Politecnico di Milano
School of Industrial and Information Engineering
Master of Science in Aeronautical Engineering
Aerospace Science and Technology Department

Optimal and robust UAV state estimation based on Gps and optical flow

Advisor: Prof. Marco LOVERA
Co-Advisor: Eng. Mattia GIURATO

Thesis by:
Simone MUSACCHIO Matr. 863436

Academic Year 2017–2018

A Milano e al mio grande Sud.

Acknowledgments

Questo lavoro di tesi rappresenta il culmine di un lungo percorso, caratterizzato dal sostegno di numerose persone importanti che ritengo opportuno ringraziare.

Il primo sentito ringraziamento va al Professor Lovera Marco che mi ha permesso di intraprendere questo tortuoso ed eccitante lavoro, indicandomi sempre la giusta strada da percorrere e dandomi la possibilità di arricchirmi con i suoi preziosi insegnamenti.

Per questo lavoro di tesi va ringraziato Mattia Giurato che mi ha accompagnato in questo mondo dei droni sin dal primo giorno, mettendo a mia disposizione la sua passione ed esperienza.

Non bisogna dimenticare di ringraziare con entusiasmo tutti i ragazzi del laboratorio ASCL che hanno accompagnato questo percorso di tesi con piena disponibilità al supporto e al contempo, allietato con grandi sorrisi le lunghe giornate trascorse in laboratorio.

Questi anni milanesi non sarebbero stati gli stessi senza chi ha dato ANIMA al mio soggiorno e che spero di non perdere mai, e quindi un enorme GRAZIE: ai ragazzi di Soave e di NZLT, compagni di viaggio, di serate, di degrado, di cene, di pranzi, di lacrime e di sorrisi, di accese discussioni e infuocate risate...semplicemente compagni di vita; agli amici di On-board che hanno travolto con incommensurata positività questi miei ultimi periodi milanesi, con l'augurio di avervi sempre accanto; a voi Püties che avete corso al mio fianco nelle folli serate meneghine e nelle tavolate pregne di gioia e ricordi, siete unici; a chi, sin dall'inizio, con affetto e sorrisi, ha catapultato questo caro terroncino nelle proprie vite milanesi, Fillo, Ema, Leo, Ale, Sava.

Se mai vi dovessero chiedere cosa sia un uomo senza le sue radici, rispondetegli: il nulla. E quindi un immenso ringraziamento a chi è stato protagonista della mia infanzia e crescita, punto fermo dei miei ricordi, compagno nei momenti di luce e di buio...GRAZIE a voi Amici di Cerzeto, siete la mia fortezza.

Quanto è doveroso ringraziare te, VM...che ti sei preso gli anni della mia formazione, li hai inghiottiti e li hai trasformati in splendide amicizie che porterò con me per il resto della mia vita.

Un immenso ringraziamento a tutti i miei zii, cugini, nonni che hanno sempre remato a mio favore e che per sempre mi sosterranno, creditori a vita del mio affetto... siete la tavola imbandita, siete il focolare d'inverno, siete l'abbraccio più

vero, siete la mia Famiglia. Un particolare e smisurato ringraziamento a chi, sin dal primo giorno di questo mio lungo viaggio lontano da casa, mi ha accolto come un figlio ed un fratello, diventando il mio rifugio nei momenti difficili e il mio forziere di affetto e sorrisi...grazie Zia Anna, Zio Valdemiro, Francesco e Martina, per sempre al vostro fianco.

Se il cuore si aprisse, urlerebbe GRAZIE ogni volta che tu respiri... e quindi grazie a te, Sofia, compagna di sogni, energia fresca nella salita, luce nei momenti più bui, ali di questa bisognosa carlinga.

E infine il GRAZIE più grande di tutti a chi rappresenta l'alpha e l'omega di questo lungo percorso, chi sarà per sempre detentore del mio amore e della mia gratitudine, chi mi ha spinto sin dal primo giorno e chi mi ha reso l'uomo che sono... questo risultato è innanzitutto il vostro, siete i miei eroi Mamma e Papà.

Abstract

Nowadays the UAVs (Unmanned Aerial Vehicles) represent a wide field of research and development because of the versatility shown in a lot of application fields, both military and civilian. This leads the UAVs to face up with different and more insidious environments of work, so, in the last years, the navigation system of drones has continuously improved with new kinds of sensors and technologies. One of the most challenging way is about the integration of optical technologies in order to equipped the UAV with the vision. The idea to compute a quantity called optical flow vector, that represents the relative motion of the environment around the observer, and integrate it in a position and velocity estimation process, could improve and make robust the pre-existing navigation system, allowing the autonomous flight of the UAV in cluttered or indoor environment.

The purpose of this thesis is to integrate an optical flow sensor, called Px4Flow, on board of an UAV, already equipped with an inertial measurement unit and a GPS receiver, and to implement a Kalman filter, able to manage this integration, that gives a reliable estimation of the position and the velocity of the drone. The first goal to achieve is to identify the output of the sensor by flight test activity and to find a relation between the optical flow and the UAV velocity that we want to estimate. The found relation must be integrated in the mathematical model. For the estimation, in detail, a Kalman filter based on GPS and optical flow measurements has been implemented. The most important requirement, in order to reach the purpose of the thesis, is to implement a Kalman filter able to manage the multi-rate problem given by the different sampling frequency of the sensors. The thesis presents also a supplementary implementation of the H_∞ filter in a suitable form for the already implemented architecture of the Kalman filter. The aim of this addition, is to give the possibility to change the performance of the filter, changing the tuning of the measurement covariance matrix.

The finalization of this work consists of applying the implemented filters on real data sets collected during indoor and outdoor flights, and evaluate the estimation results. What we want to state is: if obtaining reliable estimates of position and velocity in outdoor and indoor environment using the optical flow measurements is possible; if the GPS and optical flow measurements can be fused together in a Kalman filter, making the estimate more robust.

Sommario

Al giorno d'oggi, gli UAVs (Unmanned Aerial Vehicls), comunemente noti col nome di droni, rappresentano un grosso ambito di ricerca e sviluppo, grazie alla versatilità che, questi velivoli, hanno dimostrato di avere in applicazioni militari e civili. Questo ha portato i droni ad essere applicati in diverse condizioni ambientali sempre più insidiose, e di conseguenza a cercare di migliorare il loro sistema di navigazione con sempre più nuove tecnologie e sensori. Una delle idee maggiormente stimolanti è quella di introdurre a bordo tecnologie di tipo ottico, in modo tale da donare la "vista" al velivolo. L'idea di calcolare una nuova grandezza, chiamata vettore optical flow, e di integrarla in un processo di stima di posizione e velocità, potrebbe migliorare e rendere più robusto il preesistente sistema di navigazione, permettendo, così, il volo autonomo del drone in ambienti congestionati o al chiuso. L'obiettivo di questa tesi è di utilizzare un sensore, chiamato Px4Flow, atto alla misurazione dell'optical flow, ed integrarlo a bordo di un drone già munito di una IMU (inertial measurement unit) e di un'antenna GPS; quindi implementare un filtro di Kalman, in grado di gestire questa integrazione, che ci restituisca una stima affidabile di posizione e velocità del drone. Il primo obiettivo è identificare l'output del suddetto sensore ottico, eseguendo un'attività di test di volo e trovare quindi una relazione tra il vettore optical flow e la velocità che vogliamo stimare. Una volta trovata tale relazione si procederà ad integrarla all'interno del modello matematico su cui verrà utilizzato il filtro di Kalman. Il requisito fondamentale, è implementare un filtro di Kalman che utilizzi le misurazioni del GPS e di optical flow e che sia in grado di gestire il problema di "multi-rate" dovuto alle diverse frequenze di campionamento dei vari sensori. Il lavoro di tesi prevede anche la supplementare implementazione di un filtro H_∞ in una forma adattabile all'architettura del suddetto filtro di Kalman, con l'obiettivo di fornire la possibilità di far variare le performance del filtro, agendo sulla matrice di covarianza delle misurazioni. Infine il lavoro è stato finalizzato, applicando il filtro implementato sui dati reali raccolti durante una campagna di test di volo interni ed esterni, e valutandone quindi i risultati. Le conclusioni a cui vogliamo giungere sono: determinare se è possibile ottenere stime affidabili di posizione e velocità, utilizzando le misurazioni di optical flow, in ambienti interni ed esterni; determinare se il GPS e l'optical flow possono essere utilizzati insieme all'interno di un filtro di Kalman, rendendo la stima più robusta.

Contents

Acknowledgments **I**

Abstract **III**

Sommario **V**

List of figures **XI**

List of tables **XV**

Introduction **1**

1 Problem formulation **3**
1.1 Navigation system definition . . . 3
1.2 Reference frames . . . 4
1.2.1 Geodetic spherical frame . . . 4
1.2.2 Navigation frame (Earth, NED) . . . 5
1.2.3 Body frame . . . 5
1.3 Rotation formalism . . . 6
1.3.1 Euler angles . . . 6
1.3.2 Quaternions . . . 8
1.4 Different navigation system configurations . . . 10
1.4.1 Inertial navigation system . . . 10
1.4.1.1 Advantages . . . 10
1.4.1.2 Disadvantages . . . 11
1.4.2 Inertial and satellite radio navigation system . . . 11
1.4.2.1 Work principles . . . 11
1.4.2.2 Sources of errors . . . 12
1.4.2.3 Advantages . . . 13
1.4.2.4 Disadvantages . . . 14
1.4.3 Inertial, satellite radio and optical navigation system . . . 14
1.4.3.1 Advantages . . . 14
1.4.3.2 Disadvantages . . . 15

2 Optical flow **17**
2.1 Definition of optical flow . . . 17
2.2 Computation of the optical flow . . . 18
2.3 Integration of the optical flow in the navigation system . . . 19
2.3.1 Angular compensation . . . 19
2.3.2 Ground altitude scaling . . . 21
2.3.3 Integration of the optic flow in the navigation system . . . 21
2.4 Optical flow equations . . . 22

3 Filtering estimation theory **27**
3.1 Estimation theory . . . 27
3.1.1 General overview . . . 27
3.2 Filtering theory . . . 29
3.2.1 Discrete-time Kalman filter . . . 30
3.2.2 $\mathbf{H}_\infty$ filter . . . 32

4 Position and velocity filtering **37**
4.1 Mathematical model . . . 37
4.2 Implemented Kalman filter . . . 41
4.2.1 Data pre-processing . . . 41
4.2.2 Initialization . . . 41
4.2.3 Tuning . . . 42
4.2.4 Multi-rate management: sequential update . . . 44
4.3 Implemented $\mathbf{H}_\infty$ filter . . . 46

5 Experimental set-up **47**
5.1 Hardware set-up . . . 47
5.1.1 Optical flow sensor: Px4Flow . . . 48
5.1.1.1 Technical properties: . . . 48
5.1.1.2 Dimensions and connectors scheme: . . . 49
5.1.2 On board integration . . . 49
5.1.2.1 Design and realization of the platform . . . 50
5.2 Pre-test activities . . . 51

6 Experimental results **55**
6.1 Covariance characterisation . . . 55
6.1.1 Procedure description . . . 55
6.1.2 Test results . . . 56
6.2 Estimate results . . . 58
6.2.1 Flight tests description . . . 59
6.2.1.1 Outdoor flight . . . 59
6.2.1.2 Indoor flight . . . 60
6.2.2 GPS based Kalman filter outdoor . . . 61

6.2.3 Optical flow based Kalman filter outdoor 67
6.2.4 GPS and optical flow based Kalman filter outdoor 71
6.2.5 Results comparison: outdoor flight 79
6.2.6 Optical flow based Kalman filter indoor 88
6.2.7 GPS and optical flow based H_∞ filter outdoor 92

Conclusions **95**

List of Figures

1.1 The WGS 84 reference frame [1] 5
1.2 Rotation singularity 8
1.3 Trilateration of satellites 12
1.4 Multipath signals 13

2.1 Optical flow (flying forward) [2] 17
2.2 Optical flow (flying to the left) [2] 18
2.3 Insect optical flow 19
2.4 Block matching by SAD 20
2.5 Multi-copter coupling between translational and rotational motion 20
2.6 Apparent motion due to rotation 21
2.7 Scheme for the on board integration of the optical flow 22
2.8 Optical flow and velocity relation 23
2.9 Flow sense 24

3.1 Estimation approaches 29
3.2 Example of P trend 32

4.1 Measurements availability check 45

5.1 Px4Flow sensor 48
5.2 Connectors and dimensions 49
5.3 Design of the platform 50
5.4 Px4Flow assembled on the UAV 51
5.5 Px4Flow orientation 51
5.6 Lens calibration 53

6.1 PSD analysis for accelerometers 56
6.2 PSD analysis for optical flow 57
6.3 PSD analysis for GPS position 58
6.4 Outdoor test 59
6.5 Px4Flow quality parameter: outdoor flight 60
6.6 Indoor test 60
6.7 Px4Flow quality parameter: indoor flight 61

6.8 Comparison: estimated x-position and GPS x-position 62
6.9 Comparison: estimated y-position and GPS y-position 62
6.10 Comparison: estimated z-position and GPS z-position 63
6.11 Comparison: estimated x-velocity and GPS x-velocity 64
6.12 Comparison: estimated y-velocity and GPS y-velocity 64
6.13 Comparison: estimated z-velocity and GPS z-velocity 65
6.14 State estimate innovations 65
6.15 Comparison: Innovations PSD and GPS measurements PSD . . . 66
6.16 Innovations PSD at high frequency 66
6.17 Kalman gains due to GPS measurements 67
6.18 Comparison: estimated x-velocity and GPS x-velocity 68
6.19 Comparison: estimated y-velocity and GPS y-velocity 69
6.20 State estimate innovations 69
6.21 Comparison: Innovations PSD and optic flow measurements PSD 70
6.22 Innovations PSD at high frequency 70
6.23 Kalman gains due to optical flow measurements 71
6.24 Comparison: estimated x-position and GPS x-position 72
6.25 Comparison: estimated y-position and GPS y-position 72
6.26 Comparison: estimated z-position and GPS z-position 73
6.27 Comparison: estimated $x$velocity and GPS x-velocity 73
6.28 Comparison: estimated y-velocity and GPS y-velocity 74
6.29 Comparison: estimated z-velocity and Gps z-velocity 74
6.30 Position state estimate innovations 75
6.31 Velocity state estimate innovations 75
6.32 Comparison: Innovations PSD and GPS measurements PSD . . . 76
6.33 Comparison: Innovations PSD and optic flow measurements PSD 76
6.34 Position innovations PSD at high frequency 77
6.35 Velocity innovations PSD at high frequency 77
6.36 Kalman gains due to GPS measurements 78
6.37 Kalman gains due to optic flow measurements 78
6.38 Compared results: position in x 79
6.39 Compared results (zoom): position in x 80
6.40 Position-x error between the estimate by using GPS + optical flow and GPS only 80
6.41 Compared results: position in y 81
6.42 Compared results (zoom): position in y 82
6.43 Position-y error between the estimate by using GPS + optical flow and GPS only 82
6.44 Compared results: position in z 83
6.45 Compared results (zoom): position in z 83
6.46 Position-z error between the estimate by using GPS + optical flow and GPS only 84
6.47 Compared results: velocity in x 84

6.48 Velocity-x error between the estimate by using GPS + optical flow and optical flow only 85
6.49 Compared results: velocity in y 85
6.50 Velocity-y error between the estimate by using GPS + optical flow and optical flow only 86
6.51 Innovations PSD comparison for position 87
6.52 Innovations PSD comparison for velocity 87
6.53 Kalman gain comparison for position in x 88
6.54 Comparison: estimated x-velocity and Mocap x-velocity 89
6.55 Comparison: estimated y-velocity and Mocap y-velocity 89
6.56 Velocity state estimate innovations 90
6.57 Velocity state estimate innovations without $\pm$ 3std boundaries . . 90
6.58 Comparison: Innovations PSD and optical flow measurements PSD 91
6.59 Velocity innovations PSD at high frequency 91
6.60 Estimate x position: H_∞ with different thetas 92
6.61 Estimate y position: H_∞ with different thetas 93
6.62 Estimate z position: H_∞ with different thetas 94

List of Tables

6.1 Accelerometers variance values in $(m/s^2)/\sqrt{Hz}$ 57
6.2 Optical flow variance values in $rad/\sqrt{Hz}$ 57
6.3 Gps variance values $m/\sqrt{Hz}$ 58

Introduction

UAV (Unmanned Aerial Vehicle), usually called drone, is the definition for an aircraft without a pilot aboard. This kind of vehicle was introduced, before the WWII, as moving target used to train the gun-shot of military ships and aircraft of USA military forces.
At the beginning the drones were remotely piloted, nowadays, with the improvement of technology, they turn into vehicles able to autonomously follow a flight trajectory and this allows them to increase the their range of applications. Thanks to this ability to execute autonomous flight and to the improved performances, nowadays, there are a lot of activities which can benefit from the UAVs:

- **Military and Security Field:** terrorism fight, ISR (Intelligence, Surveillance and Reconnaissance), DDD Roles (Dangerous, Dirty, Dull) , National Security, Radar Jamming, Ground Attack, Missile Defense, Search and Rescue;
- **Civil and Scientific Field:** agriculture, hobby, 3D mapping, surveillance, weather monitoring, journalism and photography, products delivery and logistic, scientific research.

What a drone needs to execute an autonomous flight, is a Flight Control Unit (FCU) that allows to correct the behaviour of the vehicle in order to reach the desired position, velocity, attitude, etc. (talking about multi-copter UAVs , due to their very fast dynamics, the FCU becomes necessary also for remotely piloted flight). To correct the behaviour of the drone the FCU needs to have information about the state of the drone (position, velocity, acceleration, attitude, etc.) during the whole flight. What makes this possible is a navigation system that uses the integration of different sensors to reach the complete information about the flight: accelerometer to define the attitude and to know the accelerations, gyroscopes for a direct measurements of angular rate, barometer for the altitude, and all the others sensors that could complete the information. All the measurements of these sensor are fused in a state estimation process.

What really changes the navigation system and allows the wide growth of UAVs in the civil and military field, is the development of GPS technology, giving us a direct position measurement available all around the world.

Some of the UAV applications presented above underline the need to have a good operative UAV in cluttered environment like urban and indoor environment, where the GPS loses its efficiency, thus the navigation system cannot rely position and velocity measurements; in this scenario the vehicle does not have the necessary elements for the state estimation to execute an autonomous flight and from this reason, a new idea was born : to provide a completely independent sensor based on the optical technology, that becomes the eye of the drone. The idea is to compute a new vector called optical flow, that describes the apparent motion of the environment with respect to the UAV, and to use this measurement to obtain the velocity of the drone. Nowadays this optical technology begins to be enveloped and different sensors exist which can compute the optical flow vector. The aim of this thesis work is to obtain a navigation system that includes the optical flow sensor, and based on a filter able to fuse the GPS and optical flow measurements in order to obtain a reliable estimate of UAV position and velocity. The application of this filter on real data sets can gives answers about the possibility to rely on this estimation to navigate during outdoor and indoor flights.

To reach this purpose the optical flow sensor, named Px4Flow, must be integrated on board and used to supply optical flow measurements for the state estimation. The implementation of a Kalman filter is needed to reach reliable estimate results, managing the multi-rate sampling frequencies of the sensors. The flight test campaign, consisting of outdoor and indoor flight tests, supplies the logged data sets useful for the application of the implemented Kalman filter.

The structure of this thesis is presented in this order:

- formulation of the navigation problem and the description of the different navigation systems;
- definition of the optical flow and identification of the relation between the optical flow sensor output and the UAV velocity;
- presentation of the estimation theory with focus on the discrete-time Kalman filter theory and definition of a form of the H_∞ filter suitable with the Kalman filter architecture;
- description of the implemented mathematical model, Kalman filter and H_∞ filter;
- description of the experimental set-up including hardware integration of the optical flow sensor on the UAV, and its lens calibration;
- presentation of experimental results.

Chapter 1

Problem formulation

1.1 Navigation system definition

Navigation is an ancient discipline that consists in the ability to know and monitor the position of a craft or a vehicle moving from a place to another. In the aeronautical world it is essential since the first decades of the twentieth century and it have had a continuous growth correlated to the improvements reached in the field of sensors. The earliest aircraft were navigated visually for most of the time, so the two main aims of navigation, knowledge of position and velocity of the aircraft and the knowledge of its attitude, were achieved in that way or by "data" from very simple but efficient sensors as a barometer for the pressure altitude, anemometer for the airspeed, magnetic compass for the heading, artificial horizons and turn-and-bank indicators to help the aviator to "estimate" the attitude especially when the view is poor because of clouds or other natural or artificial reasons. This pack of primitive sensors allows the pilot to know the state of the vehicle, especially in the short period; for the long period navigation he had not other advices in addition to the compass, the maps and some natural or artificial landmarks on the ground. One of the first techniques developed to improve the estimation of the position during flight, is "dead-reckoning" computation. The idea of this method is to know, with the highest precision possible, the value of the speed of the vehicle (without the wind speed contribute) and its true heading, to compute the North and the East components of the velocity; from this, the computation of the position by mathematical integration is possible. This method gave, of course, before the advent of other technology as the GPS, the best estimation between two fixed point. The cumulative errors given by the integration affect the estimate proportionally to the length of the journey and for this reason step by step, different advices were introduced to have more fixed position information during a flight (think to the large envelope of radio-navigation systems during and after the WWII; for more informations [3].

The field of this thesis work is the navigation of UAVs that since their first debuts during the two world wars, need a navigation system especially for the

absence of an on-board pilot. Focusing on this category of vehicles we can talk about three different combinations of sensors and methods representing a navigation systems :

- inertial navigation;
- inertial plus satellite-radio navigation;
- inertial plus satellite-radio navigation plus vision systems.

Before to explain the pros and cons of these navigation systems, it is better to describe in this chapter some concepts and formalism useful to understand the following discussions.

1.2 Reference frames

1.2.1 Geodetic spherical frame

Nowadays the model used for the Geodetic spherical frame is the WGS 84. The WGS 84 coordinate system is a right-handed, Earth-fixed orthogonal frame, based on an ellipsoid formulated in the 1984, that approximates very well the surface of the Earth. According to the definition given by the U.S Defense Mapping Agency [1], it is possible to state what is described in Figure 1.1. The origin is placed on the Earth's center of mass; using the definitions given by the International Earth Rotation and Reference Systems Service (IERS), the $\hat{z}$ axis has the direction of the IERS Reference Pole (IRP); the $\hat{x}$ axis is generated by the intersection of the IERS Reference Meridian (IRM) and the plane passing through the origin and normal to the $\hat{z}$. The $\hat{y}$ axis completes a right-handed orthogonal coordinate system. The identification of the position of a vehicle in this reference frame is done through three quantities:

- longitude $\longrightarrow$ the angular distance with respect to the Greenwich Meridian, that is the meridian of reference;
- latitude $\longrightarrow$ the angular distance with respect to the Equatorial plane;
- altitude $\longrightarrow$ the distance from the origin of the system to the vehicle.

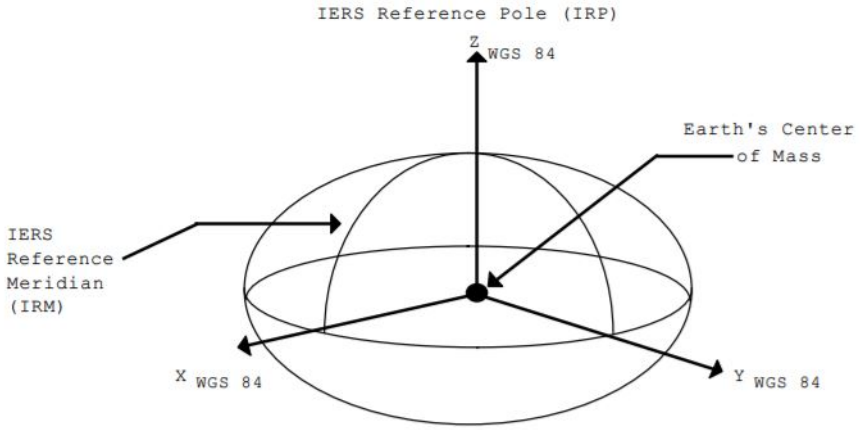


Figure 1.1: The WGS 84 reference frame [1]

1.2.2 Navigation frame (Earth, NED)

Known also as "Earth frame" , it is a right-handed orthogonal frame with the origin placed on a generic point on the Earth's surface, for example the place of take-off. The first and second axes $\hat{x}_n$, $\hat{y}_n$ define the horizontal plane tangent to the surface of earth for altitude equal to zero and parallel to it for altitude higher than zero. The $\hat{x}_n$ axis points to the North and the $\hat{y}_n$ axis points to the East; the third axis , $\hat{z}_n$, has the direction of the gravity, pointing downwards. For these features it is often referred as NED (North-East-Down). (For more information [4])

1.2.3 Body frame

This right-handed orthogonal frame has the origin on the center of gravity of the aircraft. The first axis, $\hat{x}_b$ is defined parallel or coincident to the longitudinal axis of the aircraft body, pointing to the front. The $\hat{z}_b$ points downwards lying on the longitudinal plane while the $\hat{y}_b$ points to the right creating a right-handed orthogonal system. The main property of this coordinate system is its being attached to the vehicle, changing its orientation with it. This make it very useful for the writing of the dynamics equation thanks to its constant inertial characteristics. (For more information [4])

1.3 Rotation formalism

1.3.1 Euler angles

All the reference frames described in the previous section, could be involved in a navigation system, especially because of the different nature of the data set coming from the sensors. For this reason, there is the need to switch from one of these Cartesian coordinate systems to another, and this is based on the definition of three independent parameters that allow us to describe the relative orientation between two different reference frames. The Euler angles (ϕ, θ, ψ) are three independent angular parameters that can reach this aim.

Adopting the to-from notation, a rotation matrix from system E to system D (considering the two systems with the origins in the same place with differently oriented axes) might be named R_E^D. Thus, a vector v_E in system E can be resolved to system D, that is v_D through the matrix operation:

$$v_D = R_E^D v_E \tag{1.1}$$

The matching between the two set of axes is obtained by a sequence of three rotations (one per axis). In light of this it is easy to understand how these rotation matrices work: the rotation about the X axis does not change the component of the vector directed along the X axis, but it rotates the Y and Z components. The rotation matrix that does this transformation is

$$R_X(\Phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}. \tag{1.2}$$

In a similar fashion, fixing a different axis each time we obtain the other two rotation matrices:

$$R_Y(\Theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \tag{1.3}$$

$$R_Z(\Psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1.4}$$

The concept of rotation of a coordinate system with respect to another needs the definition of a triad of axes as the non-rotating one; it will represent the starting point for the rotations of the other triads. One of the most important properties of this approach is the orthonormality of the rotation matrices, which

means orthogonality and unit magnitude of the columns of matrices seen as vectors. Mathematically this is expressed as:

$$\begin{aligned} R_X^{-1}(\phi) &= R_X^T(\phi), \\ R_Y^{-1}(\theta) &= R_Y^T(\theta), \\ R_Z^{-1}(\psi) &= R_Z^T(\psi). \end{aligned} \tag{1.5}$$

In this way we are able to manage any number of rotations.
The resulting cascade can be reduced to a rotation about just three axes. The matrix that performs this specific action is called an Euler rotation matrix and has the following definition:

$$T_E^B(\phi, \theta, \psi) = R_X(\phi) R_Y(\theta) R_Z(\psi)\,. \tag{1.6}$$

The subscripts B and E stand for "Body" and "Earth", respectively. The matrix T_E^B resolves an Earth-based vector to body axes:

$$T_E^B(\phi, \theta, \psi) = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ S_\phi S_\theta C_\psi - C_\phi S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & S_\phi C_\theta \\ C_\phi S_\theta C_\psi - S_\phi S_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi & C_\phi C_\theta \end{bmatrix}, \tag{1.7}$$

where a shorthand notation, which is $C_a = \cos(a)$ and $S_a = \sin(a)$, has been adopted.
The following is the transformation related to velocity vectors mathematically explained:

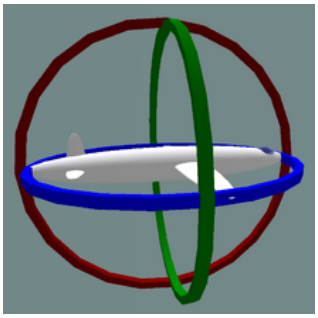
$$r_e = \begin{bmatrix} N \\ E \\ D \end{bmatrix}, v_e = \begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \end{bmatrix} = \dot{r_e}, \tag{1.8}$$

$$v_b = T_E^B(\phi, \theta, \psi) v_e = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{1.9}$$

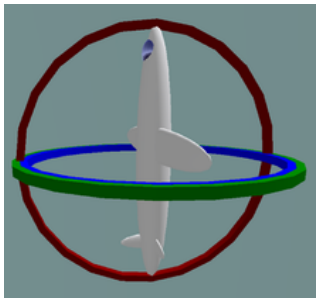
where r_e is the position vector of the aircraft center of gravity in inertial (Earth) axes, v_e is the velocity of the aircraft with respect to the Earth, and v_b is the inertial linear velocity of the aircraft, resolved to body axes.

We can also define the vector of angular position of the aircraft body frame with respect to the Earth, resolved to the Earth where the elements are the respectively roll, pitch and yaw angles.

$$\alpha_e = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \tag{1.10}$$



(a) No gimbal lock



(b) Gimbal lock

Figure 1.2: Rotation singularity

1.3.2 Quaternions

The three equations coming from the Euler rotation matrix represent a convenient tool to manage the rotation between different triads of axes but the system they compose has singularity issues. This mathematical singularity known as "gimbal lock", takes its name from the correlated physical phenomena observable in the old mechanical gyroscopes which use three gimbals (one for each axis) to give information about the attitude. Using this kind of sensor, when the vehicle executes a vertical flight, the pitch gimbal and the yaw one become aligned; this lead to a situation where it is impossible to distinguish between a rotation in roll or in yaw, as we can see in Figure 1.2.

It is well known that no global (singularity-free) three-dimensional parametrisation exist for rigid body attitude, but it is possible to formulate a four dimensional, singularity-free parametrizations to apply to this problem. Considering the latter type, the most common and widely used formulation is the quaternion. It can be defined as follows:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}. \tag{1.11}$$

It can be explained in terms of Euler axes (e) and angle (ϑ):

$$q = \begin{bmatrix} \rho \\ q_4 \end{bmatrix}, \tag{1.12}$$

where

$$\rho = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T = e \sin \frac{\vartheta}{2}, \tag{1.13}$$

$$q_4 = \cos\frac{\vartheta}{2}\,. \tag{1.14}$$

In order to manage the rotations between different coordinate systems, using the quaternion formulation, an attitude matrix is defines as:

$$A(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}. \tag{1.15}$$

For a better knowledge of this parametrization it is better to specify some useful properties. The q has a unit-norm constraint expressed as:

$$q^T q = 1, \tag{1.16}$$

in this way , from a geometric point of view, the set of all admissible quaternions spans the unit-sphere in the four dimensional Euclidean space R^4.

The unit quaternion is given by $[0\ 0\ 0\ 1]^T$ that means $A = I_3$. While the inverse of a quaternion is given by:

$$q^{-1} = \begin{bmatrix} -\rho \\ q_4 \end{bmatrix}. \tag{1.17}$$

Different quaternions represent different attitude states, thus different attitude matrices represent different rotations. It could be necessary to concatenate products by these matrices. The product and quotient operation for two quaternions q' and q'':

$$\begin{gathered} A(q'')A(q') \Leftrightarrow q'' \otimes q' \\ A(q'')A(q')^T \Leftrightarrow q'' \otimes q'^{-1} \end{gathered} \tag{1.18}$$

with the quaternion product operator :

$$q'' \otimes q' = \begin{bmatrix} q_4'' & q_3'' & -q_2'' & q_1'' \\ -q_3'' & q_4'' & q_1'' & q_2'' \\ q_2'' & -q_1'' & q_4'' & q_3'' \\ -q_1'' & q_2'' & -q_3'' & q_4'' \end{bmatrix} q'. \tag{1.19}$$

For the application of this thesis work this discussion about the quaternion formulation is sufficient. For more details, especially about the time evolution so the time derivative, it is suggested to see on [4].

1.4 Different navigation system configurations

In the following paragraphs the three types of navigation systems introduced above are discussed to focus on the motivation of this thesis work and how an optical sensor could theoretically be a good choice to improve the pre-existent systems for UAVs.

1.4.1 Inertial navigation system

The inertial navigation system (INS) represents one of the earliest examples of multi-sensors configuration. It evolved from the fire control technology implemented for the guidance of missiles since the years before WWII, and from the marine gyrocompass (a sensor that uses the efficiency of gyroscopes in the short period and the reliability of compass measurement in long range missions). At the beginning this kind of system had much fortune in ships , as well as in missiles, especially because of its size and weight, but with the improvement of such technology the INS started to appear also on aircraft. Its operative principle is to recover information of position, velocity and attitude starting from measurements supplied by accelerometers, gyroscopes and, sometimes, a compass. For the first time in history a computer on board became necessary (analog and then digital) able to fuse all these different data and give back the desired information. What the computer did on data, basically was to take the measured acceleration and , like in the dead-reckoning technique, integrate it in time twice, obtaining first the velocity and then the position, while the information about the attitude was recovered from gyroscopes, which measure the angular velocity, and from the compass for the heading. This process must be done on all the three axis of the chosen reference system. At the beginning this was possible thanks to the mechanical gimbals which isolate the sensors from the rotations of the vehicle, in order to fix accelerometers and gyroscopes on a unique orientation, today thanks to the technological improvements it is possible to benefit from the strap-down inertial system. This modern sensor platform is fixed to the aircraft and this lead to have measurements with respect to the body frame (giving us the possibility to use the projection of the gravity acceleration, felt by the accelerometer, on the three axis, in order to obtain the attitude of the body frame with respect to the navigation one). The actual strap-down platform can be very light in weight and power consumption, precise and cheap, thus ideal for the navigation system of a UAV; it is appropriate to analyse the pros and cons of the pure inertial navigation system.

1.4.1.1 Advantages

- High data rates and high bandwidths;

- completely independent from the external world, so non-jammable;
- its data are reliable at all latitudes of the Earth (including the polar regions);
- the most accurate advice for the measuring of the azimuth.

1.4.1.2 Disadvantages

- Due to the cumulative errors given by the integration steps, the position and velocity information degrades with time;
- the accuracy of the navigation information could be affected by the dynamics of the vehicle.

1.4.2 Inertial and satellite radio navigation system

The main disadvantage of the INS is the degradation of the estimation of position and velocity with time. Of course this represents a huge problem for long range missions, so, during the decades, the technological progress tried to fix it implementing the radio navigation system. It means the creation of a ground segment able to transmit, through radio signals, information of position and velocity to the vehicle, reducing the time interval in which the navigation must rely only on the inertial platform. During the second half of the 20^{th} century, great improvements have been made in this direction, but there was the need to implement something able to supply measurements of position and velocity all over the world. The answer has been the use of satellites. The United States of America created the first satellite navigation system, the NAVSTAR Global Positioning System, better known as the GPS (nowadays others systems and constellations exist, co-working for the GNSS, Global Navigation Satellite System).

1.4.2.1 Work principles

Focusing on its working principles, the satellite radio navigation system is basically composed by a space vector (satellites), ground vector (ground control station for the manage and the upload), a user vector represented by a receiver antenna. The latter receives the signal and computes the distance from the satellite, knowing the velocity of the signal and the time interval used to reach the target. The precise identification of the position is based on the concept of trilateration: the computed distance from the satellite represents the radius of a sphere, so it can not define a single point, but the intersection of three spheres allow to find the position of the vehicle, as seen in Figure 1.3.

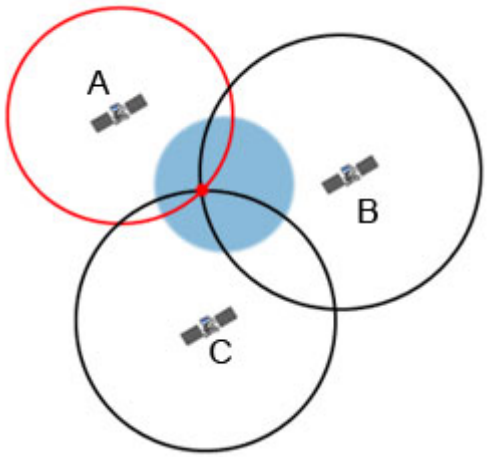


Figure 1.3: Trilateration of satellites

The process described above is based on the assumption that the epoch times of the satellites and the receiver are known and synchronized; but the clock of the user cannot have the same accuracy of the atomic ones integrated on the satellites. For this reason there is the need to continuously correct this time information of the receiver, and this means to add another unknown variable to the system. It is clear that a fourth satellite is necessary to execute this time correction. In light of this we can state that the space segment must supply at least four satellites to the receiver.

This is acceptable in the ideal case where the intersection of three spheres gives back a unique point. This is not true in reality where this intersection has as result a small region around the exact solution point. To reduce this uncertainty region more spheres must be intersected, thus other satellites need to be available in the observable sky by the receiver. Focusing on the Gps's constellation, it consists of 24 operative satellites distributed on six orbits, so four on each orbit, equally spaced (60° apart). This large space segment allows to have always a sufficient number of satellites to reach a position and velocity measurement. The choice of visible satellites is crucial to reduce the uncertainty and the criteria used to chose, by the receiver, is their spread in the sky. Operatively it is translated into a numeric parameter named DOP (Dilution of Precision) that suggests which satellites are better to chose. See [3].

1.4.2.2 Sources of errors

- Ionospheric refraction, due to the high ionization, thus large presence of ions and free floating electrons, of the atmosphere in its outermost layer;

- tropospheric refraction, in the innermost layer of the atmosphere where the satellite signal is affected in function of temperature, pressure and humidity;
- multi-path, that is an error given by the reflection of signals on the environment around the receiver as easily represented in Figure 1.4;
- receiver noise;
- time error;
- typology of the used signal. Depending of what user we are, we are authorized to receive a particular signal.

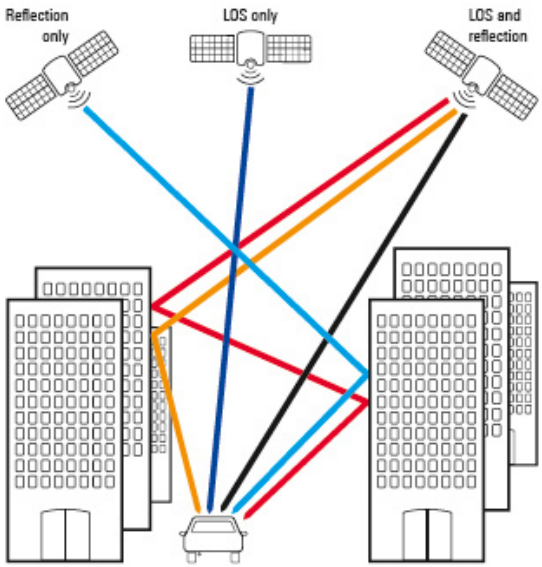


Figure 1.4: Multipath signals

With the modernization of the system and the implementation of new techniques, some of these errors have been reduced or completely compensated, but the source of error for this navigation system are numerous. For a complete discussion on the errors and correlated compensation, read on [3]

Below, pros and cons of this navigation system are summarized.

1.4.2.3 Advantages

- The use of the GPS gives a direct measurement of position and velocity, fixing the dead-reckoning of the inertial system, and for a UAV this means the possibility of outdoor automatic and autonomous flight;

- it adds an accurate information of time;
- a space segment for the transmitters allows us to have this information in every geographic region on the Earth.

1.4.2.4 Disadvantages

- slow rate of information, especially if it is compared with the inertial sensors;
- possibility to be jammed;
- not reliable indoor or in very clustered environments;
- actually the most reliable and efficient satellites system is the GPS and it is a property of the United States Department of Defense, which for any reason can decide to put it out of service or to degrade the signal.

1.4.3 Inertial, satellite radio and optical navigation system

The disadvantages shown by the satellite navigation system become worse if we focus on the field of application of this thesis work, UAV navigation. For a drone it is more probable to execute missions indoor or in a cluttered environment such as the urban one, and if the aim is to reach autonomous flight for this kind of vehicle, it is necessary to think about a new generation of sensors that can fix this issue. Thanks to the large improvements in visual technologies and image processing, the idea to use the vision to accomplish this task was born. The integration of the inertial-satellite navigation system with an optical sensor able to give position and velocity information in scenarios where usually the Gps loses its efficiency, could increase the field of application of UAVs. The other important aspect is the possibility to have a completely independent navigation system, without any kinds of external infrastructure, and always reliable. Nowadays this new kind of optical sensor has become popular and basically what the research has done is to give to the drone the ability to use a visual course guidance navigation as the first pilots. The idea is to measure how the vehicle is moving, observing how the environment has moved with respect to it. The leading actor of this navigation system is the "optical flow"; the quantity that represents the apparent motion of the scenario around the UAV. The optical flow vector will be deeply discussed in the next chapter. In the following there is a brief summary of the pros and cons that we expect before to implement and test this navigation system.

1.4.3.1 Advantages

- Position and velocity information indoor and in cluttered environment;
- completely independent navigation system;

- data rate higher than data from GPS.

1.4.3.2 Disadvantages

- Degraded quality of measurements with bad visual condition (weather or brightness condition);
- altitude limitation;
- need of features to observe to feel the apparent motion of the scenario, so not completely smooth surface.

Chapter 2

Optical flow

2.1 Definition of optical flow

The concept of optical flow can be described as the apparent motion of what belongs to the environment all around the observer. When we move, the perception of visual world give us a continuous feedback of the motion allowing us to know the depth and the relative velocity of an image.[5]

With this information, our brain is able to estimate our position and velocity. The study of this phenomenon is not a modern activity, in fact the first person who focused his knowledge and efforts to give a scientific and psychological explanation to it, was James J. Gibson with his first work in 1950 : "The Perception of the Visual World". His idea was born right from the world of vehicles. During World War II , Gibson served in the USA Army Air Force, in the Aviation Psychology Program, where his aim was to understand how the pilots fly without instrumentation, simply based on what they see and how the scenario changes out of their cockpit. [2]

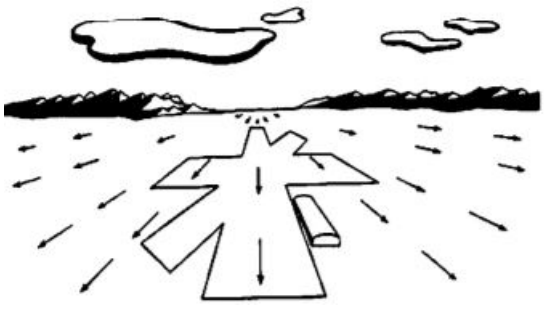


Figure 2.1: Optical flow (flying forward) [2]

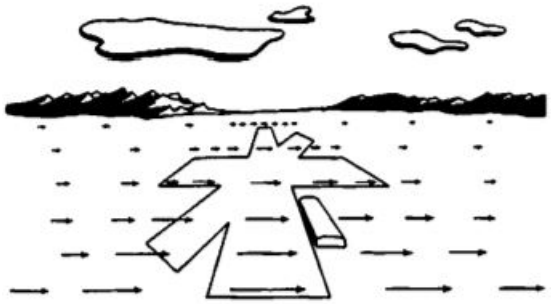


Figure 2.2: Optical flow (flying to the left) [2]

The Figure 2.1 and Figure 2.2 were drawn by Gibson, after he spent entire days observing and analysing dozens of flights, and they represent respectively the motion of the scenario while the aircraft goes forward (Figure 2.1) and the motion of the scenario while the aircraft goes to the left (Figure 2.2). These two drawings are a real simple and clear explanation of what optical flow means; here the scientist tried to represent the apparent motion of each portion of the scenario observed by the pilot, using this drawn arrows that show the direction and the sense of this field of motion, and also its magnitude: looking at the two pictures, we can immediately appreciate , thanks to the size of the arrows, that an image closer to the observer moves faster.

Another example of how much reliable could be the visual motion for navigation, comes from flying insects (Figure 2.3). They rely mainly on optical flow to navigate efficiently during their flight maneuvers such as: terrain following, tunnel crossing, adjusting their speed in very cluttered environment. [6]

2.2 Computation of the optical flow

The application of this principle to UAV navigation needs the definition of the optical flow as a field of vectors.

The necessary elements to do that are: a camera able to catch the apparent motion of the environment frame by frame and an algorithm to compute the optic flow vector. There are different techniques for this computation (classified as differential, matching, energy-based, phase based) but in this work we are not going to focus on their mathematical equations. It is useful to know just the basic idea used to obtain the optical flow vector by the sensor, chosen for this work, *i.e.*, the Px4Flow. In this case optical flow estimation is based on the Sum of Absolute Differences (SAD) block matching [7]. It consists in the observation of a search area of ± 4 pixels with respect to the point given by the intersection of the

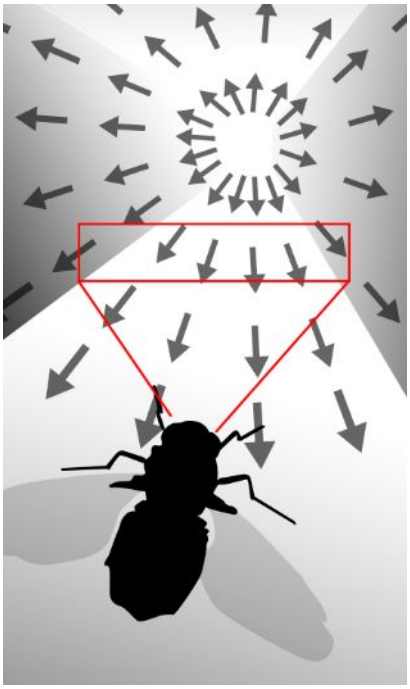


Figure 2.3: Insect optical flow

observed surface and the perpendicular line coming from the centre of the lens of the camera. The 8x8 pixels block is defined and compared frame by frame, looking at how much a single pixel has moved from an image to the next. The SAD is used to identify the same portion between the old frame and the new one (for a better understanding see Figure 2.4).

2.3 Integration of the optical flow in the navigation system

For our purposes the camera has been fixed in a looking downward position to use the features of the floor for the block matching algorithm. The aim of this work is to take the output optical flow vector computed into the sensor and use it as and indirect measurement of the translational velocity (x,y plane) of the UAV. It is necessary to understand its behaviour to execute the sensors integration in the right way.

2.3.1 Angular compensation

The optic flow data, supplied by the sensor, represent the relative motion of the environment around the drone and for this reason they don't take into account if

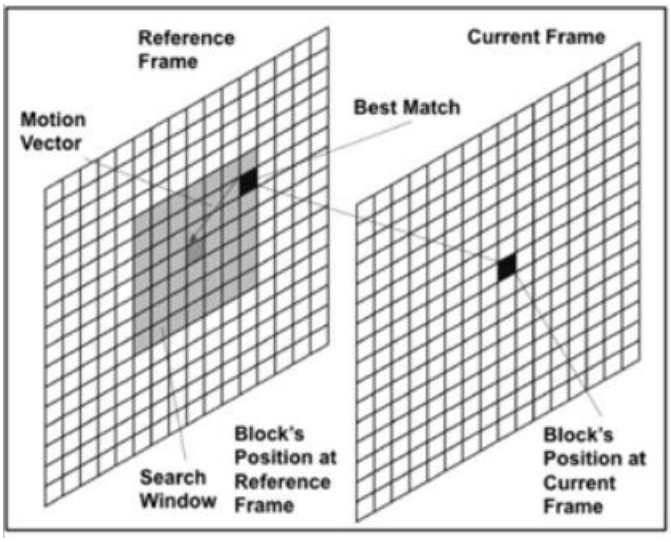


Figure 2.4: Block matching by SAD

this apparent motion is due to a translational or an angular velocity of the UAV. A multi-copter flies thanks to the thrust given by its propellers. When the drone is levelled this thrust is vertical and it is possible to stay in hover or to reach a chosen altitude. If we want to follow a different trajectory the multi-copter has to rotate and generate an horizontal component of the thrust, like in the scheme of Figure 2.5. For this reason, for the conventional configuration of a multi-copter UAV, is impossible to execute a translation without a rotation in pitch or roll (depending on the direction of the translation).

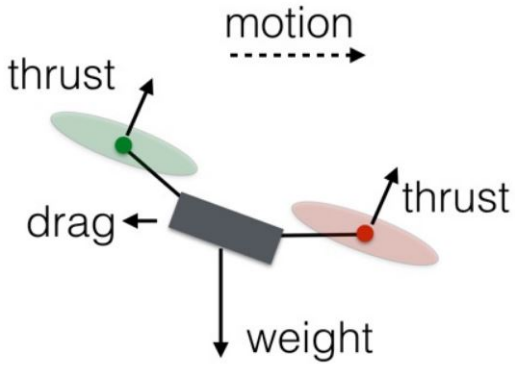


Figure 2.5: Multi-copter coupling between translational and rotational motion

For the camera a change in attitude is a motion of the image of the floor(Figure 2.6), as well as a change in longitudinal or lateral position, so it becomes necessary to post-process the optical flow data in order compensate the contribution of the rotation. To accomplish this task the attitude of the UAV must be known. In

particular the angular velocity about the $\hat{x}$ and $\hat{y}$ body axes can be measured using the set of gyroscopes that are implemented on board.

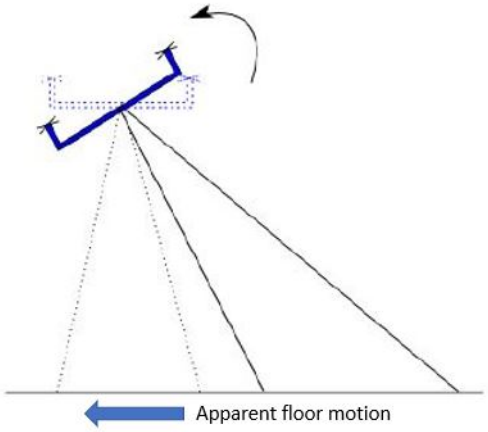


Figure 2.6: Apparent motion due to rotation

2.3.2 Ground altitude scaling

A moving object could appear more or less faster depending on the distance from the observer and this affect the optical flow computation. To understand how, it could be useful to think that the camera works the images placed on an imaginary sphere that have the centre equivalent to the position of the camera, and the radius equal to the ground distance. The same value of optic flow could be the result of a faster motion with higher ground distance , or a slower motion with a lower ground distance. This can completely void the estimation of the velocity, so the navigation system must have the measurement of the distance from the flow at each time instant. It is important to underline that what we need is something as a sonar or laser sensor that lets us to know the ground distance, not a barometer because it is necessary to feel every single vertical change of the observed floor surface.

2.3.3 Integration of the optic flow in the navigation system

The chosen scheme to integrate the optical flow with the other sensors, described in Figure 2.7, requires:

- optical flow sensor for the optical flow vector ($flow$); acc_m supplied by the accelerometers; gyroscopes for the angular velocities (w_m); a sonar sensor for the ground distance (h), GPS receiver to have a comparison with the

velocities estimated from the optical flow data or to implement a position and velocity estimation with the redundancy of the two sensors.

- the attitude inputs ($\hat{q}$) needed for the position ($\hat{p}$) and velocity ($\hat{v}$) estimation, supplied by an attitude estimator.

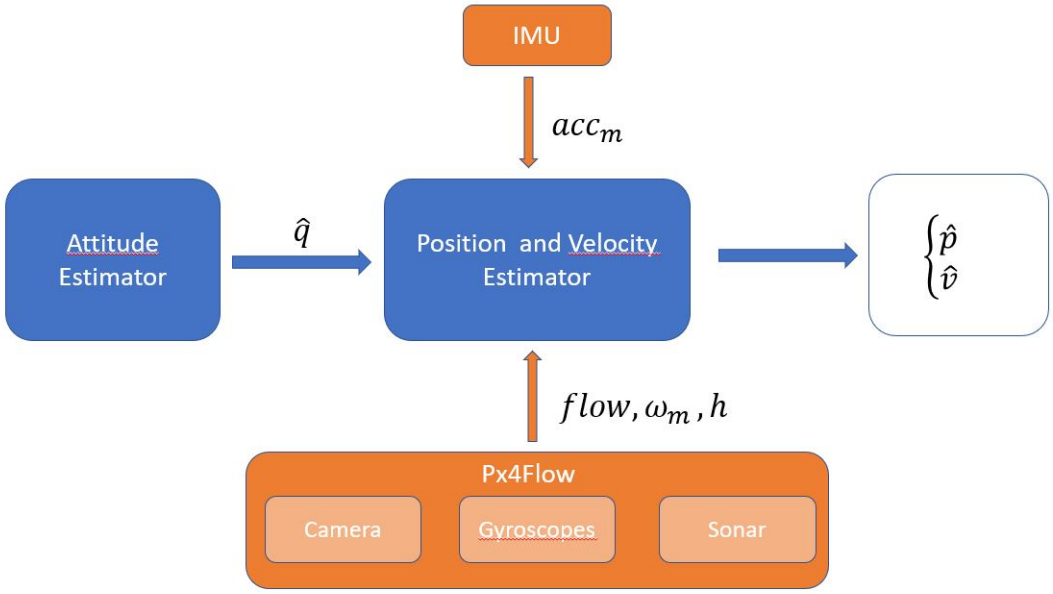


Figure 2.7: Scheme for the on board integration of the optical flow

2.4 Optical flow equations

What we need from the optical flow sensor is an output useful to indirectly estimate the velocity. The optical flow is definitely related to it but we don't know how and there is not a literature that can suggest some kind of equation. The procedure to identify the output of the sensor and its relation with the velocity of the UAVs, has been based on:

- Knowledge of the physics of the optical flow (explained in Chapter 1);
- knowledge of the firmware implemented in the sensor;
- comparison with the available velocity measurements coming from the GPS.

The optical flow could be seen as a rotation of the environment sphere around the observer. For this reason we can suppose it, dimensionally, as an angular velocity expressed in $[rad/s]$.

The velocity, that is the target, is expressed in $[m/s]$; so, according to the kinematic equation of a generic rotational motion as shown in Figure 2.8 , it is possible to write:

$$v_{floor} = \varepsilon\, h\,, \tag{2.1}$$

$$v = -v_{floor}\,, \tag{2.2}$$

with ε = optical flow vector; v = UAV velocity; v_{floor} = floor apparent velocity; h = distance between the floor and the camera.

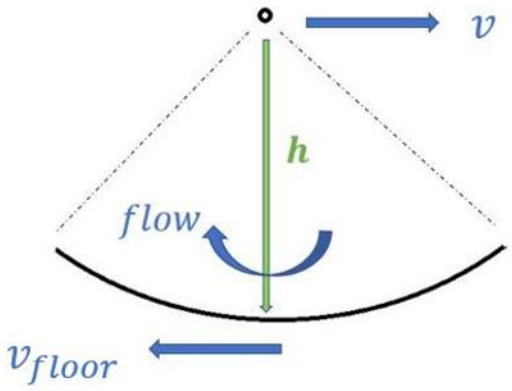
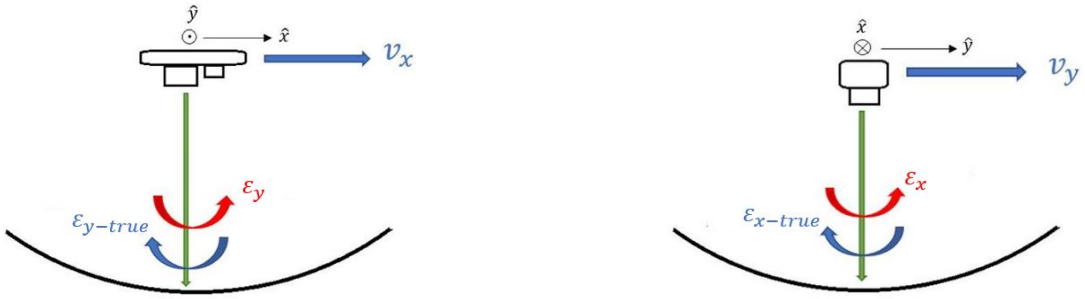


Figure 2.8: Optical flow and velocity relation

Taking into account the reference system orientation and both axes of interest:

$$\begin{cases} v_x = \varepsilon_y\, h & (2.3) \\ v_y = -\varepsilon_x\, h\,. & (2.4) \end{cases}$$

The different sign in the two equations above is due to the reference system as represented in Figure 2.9 (a) and Figure 2.9 (b) to the way used by the sensor to compute the optical flow. In Figure 2.9 (a) the drone is moving forward and this means, in UAV body reference frame, along the $\hat{x}$ axis, and this leads to an apparent translational velocity of the floor in the opposite direction ($-\hat{x}$). The floor could be approximated as a little portion of the circumference around the sensor that, rotating, generates the optical flow (the flowing images in front of the camera). If we consider the flow vector as an angular velocity around the $\hat{y}$ axis (ε_{y-true} in the figure), it will be negative in this case (its spin axis points in the opposite direction of $\hat{y}$); but the sensor is implemented to give, as an output, the optical flow vector with a change in sign (ε_y in the figure), to allow the direct



(a) Flow moving forward

(b) Flow moving to the right

Figure 2.9: Flow sense

recover of the UAV velocity , without passing through the floor apparent velocity. In light of this a positive value of flow along the $\hat{y}$ axis is correlated to a positive value of velocity in the $\hat{x}$ axis.
The visual circumference around the camera, according to how it was defined above, has radius equal to the ground distance h. For our application, also h will be supplied by a sensor, in particular by an on-board sonar. Multiplying the introduced vectors it is possible to write equation (2.3).

In Figure 2.9 (b) where the drone is moving to the right, so along the $\hat{y}$ body axis, it is clear that, in this situation, the flow vector spin axis is equal to $\hat{x}$ in positive sense(ε_{x-true} in the figure); but according to the same reasoning, the optical flow from the sensor is available with the opposite sign (ε_x in the figure), so we obtain equation (2.4), that shows how a positive value of velocity in $\hat{y}$ axis is correlated to a negative value of the flow around the $\hat{x}$ axis.

Comparing the output data of the optical flow sensor with the measured angular velocities, we observe there is a constant offset between them. The knowledge of the firmware implemented on the sensor suggests to us that the sensor computes the optical flow integrating the angular velocity of the flowing images on camera, so its output is a delta angle. The constant offset is due to the integration time step equal to 0,1 [s]. The equations become:

$$\begin{cases} v_x = \dfrac{flow_y}{dt}\, h & (2.5) \\ v_y = \dfrac{-flow_x}{dt}\, h \quad , & (2.6) \end{cases}$$

with $dt_{Px4Flow} = 0.1s$.
Until now, we have assumed that the UAV does not change its attitude, but, as described above, it is impossible for a conventional multirotor UAV (with no-

tilting arms), to execute a translational motion without an attitude variation. In the equations (2.5) and (2.6) it is necessary to introduce the angular compensation discussed before, to avoid a wrong estimation of the velocity. The idea is to subtract the portion of optical flow generated by the rotation, keeping in mind that a positive angular velocity leads to a negative optic flow value and vice versa; so to take the angular rate information from the on-board gyroscopes and act as follows:

$$\begin{cases} v_x = (\dfrac{flow_y}{dt} + \omega_y)\, h & (2.7) \\ v_y = -(\dfrac{flow_x}{dt} + \omega_x)\, h & (2.8) \end{cases}$$

where ω_x is the angular velocity in x body axis and ω_y is the angular velocity in y body axis.

Equations (2.7) , (2.8) represent the ultimate relations between the velocities in $\hat{x}$ and $\hat{y}$ body axis and the optical flow vector. Rewriting and making them explicit with respect to the two component of the flow vector we will have the following equations to use in the mathematical model that we are going to use for the estimation problem. The equations are:

$$\begin{cases} flow_x = (-\dfrac{v_y}{h} - \omega_x)\, dt & (2.9) \\ flow_y = (\dfrac{v_y}{h} - \omega_x)\, dt \quad . & (2.10) \end{cases}$$

Chapter 3

Filtering estimation theory

In this chapter we briefly describe the estimation theory and the different ways to approach it, then focusing on our interest: the Kalman filter theory. All the equation and requirements of the Kalman filter algorithm are presented and described. In the last part of this chapter the H_∞ filter theory is presented focusing on its suitable form for the Kalman filter architecture.

3.1 Estimation theory

This chapter describes in general the state estimation theory, focusing on the filtering technique, that allows a real-time state estimation useful for on board application where it is necessary to have reliable information about the state vector (composed by position, velocity, attitude and all the other quantities that we want to estimate) at each time instant. In light of what has been described in the Chapter 1, it is clear that the navigation system could consists of various configurations of sensors with their advantages and disadvantages. The estimation theory gives us the tools to overcome individual sensor deficiencies obtaining a multi-sensor system designed to provide reliable and accurate estimation of the vehicle state vector, for all required flight conditions.

3.1.1 General overview

The aim of a navigation system is to allow the knowledge of the state vector of the vehicle at each time instant. In an ideal world this could be achievable simply reading the measurements of the on board sensors. In the reality , on the contrary, there are a lot of issues which make this simply way not executable. For example the noise of the sensors that affects the measurements, and all the other items which make the problem stochastic and not deterministic; for this reason there is the need to approach the problem in a statistical and probabilistic way, in order to take into account the uncertainties of the whole system. The other advantage to solve the navigation problem in this way is to have information about the state

variables with the same rate and, in most of the cases, with an higher rate with respect to data rate of the single sensor.

In a stochastic system, there are random variables in the system equations, so the envelope in time of the state variables must be estimated using a statistical method. There are many possible formulations of the state estimation problem, and before to briefly described them, focusing later on the state estimation for our application, it is better to define, in this paragraph, what is meant with stochastic problem. The problem can be formulated in different ways depending on the modelling assumptions:

- continuous-time dynamics and continuous-time measurement process;
- continuous-time dynamics and discrete-time measurement process;
- discrete-time dynamics and discrete-time measurement process;

According to what we have done in this thesis work, we are going to formulate the problem as a discrete-time dynamics with discrete-time measurement process. We introduce a generic discrete-time linear stochastic system modelled as suggested in [8]:

$$x(k) = \Phi(k-1)x_(k-1) + \Gamma(k-1)u(k-1) + \Gamma_w(k-1)w(k-1)\,, \tag{3.1}$$

$$y(k) = H(k)x(k) + D(k)u(k) + v(k) \qquad i = 1, 2, ..., N\,, \tag{3.2}$$

where x is the state vector, y is the output vector, z is the measurement vector and u is the input vector. The two noises w and v are, respectively, the process noise, accounting for disturbances and model uncertainty and the measurements noise acting on the output of the system. It is important to state that they are two white Gaussian noise processes. The Bayesian model for uncertainties is given as:

$$E[x(0)] = \bar{x}_0 \qquad E\{[x(0) - \bar{x}_0][x(0) - \bar{x}_0]^T\} = P_0\,, \tag{3.3}$$

$$E[w(i)] = 0 \qquad E[w(i)w^T(j)] = Q(i)\delta_{ij}\,, \tag{3.4}$$

$$E[v(i)] = 0 \qquad E[v(i)v^T(j)] = R(i)\delta_{ij}\,, \tag{3.5}$$

where $x(0)$, $w(i)$ and $v(i)$ are uncorrelated.

In addition, we introduce and define some notation useful for the understanding:

- $\hat{x}_k$ indicates the estimate of the state at the $t_k = k\Delta t$ time instant;
- e_k is the error vector in the estimate $\hat{x}_k$, defined as $e_k = x(k) - \hat{x}_k$;

- the state error covariance matrix is $P_k = E[e_k e_k^T]$.

For the system specified above, assuming to have data over the time interval from 0 to t_k, we can formulate three different estimation problems (see Figure 3.1):

- for $t > t_k \Rightarrow$ prediction, estimates how something will be in the future, based on what informations are available until now;
- for $t = t_k \Rightarrow$ filtering, for the estimation in the current time instant, thus useful for real-time applications;
- for $0 < t < t_k \Rightarrow$ smoothing, that can increase the accuracy of the estimation but knowing the whole data set, thus not applicable in for real-time purposes.

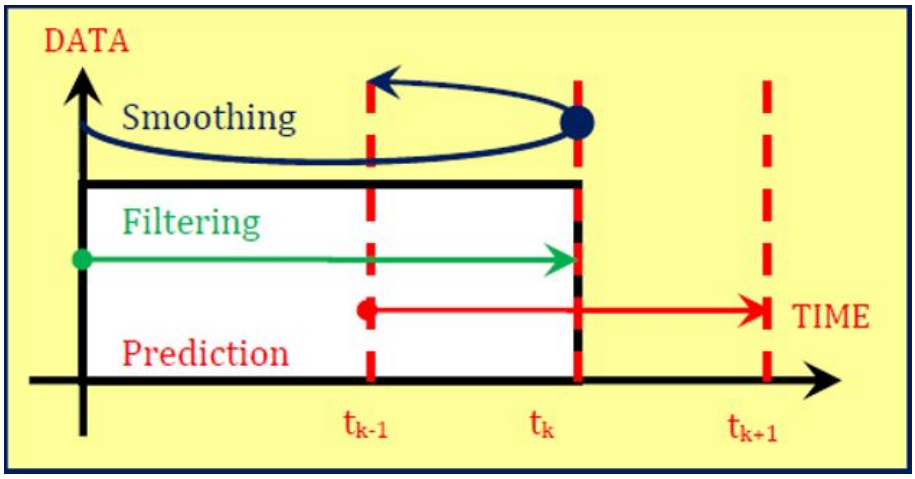


Figure 3.1: Estimation approaches

3.2 Filtering theory

The navigation system is something that the vehicle needs during the flight to know, at each time instant, its position, velocity and attitude; for this reason it is a real-time application of the estimation problem. According to our aims, in this paragraph the focus goes on the filtering theory and the on-line algorithm used in this work, to execute it.

3.2.1 Discrete-time Kalman filter

The Kalman filter is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-time linear filtering problem. First of all we have to state that the Kalman filter represents an optimal estimator if all the noises are Gaussian, and it means that the noises have a probability density function (PDF) equal to the probability density function of the normal distribution, which is also known as the Gaussian distribution. If this hypothesis is not confirmed, the Kalman filter represents the best linear estimator (non-linear estimators may be better). Below we will describe the formulation of the Kalman filter, basing on the state space model introduced in equations (3.1) (3.2). Concerning the stochastic terms, the process noise w and the measurements noise v represent white Gaussian noises with covariance matrices, respectively Q and R. Some assumption have to be defined, for the sake of simplicity:

- w and v are assumed uncorrelated;
- R and Q are assumed to be strictly positive definite.

It is fundamental to notice that, concerning the convergence to a steady-state solution for the recursive estimator, there are some minimal assumption needed:

- Observability of Φ and H;
- Controllability of Φ and $Q^{1/2}$.

If these two assumptions are satisfied, the asymptotic stability of the plant model is not necessary, as the filter still guarantees its stability.

The base idea of the Kalman filter, is to supply a recursive estimate ($\hat{x}$) of the state variables vector (x) at time k, using input (u) and output (y) data up to the same time instant k, in order to minimize the variance of the estimation error. We can imagine the solution to this problem as a particular form of Luenberger observer

$$\hat{x}_k = \Phi_{k-1}\hat{x}_{k-1} + K_k(y_k - \hat{y}_k)\,, \tag{3.6a}$$

$$\hat{y}_k = H_k\hat{x}_k, \tag{3.6b}$$

where the Kalman gain K is computed so as to minimize the variance of the estimation error

$$E[(x_k - \hat{x}_k)^T(x_k - \hat{x}_k)]\,. \tag{3.7}$$

In this discussion the Kalman filter is presented in its predictor-corrector form; in this way the estimation of the state vector is carried out in two different steps :

- the prediction step at time k consists of a one-step-ahead in the timeline, simply based on the propagation given by the plant model starting from the corrected state estimate in time $k-1$;
- the correction step at time k uses the computed Kalman gain, correcting the predicted state from the same time instant k.

The forward propagation through the deterministic part of the model, gives us the prediction step

$$\hat{x}_k(-) = \Phi_k \hat{x}_{k-1}(+) + \Gamma_k u_k \,. \tag{3.8}$$

The forward propagation step must be applied also in terms of covariances, so

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q_{k-1} \,, \tag{3.9}$$

The best structure for the correction step turns out to be of Luenberger types, defined as follows:

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k(y_k - \hat{y}_k) \,, \tag{3.10}$$

where

$$\hat{y}_k = H_k \hat{x}_k(-) \,. \tag{3.11}$$

The same idea to extend the prediction also to the covariance matrix, is valid also for the propagation step. The covariance matrix is updated as

$$P_k(+) = [I - K_k H_k] P_k(-). \tag{3.12}$$

The Kalman filter algorithm needs the numerical positive definiteness of the covariance matrix for the estimation error in order to not have stability issues, and to easily achieve this condition there is a different way to write the corrected P. This equation, known as Joseph form of the covariance update, is defined as follows:

$$P_k(+) = [I - K_k H_k] P_k(-) [I - K_k H_k]^T + K_k R_k K_k^T \,. \tag{3.13}$$

What we expect from the state error covariance matrix is that during the prediction step it becomes higher, because of the uncertainty that characterize a propagation step. The aim of the correction step is to reduce the values of the covariance matrix that means the error is going to a convergence, as shown in Figure 3.2.

We can notice that the main character of the correction step is the Kalman gain, which is updated in this way

$$K_k = P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1} \,. \tag{3.14}$$

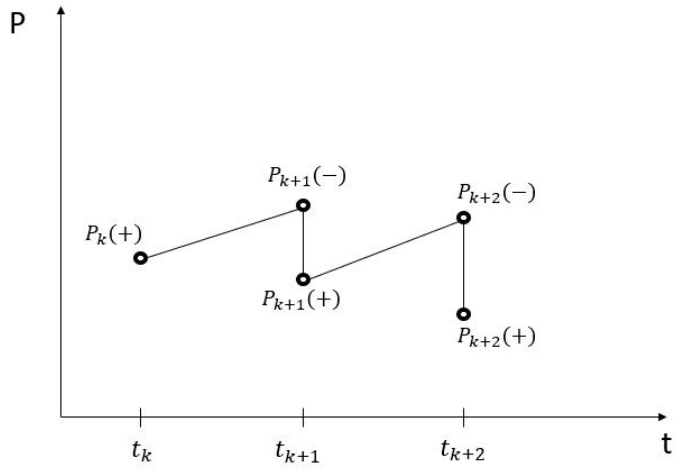


Figure 3.2: Example of P trend

The above presentation of the general solution to the Kalman filter, assumes that all the measurements of the model, are sampled at the same frequency, and that are available simultaneously at the same time instant k. This is an ideal condition difficult to reach in the reality, where measurements are supplied by different sensors with different sampling time. In this thesis the chosen solution to this issue is the sequential update. Its principle is to execute a correction step for each measurements. It will be described in the next chapter with the correlated advantages.

3.2.2 H_∞ filter

The presented Kalman filter is defined as the optimal solution for the filtering problem, assuming that the stochastic elements of the model (w and v) are white Gaussian noise with known covariance matrix, respectively, Q and R; but this assumptions is difficult to be confirmed in practice, so, over the years, others solution have been proposed. In this work the best-known alternative has chosen, the H_∞ filter. The purpose is to implement a formulation of this filter, able to be adapted to the pre-implemented structure of the Kalman filter. In this paragraph, a brief description of the theory of the H_∞ filter will be presented, and its formulation chosen for this work.

The idea is to have a filter able to optimize the state estimate solution according to a chosen cost function, stated that the absolute optimal solution is not reachable by the Kalman filter, due to the non-Gaussian noises w and v. The filtering problem is the estimation of

$$z_k = L_k x_k \,, \tag{3.15}$$

where z_k is named performance outputs vector and represents the quantities respect of which the optimization is executed, L_k is defined by the user and x_k represent the state vector. The cost function to minimize is formulated as follows:

$$J_1 = \frac{\Sigma_{k=0}^{N-1} \|z_k - \hat{z}_k\|_{S_k}^2}{\|x_0 - \hat{x}_0\|_{P_0^{-1}}^2 + \Sigma_{k=0}^{N-1}(\|w_k\|_{Q_k^{-1}}^2 + \|v_k\|_{R_k^{-1}}^2)}\,, \tag{3.16}$$

where $\hat{x}_0$ is the estimate of the initial state; P_0, Q_k and R_k is the already presented user-defined covariance matrices; S_k is the new symmetric, positive definite user-defined matrix; and at last we define the annotation

$$\|S_k\|_{R_k}^2 = S_k^T R_k S_k\,. \tag{3.17}$$

According to the theory, the direct minimization of the cost function J_1 is not reachable, so the approach is to look at the problem as

$$J_1 < \frac{1}{\theta}\,, \tag{3.18}$$

where θ is a performance bound defined by the user.

The problem can be reformulated as

$$J = -\frac{1}{\theta}\|x_0 - \hat{x}_0\|_{P_0^{-1}}^2 + \sum_{k=0}^{N-1}\|z_k - \hat{z}_k\|_{S_k}^2 - \frac{1}{\theta}\sum_{k=0}^{N-1}(\|w_k\|_{Q_k^{-1}}^2 + \|v_k\|_{R_k^{-1}}^2) \tag{3.19}$$

requiring to satisfy the condition

$$J < 1\,. \tag{3.20}$$

The described optimization problem is a min-max one, which can be defined as

$$J^* = \min_{\hat{z}_k} \max_{w_k, v_k, x_0} J\,, \tag{3.21}$$

which means to provide the best performance estimate ($\hat{z}_k$) considering the worst case about noise processes (w_k, v_k) and initial condition (x_0).

For our purposes, we want a formulation of the H_∞ filter, suitable with the Kalman filter structure (as a guideline for the following discussion [9] has used). The latter, by means of suitable manipulations can be written as

$$\hat{x}_{k+1}(-) = \Phi_k \hat{x}_k(-) + \Phi_k K_k(y_k - H_k \hat{x}_k(-))\,, \tag{3.22}$$

$$K_k = P_k(-)H_k^T[H_k P_k(-) H_k^T + R_k]^{-1}\,, \tag{3.23}$$

$$P_k(-) = \Phi_{k-1} P_{k-1}(+)\Phi_k^T - 1 + Q_{k-1}\,, \tag{3.24}$$

$$P_k(+) = [I - K_k H_k] P_k(-)\,. \tag{3.25}$$

can be easily manipulated and rewritten as

$$\hat{x}_{k+1}(-) = \Phi_k \hat{x}_k(-) + \Phi_k K_k(y_k - H_k \hat{x}_k(-)) \,, \tag{3.26}$$
$$K_k = P_k(-)[I + H_k^T R_k^{-1} H_k P_k(-)]^{-1} H_k^T R_k^{-1} \,, \tag{3.27}$$
$$P_{k+1}(-) = \Phi_k P_k(-)[I + H_k^T R_k^{-1} H_k P_k(-)]^{-1} \Phi_k^T + Q_k \,. \tag{3.28}$$

The H_∞ filter is given by

$$\bar{S}_k = L_k^T S_k L_k \,, \tag{3.29}$$
$$\hat{x}_{k+1} = \Phi_k \hat{x}_k + \Phi_k K_k(y_k - H_k \hat{x}_k) \,, \tag{3.30}$$
$$K_k = P_k[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k]^{-1} H_k^T R_k^{-1} \,, \tag{3.31}$$
$$P_{k+1} = \Phi_k P_k[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k]^{-1} \Phi_k^T + Q_k \,. \tag{3.32}$$

From this formulation it is clear that if $\theta \longrightarrow 0$ then the gain of the H_∞ filter becomes equal to the gain of Kalman filter; on the contrary if θ is chosen larger and larger, the two filters become increasingly different and it means imposing tighter bounds on the cost function J.

For the tuning of the H_∞ filter, according to our purposes, the choice has been the following:

$$z_k = y_k \,, \tag{3.33}$$

which means

$$L_k = H_k \,. \tag{3.34}$$

After this choice, the tuning parameter will be only the diagonal elements of the matrix S_k, which could be chosen, for example to provide scaling factors among the outputs:

$$S_k = \begin{bmatrix} S_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & S_l \end{bmatrix} . \tag{3.35}$$

In order to have a completely suitable formulation of the H_∞ filter with the Kalman gain structure, we need to recover a predictor/corrector form. Below the resulting formulation is defined (for deeper discussion see [10]) :

$$\hat{x}_{k+1} = \Phi_k \hat{x}_k + \Phi_k K_k (y_k - H_k \hat{x}_k)\,, \tag{3.36}$$

$$K_k = P_k [I + H_k^T R_k^{-1} H_k P_k]^{-1} H_k^T R_k^{-1}\,, \tag{3.37}$$

$$P_{k+1} = \Phi_k P_k [I + H_k^T \bar{R}_k^{-1} H_k P_k]^{-1} \Phi_k^T + Q_k\,, \tag{3.38}$$

having introduced the new "covariance" matrix

$$\bar{R}_k = [-\theta S_k + R_k^{-1}]^{-1}\,, \tag{3.39}$$

from which it is easy to observe that $\bar{R}_k \longrightarrow R_k$ for $\theta \longrightarrow 0$, reducing to the measurements covariance matrix of the Kalman filter formulation.

With R_k assumed diagonal, as in our work, and in view of the definition of S_k, also $\bar{R}_k$ is diagonal. Its diagonal elements, according to the equation (3.39), can be written as

$$\bar{R}(i,i)_k = -\theta S_i + \frac{1}{R_k^{[i]}}\,. \tag{3.40}$$

Also in this case, for the well-posedness of the filter it necessary that

$$\bar{R}_k > 0\,, \tag{3.41}$$

and it means

$$\min_{i=1,\dots l} -\theta S_i + \frac{1}{R_k^{[i]}} > 0\,, \tag{3.42}$$

so

$$\theta < \min_{i=1,\dots l} \frac{1}{S_i R_k^{[i]}}\,, \tag{3.43}$$

with l = length of the diagonal of $\bar{R}_k$.

This could be a useful guideline for the tuning of the H_∞ filter.

Chapter 4

Position and velocity filtering

In this chapter there is the description of what has been implemented in practice, in order to reach a position and velocity state estimate based on GPS and optical flow measurements, according to the filtering theory illustrated in the Chapter 3. The discussion about every choice and assumption will be described below.

4.1 Mathematical model

The used model is a discrete-time kinematic one, based on the relation between position, velocity and acceleration. It is written as follow:

$$\begin{cases} p_k = p_{k-1} + dt\, v_{k-1} \\ v_k = v_{k-1} + dt\, a_k\,. \end{cases} \tag{4.1}$$

Where k is the current time instant; p, v and a are respectively the position, the velocity and the acceleration of the UAV; dt defined as the time step of our discretization. This equations system represents the time propagation model of the problem. We can observe that, in order to pass from the continuous-time model to the discrete one, some choices have been done. The first equation of the system (4.1) could be seen as a commonly known Backward Euler method that give us a time propagation that depends only on past value of the state variables.

On the contrary, for the second equation of the system (4.1), the chosen discretization uses the acceleration at the current time instant, giving back a Forward Euler formulation, where the propagation in time of the left side state variable depends on both current e past quantities. Usually this leads to a heavier computational load because of the non linearity of the system (for insights see [11]), receiving back a best numerical stability situation. In our case the computational load remains the same and the equation linear, due to the following choice about the acceleration. In the design phase of the mathematical model, the possibility to implement a system of two state variables rather than three, has taken into

account and it has become the final strategy. It means to define the position and the velocity as state variable, and the acceleration as an input reliable from the accelerometers. This choice has two main effects:

- the problem to estimate two state variables rather than three;
- the process noise w of the model is given by the accelerometers standard deviation and it means less possibility of free tuning.

Starting from this model, we define a state-space representation of the model focusing on each of its elements. Using the same notation introduced in the previous chapter, the state space model is defined as

$$x_k = \Phi_{k-1} x_{k-1} + \Gamma_{k-1} u_{a\,k-1} + G_{k-1} w_{k-1}\,, \tag{4.2a}$$
$$y_k = H_k x_k + D_k u_{w\,k} + v_k\,. \tag{4.2b}$$

The first equation represents the system (4.1), thus we define the x as the state vector containing the two state variables, position and velocity in their three components, so we can write

$$x = \begin{bmatrix} p_x \\ p_y \\ p_z \\ v_x \\ v_y \\ v_z \end{bmatrix}_{NED}\,, \tag{4.3}$$

where p_x, p_y, p_z are the three components of UAV position on the three axes of the NED reference frame; v_x, v_y, v_z are the three components of UAV velocity in the NED reference frame.

Φ is the state transition matrix that allows us to execute the time propagation of the state vector. According to the mathematical model it is defined as follow:

$$\Phi = \begin{bmatrix} I_3 & dtI_3 \\ O_3 & I_3 \end{bmatrix}\,, \tag{4.4}$$

where the subscript "3" indicates a 3x3 dimension for the related element.

In the state space representation above (equations (4.2a) (4.2b)), we can notice that the input vector have been split into two vectors: one for the inputs acting on the state equation and one for that about the output equation. This have been done simply to get more comfort during the implementation of the filter. In view of this we have:

$$u_a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{NED}\,, \tag{4.5}$$

where a_x, a_y, a_z are the three components of the acceleration in NED reference frame, which excite the state model;

$$u_\omega = \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}_{BODY} , \tag{4.6}$$

with ω_x, ω_y equal to the angular velocity in two axes of the body reference frame, necessary to execute the angular compensation of the optical flow measurements, as discussed in 2.3.1. For the angular velocity measurements we use the gyroscopes mounted on board, in particular we can choose between those integrated on the IMU(inertial measurements unit) or those from the optical flow sensor. We have chosen the latter and the reason is about their sampling frequency; in order to have a more comfortable implementation of the angular compensation for the optic flow is convenient to use a measurement of angular velocity sampled at the same frequency of the flow vector.

Γ is the coupling matrix between input vector and state variables vector, and for our model it is defined as

$$\Gamma = \begin{bmatrix} O_3 \\ dtI_3 \end{bmatrix} , \tag{4.7}$$

that shows how the acceleration input excites directly only the velocity. At the beginning of the section we have stated that the random process is given to the model by the acceleration input (in the next section the characterization process will be described). What is necessary to notice for the model is that

$$a_m = a + w_a \, , \tag{4.8}$$

$$\Rightarrow a = a_m - w_a \, , \tag{4.9}$$

where a is the true acceleration that in an ideal case must be used as input in our system, but of course, what is available for this task is the measured acceleration a_m. The accelerometers return, as it is well known, a measurement with noise w_a. Introducing the equation (4.9) in our model we will have a matrix (G) that couples the random process noise to the state variables, defined as follow:

$$G = \begin{bmatrix} O_3 \\ -dtI_3 \end{bmatrix} . \tag{4.10}$$

The second equation ((4.2b)) of the state space model represents the measured outputs and how they are coupled with the states and the inputs of the system. For our application the measured output vector y rely on the measurements of

the GPS and the optical flow sensor (Px4Flow), and we define it as

$$y = \begin{bmatrix} pos_x \\ pos_y \\ pos_z \\ flow_x \\ flow_y \end{bmatrix}, \tag{4.11}$$

where pos_x, pos_y, pos_z come from the GPS and we want them representing the UAV position components on the three axes of the NED reference frame, for this reason it is necessary to convert the GPS measurements from the geodetic frame to the NED one; $flow_x$ and $flow_y$ are the measurements of the flow vector in body axis. The matrix that links the state to the output is the measurements sensitivity matrix H and it is defined as follow:

$$H = \begin{bmatrix} I_3 & O_3 \\ O_{2x3} & H_{flow} \end{bmatrix}, \tag{4.12}$$

where H_{flow} is, according to what is explained in 2.4, the matrix that represents the relation between the measurements of the optical flow sensor and the velocities of the state vector; it is defined as

$$H_{flow} = \begin{bmatrix} 0 & -\frac{dt_{flow}}{h} \\ \frac{dt_{flow}}{h} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} A_{i2b}\,, \tag{4.13}$$

where dt_{flow} is the integration time implemented on the sensor firmware as specified in Chapter 2, and it is equal to 0.1 seconds; h is the distance from the ground; the second matrix is a selector matrix, necessary to link the flow vector only to the v_x and v_y component; A_{i2b} is the attitude matrix defined as in (1.15) and it is used in order to rotate the velocity vector from the NED inertial frame to the body frame.

The last matrix to be defined is D that describes the relation between the angular velocity seen as an input and the flow measurements, in order to ensure the angular compensation. It is stated as

$$D = \begin{bmatrix} -dt_{px4Flow} I_2 \end{bmatrix}. \tag{4.14}$$

Before to discuss about the implementation of the Kalman filter, an observation is necessary, about the above defined matrices of the model. While all the other matrices remain constant during the time iterations, the H matrix is time-variant due to the presence of the attitude matrix. The latter is defined through the estimate quaternion coming from an attitude estimator, so at each time instant the attitude matrix must be re-defined to ensure the following of the actual attitude of the UAV, thus rotating the velocity vector in the right way.

In view of this, the annotation will expect the use of the subscript (k) only for the H matrix and not for the other matrices of the mathematical model.

4.2 Implemented Kalman filter

According to the Kalman filter theory presented in Chapter 3, this section will focus on the practical implementation of the filter equations, with all the needs required by the reality.

4.2.1 Data pre-processing

Before to use the sensor data as measurements (y) and inputs (u), their rotation into the appropriate reference frames must be executed. In particular two data sets need to be rotated:

- GPS data from the geodetic reference frame to the NED one;
- acceleration data from the body reference frame to the NED one.

About the angular compensation of the optical flow, one observation must be noticed: the Px4Flow , as stated in 2.4, gives back its measurements as a delta angle and not as an angular velocity, so to use its gyroscopes outputs as something expressed in $[rad/s]$ we need to divide them for dt_{flow}. In this way it is not necessary to change the H_{flow}, so it is possible to have a model usable also if we decide to use the gyroscopes of the IMU.

The last adjustment to do is about the measurement of the ground distance. It is obtained by the sonar integrated in the Px4Flow and it is used into the H_{flow} matrix as a denominator. For this reason, to avoid singularity problem due to $h = 0$, we have chosen to put a lower bound to this measurement, equal to $h_{min} = 0.3[m]$ (value that represents also the minimum operative distance from an observed surface, for the lens of the Px4Flow camera).

4.2.2 Initialization

The Kalman filter algorithm needs an initial guess for the state vector (x_0). The choice of this initial estimate should be based on prior knowledge, if available, of the typical average values for the individual components of the state. The state vector in our problem represents the position and the velocity of the UAV, so, reasonably, it can be initialize equal to zero. The initial covariance (P_0), on the other hand, is used to express the confidence in the initial state vector; more we trust in the initial guess and smaller the initial covariance values will be; on the contrary if we don't have prior knowledge about the initial state variables, the values of P_0 may be higher. From the equation of the Kalman gain (3.14), it

is clear how this affects the work of the filter; large values of the state covariance matrix imply large gains, which pick up the value of the state estimate far from the initial guess. For this thesis work the chosen values for x_0 and P_0 are:

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \end{bmatrix}, \qquad P_0 = 10^{-6} I_6 \,. \tag{4.15}$$

4.2.3 Tuning

The tuning of the Kalman filters consists of choosing the appropriate values for the covariance matrices Q and R. It is a problem which has been studied extensively in the literature but there is not systematic approach to do that and it depends a lot on the preliminary knowledge of the designer about the model. Usually the R matrix is easier to be defined, due to do the fact that it represents the measurement noise covariance, so it is strictly linked to the data from the sensors. On the contrary the Q matrix is related to the process noise that could be the effect of a lot of elements. In view of this, the R matrix in our case has been defined thanks to the variances of the analyzed data coming from the sensor. It is, reasonably, assumed diagonal, and it means that the noise of a certain measurements gives uncertainty only on the output correlated to that measurement, so each noise is independent. The measurement covariance matrix is:

$$R = \begin{bmatrix} R_{gps} & O_{3x2} \\ O_{2x3} & R_{flow} \end{bmatrix}, \tag{4.16}$$

with

$$R_{gps} = \begin{bmatrix} \sigma^2(gps_x) & \cdots & 0 \\ \vdots & \sigma^2(gps_y) & \vdots \\ 0 & \cdots & \sigma^2(gps_z) \end{bmatrix}, \qquad R_{flow} = \begin{bmatrix} \sigma^2(flow_x) & 0 \\ 0 & \sigma^2(flow_y) \end{bmatrix}, \tag{4.17}$$

where $\sigma^2(\bullet)$ indicates the variance of the measurement "$\bullet$" (defined as Var($\bullet$)). The estimate problem is formulated as a discrete-time problem, so it is necessary to use discrete covariance matrix and as in our case, if the computation of the sensor variances is done in continuous time, we need to act as follow:

$$R_d = R/dt \,. \tag{4.18}$$

The process noise, as explained above, is more difficult to define, but in our problem, thanks to the choice to use the acceleration as an input we can have a really good approximation of the real Q starting from the value of the accelerometer noise. Looking at the mathematical model (4.1), the acceleration variance seems to affect only the velocity, but if we want a more real values of Q, we have to propagate the covariance thanks to the following formulation (stating that the below discussion is done for one generic axis and it is the same for all the direction of the reference frame):

$$Q_k = cov(w_k) = \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} G Q_w G^T e^{A^T(t_{k+1}-\tau)} d\tau \,, \tag{4.19}$$

with

$$Q_w = \sigma_{acc}^2(\tau) \,, \tag{4.20}$$

where τ is the time variable and σ_{acc}^2 represents the variance of the acceleration measurement. Notice that the equation.(4.19) involves matrix from the continuous-time state space model that we report here for sake of clarity:

$$\dot{x} = Ax + Bu + Gw \,, \tag{4.21}$$

that in detail it is written as

$$\left\{ \begin{array}{c} \dot{p} \\ \dot{v} \end{array} \right\} = \left[\begin{array}{cc} 0 & 1 \\ 0 & 0 \end{array} \right] \left\{ \begin{array}{c} p \\ v \end{array} \right\} + \left[\begin{array}{c} 0 \\ 1 \end{array} \right] a + \left[\begin{array}{c} 0 \\ -1 \end{array} \right] w_a \,. \tag{4.22}$$

Solving the analytic equation (4.19) we obtain a discrete covariance matrix Q_d for the single axis equal to

$$Q_d = \left[\begin{array}{cc} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{array} \right] \sigma_{acc}^2 \,, \tag{4.23}$$

where Δt is the time interval between two time instants. From this formulation it is clear that now the problem is to identify experimentally the variance of the accelerometer for each axis (σ_{acc}^2). The procedure and results will be shown in the Chapter 6.

4.2.4 Multi-rate management: sequential update

In Chapter 3 we have underlined how the generic theory of the Kalman filter is based on the assumption of having all the data set at the same rate. On the UAV used for this thesis there is a large group of sensors, which are involved in the Kalman filtering process. Due to their different nature, they have different sampling frequencies, making our filtering problem a multi-rate one. In particular the sensors has the following sampling frequencies:

- accelerometers sample at $125\,Hz$;
- Px4Flow (that gives data about optical flow, angular rate and ground distance) samples at $10\,Hz$;
- Gps samples at $5\,Hz$.

In order to manage this multi-rate problem, a sequential update approach is used. In our case it is executed following these instructions:

- the discrete-time Kalman filter is run at the fastest update frequency, namely the accelerometers one ;
- at each time update, the availability of each measurement is verified;
- only the available measurements are used for the correction step of the filter.

This approach is possible to follow due to the matrix of the covariance measurement noise R assumed to be diagonal. This allow us to use two independent blocks of the diagonal of R for the different sensors measurements used to correct the estimate of the state vector.

The structure of the prediction and correction process of the Kalman filter implemented for our work could be resumed as follow:

- prediction step

$$\hat{x}_k(-) = \Phi_k \hat{x}_{k-1}(+) + \Gamma_k u_k \,, \tag{4.24}$$

$$P_k(-) = \Phi_{k-1} P_{k-1}(+) \Phi_{k-1}^T + Q_{k-1} \,; \tag{4.25}$$

- GPS correction step

$$K_{gps\,k} = P_k(-) H_{gps\,k}^T [H_{gps\,k} P_k(-) H_{gps\,k}^T + R_{gps\,k}]^{-1} \,, \tag{4.26}$$

$$P_k(+) = [I - K_{gps\,k} H_{gps\,k}] P_k(-) [I - K_{gps\,k} H_{gps\,k}]^T + K_{gps\,k} R_{gps\,k} K_{gps\,k}^T \,, \tag{4.27}$$

$$\hat{x}_k(+) = \hat{x}_k(-) + K_{gps\,k} (y_k - \bar{H}_{gps\,k} \hat{x}_{1:3\,k}(-)) \,, \tag{4.28}$$

where $H_{gps\,k}$ is the sub-matrix of $H(i,j)$, with $i = 1,...,3$ and $j = 1,...,6$; and $\bar{H}_{gps\,k}$ is the sub-matrix of $H(i,j)$ with $i = 1,...,3$ and $j = 1,...,3$; notice that the innovation error (element that multiply the $K_{gps1,k}$) in equation (4.28) is executed using only the state vector components related to the position;

- optical flow correction step

$$K_{of\,k} = P_k(-)H_{of\,k}^T[H_{of\,k}P_k(-)H_{of\,k}^T + R_{of\,k}]^{-1}\,, \tag{4.29}$$

$$P_k(+) = [I - K_{of\,k}H_{of\,k}]P_k(-)[I - K_{of\,k}H_{of\,k}]^T + K_{of\,k}R_{of\,k}K_{of\,k}^T\,, \tag{4.30}$$

$$\bar{y}_k = y_k - Du_\omega\,, \tag{4.31}$$

$$\hat{x}_k(+) = \hat{x}_k(-) + K_{of\,k}(\bar{y}_k - \bar{H}_{of\,k}\hat{x}_{4:5\,k}(-))\,, \tag{4.32}$$

where $H_{of\,k}$ is the sub-matrix of $H(i,j)$, with $i = 4,...,5$ and $j = 1,...,6$; and $\bar{H}_{of\,k}$ is the sub-matrix of $H(i,j)$ with $i = 4,...,5$ and $j = 4,...,6$; notice that the innovation error (element that multiply the $K_{of1,k}$) in equation (4.32) is executed using only the state vector components related to the velocity in x and y axes.

Following the above structure, we obtain an estimate state vector at the rate of the Kalman filter, with a correction based on the availability of the sensors measurements, thus on their rate. This allows a perfect management of the multi-rate problem and a lower computational load during the correction step.

The last thing to describe is the way to check the availability of the measurements in a real-time problem as the Kalman filter. It is illustrated in the following block diagram:

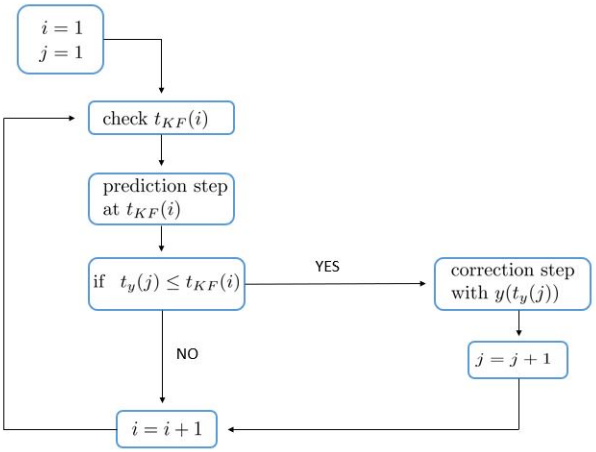


Figure 4.1: Measurements availability check

where t_{KF} is the time-line vector of the Kalman filter and the i is the index that indicates the element of this vector; in the same way t_y represents the time-line vector of the measurement of which we want to check the availability and j indicates the element of the vector; in the block of the correction step , $y(t_y(j))$ is the measurement taken at the time instant equal to $t_y(j)$.

4.3 Implemented $\mathbf{H}_\infty$ filter

The implementation of the H_∞ filter has not critical aspects thanks to the effort done to reach the formulation discussed in 3.2.2. The problem is reduced to write the "new covariance" matrix $\bar{R}$ according to the equation (3.39) where the only parameter that needs to be tuned is the θ. Substantially, in the code, a flag has been included in order to allow us to chose which one between the two filters we want to use for the estimate; the flag drives us in a section of the implemented code where we can tune the θ parameter in order to change the filter performances.

Chapter 5

Experimental set-up

In this chapter the experimental set-up is described. In particular a brief description of the used UAV is done followed by a description of the used optical flow sensor, Px4Flow. The on board integration of the optical flow sensor is presented focusing on the hardware set-up. In the last part of this chapter, the required pre-test activity are described.

5.1 Hardware set-up

The machine used for the experimental activity of this work, is an hexa-copter UAV, designed and become operative during previous works executed in the ASCL (Aerospace System & Control Laboratory) of Politecnico di Milano. The drone has the following properties:

- configuration $\rightarrow$ six propellers ;
- TOW $\rightarrow$ 900 g ;
- flight duration $\rightarrow$ 20 minutes ;
- payload $\rightarrow$ 350 g .

The optic flow sensor, main character of this thesis work, must be installed co-existing with an already implemented pack of sensors. This is composed by :

- IMU platform (3-axis accelerometers, 3-axis gyroscopes, magnetometer for the heading);
- barometer;
- GPS antenna.

5.1.1 Optical flow sensor: Px4Flow

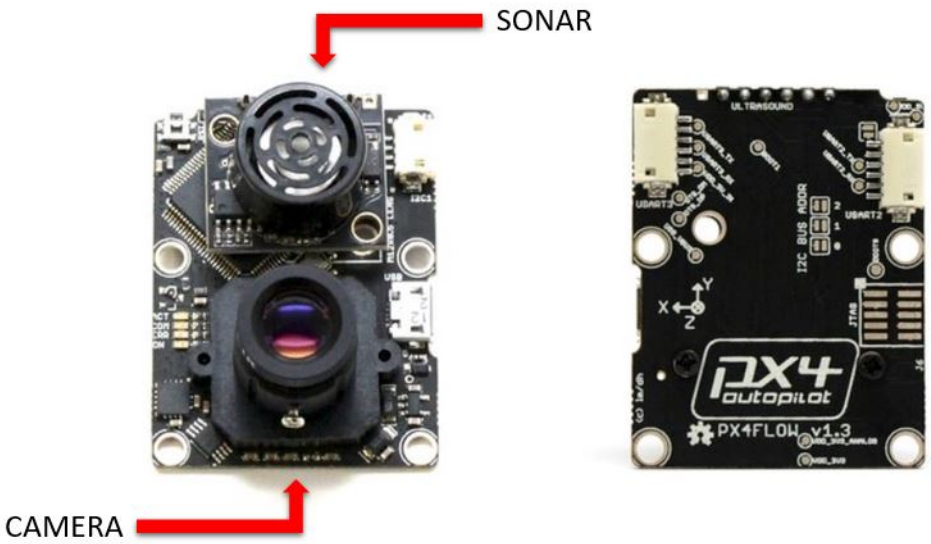


Figure 5.1: Px4Flow sensor

Px4Flow, introduced in Figure 5.1, is an optical flow smart camera created to calculate optic flow from the observation of images. It is not designed to capture images like a common web-cam but it is able to supply the video just for calibration proposal (not when it is working on the UAV). Unlike many mouse sensors, it also works indoors and in low outdoor light conditions without the need for an illumination LED, but it is also very sensitive to high light conditions due to its very high frequency of optical flow calculation, 400 Hz.

The Px4flow is not just a camera, but it is composed by all the necessary to comfortably use the optical flow data. On board we can appreciate a set of gyroscopes for the angular compensation and a sonar for the ground altitude scaling.

5.1.1.1 Technical properties:

- 168 MHz Cortex M4F CPU (128 + 64 KB RAM);
- 16 mm M12 lens;
- Size 45.5 mm x 35mm;
- Power consumption 115mA / 5V;

- On board 16bit gyroscope up to 2000°/s;
- On board sonar;
- USB bootloader and power option.

5.1.1.2 Dimensions and connectors scheme:

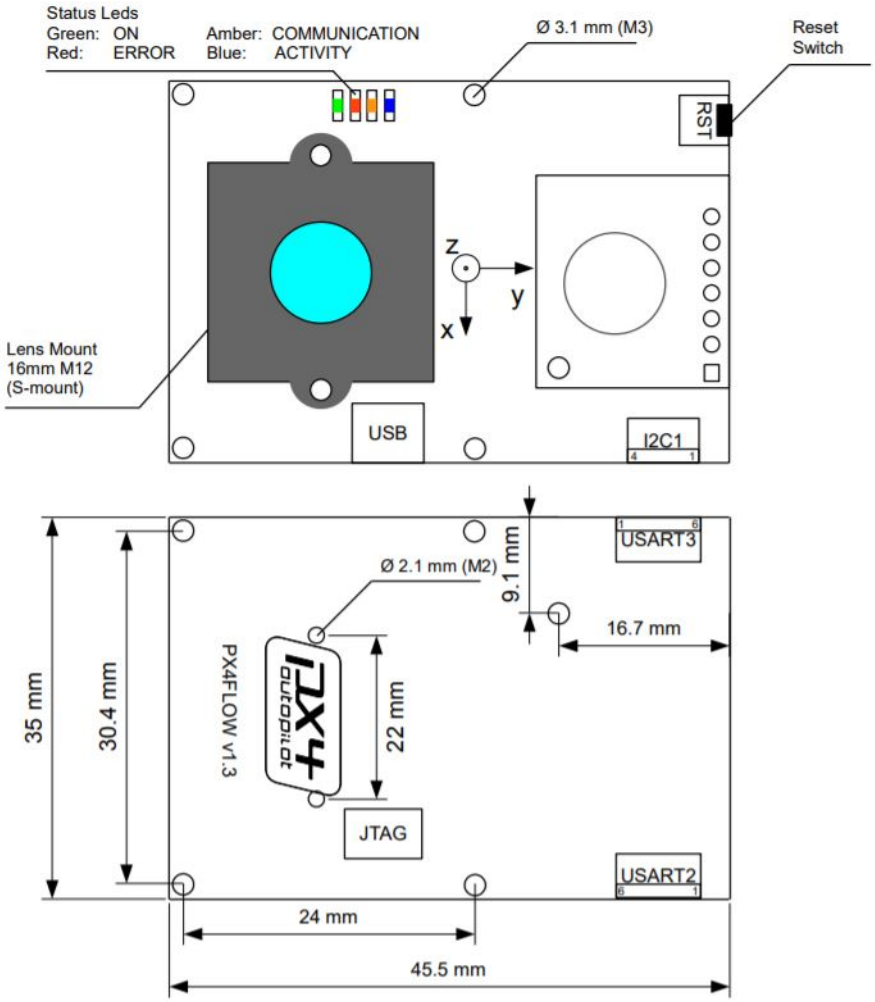


Figure 5.2: Connectors and dimensions

5.1.2 On board integration

The choice to use the hexa-copter has taken for reason of comfortable allocation possibility and with respect to the requirements of payload and power indicated

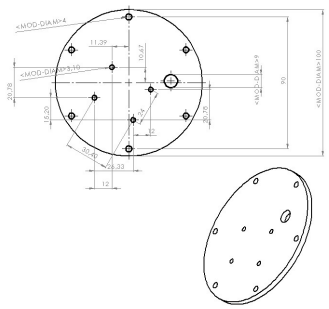
by the technical specifications of the sensor in 5.1.1.1.
The Px4Flow needs to be oriented looking downward; and the chosen UAV has the space to allocate a new rigid platform to host the new sensor. The supplementary platform gives us the possibility to set the new electronic component without taking space to the pre-existing sensors and modules.

5.1.2.1 Design and realization of the platform

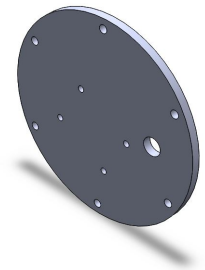
The platform requires few very important things:

- it must not give a big increase of weight to the structure;
- it must supply enough space for the allocation of the Px4Flow
- it must ensure the necessary opening to allow the wiring between the sensor and the flight controller placed in the core of the UAV.

The design , shown in Figure 5.3, has made using the 3D mechanical CAD program SolidWorks, obtaining the file that allow to generate a gcode on Cambam software, necessary to use the milling machine for the realization of the platform. The chosen material for this item is the carbon fiber that ensures a good compromise between weight and good mechanical properties. For the final assembly, the procedure suggested in the on-line guide has been followed; where one of the most important recommendation is about the orientation of the sensor, in order to be coherent with the orientation (so with the reference frame) of the Flight Control Unit, as it is described in the Figure 5.5 below.

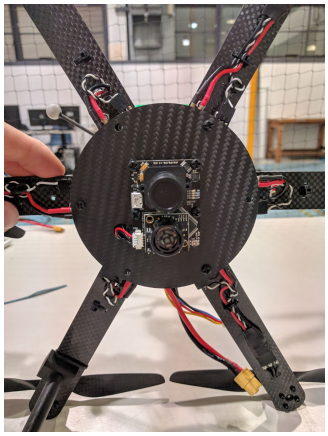


(a) 2D sketch of the platform



(b) 3D model of the platform

Figure 5.3: Design of the platform



(a) Bottom view

(b) Side view

Figure 5.4: Px4Flow assembled on the UAV

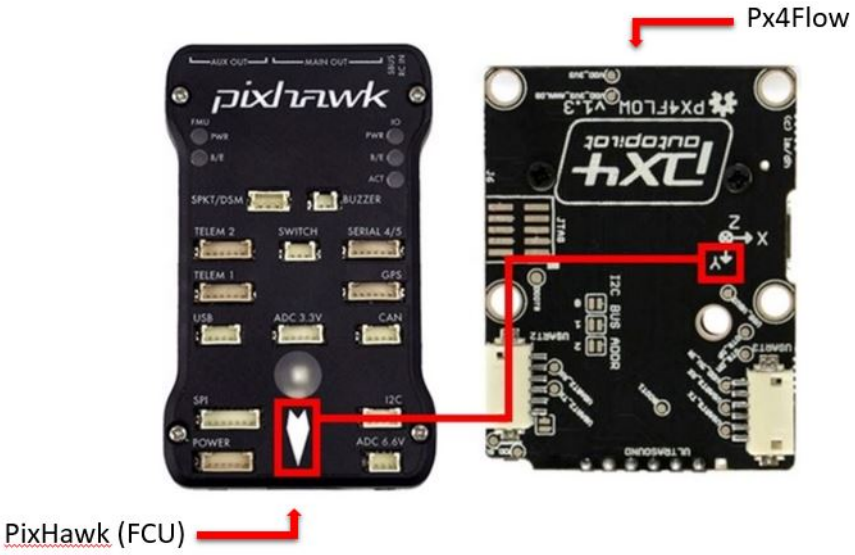


Figure 5.5: Px4Flow orientation

5.2 Pre-test activities

There are some procedures to follow before doing a flight test, in order to have a valid data set. The program QGroundcontrol gives us an appropriate ground station for the following described activities. First of all, what needs to be done is the upload of the firmware on the PixHawk (Flight Control Unit) and the

Px4Flow; the firmware could be the default one or a tuned one depending on what we desire from the machine.

The second step is to calibrate all the sensors on board. In this description our focus goes on the calibration of the Px4Flow lens; Its camera needs to be focused on the image, taking into account the range of ground distance of the test that we want to execute. Its ability to catch good images depends a lot on this, and a poor vision quality could completely affect the optical flow data. A parameter named "Quality" is available from the messages of the Px4Flow, and logging it, we can have a qualitative measurement of how well the camera is catching images useful for the computation of the optic flow. The sensor is set to work only when this parameter is bigger than a chosen threshold. There are some elements which generate variation on the quality parameter and their management allow us to keep it higher , thus to have a reliable data set for our analysis. Those elements are: brightness, surface roughness and colour, lens calibration. The latter is where we can and must act. The procedure is executable on the QGroundcontrol and it requires :

- to plug-in the Px4Flow to the ground station;
- to rotate the lens, as a common camera, till the images shown on the screen have a very good resolution (for example , like in Figure 5.6, pointing the camera on the keyboard of a computer in order to use the readability of the alphanumeric symbols as a feedback for the focus calibration.

When the above procedure are executed, the UAV, equipped with the Px4Flow, is ready to fly and we are ready to do the chosen flight tests.

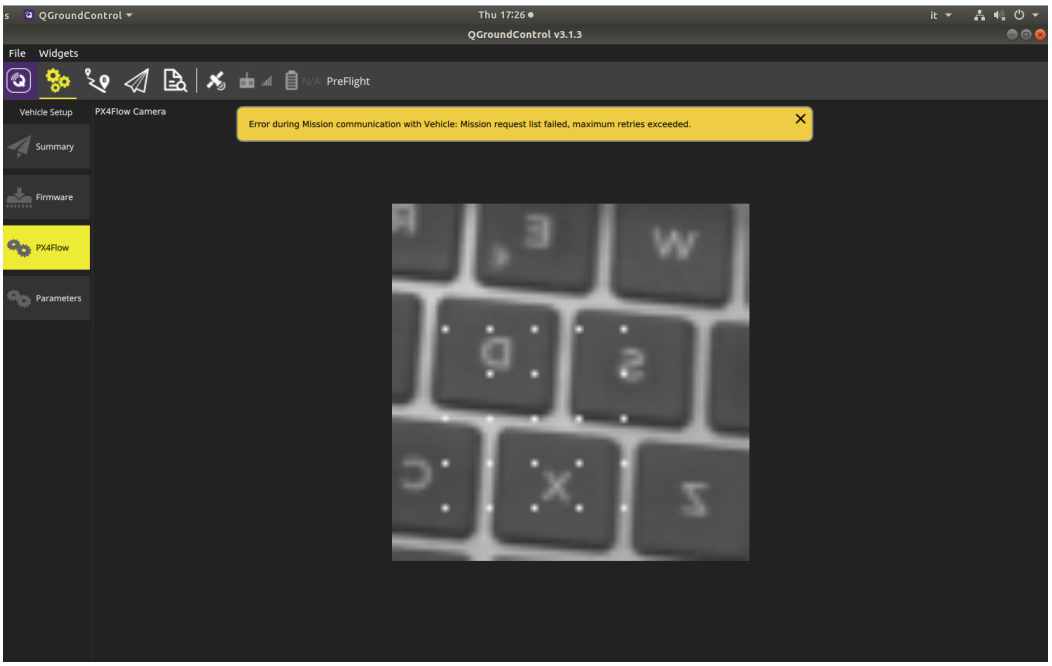


Figure 5.6: Lens calibration

Chapter 6

Experimental results

First of all in this chapter the procedure and the test results used to obtain the values of the sensors measurements variances are explained. In the second part all the results obtained by applying the implemented filters on the real data sets collected during the flight testing activity are shown and discussed.

6.1 Covariance characterisation

The paragraph 4.2.3 describes how to define the covariance matrices Q and R, and we can observe that the way to proceed is to find a physical value for the variances of the sensors measurements. Below the experimental procedure to reach them and the results are shown.

6.1.1 Procedure description

According to our implementation of the Kalman filter we need the continuous sensors measurements variances, so the evaluation of it by the power spectral density (PSD) analysis is necessary. The procedure to follow is: compute a vector as the difference between the measurements data and the mean value of the same data set (this operation allows us to identify only the noise of the measurements); plot the trend of the square root of the one-sided PSD of this new vector with respect to frequency; from the resulting plot we can properly identify a value to approximate the model of the sensor noise with a straight horizontal line passing through that value; for the approximation of the sensor noise we have decide to take a mean value of the plotted square root of the one-sided PSD, described above, observing the set of frequency from the lowest one to the $1Hz$ one, in order to focus the problem on a set of frequencies similar with the range of the UAV dynamics. According to this, to obtain the variance value we have to act as follows:

in the resulting plots (*i.e.* in Figures 6.1 , 6.2, 6.3) we read the values of

σ multiplied by $\sqrt{2}$ (where σ is the standard deviation of the measurement); so computing the mean value in the above described range of frequencies, we have

$$m = \sqrt{2}\sigma\,, \tag{6.1}$$

manipulating it to obtain the standard deviation

$$\sigma = \frac{m}{\sqrt{2}}\,, \tag{6.2}$$

thus the variance

$$Var = \sigma^2\,. \tag{6.3}$$

6.1.2 Test results

The tests for this aim have planned in order to have the most reliable variance values for on board sensors. The results, shown below, come from the best flight tests for this purpose.

The flight used to characterize the accelerometer and flow sensor variances, is executed outdoor in perfect visibility conditions for the lens of the Px4Flow. In Figure 6.1 the square root of the accelerometer measurements PSD, in the three body axis, is shown; while the square root of the flow PSD in x and y body axes is represented in Figure 6.2.

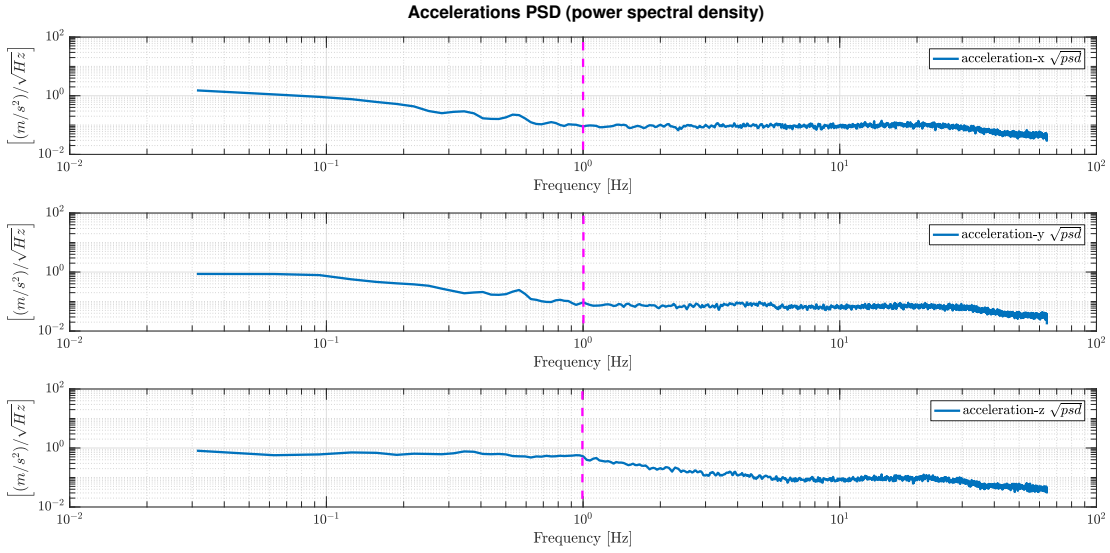


Figure 6.1: PSD analysis for accelerometers

Table 6.1: Accelerometers variance values in $(m/s^2)/\sqrt{Hz}$

	$\mathbf{acc_x}$	$\mathbf{acc_y}$	$\mathbf{acc_z}$
m	0.34	0.27	0.65
Var	0.0578	0.0364	0.2112

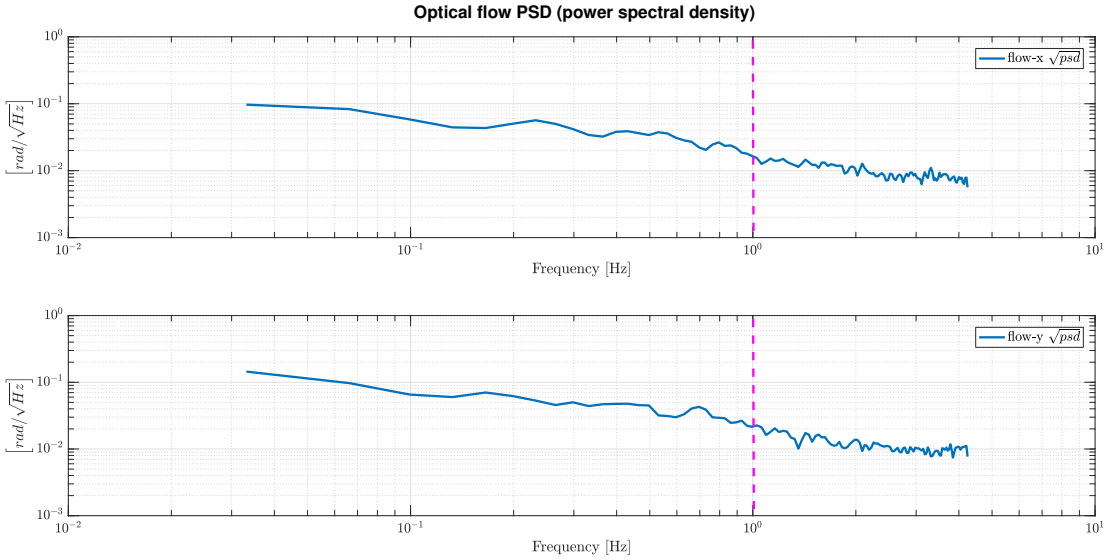


Figure 6.2: PSD analysis for optical flow

Table 6.2: Optical flow variance values in $rad/\sqrt{Hz}$

	$\mathbf{flow_x}$	$\mathbf{flow_y}$
m	0.0375	0.0476
Var	0.0007	0.0011

In Table 6.1 and Table 6.2 are reported the numerical values obtained using the described procedure. Putting them in our covariance matrices, make the tuning of the Kalman filter coherent with the reality.

For the definition of the GPS variances we have planned a different test in order to obtain more reliable results. This test has been executed following these instructions:

- UAV placed on a 3 meters tall platform to avoid multipath GPS error;
- data logged with the no-flying UAV but with armed propellers, in order to have disturbance conditions similar to the flight mode.

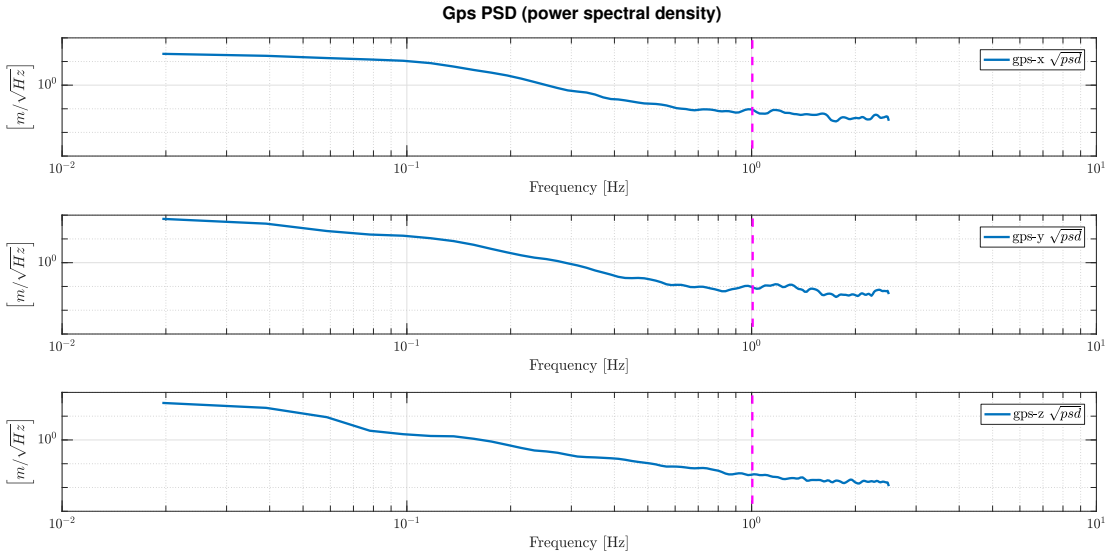


Figure 6.3: PSD analysis for GPS position

Table 6.3: Gps variance values $m/\sqrt{Hz}$

	$\mathbf{gps_x}$	$\mathbf{gps_y}$	$\mathbf{gps_z}$
m	0.2739	0.432	0.968
Var	0.0375	0.0933	0.4685

From the trend of the square root of the PSD plotted in Figure 6.3 with respect to frequency, we can extrapolate the value of m, thus of the variance (Var) reported in Table 6.3 , in order to complete the set of useful sensor variances for the tuning.

6.2 Estimate results

The performance of the implemented Kalman filter and H_∞ filter has been tested thanks to data sets collected during a large number of flight tests. In this chapter, we present the results related to the tests evaluated to be the best to shown the completeness of the performance, results and limitation of this Gps and optical flow based estimation filter. Five cases will be discussed and described:

- Gps based Kalman filter in outdoor flight;
- Gps and optical flow based Kalman filter in outdoor flight;
- optical flow based Kalman filter in outdoor flight;
- optical flow based Kalman filter in indoor flight;

- Gps and optical flow based H_∞ filter in outdoor flight.

The reader must be advised that all the results about the position and velocity state variables are expressed in NED reference frame. Before discussing the results, below we describe the two flight tests used to apply the implemented filters and get the estimate results.

6.2.1 Flight tests description

6.2.1.1 Outdoor flight

(a) Outdoor scenario

(b) Outdoor surface

Figure 6.4: Outdoor test

The outdoor flight has performed at the heliport of the Bovisa campus of Politecnico di Milano, in order to reduce or completely avoid the multipath error for the GPS. The weather conditions were optimal in order to have a high quality of the Px4Flow measurements that means a high visibility of its camera. The second element that contributes to the high quality of the optical flow data is the outdoor surface, as we can appreciate in Figure 6.4 . The duration of the whole flight is equal to 11 minutes and 40 seconds (700 seconds). In Figure 6.5 the parameter Px4Flow quality is shown and indicates how good the Px4flow camera is observing the ground surface as described in paragraph 5.2 . As we can see, there is a threshold that indicates when the optical flow is considered completely reliable, and it is equal to 250. We can observe that during this outdoor flight, the chosen environment conditions ensure high quality of the optical flow measurements, so we expect good results from the filter.

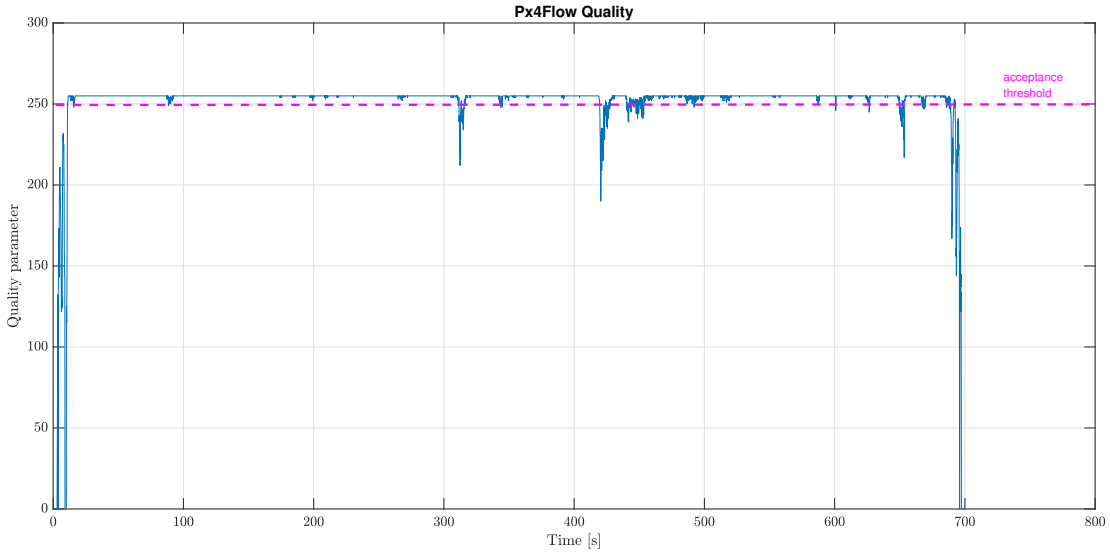
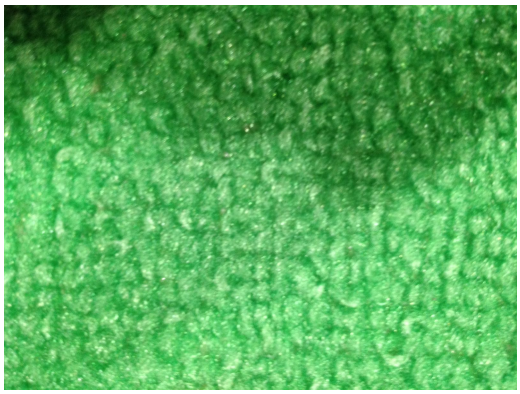


Figure 6.5: Px4Flow quality parameter: outdoor flight

6.2.1.2 Indoor flight

(a) Indoor scenario



(b) Indoor surface

Figure 6.6: Indoor test

In our laboratory, equipped with the security cage specific for indoor flight test, the ground surface is extremely smooth and with a high light reflectivity. This makes the usage of the Px4flow to collect optical flow data impossible. In order to reach better conditions, we have tried several solutions but we have obtained the best visual capability results, using simple coloured carpets. In Figure 6.6 we can appreciate the roughness of the carpet surface that increases the quality parameter for the indoor flight. The quality parameter is shown in Figure 6.7, where we can see that, in this indoor case, the optical flow measurements are less

reliable, so we expect worse results from the filtering. In future works the first thing to do, could be to try to find surface conditions which allow us to have high quality optical flow measurements in an indoor environment, equipped with the necessary facilities to execute flight testing (safety cage, motion capture system).

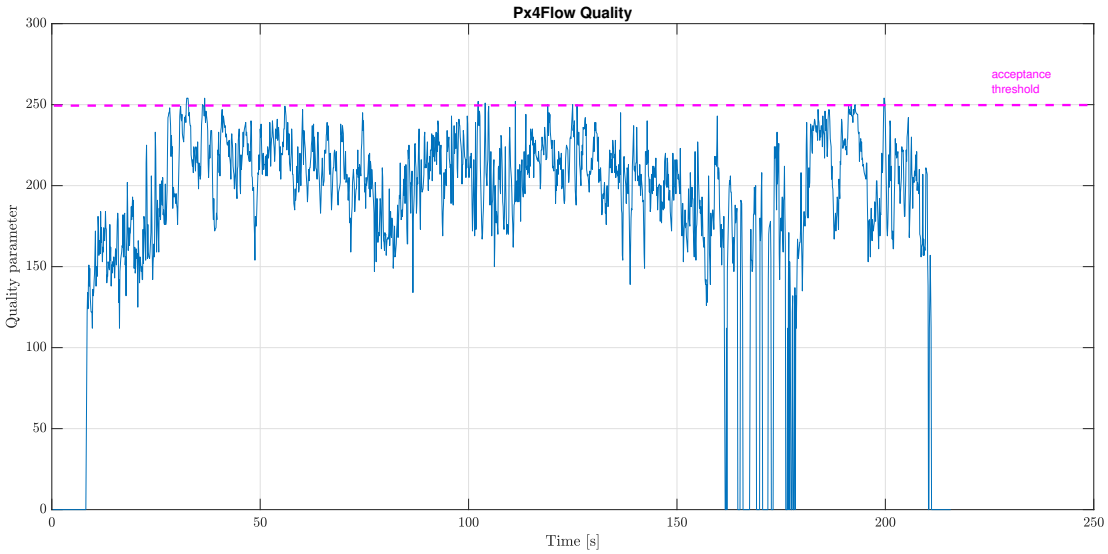


Figure 6.7: Px4Flow quality parameter: indoor flight

6.2.2 GPS based Kalman filter outdoor

The following results are obtained by applying the implemented Kalman filter on the data set collected during the outdoor flight described in 6.2.1.1, by using only the Gps measurements for the correction step.

The Gps based Kalman filter for the estimate of the UAV position and velocity gives good estimates. In particular we can observe that the estimates of position along x (Figure 6.8) and y (Figure 6.9) NED axes, are similar to the Gps measurements but not exactly the same; this means that the filter is reaching an estimate fusing the acceleration measurements (as inputs in the mathematical model) and the GPS measurements. We cannot state if it is an absolutely correct estimate, because the only reference is the output of the Gps , that has its intrinsic errors, but of course the Kalman filter is working very well.

A different behaviour can be observed along the z axis (Figure 6.10). We know that GPS measurements are less reliable along the z direction (altitude), so we expected these results. Nevertheless the estimate could be considered satisfying.

The estimate velocities are compared with the velocity information given by the Gps, but the implemented Kalman filter does not fuse direct measurements of the velocities. In the two comparisons, Figure 6.11 for the x velocity and Figure 6.12 for the y velocity, we can observe a coherent behaviour between the

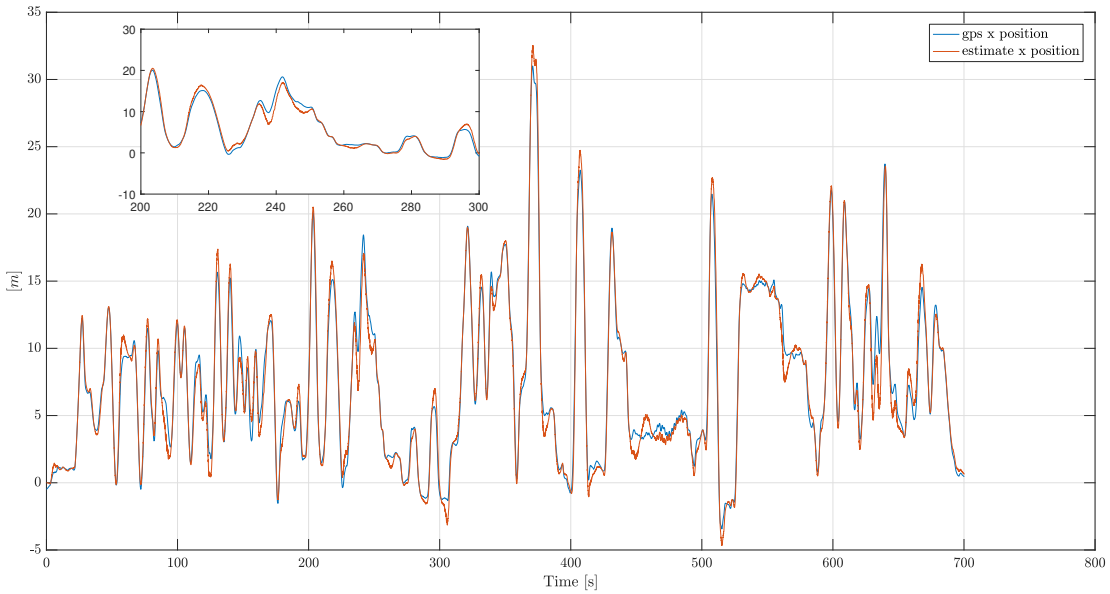


Figure 6.8: Comparison: estimated x-position and GPS x-position

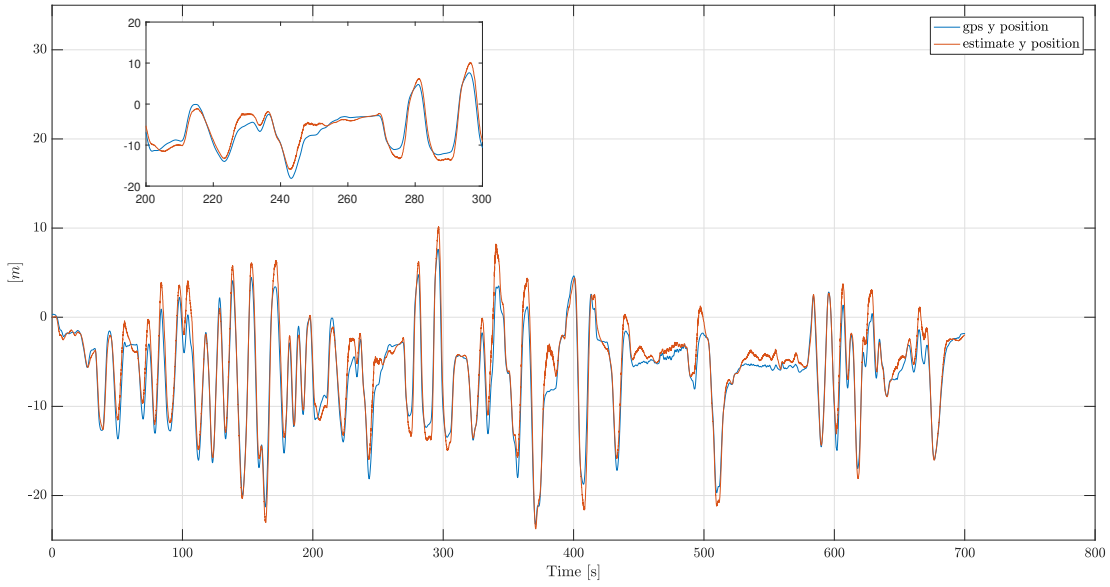


Figure 6.9: Comparison: estimated y-position and GPS y-position

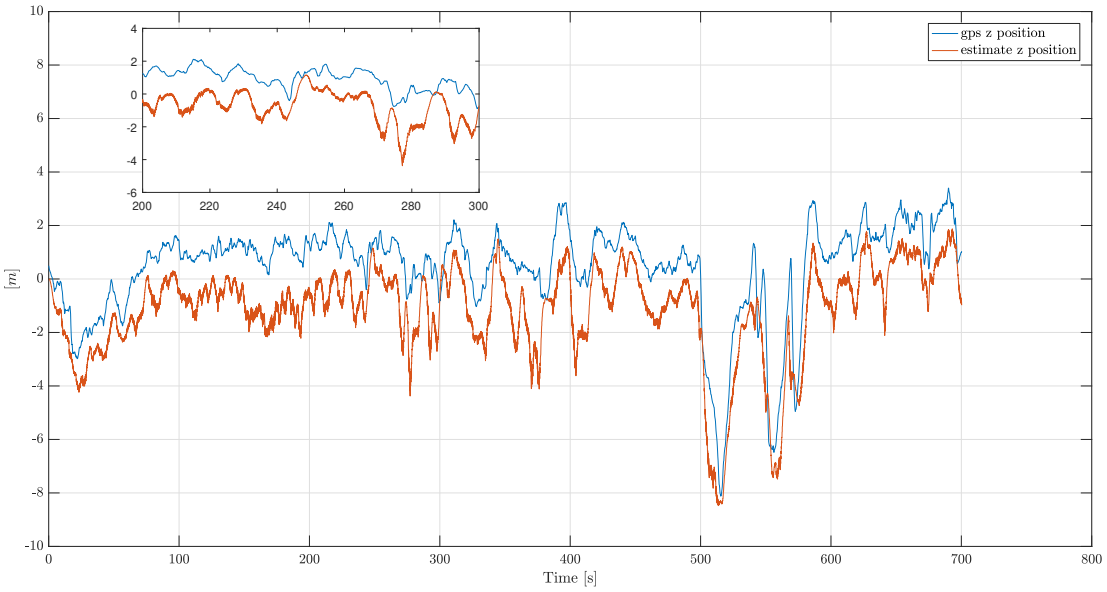


Figure 6.10: Comparison: estimated z-position and GPS z-position

estimates and the references. About the z axis, as expected, the gap between the GPS measurement and the estimate became bigger (Figure 6.13).

A quantity that could show better if the Kalman filter is working in a good way, is the estimation error (also called innovation), that, as we can see in eq.(3.10), multiplied for the Kalman gain, gives us the correction of the state vector. The right behaviour of the innovation that we want to observe consist of:

- a white noise, zero mean value process, that suggests the filter is working right (see [12] for the proof) ;
- a trend that stay bounded in the plus-minus three standard deviation ($\pm 3std$) band in order to ensure the normal distribution of the innovations. The standard deviation has been calculated from the values of the P state covariance matrix.

For this case, the application of the Kalman filter on the available data set, gives us the innovation results in Figure 6.14 . Except for a few moments, we can observe the innovations staying within the boundaries, and these results underline a good tuning of the Kalman filter. Another results that gives us a positive feedback of the good working of the GPS based Kalman filter in this outdoor flight, is the trend of the power spectral density (PSD) of the innovations. What we expect from a good filter, is to get an innovation PSD with lower value than the PSD of the available measurements. This behaviour is observed in the results in Figure 6.15 as further evidence of the a good working Kalman filter.

An observation must be done about the peaks which can be seen in the innovations PSD behaviour at high frequency.

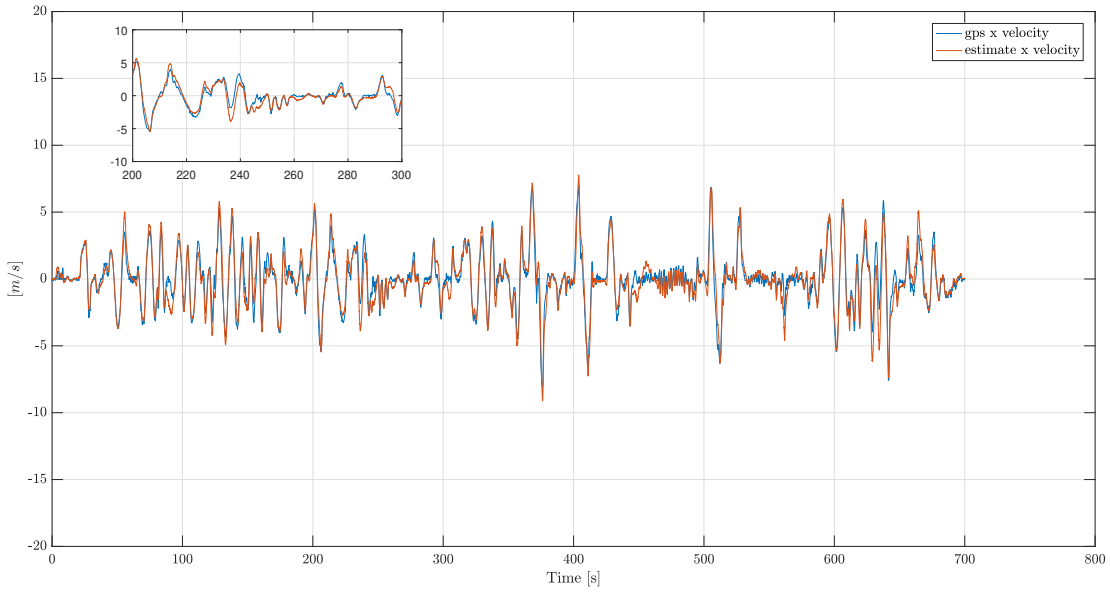


Figure 6.11: Comparison: estimated x-velocity and GPS x-velocity

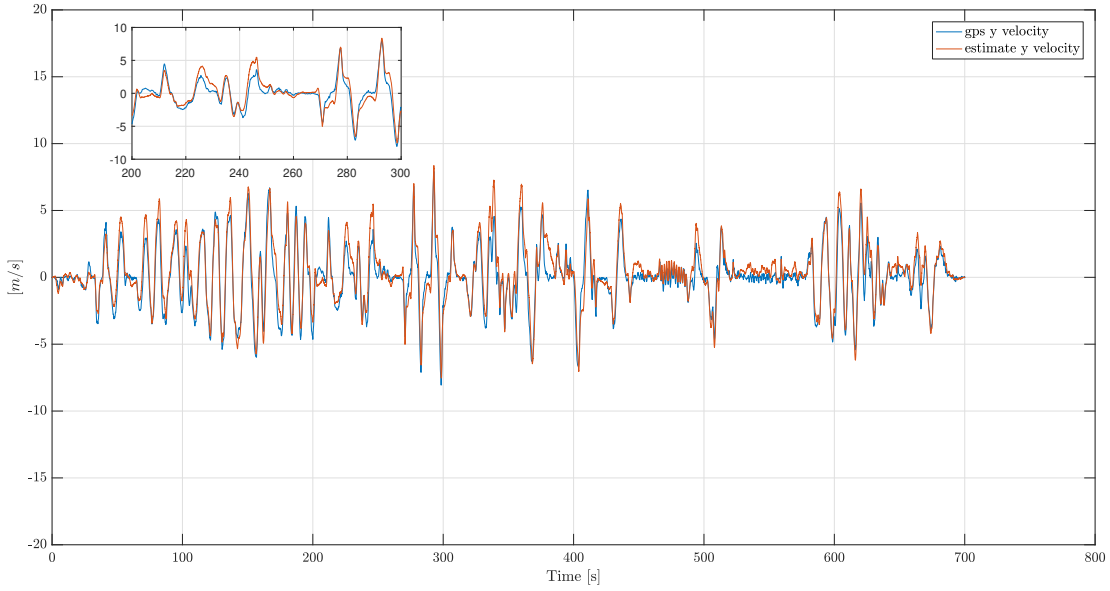


Figure 6.12: Comparison: estimated y-velocity and GPS y-velocity

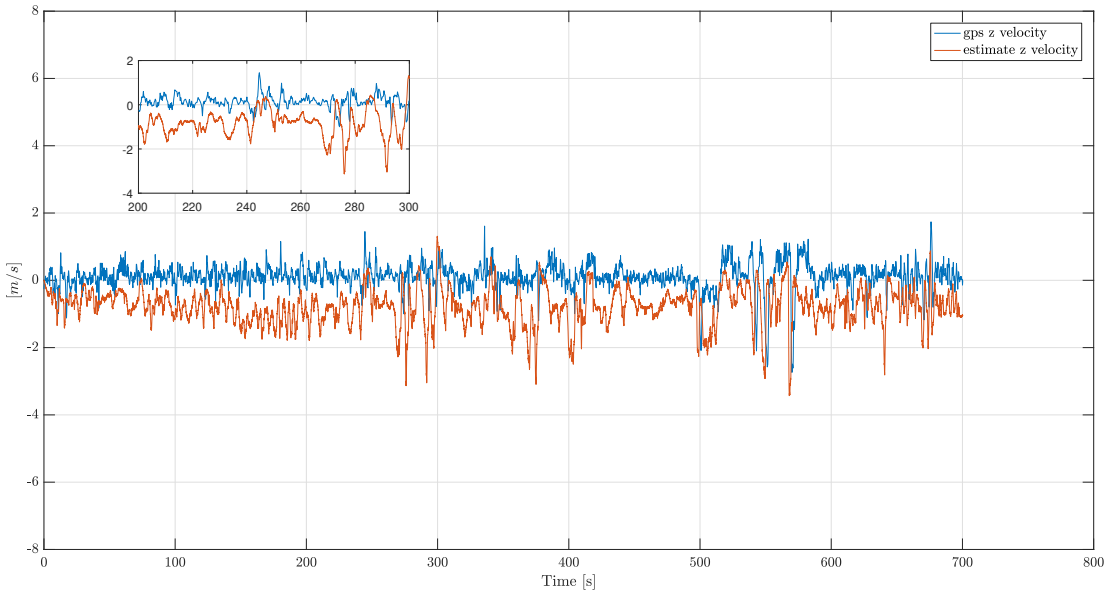


Figure 6.13: Comparison: estimated z-velocity and GPS z-velocity

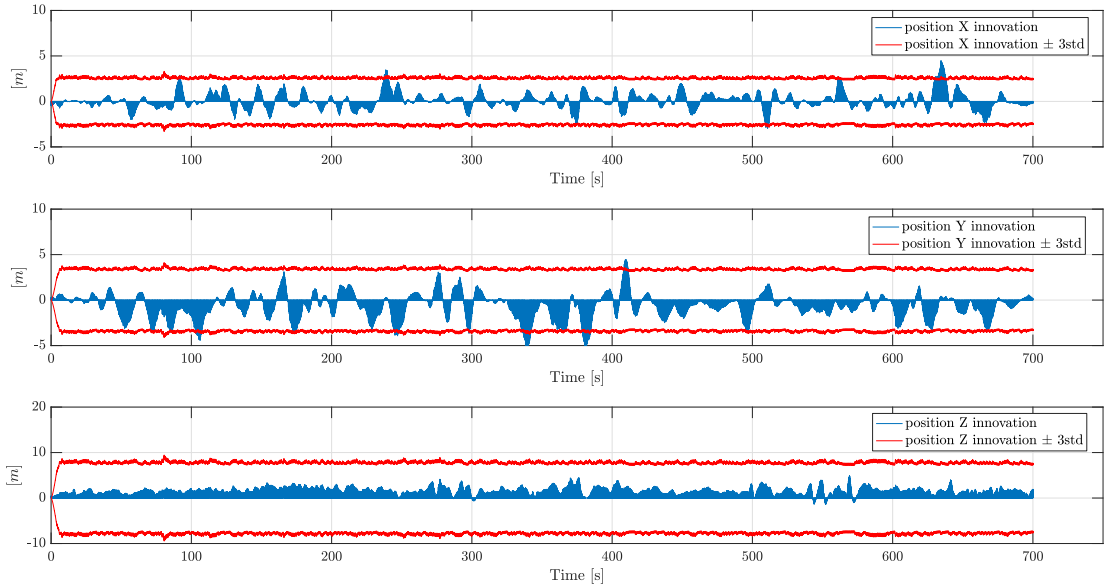


Figure 6.14: State estimate innovations

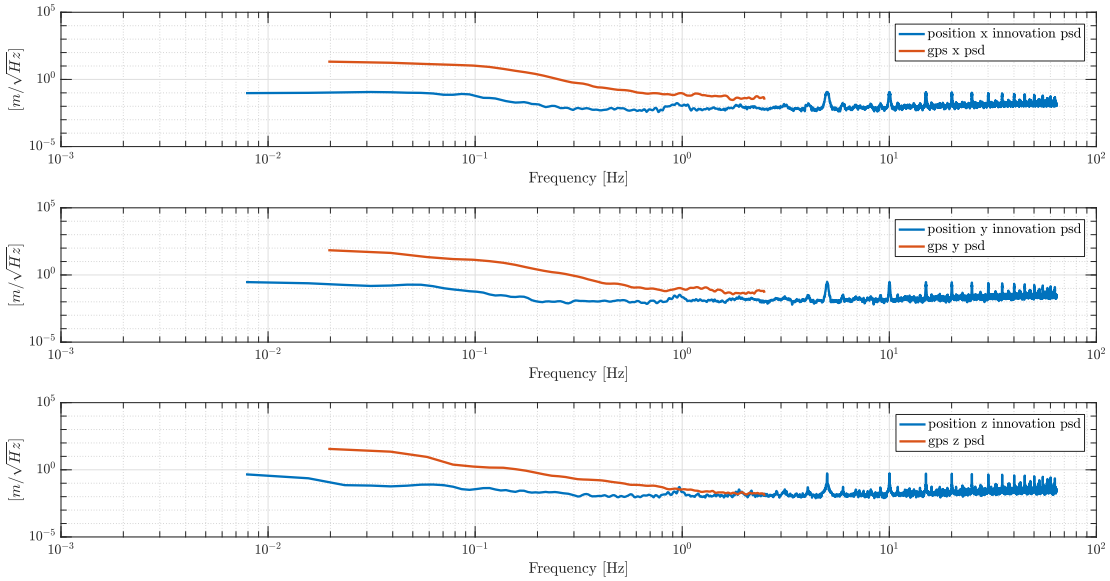


Figure 6.15: Comparison: Innovations PSD and GPS measurements PSD

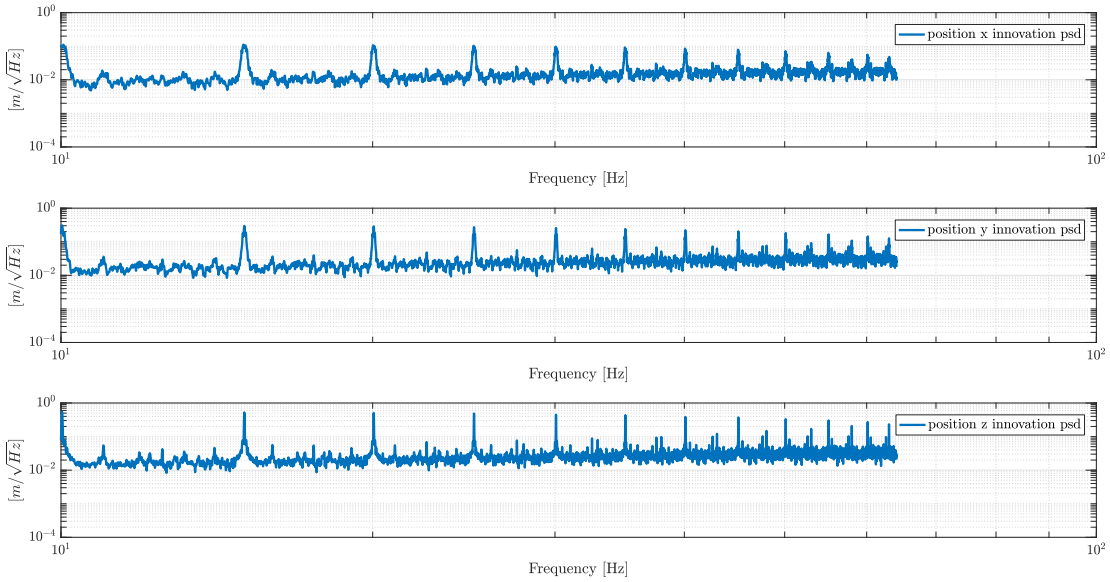


Figure 6.16: Innovations PSD at high frequency

In Figure 6.16 the plot is focused on high values of frequency and show us an harmonic presence of these peaks, about every 5 Hz. This is a typical phenomenon of a multi-rate Kalman filter. In an ideal case where all the involved measurements have the same rate of the Kalman filter, the innovation PSD at high frequency will converge to a value. In our case the measurements y are not always available, in particular in this case the innovation for the correction step is executed thanks to the GPS measurements which are sampled at $5Hz$. This explains the periodicity of the peaks that appear after the $5Hz$ that we can observe in Figure 6.16.

At last we report in Figure 6.17 the Kalman gain components for every time instant, and it is possible to see all the components of the Kalman gain matrix converge quickly around a constant value.

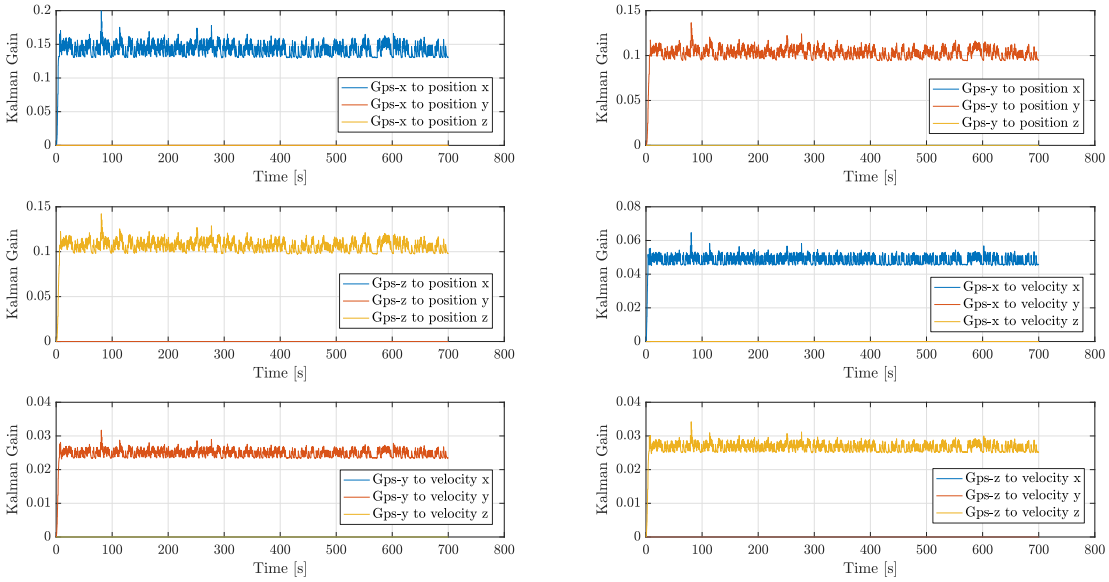


Figure 6.17: Kalman gains due to GPS measurements

In view of this results we can state that the implemented Kalman filter gives a reliable estimate of the position and the velocity of the UAV, during an outdoor flight when it is based on Gps measurements. The addition of another sensor to improve the estimate results along the z axis, could be necessary.

6.2.3 Optical flow based Kalman filter outdoor

In this paragraph we present the results obtained applying the Kalman filter to the data set collected during the outdoor flight described in 6.2.1.1, by using only the optical flow measurements for the correction step. The optical flow measurements allow us to have indirect measurements of the velocity, as stated in equation (2.9) and equation (2.10), so the position cannot be estimated in a right way, being obtained by integrating the velocity with no direct measurements which could

correct it. Indeed it is clear that the position states are not observable from the flow measurements. For this reason we must evaluate the results of this scenario by looking only at the estimates velocities. Of course there is not observability from the optical flow measurements also for the velocity along the z axis, so its estimate is not reliable as well.

In Figure 6.18 the estimated velocity along the x NED axis is compared to the NED velocity in x obtained from the GPS measurements. It is important to underline that the data sets supplied by the GPS are used only as a reference but they are not involved in the estimation. It is possible to observe an error between the two plotted velocities, that in some time interval could become higher, as represented in the zoom box on Figure 6.18 , but for most of the flight it is very small.

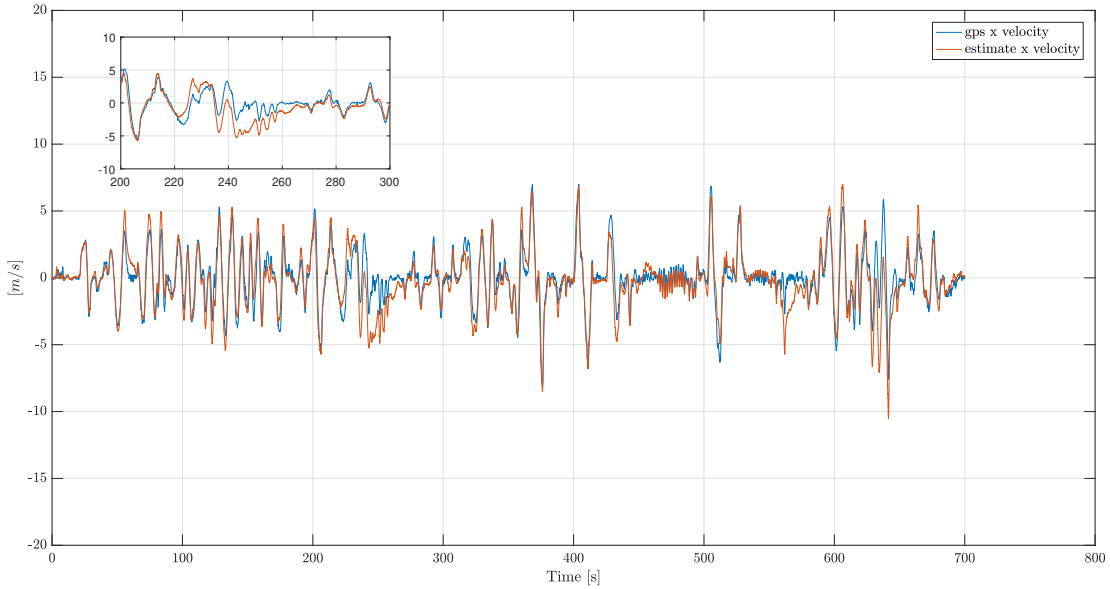


Figure 6.18: Comparison: estimated x-velocity and GPS x-velocity

The same considerations can be done for the velocity along the y NED axis, by looking at the Figure 6.19.

In Figure 6.20 the innovations about the velocities in x and y are shown. In this case it seems not to be a completely zero mean process but it always stays within the $\pm$ 3std boundaries. In future works could be possible to take advantages of this big margin between the boundaries and the innovations, pushing up the Kalman filter performances by acting on the tuning.

In the Figure 6.21 we can appreciate the PSD of the innovation compared with the PSD of the optical flow measurements. It is possible to state that the innovation PSD values are lower than the optical flow measurements PSD values, and this is an other indicator of the good working of the filter. Zooming on the PSD related to high frequencies (Figure 6.22) we can observe the same PSD

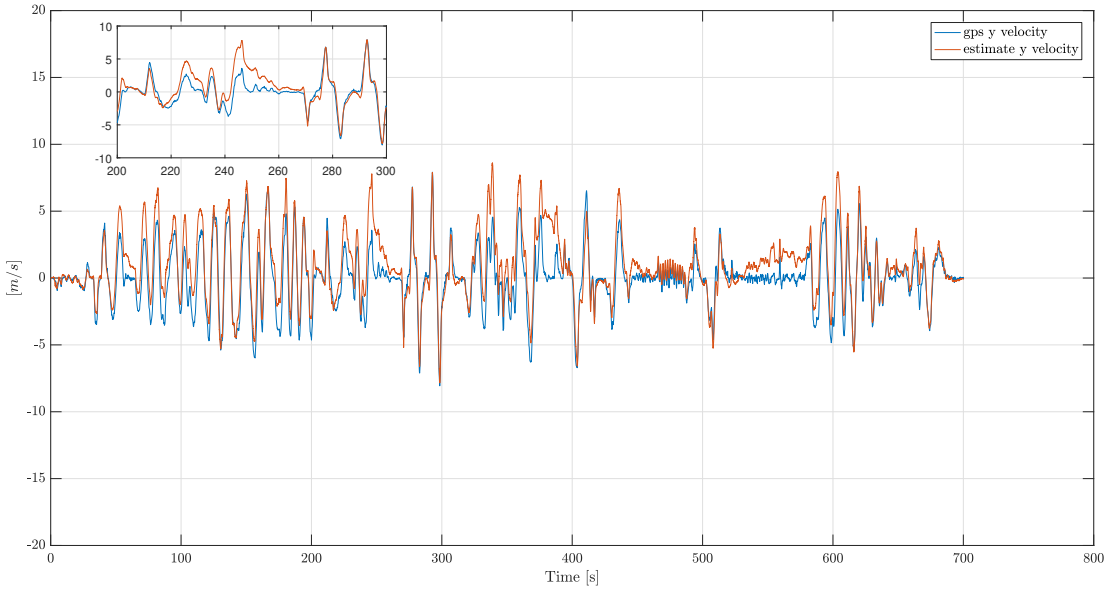


Figure 6.19: Comparison: estimated y-velocity and GPS y-velocity

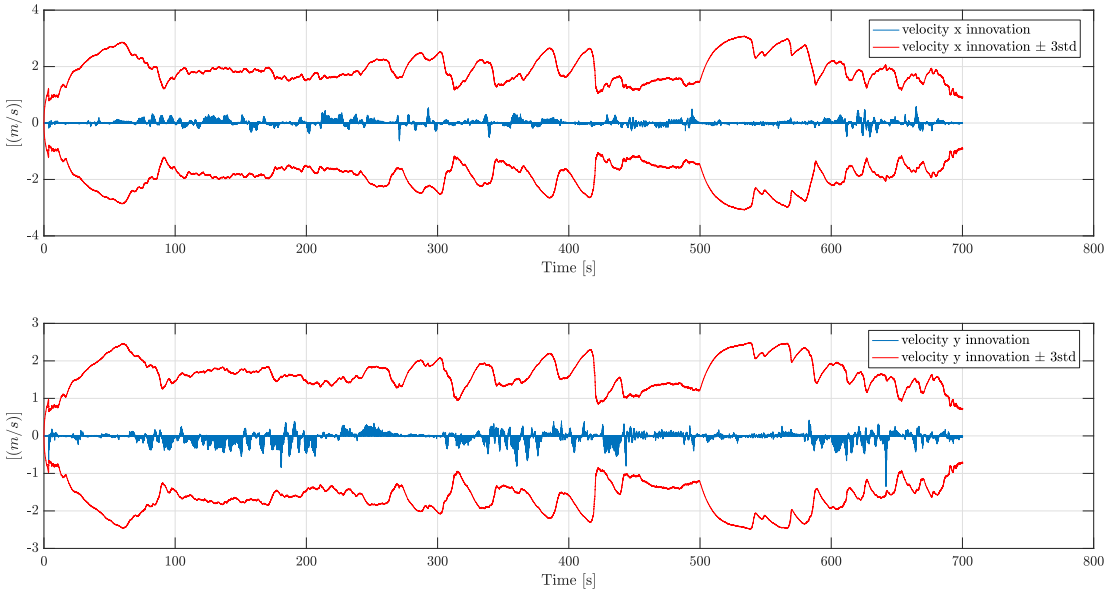


Figure 6.20: State estimate innovations

behaviour discussed in the results of GPS based Kalman filter, in paragraph 6.2.2 . In this case the periodic trend of the peaks is given by the optical flow sample frequency that is equal to $10Hz$.

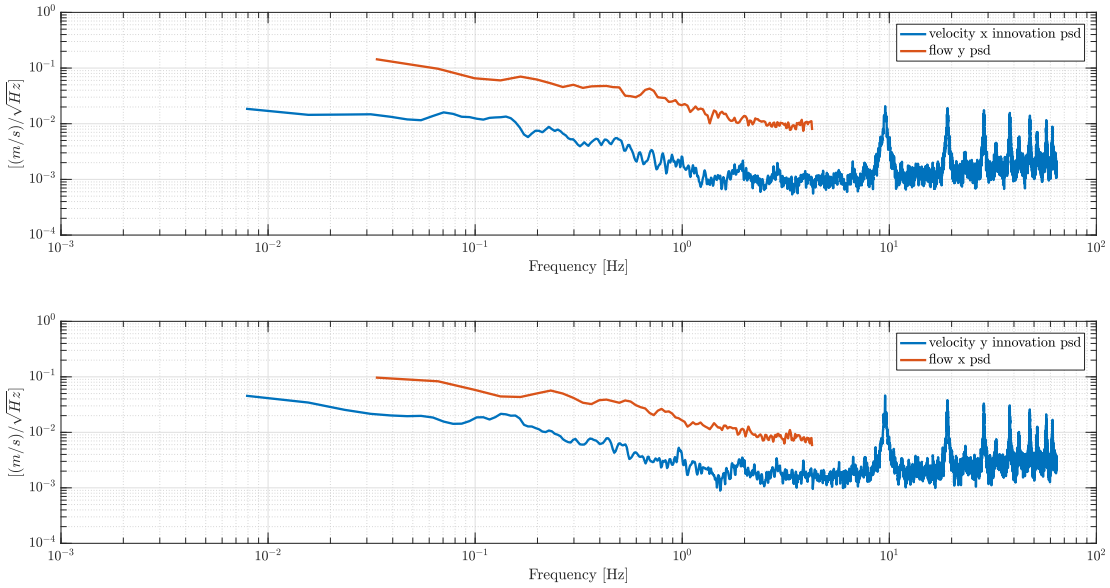


Figure 6.21: Comparison: Innovations PSD and optic flow measurements PSD

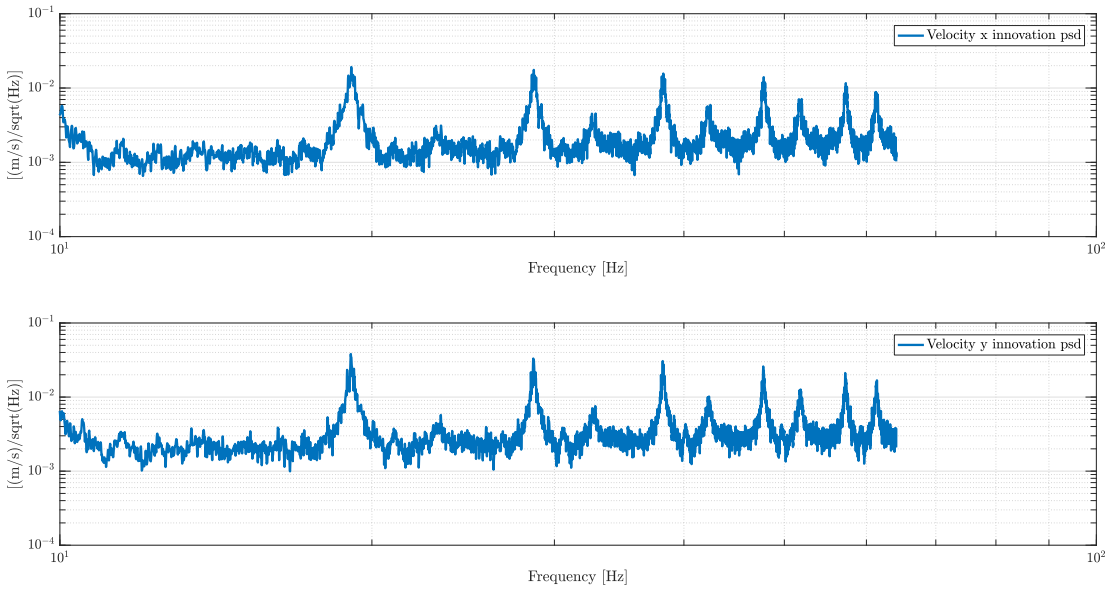


Figure 6.22: Innovations PSD at high frequency

In Figure 6.23 the Kalman gain from the optical flow correction block, are shown. Their strange behaviour is due to the attitude matrix of this flight; the

attitude is involved through the H matrix (Eq.(4.1)) that, in turn, affects the Kalman gain calculate as we can see in Eq. (3.14).

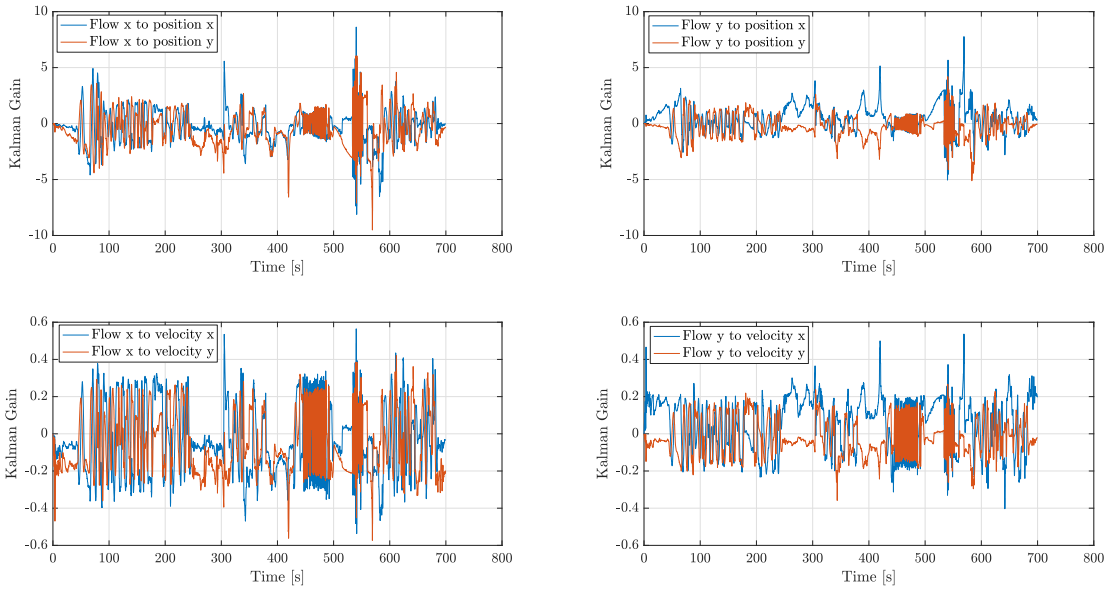


Figure 6.23: Kalman gains due to optical flow measurements

In view of the above discussion, we can state that the optical flow based Kalman filter for an outdoor flight with good visibility conditions, can give reliable estimate velocities in x and y NED axis.

6.2.4 GPS and optical flow based Kalman filter outdoor

The below results are obtained by applying the implemented Kalman filter on the data set logged during the outdoor flight described in 6.2.1.1, using both GPS and optical flow measurements for the correction step. This represents the probable scenario for an application of this technology: the UAV able to rely both on the GPS and optical flow sensor to execute its autonomous flight.

In order to have a comparison to evaluate the estimate results, the GPS measurements are used as a reference, but it is good to underlined that the correction step is implemented to use the GPS position measurements and the optical flow measurements.

The Figure 6.24 shows the estimate position along the x axis compared with the GPS information. The Figure 6.25 does the same for the estimate position in y axis.

Of course the GPS reference cannot bee seen as the perfect one, being a measurement, thus, not completely correct; but this comparison give us the idea that the filter is working good.

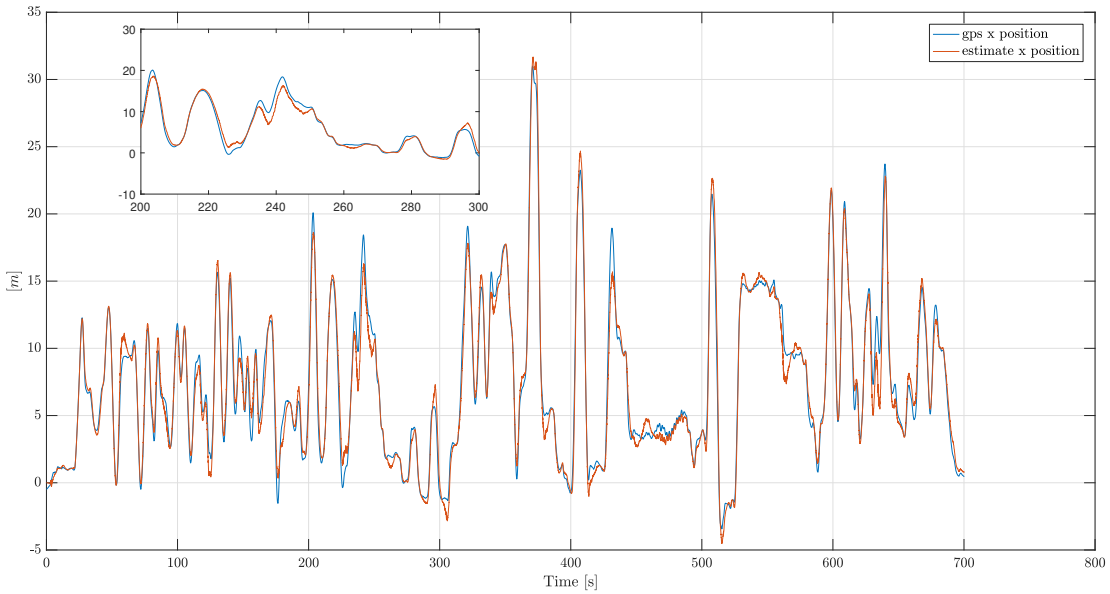


Figure 6.24: Comparison: estimated x-position and GPS x-position

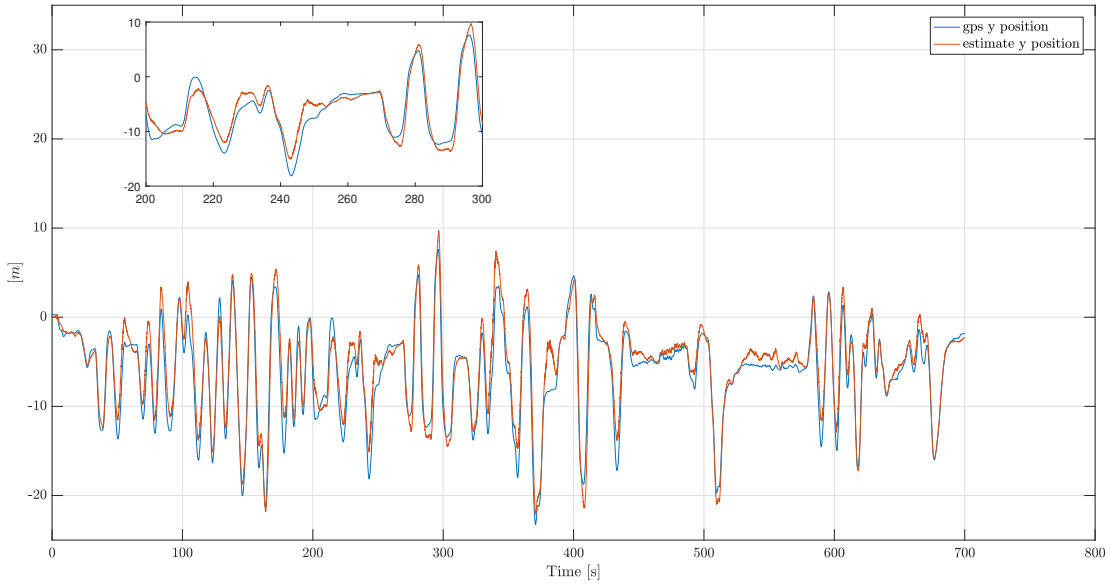


Figure 6.25: Comparison: estimated y-position and GPS y-position

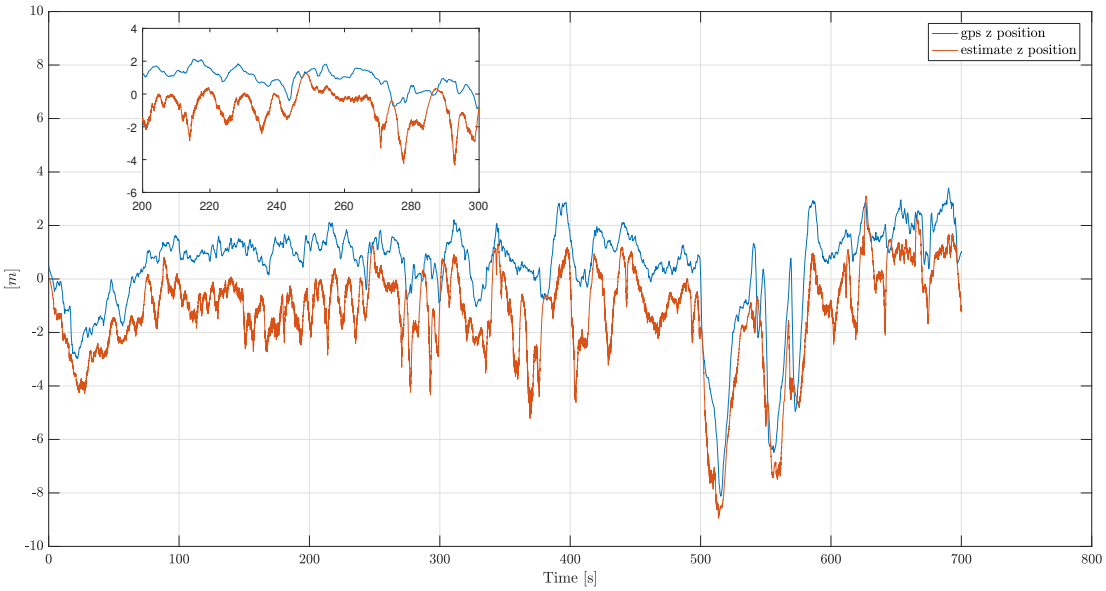


Figure 6.26: Comparison: estimated z-position and GPS z-position

The Figure 6.26 shows what we expect about the z axis; the error between the estimate z position and the GPS measurement is significant but it is well known that the worse performances of the GPS are along the z axis.

Using both the sensor, the estimate results about the velocities, are good, as it is possible to observe in Figure 6.27 for the x axis and in Figure 6.28 for the y axis.

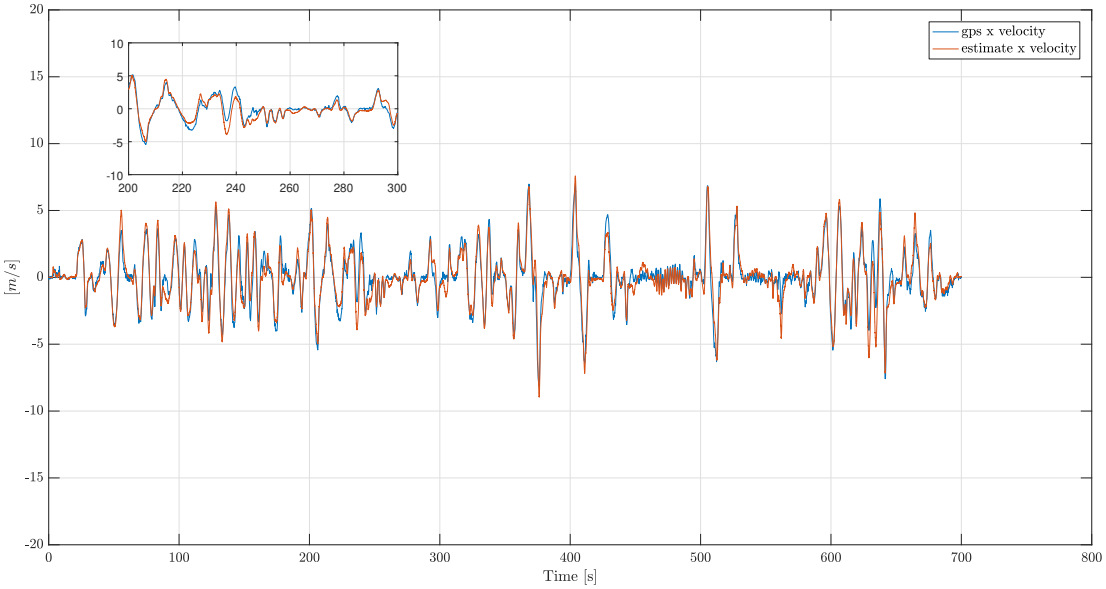


Figure 6.27: Comparison: estimated $x$velocity and GPS x-velocity

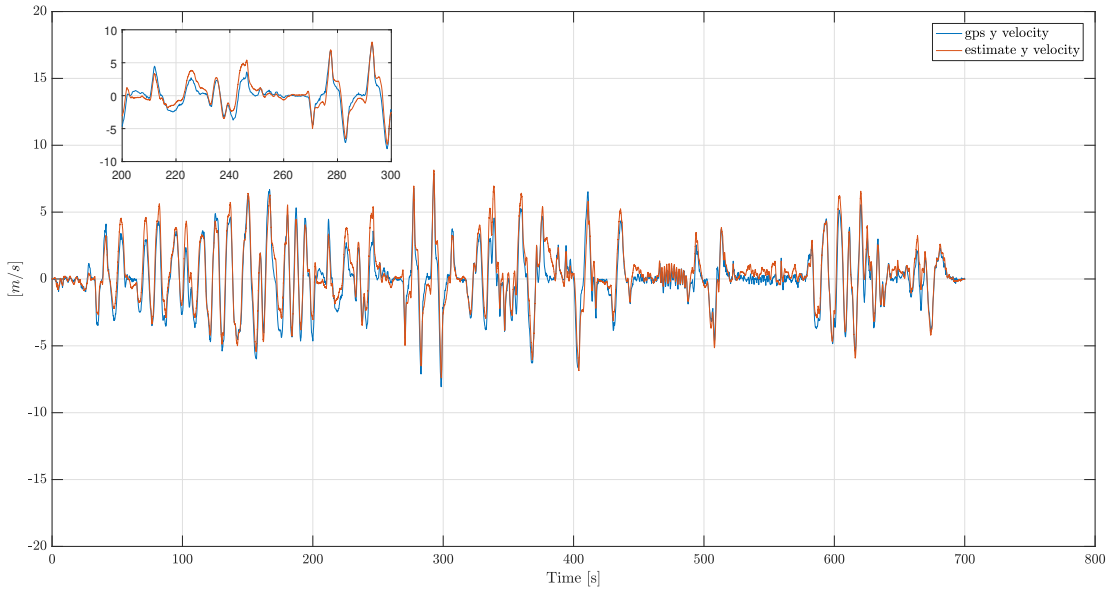


Figure 6.28: Comparison: estimated y-velocity and GPS y-velocity

The bad behaviour along the z axis is underlined in Figure 6.29 where the estimate results about the z velocity are compared with the GPS measurements.

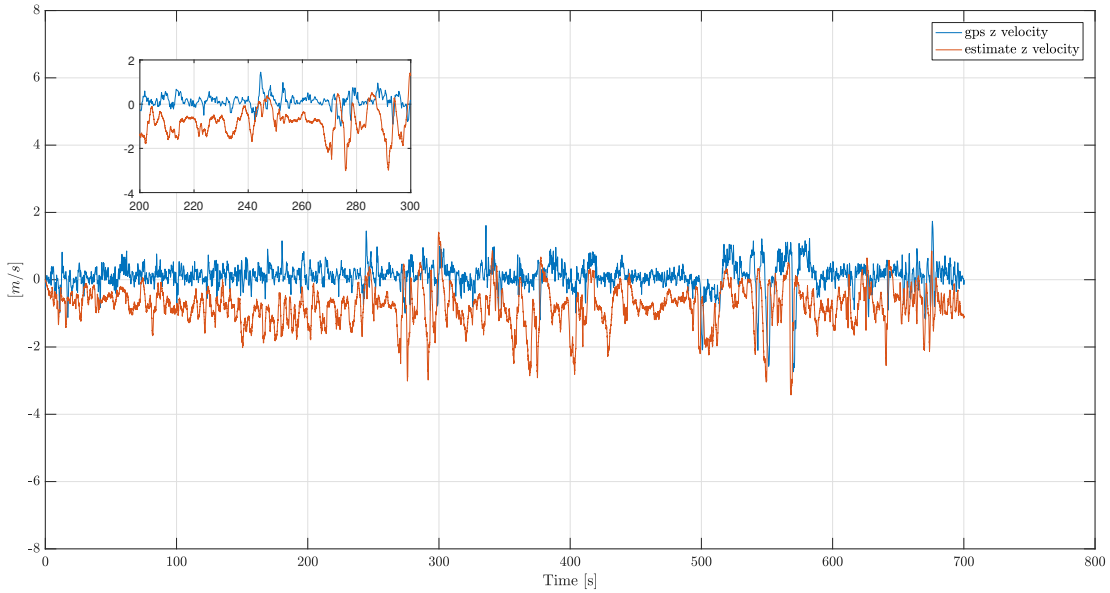


Figure 6.29: Comparison: estimated z-velocity and Gps z-velocity

In order to ensure that the Kalman filter is working right, the innovation behaviour must be observed. In Figure 6.30 the innovations about the position state variables are represented with the boundaries composed by the $\pm 3std$ band. As we can see, the estimate innovations stay within the boundaries for most of the

time of the flight, certifying the good tuning of the filter. The same is represented in Figure 6.31 about the velocity state estimate innovations. In this case we can see a large margin to be used in order to increase the filter performance.

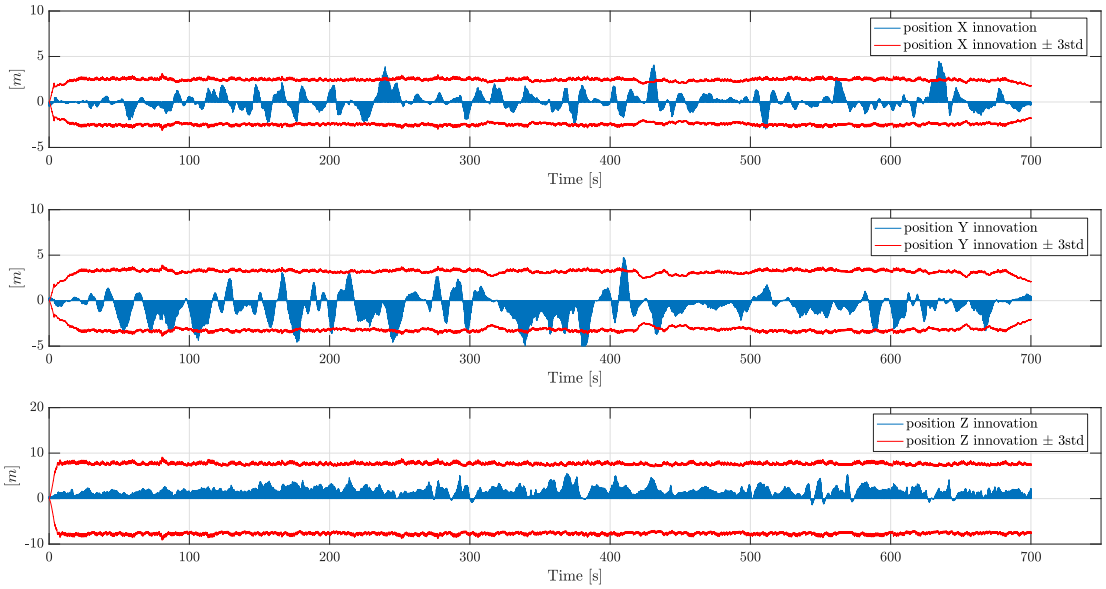


Figure 6.30: Position state estimate innovations

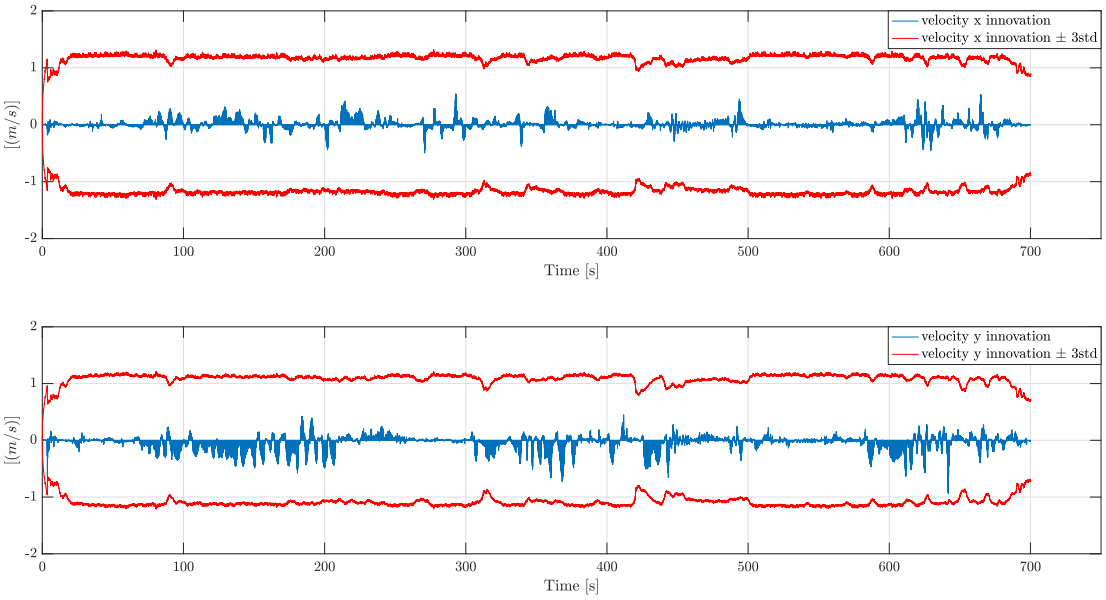


Figure 6.31: Velocity state estimate innovations

In Figure 6.32 we can observe and analyse the innovations from the PSD point of view. This graph shows an innovation PSD lower than the GPS measurements

PSD for all the three components of the position. This means the Kalman filter is doing its work. The same behaviour can be observed for the velocity innovations PSD, in Figure 6.33, with respect to the optical flow measurements PSD.

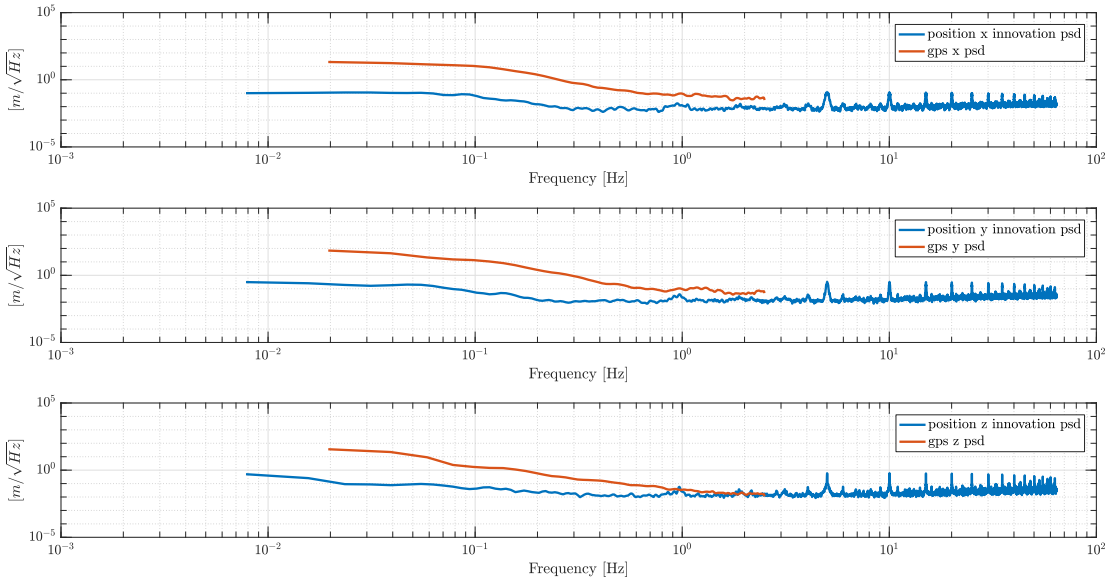


Figure 6.32: Comparison: Innovations PSD and GPS measurements PSD

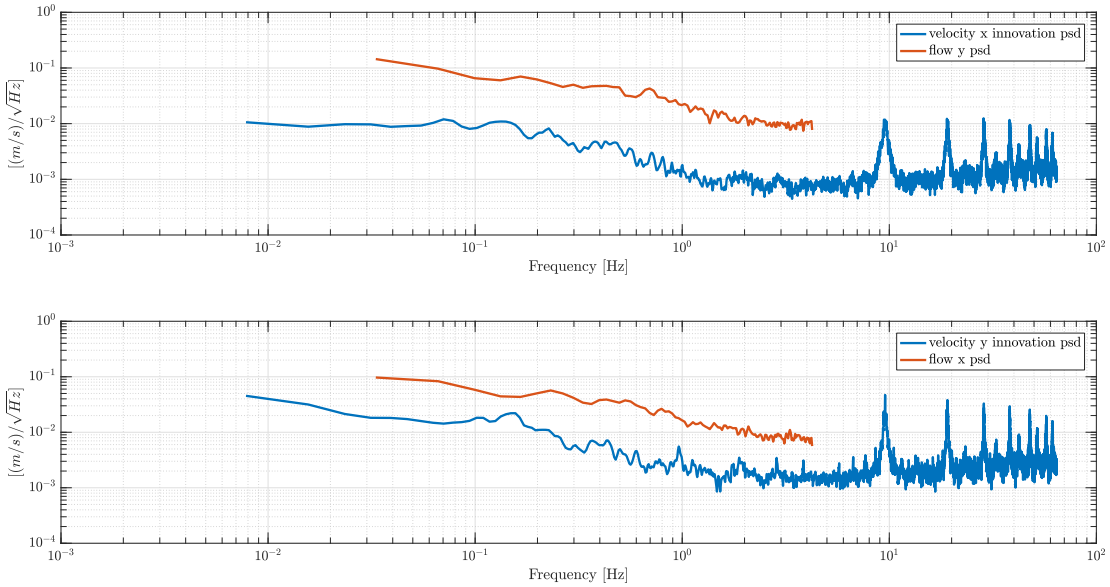


Figure 6.33: Comparison: Innovations PSD and optic flow measurements PSD

The typical multi-rate behaviour can be observed, focusing on the innovation PSD at high frequencies. The Figure 6.34 shows what happen to the position

innovations PSD at high frequency; as we can see, the trend of the PSD is characterized by periodical big peaks, consequence of the GPS sampling frequency equal to $5Hz$, but also other small peaks can bee observed and they are consequences of the combination of the effects given by the sampling time of both the sensors (GPS and optical flow sensor). This effects are visible also in Figure 6.35, where the velocity innovations PSD at high frequency are represented.

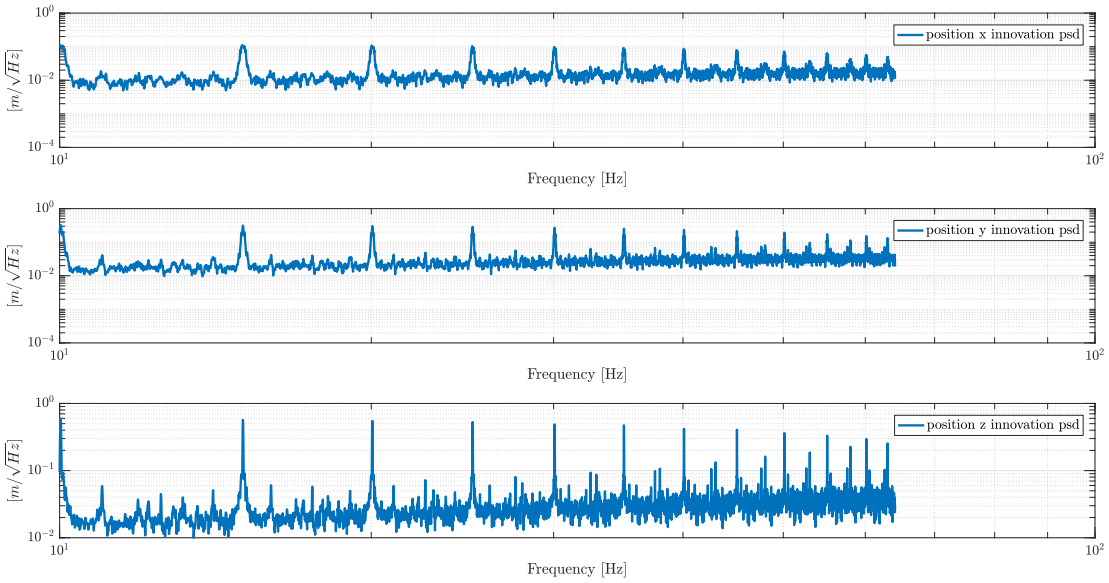


Figure 6.34: Position innovations PSD at high frequency

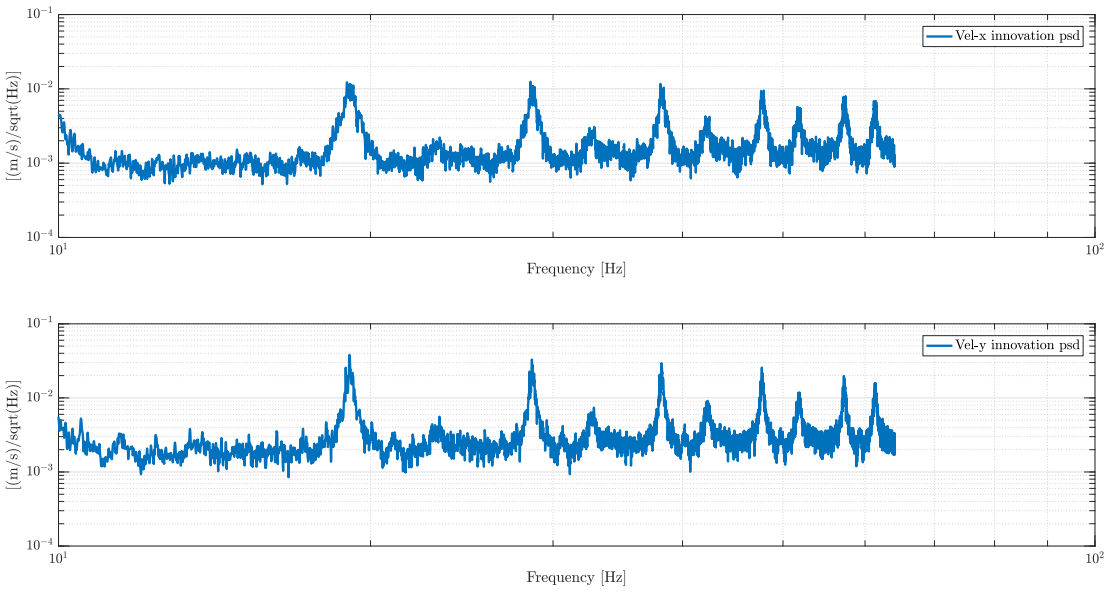


Figure 6.35: Velocity innovations PSD at high frequency

At the end of this paragraph we focus our attention on the Kalman gain contributes given by the two block of used measurements (GPS positions and optical flow). In Figure 6.36 the Kalman gains due to the GPS measurements are shown. In Figure 6.37 we can appreciate the Kalman gain due to the optical flow based correction step.

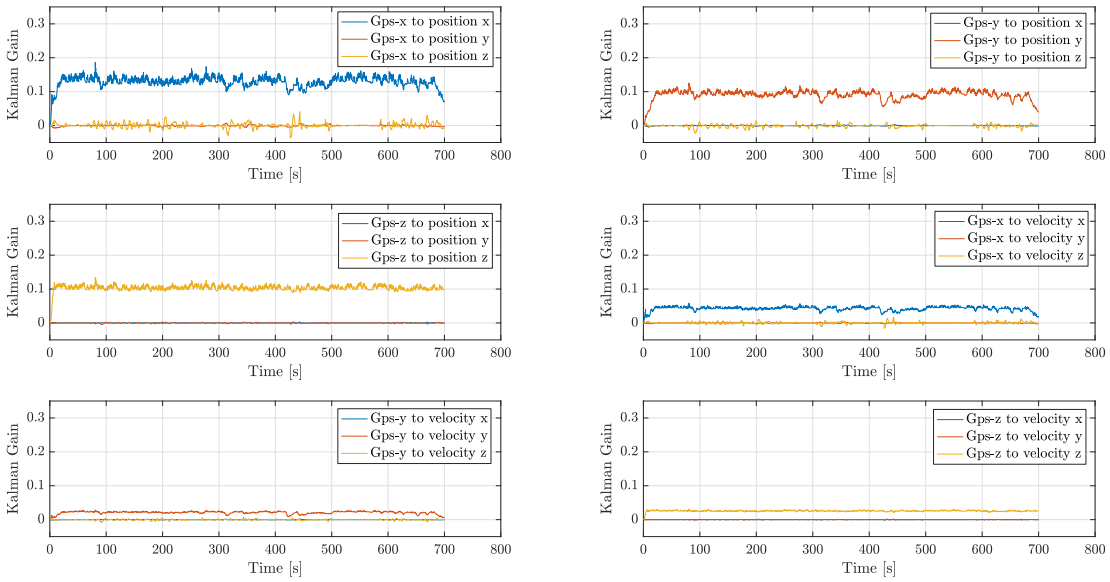


Figure 6.36: Kalman gains due to GPS measurements

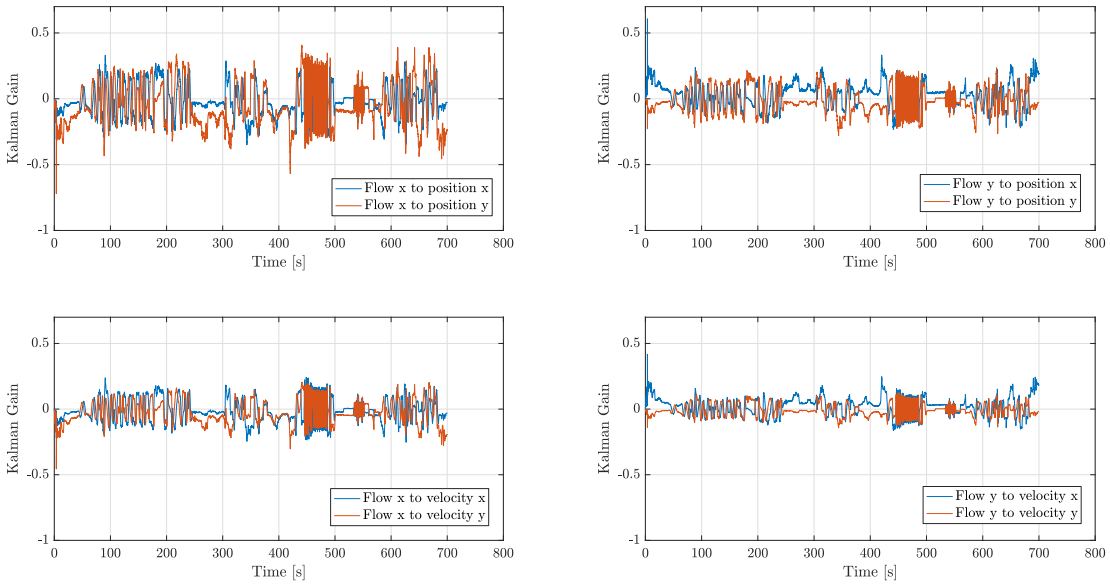


Figure 6.37: Kalman gains due to optic flow measurements

In view of the above discussion, we can state that an optical flow and GPS

based Kalman filter for an outdoor flight with good visibility conditions, can give reliable estimate position and velocity in x and y NED axes while along the z one could be necessary an other kind of sensor to improve the estimate results.

6.2.5 Results comparison: outdoor flight

In this paragraph we want to give a more complete understanding of the estimation results, by comparing the estimates obtained in the different cases discussed above. The comparison between the Kalman filter results when it uses different kind of data sources, is done on the outdoor flight data sets.

The Figure 6.38 shows the comparison between: the estimate position in x when the filter uses both the GPS and the optical flow measurements, and the estimate position in x when the filter uses only the GPS data for the correction step. The results are very similar and the data set is very long, so it is difficult to see any differences. For this reason in Figure 6.39 the same comparison is represented in a 40 seconds interval of the flight, adding the x position from GPS raw data as a reference. It is possible to observe that the two estimates are different. The magnitude of this difference is readable from the Figure 6.40, where the error between the two estimates is represented; this error is not negligible.

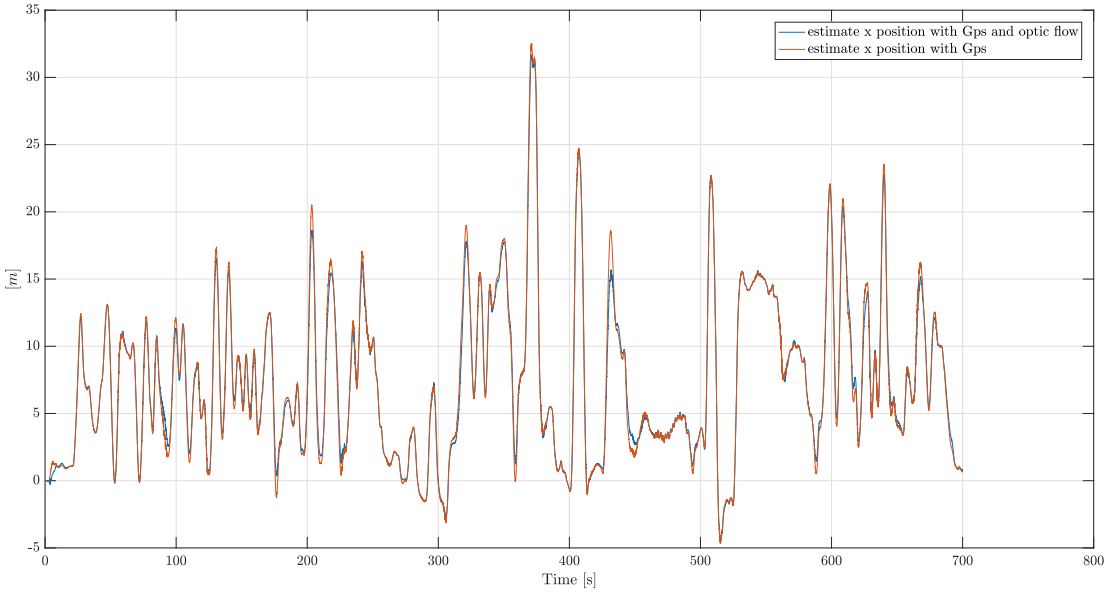


Figure 6.38: Compared results: position in x

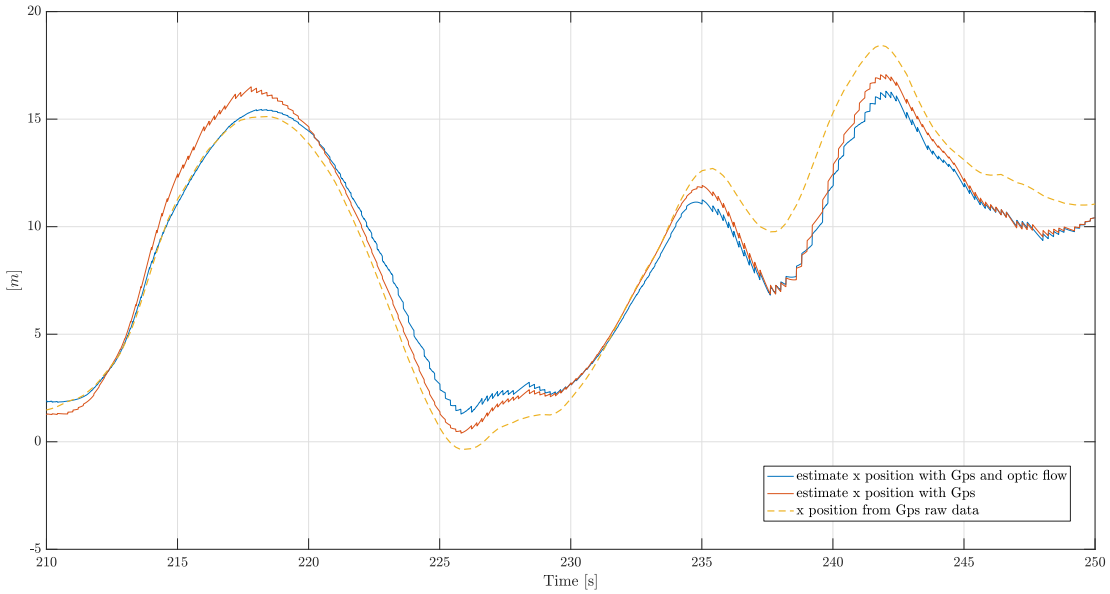


Figure 6.39: Compared results (zoom): position in x

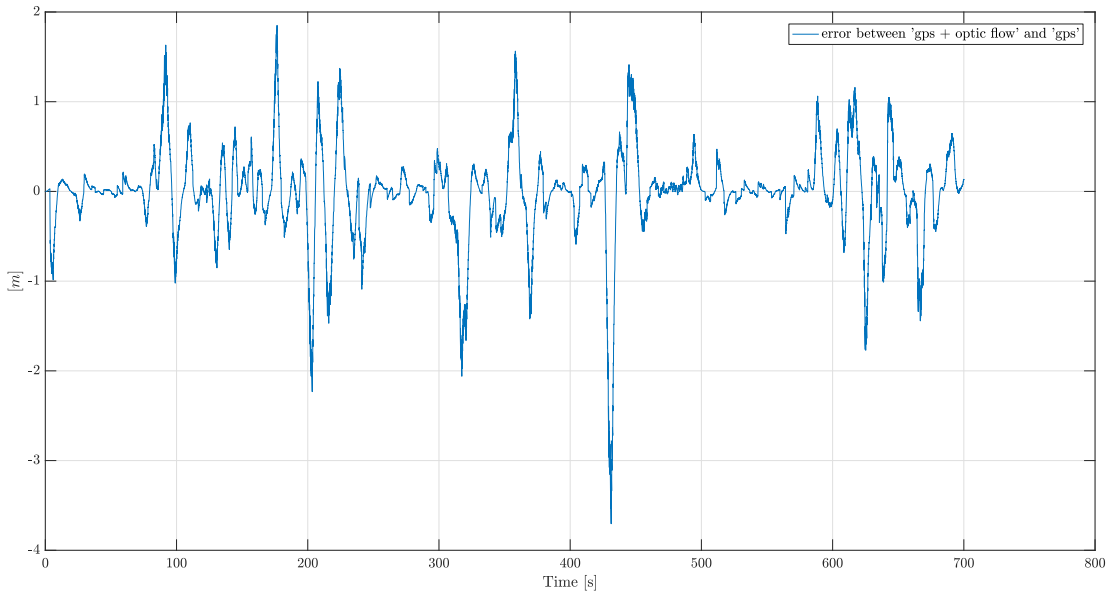


Figure 6.40: Position-x error between the estimate by using GPS + optical flow and GPS only

The Figure 6.41 shows the comparison between: the estimate position in y when the filter uses both the GPS and the optical flow measurements, and the estimate position in y when the filter uses only the GPS data for the correction step. The results are similar. In Figure 6.42 the two estimates are represented in a 40 seconds interval of the whole flight. In this figure we can appreciate

the differences between the two estimates and how far they are from the GPS measurements of the position in y. In order to see how much the optical flow affects the estimate position with respect to the case when only the GPS is used for the estimation, we represent the error between the two estimates in Figure 6.43.

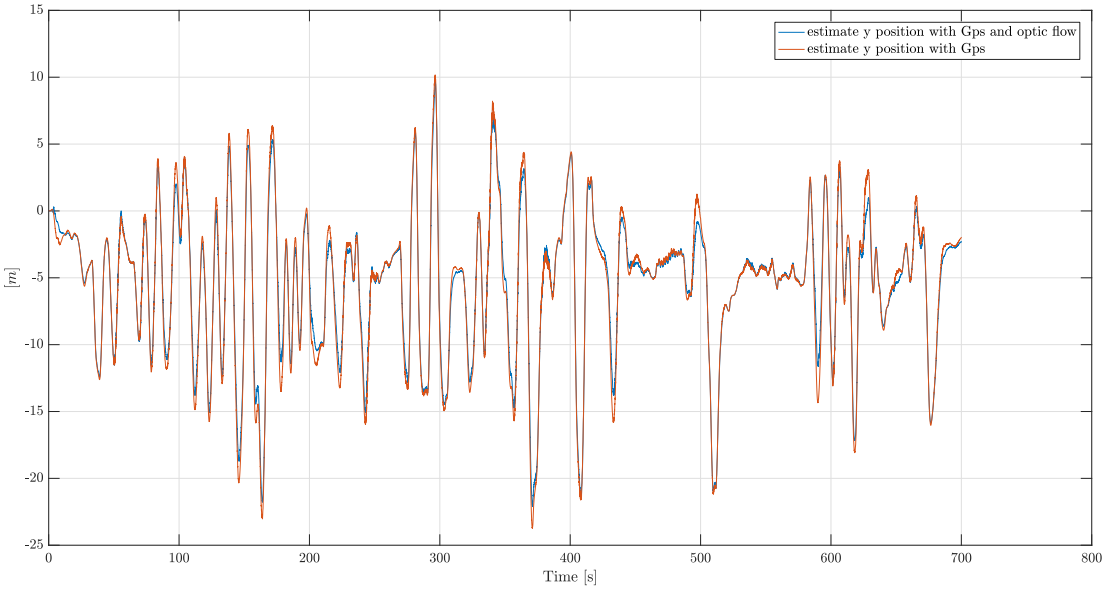


Figure 6.41: Compared results: position in y

The Figure 6.44 shows the comparison between: the estimate position in z when the filter uses both the GPS and the optical flow measurements, and the estimate positions in z when the filter uses only the GPS data for the correction step. The Figure 6.45 shows that both the estimates are far from the GPS raw data, but we already stated that the performances of our Kalman filter decrease along the z axis. In Figure 6.46 we can appreciate the error between the two estimates along the z axis.

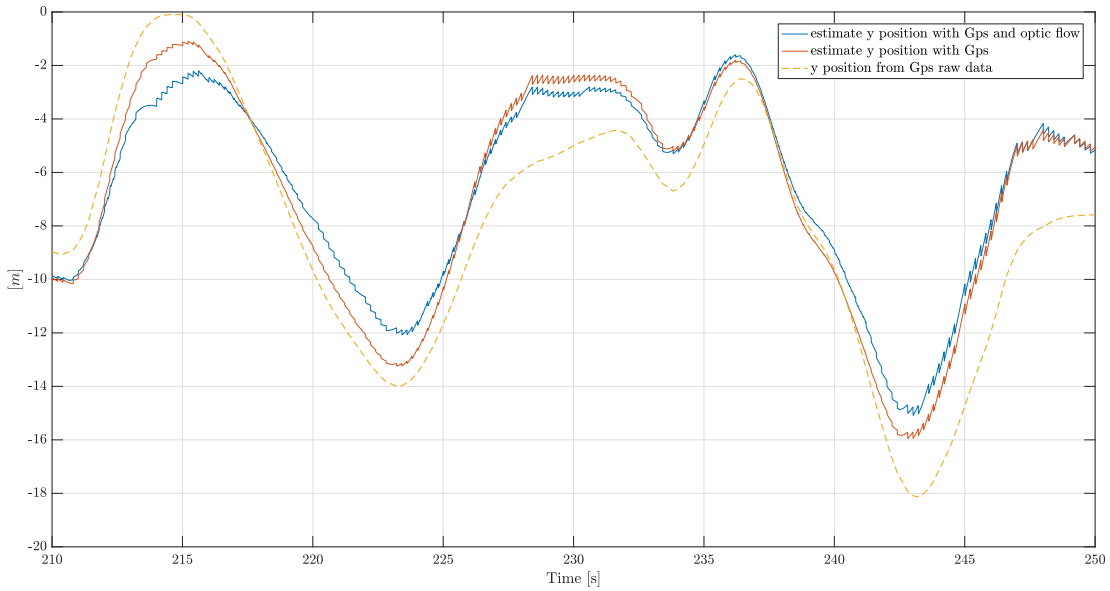


Figure 6.42: Compared results (zoom): position in y

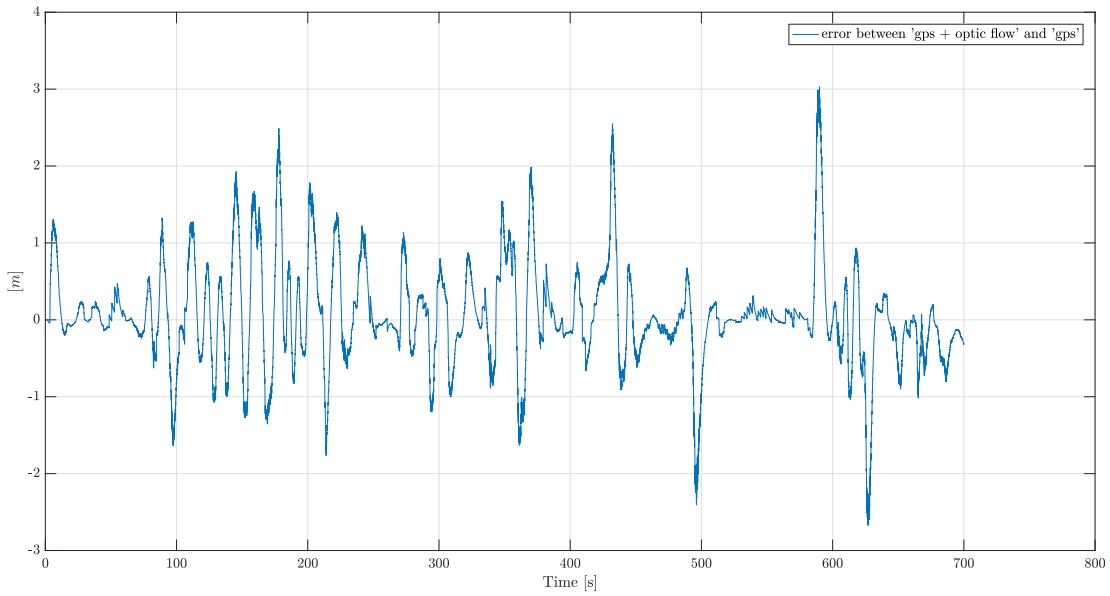


Figure 6.43: Position-y error between the estimate by using GPS + optical flow and GPS only

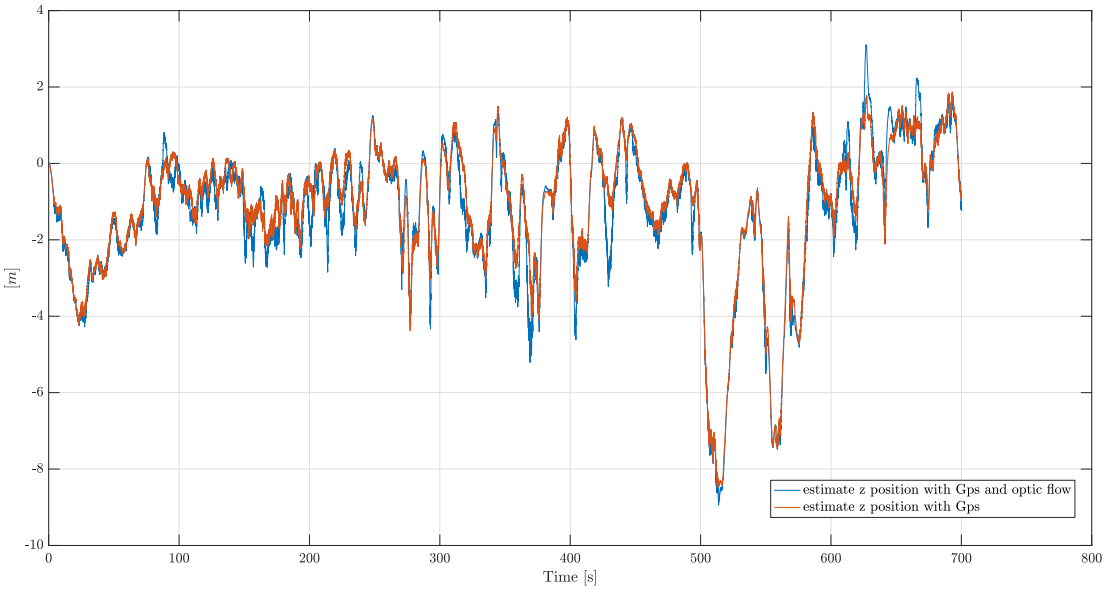


Figure 6.44: Compared results: position in z

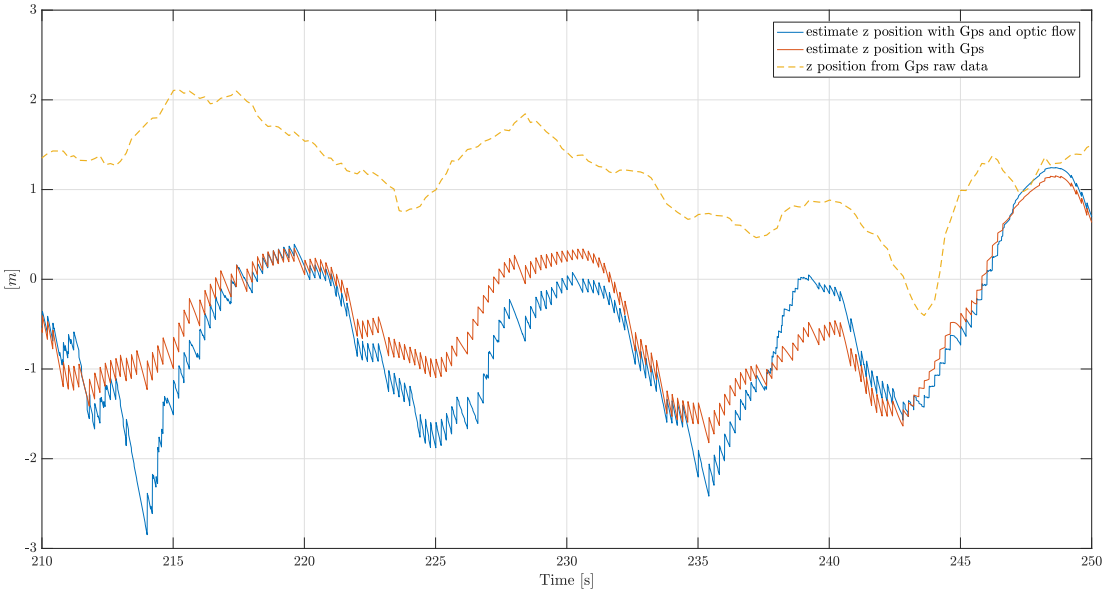


Figure 6.45: Compared results (zoom): position in z

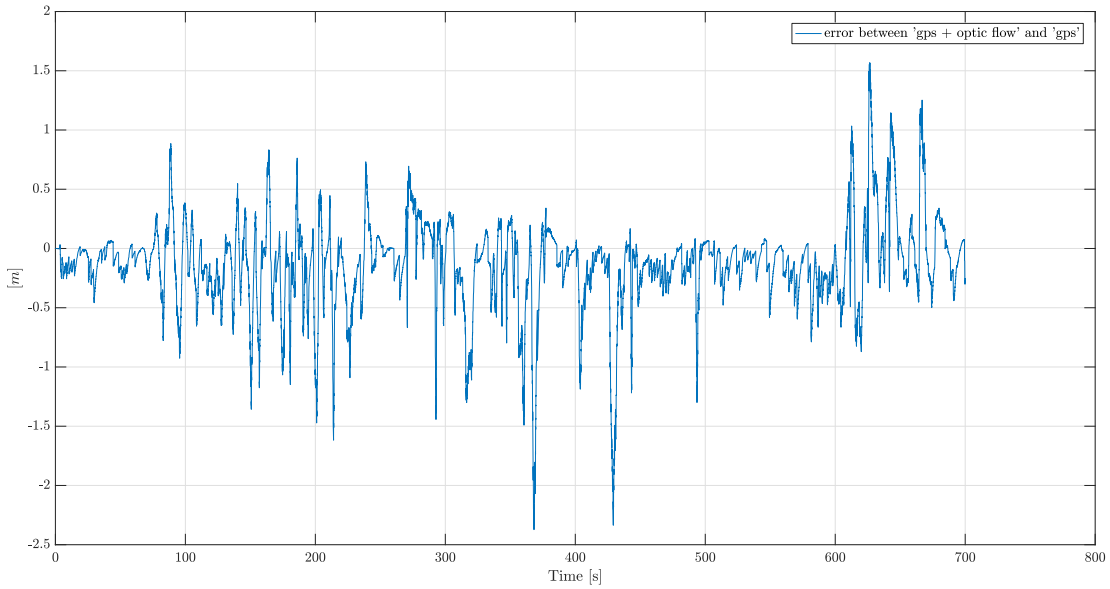


Figure 6.46: Position-z error between the estimate by using GPS + optical flow and GPS only

The Figure 6.47 shows the comparison between: the estimate velocity in x when the filter uses both the GPS and the optical flow measurements, and the estimate velocity in x when the filter uses only the optical flow data from the Px4Flow for the correction step. The two estimates are slightly different, and we can have a quantitative analysis of this difference, looking at the Figure 6.48 in which the error between the two estimates is represented.

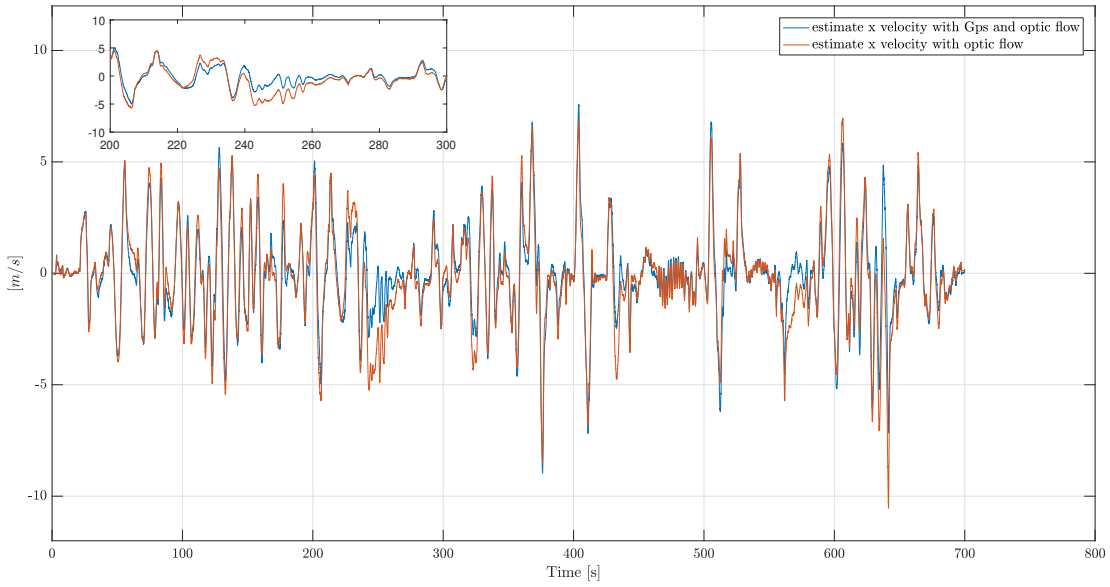


Figure 6.47: Compared results: velocity in x

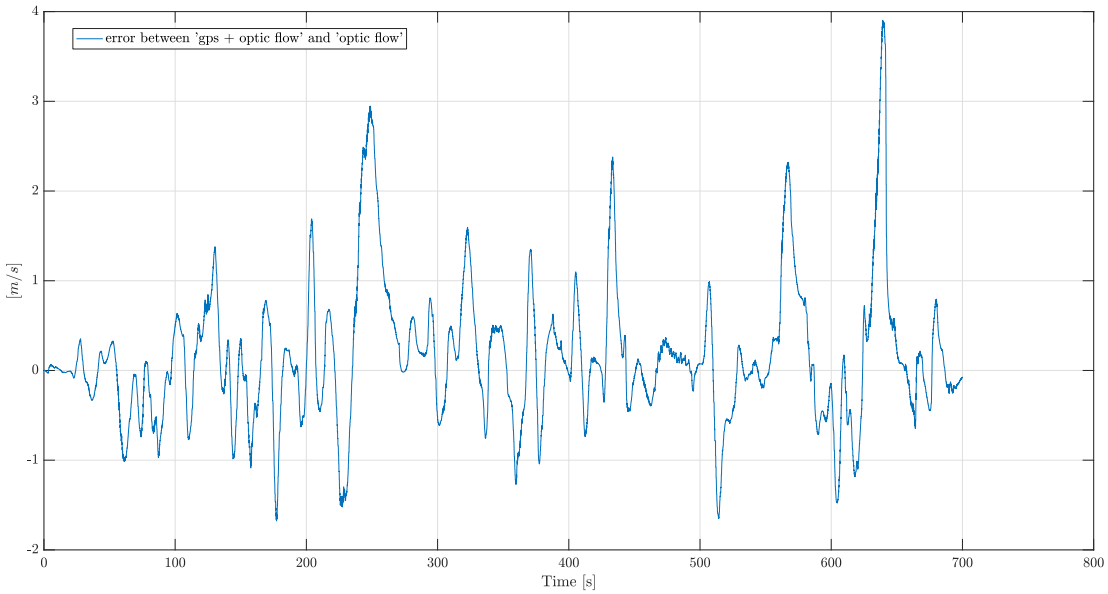


Figure 6.48: Velocity-x error between the estimate by using GPS + optical flow and optical flow only

The Figure 6.49 shows the comparison between: the estimate velocity in y when the filter uses both the GPS and the optical flow, and the estimate velocity in y when the filter uses only the optical flow data from the Px4Flow for the correction step. In the zoom box of this figure we can appreciate the differences between the two estimates. The Figure 6.50 helps us to evaluate this error between the results of the two way to execute the estimation of the velocity in the y axis.

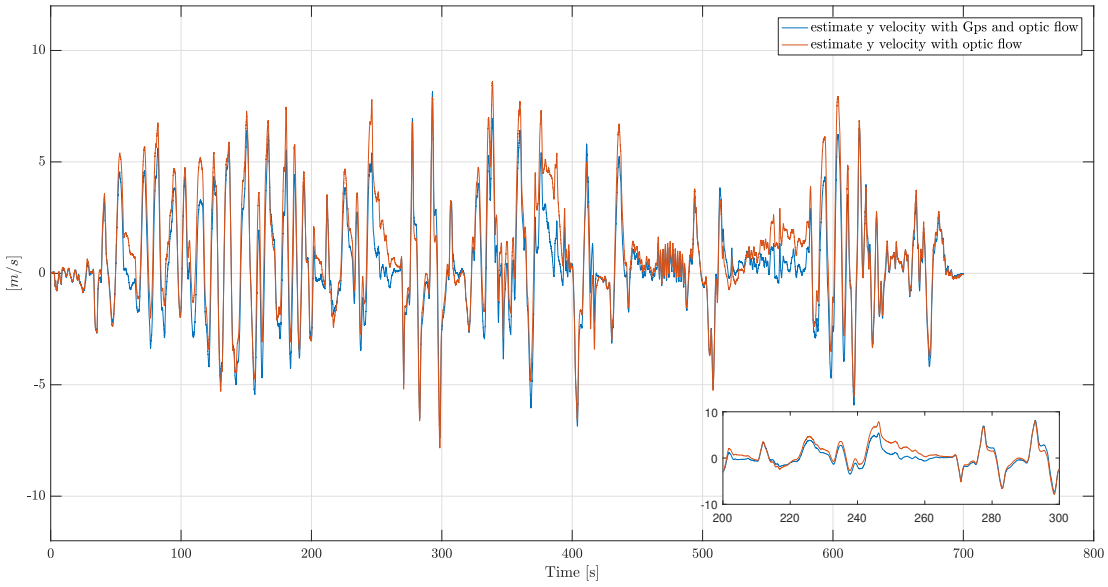


Figure 6.49: Compared results: velocity in y

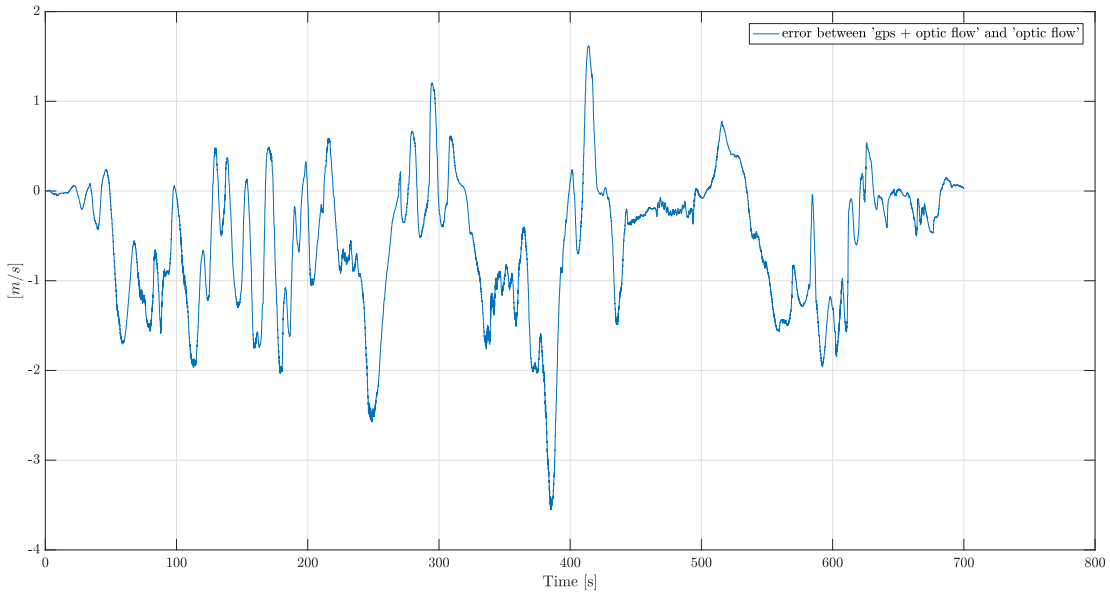


Figure 6.50: Velocity-y error between the estimate by using GPS + optical flow and optical flow only

From the above results we can state that, of course, the implemented Kalman filter works differently according to which set of sensors use, to estimate the position and velocity. We cannot state, which one of the discussed cases work better, because of we don not have an absolutely correct reference, both for the position and for the velocity.

In Figure 6.51 the innovations PSD for the position are compared, analysing the results of the Kalman filter based on the GPS and optical flow data and the results of the Kalman filter based only on the GPS measurements. The innovations PSD in the two cases are very similar, so we cannot observe any differences in the filter working.

In Figure 6.52 the velocity innovations PSD is shown. We compare the results of the Kalman filter when it uses both the GPS and the optical flow measurements, with the results obtained by the Kalman filter based only on the optical flow measurements. We can observe innovations PSD value lower in the case of the availability of both the GPS and the optical flow; so, in view of this, we can state that with the combination of the two sensors we obtain a better estimate of the velocity in x and y axes.

The last interesting observation to do is about the Kalman gain. In Figure 6.53 we compare: the Kalman gain component due to the GPS, related to the x position, in the case when the filter uses both the sensors to correct the estimate; the Kalman gain component due to the GPS, related to the x position, when the filter uses only the GPS measurements. As we can see, in the first case (blue line) the Kalman gain is more coloured, with respect to the second case (orange line), and not stable around a mean value. This is due to the correction step executed

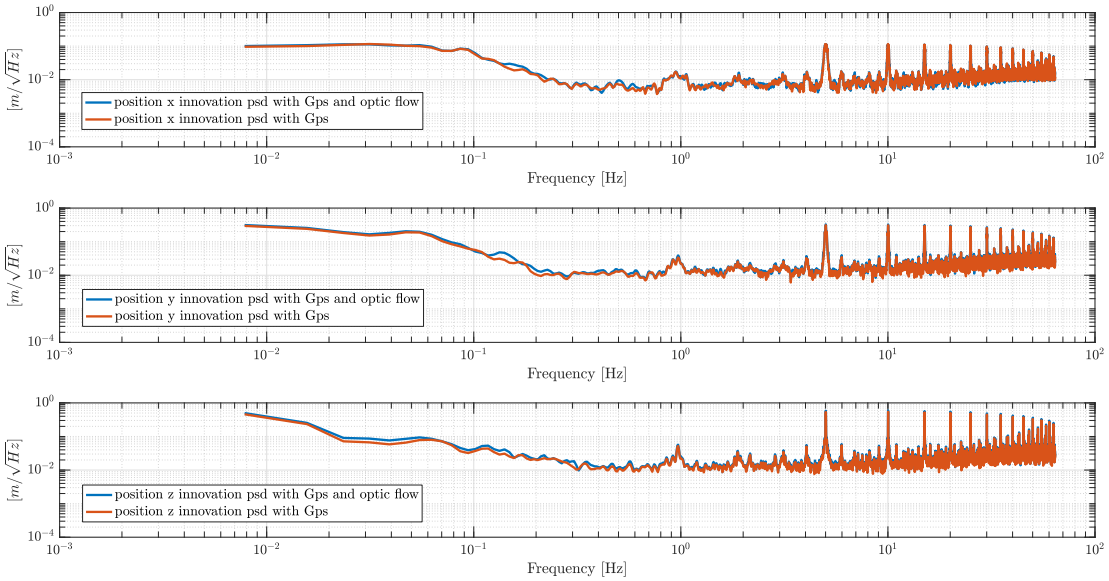


Figure 6.51: Innovations PSD comparison for position

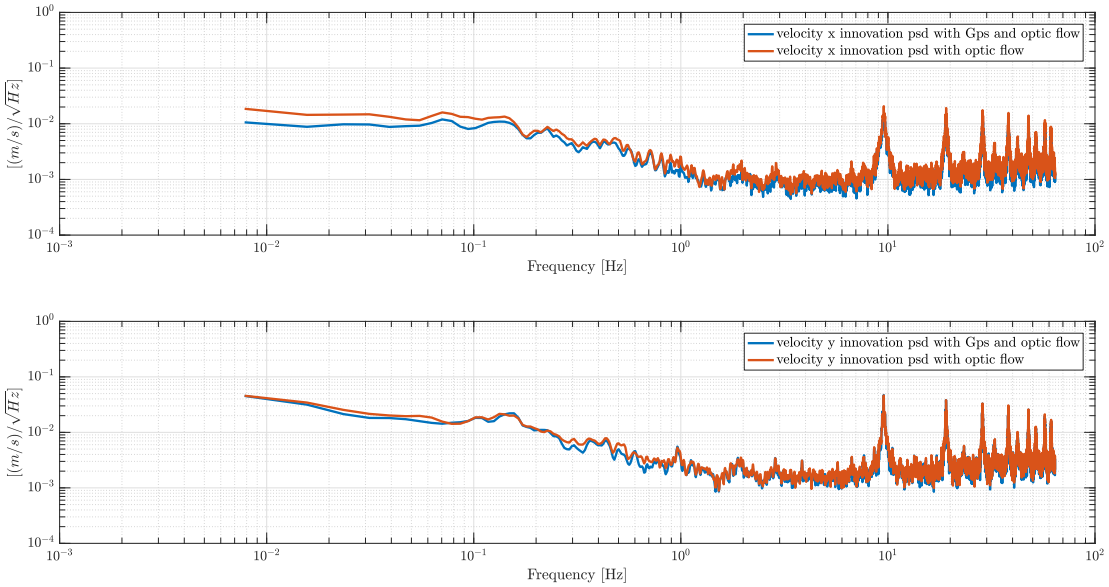


Figure 6.52: Innovations PSD comparison for velocity

by the optical flow measurements, when the P covariance matrix, became non-diagonal. It is function of the H matrix (Eq.(3.2.1)), and this is a full matrix because of the attitude matrix in H_{flow}, as we can see in equation (4.1).

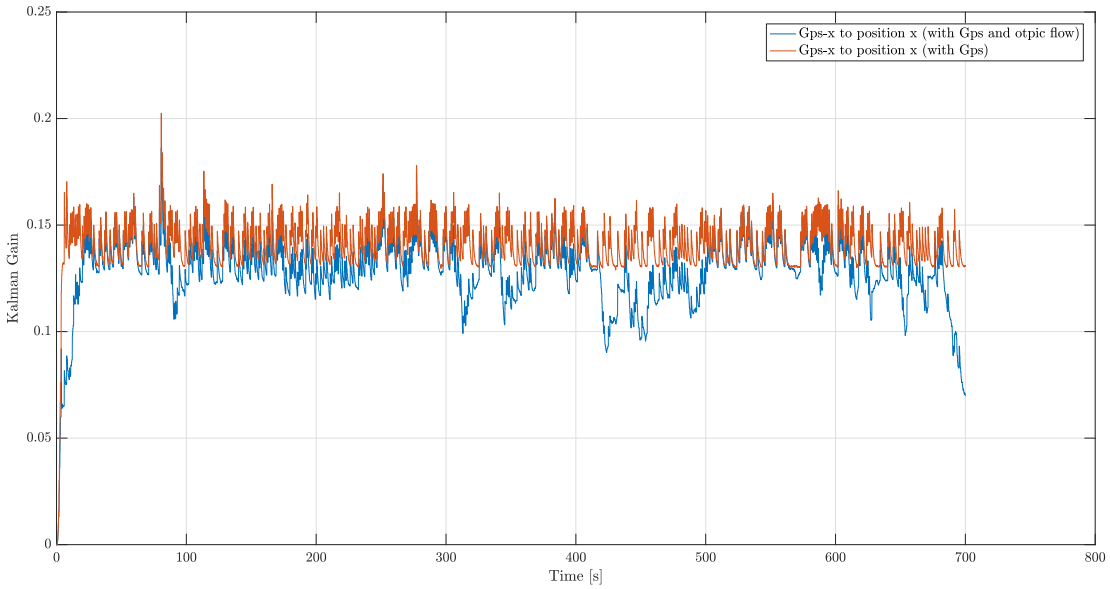


Figure 6.53: Kalman gain comparison for position in x

6.2.6 Optical flow based Kalman filter indoor

Here, we present the estimate results obtained by applying the implemented Kalman filter on the data set logged during the indoor flight described in 6.2.1.2. The correction step is executed by using only the optical flow measurements in order to test if it possible to rely on them to execute an indoor flight without any other facilities.

In Figure 6.54 the estimate velocities along the x axis is compared to the x velocity measured by the motion capture system named Optitrack (Mocap), the laboratory facility that allows the indoor flight. Mocap measurements can be considered more reliable than the GPS measurements, and they represent values very close to the real values of position and velocity. For this reason if the estimate velocities does not match the mocap measurements of velocity, we are sure that the filter is not working in the right way. Looking at Figure 6.54 we can observe that the Kalman filter is not estimating the x velocity in the right way. The same wrong estimate can be observed from Figure 6.55 where the velocity along the y axis is compared to the mocap y velocity measurement.

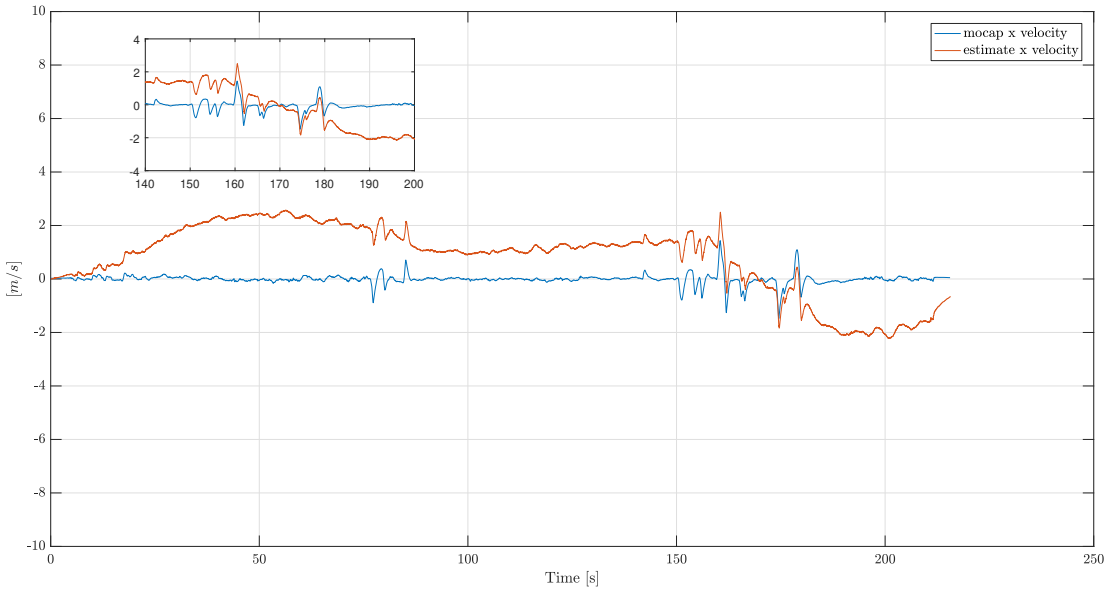


Figure 6.54: Comparison: estimated x-velocity and Mocap x-velocity

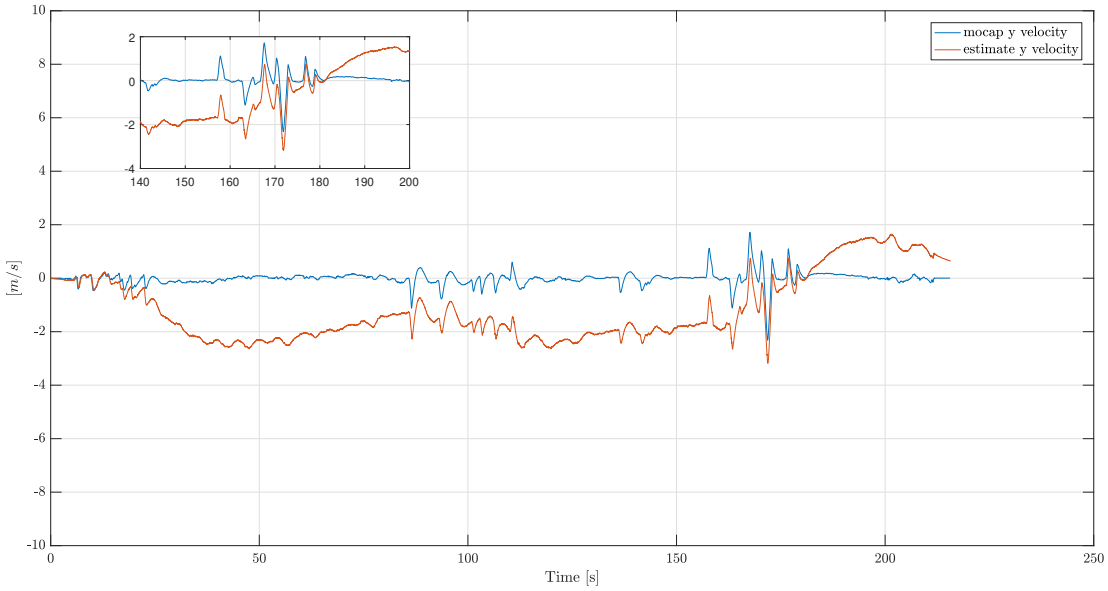


Figure 6.55: Comparison: estimated y-velocity and Mocap y-velocity

The best evidence of the not good working of the filter is given by the behaviour of the P covariance matrix. In Figure 6.56 the red lines may represent the $\pm 3std$ band around the innovations. The standard deviation (std) is computed as the square root of the diagonal elements of the P matrix. The very strange behaviour shown in Figure 6.56 underline that something is not working right in the Kalman filter.

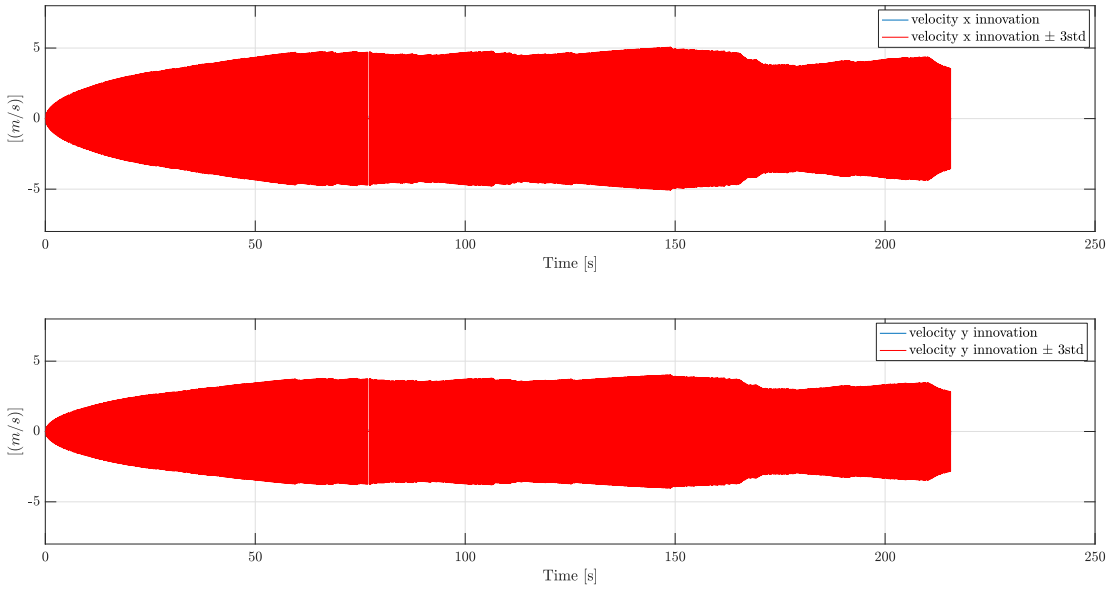


Figure 6.56: Velocity state estimate innovations

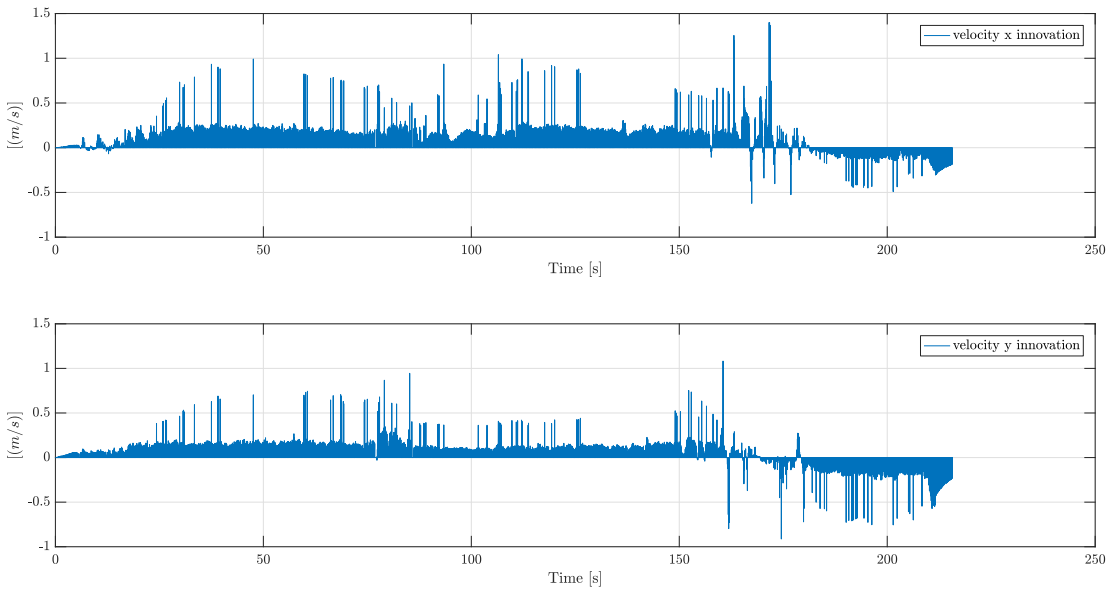


Figure 6.57: Velocity state estimate innovations without ± 3std boundaries

Isolating the velocity innovations representation in Figure 6.57, we can notice again a very strange behaviour, completely different from the innovations represented for the other above cases.

The Figure 6.58 shows the comparison between the innovations PSD and the optical flow measurements PSD. The innovation PSD is lower than the measurements PSD, so the filter is doing its work from this point of view, but we have already stated its bad working for this case.

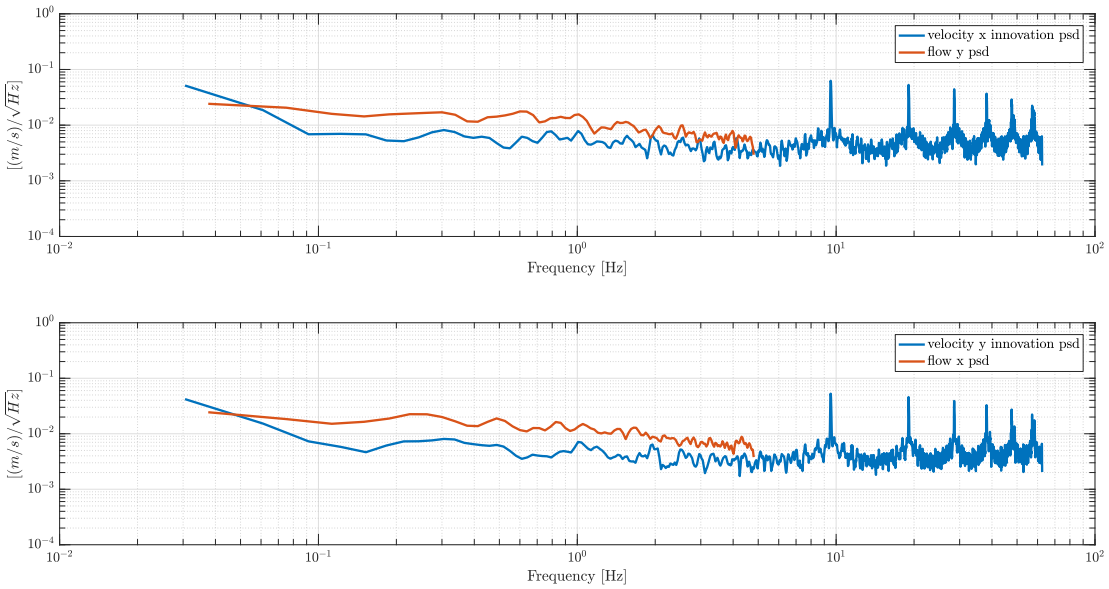


Figure 6.58: Comparison: Innovations PSD and optical flow measurements PSD

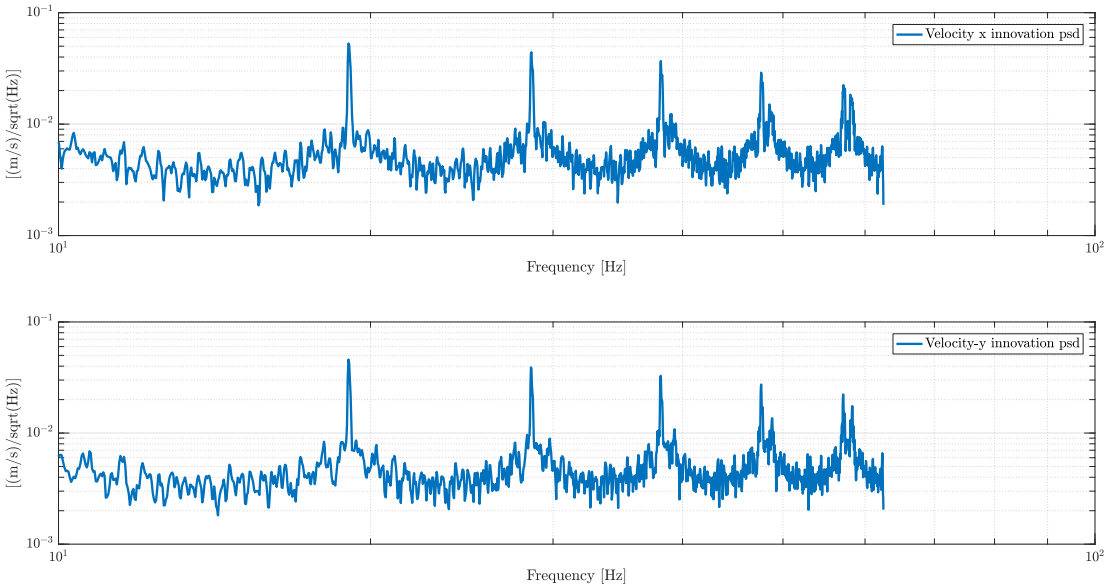


Figure 6.59: Velocity innovations PSD at high frequency

In Figure 6.59 we can observe the velocity innovation PSD at high frequency, and we can appreciate the effect of the multi-rate of the sensors, in particular in this case the peaks appear every $10Hz$ interval, due to the sampling frequency of the optical flow measurements.

The above results allow us to state that an optical flow based Kalman filter does not work good on a data set from an indoor flight with the environment conditions described in 6.2.1.2 . This was expected looking at the quality parameter in Figure

6.7 . In order to improve these estimate results in future works, the first thing to do is to try to find visibility conditions which ensure an high quality of the optical flow sensor indoor; from the filter point of view, the challenge to face with, is to find a way to involve the quality parameter in the tuning part.

6.2.7 GPS and optical flow based H_∞ filter outdoor

The implemented H_∞ filter suitable for the Kalman filter architecture, allows to modify the tuning of the R covariance matrix in order to change the performance of the filters. As we can see from the theory discussed in paragraph 3.2.2, the problem is to tune θ. Remembering that for $\theta \longrightarrow 0$ the H_∞ filter becomes equal to the Kalman filter.

In this paragraph the estimation results for different θ are presented, in order to show how the performances of the filter can bee modified according to the aim of the work. For the following results the chosen scenario is the outdoor flight described in 6.2.1.1 , with the GPS and optical flow measurements used for the correction step of the filter. The condition representing a guideline to tune θ is defined in equation (3.43); it ensures the well-posedness of the $\bar{R}$ matrix (equation (3.41)). In order to have a wide panoramic of how the filter performances change for different θ, we have chosen: $\theta = 0.5$, $\theta = 0.9$ which respect the condition of well-posedness of $\bar{R}$; $\theta = 10$ that does not ensure the same well-posedness condition.

In Figure 6.60 we compare the estimate x position for different thetas, taking into account also $\theta = 0$ that means original Kalman filter. In this case, with these values, the estimates change but there are not big differences between them.

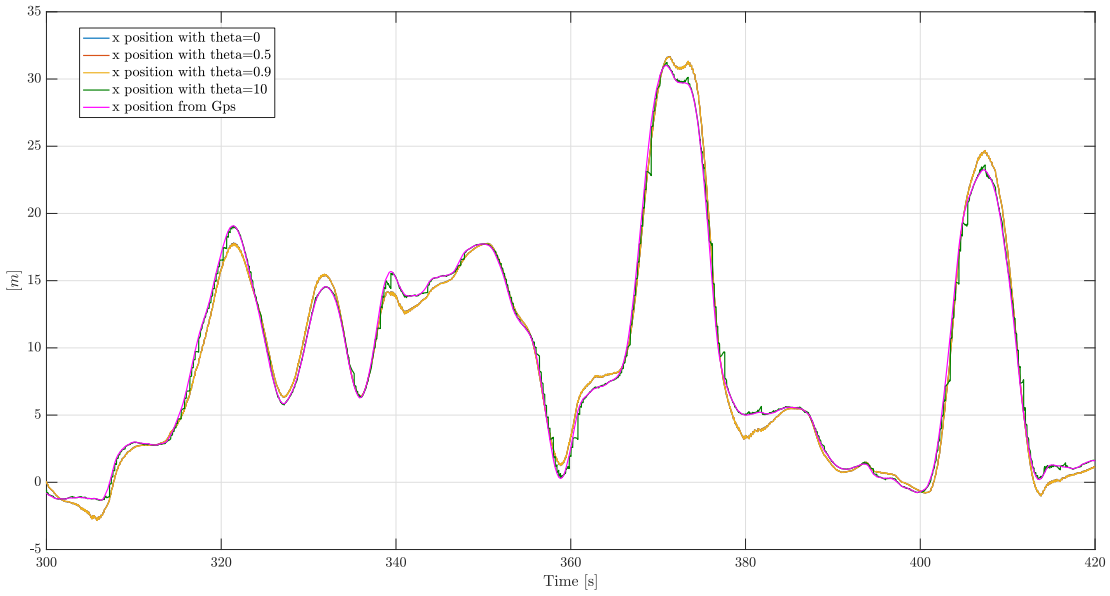


Figure 6.60: Estimate x position: H_∞ with different thetas

The Figure 6.61 shows the comparison between the estimates of y position for different theta values. In this case, for example, increasing the value of theta to $\theta = 0.9$ (thus, getting away from the original Kalman filter) we obtain higher error with respect to the raw GPS data. If we increase more the theta value, using $\theta = 10$, not ensuring the well-posedness condition, we obtain an estimate similar to the Kalman filter one. This is just to show how the H_∞ filter can be used to change the filtering performances.

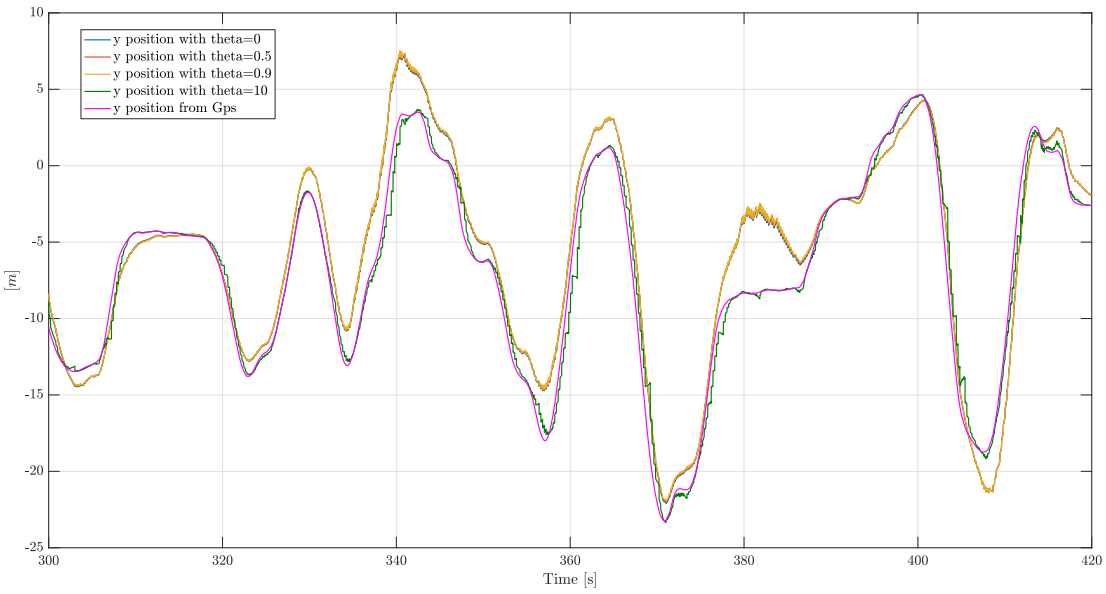


Figure 6.61: Estimate y position: H_∞ with different thetas

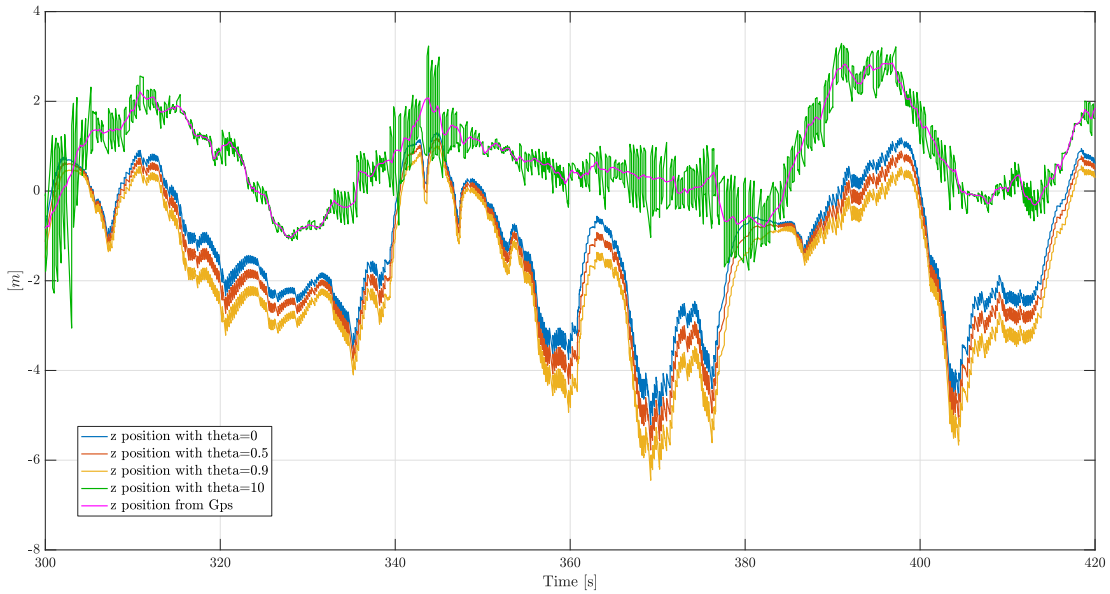


Figure 6.62: Estimate z position: H_∞ with different thetas

In Figure 6.62 we compare the results about the estimate z position. As we can see, in this case the trend is different: increasing the value of theta, the error with respect to the raw GPS data decreases.

In view of the above results, we can appreciate how the implemented form of the H_∞ filter, could be a useful tool to modify the Kalman filter performances.

Conclusions

The aim of this thesis was to reach a position and velocity estimation implementing a Kalman filter based on GPS and optical flow measurements, to improve the pre-existing UAV navigation system.

A new optical flow sensor, called Px4Flow, has been integrated on-board the UAV and its output has been identified by flight testing activity. The next step has been to find the relation between the optical flow sensor output and the UAV velocity in the NED reference frame, in order to involve it in the mathematical model.

A Kalman filter based on both the GPS and the optical flow, able to manage the multi-rate problem given by the different sensors, has been implemented. The tuning of the Kalman filter has executed using the variances of the involved sensors which have been defined through new testing activity on the UAV. A H_∞ filter has implemented in a suitable form with the already implemented architecture of the Kalman filter. This supplementary implementation has been done in order to have the possibility to change the performances of the original Kalman filter.

A flight test activity, outdoor and indoor, has been executed to collect the real data sets on which the filter has been applied. From the estimation results obtained on the real data, different conclusions have been reached: the implemented Kalman filter, based on Gps and optical flow, for an outdoor flight, gives reliable estimates of position and velocity when it uses both the sensors; also the implemented Kalman filter, using only the GPS during an outdoor flight, gives reliable estimates of position and velocity; the Kalman filter, using only the optical flow outdoor, gives good estimates of the velocity in x and y NED axis. In view of this, it is difficult to state if the addition of the optical flow measurements, improves the estimation results because as a reference we do not have absolute values of position and velocity during the outdoor flight. On the other hand, we can state that the possibility to have two sensors that contributes to the velocity estimate, instead of one, makes more robust the optimal estimate of the Kalman filter.

The application of the implemented Kalman filter on the indoor flight data, using only the optical flow, does not give the same good results. The surface of the available indoor environment does not ensure good visibility conditions for the Px4Flow camera, so different solutions, to cover the floor, have been tried, improving the quality of the optical flow measurements. The improved optical

flow quality is not enough to ensure a reliable estimate for the velocity for an indoor flight.

Concluding, some considerations about the future developments, could be done:

- find a surface able to ensure the best visibility conditions for the Px4Flow camera for the indoor flight;
- evaluate the possibility to involve the quality parameter of the Px4Flow in the implemented Kalman filter in order to take it into account to define the uncertainty of the optical flow measurements;
- model the dynamic of the sensors and involve it in the Kalman filter in order to obtain not approximated variances values.

Bibliography

[1] U.S Defense Mapping agency. *World Geodetic System 1984(WGS 84)-Its Definition and Relationship with Local Geodetic Systems*. Maryfield, VA: Washington, DC, USA, 1991.

[2] James E Cutting. Images, imagination, and movement: Pictorial representations and their development in the work of James Gibson. 2000.

[3] Myron Kayton and Walter R Fried. *Avionics navigation systems*. John Wiley & Sons, 1997.

[4] Carlo E.D. Riboldi Giorgio Guglieri. *Introduction to Flight Dynamics*. Torino, TO, Italy, 2014.

[5] James J. Gibson. *The Perception of the Visual World*. Houghton Mifflin Company, Boston, MA, USA, 1950.

[6] Franck Ruffin Julien R. Serres. Optic flow-based collision-free strategies: From insects to robots. 2017.

[7] Petri Tanskanen Marc Pollefeys Dominik Honegger, Lorenz Meier. An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications . 2013.

[8] Vladislav Klein and Eugene A Morelli. *Aircraft system identification: theory and practice*. American Institute of aeronautics and astronautics Reston, VA, 2006.

[9] M Giurato, F Haydar, M Lovera, G Sechi, et al. Very high accuracy attitude determination for los steering. In *10th International ESA Conference on Guidance, Navigation & Control Systems*, pages 1–15, 2017.

[10] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. 2006.

[11] François E Cellier and Ernesto Kofman. *Continuous system simulation*. 2006.

[12] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. 2006.