithm. In: BRANCO, K.; PINTO, A.; PIGATTO, D. (Ed.). **Communication in Critical Embedded Systems: WoCCES 2013, WoCCES 2014, WoCCES 2015, WoCCES 2016, Revised Selected Papers**. Brazil: Springer International Publishing, 2017. p. 63–80. Citations on pages 34, 86 e 144.

PELIZER, G. V.; SILVA, N. B. F. da; BRANCO, K. R. L. J. Comparison of 3D path-following algorithms for unmanned aerial vehicles. In: **2017 International Conference on Unmanned Aircraft Systems (ICUAS)**. Miami, FL, USA, 2017. p. 498–505. Citations on pages 34 e 144.

PENNEC, X. **Computing the Mean of Geometric Features Application to the Mean Rotation**. 1998. Research report RR-3371. Citation on page 67.

PERKINS, C.; HAGE, R. **Airplane Performance, Stability and Control**. USA: Wiley, 1949. ISBN 9780471680468. Citations on pages 48 e 54.

RITZ, R.; D'ANDREA, R. A global controller for flying wing tailsitter vehicles. In: **2017 IEEE International Conference on Robotics and Automation (ICRA)**. Singapore, 2017. p. 2731–2738. Citations on pages 31, 39, 41, 42 e 45.

ROGERS, R. **Applied Mathematics in Integrated Navigation Systems**. USA: American Institute of Aeronautics and Astronautics, 2003. (AIAA education series, v. 1). ISBN 9781563476563. Citations on pages 50 e 62.

ROSAS, V.; LOZANO, R.; TORRES, J.; ZAMUDIO, Z. Modeling and control of a convertible airplane. In: **System Theory, Control and Computing (ICSTCC), 2012 16th International Conference on**. Sinaia, Romania, 2012. p. 1–6. Citation on page 39.

SADHANA, U.; KUMARI, B. L.; RAJU, K. P. Performance analysis of extended kalman filter and unscented kalman filter for doppler bearing passive target tracking. **International Journal of Power Control Signal and Computation (IJPCSC)**, v. 4, n. 2, p. 157–164, Apr 2012. ISSN 0976-268X. Citation on page 61.

SADRAEY, M.; COLGREN, R. Two DOF robust nonlinear autopilot design for a small UAV using a combination of dynamic inversion and h-infinity loop shaping. In: **AIAA Guidance, Navigation, and Control Conference and Exhibit**. San Francisco, CA, USA, 2005. p. 6402. Citation on page 116.

SAEED, A. S.; YOUNES, A. B.; ISLAM, S.; DIAS, J.; SENEVIRATNE, L.; CAI, G. A review on the platform design, dynamic modeling and control of hybrid uavs. In: **2015 International Conference on Unmanned Aircraft Systems (ICUAS)**. Denver, CO, USA, 2015. p. 806–815. Citations on pages 31 e 40.

SERVODATABASE.COM. **Hobbico CS-12MG - High-Speed Micro MG Servo**. 2018. Accessed: 27/02/2018. Available: <<https://servodatabase.com/servo/hobbico/cs-12mg>>. Citation on page 138.

SICILIANO, B.; SCIAVICCO, L.; VILLANI, L. **Robotics : modelling, planning and control**. London: Springer, 2009. (Advanced Textbooks in Control and Signal Processing). 013-81159. ISBN 1-8462-8641-7. Citations on pages 33, 102, 109 e 113.

SIEBERLING, S.; CHU, Q. P.; MULDER, J. A. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. **Journal of Guidance, Control, and Dynamics**, American Institute of Aeronautics and Astronautics, Reston, US-VA, v. 33, n. 6, p. 1732–1742, November–December 2010. Citations on pages 33, 110 e 114.

SILVA, N.; WILSON, D.; BRANCO, K. Performance evaluation of the extended Kalman filter and unscented Kalman filter. In: **International Conference on Unmanned Aircraft Systems (ICUAS)**. Denver, CO, USA, 2015. p. 733–741. Citations on pages 34 e 144.

SILVA, N. B. F.; FONTES, J. V. C.; INOUE, R. S.; BRANCO, K. R. L. J. C. Development of a fixed-wing vertical takeoff and landing aircraft as an autonomous vehicle. In: **2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)**. Curitiba, Brazil, 2017. p. 1–6. Citations on pages 34 e 144.

_____. Dynamic inversion and gain-scheduling control for an autonomous aerial vehicle with multiple flight stages. **Journal of Control, Automation and Electrical Systems**, v. 29, n. 3, p. 328–339, Jun 2018. ISSN 2195-3899. Citations on pages 34 e 144.

SILVA, N. B. F.; MARCONATO, E. A.; BRANCO, K. R. L. J. C. AVALON: definition and modeling of a vertical takeoff and landing UAV. **Journal of Physics: Conference Series**, v. 633, n. 1, p. 012125, 2015. Citations on pages 34, 47 e 144.

SIMPLICIO, P. Helicopter nonlinear flight control using incremental nonlinear dynamic inversion. **Master's Thesis, Delft University of Technology**, 2011. Citation on page 116.

SINHA, P.; ESDEN-TEMPSKI, P.; FORRETTE, C. A.; GIBBONEY, J. K.; HORN, G. M. Versatile, modular, extensible vtol aerial platform with autonomous flight mode transitions. In: **Aerospace Conference, 2012 IEEE**. Big Sky, MT, USA, 2012. p. 1–17. ISSN 1095-323X. Citations on pages 39, 43 e 45.

SMEUR, E. J.; CHU, Q.; CROON, G. C. de. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. **Journal of Guidance, Control, and Dynamics**, American Institute of Aeronautics and Astronautics, 2015. Citation on page 110.

SNELL, S. A.; ENNS, D.; GARRARD, W.; L, J. A. M. Nonlinear inversion flight control for a supermaneuverable aircraft. **Journal of guidance, control, and dynamics**, v. 15, n. 4, p. 976–984, 1992. Citation on page 116.

SOLA, J. **Quaternion kinematics for the error-state Kalman filter**. 2017. Accessed: 10/01/2018. Available: <<https://hal.archives-ouvertes.fr/hal-01122406>>. Citations on pages 64, 110 e 167.

SONG, H.; UNDERWOOD, C. A Mars VTOL Aerobot - Preliminary Design, Dynamics and Control. In: **IEEE Aerospace Conference**. Big Sky, MT, USA, 2007. p. 1–14. ISSN 1095-323X. Citation on page 39.

SPEKTRUM. **DX8 Instruction Manual**. 2018. Citation on page 138.

ST-PIERRE, M.; GINGRAS, D. Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system. In: **IEEE Intelligent Vehicles Symposium**. Parma, Italy, 2004. p. 831–835. Citation on page 61.

STEVENS, B.; LEWIS, F.; JOHNSON, E. **Aircraft Control and Simulation**. 3. ed. USA: Wiley, 2016. ISBN 9781118870983. Citation on page 51.

STONE, R.; ANDERSON, P.; HUTCHISON, C.; TSAI, A.; GIBBENS, P.; WONG, K. Flight testing of the T-wing tail-sitter unmanned air vehicle. **Journal of Aircraft**, v. 45, n. 2, p. 673–685, 2008. ISSN 00218669. Citations on pages [39](#), [40](#), [45](#) e [109](#).

SUJIT, P.; SARIPALLI, S.; SOUSA, J. B. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles. **IEEE Control Systems**, v. 34, n. 1, p. 42–59, Feb 2014. ISSN 1066-033X. Citation on page [83](#).

TA, D.; FANTONI, I.; LOZANO, R. Modeling and control of a convertible mini-UAV. In: **IFAC Proceedings Volumes (IFAC-PapersOnline)**. Milano, 2011. v. 18, p. 1492–1497. ISBN 9783902661937. ISSN 14746670. Citations on pages [39](#), [42](#) e [45](#).

TA, D. A.; FANTONI, I.; LOZANO, R. Modeling and control of a tilt tri-rotor airplane. In: **2012 American Control Conference (ACC)**. Montreal, QC, Canada, 2012. p. 131–136. ISSN 0743-1619. Citation on page [39](#).

TITTERTON, D.; WESTON, J.; ENGINEERS, I. of E. **Strapdown Inertial Navigation Technology, 2nd Edition**. UK: Institution of Engineering and Technology, 2004. ISBN 9780863413582. Citations on pages [62](#) e [63](#).

U.S. GOVERNMENT ACCOUNTABILITY OFFICE (U.S. GAO). **Unmanned aircraft systems: federal actions needed to ensure safety and expand their potential uses within the national airspace system**. 2008. GAO-08-511. Citation on page [35](#).

VALAVANIS, K. P. **Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy**. 1st. ed. USA: Springer Publishing Company, Incorporated, 2007. ISBN 1402061137, 9781402061134. Citations on pages [31](#) e [35](#).

VALAVANIS, K. P.; VACHTSEVANOS, G. J. **Handbook of Unmanned Aerial Vehicles**. USA: Springer Publishing Company, Incorporated, 2015. ISBN 9048197082, 9789048197088. Citations on pages [32](#) e [37](#).

VENKATRAMAN, K.; MANI, V.; KOTHARI, M.; POSTLETHWAITE, I.; GU, D.-W. A suboptimal path planning algorithm using rapidly-exploring random trees. **International Journal of Aerospace Innovations**, Multi Science Publishing, v. 2, n. 1-2, p. 93–104, 2010. Citations on pages [33](#), [83](#) e [90](#).

VERLING, S.; WEIBEL, B.; BOOSFELD, M.; ALEXIS, K.; BURRI, M.; SIEGWART, R. Full attitude control of a vtol tailsitter uav. In: **2016 IEEE International Conference on Robotics and Automation (ICRA)**. Stockholm, Sweden, 2016. p. 3006–3012. Citations on pages [39](#), [42](#) e [45](#).

WANG, K.; KE, Y.; LAI, S.; GONG, K.; TAN, Y.; CHEN, B. M. Model-based optimal auto-transition and control synthesis for tail-sitter uav kh-lion. In: **2017 13th IEEE International Conference on Control Automation (ICCA)**. Ohrid, Macedonia, 2017. p. 541–547. Citations on pages [31](#) e [39](#).

WANG, W.; ZHU, J.; KUANG, M. Design, modelling and hovering control of a tail-sitter with single thrust-vectoring propeller. In: **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. Vancouver, BC, Canada, 2017. p. 5971–5976. Citations on pages [31](#) e [39](#).

XILI, Y. b.; YONG, F. c.; JIHONG, Z. d. Transition flight control of two vertical/short takeoff and landing aircraft. **Journal of Guidance, Control, and Dynamics**, v. 31, n. 2, p. 371–385, 2008. ISSN 07315090. Citation on page [39](#).

ZHANG, B.; ZHU, X.; ZHOU, Z.; XU, M. Design of nonlinear flight control law for a tailless unmanned aerial vehicle. In: **2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)**. Xi'an, China, 2013. p. 1–4. ISSN 2159-3442. Citation on page [109](#).

ZHANG, C.; KOVACS, J. The application of small unmanned aerial systems for precision agriculture: a review. **Precision Agriculture**, Springer US, v. 13, n. 6, p. 693–712, 2012. ISSN 1385-2256. Citation on page [32](#).

ZHANG, C.; ZHU, J.; LIU, K.; YUAN, X.; GAO, Y. Model and longitudinal hover control of a conceptual thrust-vectorred unmanned tail-sitter. In: **Control and Automation (ICCA), 2013 10th IEEE International Conference on**. Hangzhou, China, 2013. p. 1714–1717. ISSN 1948-3449. Citation on page [39](#).

ZHOU, J.; LYU, X.; LI, Z.; SHEN, S.; ZHANG, F. A unified control method for quadrotor tail-sitter uavs in all flight modes: Hover, transition, and level flight. In: **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. Vancouver, BC, Canada, 2017. p. 4835–4841. Citations on pages [31](#) e [39](#).

AVALON AERODYNAMIC COEFFICIENTS

This appendix presents AVALON's aerodynamic coefficients obtained through Tornado's Vortex Lattice Method, as defined in Subsection 3.3.1. Tornado separates and normalises these coefficients to find the derivatives used in the dynamic and kinematic modelling, but for the sake of brevity, only the coefficients are shown here.

Figures 78 and 79 show C_{Lift} coefficient for intervals of α and β and for intervals of α and $\|\mathbf{v}_{rel}\|$ respectively. It is possible to see that C_{Lift} increases with α , but does not vary much with β and $\|\mathbf{v}_{rel}\|$.

Figure 78 – C_{Lift} coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figures 80 and 81 present C_{Drag} coefficient for intervals of α and β and for intervals of α and $\|\mathbf{v}_{rel}\|$ respectively. Drag coefficient varies symmetrically with α and β .

Figure 79 – C_{Lift} coefficient for a fixed sideslip of 0° and intervals of α and $\|\mathbf{v}_{rel}\|$ resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figure 80 – C_{Drag} coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figures 82 and 83 depict C_Y coefficient for intervals of α and β and for intervals of α and $\|\mathbf{v}_{rel}\|$ respectively. C_Y not only has a stronger influence of β , but also alters with α .

Figures 84 and 85 illustrate C_L coefficient for intervals of α and β and for intervals of α and $\|\mathbf{v}_{rel}\|$ respectively. C_L changes considerably with α and β .

Figures 86 and 87 present C_M coefficient for intervals of α and β and for intervals

Figure 81 – C_{Drag} coefficient for a fixed sideslip of 0° and intervals of α and $\|\mathbf{v}_{rel}\|$ resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figure 82 – C_Y coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figure 83 – C_Y coefficient for a fixed sideslip of 0° and intervals of α and $\|\mathbf{v}_{\text{rel}}\|$ resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figure 84 – C_L coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figure 85 – C_L coefficient for a fixed sideslip of 0° and intervals of α and $\|\mathbf{v}_{\text{rel}}\|$ resulted from Tornado's Vortex Lattice Method.

Source: Research data.

of α and $\|\mathbf{v}_{\text{rel}}\|$ respectively. C_M is more influenced by α .

Figure 86 – C_M coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figures 88 and 89 show C_N coefficient for intervals of α and β and for intervals of α and $\|\mathbf{v}_{\text{rel}}\|$ respectively. C_N not only alters strongly with β , but also changes with α .

Figure 87 – C_M coefficient for a fixed sideslip of 0° and intervals of α and $\|\mathbf{v}_{\text{rel}}\|$ resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figure 88 – C_N coefficient for a fixed velocity of 15 m/s and intervals of α and β resulted from Tornado's Vortex Lattice Method.

Source: Research data.

Figure 89 – C_N coefficient for a fixed sideslip of 0° and intervals of α and $\|\mathbf{v}_{\text{rel}}\|$ resulted from Tornado's Vortex Lattice Method.

Source: Research data.

It can be noticed that even for fixed values of one variable the coefficients vary with the change of the others, which reinforces the contribution of evaluating the derivatives in different flight conditions.

EKF JACOBIANS

This appendix presents the Jacobian matrices from AVALON's EKF implementation. The matrix $\mathbf{A}_k$ in (B.1) is the Jacobian of the process model represented by (4.14)-(4.16) with respect to $\mathbf{x}$ evaluated at $\tilde{\mathbf{x}}_k$, where $\mathbf{I}_{3 \times 3}$ is an identity matrix with size 3×3 , $\mathbf{O}_{3 \times 3}$ is a zero matrix with size 3×3 , $\mathbf{M}_{\tilde{\mathbf{a}}\mathbf{q}}$ is defined by (B.2), $\mathbf{M}_{\tilde{\omega}\mathbf{q}}$ is defined by (B.3), $\mathbf{M}_{\tilde{\omega}\mathbf{q}ii}$ are the diagonal elements of $\mathbf{M}_{\tilde{\omega}\mathbf{q}}$ defined by (B.4), $\tilde{a}_x$, $\tilde{a}_y$ and $\tilde{a}_z$ are the elements of the vector $\tilde{\mathbf{a}}$, i.e., $\tilde{\mathbf{a}} = [\tilde{a}_x, \tilde{a}_y, \tilde{a}_z]^T$, α_{rot} is the angle of the rotation $\tilde{\omega}\Delta t_f$ defined by (B.5) and $\mathbf{u}_{rot} = [u_{rot,x}, u_{rot,y}, u_{rot,z}]^T$ are the axes of this rotation defined by (B.6). The matrix $\mathbf{A}_k$ is also known as the transition matrix for the EKF implementation and its component $\mathbf{M}_{\tilde{\mathbf{a}}\mathbf{q}}$ is the transition matrix for the angular error dynamics based on Rodrigues rotation formula (SOLA, 2017).

$$\mathbf{A}_k = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \Delta t_f \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 4} \\ \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \Delta t_f \mathbf{M}_{\tilde{\mathbf{a}}\mathbf{q}3 \times 4} \\ \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{M}_{\tilde{\omega}\mathbf{q}3 \times 4} \end{bmatrix} \quad (\text{B.1})$$

$$\mathbf{M}_{\tilde{\mathbf{a}}\mathbf{q}} = 2 \begin{bmatrix} \tilde{a}_x q_1 + \tilde{a}_y q_4 - \tilde{a}_z q_3 & \tilde{a}_x q_2 + \tilde{a}_y q_3 + \tilde{a}_z q_4 & -\tilde{a}_x q_3 + \tilde{a}_y q_2 - \tilde{a}_z q_1 & -\tilde{a}_x q_4 + \tilde{a}_y q_1 + \tilde{a}_z q_2 \\ -\tilde{a}_x q_4 + \tilde{a}_y q_1 + \tilde{a}_z q_2 & \tilde{a}_x q_3 - \tilde{a}_y q_2 + \tilde{a}_z q_1 & \tilde{a}_x q_2 + \tilde{a}_y q_3 + \tilde{a}_z q_4 & -\tilde{a}_x q_1 - \tilde{a}_y q_4 + \tilde{a}_z q_3 \\ -\tilde{a}_x q_3 - \tilde{a}_y q_2 + \tilde{a}_z q_1 & \tilde{a}_x q_4 - \tilde{a}_y q_1 + \tilde{a}_z q_2 & \tilde{a}_x q_1 + \tilde{a}_y q_4 - \tilde{a}_z q_3 & \tilde{a}_x q_2 + \tilde{a}_y q_3 + \tilde{a}_z q_4 \end{bmatrix} \quad (\text{B.2})$$

$$\mathbf{M}_{\tilde{\omega}\mathbf{q}} = \begin{bmatrix} \mathbf{M}_{\tilde{\omega}\mathbf{q}ii} & -u_{rot,x} \sin \alpha_{rot} & -u_{rot,y} \sin \alpha_{rot} & -u_{rot,z} \sin \alpha_{rot} \\ u_{rot,x} \sin \alpha_{rot} & \mathbf{M}_{\tilde{\omega}\mathbf{q}ii} & u_{rot,z} \sin \alpha_{rot} & -u_{rot,y} \sin \alpha_{rot} \\ u_{rot,y} \sin \alpha_{rot} & -u_{rot,z} \sin \alpha_{rot} & \mathbf{M}_{\tilde{\omega}\mathbf{q}ii} & u_{rot,x} \sin \alpha_{rot} \\ u_{rot,z} \sin \alpha_{rot} & u_{rot,y} \sin \alpha_{rot} & -u_{rot,x} \sin \alpha_{rot} & \mathbf{M}_{\tilde{\omega}\mathbf{q}ii} \end{bmatrix} \quad (\text{B.3})$$

$$\mathbf{M}_{\tilde{\omega}\mathbf{q}_{ii}} = \cos \alpha_{rot} - \|\mathbf{u}_{rot}\|^2 (\cos \alpha_{rot} - 1) \quad (\text{B.4})$$

$$\alpha_{rot} = \|\tilde{\omega} \Delta t_f\| \quad (\text{B.5})$$

$$\mathbf{u}_{rot} = \frac{\tilde{\omega} \Delta t_f}{\alpha_{rot}} \quad (\text{B.6})$$

The matrix $\mathbf{W}_k$ in (B.7) is the Jacobian of the process model with respect to $\mathbf{w} = [\mathbf{w}_{\tilde{\mathbf{a}}}, \mathbf{w}_{\tilde{\omega}}]^T$ evaluated at $\tilde{\mathbf{x}}_k$, where $\mathbf{M}_{\mathbf{w}_{\tilde{\omega}}}$ is defined by (B.8)-(B.11) and $\tilde{P}$, $\tilde{Q}$ and $\tilde{R}$ are the elements of $\tilde{\omega}$, i.e., $\tilde{\omega} = [\tilde{P}, \tilde{Q}, \tilde{R}]^T$.

$$\mathbf{W}_k = \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{w}} = \begin{bmatrix} \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \\ \Delta t_f \mathbf{R}_{\mathbf{b}}^i & \mathbf{O}_{3 \times 3} \\ \mathbf{O}_{4 \times 3} & \mathbf{M}_{\mathbf{w}_{\tilde{\omega}}} \end{bmatrix} \quad (\text{B.7})$$

$$\mathbf{M}_{\mathbf{w}_{\tilde{\omega}}} = \begin{bmatrix} \mathbf{q} \otimes \mathbf{M}_{\mathbf{w}_{\tilde{\omega}},1} & \mathbf{q} \otimes \mathbf{M}_{\mathbf{w}_{\tilde{\omega}},2} & \mathbf{q} \otimes \mathbf{M}_{\mathbf{w}_{\tilde{\omega}},3} & \mathbf{q} \otimes \mathbf{M}_{\mathbf{w}_{\tilde{\omega}},4} \end{bmatrix} \quad (\text{B.8})$$

$$\mathbf{M}_{\mathbf{w}_{\tilde{\omega}},1} = \begin{bmatrix} \frac{-\Delta t_f^2 \tilde{P} \sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{2 \|\tilde{\omega}\|} \\ \frac{\sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|} - \frac{\tilde{P}^2 \sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} + \frac{\Delta t_f \tilde{P}^2 \cos(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{2 \|\tilde{\omega}\|^2} \\ \frac{\Delta t_f \tilde{P} \tilde{Q} \cos(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{2 \|\tilde{\omega}\|^2} - \frac{\tilde{P} \tilde{Q} \sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} \\ \frac{\Delta t_f \tilde{P} \tilde{R} \cos(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{2 \|\tilde{\omega}\|^2} - \frac{\tilde{P} \tilde{R} \sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} \end{bmatrix} \quad (\text{B.9})$$

$$\mathbf{M}_{\mathbf{w}_{\tilde{\omega}},2} = \begin{bmatrix} \frac{-\Delta t_f^2 \tilde{Q} \sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{2 \|\tilde{\omega}\|} \\ \frac{\Delta t_f \tilde{P} \tilde{Q} \cos(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{2 \|\tilde{\omega}\|^2} - \frac{\tilde{P} \tilde{Q} \sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} \\ \frac{\sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|} - \frac{\tilde{Q}^2 \sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} + \frac{\Delta t_f \tilde{Q}^2 \cos(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{2 \|\tilde{\omega}\|^2} \\ \frac{\Delta t_f \tilde{Q} \tilde{R} \cos(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{2 \|\tilde{\omega}\|^2} - \frac{\tilde{Q} \tilde{R} \sin(0.5 \Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} \end{bmatrix} \quad (\text{B.10})$$

$$\mathbf{M}_{\mathbf{w}\tilde{\omega},3} = \begin{bmatrix} \frac{-\Delta t_f^2 \tilde{R} \sin(0.5\Delta t_f^2 \|\tilde{\omega}\|)}{2\|\tilde{\omega}\|} \\ \frac{\Delta t_f \tilde{P} \tilde{R} \cos(0.5\Delta t_f^2 \|\tilde{\omega}\|)}{2\|\tilde{\omega}\|^2} - \frac{\tilde{P} \tilde{R} \sin(0.5\Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} \\ \frac{\Delta t_f \tilde{Q} \tilde{R} \cos(0.5\Delta t_f^2 \|\tilde{\omega}\|)}{2\|\tilde{\omega}\|^2} - \frac{\tilde{Q} \tilde{R} \sin(0.5\Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} \\ \frac{\sin(0.5\Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|} - \frac{\tilde{R}^2 \sin(0.5\Delta t_f^2 \|\tilde{\omega}\|)}{\Delta t_f \|\tilde{\omega}\|^3} + \frac{\Delta t_f \tilde{R}^2 \cos(0.5\Delta t_f^2 \|\tilde{\omega}\|)}{2\|\tilde{\omega}\|^2} \end{bmatrix} \quad (\text{B.11})$$

For AVALON's EKF, there is two update stages that occurs according to the availability of data, with priority to the GPS that has a slower frequency. Therefore, there are two observation matrices, $\mathbf{H}_{\mathbf{k},\text{GPS}}$, which is the Jacobian of the GPS observation model with respect to $\mathbf{x}$ evaluated at $\tilde{\mathbf{x}}_{\mathbf{k}}$ and $\mathbf{H}_{\mathbf{k},\text{MagPre}}$, which is the Jacobian of the magnetometer and barometer observation model represented by (4.20) with respect to $\mathbf{x}$ evaluated at $\tilde{\mathbf{x}}_{\mathbf{k}}$. $\mathbf{H}_{\mathbf{k},\text{GPS}}$ is defined by (B.12) and $\mathbf{H}_{\mathbf{k},\text{MagPre}}$ is defined by (B.13), where $\mathbf{M}_{\text{Mag}}$ is represented by (B.14).

$$\mathbf{H}_{\mathbf{k},\text{GPS}} = \frac{\partial \mathbf{h}_{\text{GPS}}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 7} \end{bmatrix} \quad (\text{B.12})$$

$$\mathbf{H}_{\mathbf{k},\text{MagPre}} = \frac{\partial \mathbf{h}_{\text{MagPre}}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & \mathbf{O}_{1 \times 3} & \mathbf{M}_{\text{Mag}} \\ 0 & 0 & -1 & \mathbf{O}_{1 \times 3} & \mathbf{O}_{1 \times 4} \end{bmatrix} \quad (\text{B.13})$$

$$\mathbf{M}_{\text{Mag}} = \begin{bmatrix} - \left(\frac{2q_2}{2q_3^2 + 2q_4^2 - 1} - \frac{4q_3(2q_1q_4 + 2q_2q_3)}{(2q_3^2 + 2q_4^2 - 1)^2} \right) \left(\frac{(2q_3^2 + 2q_4^2 - 1)^2}{(2q_3^2 + 2q_4^2 - 1)^2 + 4(q_1q_4 + q_2q_3)^2} \right) \\ - \left(\frac{2q_1}{2q_3^2 + 2q_4^2 - 1} - \frac{4q_4(2q_1q_4 + 2q_2q_3)}{(2q_3^2 + 2q_4^2 - 1)^2} \right) \left(\frac{(2q_3^2 + 2q_4^2 - 1)^2}{(2q_3^2 + 2q_4^2 - 1)^2 + 4(q_1q_4 + q_2q_3)^2} \right) \\ - \frac{2q_4(2q_3^2 + 2q_4^2 - 1)}{((2q_3^2 + 2q_4^2 - 1)^2 + 4(q_1q_4 + q_2q_3)^2)} \\ - \frac{2q_3(2q_3^2 + 2q_4^2 - 1)}{((2q_3^2 + 2q_4^2 - 1)^2 + 4(q_1q_4 + q_2q_3)^2)} \end{bmatrix}^T \quad (\text{B.14})$$