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

Kalman filtering for UAV navigation using optical flow

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

Thesis by:
Kerlus BSTAOROS Matr. 884582

Academic Year 2019–2020

Alla mia grande famiglia sparsa per tutto il mondo.

Acknowledgments

Questo lavoro di tesi rappresenta la conclusione di un percorso durato anni, che non sarebbe stato possibile se non grazie all'aiuto e alla pazienza di molte persone che mi sento di voler ringraziare.
Il primo ringraziamento va al Professore Marco Lovera, che mi ha dato la possibilità di approfondire e sviluppare un argomento affascinante e interessante come quello sugli UAV. Lo ringrazio anche per i continui consigli che mi ha dato per procedere al meglio nel lavoro.
Ringrazio Mattia Giurato che fin dall'inizio mi ha seguito, dandomi la sua disponibilità e trasmettendomi le sue conoscenze, importanti per me per entrare meglio nell'ottica del lavoro. Ringrazio inoltre Simone Panza per la sua disponibilità ad aiutarmi a superare le difficoltà che ho incontrato durante il lavoro in laboratorio.
Un grande ringraziamento va a tutti i ragazzi che ho incontrato nel laboratorio ASCL, per il contributo che ciascuno di loro ha dato per rendere migliori le giornate passate in laboratorio.
Il percorso universitario, durato diversi anni non poteva essere lo stesso senza la compagnia delle persone con cui ho passato la maggior parte del tempo. In ordine di quando li ho conosciuti un particolare ringraziamento va a: Frank, Gaia, Giorgia, Simo, Fabri, Loris, Flavio, Francesco, Lore, Gregorio, Giovanni e Fra, e molti altri con cui sono stato in contatto, per aver reso più leggere e divertenti le infinite giornate in università, durante pause e lezioni; senza dimenticare il tempo trascorso insieme a festeggiare ricorrenze pagane varie.
Le persone che fin dal primo giorno di università fino all'ultimo, mi sono state vicine, e che non capirò mai come abbiano fatto a supportarmi e sopportarmi come hanno fatto loro, nei tanti momenti di difficoltà che ho incontrato, sono la mia famiglia. A partire dai miei zii, le mie zie e i miei cugini che per qualunque cosa sono stati disponibili, va un enorme ringraziamento; senza di loro non sarebbe stato possibile attraversare questo percorso in maniera tanto liscia. Un particolare ringraziamento va ai miei nonni, che malgrado la loro lontananza, è come se li avessi avuti sempre al mio fianco. Un grande grazie va ad Abuna Marco per tutto ciò che ha fatto per me in questo lungo percorso.
Infine un grande grazie va a mia sorella Mariam durante le stressanti giornate delle sessioni di esame e per essere stata anche una grande amica con cui passare tante ore a conversare; ai miei genitori Evelen ed Emad per aver fatto in modo

che io potessi pensare a impegnarmi in questo percorso; grazie ai tanti sacrifici che avete fatto nella vostra vita per dare questa occasione a me, GRAZIE.

Abstract

Nowadays the drone tehnology is used in several activities, civil and military, then it represents a wide field of research and development. Despite the UAVs are still in the infancy stage, in terms of mass adoption and usage, respect to what they can offer, they have already exploited in the industries, decreasing the workload and the production costs, imoproving accuracy, and resolving some security issues. Moreover the UAVs are extensively used for photography, journalism and to capture films; for shipping and delivery; to penetrate extreme enviroments (like mines) and for other several activities. Therefore the navigation system has become an important player, which has been subjected to continious developments and improvements with new kinds of sensors and technologies. In this historical period, the use of optical technologies in order to equipped the UAV with the vision, is of large use. The integration of the aforesaid technology returns an improvement and more robustness to the pre-existing navigation system, allowing the autonomous flight of the UAV in cluttered or indoor enviroment.

The purpose of the thesis is to integrate the GPS with the optical sensor, in order to improve the estimation process keeping into account that the noise provided by the sensors is not well known. For this reason three versions of the Kalman filter will be implemented: in the first version the noise will be modeled as a white noise, *i.e.* the variances will be considered as constants; the second version will keep into account the dynamics of the noises, by means of shape filters, namely systems which take as input a white noise and return as output the noise of interest; the third version of the Kalman filter is based on an adaptive algorithm, in which at each step the variance of the noises is modified according to a certain mathematical law.

To reach the aforesaid purposes, the first thing made is to recognize the relation between the output of the optical sensor and the descriptive states of the UAV; then the kinematic model of the UAV and the output equation, which link the measurements provided by the sensor and the state variables, are written; subsequentely the discrete Kalman filter algorithm is drafted and applied to the UAV state space system based on the kinematic equation. Then, for each version of the Kalman filter a path will be followed: for the augmented one, the shape filters are realized, and then the state vector of the UAV is augmented with the states of the shape filters; instead, for the adaptive filter, the base algorithm is modified. Prac-

tically the augmented filter is obtained making a re-tuning of the first version of the Kalman filter (that in which the noises are considered white noises), while the adaptive Kalman filter is obtained by modifying the algorithm of the first version of the Kalman filter, adding the equations according to which the variance of the noises is adapted. Once the three Kalman algorithms are implemented the flight test activities are decided and performed. Finally, when the data of the flights are collected, the filtering process is executed.
The aim of the work is to evaluate the action of the different versions of the filter, namely if the estimation results are reliable or less in several conditions; and more important if the addition of the dynamics of the noises (second version of the filter) or the use of an adaptive version (third version of the filter), can improve the estimation performances when the UAV is let to fly in bad conditions for the optical technology.

Sommario

Oggigiorno la tecnologia dei droni è usata in molte attività, civili e militari, perciò rappresenta un largo campo di ricerca e sviluppo. Benché gli UAV siano ancora nella loro fase iniziale, in termini di quantità adottata e uso, rispetto a quello che possono offrire, sono stati sfruttati nell'industria diminuendo il carico di lavoro e i costi di processo, migliorando l'accuratezza, e risolvendo problemi di sicurezza. Inoltre gli UAV sono estensivamente usati per la fotografia, giornalismo e per catturare film; per spedizioni e consegne; per penetrare ambienti estremi (come miniere) e per altre molte attività. Quindi il sistema di navigazione è diventato un importante giocatore, che è stato soggetto a continui sviluppi e miglioramenti con nuove tipologie di sensori e tecnologie. In questo periodo storico, l'uso di tecnologie ottiche che permettano all'UAV la visione, sono di largo uso. L'integrazione delle suddette tecnologie portano un miglioramento e una maggiore robustezza al pre-esistente sistema di navigazione, permettendo il volo autonomo dell'UAV in ambienti chiusi.

Il proposito di questa tesi è di integrare le misure del GPS con quelle del sensore ottico, in modo da migliorare il processo di stima tenendo in conto che il rumore fornito dai sensori non è ben noto. Per questa ragione tre versioni del filtro di Kalman saranno implementate: nella prima versione il rumore sarà modellato come rumore bianco, cioè le varianze in gioco saranno considerate costanti; la seconda versione terrà in conto la dinamica dei rumori, per mezzo dei filtri di forma, cioè sistemi che prendono come input un rumore bianco e restituiscono come output il rumore di interesse; la terza versione del filtro di Kalman è basata su un algoritmo adattivo, in cui ad ogni step la varianza del rumore è modificata secondo una certa legge matematica.

Per il raggiungimento del suddetto obiettivo, la prima cosa da fare è riconoscere la relazione tra l'output del sensore ottico e gli stati descrittivi dell'UAV; quindi il modello cinematico dell'UAV e l'equazione degli output, che lega le misure fornite dai sensori e le variabili di stato, sono scritte; successivamente l'algoritmo del filtro di Kalman discreto è redatto e applicato al sistema agli stati dell'UAV, basato sulle equazioni della cinematica. Per ciascuna versione del filtro di Kalman, un diverso procedimento sarà eseguito: per il filtro aumentato, sono realizzati i filtri di forma, e successivamente il vettore delle variabili di stato dell'UAV è aumentato con gli stati dei filtri di forma; per il filtro adattivo, l'algoritmo di base è invece

modificato. Praticamente il filtro aumentato è ottenuto attraverso una ritaratura della prima versione del filtro di Kalman (quella in cui i rumori erano considerati bianchi), mentre il filtro di Kalman adattivo è ottenuto modificando l'algoritmo della prima versione del filtro di Kalman, aggiungendo le equazioni secondo le quali la varianza dei rumori è adattata. Una volta che i tre algoritmi di Kalman sono implementati, le attività di volo sono decise e condotte. In conclusione, quindi i dati di volo sono raccolti e il processo di filtraggio è eseguito.
L'obiettivo di questo lavoro è di valutare l'azione delle differenti versioni del filtro, cioè se i risultati di stima sono affidabili o meno in diverse condizioni; e più importante se l'aggiunta della dinamica dei rumori (seconda versione del filtro) o l'utilizzo di un approccio adattivo (terza versione del filtro), possono migliorare le prestazioni della stima, quando l'UAV è lasciato volare in cattive condizioni per la tecnologia ottica.

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 frame 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 7

1.4 Different navigation system configurations 9

1.4.1 Inertial navigation system 9

1.4.2 Inertial and satellite radio navigation system 11

1.4.3 Work principles 11

1.4.4 Inertial, satellite radio and optical navigation system . . . 13

2 Optical flow **15**

2.1 Definition of optical flow 15

2.2 Computation of the optical flow 17

2.3 Integration of the optical flow in the navigation system 18

2.3.1 Angular compensation 18

2.3.2 Ground altitude scaling 19

2.3.3 Integration of the optic flow in the navigation system . . . 20

2.4 Optical flow equations 21

3 Kalman Filter for position and velocity estimation 25

3.1 Mathematical model 25
3.2 Implemented Kalman Filter 31
3.2.1 Data pre-processing 31
3.2.2 Initialization 32
3.2.3 Tuning 32
3.2.4 Multi-rate management: sequential update 34

4 Augmented Kalman Filter 37

4.1 Mathematical model 37
4.1.1 Description of the noise dynamics 37
4.1.2 Augmented system 39
4.1.3 Discrete-time domain description 43
4.2 Implemented augmented Kalman filter 44
4.2.1 Data pre-processing 45
4.2.2 Initialization 45
4.2.3 Tuning 45
4.2.4 Multi-rate management: sequential update 47
4.3 Conclusions 48

5 Adaptive Kalman Filter 49

5.1 Mathematical problem 49
5.1.1 Description of the noise 49
5.2 Implemented adaptive Kalman filter 52
5.2.1 Data pre-processing 52
5.2.2 Initialization 53
5.2.3 Multi-rate management: sequential update 53
5.3 Conclusion 56

6 Experimental set up 57

6.1 UAV and Px4Flow set-up 57
6.1.1 Optical flow sensor: Px4Flow 58
6.1.2 On board integration 59
6.2 Sensor characterization 60
6.3 Pre-test activities 66

7 Filtering results 69

7.1 Estimate results: augmented Kalman filter 69
7.1.1 Outdoor flight 69
7.1.2 Indoor flight 79
7.2 Estimate results: adaptive Kalman filter 97
7.2.1 Outdoor flight 98

7.2.2 Indoor flight 105
7.3 Estimate results: augmented vs adaptive Kalman filter 123
7.4 Conclusions 128

Conclusions **133**

List of Figures

1.1 The WGS 84 reference frame [2] 5
1.2 Rotation singularity 8
1.3 Trilateration of satellites 12
1.4 Multipath signals 13

2.1 Optical flow (flying forward) [6] 16
2.2 Optical flow (flying to the left) [6] 16
2.3 Insect optical flow 17
2.4 Block matching by SAD 18
2.5 Multi-copter coupling between translational and rotational motion 19
2.6 Apparent motion due to rotation 19
2.7 Scheme for the on board integration of the optical flow 20
2.8 Optical flow and velocity relation 21
2.9 Flow sense 23

3.1 Measurements availability check 36

6.1 Px4Flow sensor 58
6.2 Dimensions and connectors 59
6.3 Accelerometer noise 60
6.4 Optical flow sensor noise 61
6.5 GPS noise 62
6.6 MoCap noise 62
6.7 Shape filter realization of the acceleration noise 63
6.8 Shape filter realization of the optical flow noise 64
6.9 Shape filter realization of the GPS noise 64
6.10 Shape filter realization of the MoCap noise 65

7.1 Position estimation along x-axis 70
7.2 Position estimation along y-axis 71
7.3 Position estimation along z-axis 71
7.4 Comparison of discrepancies ”estimation-measurement” for position 72
7.5 Velocity estimation along x-axis 72
7.6 Velocity estimation along y-axis 73

7.7 Velocity estimation along z-axis 73
7.8 Comparison of discrepancies ”estimation-measurement” for velocity 74
7.9 Discrepancy between estimations 75
7.10 Innovation $PSDs$ for position 75
7.11 Innovation $PSDs$ for f 76
7.12 Position innovations 77
7.13 Optical flow innovations 77
7.14 Kalman gain due to the position measurements 78
7.15 Kalman gain due to the optical flow measurements 78
7.16 Quality of the optical flow 79
7.17 Position estimation along x-axis 80
7.18 Position estimation along y-axis 80
7.19 Position estimation along z-axis 81
7.20 Comparison of discrepancies ”estimation-measurement” for position 82
7.21 Velocity estimation along x-axis 82
7.22 Velocity estimation along y-axis 83
7.23 Velocity estimation along z-axis 83
7.24 Comparison of discrepancies ”estimation-measurement” for velocity 84
7.25 Discrepancy between estimations 84
7.26 Innovation $PSDs$ for position 85
7.27 Innovation $PSDs$ for f 86
7.28 Position innovations 86
7.29 Optical flow innovations 87
7.30 Kalman gain due to the postion measurements 87
7.31 Kalman gain due to the optical flow measurements 88
7.32 Quality of the optical flow 88
7.33 Position estimation along x-axis 89
7.34 Position estimation along y-axis 90
7.35 Position estimation along z-axis 90
7.36 Comparison of discrepancies ”estimation-measurement” for position 91
7.37 Velocity estimation along x-axis 91
7.38 Velocity estimation along y-axis 92
7.39 Velocity estimation along z-axis 92
7.40 Comparison of discrepancies ”estimation-measurement” for velocity 93
7.41 Discrepancy between estimations 93
7.42 Innovation $PSDs$ for position 94
7.43 Innovation $PSDs$ for optical flow 94
7.44 Position innovation 95
7.45 Optical flow innovation 95
7.46 Kalman gain due to the position measurements 96
7.47 Kalman gain due to the position measurements 96
7.48 Quality of the optical flow 97
7.49 Position estimation alog x-axis 98

7.50 Position estimation along y-axis 98
7.51 Position estimation along z-axis 99
7.52 Comparison of discrepancies ”estimation-measurement” for position 100
7.53 Velocity estimation along x-axis 100
7.54 Velocity estimation along y-axis 101
7.55 Velocity estimation along z-axis 101
7.56 Comparison of discrepancies ”estimation-measurement” for velocity 102
7.57 Discrepancy between estimations 102
7.58 Innovation $PSDs$ for position 103
7.59 Innovation $PSDs$ for optical flow 103
7.60 Position innovation 104
7.61 Optical flow innovation 104
7.62 Kalman gain due to the position measurements 105
7.63 Kalman gain due to the optical flow measurements 106
7.64 Position estimation along x-axis 106
7.65 Position estimation along y-axis 107
7.66 Position estimation along z-axis 107
7.67 Comparison of discrepancies ”estimation-measurement” for position 108
7.68 Velocity estimation along x-axis 109
7.69 Velocity estimation along y-axis 109
7.70 Velocity estimation along z-axis 110
7.71 Comparison of discrepancies ”estimation-measurement” for velocity 110
7.72 Discrepancy between estimations 111
7.73 Innovation $PSDs$ for position 111
7.74 Innovation $PSDs$ for optical flow 112
7.75 Position innovation 112
7.76 Optical flow innovation 113
7.77 Kalman gain due to the position measurements 114
7.78 Kalman gain due to the optical flow measurements 114
7.79 Position estimation along x-axis 115
7.80 Position estimation along y-axis 115
7.81 Position estimation along z-axis 116
7.82 Comparison of discrepancies ”estimation-measurement” for position 116
7.83 Velocity estimation along x-axis 117
7.84 Velocity estimation along y-axis 117
7.85 Velocity estimation along z-axis 118
7.86 Comparison of discrepancies ”estimation-measurement” for velocity 119
7.87 Discrepancy between estimations 119
7.88 Innovation $PSDs$ for position 120
7.89 Innovation $PSDs$ for optical flow 120
7.90 Position innovation 121
7.91 Optical flow innovation 121
7.92 Kalman gain due to the position measurements 122

7.93 Kalman gain due to the optical flow measurements 122
7.94 Comparison: KF vs AugKF vs AdKF for position 123
7.95 Comparison: KF vs AugKF vs AdKF for velocity 124
7.96 Comparison: KF vs AugKF vs AdKF innovation *PSDs* for position 125
7.97 Comparison: KF vs AugKF vs AdKF innovation *PSDs* for optical flow 125
7.98 Comparison: KF vs AugKF vs AdKF for position 126
7.99 Comparison: KF vs AugKF vs AdKF for velocity 127
7.100 Comparison: KF vs AugKF vs AdKF innovation *PSDs* for position 127
7.101 Comparison: KF vs AugKF vs AdKF innovation *PSDs* for optical flow 128
7.102 Comparison: KF vs AugKF vs AdKF for position 129
7.103 Comparison: KF vs AugKF vs AdKF for velocity 129
7.104 Comparison: KF vs AugKF vs AdKF innovation *PSDs* for position 130
7.105 Comparison: KF vs AugKF vs AdKF innovation *PSDs* for optical flow 130

List of Tables

6.1 Flights for noise characterization 60
6.2 Noise variances 66
6.3 α, c_Q and c_R 66

Introduction

UAV (Unmanned Aerial Vehicle), usually called drone, is the definition for an aicraft in which the pilot does not sit or stand within the vehicle itself. Following the aforesaid definition, the earliest unmanned aerial vehicle in the history of the drones was adopted by the austrian soldiers to attack Venice. Those were unmanned balloons filled with explosives. Important UAV technology improvements have been observed during the World War II and the Cold War. Then, the first use of the unmanned vehicles, focused on belligerent purposes: to spy and recon the enemy's position, to jam communications, and to act as decoys that would prevent the loss of pilot life. In recent times, the drones have been adopted for commercial purposes. An example is their use to spray pesticides on farms, 3D mapping and other several activities.

Then, nowadays there are a lot of activities which can benefit from the UAVs: in military and security field; in civil and scientific field.

What a drone need for the execution of 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, velocty, attitude, etc. To perform this activity, the FCU needs the informations about the state of the drone (position, velocity, acceleration at which is subjected, attitude, etc.) during the whole flight. This is possible thank to the navigation system, that using different sensors, and by means of their integration, permit to the drone to reach the knowledge of the informations about the flight. All the measurements by the sensors are fused in a state estimation process.

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

As said previously, the UAVs can be adopted in extreme enviroments, then is necessary that they work in cluttered places, where the GPS loses its efficiency; then the navigation system cannot rely position and velocity measurements. From this weak point a new idea was born: provide a sensor based on the optical technology, the Px4Flow.

The aim of this thesis is to take the Kalman filter algorithm already developed in ASCL laboratory by the author Simone Musacchio in [1], based on the integration of the measurements provided by the GPS and the Px4Flow, and to model

the noises provided by both the sensors. This is made by developing two different versions of the Kalman filter: Augmented Kalman Filter in which the dynamics of the noises is considered; Adaptive Kalman Filter which is based on an algorithm in which the variance of the noises is adapted, according to a certain mathematical law. In the implementation of the filter algorithms, the multi-rate problem due to the different sensor sampling frequencies, is treated in the same way seen in the previous thesis work in [1], namely by using the sequential update method. The implemented filters are used to perform the estimation of the UAV states, using as flight data, those logged during the flights. Practically the estimation process is performed post-flight.
The structure of the thesis is as follows:

- 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 mathematical model and of the Kalman filter algorithm;
- presentation of the mathematical model and of the Augmented Kalman filter algorithm;
- presentation of the mathematical model and of the Adaptive Kalman filter algorithm;
- description of the experimental set-up including hardware integration of the optical flow sensor on the UAV, and its lens calibration; the shape filter realzations are shown;
- presentation of the experimental results.

Chapter 1

Problem formulation

In the following chapter the purpose is to introduce the problem of the UAV navigation, namely how to collocate it, in space and time. This operation involves the definition of different reference frames as well as the use of a navigation system, which will be explained in this section. Then the mathematical formalism exploited to develop this part of the thesis work will be illustrated, followed by a comparison among the different navigation systems and more precisely their advantages and disadvantages.

1.1 Navigation system definition

One of the pricipal needs of a craft or a vehicle moving from a place to another, is to know its position and velocity. The navigation is the discipline by means which this request is satisfied. Along the time several systems were adopted to allow the navigation, and thanks to the technology developments, these become more precise and faster. For example one of the first forms of navigation used for the earliest aircrafts, was the astronavigation, in which the sun, moon and stars were used to determine the position; that is visually navigation. Subsequently pretty simple but efficient sensors were used, like the 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 step ahead has been made by the "dead-reckoning" computation, used to improve the estimation of the position during the flight. Its idea is based on the knowledge, with the highest precision possible, of the speed value of the vehicle (without the wind speed contribute) and of its true heading, for the computation of the North and the East compo-

nents of the velocity; from this, the computation of the position by mathematical integration is possible. This implies a cumulative error given by the integration, which inluences the estimate proprtionally 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 [4]).
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 the exposure of the aforesaid navigation systems, it is better to explain respect which reference frame the navigation of the UAV is defined.

1.2 Reference frame

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 ([5]), 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 $\hat{z}$. 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 ⟶ the angular distance with respect to the Greenwhich Meridian, that is the meridian of reference;
- latitude ⟶ the angular distance with respect to the Equatorial plane;
- altitude ⟶ the distance from the origin of the system to the vehicle.

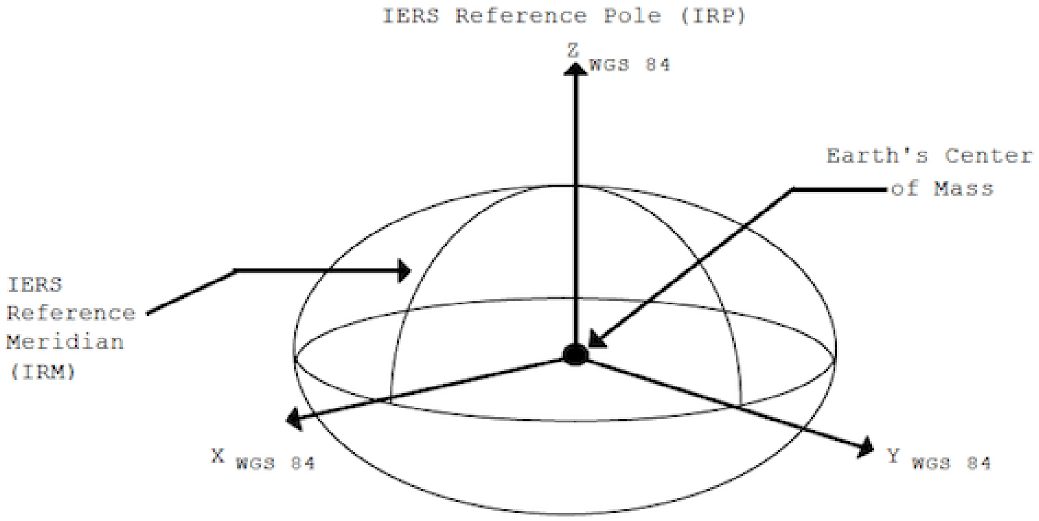


Figure 1.1: The WGS 84 reference frame [2]

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 [3])

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 $\hat{x}_b$ points downwards lying on the longitudinal plane while the $\hat{x}_b$ 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 makes it very useful for the writing of the dynamics equation thanks to its constant inertial characteristics. (For more information [3])

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

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

In a similar fashion, the other two rotation matrices are the following:

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

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

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:

$$\begin{aligned} R_{\psi}^{-1} &= R_X^T(\phi) && (1.5)\\ R_{\theta}^{-1} &= R_Y^T(\theta) && (1.6)\\ R_{\psi}^{-1} &= R_Z^T(\psi). && (1.7) \end{aligned}$$

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:

$$T_E^B = R_X(\phi)R_Y(\theta)R_Z(\psi) \tag{1.8}$$

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

$$T_E^B(\phi,\theta,\psi) = \begin{bmatrix} C_\theta C_\psi & C_\theta S_\psi & -S_\theta \\ S_\phi S_\theta C_\psi - C_\phi S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & S_\phi C_\theta \\ C_\phi S_\theta C_\psi - S_\phi S_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi & C_\phi C_\theta \end{bmatrix}, \tag{1.9}$$

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 mathematycally explained:

$$r_e = \begin{bmatrix} N \\ E \\ D \end{bmatrix}, v_e = \begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \end{bmatrix} = \dot{r}_e \tag{1.10}$$

$$v_b = T_E^B(\phi,\theta,\psi)v_e = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \tag{1.11}$$

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.

$$\alpha_e = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \tag{1.12}$$

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

(a) No gimbal lock

(b) Gimbal lock

Figure 1.2: Rotation singularity

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 informations 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:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}. \tag{1.13}$$

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

$$q = \begin{bmatrix} \rho \\ q_4 \end{bmatrix}, \tag{1.14}$$

where

$$\rho = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^T = e \sin(\frac{\theta}{2}), \tag{1.15}$$

$$q_4 = \cos(\frac{\theta}{2}) \tag{1.16}$$

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

$$A(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 + q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{1.17}$$

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

$$q^T q = 1, \tag{1.18}$$

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:

$$q^{-1} = \begin{bmatrix} -\rho \\ q_4 \end{bmatrix}. \tag{1.19}$$

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 $q^{''}$:

$$A(q^{''})A(q^{'}) \leftrightarrow q^{''} \otimes q^{'} \tag{1.20}$$

$$A(q^{''})A(q^{'})^T \leftrightarrow q^{''} \otimes q^{'-1} \tag{1.21}$$

with the quaternion product operator:

$$q^{''} \otimes q^{'} = \begin{bmatrix} -q_4^{''} & q_3^{''} & -q_2^{''} & q_1^{''} \\ q_3^{''} & q_4^{''} & q_1^{''} & q_2^{''} \\ q_2^{''} & -q_2^{''} & q_4^{''} & q_3^{''} \\ -q_1^{''} & q_1^{''} & -q_3^{''} & q_4^{''} \end{bmatrix} q^{'}. \tag{1.22}$$

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 [3].

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.

Advantages

- High data rated and high bandwidths;
- completely independent from the external world, do 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.

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 20th 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).

1.4.3 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. 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

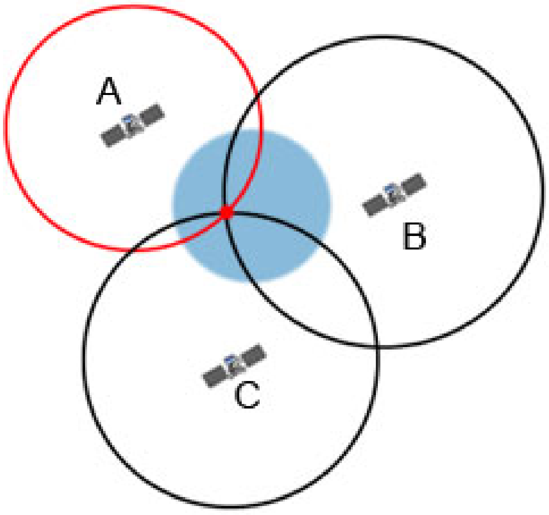


Figure 1.3: Trilateration of satellites

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 [4].

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 on what user we are, we are authorized to receive a particular signal.

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 [4].

Below, pros and cons of this navigations system are summerized.

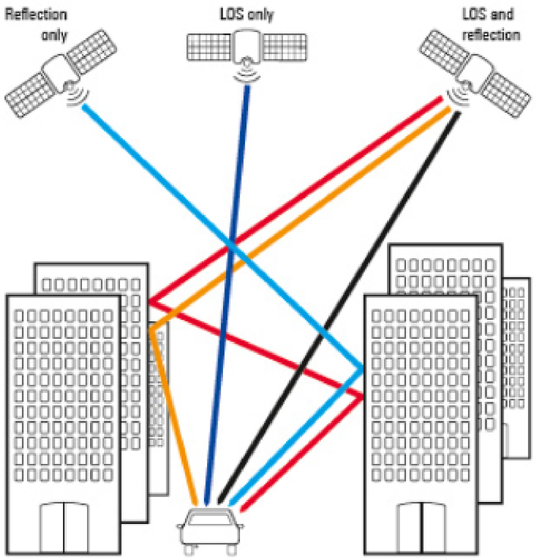


Figure 1.4: Multipath signals

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.

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.

1.4.4 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.

Advantages

- Position and velocity information indoor and in cluttered environment;
- completely independent navigation system;
- data rate higher than data from GPS.

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

This work of thesis focuses on the navigation of a UAV based on some sensors; the purpose of this chapter is to present the physics of the optical flow sensor. In particular it is shown how the optical flow is defined, how it works and how it is integrated in the navigation system. The section ends with the equation used to describe the measurements of the optical flow.

2.1 Definition of optical flow

The concept of optical flow (in the following denoted as 'f') 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 [6]. The Figures 2.1 and 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 f 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

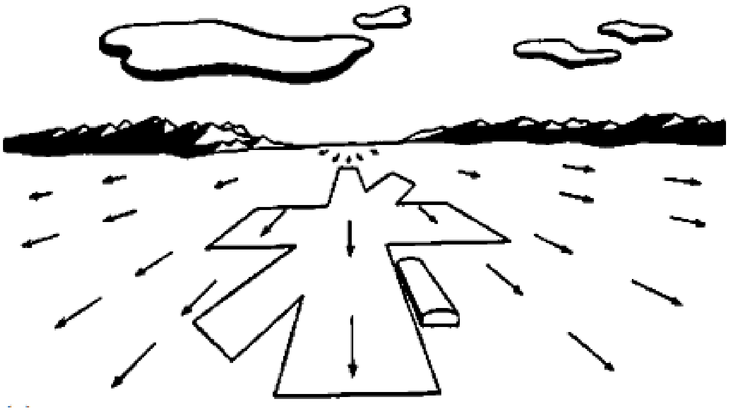


Figure 2.1: Optical flow (flying forward) [6]

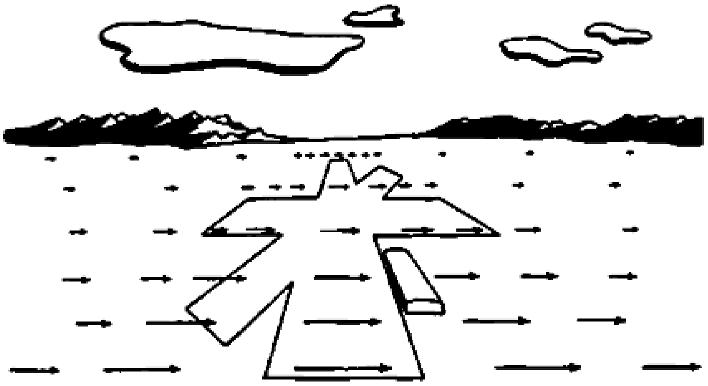


Figure 2.2: Optical flow (flying to the left) [6]

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 f to navigate efficiently during their flight maneuvers such as: terrain following, tunnel crossing, adjusting their speed in very cluttered environment. (See [7])

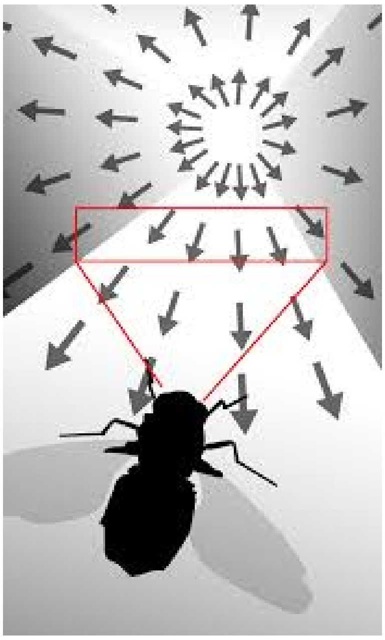


Figure 2.3: Insect optical flow

2.2 Computation of the optical flow

The application of this principle to UAV navigation needs the definition of the f 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 f vector by the sensor, chosen for this work, *i.e.*, the Px4Flow. In this case f estimation is based on the Sum of Absolute Differences (SAD) block matching [8]. It consists in the observation of a search area of ± 4 pixels with respect to the point given by the intersection of the 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).

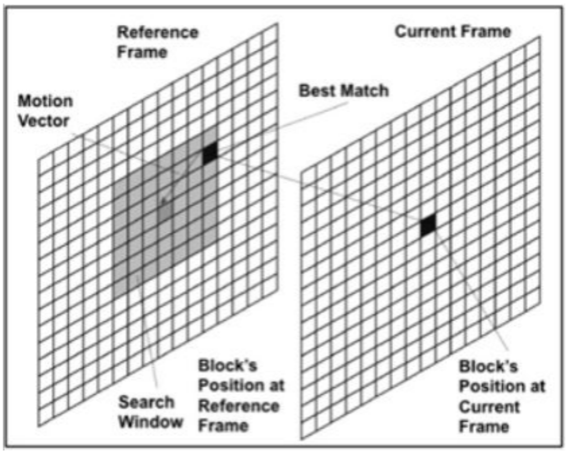

Reference Frame
Current Frame
Motion Vector
Best Match
Block's Position at Reference Frame
Search Window
Block's Position at Current Frame

Figure 2.4: Block matching by SAD

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 f vector computed into the sensor and use it as an 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 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). 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 f data in order

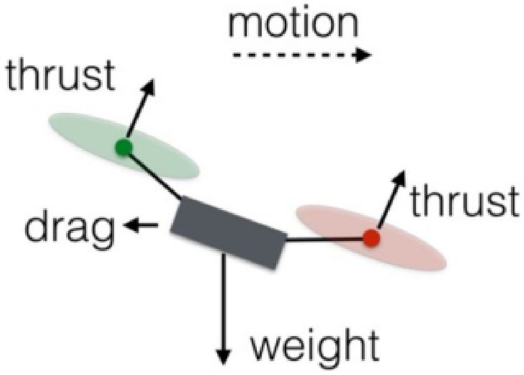


Figure 2.5: Multi-copter coupling between translational and rotational motion

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.

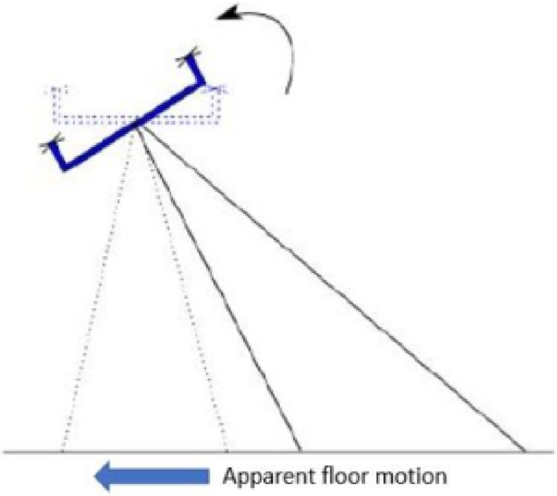


Figure 2.6: Apparent motion due to rotation

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 f 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 f with the other sensors, described in Figure 2.7, requires:

- f sensor for the f 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 f 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.

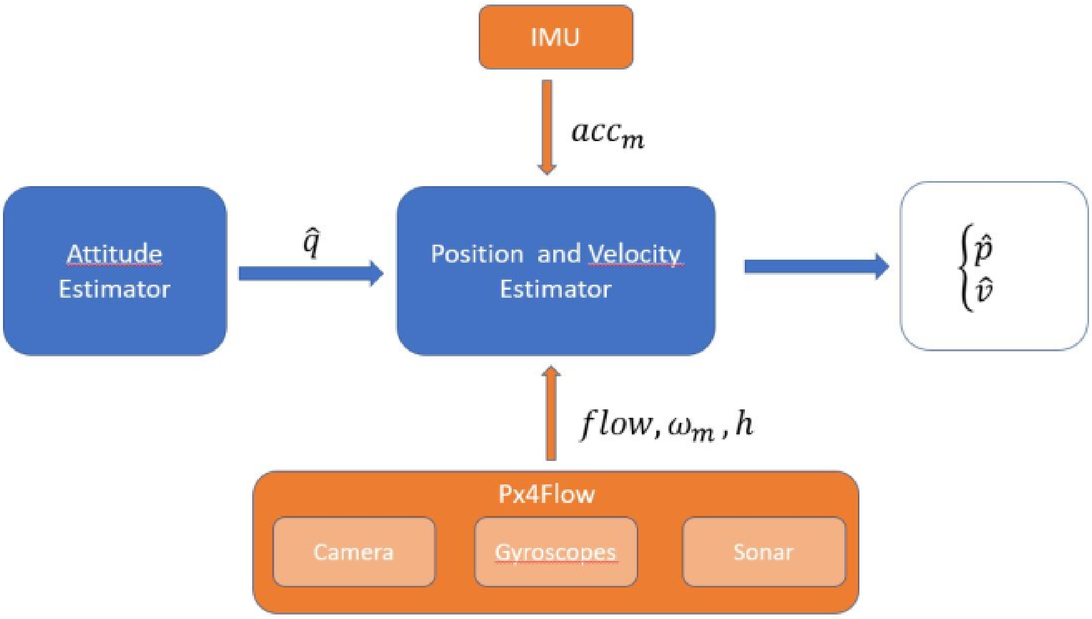


Figure 2.7: Scheme for the on board integration of the optical flow

2.4 Optical flow equations

What we need from the f sensor is an output useful to indirectly estimate the velocity. The f 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 f (explained in Chapter 1);
- knowledge of the firmware implemented in the sensor;
- comparison with the available velocity measurements coming from the GPS.

The f 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:

$$v_{floor} = \epsilon h \tag{2.1}$$

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

with $\epsilon = f$ vector; $v =$ UAV velocity; $v_{floor} =$ floor apparent velocity; $h =$ distance between the floor and the camera. Taking into account the reference system orientation and both axes of interest:

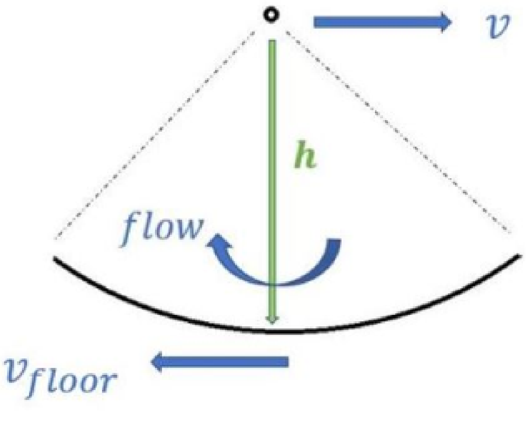


Figure 2.8: Optical flow and velocity relation

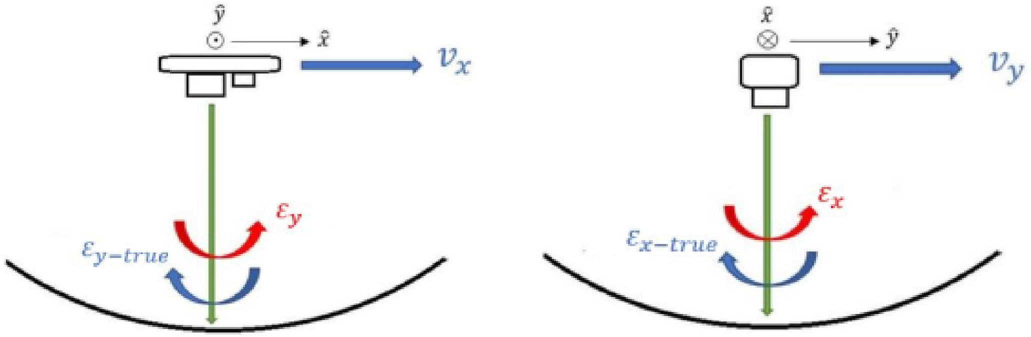
$$\begin{cases} v_x = \epsilon_y h \\ v_y = -\epsilon_x h \end{cases} \tag{2.3}$$

The different sign in the two equations above is due to the reference system as represented in Figure 2.9a and Figure 2.9b to the way used by the sensor to compute the f. In Figure 2.9a 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 f (the flowing images in front of the camera). If we consider the flow vector as an angular velocity around the $\hat{x}$ axis (ϵ_{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 f vector with a change in sign (ϵ in the figure), to allow the direct 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 the first equation of (2.3).
In Figure 2.9b 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_{true} in the figure); but according to the same reasoning, the f from the sensor is available with the opposite sign (ϵ_x in the figure), so we obtain the second equation in (2.3), 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 f 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 f 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:

$$\begin{cases} v_x = \frac{flow_y}{dt} h \\ v_y = \frac{-flow_x}{dt} h, \end{cases} \tag{2.4}$$

with $dt_{Px4Flow} = 0.1\ s$. 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 (2.4) 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 f 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:

$$\begin{cases} v_x = (\frac{flow_y}{dt} + \omega_y) h \\ v_y = -(\frac{flow_x}{dt} + \omega_x) h, \end{cases} \tag{2.5}$$



(a) Flow moving forward

(b) Flow moving to the right

Figure 2.9: Flow sense

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

The equations in (2.5) represent the ultimate relations between the velocities in $\hat{x}$ and $\hat{y}$ body axis and the f 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:

$$\begin{cases} flow_x = (-\frac{v_y}{h} - \omega_y)dt \\ flow_y = (\frac{v_x}{h} - \omega_x)dt, \end{cases} \tag{2.6}$$

Chapter 3

Kalman Filter for position and velocity estimation

In this chapter the algorithm of the Kalman filter that has been implemented, to obtain position and velocity estimates, based on GPS (for outdoor flight) or MoCap (for indoor flight) and f measurements is explained.

3.1 Mathematical model

The mathematical model exploited to describe the system, is a kinematic model based on the relationship among position, velocity and acceleration. The kinematic model is written respect to the NED frame as explained in Chapter 1. The continuous description of the kinematic relationships is reported only for one dimension and it is as follows:

$$\begin{cases} \dot{p}_i = v_{i0} + v_i \\ \dot{v}_i = a_i \end{cases} \tag{3.1}$$

where p_i is the position, v_i the velocity and a_i the acceleration of the UAV along the axis 'i', t_0 the intial time while v_{i0} the initial condition of the velocity (which is considered null as will be explained in the following section "Implemented Kalman Filter").
In system (3.1) the acceleration has been considered as an input of the system, instead of its internal state like actually p and v; this implies that it is used as a datum. In particular the acceleration is provided by the accelerometer.
As already said, the kinematic equation is written in the NED reference, while the acceleration measurements are provided in BODY frame; then a rotation of the accelerations must be performed before their use.
After the rotation, the system (3.1) must be rewritten using the acceleration measure. The true acceleration a_i has the following relationship with the measured

one:

$$a_{im} = a_i + w_{ia} \tag{3.2}$$

or practically:

$$\Rightarrow a_i = a_{im} - w_{ia}, \tag{3.3}$$

where a_{im} is the measured acceleration while w_{ia} is the noise of the accelerometer. The system (3.1) becomes:

$$\begin{cases} \dot{p_i} = v_i \\ \dot{v_i} = (a_{im} - w_{ia}) \end{cases} \tag{3.4}$$

For the implementation of the filter algorithm, is necessary the state-space descritpion of the kinematic system in (3.4), this time considered for the three dimensions:

$$\dot{x} = Ax + Bu_a + B_d w_a. \tag{3.5}$$

In (3.5) x is the vector of the state variables; u the input vector; A the state space matrix; B the input matrix; B_d the matrix that connect the process disturbance to the state variables while w_a the process disturbance.

More in detail:

$$x = \begin{bmatrix} p_x \\ p_y \\ p_z \\ v_x \\ v_y \\ v_z \end{bmatrix}_{NED} ; \tag{3.6}$$

the input vector u_a:

$$u_a = \begin{bmatrix} a_{xm} \\ a_{ym} \\ a_{zm} \end{bmatrix}_{NED} ; \tag{3.7}$$

while the disturbance:

$$w_a = \begin{bmatrix} w_{xa} \\ w_{ya} \\ w_{za} \end{bmatrix}_{NED} . \tag{3.8}$$

About the matrices:

$$A = \begin{bmatrix} O_3 & I_3 \\ O_3 & O_3 \end{bmatrix}; \tag{3.9}$$

the input matrix:

$$B = \begin{bmatrix} O_3 \\ I_3 \end{bmatrix}; \tag{3.10}$$

while B_d:

$$B_d = \begin{bmatrix} O_3 \\ -I_3 \end{bmatrix}. \tag{3.11}$$

As it is possible to see from (3.6), (3.7) and (3.8) the subscript "NED" has been used; this to underline that the equation of the state variables is written in NED inertial frame as was explained in Chapter 1 and previously recalled. Now that the state variables equation has been written, remains the output equation. The output equations are those which link the available measurements to the state variables.
In particular for the GPS measures, as well as for the MoCap measures, the equation which link the measure to the state, along a generic axis is:

$$p_{mi} = p_i + v_{igps} \tag{3.12}$$

being p_{mi} the measure along the axis 'i' and v_{igps} the corresponding noise. Is rightful to recall that the state variables are described in the NED inertial frame, then the measurements from the GPS must be projected on that frame before their introduction in the filter.
About the measurements provided by the Px4Flow, the relation between f and velocity, is as shown in Chapter 2. In this case the choice of rotating the velocity state variable to the BODY frame rather than the otpical flow measurements to the NED frame, has been made; therefore in the relation between the f measures and the velocity state variable, a rotation must be included.
Moreover in the measurement equations of the optical flow, is necessary to keep into account the influence of the angular velocities on the f measurements; this has been made thank to a compensation.
Then, considering the aforesaid aspects the output equation is written in compact form as:

$$y = Cx + Du_w + D_d v. \tag{3.13}$$

The vector of the measurements y is made as follows:

$$y = \begin{bmatrix} p_{mx} \\ p_{my} \\ p_{mz} \\ flow_x \\ flow_y \end{bmatrix}; \tag{3.14}$$

and the compensation has been made by means of the vector u_w:

$$u_w = \begin{bmatrix} w_x \\ w_y \end{bmatrix}_{BODY}. \tag{3.15}$$

For the angular velocity measurements the gyroscopes mounted on board have been adopted. Actually a choice between two sensors has been made: those integrated on the IMU (inertial measurements unit) and those from the f sensor. The latter have been chosen and the reason is about their sampling frequency; in order to have a more comfortable implementation of the angular compensation, it is convenient to use measurements sampled at the same frequency as the flow vector; finally the vector v is that of the noises introduced in the sytem by the GPS or MoCap and Px4Flow sensors:

$$v = \begin{bmatrix} v_{gpsx} \\ v_{gpsy} \\ v_{gpsz} \\ v_{flowx} \\ v_{flowy} \end{bmatrix}. \tag{3.16}$$

For being the explanation of how is made the matrix C more clear, it is important to keep into account that the measurements of the f are made respect to a reference frame BODY while the state variable velocity is described respect to a NED reference frame; then in the definition of matrix C is necessary to rotate the velocity from the NED inertial frame to the BODY one, by using the quaternions (as shown in Chapter 1). Therefore, matrix C on the contrary of all the other matrices, is time-varying due to its dependence on the attitude matrix. In fact the attitude matrix is described through the time varying quaternion estimates, coming from an on board attitude estimator (Extended Kalman Filter); therefore at each time instant, C must be re-defined to ensure the chasing of the actual attitude of the UAV, with the purpose of rotating the velocity vector in the right way. At the basis of this argument, the assumption that the estimate of the attitude, given by the on board estimator is comparable to the true attitude, has been made:

$$C = \begin{bmatrix} I_3 & O_3 \\ O_{2x3} & C_{flow} \end{bmatrix}, \tag{3.17}$$

where

$$C_{flow} = \begin{bmatrix} 0 & -\dfrac{dt_{flow}}{h} \\ \dfrac{dt_{flow}}{h} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} A_{i2b}(t). \tag{3.18}$$

The Px4Flow firmware presents an implemented integration time which must be kept into account as explained in Chapter 2: dt_{flow}. Its value is 0.1 seconds; h is the distance from the ground to the belly of the UAV; the second matrix is a selector matrix, that has been used to link the flow vector only to the components of the velocity along the x-axis and y-axis, respectively v_x and v_y (see Chapter 2, equation 2.6); A_{i2b}(t) is the attitude matrix that permit the rotation of the velocity from the NED to the BODY frame, as said before. Therefore C is a time-varying matrix.
Finally the matrices D and D_d are respectively:

$$D = \begin{bmatrix} O_{3x2} \\ -dt_{flow} I_{2x2} \end{bmatrix} \tag{3.19}$$

and

$$D_d = \begin{bmatrix} I_{5x5} \end{bmatrix}. \tag{3.20}$$

The state space system is then the following one:

$$\begin{cases} \dot{x} = Ax + Bu_a + B_d w_a \\ y = C(t)x + Du_w + D_d v. \end{cases} \tag{3.21}$$

In the state space representation above the choice of splitting the input vector into two vectors has been made: one for the inputs acting on the state equation and one for those which work on the output equation. This have been done simply to get more comfort during the implementation of the filter.
Since the Kalman filter to be implemented, is a discrete algorithm, the (3.21) has been discretized yielding:

$$\begin{cases} x_k = \Phi x_{k-1} + \Gamma u_{ak-1} + G w_{k-1} \\ y_k = H_k x_k + D u_{wk} + D_d v_k. \end{cases} \tag{3.22}$$

The system (3.22) explains the evolution of the continuous system (3.21) at each step k. The first equation in the latter system, has been obtained by approximating the derivative $\dot{x}$ as:

$$\dot{x} \simeq \frac{x_k - x_{k-1}}{dt}, \tag{3.23}$$

where dt is the time step used in the filter algorithm. The vectors u_a and w_a remain the same, but described at the steps k.
By considering (3.23) the following relationships have been obtained: the transition matrix Φ is obtained by the state matrix A as:

$$\Phi = I_6 + dtA; \tag{3.24}$$

the discrete input matrix is:

$$\Gamma = dtB; \tag{3.25}$$

while the matrix G, which connect the process disturbance to the state variables is the following:

$$G = dtB_d. \tag{3.26}$$

The output equation of the system (3.22), from a practical point of view, is the same seen in (3.21), but in this case the time-varying quantities exist only at the steps k. Since $C(\mathrm{t})$ is a time-varying matrix, the new matrix H_k has been defined. H_k is $C(\mathrm{t})$ evaluated at each step k.

In this section, up to now, the descriptive equations of the UAV dynamics and of the measurements, have been derived by following these assumptions:

- the UAV dynamics is described by two state variables: p and v;
- a is an input of the system provided by the accelerometer;
- the measurements provided by the accelerometer and the GPS are rotated in NED frame, for being compatible with the state space variables, which are described in NED;
- in the measurement equations of the optical flow, the state variable velocity has been rotated in BODY frame, rather than the f measurements in NED frame, for convenience.

The main implications of these choices are:

1. the estimation problem is of two variables, instead of three;
2. the process disturbance w_a of the model is given by the accelerometer noise.

Since the acceleration has been considered as the only input for the state equation, it reprensents the only source of noise. Is rightful to underline that the noise of the accelerometer measurements is a random process, for this reason the state variables dynamics will contain, also a random behaviour. The random behaviour of the accelerometer noise is described by the standard deviation of its time history.

3.2 Implemented Kalman Filter

This section focuses on the practical implementation of the filter algorithm, with all the needs required by the reality.

3.2.1 Data pre-processing

Before using the sensors data as measurements and inputs for the filter, namely y and u respectively, their rotation into the appropriate frame of reference must be executed. More precisely two data sets need to be rotated:

- GPS data from the GEODETIC reference frame to the NED one;
- acceleration data from the BODY frame to the NED frame.

Another data set, about which is necessary to spend an explanation is that of the angular velocity, used for the compensation of the f measurement. The Px4Flow sensor, as stated in Chapter 2, provides an angle and not an angular velocity, then in order to use its gyroscope output in the form of an angular velocity (rad/s), it has been divided by dt_{flow}. With this approach the implemented model can be likewise used with the measurements of the IMU; then there is not the necessity to change, in dependence on the sensor used, the matrix H_{flow}.
The last data set that needs an adjustment is that of the ground distance. That measure is provided by the sonar integrated in the Px4Flow, and it appears in matrix H_{flow}. Unfortunately the ground distance is at the denominator (see equation (3.18)), so to avoid singularity problems due to $h = 0\ m$, a lower bound to this measurement, equal to $h_{min} = 0.3\ m$ (actually the value that represents also the minimum operative distance from an observed surface, for the lens of the Px4Flow camera) has been imposed.

3.2.2 Initialization

Actually for some flights, an initial condition different from zero is likely; in that case a possible option is to use the initial value of the time history to be estimated. The initial covariance P_0, is used to express the confidence in the initial state vector; more trust in the initial guess and smaller the initial covariance values will be; on the contrary less trust in the initial guess is transleted in a bigger initial covariance value. The effect of the chosen P_0, or more properly that has been used as consequence of x_0, is clear keeping into account the generic discrete equation of the Kalman gain (see *e.g.* the equations (3.38) and (3.41) of the sub-section 3.2.4: "Multi-rate management: sequential update"); in fact large values of the state covariance matrix implies 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:

$$x_0 = 0_{6\times1}, \qquad P_0 = 10^{-6} I_6. \tag{3.27}$$

3.2.3 Tuning

This phase consists into choosing the appropriate values for the covariance matrices Q and R. It is an issue that has been studied extensively in the literature but there is not an exact systematic approach to realize it. It depends on the knowledge of the model by the designer, and also on the choices made by him. For a better understanding of the approach that has been followed, here there is a recall of what Q and R are:

- Q is the covariance matrix related to process noise;
- R is the covariance matrix related to measurement noise.

Generally, matrix R is more simple to be found, since it is directly linked to the data from the sensors. Conversely matrix Q has a slighlty more complicated computation. Beginning by R, which has a more immediate description, its structure is shown in equation (3.28):

$$R = \begin{bmatrix} R_{gps} & 0_{3x2} \\ 0_{2x3} & R_{flow} \end{bmatrix}, \tag{3.28}$$

being:

$$R_{gps} = \begin{bmatrix} \sigma^2_{gps_x} & \cdots & 0 \\ \vdots & \sigma^2_{gps_y} & \vdots \\ 0 & \cdots & \sigma^2_{gps_z} \end{bmatrix}, \qquad R_{flow} = \begin{bmatrix} \sigma^2_{flow_x} & 0 \\ 0 & \sigma^2_{flow_y} \end{bmatrix}, \tag{3.29}$$

where σ_*^2 stands for the variance of the measurement '$*$', denoted in the following also as $Var(*)$. Note that matrix R is diagonal, and this is due to the assumption that the noise of a certain measurement, does not affect the others; namely it adds uncertainty only to the output correlated to that measurement. Since the problem had been formulated as a discrete-time problem, if the computation of the sensor variances is made in continuous-time domain, is necessary to scale R as follows:

$$R_k = R/dt. \tag{3.30}$$

Now, how the matrix Q has been defined is explained. The process noise, presents more difficulties due to the fact that is necessary to discretize the continuous covariance matrix Q_w. This is made by the following integration:

$$Q_k = cov(w_k) = \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)} G Q_w G^T e^{A^T(t_{k+1}-\tau)} d\tau, \tag{3.31}$$

with

$$Q_w = \sigma_{acc}^2(\tau), \tag{3.32}$$

where τ is the time variable of the convolution integral and σ_{acc}^2 the variance of the accelerometer channel considered in the computation. Note that equation (3.31) involves matrices from the continuous-time state space model; is convenient to report them here:

$$\dot{x} = Ax + Bu + Gw, \tag{3.33}$$

which in detail is written as

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_m + \begin{bmatrix} 0 \\ -1 \end{bmatrix} w_a. \tag{3.34}$$

By solving equation (3.31), the discrete covariance matrix Q_k is obtained for the single axis, and is equal to:

$$Q_k = \begin{bmatrix} \dfrac{\Delta t^3}{3} & \dfrac{\Delta t^2}{2} \\ \dfrac{\Delta t^2}{2} & \Delta t \end{bmatrix} \sigma_{acc}^2, \tag{3.35}$$

where Δt is the interval between two consecutive discrete-time instants. As it is possible to see from the resulting Q_k, the variance of the accelerometer affects, not only the velocity variable, but also the position variable, on the contrary of what the kinematic relation states. This is the effect of the measurement covariance matrix discretization.
At this point the problem is to identify experimentally the variance of the accelerometer for each axis. The procedure and the results will be shown in Chapter 6.

3.2.4 Multi-rate management: sequential update

Actually in this thesis, a large group of sensors has been used in the filtering process, and this implies to handle a multi-rate problem. In detail the sensors have the following sampling frequencies:

1. accelerometer sampled at 125 *Hz*;
2. Px4Flow sampled at 10 *Hz*;
3. GPS sampled at 5 *Hz*.

A possible path to follow, for the management of multi-rate problems, which has been used in this work, is the so called sequential update approach. It consists in:

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

Below is reported the structure of the prediction and the correction process:

- prediction step:

$$\hat{x_k}(-) = \Phi_k \hat{x}_{k-1}(+) + \Gamma_k u_k, \tag{3.36}$$

$$P_k(-) = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T(+) + Q_{k-1}. \tag{3.37}$$

- GPS correction step:

$$K_{gpsk} = P_k(-)H_{gpsk}^T[H_{gpsk}P_k(-)H_{gpsk}^T + R_{gpsk}]^{-1}, \tag{3.38}$$

$$P_k(+) = [I - K_{gpsk}H_{gpsk}]P_k(-)[I - K_{gpsk}H_{gpsk}]^T + K_{gpsk}R_{gpsk}K_{gpsk}^T, \tag{3.39}$$

$$\hat{x_k}(+) = \hat{x_k}(-) + K_{gpsk}(y_k - \bar{H}_{gpsk}\hat{x}_{1:3k}(-)), \tag{3.40}$$

where H_{gpsk} is the sub-matrix oh $H(i,j)$, with $i\ = 1,\cdots,3$ and $j\ = 1,\cdots,6$; $\bar{H}_{gpsk}$ is the sub-matrix of *H(i,j)* with $i\ = 1,\cdots,3$ and $j\ = 1,\cdots,3$; notice that the innovation error (element that multiply the K_{gpsk}) in equation (3.40) is executed using only the state vector components related to the position.

- f correction step:

$$K_{ofk} = P_k(-)H_{ofk}^T[H_{ofk}P_k(-)H_{ofk}^T + R_{ofk}]^{-1}, \tag{3.41}$$

$$P_k(+) = [I - K_{ofk}H_{ofk}]P_k(-)[I - K_{ofk}H_{ofk}]^T + K_{ofk}R_{ofk}K_{ofk}^T, \tag{3.42}$$

$$\bar{y_k} = y_k - Du_w, \tag{3.43}$$

$$\hat{x_k}(+) = \hat{x_k}(-) + K_{ofk}(\bar{y_k} - \bar{H}_{ofk}\hat{x}_{4:5k}(-)), \tag{3.44}$$

where H_{ofk} is the sub-matrix oh $H(i,j)$, with $i\ = 4,5$ and $j\ = 1,\cdots,6$; $\bar{H}_{ofk}$ is the sub-matrix of *H(i,j)* with $i\ = 4,5$ and $j\ = 4,\cdots,6$; notice that the innovation error (element that multiply the K_{ofk}) in equation (3.44) is executed using only the state vector components related to the velocity in x and y axes.

The structure just explained, consists of a prediction of the state vector, at the rate of the accelerometer, with a correction based on the availability of the measurements provided by the GPS or MoCap and f sensors, dependent on their rate.
With the purpose to be more clear, the sequential update approach is illustrated with the diagram in Figure 3.1, where t_{KF} is the time vector which represents the time history of the filtering; i is the index that indicates the instant at which the filter is operating; t_y reprensents the time history of the measurement and j is the time instant of t_y, at which is made the test of availability. Note that the time histories are already provided, in other words the filter is working not in real time. In the correction step block, $y(t_y(j))$ is the measure taken correspondent to the time instant which passed the availability test of the measure, *i.e.* $t_y(j)$.

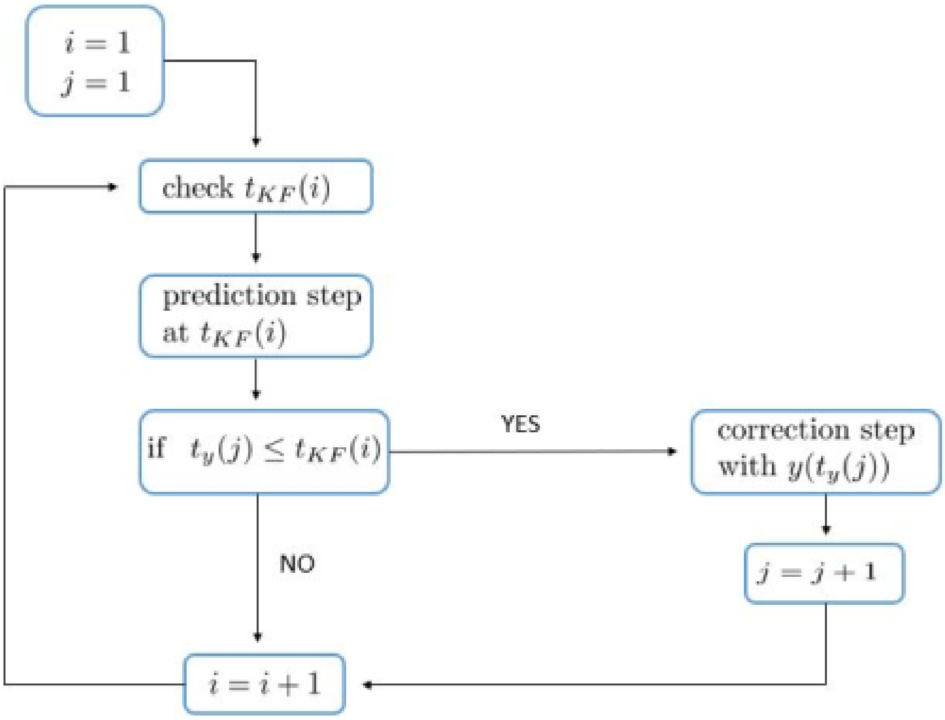

$i = 1$
$j = 1$
check $t_{KF}(i)$
prediction step at $t_{KF}(i)$
if $t_y(j) \leq t_{KF}(i)$
YES
correction step with $y(t_y(j))$
$j = j + 1$
NO
$i = i + 1$

Figure 3.1: Measurements availability check

Chapter 4

Augmented Kalman Filter

This chapter wants to explain why the state-space system has been augmented, and what other variables have been used to make possibile this request. The mathematical treatment is shown and then how becomes the final description of the system to be filtered, namely which state variables are used and which from them are filtered.

4.1 Mathematical model

In Chapter 3 the state-space description of the UAV dynamics, and the Kalman filtering algorithm have been shown. In that discussion, the noises of the process and those of the measurements have been introduced by means of a constant value, the variance, which it has been considered as a sufficiently complete characterization of the noises model; in other terms the definition of Q_k and R_k has been made by considering the noises as white noises. Actually each sensor for itself nature, has its proper dynamics which translates into a dynamical behaviour of the noise that it provides. For this reason is interesting to understand how becomes the description of the system, by keeping into account the dynamics of the noises, and certainly how behaves the filtering process of the new system which has been augmented.

4.1.1 Description of the noise dynamics

To understand how the noise has been treated, with the aim of obtaining the description of its dynamics, the passage from time-domain to frequency-domain is necessary. In fact in the frequency domain is more natural to analyze and feature the noise behaviour. In particular having the measurement data set m, from the sensor, it has been treated like this: (actually in this part here, when the word 'measurement' is used, it refers to the measurements resulting from those particular flights, made ad-hoc, to provide a measure of the noise. In particular these fligths don't contain important flight dynamics, since they have been chosen exactly to isolate the noise. With the purpose to be more clear, an example of flight made ad-hoc to asses the noise of an accelerometer is made: "hover"; in fact in hover the dynamics of the flight practically is not present, while only the noise is provided. In Chapter 6 there will be the presentation of the performed flights for the characterization of each sensor):

- the power spectral density of m, PSD_m (in the following denoted as PSD_* to indicate the power spectral density of the variable "$*$") has been calculated.

Generally there are two possibilities:

1. the PSD has the same amplitude for each frequency, or approxiamtely like that, then that noise behaves as white noise;
2. the PSD changes its amplitude along the frequency, and in this case it is not a white noise.

In the first case, a description like that presented in the previous chapter (Chapter 3), is satisfying; but in the second one no more, and the formulation of the problem, to achieve the drafting of the algorithm, becomes less trivial.
Practically what it has been made is to transform the not white noise in a white noise by means the addition of a new system, which has this particularity:

$$PSD_m(j\omega) = |H(j\omega)|^2 \, PSD_{wn}(j\omega), \tag{4.1}$$

being PSD_{wn} the *PSD* of a white noise, which for its definition is constant (for convenience is considered unitary), and $H(j\omega)$ the frequency response of that particular system. Being the PSD_{wn} unitary:

$$PSD_m(j\omega) = |H(j\omega)|^2 . \tag{4.2}$$

In other terms the dynamics of the noise has been treated as a white noise filtered by the system whose frequency response is $H(j\omega)$. The new system, so defined, is called in some literatures "Shape filter", and also in this work of thesis; namely it takes as input a white noise, and shape it to give the noise of interest. Clearly all this is referred to the frequency domain, since the problem now is treated using *PSDs*, as state in (4.1).
The next step made is the in time domain realization of the aforesaid system. As said just before, for the realization of the shape filter a unitary PSD of the white noise has been imposed. The filter that has been used for the shape filter state-space realization, is of the second order (for reasons that will be explained in the Chapter 6):

$$\begin{cases} \dot{x_1} = A_{11}x_1 + A_{12}x_2 + B_1 w_n \\ \dot{x_2} = A_{21}x_1 + A_{22}x_2 + +B_2 w_n \\ y_F = C_1 x_1 + C_2 x_2 + D w_n \end{cases} \tag{4.3}$$

where, x_1 and x_2 are the internal state variables of the shape filter (*i.e.* the dynamics of the noise of each sensor is decribed by two state variables); A_* are the components of the state matrix; B_* are the components of the matrix that link the input to the state variables; the input as explained before is the white noise, while the output y_F is the data set measurement, m; C_* are the components of the matrix that connect the state variables to the output and D is the scalar, which link the input to the output.
Note that for each sensor, a shape filter must be developed, and that the measurements data set from accelerometer and GPS sensors, must be rotated from the BODY and the GEODETIC frames respectively, to the NED frame, while those from the f sensor, are kept in the BODY frame as explained in the previous chapter.

4.1.2 Augmented system

In the previous subsection the method through which the noise dynamics of a sensor has been defined, it has been shown. Since eight measures are used (three from the GPS, two from f sensor and three from the accelerometers), the just seen process has been repeated eight times (and in case of indoor flights the MoCap is used, then the process is performed for the correpondent three measures). The next step that must be carried out is the addition of the noise dynamics state space description, to the system of the UAV seen in the previous chapter (Chapter 3, in (3.21)).
To explain better the method, the state space equations of the UAV dynamics are recalled:

$$\begin{cases} \dot{x} = Ax + Bu_a + B_d w_a \\ y = C(t)x + Du_w + D_d v. \end{cases} \tag{4.4}$$

In (4.4) the elements of w_a and v have been substituted by the output of the sensor shape filter correspondent to the noise considered (remember that the shape filters for how they were defined, return as output the noises of interest).

- For each accelerometer the substitution is:

$$w_{ia} = y_{Fia} = C_{Fia} x_{Fia} + D_{Fia} w_n, \tag{4.5}$$

where w_{ia} is the noise provided by the accelerometer along the axis 'i'; y_{Fia} is the output of the shape filter corresponding to the accelerometer along 'i', while C_{Fia}, x_{Fia} and D_{Fia} are the quantities involved in the shape filter output equation, as shown in (4.3).

- Similarly for each channel of the GPS, the substitution is:

$$v_{igps} = y_{Figps} = C_{Figps}x_{Figps} + D_{Figps}w_n, \tag{4.6}$$

- and for each Px4Flow channel is:

$$v_{iflow} = y_{Fiflow} = C_{Fiflow}x_{Fiflow} + D_{Fiflow}w_n, \tag{4.7}$$

being v_{igps} the components of v due to the GPS noises, and v_{iflow} those due to the Px4Flow noises.
Now the augmented system will be described following this nomenclature:

$$\begin{cases} \dot{x}_{aug} = A_{aug}x + B_{aug}u_a + B_{augd}w_n \\ y_{aug} = C_{aug}x + D_{aug}u_w + D_{augn}w_n \end{cases} \tag{4.8}$$

As is shown in the latter system, the inputs of the system to be filtered, are no more w_a and v, but only white noises, as wanted.
Considering all the sensors used (three accelerometers: x, y and z; GPS has three channels: x, y and z, while f two channels: x and y), and that the shape filter realization is of the second order, the state variables vector becomes:

$$x_{aug} = [p_x, p_y, p_z, v_x, v_y, v_z, x_{1accx}, x_{2accx}, x_{1accy}, x_{2accy}, x_{1accz}, x_{2accz}, x_{1gpsx}, \tag{4.9}$$
$$x_{2gpsx}, x_{1gpsy}, x_{2gpsy}, x_{1gpsz}, x_{2gpsz}, x_{1ofx}, x_{2ofx}, x_{1ofy}, x_{2ofy}]^T, \tag{4.10}$$

i.e with twentytwo states: six of the UAV dynamics and sixteen due to the dynamics of the noises. The state matrix of the augmented system, becomes:

$$A_{aug} = \begin{bmatrix} A_{6x6} & \tilde{C}_{6x16} \\ O_{16x6} & A_{sf16x16} \end{bmatrix}_{22x22}; \tag{4.11}$$

on the diagonal, there are the state matrix of the system not augmented and that resulting from the shape filters (in the following when $*_{sf}$ is adopted, is for indicating the quantity '$*$' of the shape filter) with:

A_{6x6} as seen previously (in Chapter 3, equation (3.9)), which is the state matrix of the system not augmented;

$$\tilde{C} = \begin{bmatrix} O_{3x6} & O_{3x10} \\ -C_{sfacc3x6} & O_{3x10} \end{bmatrix}_{6x16}, \tag{4.12}$$

where C_{sfacc} is as follows:

$$C_{sfacc} = \begin{bmatrix} \boxed{C_{accx}} & O_{1x2} & O_{1x2} \\ O_{1x2} & \boxed{C_{accy}} & O_{1x2} \\ O_{1x2} & O_{1x2} & \boxed{C_{accz}} \end{bmatrix}; \tag{4.13}$$

as seen in the description of the noise dynamics (4.3), C_* are the matrices of the shape filter, that connect the state variables to the output. In (4.11) matrix $\tilde{C}$ appears in the first row of the matrix A_{aug}, because the noise of the accelerometers appears in the state variables equation of the UAV; while A_{sf} is a blockdiagonal matrix, composed as following:

$$A_{sf} = \begin{bmatrix} \boxed{A_{accx}} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} \\ O_{2x2} & \boxed{A_{accy}} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} \\ O_{2x2} & O_{2x2} & \boxed{A_{accz}} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} \\ O_{2x2} & O_{2x2} & O_{2x2} & \boxed{A_{gpsx}} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} \\ O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & \boxed{A_{gpsy}} & O_{2x2} & O_{2x2} & O_{2x2} \\ O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & \boxed{A_{gpsz}} & O_{2x2} & O_{2x2} \\ O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & \boxed{A_{ofx}} & O_{2x2} \\ O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & O_{2x2} & \boxed{A_{ofy}} \end{bmatrix}_{16x16}, \tag{4.14}$$

being A_* the state matrix of the shape filter realization, seen in (4.3), for each sensor.
The matrix that connect the input to the state variables becomes B_{aug}:

$$B_{aug} = \begin{bmatrix} B \\ O_{16x3} \end{bmatrix}_{22x3} \tag{4.15}$$

being B the same shown in (3.10).

Matrix B_{augd} is made as is shown below:

$$B_{augd} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -D_{accx} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -D_{accy} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -D_{accz} & 0 & 0 & 0 & 0 & 0 \\ \boxed{B_{accx}} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} \\ O_{2x1} & \boxed{B_{accy}} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} \\ O_{2x1} & O_{2x1} & \boxed{B_{accz}} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} \\ O_{2x1} & O_{2x1} & O_{2x1} & \boxed{B_{gpsx}} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} \\ O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & \boxed{B_{gpsy}} & O_{2x1} & O_{2x1} & O_{2x1} \\ O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & \boxed{B_{gpsz}} & O_{2x1} & O_{2x1} \\ O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & \boxed{B_{ofx}} & O_{2x1} \\ O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & O_{2x1} & \boxed{B_{ofy}} \end{bmatrix}, \tag{4.16}$$

being B_* the matrices that link the white noise to the state variables, as seen in (4.3), for each sensor, while D_* the scalars that connect the white noise to the output, but only for the accelerometers.
About the output equation in (4.8), C_{aug} is made as follows:

$$C_{aug} = \begin{bmatrix} I_{3x3} & O_{3x3} & O_{3x5} & C_{sfgps} & O_{3x4} \\ O_{2x3} & C_{flow} & O_{2x5} & O_{2x6} & C_{sfflow} \end{bmatrix}; \tag{4.17}$$

where $C_{aug}(1:5, 1:6)$ is equivalent to the matrix, already seen in (3.21), while C_{sfgps} and C_{sfof} are two block diagonal matrices:

$$C_{sfgps} = \begin{bmatrix} \boxed{C_{gpsx}} & O_{1x2} & O_{1x2} \\ O_{1x2} & \boxed{C_{gpsy}} & O_{1x2} \\ O_{1x2} & O_{1x2} & \boxed{C_{gpsz}} \end{bmatrix}, \tag{4.18}$$

and

$$C_{sfof} = \begin{bmatrix} \boxed{C_{gpsx}} & O_{1x2} \\ O_{1x2} & \boxed{C_{gpsy}} \end{bmatrix}, \tag{4.19}$$

where C_* is the matrix that link the state variables to the output of the realization of the shape filter, for the sensor considered, following as always the

symbols adopted in (4.3).
The matrix D_{aug} is made like this:

$$D_{aug} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -dt_{flow} & 0 \\ 0 & -dt_{flow} \end{bmatrix}. \tag{4.20}$$

In the end the matrix D_{augn}, which connect the w_n to the output of the augmented system, is shown:

$$D_{augn} = \begin{bmatrix} 0 & 0 & 0 & D_{gpsx} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D_{gpsy} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{gpsz} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D_{ofx} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_{ofy} \end{bmatrix}, \tag{4.21}$$

where D_* refers again to the sensor used.

4.1.3 Discrete-time domain description

In the past subsections a model to describe the dynamics of the noises, provided by the sensors, has been presented thank to the use of shape filters. Moreover how the initial system has been augmented by the shape filter dynamics, to obtain a system which noises were no more, with their dynamics but simple white noises, has been shown; to make more simple the treatment and for handling better the problem, the shape filters realization has been made in continious domain. Since the implementation of the filter is made through a discrete algorithm, then is necessary to pass to the discrete description.
The model which dynamics must be filtered with the Kalman filter, is recalled from the previous subsection:

$$\begin{cases} \dot{x}_{aug} = A_{aug}x + B_{aug}u_a + B_{augd}w_n \\ y_{aug} = C_{aug}x + D_{aug}u_w + D_{augn}w_n. \end{cases} \tag{4.22}$$

To pass from this continious form to the discrete one, the same approach already seen in the previous chapter (Chapter 3), to discretize the initial system of the only UAV dynamics has been used, *i.e.* :

$$\begin{cases} x_{augk} = \Phi_{augk-1}x_{augk-1} + \Gamma_{augk-1}u_{ak-1} + G_{augk-1}w_{nk-1} \\ y_{augk} = H_{augk}x_{augk} + D_{augk}u_{wk} + D_{augn}w_{nk}. \end{cases} \tag{4.23}$$

The first equation represents the state-space equation in (4.22) dicretized, while the second one, represents the measured outputs and how they are coupled

with the states. Clearly the outputs of the augmented system are the same of the not augmented one, since those measures are the only available.
Below are reported how the elements of (4.23) have been obtained.
x_{aug} is the same vector seen before in (4.10) but existent only at the steps k; Φ_{aug} is obtained as is shown below:

$$\Phi_{aug} = I_{22x22} + A_{aug}dt, \tag{4.24}$$

where I_{22x22} is used to indicate a diagonal matrix of 1, with 22 x 22 size, while dt is the discrertization step. Γ_{aug} is the matrix obtained as:

$$\Gamma_{aug} = B_{aug}dt. \tag{4.25}$$

The matrix G_{aug} represents the input matrix, which has been obtained as:

$$G_{aug} = B_{augd}dt. \tag{4.26}$$

y as before, is composed by these components:

$$y = \begin{bmatrix} p_{mx} \\ p_{my} \\ p_{mz} \\ flow_x \\ flow_y \end{bmatrix}. \tag{4.27}$$

H_{aug} is the matrix C_{aug}, but considered only at the steps k, and the thing to keep into account about it, is that it is a time varying matrix, due to its dependence on the Euler angles, as said before. Practically D_{aug} and D_{augn} remain the same.

4.2 Implemented augmented Kalman filter

Since the mathematical model of the system to be filtered, has been obtained, it is time to focus on the practical issues of the implementation of the filter aglorithm, with all the needs required by the reality.

Since a lot of aspects have been covered by the previous chapter (Chapter 3, section 3.2), in this section it has been decided to analyze more in detail only those aspects that are different; therefore the aspects in common are visited rapidly.

4.2.1 Data pre-processing

Also for the augmented Kalman filter some sensors need a special treatment. More precisely two data sets need to be rotated as said in the previous chapter: those that are provided by the GPS and the accelerometer. Moreover about the angular velocity is necessary to keep into account the fact that the altitude appears at the denominator, in matrix H_{flow}; then in case of $h = 0\ m$, to avoid singularities, a lower bound has been imposed, equal to $h_{min} = 0.3\ m$.
The augmented filter, for running, needs the knowledge of the dynamics of the noises. In this part the shape filters have been realized, by using the data set resulting by the flights made ad-hoc to emphasize the noises respect to the dynamics of the flights. In particular having the PSD of the measures, that second order system which frequency response is similar enough to the PSD has been detected. This work for each set of measurements has been performed: for the three measures of the accelerometer, the three of the GPS (or of the MoCap if indoor flights are tested) and the two of the Px4fFlow.

4.2.2 Initialization

About the inizializaton of the filter is necessary to impose the initial condition, from which the filter must start to run. Respect to the system in which only the dynamics of the UAV is described, there is not differencies, if not for the fact that the state variables are now twentytwo.

$$x_{aug0} = 0_{22\times 1}, \qquad P_{aug0} = 10^{-6} I_{22}. \tag{4.28}$$

4.2.3 Tuning

In this part here, respect to the just shown subsections, is necessary to spend more attention. In fact respect to the previous chapter (Chapter 3 section 3.2.3), for the augmented system, also the dynamics of the noises must be kept into account.

Starting from Q_{aug}, below is shown the integral used to compute its discretization:

$$Q_{augk} = \int_{t_k}^{t_{k+1}} e^{A_{aug}(t_{k+1}-\tau)} B_{augd} Q_{wn} B_{augd}^T e^{A_{aug}^T(t_{k+1}-\tau)} d\tau, . \tag{4.29}$$

As said before, for the realization of the shape filter a unitary variance has been imposed, therefore Q_{wn} is a unitary diagonal matrix 8 by 8 (eight is the number of measures involved: three accelerometers, three GPS channels and two Px4Flow channels). Then the integral that has been solved is:

$$Q_{augk} = \int_{t_k}^{t_{k+1}} e^{A_{aug}(t_{k+1}-\tau)} B_{augd} B_{augd}^T e^{A_{aug}^T(t_{k+1}-\tau)} d\tau, . \quad (4.30)$$

Since the augmented system has more complex structure, due to the presence of the dynamics of the noises, the integral just shown, has been solved imposing an approximation: in particular considering the step size used by the Kalman filter algorithm, small enough (namely the quantity $dt = t_{k+1}$- t_k small enough), the integral has been approximated as follows:

$$Q_{augk} = cov(w_{nk}) = B_{augd} B_{augd}^T dt. \quad (4.31)$$

Is rightful to say, that the matrices used up to now, are those used to describe the augmented system (4.22) in the state space form, which is recalled here for more ease:

$$\begin{cases} \dot{x}_{aug} = A_{aug}x + B_{aug}u_a + B_{augd}w_n \\ y_{aug} = C_{aug}x + D_{aug}u_w + D_{augn}w_n \end{cases} \quad (4.32)$$

The way followed for discretizing R_{aug}, is similar:

$$R_{augk} = \int_{t_k}^{t_{k+1}} (C_{aug} e^{A_{aug}(t_{k+1}-\tau)} B_{augd} + D_{augn}\delta(t_{k+1}-\tau)) R_{wn} \\ (C_{aug} e^{A_{aug}(t_{k+1}-\tau)} B_{augd} + D_{augn}\delta(t_{k+1}-\tau))^T d\tau, \quad (4.33)$$

where the nomenclature used is the same seen in (4.32), and δ refers to the operator "delta of Dirac", namely the impulse function. As before, R_{wn} is a unitary 5 x 5 matrix (five is the number of sensors exploited for the estimation process: three of the GPS and two of the Px4Flow), and considering sufficiently small dt the equation yields:

$$R_{augk} = C_{aug} B_{augd} B_{augd}^T C_{aug}^T dt + C_{aug} B_{augd} D_{augn}^T + D_{augn} B_{augd}^T C_{aug}^T + \\ + D_{augn} D_{augn}^T \frac{1}{dt}. \quad (4.34)$$

In the implementation of the augmented Kalman filter, the first term after the equal, has not been considered, since it is very small respect to the other terms; the following two terms are considered only if the process noise and the measurements noise are considered correlated (as actually are, since as is shown in the system (4.32) the noise input of the state equation and that of the output equation, is the same. But in this case the implementation of the Kalman filter becomes more complex due to the correlation). Note that in case of correlation the covariance matrix R_{augk} becomes dependent on time, because of C_{aug} (which is a function of time as seen previously); in the end, the last term is the only that has been used to compute the covariance matrix R_{augk}.

4.2.4 Multi-rate management: sequential update

Also for the augmented system the same approach used previously considering only the dynamics of the UAV has been used: the sequential update method. It has been implemented with the purpose to handle the different rates at which the measures are provided to the filter. Practically the algorithm used in this section is the same shown in chapter 3, but with slight differencies to keep into account the presence of the news states, given by the shape filters. Moreover to make correctely the estimation process a compensation of the dyamics of the noises is necessary.
The structure of the process is reported below:

- prediction step

$$\hat{x}_{augk}(-) = \Phi_{augk}\hat{x}_{augk-1}(+) + \Gamma_{augk}u_k, \tag{4.35}$$

$$P_{augk}(-) = \Phi_{augk-1}P_{augk-1}\Phi^T_{augk-1}(+) + Q_{augk-1}, \tag{4.36}$$

- GPS correction step

$$K_{auggpsk} = P_{augk}(-)H^T_{auggpsk}[H_{auggpsk}P_{augk}(-)H^T_{auggpsk} + R_{auggpsk}]^{-1}, \tag{4.37}$$

$$\begin{aligned} P_{augk}(+) = [I - K_{auggpsk}H_{auggpsk}]P_{augk}(-)[I - K_{auggpsk}H_{auggpsk}]^T + \\ + K_{auggpsk}R_{auggpsk}K^T_{auggpsk}, \end{aligned} \tag{4.38}$$

$$\hat{x}_{augk}(+) = \hat{x}_{augk}(-) + K_{auggpsk}(y_k - (\bar{H}_{auggpsk}\hat{x}_{1:22k}(-) + \bar{H}_{augnoisegpsk}\hat{x}_{13:18k}(-))), \tag{4.39}$$

where $H_{auggpsk}$ is the sub-matrix of $H_{aug}(i,j)$, with $i = 1,\cdots,3$ and $j = 1,\cdots,22$; $\bar{H}_{auggpsk}$ is the sub-matrix of $H_{aug}(i,j)$ with $i = 1,\cdots,3$ and $j = 1,\cdots,3$, while $\bar{H}_{augnoisegpsk}$ is the sub-matrix of $H_{aug}(i,j)$ with $i = 1,\cdots,3$ and $j = 13,\cdots,18$ (state variables of the GPS noises shape filters, which are involved in the equation of the output for the three position measurements);

- f correction step

$$K_{augofk} = P_{augk}(-)H^T_{augofk}[H_{augofk}P_{augk}(-)H^T_{augofk} + R_{augofk}]^{-1}, \tag{4.40}$$

$$P_{augk}(+) = [I - K_{augofk}H_{augofk}]P_{augk}(-)[I - K_{augofk}H_{augofk}]^T + K_{augofk}R_{augofk}K^T_{augofk}, \tag{4.41}$$

$$\bar{y}_k = y_k - D_{aug}u_w, \tag{4.42}$$

$$\hat{x}_{augk}(+) = \hat{x}_{augk}(-) + K_{augofk}(\bar{y}_k - (\bar{H}_{augofk}\hat{x}_{4:5k}(-) + \bar{H}_{augnoiseofk}\hat{x}_{19:22k}(-))), \tag{4.43}$$

where H_{augofk} is the sub-matrix of $H_{aug}(i,j)$, with $i\ = 4{,}5$ and $j\ = 1{,}\cdots{,}22$; $\bar{H}_{augofk}$ is the sub-matrix of $H_{aug}(i,j)$ with $i\ = 4{,}5$ and $j\ = 4{,}\cdots{,}6$, while $\bar{H}_{augnoiseofk}$ is the sub-matrix of $H_{aug}(i,j)$ with $i\ = 4{,}5$ and $j\ = 19{,}\cdots{,}22$ (state variables of the Px4Flow noises shape filters, involved in the equation of the output for the two f measurements).
Then the principal differencies, respect to the only UAV dynamics estimation problem, are in the equations of the states update (4.39) and (4.43).

4.3 Conclusions

In this chapter it has been explained that as well as the dynamics of the UAV, that of the noises provided by the sensors has been considered. With this purpose intoducing the shape filters has been necessary, which realization has been made starting from the imposition of unitary variance. For each measure a second order filter has been adopted. Then the additon of the dynamics of the noises to the UAV dynamics, has been performed, giving an augmented system, which becomes the model used for the estimation problem. Then the new edition of the algorithm which includes the noise dynamics compensation, has been shown.

Chapter 5

Adaptive Kalman Filter

In this chapter is presented why an adaptive Kalman algorithm has been implemented , for the filtering problem. In particular the model used for the estimation and the algorithm implemented are shown. As in the other chapters, also in this one the mathematical model will be treated, all the assumptions made to describe it, and in the end the issues encountered in the real implementation.

5.1 Mathematical problem

In the implementation of the Kalman algorithm, an important quatity from which the performaces of the estimation, heavly depend is the covariance used for decribing the noise. About the noise is rightful to say that not always, indications to model it are available. For this reason an important aspect to keep in mind is how to handle the noise. In Chapter 3 the noises have been described as white noises, while in the previous one (Chapter 4) their dynamics has been considered; in this chapter another approach has been thought: an adaptive description of the covariance matrices. More precisely the matrices Q and R, for each step of the Kalman algorithm, on the basis of the availability of the measures given by the sensors, are updated according to a certain mathematical law.

5.1.1 Description of the noise

For the sake of being able to describe how Q and R are adapted, is necessary to recall the system to be filtered. As for the case of the not augmented system, in the case of the adaptive approach, the model adopted, includes only the dynamics of the UAV; more precisely six states: three for the position and three for the velocity. Then, there are not differencies respect to the description seen in Chapter 3. For more ease, the same is reported below:

$$x_k = \Phi_{k-1} x_{k-1} + \Gamma_{k-1} u_{ak-1} + G_{k-1} w_{k-1}, \tag{5.1a}$$

$$y_k = H_k x_k + D_k u_{wk} + v_k. \tag{5.1b}$$

The state space model just shown, is a description of the UAV dynamics respect to the NED frame, in which the acceleration is seen as an external input, as well as the the angular velocity, which appears in the equation of the outputs, (5.1b). This latter is used to compensate the measure of the f. The state variables equation, comes from the kinematic relationship among position, velocity and acceleration. Instead, the second one originates as conseguence of the knowledge of the sensors used: the GPS provides a direct measure of the position, while the Px4f gives a measure which has a particular relation with the velocity, as previously seen in Chapter 2. Note that all the variables involved in (5.1a) and (5.1b) are the same seen in the chapter 3.
Now that the system under examination is recalled, the method used to update the covariance matrices is presented.
Q_{ad} and R_{ad} in the following are adopted to indicate the covariance matrices Q and R respectively, subjected to the adaptive algorithm.
The method which has been used to adapt Q_{ad} at each iteration k, consists in this equation:

$$Q_{adk} = \alpha Q_{adk-1} + (1-\alpha)(K_{adk} d_k d_k^T K_{adk}^T), \tag{5.2}$$

where d_k is the innovation:

$$d_k = y_k - H_k \hat{x}_k^-, \tag{5.3}$$

in other terms it is the difference between the most modern measurement and its predective value; K_{adk} is the Kalman gain matrix of the adaptive algorithm; α is a term which on the basis of its value gives more weight to the old value of Q_{adk} or to the innovation. As will be shown, the value of α has been sought through an optimization algorithm.
Also R_{ad} is updated with a similar method:

$$R_{adk} = \alpha R_{adk-1} + (1-\alpha)(\epsilon_k \epsilon_k^T + H_k P_{adk}^- H_k^T), \tag{5.4}$$

where ϵ_k is the residual:

$$\epsilon_k = y_k - H_k \hat{x}_k^+, \tag{5.5}$$

namely the difference between the actual measurement and its estimated value, and finally P_{adk} is the covariance matrix of the state variables. Note that, since the covariance matrices Q_{ad} and R_{ad} are iterated, they need an initial value, or better, an initial guess from which the filter starts to run.
α as just said before, needs more attention in its definition.

The chioce of α

The choice of the value α has an important influence on the resulting preformances of the implemented estimator; in fact it is directly involved in the dynamics of the filter. The investigation of α has been made following these two points:

- fixing the initial guesses of Q_{ad} and R_{ad}, which in the following, will be called Q_{ad0} and R_{ad0} respectively;
- finding a quantity dependent on α to minimize.

The quantity that has been used in this work is the mean square error (in the follwing indicated as MSE) of the state variables. More precisely the adaptive filter has been run several times using a set of possible values of α (the section "Implemented Kalman Filter" of this chapter, will go more in deep about this aspect), and the α value which presents the minimum MSE of the state variables, has been considered the optimal one to be used in the implementation of the adaptive Kalman filter.

The choice of Q_{ad0}

The next passage to think about has been about the optimal choice also of Q_{ad0}. To make possible this request the following points have been respected:

- the optimal value of α has been selected and inserted into the adaptive algorithm;
- Q_{ad0} has been rewritten as:

$$Q_{ad0} = c_Q Q, \tag{5.6}$$

where c_Q is the value which will be sought to minimize the MSE, rather than directly the matrix Q_{ad0}; and Q is the matrix already encountered when the noises had been modeled as white noises (for more insights about the noise as white noise return to Chapter 3). Practically the initial guess Q_{ad0} becomes a fraction of Q, in dependence on the value c_Q;

- the adaptive algorithm has been run for a set of c_Q.

The optimal value of c_Q is that which minimizes the MSE.

The choice of R_{ad0}

The approach used to find the optimal R_{ad0} is the same seen in the previous point. Also in this case the following points have been respected:

- the optimal value of α it has been selected and inserted into the adaptive algorithm;
- R_{ad0} has been rewritten as:

$$R_{ad0} = c_R R, \tag{5.7}$$

where c_R is the value which will be sought to minimize the MSE, rather than directly the matrix R_{ad0}; and R is the matrix already encountered when the noises had been modeled as white noises;

- the adaptive algorithm has been run for a set of c_R.

The optimal value of c_R is that which minimizes the MSE. Actually the investigation of c_Q and c_R has been made concurrently; in other words c_Q and c_R chosen as optimal, are those that selected together, provide the minimum MSE of the state variables.

5.2 Implemented adaptive Kalman filter

Up to here, it has been shown that the dynamical model used for the adaptive estimation problem, is the same used when the noise had been modeled as white noise. About the covariance matrices of the noises, the description of how them change at each step of the implemented filter has been provided; more precisely according to the equations (5.2) and (5.4) for Q_{ad} and R_{ad} respectively. Now it is time to implement the adaptive filter, and to deal with the practical issues that this leads to.

5.2.1 Data pre-processing

As previously mentioned the implementation of the adaptive filter, requires the choice of the values of α, c_Q and c_R. This research has been made using the adaptive algorithm, which it has been run until the optimal values had been discovered. Is rightful to underline that, in principle is possible to find some optimal values not acceptable; this is conseguence of possible values which compromise the positive definiteness of the covariance matrices Q_{ad0} and R_{ad0}. Therefore, this is an aspect to pay attention to.

Other important aspects, are those already encountered in the last two chapters, in which it has been explained how the data of the accelerometer and those of the GPS, need to be projected on the NED frame; while about the angular velocity (which appears in the outputs equation), has been inserted in the filter, as was provided by the sensor; in fact in this case, it is the velocity state variable to be oriented for being coherent with the data set of the Px4Flow, as is stated in Chapter 3, equation (3.18).

5.2.2 Initialization

About the initialization of the filter is necessary to impose the initial conditions, from which the filter must start to run. Unlike the augmented Kalman filter, and alike the model in which the noises had been considered white noises, the initial values are those shown below:

$$x_{ad0} = 0_{6\times1}, \qquad P_{ad0} = 10^{-6} I_6. \tag{5.8}$$

In addition to the state variables and the covariance of the state initial condition, there must be imposed the initial values Q_{ad0} and R_{ad0}; according to what it has been explained in (5.6) and (5.7) the initial values are:

$$Q_{ad0} = c_Q Q \tag{5.9}$$

and

$$R_{ad0} = c_R R, \tag{5.10}$$

being the coefficients c_Q and c_R, those optimal values sought before, while Q and R the matrices of the sensor covariances modeled as white noises.

5.2.3 Multi-rate management: sequential update

Also for the adaptive Kalman filter the same approach used previously has been used: the sequential update method. It has been implemented with the purpose to handle the different rates at which the measures are provided to the filter. Practically the idea under the algorithm presented in this section, is the same shown in chapter 3 and chapter 4, since it is the sequential update method, but with the differences for including the changes due to the adaptivity of the covariance matrices at each iteration.
Below are exposed the steps performed:

- prediction step

$$\hat{x}_{adk}(-) = \Phi_k \hat{x}_{adk-1}(+) + \Gamma_k u_k, \tag{5.11}$$

$$P_{adk}(-) = \Phi_{k-1} P_{adk-1} \Phi_{k-1}^T(+) + Q_{adk-1}, \tag{5.12}$$

- GPS correction step

$$K_{adgpsk} = P_{adk}(-) H_{gpsk}^T [H_{gpsk} P_{adk}(-) H_{gpsk}^T + R_{gpsk}]^{-1}, \tag{5.13}$$

$$d_{gpsk} = y_k - \bar{H}_{gpsk} \hat{x}_{1:3k}(-), \tag{5.14}$$

$$\hat{x_k}(+) = \hat{x_k}(-) + K_{adgpsk} d_{gpsk}, \tag{5.15}$$

$$\epsilon_{gpsk} = y_k - \bar{H}_{gpsk} \hat{x}_{1:3k}(+), \tag{5.16}$$

$$R_{adgpsk} = \alpha R_{adgpsk-1} + (1-\alpha)(\epsilon_{gpsk} \epsilon_{gpsk}^T + H_k P_{adk}^- H_k^T), \tag{5.17}$$

$$P_{adk}(+) = [I - K_{adgpsk} H_{gpsk}] P_{adk}(-) [I - K_{adgpsk} H_{gpsk}]^T + \\ + K_{adgpsk} R_{adgpsk} K_{adgpsk}^T, \tag{5.18}$$

$$Q_{adk} = \alpha Q_{adk-1} + (1-\alpha)(K_{adgpsk} d_{gpsk} d_{gpsk}^T K_{adgpsk}^T), \tag{5.19}$$

where H_{gpsk} is the sub-matrix of $H_k(i,j)$, with $i = 1,\cdots,3$ and $j = 1,\cdots,6$; $\bar{H}_{gpsk}$ is the sub-matrix of $H_k(i,j)$ with $i = 1,\cdots,3$ and $j = 1,\cdots,3$; as it is possible to see from the just shown equations involved in the algorithm, respect to the algorithm shown in chapter 3, this one presents the use of both the innovation in (5.14), and residual in (5.16).

What it has just been shown are the iterations used to estimate the state variables, by using an adaptive correction of the covariance matrices. More in particular, the prediction of the state variables ($\hat{x}_{adk}(-)$) and of the state covariance ($P_{adk}(-)$) is made, then the correction step is performed, to give the estimates of the same ($\hat{x}_{adk}(+)$ and $P_{adk}(+)$). During the estimation step, the innovation and the residual have been computed, to be used to solve the descriptive equations of Q_{adk} and R_{adk}.

The same for the measures provided by the f sensor:

- f correction step

$$K_{adofk} = P_{adk}(-)H_{ofk}^T[H_{ofk}P_{adk}(-)H_{ofk}^T + R_{ofk}]^{-1}, \tag{5.20}$$

$$\bar{y}_k = y_k - Du_w, \tag{5.21}$$

$$d_{ofk} = \bar{y}_k - \bar{H}_{ofk}\hat{x}_{4:5k}(-), \tag{5.22}$$

$$\hat{x_k}(+) = \hat{x_k}(-) + K_{adofk}d_{ofk}, \tag{5.23}$$

$$\epsilon_{ofk} = \bar{y}_k - \bar{H}_{ofk}\hat{x}_{4:5k}(+), \tag{5.24}$$

$$R_{adofk} = \alpha R_{adofk-1} + (1-\alpha)(\epsilon_{ofk}\epsilon_{ofk}^T + H_k P_{adk}^- H_k^T), \tag{5.25}$$

$$\begin{aligned} P_{adk}(+) = [I - K_{adofk}H_{ofk}]P_{adk}(-)[I - K_{adofk}H_{ofk}]^T + \\ + K_{adofk}R_{adofk}K_{adofk}^T, \end{aligned} \tag{5.26}$$

$$Q_{adk} = \alpha Q_{adk-1} + (1-\alpha)(K_{adofk}d_{ofk}d_{ofk}^T K_{adofk}^T), \tag{5.27}$$

where H_{ofk} is the sub-matrix of $H_k(i,j)$, with i = 4,5 and j = 1,$\cdots$,6; $\bar{H}_{ofk}$ is the sub-matrix of $H_k(i,j)$ with i = 4,5 and j = 4,$\cdots$,6.
It is rightful to keep in mind, that an important aspect is that about the positivity of the matrix P_{adk}: in fact for each iteration this test must be assured, and the same for the matrix of the Kalman gain.

5.3 Conclusion

In this chapter it has been exposed that one way to model the covariance of the noises, is through an adaptive approach. In particular the form of the equations, used at each iteration to update Q and R has been presented. Since these latter matrices are subject to iterations, they need initial guesses, which have been detected by minimizing an index dependent on the variables α, c_Q and c_R; those optimal values have been used for the tuning of the adaptive Kalman filter. The last part of the chapter, deals about the algorithm that has been implemented, keeping into account the multi-rate problem. Moreover it has been shown that respect to the Kalman filter with the noises described as white noise and to the augmented Kalman filter, the adaptive Kalman filter presents a slightly different estimation process, in which the innovation as well as the residual become important components of the filter.

Chapter 6

Experimental set up

This chapter is structured in two principal parts: in the first half a description of the principal properties of the adopted UAV and of the on board f sensor are presented, while in the second one the dynamical features of all the exploited sensors are shown.
In the last part of the chapter, the pre-test activities are briefly described.

6.1 UAV and Px4Flow set-up

For the experimental activity of this work, the machine on which the Kalman filter has been applied and tested, is the hexa-copter UAV. It is a drone designed and which become operative during previous works carried out in the ASCL (Aerospace System & Control Laboratory) of Politecnico di Milano. The drone has the following properties:

- configuration → six propellers;
- TOW → 900 g;
- flight duration → 20 min;
- payload → 350 g.

The f sensor, main character of this thesis, 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.

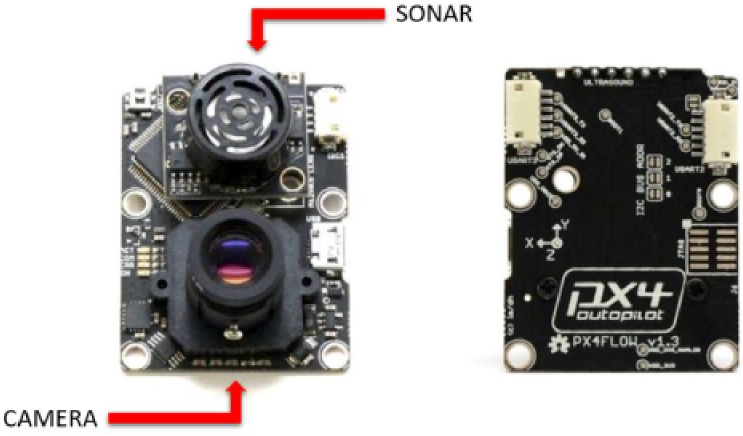


Figure 6.1: Px4Flow sensor

6.1.1 Optical flow sensor: Px4Flow

The Px4Flow sensor, introduced in Figure 6.1, is an f smart camera born for the computation of the f; this is made from the observations 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 (*i.e.* not when the sensor is operating on the UAV). Unlike many mouse sensors, it also works indoors and in low outdoor light conditions without the need for an illumination LED. Moreover it is also very sensitive to high brightness conditions due to its very high f calculation frequency, 400 *Hz*.

From the operative point of view, the Px4Flow sensor is not just a camera, but it is thought to provide comfortably the f data. In fact, on board we can appreciate a set of gyroscopes for the angular compensation and a sonar for the ground altitude scaling.

Technical properties:

- 168 *MHz* Cortex M4F CPU (128+64 *KB* RAM);
- 16 mm M12 lens;
- size 45.5 *mm* x 35 *mm*;
- power consumption 115 *mA* / 5 *V*;
- on board 16 *bit* gyroscope up to 2000° / *s*;
- on board sonar;
- USB bootloader and power option.

Dimensions and connectors scheme:

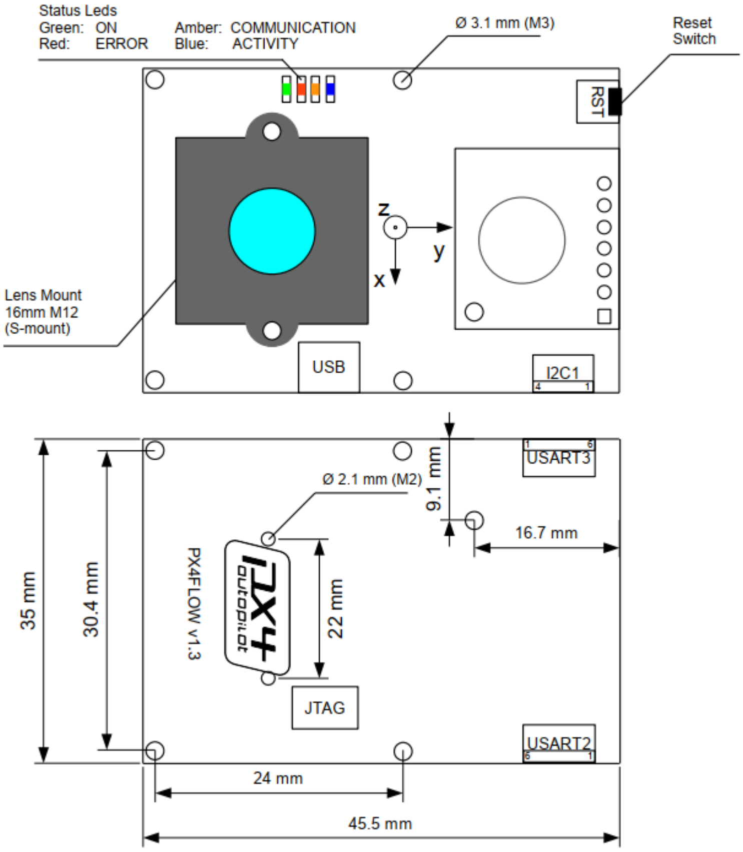


Figure 6.2: Dimensions and connectors

6.1.2 On board integration

The choice of using the hexa-copter is taken for reason of comfortable allocation possibility and for its compatibility with the requirements of payload and power indicated by the technical specifications of the Px4Flow sensor, visited in 6.1.1. The Px4Flow needs to be oriented looking downward, then on the UAV belly a new rigid platform has been allocated to host the new sensor. The supplementary platform gives the possibility to set new electronic component without taking space to the pre-existing sensors and modules. For more insights about the Design and the realization of the platform, as well as its mounting activity, see the first part of this work [1].

SENSOR	FLIGHT
Accelerometer	Hover
GPS	UAV not in flight, placed on a platform
Px4Flow	Hover
MoCap	UAV controlled in position

Table 6.1: Flights for noise characterization

6.2 Sensor characterization

In this second part, the purpose is to characterize the dynamics of the noises provided by the sensors. In particular, starting from flights made ad-hoc for not including flight dynamics, the noises are identified. In table 6.1, the flights are reported in dependence on the sensor considered.

For the characterization of the accelerometer and the f sensor noises, the UAV has been set up to fly in hover. This flight condition is natural for assessing the accelerometer noises, since it is known that in hover, the sensor should measure null acceleration; then fluctuations about that value will correspond to the noise. The same flight condition has been adopted for the Px4Flow, since in this way there is not the necessity of compensating the angular velocities (from which the f measurement depends). The resulting noises are reported in the following Figures (6.3 and 6.4):

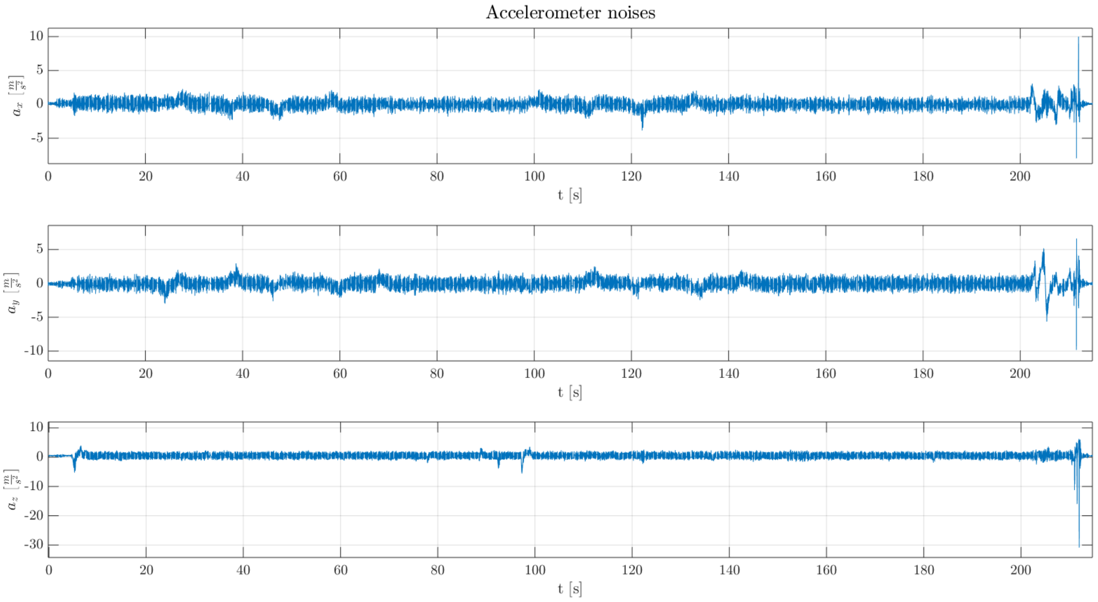


Figure 6.3: Accelerometer noise

For the GPS noises the UAV has been placed on a fixed platform to avoid multipath errors; subsequently the data have been logged with the UAV not flying,

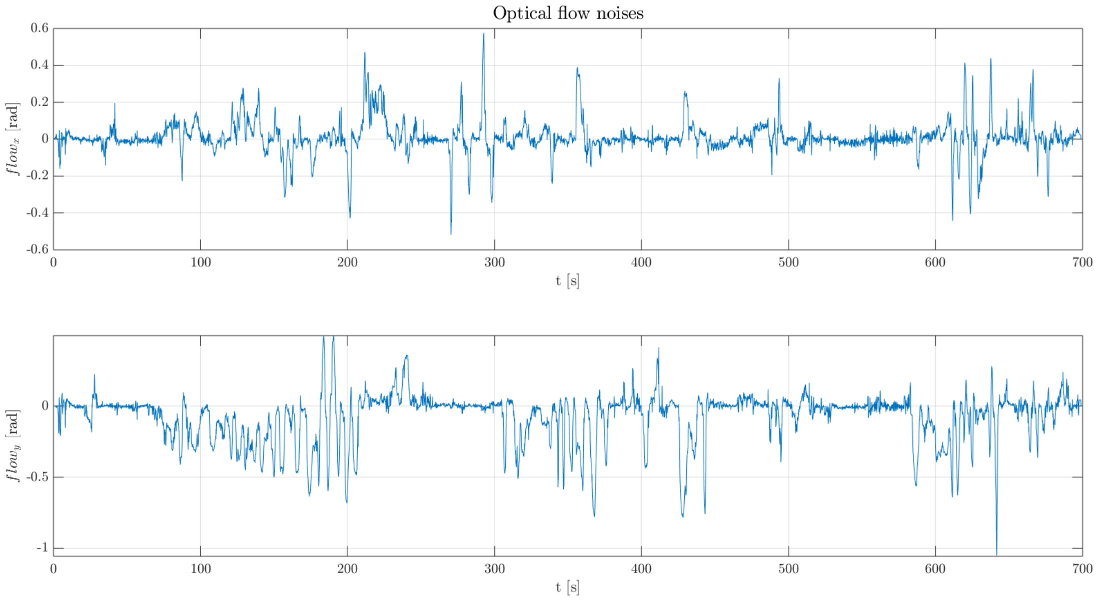


Figure 6.4: Optical flow sensor noise

but with armed propellers in order to have dirturbance conditions similar to the flight mode. The resulting time history of the noise, is marked out in Figure 6.5.

In case of indoors flights the MoCap is used rather than the GPS, for the position measurements. For the evaluation of its noise, the UAV has been controlled in position by means of the MoCap. Then also the noises of the MoCap sensor are plotted (Figure 6.6).

Now that the noises have been discovered, the step ahead is to describe them appropriately to include them in the Kalman filter. According to the Kalman filter version described in Chapter 3 the variance is sufficient; for that in Chapter 4 the shape filters are necessary while for the adaptive version, α, c_Q and c_R must be computed.

Shape filters realization

The path followed for the realization of the shape filters is the one described in Chapter 4: practically, having the time histories of the noises, their PSD mean square root have been computed, and subsequently the correspondent frequency response. A fitting of the frequency response has been sought, and among all the possible realizations the one that has been chosen is that which presents the lower order to fit adequately the frequency response.

In the following plots, the trend of the frequency response correspondent to the noises are shown (computed as explaiend above), and the realizations of the 1^{st} and the 2^{nd} order of the shape filters.

For the accelerometers the observed frequency response of the shape filters, is a

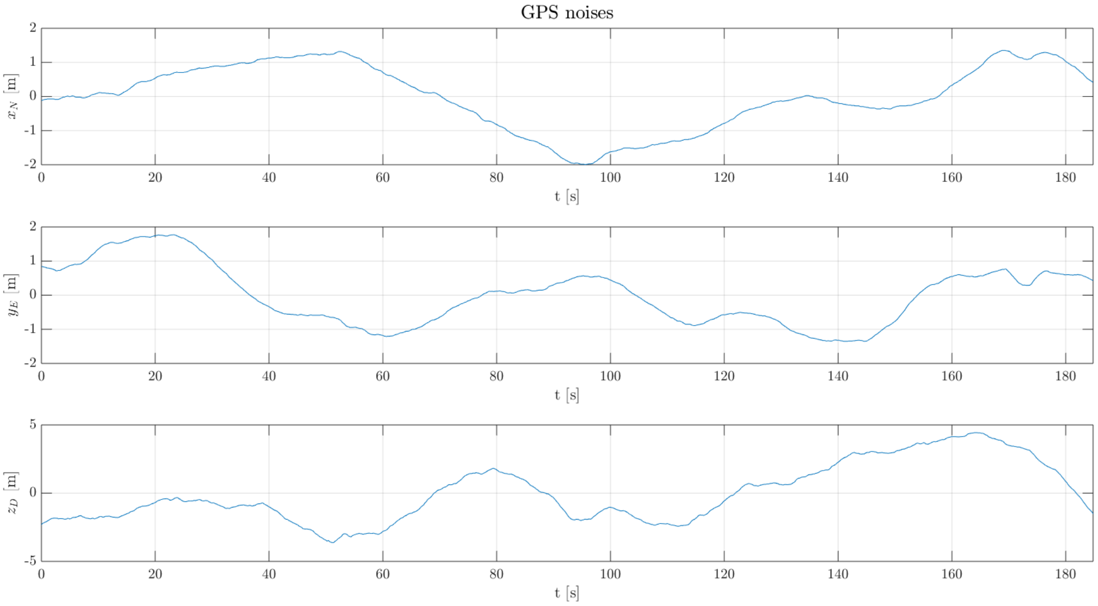


Figure 6.5: GPS noise

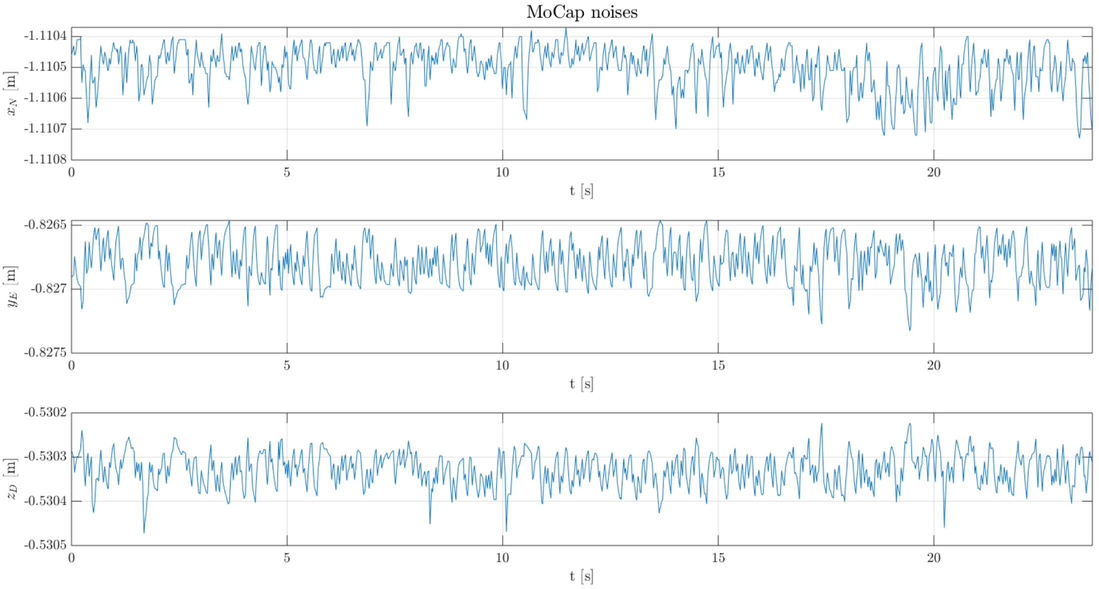


Figure 6.6: MoCap noise

low-pass realization, as shown in Figure 6.7.

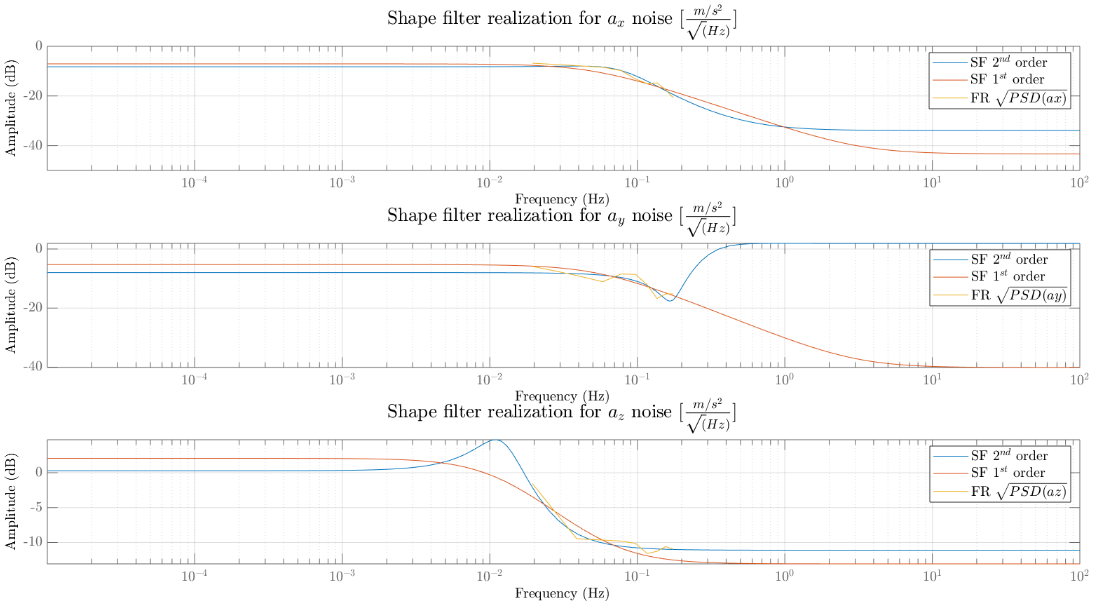


Figure 6.7: Shape filter realization of the acceleration noise

The second order shape filter realization has been chosen rather than the first order for the description of the noise dynamics, due to its better fitting. This, even if along the y-axis it is not completely satisfying; in fact the first acceptable fitting, for the y-axis, is of the 5^{th} order. On the other hand the orders 3 and 4 do not significantly improve the issue. Since a 5^{th} order realization implies too much computational load, in the implementation of the augmented Kalman filter, the 2^{nd} order realization has been adopted.

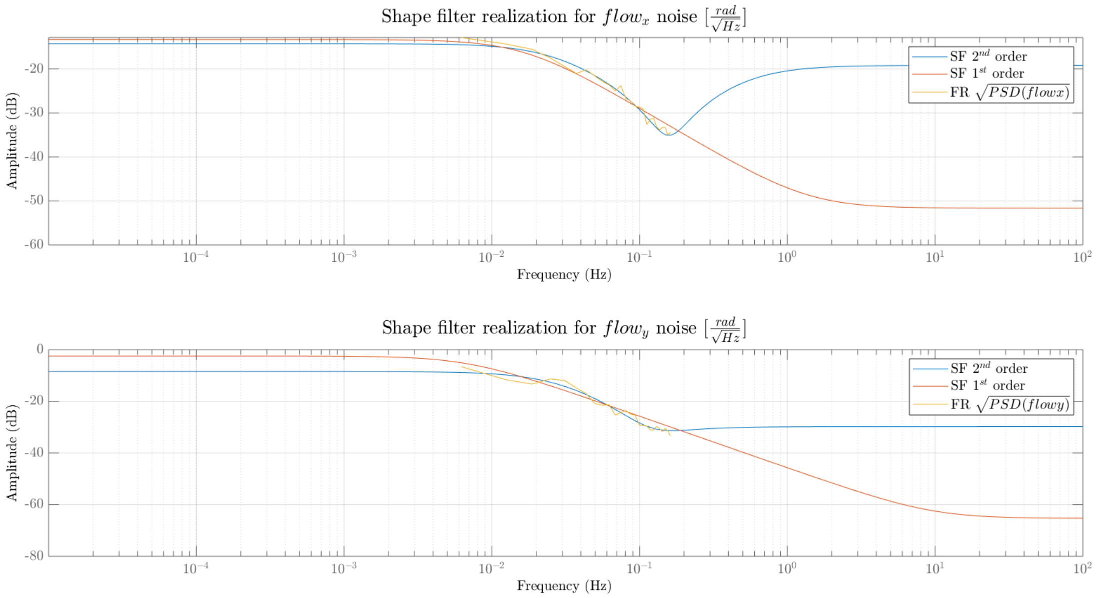
Actually, the 2^{nd} order realization has been considered a sufficient description for all the sensors.

The f sensor presents a trend which is, slightly different than the one of the accelerometer, but still a low-pass trend, as depicted in Figure 6.8.

The shape filters correspondent to the noises provided by the GPS, have the frequency response plotted in Figure 6.9. For channel z, a frequency response realization with an upper bound is shown. This make worse the fitting, but returns some improvements in the estimation of the position along the z-axis, as will be shown in Chapter 7.

Finally the results for the MoCap sensor, are marked out in Figure 6.10.

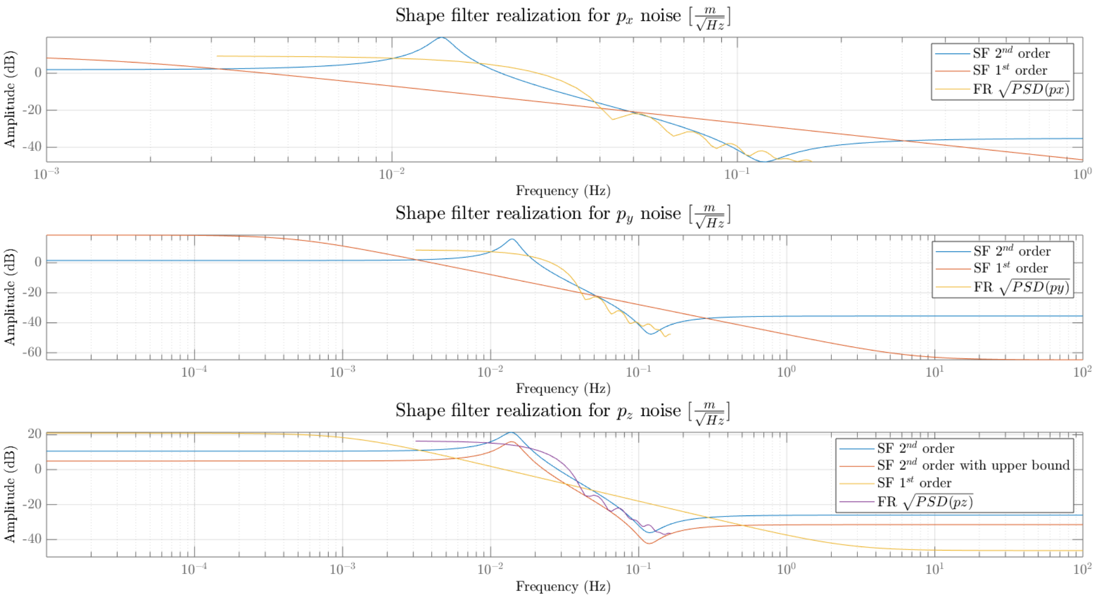
As it is possible to see from the Figures 6.9 and 6.10, the shape filter realizations of the MoCap sensor present smaller amplitude respect to those of the GPS. This is due to the fact that the MoCap appreciates variations in the order of millimeters.



Shape filter realization for $flow_x$ noise $[\frac{rad}{\sqrt{Hz}}]$

Shape filter realization for $flow_y$ noise $[\frac{rad}{\sqrt{Hz}}]$

Figure 6.8: Shape filter realization of the optical flow noise



Shape filter realization for p_x noise $[\frac{m}{\sqrt{Hz}}]$

Shape filter realization for p_y noise $[\frac{m}{\sqrt{Hz}}]$

Shape filter realization for p_z noise $[\frac{m}{\sqrt{Hz}}]$

Figure 6.9: Shape filter realization of the GPS noise

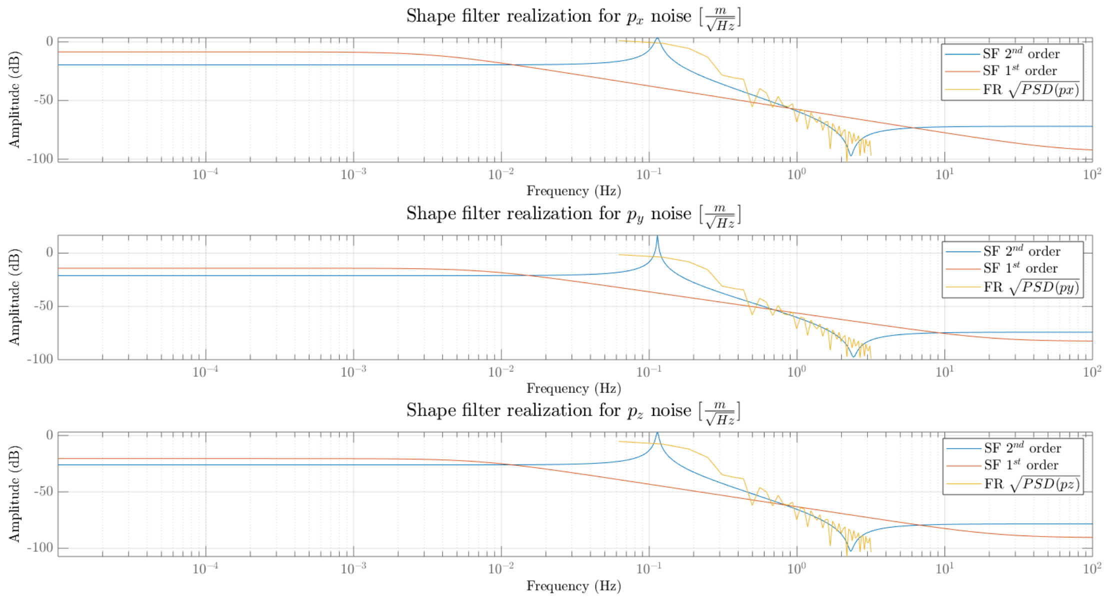


Figure 6.10: Shape filter realization of the MoCap noise

Noise variances

The estimate of the variances, when the noises are considered as white, is made using the PSD mean square root of the noises. Then the obtained quantity (which for the sake of clarity, is the same used to realize the shape filters) is treated following these passages:

- the mean value μ has been computed;
- from μ the standard deviation σ has been obtained:

$$\sigma = \frac{\mu}{\sqrt{2}}; \tag{6.1}$$

- from σ the variance has been obtained: Var $= \sigma^2$.

In table 6.2 the resulting variances of the noises are reported.

Values of α, c_Q and c_R

As said previously, α, c_Q and c_R are the results of an optimization process. Starting from α, the Kalman algorithm has been run several times for each value of α in vector $\alpha_{vect} = [0 : 0.0001 : 1]$. The value of α, for which the MSE of the states is minimized, is the optimal one. Then the research of c_Q and c_R is carried out, fixing α to the optimal value: in this case the filter is let run for the following values of c_Q and c_R: $c_{Qvect} = [10^{-4}\ 10^{-2}\ 1\ 10^{2}\ 10^{4}]$ and $c_{Rvect} = [10^{-4}\ 10^{-2}\ 1\ 10^{2}\ 10^{4}]$.

SENSOR	μ	**Var**
ACCELEROMETER	$[(m/s^2)/\sqrt{Hz}]$	$[((m/s^2)/\sqrt{Hz})^2]$
a_x	0.2708	0.0367
a_y	0.3024	0.0457
a_z	0.4285	0.0918
OF	$[rad/\sqrt{Hz}]$	$[(rad/\sqrt{Hz})^2]$
$flow_x$	0.010	$0.0025 \cdot 10^{-5}$
$flow_y$	0.1166	$0.0068 \cdot 10^{-5}$
GPS	$[m/\sqrt{Hz}]$	$[(m/\sqrt{Hz})^2]$
p_x	0.4266	0.0910
p_y	0.3932	0.0773
p_z	0.9817	0.4819
MoCap	$[m/\sqrt{Hz}]$	$[(m/\sqrt{Hz})^2]$
p_x	0.0724	0.0026
p_y	0.0539	0.0015
p_z	0.0346	0.0006

Table 6.2: Noise variances

FLIGHT	α	c_Q	C_R
OUTDOOR	0.9999	$1 \cdot 10^2$	$1 \cdot 10^{-4}$
SLOW INDOOR	0.9999	$1 \cdot 10^{-2}$	$1 \cdot 10^{-4}$
FAST INDOOR	0.9990	$1 \cdot 10^4$	1

Table 6.3: α, c_Q and c_R

The couple of c_Q an c_R which minimizes the MSE, is that sought.
In table 6.3 the characterisation of the aforesaid coefficients, for each flight, is collected.

6.3 Pre-test activities

There are some procedures to follow before doing a flight test, in order to obtain a valid data set. The program QGroundControl gives an appropriate ground station for the aforesaid activities.
First of all, the PixHawk (Flight Control Unit) and the Px4Flow firmwares have been uploaded; the firmware could be the delfault one or a tuned one depending on what is requested by the machine.
The second step is the calibration of all the on board sensors. Actually, in this description, more focus will be spent on the calibration of the PX4Flow. Its camera needs to be focused on the image, taking into account the range of altitudes in which the flight will be performed. In fact its ability of catching good images,

depends a lot on this, and a poor vision quality could completely affect the f data. A parameter named "Quality" is available from the messages of the Px4Flow, and logging it, a qualitative measure of how well the camera is catching images, helpuful for the computation of the optic flow, is provided. The sensor is set to work only when this parameter is bigger than a chosen threshold. The quality of the f is influnced by some elements: brightness, surface roughness and colour, and lens calibration. The latter is where it is possible to 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.

When the aforesaid procedure is executed, the UAV needs to catch the signal from the GPS network (if an outdoor flight is required) or from the MoCap (if an indoor flight must be performed). Then the UAV equipped with the Px4Flow is ready to fly with the certainty that the correspondent measurement data sets are coherents.

Chapter 7

Filtering results

In the previous chapters, three approaches have been exhibited for the filtering process of the UAV position and velocity: in Chapter 3 the sensor noises have been modeled as white noises; in Chapter 4 the shape filters have been used to describe the dynamics of the noises, while in Chapter 5 an adaptive algorithm, in which the variance matrices of the sensors were adapted during the filtering process, has been adopted. Applying the different versions of the Kalman filter on real UAV flights, different performances have been observed: this chapter wants to show the estimate results, and subsequently, making a comparison among them it tries to draw conclusions.

This chapter is divided in three parts: the first one will compare the estimate results of the augmented Kalman filter ('AugKF') respect to the standard Kalman filter ('KF'), while the second part will focus on the adaptive Kalman filter ('AdKF') in comparison to the KF. In the end a comparison between the AugKF and the AdKF will be performed.

7.1 Estimate results: augmented Kalman filter

In this section the results for the outdoor and the indoor flights, considering the augmented Kalman filter, will be presented.

All the estimates are based on position and flow measurements as well as the acceleration one (seen by the filter as input, as described by the mathematical model).

7.1.1 Outdoor flight

Since the UAV is in outdoor flight, the position meaurements are provided by the GPS.

In the following plots the estimates of the considered state variable, will be reprensented with respect the measurement provided by the GPS, taken as reference for evaluating if the time history of the estimates has sense. This does not state that the estimate is an absolutely correct one, since the GPS measurements contain intrinsic errors; nevertheless if the estimate is similar to the measurement, then the filter is working coherentely.

The first plot encountered is in Figure 7.1, and it marks out the trend of the position along the x-axis in NED reference. The first noticeable thing, is that the behaviour of both the estimates, the one provided by the KF and that by the AugKF, have sense; in fact either the estimates try to follow the behaviour of the position measurement provided by the GPS. In particular using the augmented Kalman algorithm is evident an improvement respect to the KF, since the estimate is very similar to the measure. For a numerical comparison of how much is the discrepancy between each estimate and the measure provided by the GPS, see the histogram in 7.4.

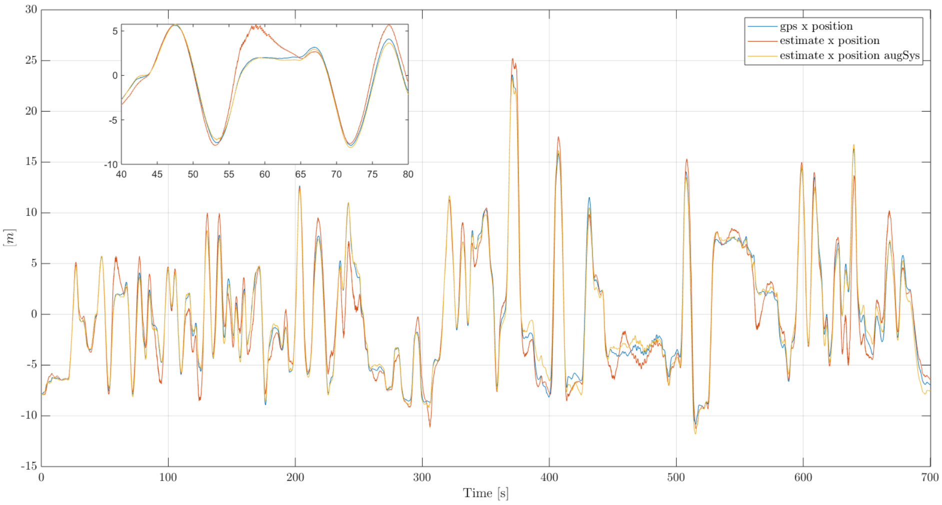


Figure 7.1: Position estimation along x-axis

In Figure 7.2 and 7.3 the estimates of the position along the y-axis and the z-axis respectively are shown . The position estimation along y is fine, but it is along the z-axis that the estimate provided by the augmented filter presents an important improvement; in fact the AugKF respect to the KF returns an estimate which is more satisfying along the z-axis, which is the critical channel. Also for the axes y and z, the discrepancy value between the estimates and the GPS measurement, are provided (histogram in Figure 7.4).

Using the GPS and the f sensors, the estimation of the velocity along x, y and z, in NED reference, is observable in the following Figures 7.5, 7.6 and 7.7 respectively. The velocities along x and y, as expected present a good estimation, but it is along the z-axis that the estimate presents an important improvement. In Figure 7.7 is possible to see that using the augmented algorithm, an improvement of the velocity estimation is evident. The discrepancy of each estimate respect to the considered measurement by the GPS, for each axis, is reprenseted in the

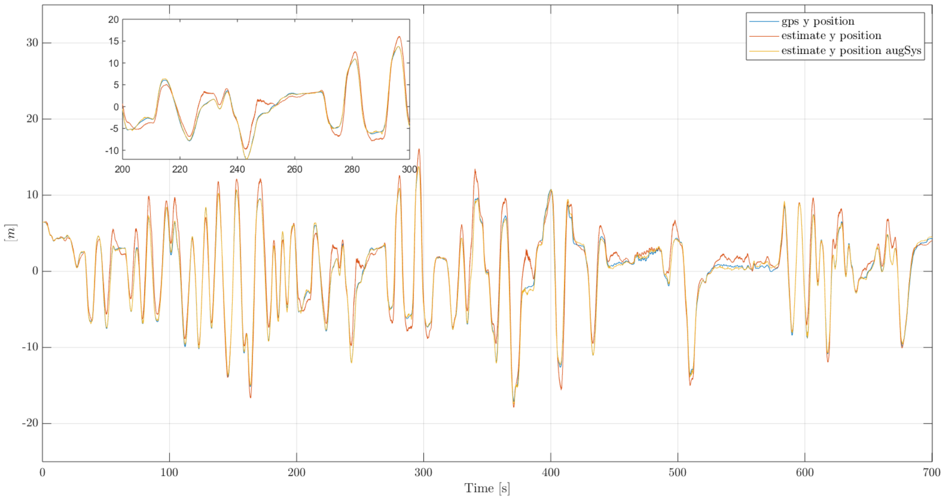


Figure 7.2: Position estimation along y-axis

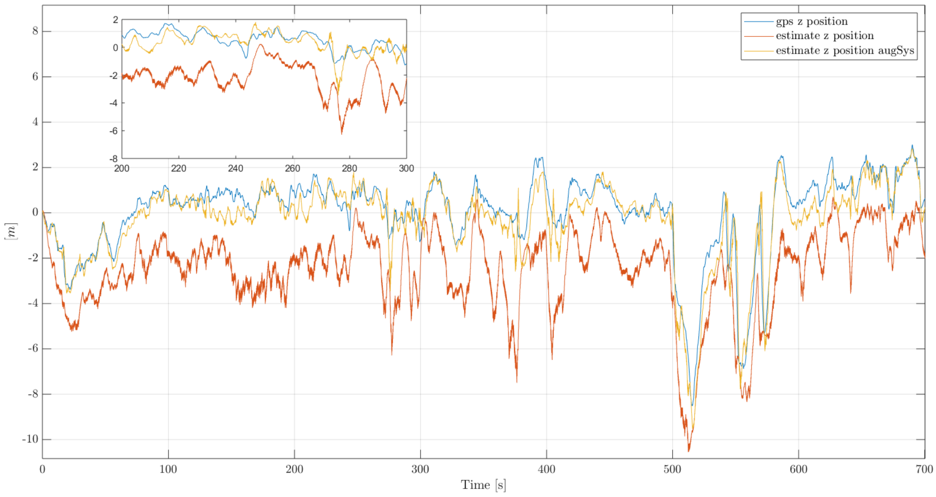


Figure 7.3: Position estimation along z-axis

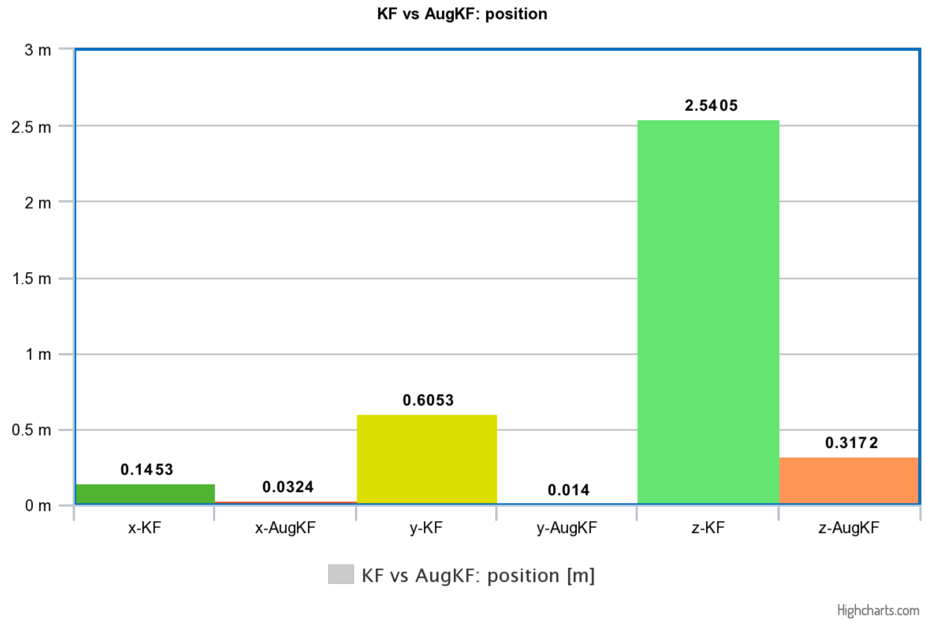


Figure 7.4: Comparison of discrepancies ”estimation-measurement” for position

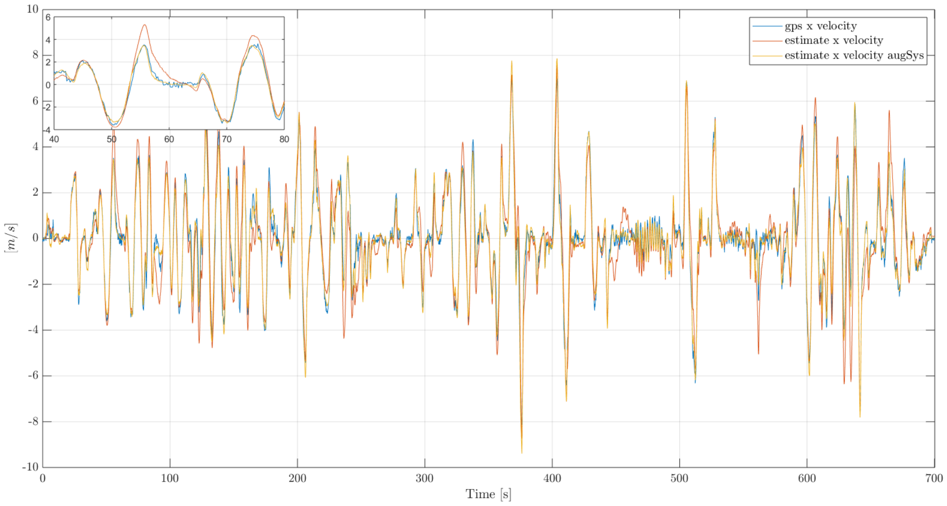


Figure 7.5: Velocity estimation along x-axis

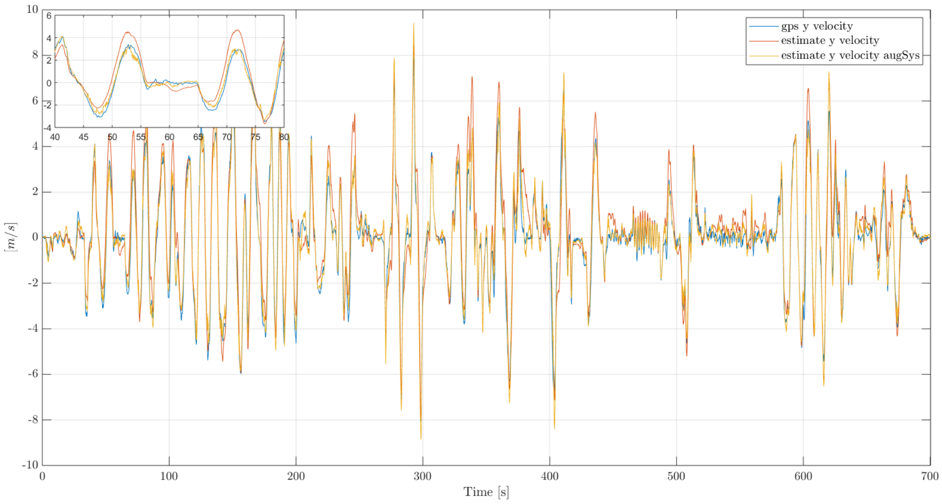


Figure 7.6: Velocity estimation along y-axis

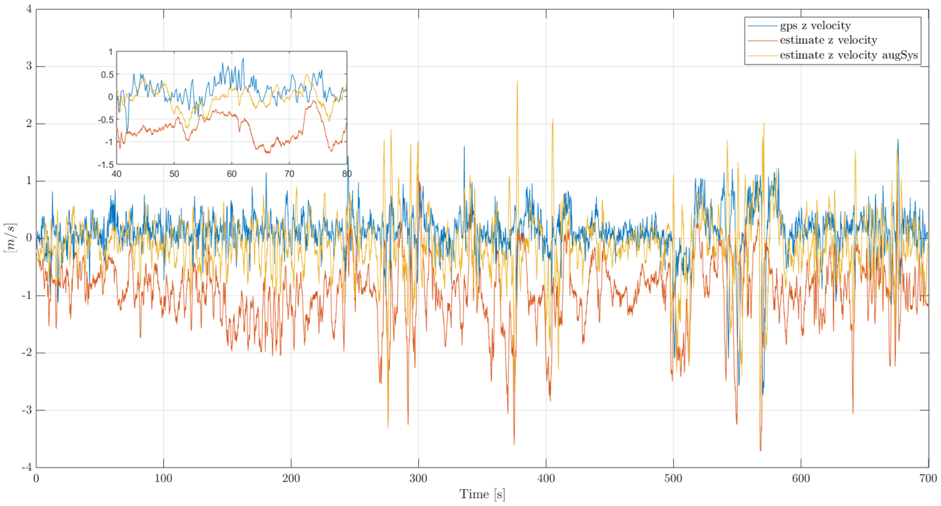


Figure 7.7: Velocity estimation along z-axis

histogram 7.8. Since each version of the Kalman filter (KF and AugKF in this case) presents a different estimation, the difference between the estimations can be of interest (see 7.9), to better handle the results of the different approaches. Up to now the goodness of the estimates has been judged only considering their behavior respect to the measures by the GPS; but a more techincal explanation is necessary. For this purpose the innovation is analyzed, and its behaviour must consist of:

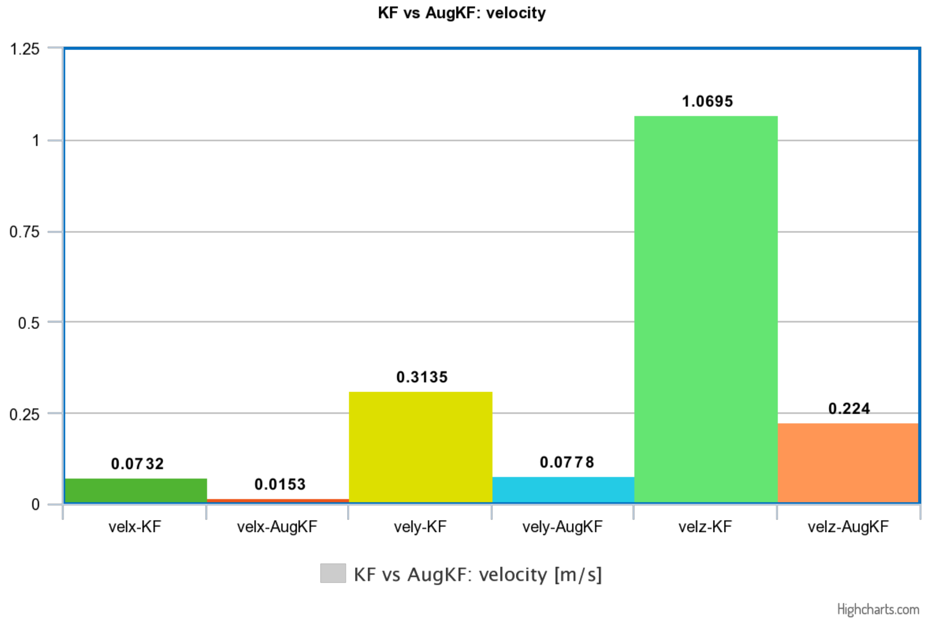


Figure 7.8: Comparison of discrepancies ”estimation-measurement” for velocity

- a white noise, zero mean value process, which suggests the filter is working right ([9]);
- a trend that stay bounded in the plus-minus three standard deviation ($\pm$ 3std) band in order to ensure the normal distribution of the innovation. The standard deviation has been calculated from the values of the P state covariance matrix.

In Figures 7.10 and 7.11 the innovation $PSDs$ correspondent to the position and the f are provided. In 7.10 as well as 7.11 is noticeable that both the innovation $PSDs$ are smaller than the PSD of the measurement; this means that the filters work well. In 7.10 is observable that the innovation PSD resulting from the AugKF is smaller than that of the KF; *i.e.* the position estimate of the augmented filter presents less variance; conversely is observable for the f measure. As said previously, this estimation problem adopts the measurements of more sensors, namely a multi-rate behaviour is expected. In fact by observing the innovation

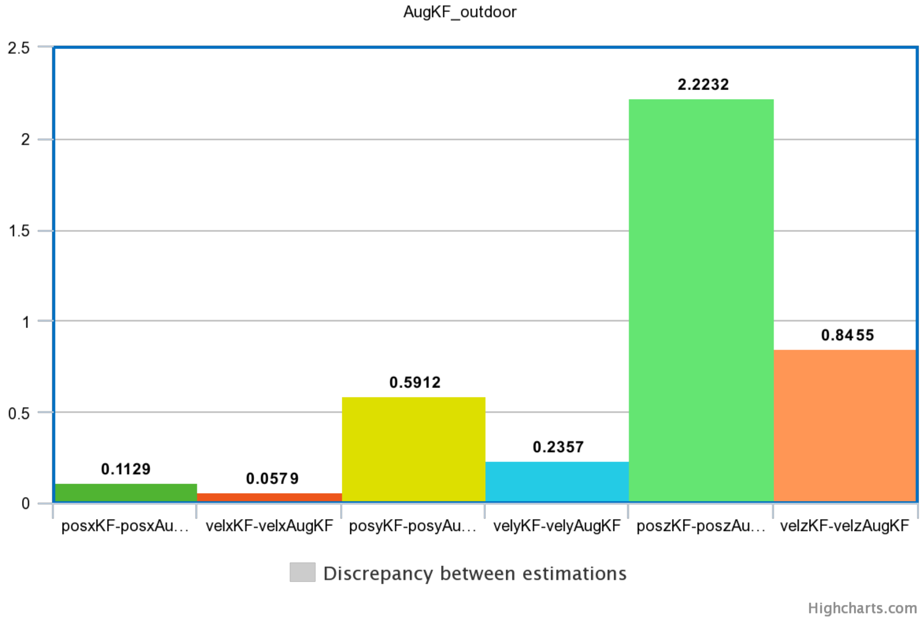


Figure 7.9: Discrepancy between estimations

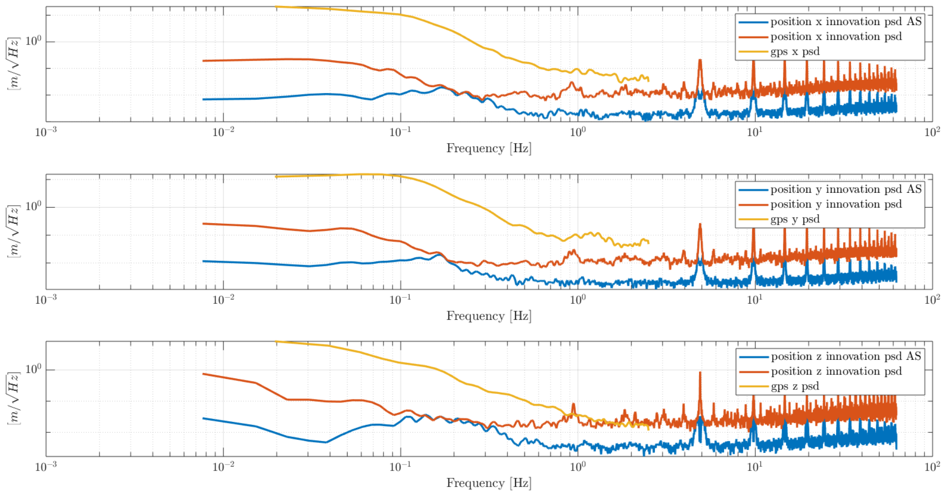


Figure 7.10: Innovation $PSDs$ for position

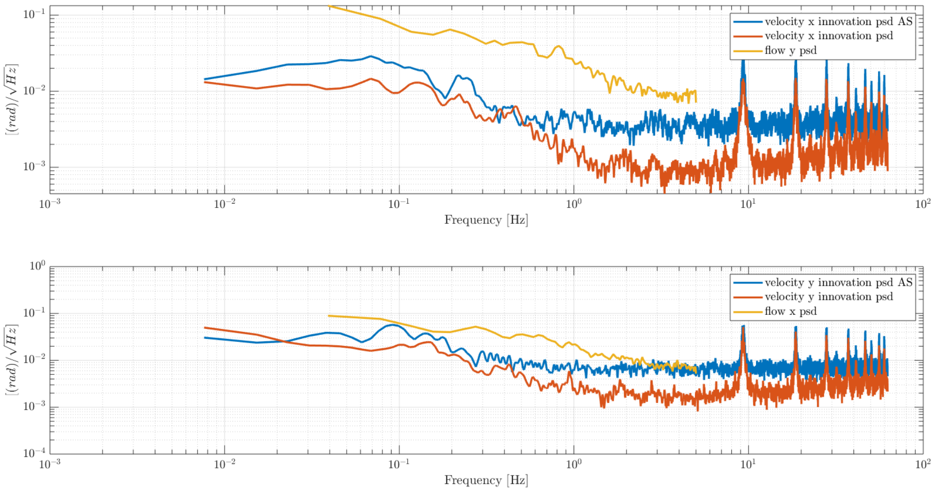


Figure 7.11: Innovation $PSDs$ for f

$PSDs$ at high frequency, some periodical peaks are noticeable: this is consequence of the GPS sampling frequency, which is equal to 5 Hz, but also other small peaks can be observed and they are consequences of the combination of the effects given by the sampling time of both the sensors (GPS and f sensor). Similarly happens for the f innovation $PSDs$, but this time the periodical peaks are due to the sampling frequency of 10 Hz.
Now is necessary to verify that the time history of the innovations of the AugKF, stay bounded in the plus-minus three standard deviation. The plots in Figure 7.12 and 7.13, show that the innovations stay within the aforesaid band, for most of the flight time, a part for some instants. Moreover in Figure 7.13 is noticeable that the three standard deviation band corresponding to the AugKF, is larger than that of the KF; this is consequence of the fact that the innovation PSD of the f resulting from the AugKF, is larger than that resulting from the KF as it is possible to see in Figure 7.11. Another parameter of interest about the filter response, is the Kalman gain. On the basis of its value the algorithm gives more weight to the most recent measurements or to the model prediction; then its time history can be of interest. In Figure 7.14 and 7.15 the Kalman gains based on position and f measurements are plotted. For the just shown results, the quality of the f measurement was the one plotted in Figure 7.16.

Keeping into account the results just shown, is possible to conlude that the AugKF algorithm provides good estimations; moreover if it is compared to the KF, it guarantees better estimations, with lower uncertainties for the position, while accpetable uncertainties for the velocity.

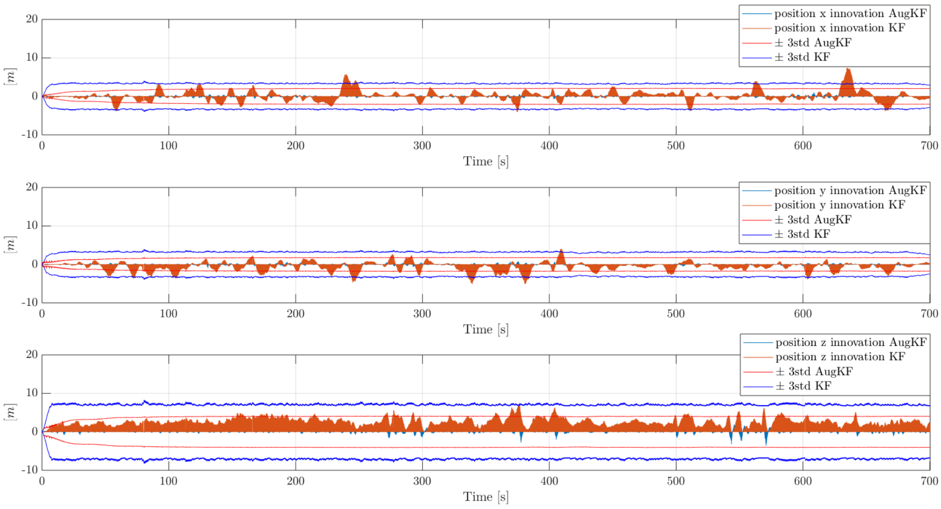


Figure 7.12: Position innovations

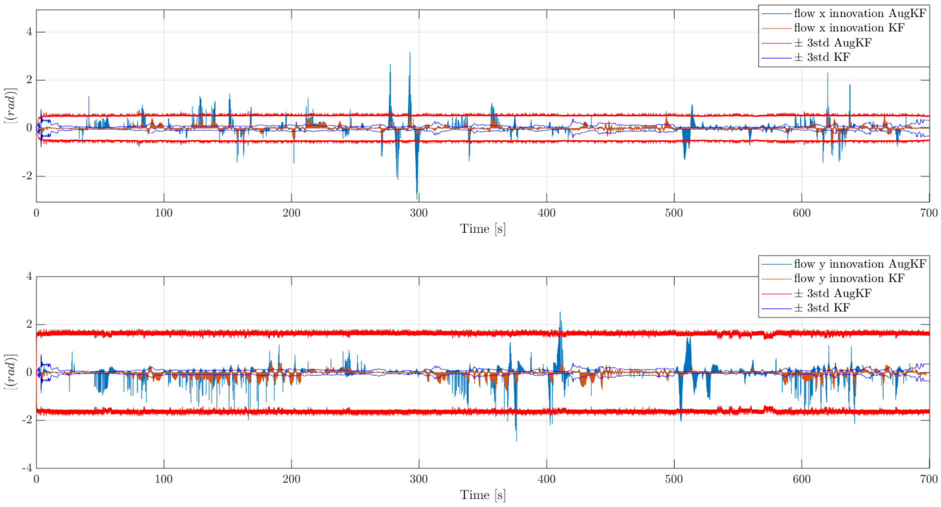


Figure 7.13: Optical flow innovations

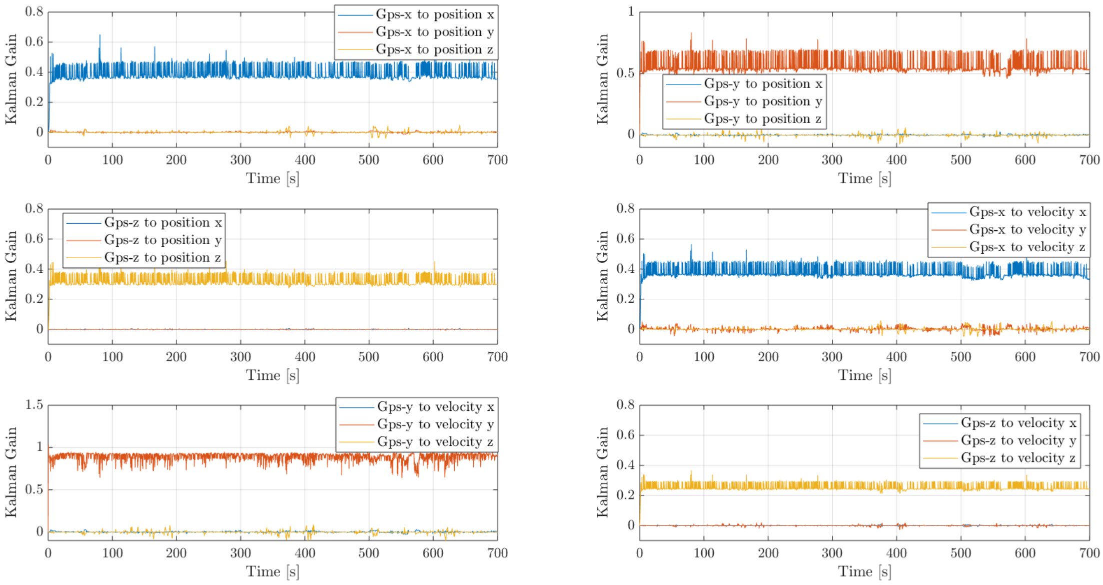


Figure 7.14: Kalman gain due to the position measurements

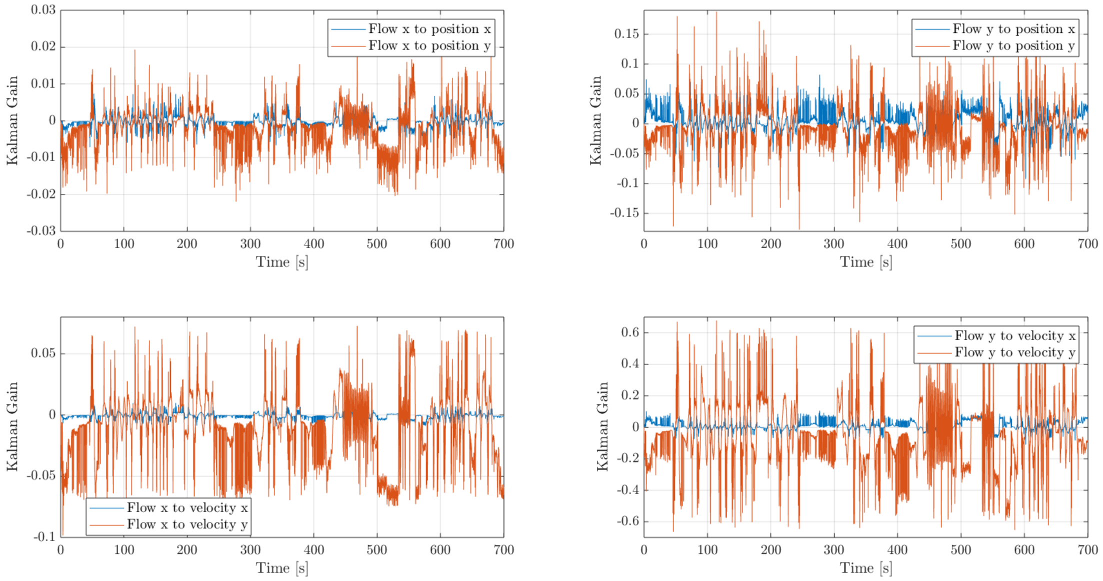


Figure 7.15: Kalman gain due to the optical flow measurements

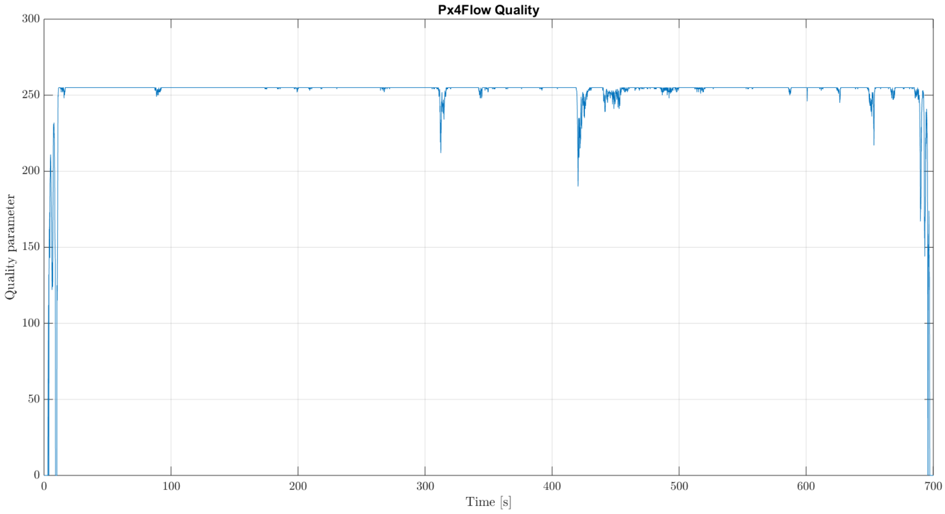


Figure 7.16: Quality of the optical flow

7.1.2 Indoor flight

Since the most critical qualities of the f measurements, have been obtained during indoor flights, two different flights have been thought: one with slow flight dynamics and the other one with fast flight dynamics. This choice has been made since the performances of the filtering process are dependent on the dynamics of the state to be estimated. The two flights are thought to let the UAV to perform the same trajectory, but at different velocity.
In the previous section an outdoor flight has been performed to test the implemented filter; in this part the purpose is to do the same thing but with an indoor flight. This difference in enviroment brings as consequence different quality of the data collected by the f sensor, implying different results in the estimation process. It is helpful to remember that for indoor flights the position measurement is provided by the MoCap sensor.

Slow flight

The estimations of p_x, p_y and p_z are shown in Figure 7.17, 7.18 and 7.19 respectively. Observing the plots of the position estimations, oscillations with small amplitudes are evident, during all the time history. This phenomenon is probably due to the fact that the MoCap sampling frequency is 100 Hz, while the Kalman filter runs at 125 Hz. Actually this behaviour is more evident for the KF. As it is possible to see from the aforesaid figures the position state estimations chase the measures from the MoCap. At first glance the filter seems to work coherently with the actual motion of the UAV. This means that the tuning of the augmented filter for the indoor test, has been made adequately. A comparison of the discrepancy between each estimate and the measurement, for each axis, is shown in the

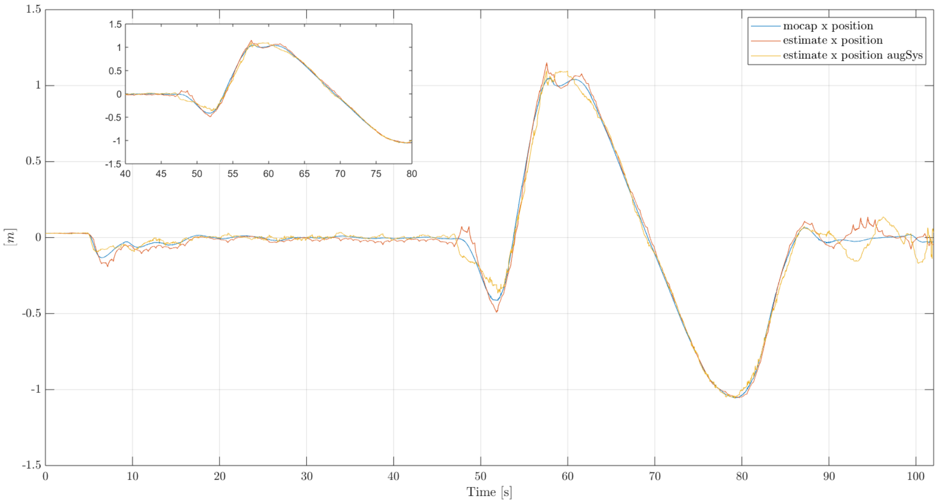


Figure 7.17: Position estimation along x-axis

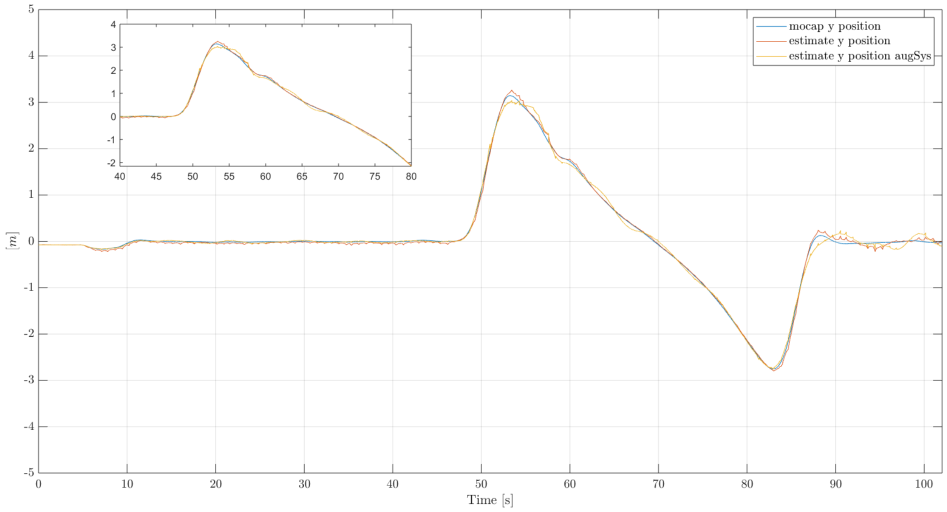


Figure 7.18: Position estimation along y-axis

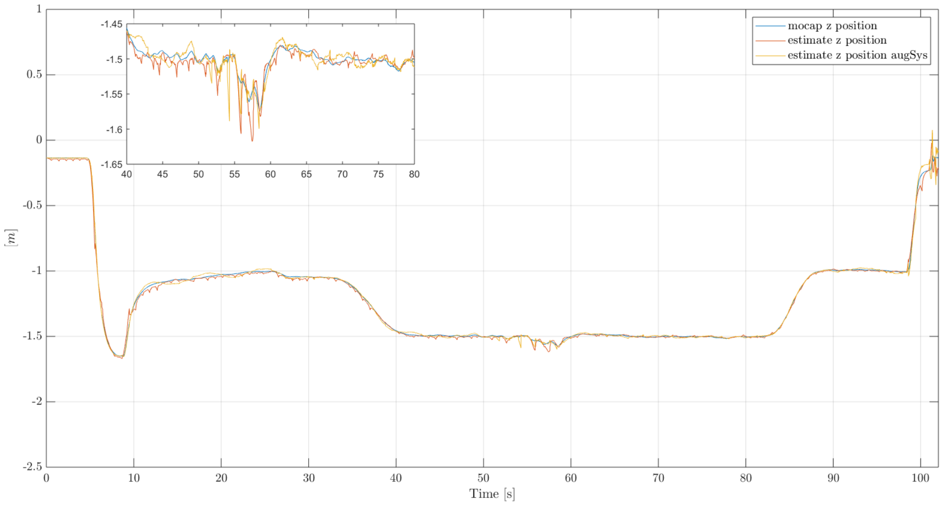


Figure 7.19: Position estimation along z-axis

histogram in Figure 7.20. Practically the augmented filter, along x and y does not improve significantly the position estimate, but a meaningful improvement is verified along the z-axis.

In Figure 7.21 and, 7.22 there are the plots of the velocity estimates v_x and v_y respectively. Even for the velocity the filter seems is working well. The time histories of the velocity estimates with respect to the velocity measurement gives us the idea that the filter is working adequately. Note how the phenomenon of the oscillations along the estimation of the velocities is evident. As shown in Figure 7.23 the estimate along the z-axis is a satisfying one.

In order to get an idea of the differences resulting by using the AugKF rather than the KF, the data in the histogram in Figure 7.24 are exhibited. As it is possible to observe the discrepancy between estimate and measurement is decreased about one order along all the axes by using the AugKF.

As before the difference between the estimations is shown in Figure 7.25.

As seen for the outdoor test, important indicators about the performances and the goodness of the work made by the filter, are the PSD and the time history of the innovation. In the following plots the innovation $PSDs$ compared to the PSD of the measures, for position (Figure 7.26) and f (Figure 7.27) are shown. For the indoor flight more than for the outdoor one, observing the innovation $PSDs$ in 7.26 is noticeable a trend which is practically constant for all the frequencies. This means that when the flight is in indoor enviroment, the innovation PSD resulting from the implemented filters (both the KF and AugKF) behave more similarly to a white noise respect to the same algorithm when tested in outdoor (this is not a generalization but an observation resulting from the analysis of the two flights examined). Another considerable aspect is about the amplitude of the innovation PSD of the AugKF respect to that of the KF; a decrease in its value implies a

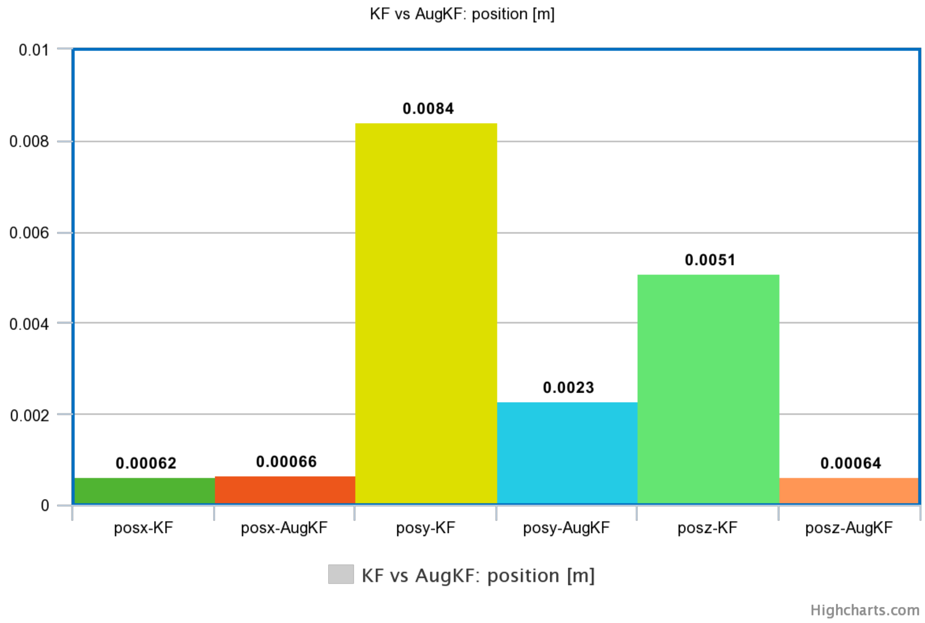


Figure 7.20: Comparison of discrepancies ”estimation-measurement” for position

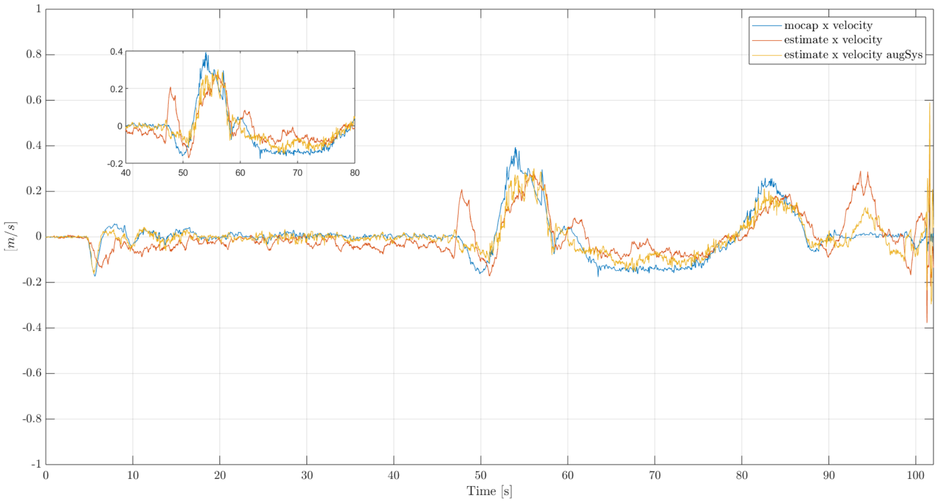


Figure 7.21: Velocity estimation along x-axis

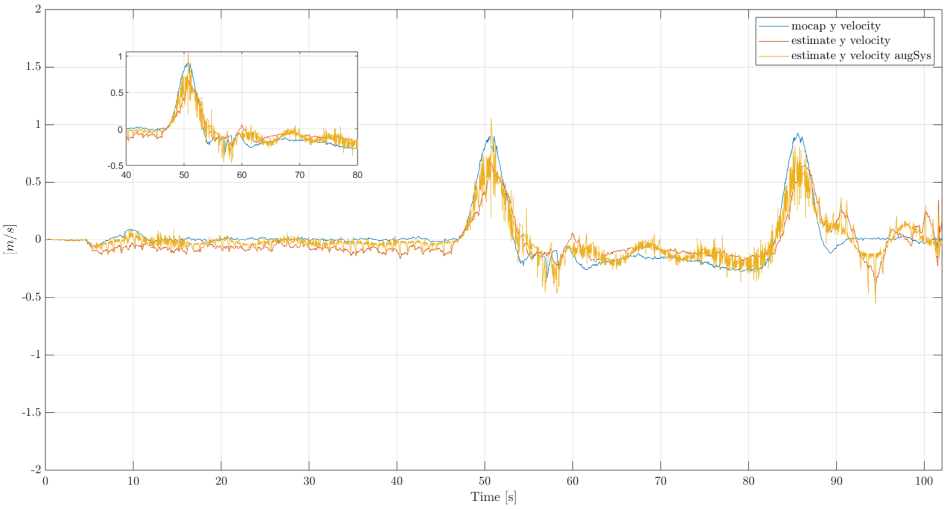


Figure 7.22: Velocity estimation along y-axis

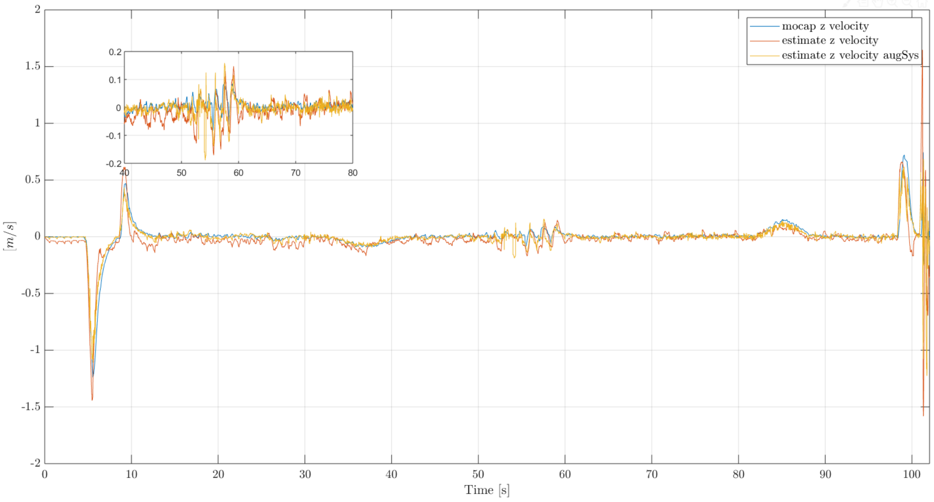


Figure 7.23: Velocity estimation along z-axis

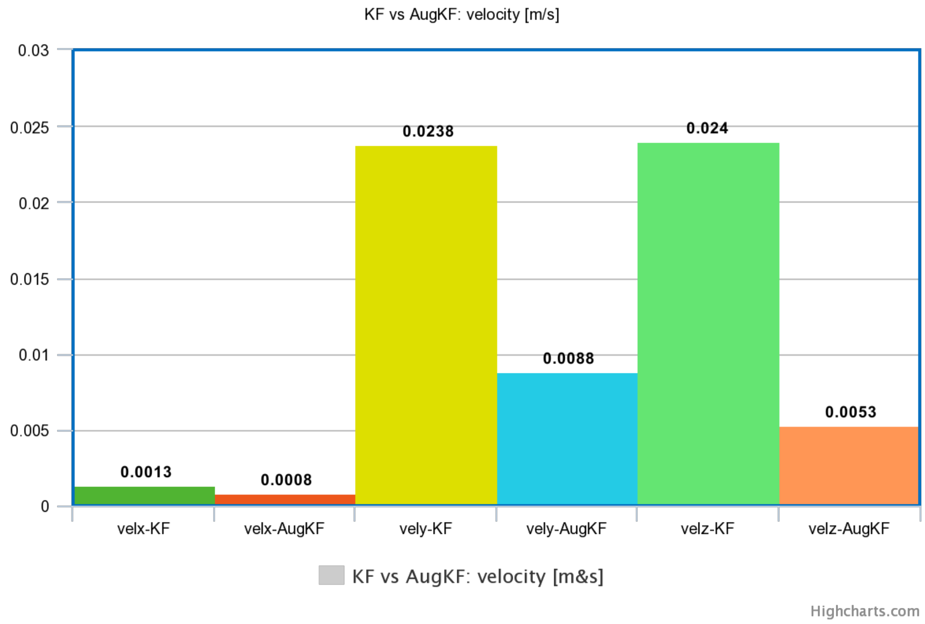


Figure 7.24: Comparison of discrepancies ”estimation-measurement” for velocity

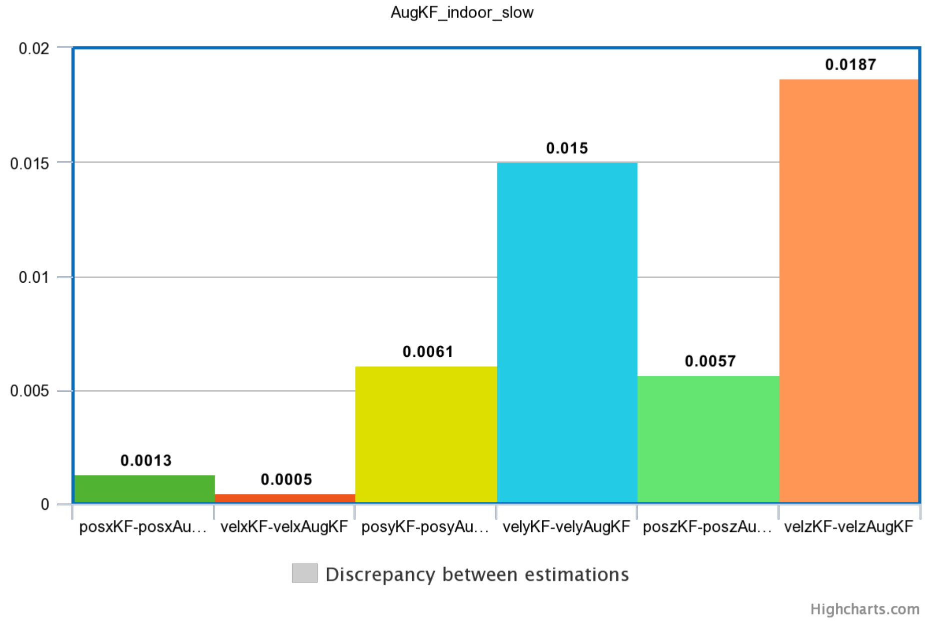


Figure 7.25: Discrepancy between estimations

better filtering process. From the same plot it is also evident that the measurement PSD becomes smaller than the PSD of the estimate, for bigger frequency in case of filtering with AugKF; then the latter provides more correct estimates for a larger interval of frequencies. In Figure 7.27 the same quantities but for the

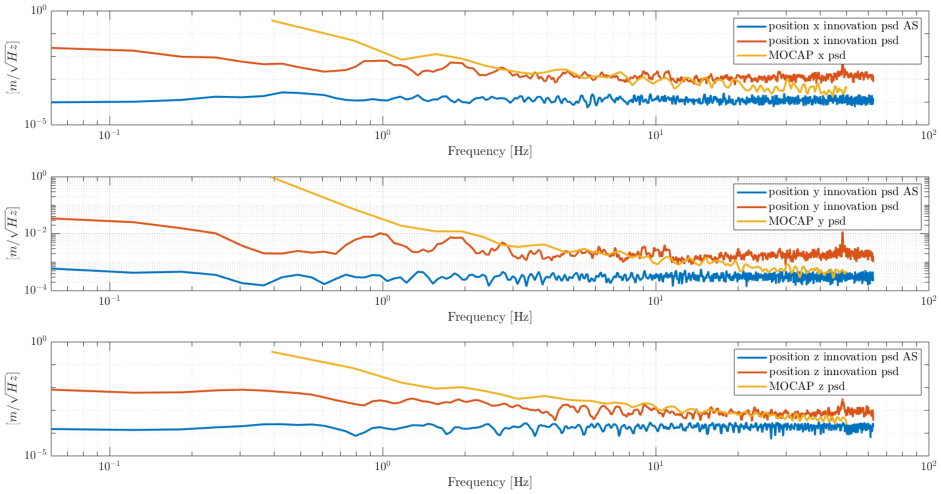


Figure 7.26: Innovation $PSDs$ for position

f are shown. Also in this case both the filters return a smaller innovation $PSDs$ amplitude respect to the PSD measurement; therefore the filters are working well. Always observing Figure 7.27 the innovation $PSDs$ resulting from KF and AugKF, are quite similar, unlike the position case. This behaviour is likely due to the fact that the matrix $\bar{H}_{augofk}$ (matrix used to define the innovation during the augmented filter algorithm in equation (4.43)) gives a heavier contribute to the definiton of the innovation than the used algorithms (KF and AugKF). In fact the elements of $\bar{H}_{augofk}$ are so small that the innovation varies slightly. The innovation when analyzed in time domain is a good performance indicator. The plots correspondent to the innovation of position and otpical flow, are marked out in Figures 7.28 and 7.29. In both the figures is evident how the AugKF presents an innovation which is bounded within the plus-minus three standard deviation band. Moreover the three standard deviation bands due to KF and AugKF are similar, then the uncertainty on the state variable estimations, using the two approaches, is similar. Finally the Kalman gains for the AugKF are shown in Figures 7.30 and 7.31. The gains due to the position measurements reach a constant value rapidly, about which oscillations are noticeable (Figure 7.30); while the gains resulting from the f measurements have a time history influenced by the Euler angles, as explained by the mathematical model seen previously (Figure 7.31), then it is difficult to draw conclusions. As said previously, the indoor flight has an effect on the quality of the measures collected by the Px4Flow. In Figure 7.32 is evident a bad quality of the f measurement, and despite this, the implemented

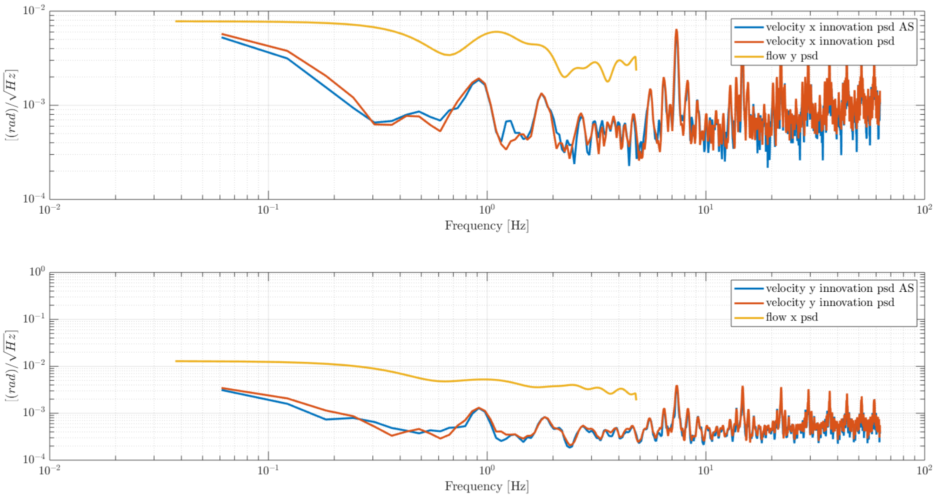


Figure 7.27: Innovation $PSDs$ for f

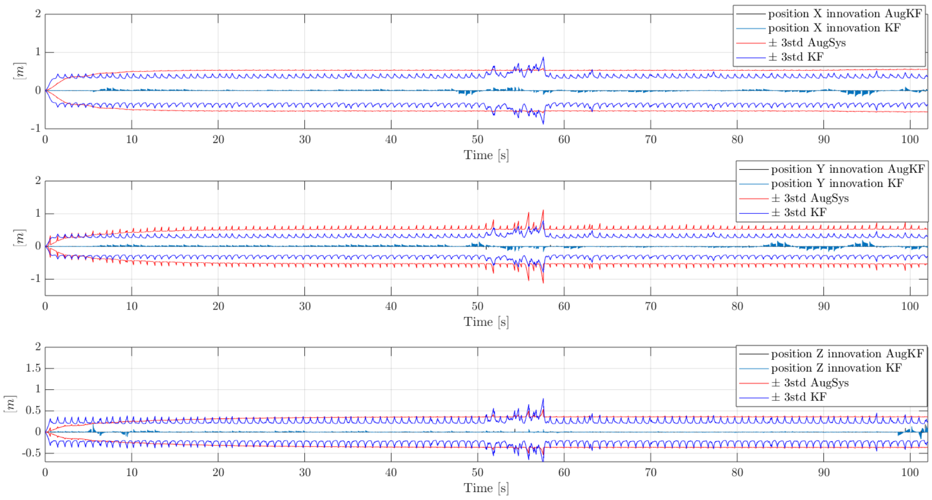


Figure 7.28: Position innovations

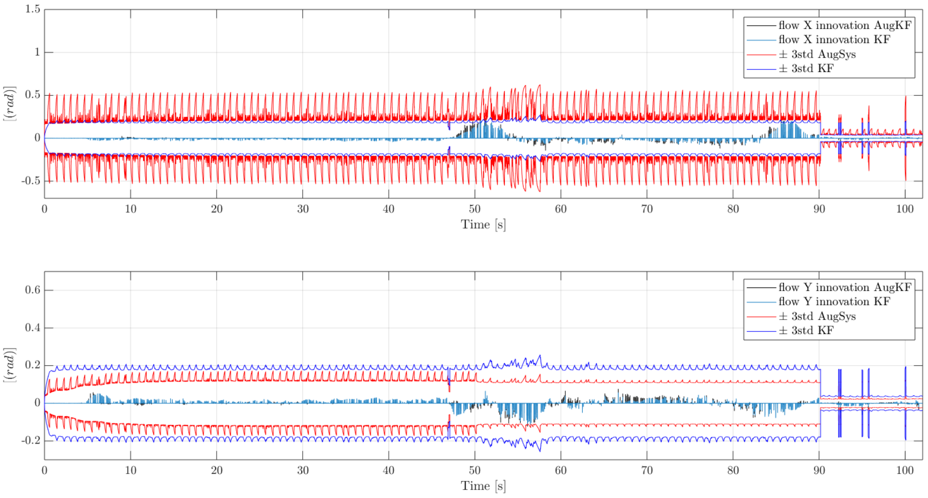


Figure 7.29: Optical flow innovations

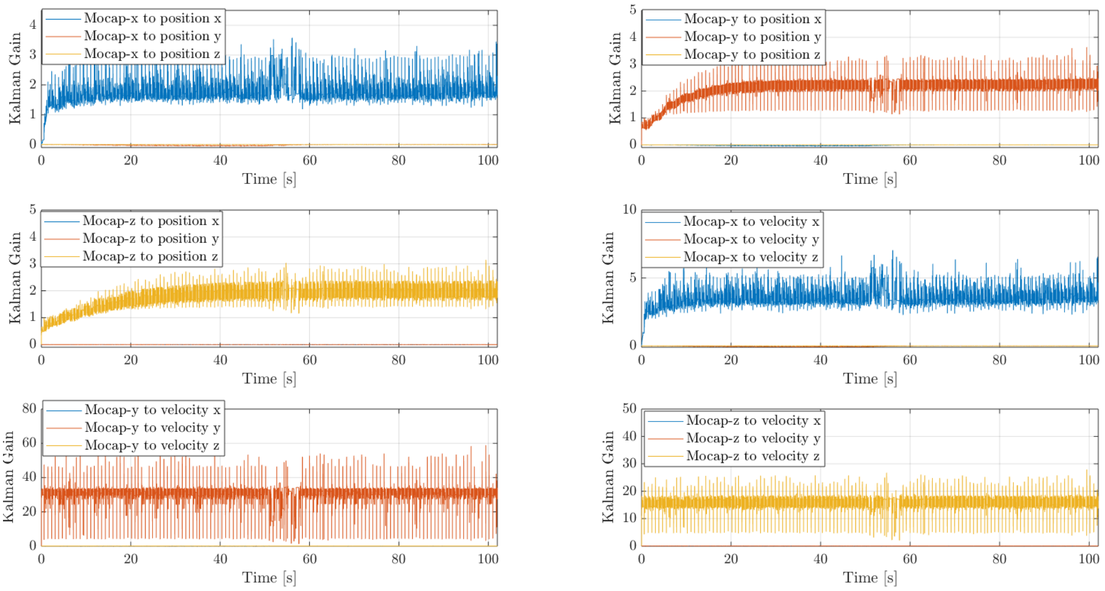


Figure 7.30: Kalman gain due to the postion measurements

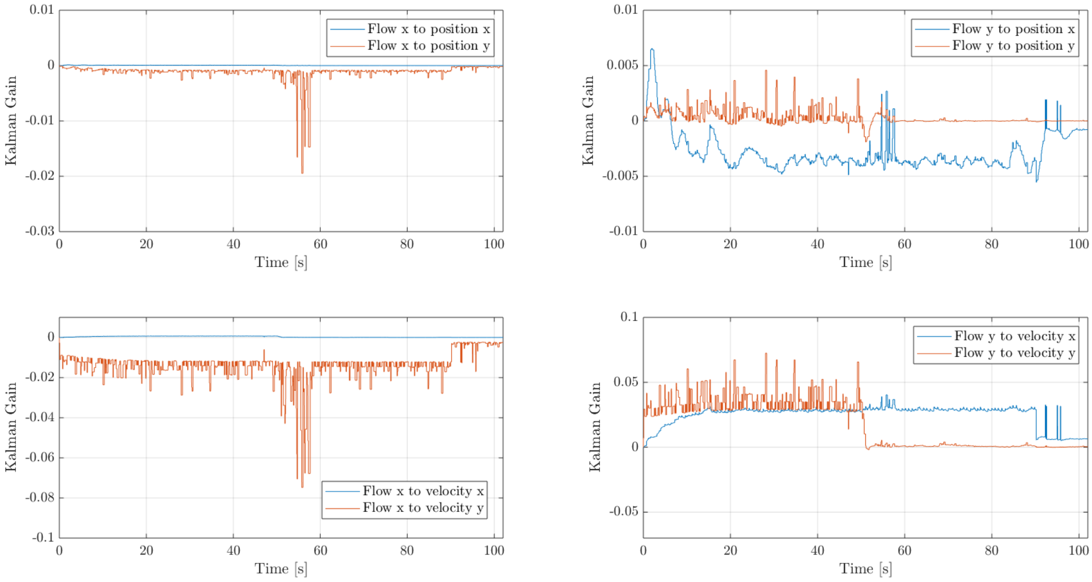


Figure 7.31: Kalman gain due to the optical flow measurements

Kalman filters have made their work correctely and the AugKF even better than the KF. And considering the results shown earlier, it is possible to conclude that the AugKF works adequetely, and that compared to the KF it provides better estimations of the UAV states.

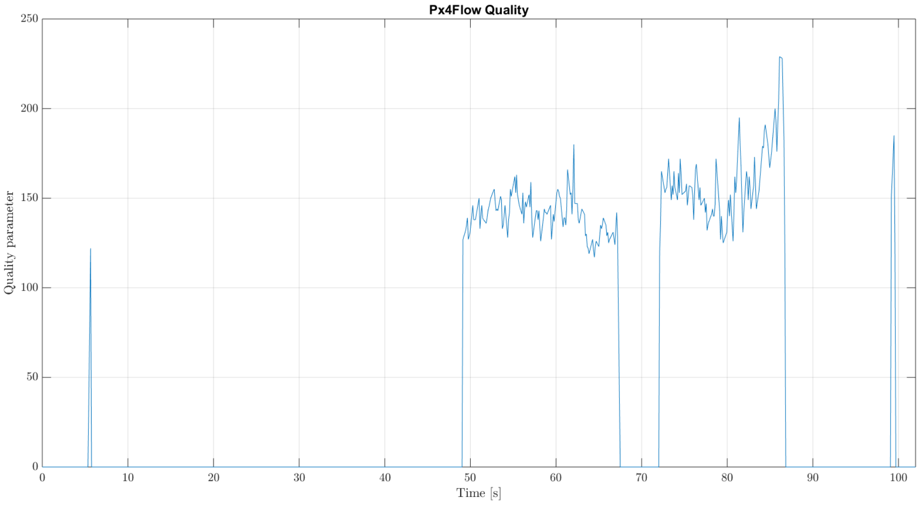


Figure 7.32: Quality of the optical flow

Fast flight

The estimates of the states p_x, p_y and p_z are reported in Figures 7.33, 7.34 and 7.35. From the plots is noticeable that the estimates provided by KF and AugKF along the axes y and z are coherent with the measurements; while the estimations along x try to follow the measurement trend, but it presents some difference respect to the measurement. The fact that the estimation in Figure 7.33 is heavly different respect to the measurement, does not mean that the estimation is bad or wrong; in fact if the filter is working well the measurement presents more uncertainty than the estimation, therefore is possible that the estimation is more true than the measurement. The discrepancy between each estimate and the measurement is collected in the histogram in Figure 7.36. From the histogram is evident a decrease of the discrepancy adopting the AugKF; this means that the augmented algorithm, respect to the KF, provides estimates which give more weight to the measure rather than to the prediction. About the estimations of the velocity see Figures 7.37, 7.38 and 7.39. The fact that the estimations in Figure 7.37 are heavly different respect to the correspondent measurement (in particular during the large oscillations), as said before, does not imply that the estimation is bad.
To get an idea about the entity of the discrepancy between each estimate and the correspondent measurement, for each axis, see Figure 7.40. The difference between the estimates is collected in the histogram in 7.41. In figures 7.42 and 7.43 the innovation $PSDs$ are plotted. As for the slow flight test case, for the innovation PSD correspondent to the position, is evident that the AugKF presents a better filtering process than that of the KF, as well as less uncertainty than that of the measurement. About the f the $PSDs$ of KF and AugKF are very similar, as seen for the slow flight. The plots correspondent to the innovation time history for the position are reported in Figure 7.44, while for the f in Figure 7.45. From

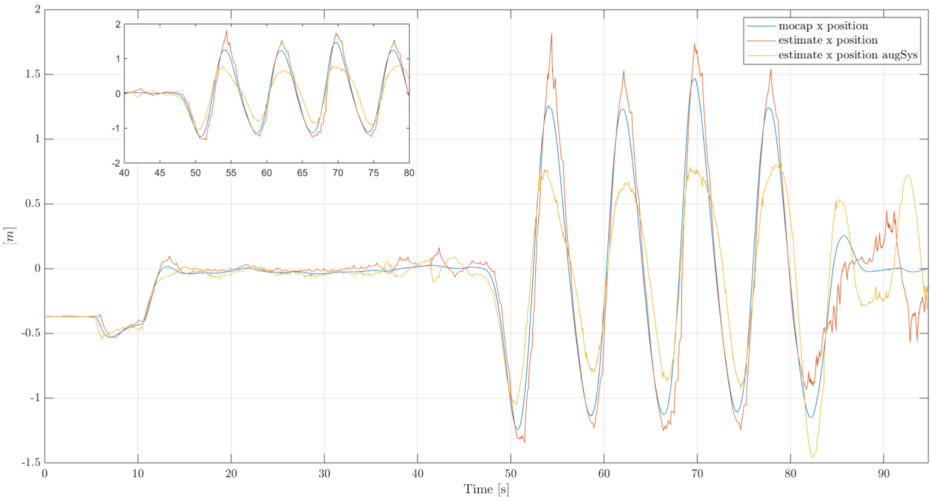


Figure 7.33: Position estimation along x-axis

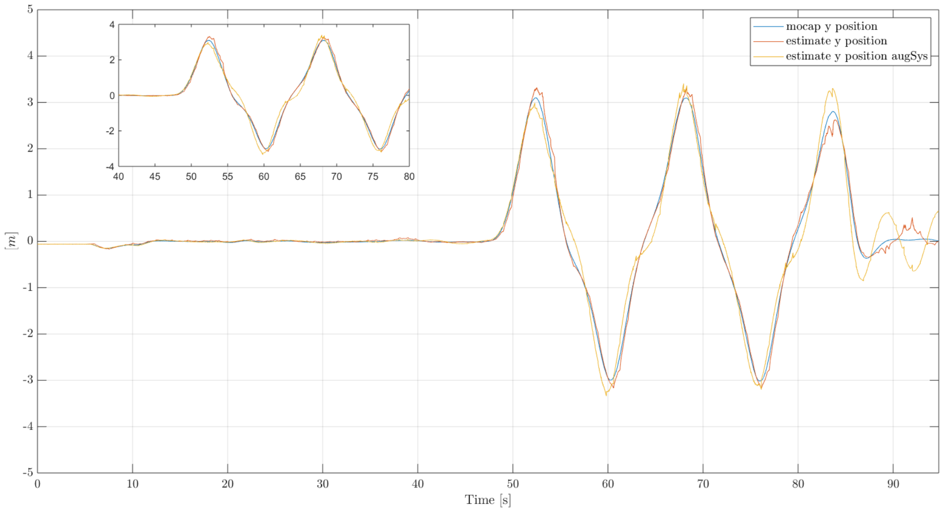


Figure 7.34: Position estimation along y-axis

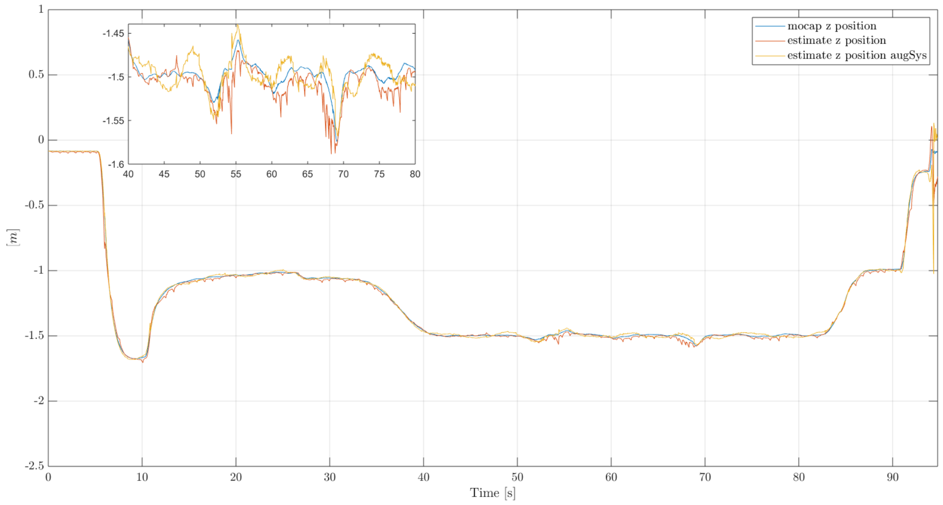


Figure 7.35: Position estimation along z-axis

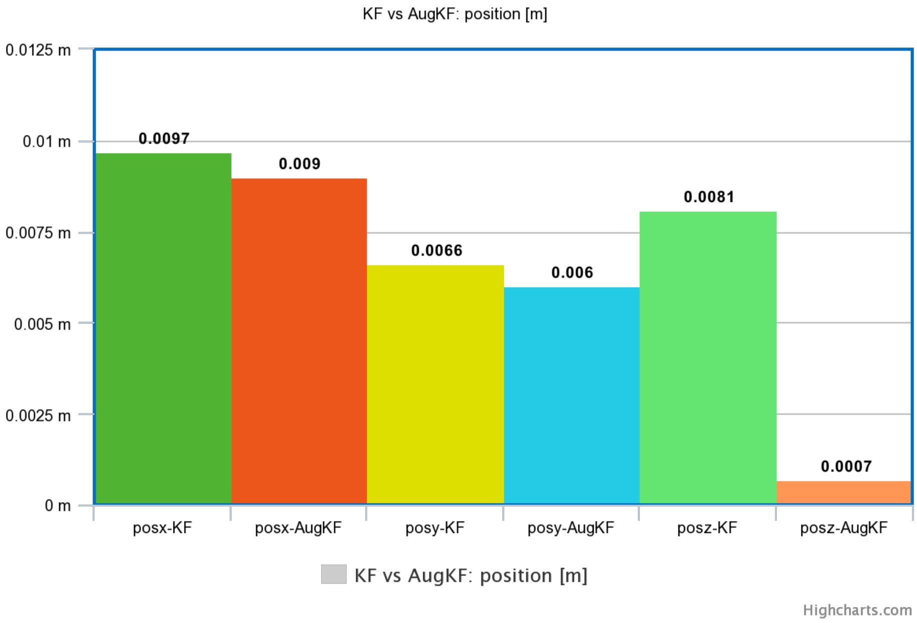


Figure 7.36: Comparison of discrepancies ”estimation-measurement” for position

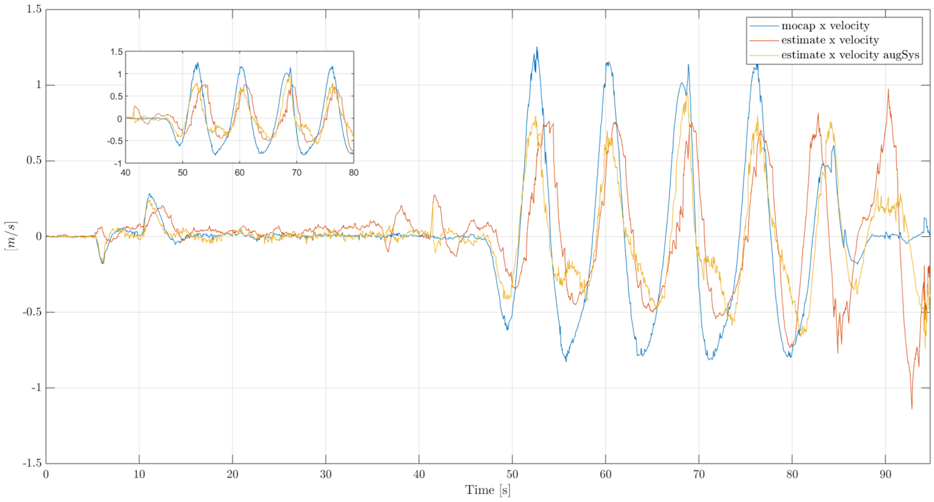


Figure 7.37: Velocity estimation along x-axis

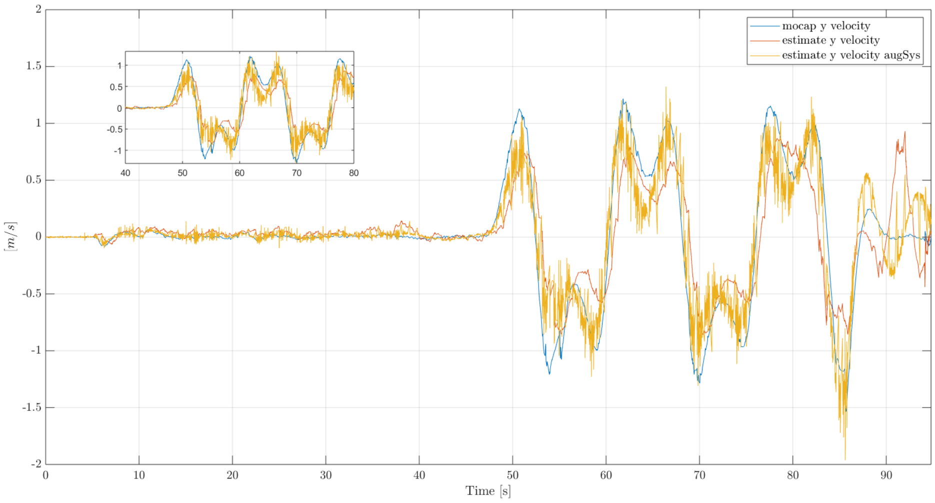


Figure 7.38: Velocity estimation along y-axis

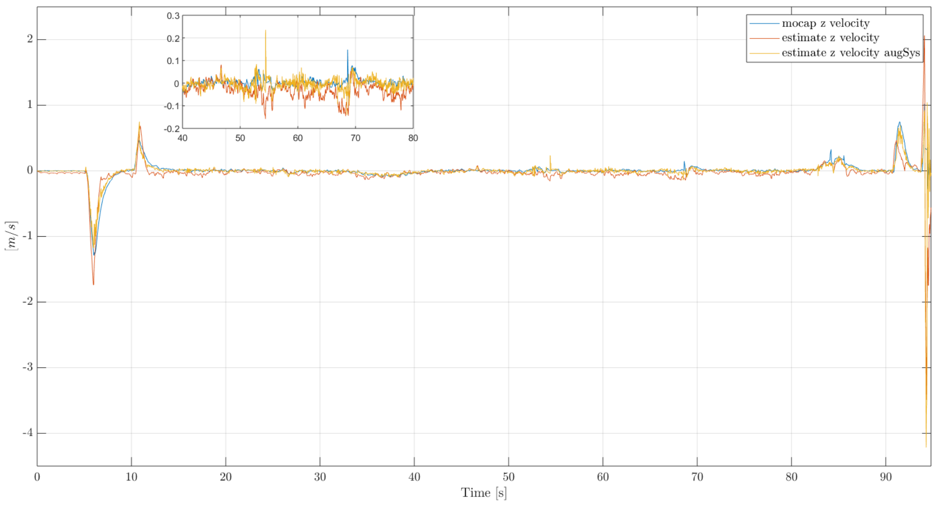


Figure 7.39: Velocity estimation along z-axis

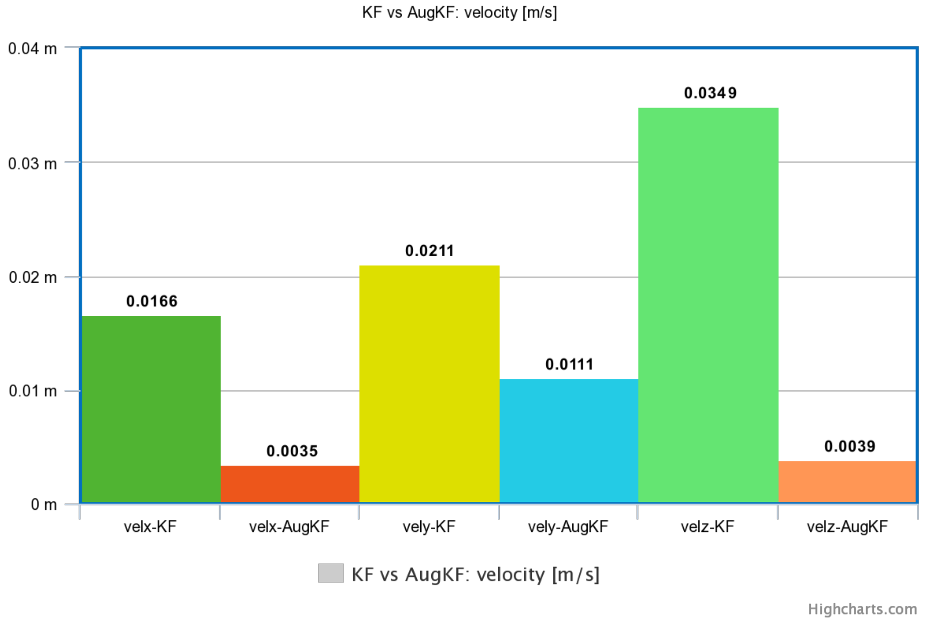


Figure 7.40: Comparison of discrepancies ”estimation-measurement” for velocity

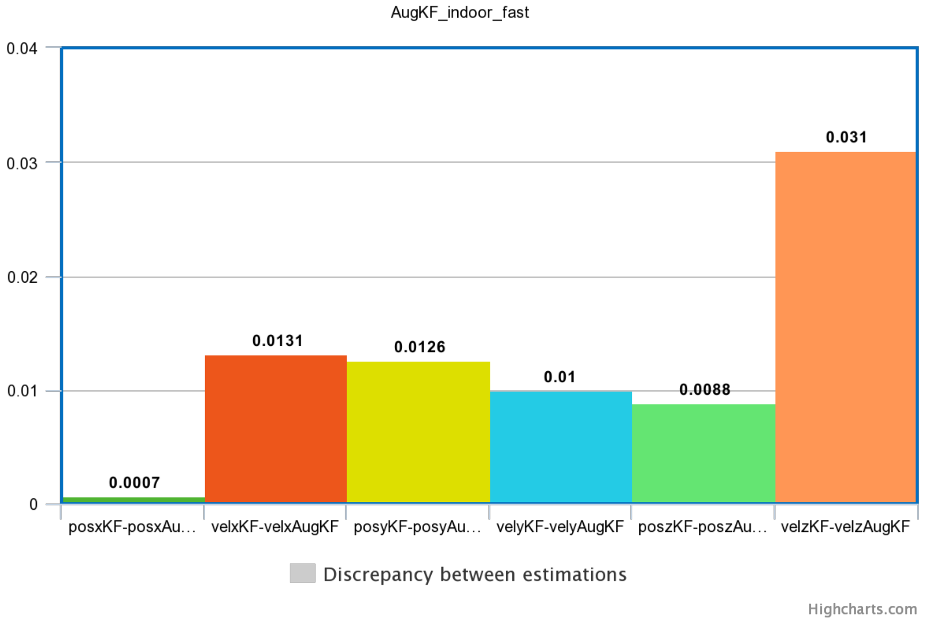


Figure 7.41: Discrepancy between estimations

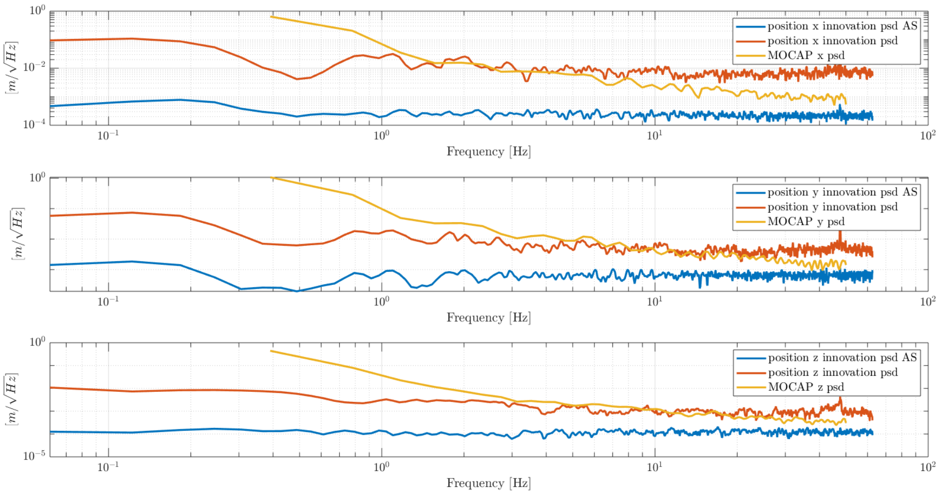


Figure 7.42: Innovation $PSDs$ for position

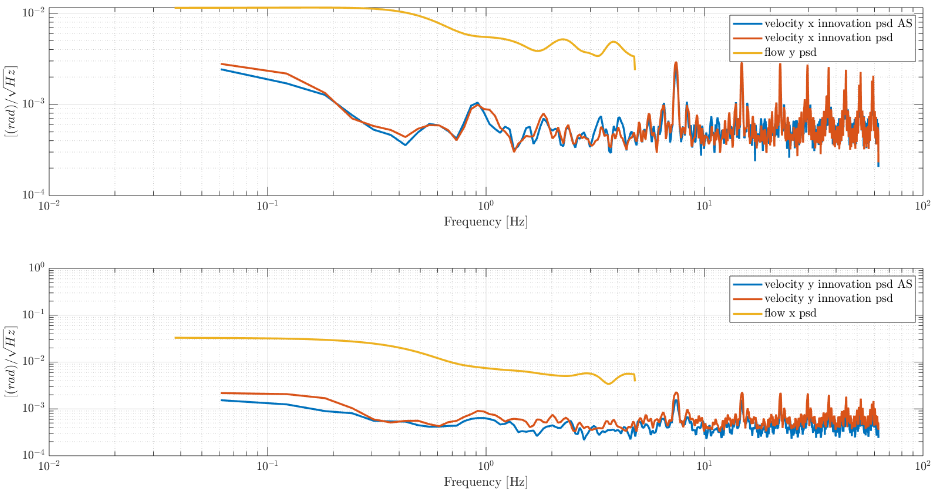


Figure 7.43: Innovation $PSDs$ for optical flow

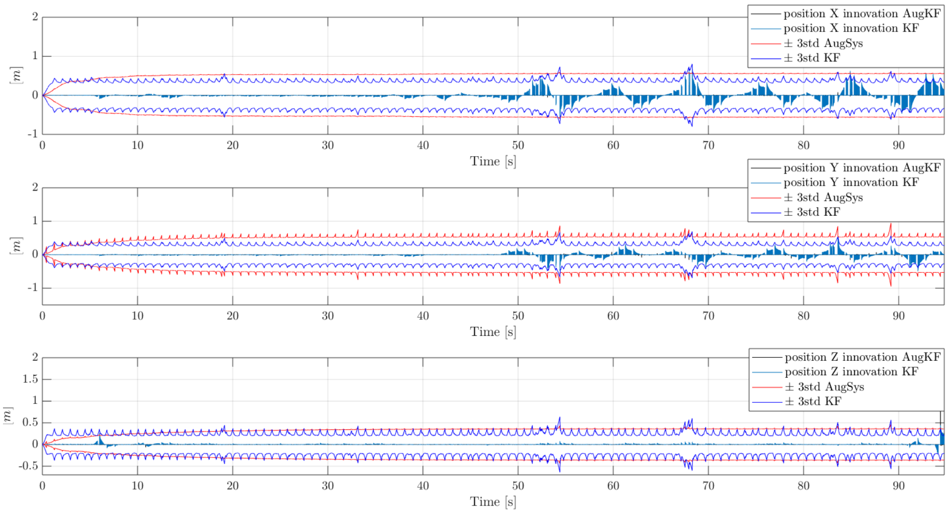


Figure 7.44: Position innovation

Figure 7.44 is observable that the innovation of the KF is bounded in the defined band, a part for some time instants; but the innovation of the AugKF remains bounded for each instant. For the f, in Figure 7.45, is evident how both the estimation processes return an innovation time history with similar amplitude. This similarity in time domain, is the other side of the behaviour observed in the plots of the f innovation $PSDs$, where both the filters had similar behaviours.

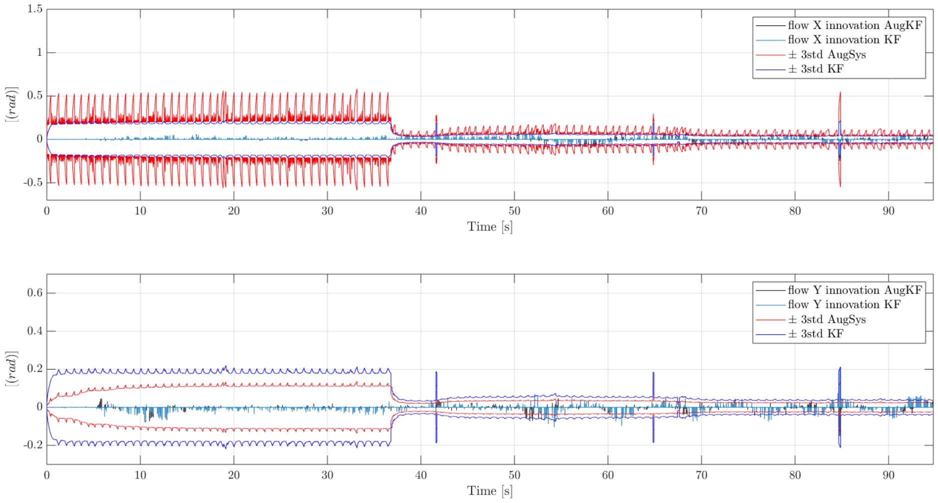


Figure 7.45: Optical flow innovation

The gains used by the AugKF during the filtering process, are represented in Figures 7.46 and 7.47. The estimation of the UAV states, have been carried out by adopting the f measurements with the quality marked out in Figure 7.48.

Up to now the estimations of the UAV states, have been realized by using the

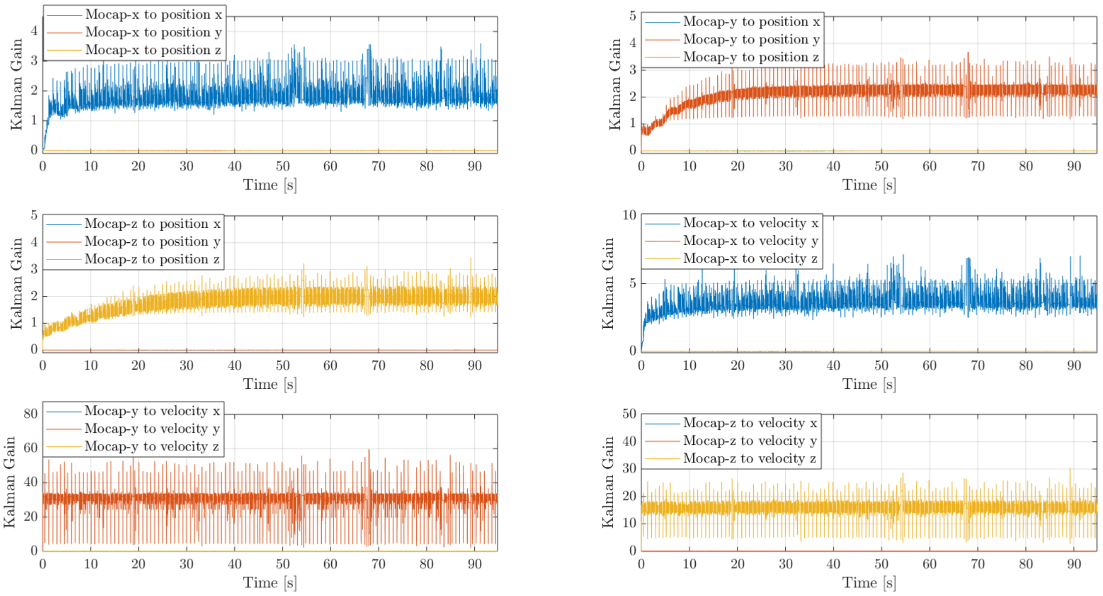


Figure 7.46: Kalman gain due to the position measurements

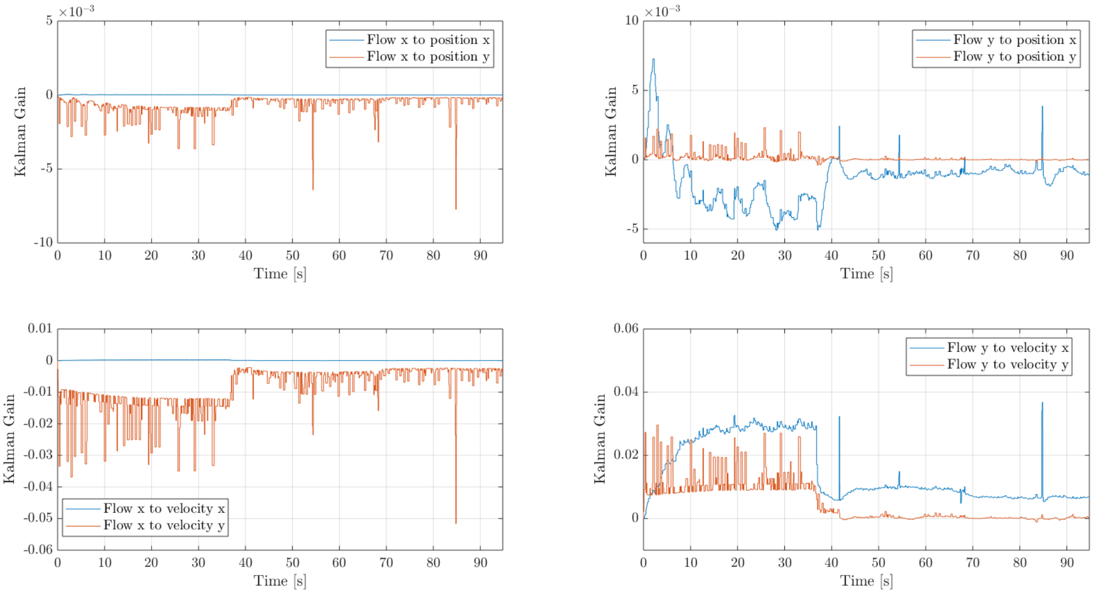


Figure 7.47: Kalman gain due to the position measurements

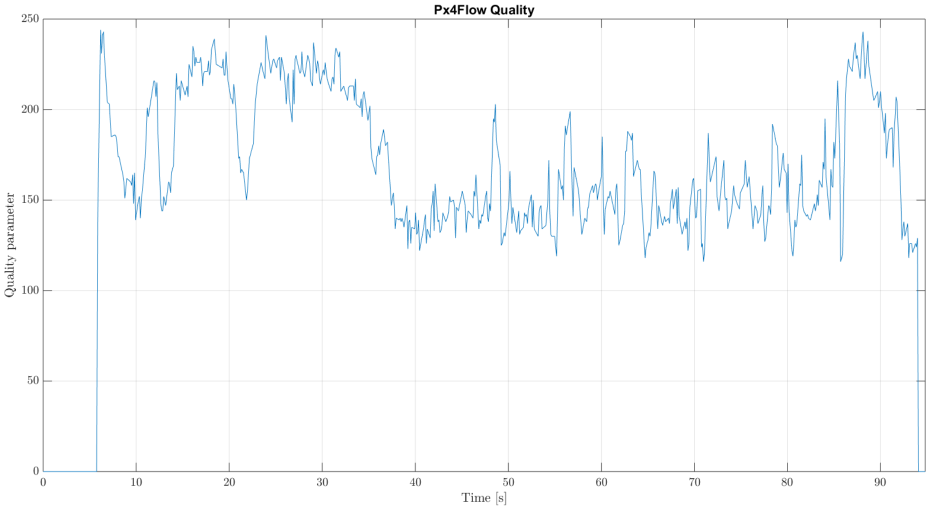


Figure 7.48: Quality of the optical flow

augmented Kalman filter, which has been compared to the KF. In particular the filters have been tested letting the UAV to fly in outdoor and indoor enviroments. In all the tests the AugKF presents coherent estimates respect to the reference, and correct estimates since the PSD of the innvations is smaller than that of the measurements, and the time history of the innovation is bounded in the plus-minus three standard deviation. This ensures that the innovation has a behavior similar to a white noise then the filter is working right. Using the AugKF respect to the KF, an important improvement of the position and velocity estimates along the axis z is obtained, and moreover a reduction of the innovation $PSDs$ amplitude. Comparing the discrepancies ”estimation-measurement” correspondent to the velocity, of the slow and the fast tests, similar results are noticeable for the two flights, when the AugKF is adopted; similar result is observable for the discrepancies correspondent to the postion when the AugKF is used. Moreover, the KF estimations and the correspondent estimations by the AugKF, when compared give the following results: for the outdoor flight there is large difference between the position estimations along z axis; while for both the indoor flights there is a large change of the velocity estimations, along z.

7.2 Estimate results: adaptive Kalman filter

In this section the same flights analyzed above, will be tested with the adaptive Kalman filter. The purpose is to understand if an adaptive algorithm will return right estimates and in that case will be compared with the version ”KF”.

7.2.1 Outdoor flight

Using the adaptive approach the estimates of the states p_x, p_y and p_z are depicted in Figures 7.49, 7.50 and 7.51. Like using the AugKF, in this approach is evident how the estimate of the position along the axis x, is coherent with the GPS measurements. The same is true for the axes y and z. Observing Figure 7.51 is noticeable an improvement of the estimate, respect to that provided by the KF. This is a good result considering that the z-axis is the most critical one since it is known that the worse performances of the GPS are along it.

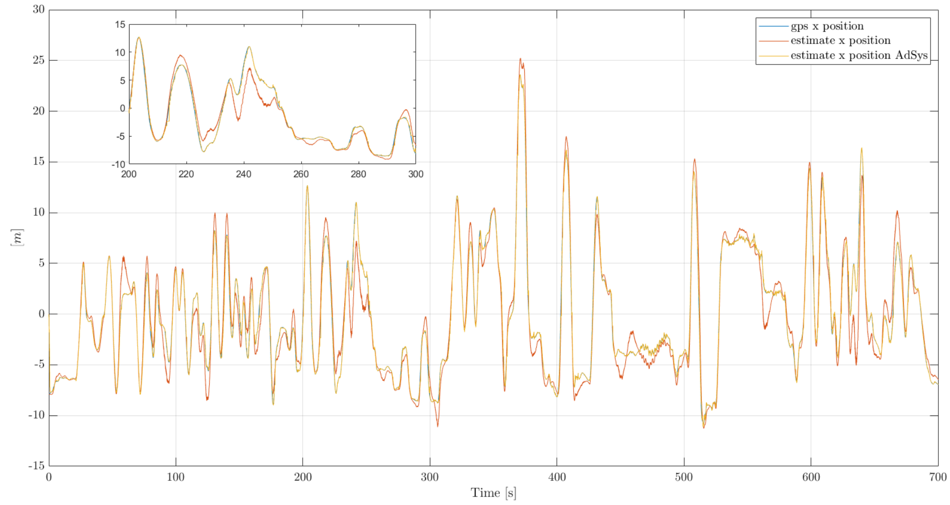


Figure 7.49: Position estimation alog x-axis

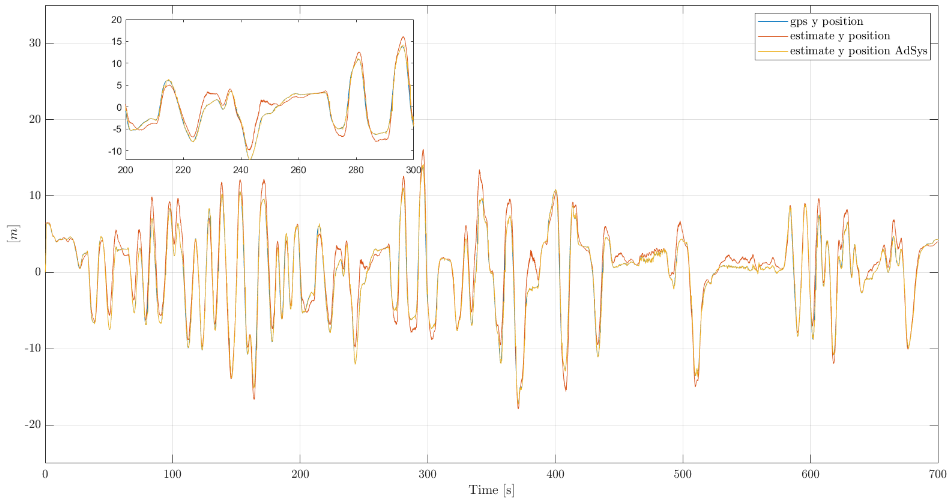


Figure 7.50: Position estimation along y-axis

The discrepancies ”estimation-measurement” for each axis, are collected in the

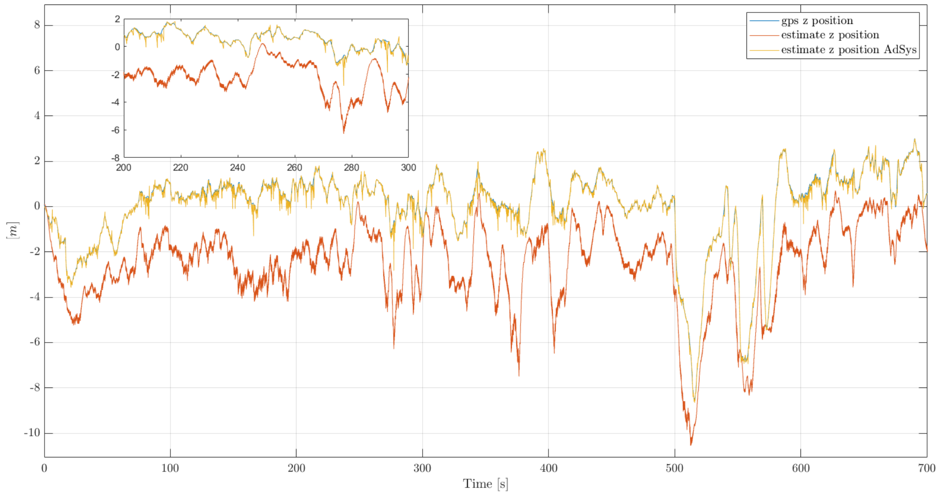


Figure 7.51: Position estimation along z-axis

histogram in Figure 7.52.

Also the estimations of the velocity along each axis seem to be correct in comparison to the measurements provided by the GPS. In Figures 7.53, 7.54 and 7.55 the states v_x, v_y and v_z respectively are marked out. The estimation along the axis z, when the AdKF is used, improves. To have an idea about the discrepancy between the estimates and the measurements see the histogram in Figure 7.56. As seen previously, the discrepancy between the different estimations, can be of interest (Figure 7.57). In order to evaluate the correctness of the work made by the filter, is necessary to observe the PSD of the innovation. Recalling what it has been already said, a good filter must imply an innovation PSD with trend similar to a white noise as well as an amplitude smaller than the PSD of the measurement. The other aspect exploited to recognize if the filter is working well or less, is the time history of the innovation. In that case the trend must be bounded between the minus-plus three standard deviation band, to ensure the normal distribution of the innovation. In Figure 7.58 the innovation PSD of the position is reported. As it is possible to see in Figure 7.58 the $PSDs$ have a trend which is almost flat; moreover both the filters, but in particular the AdKF, present innovation $PSDs$ which are smaller than those of the measurements. In Figure 7.59 the $PSDs$ correspondent to the f, are marked out, and similar conclusions are made. The time history of the innovation is plotted in Figures 7.60 and 7.61. The adaptive Kalman filter presents a position innovation which is bounded in the considered band, for all the time instants and along each axis. Similarly the KF, but for some time instants the innovation exceeds the bound. For the f the same thing happens: the innovations resulting from the KF remain bounded a part for some time instants, while those from the AdKF are always bounded.

Finally the Kalman gains are plotted in Figures 7.62 and 7.63 for the AdKF. From

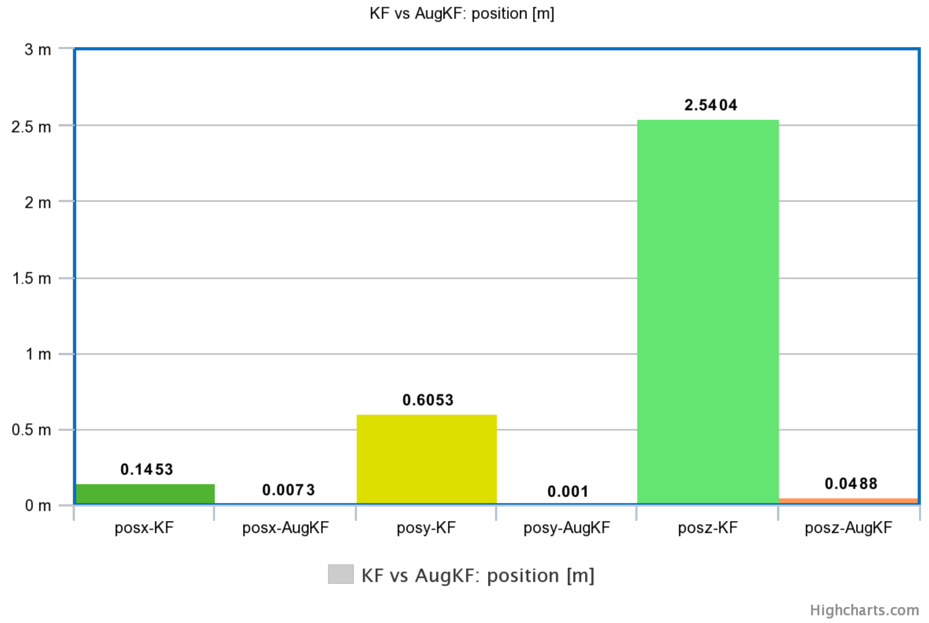


Figure 7.52: Comparison of discrepancies ”estimation-measurement” for position

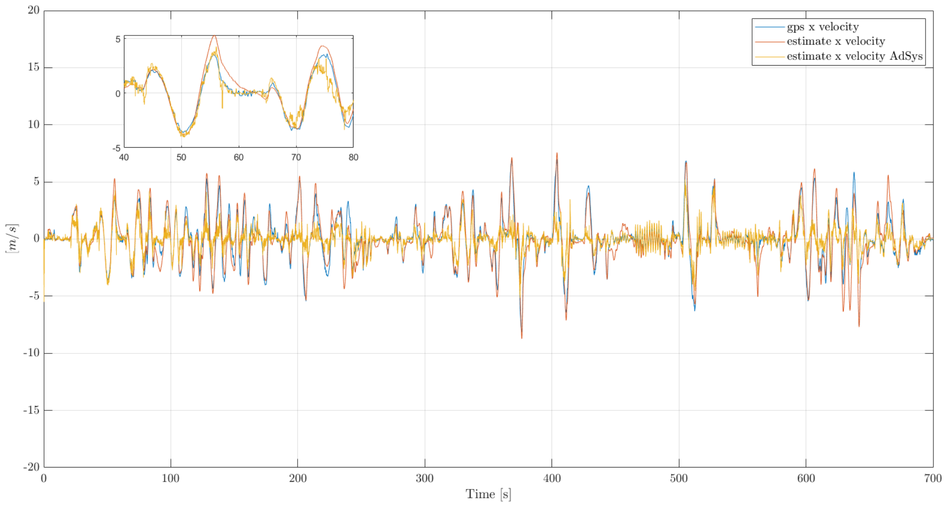


Figure 7.53: Velocity estimation along x-axis

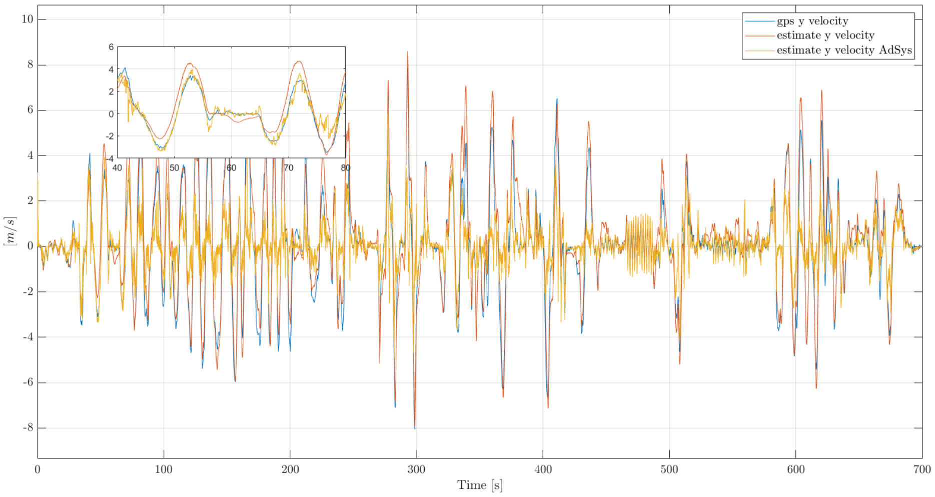


Figure 7.54: Velocity estimation along y-axis

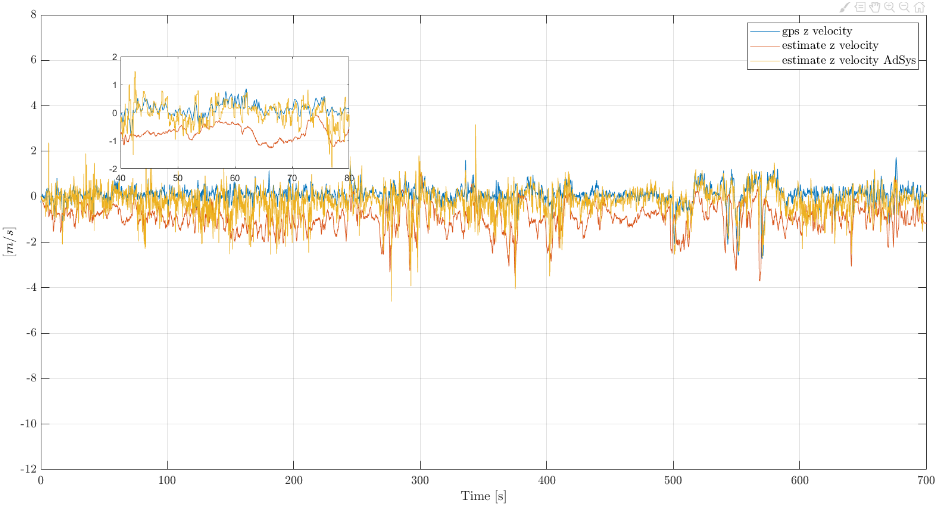


Figure 7.55: Velocity estimation along z-axis

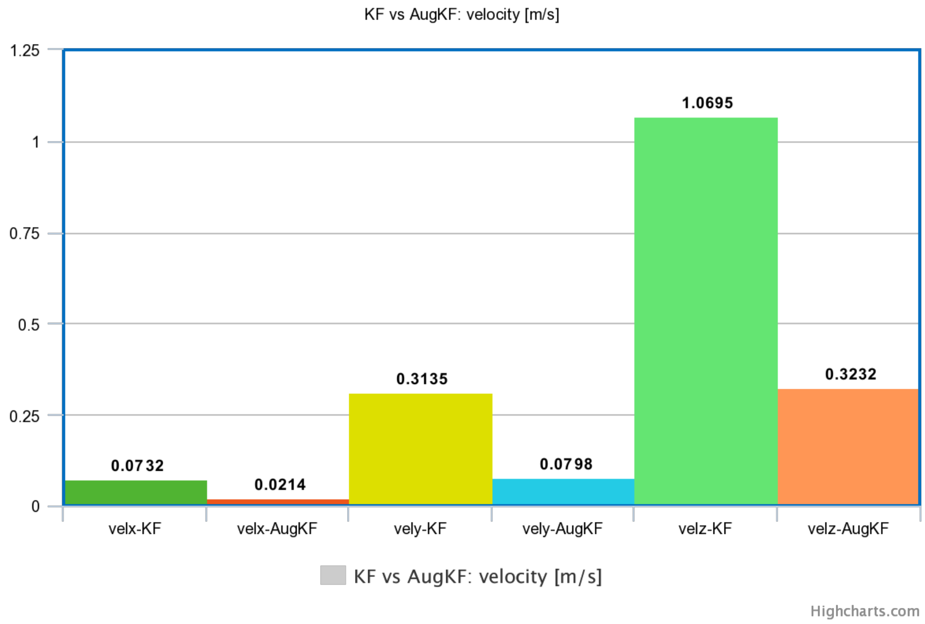


Figure 7.56: Comparison of discrepancies ”estimation-measurement” for velocity

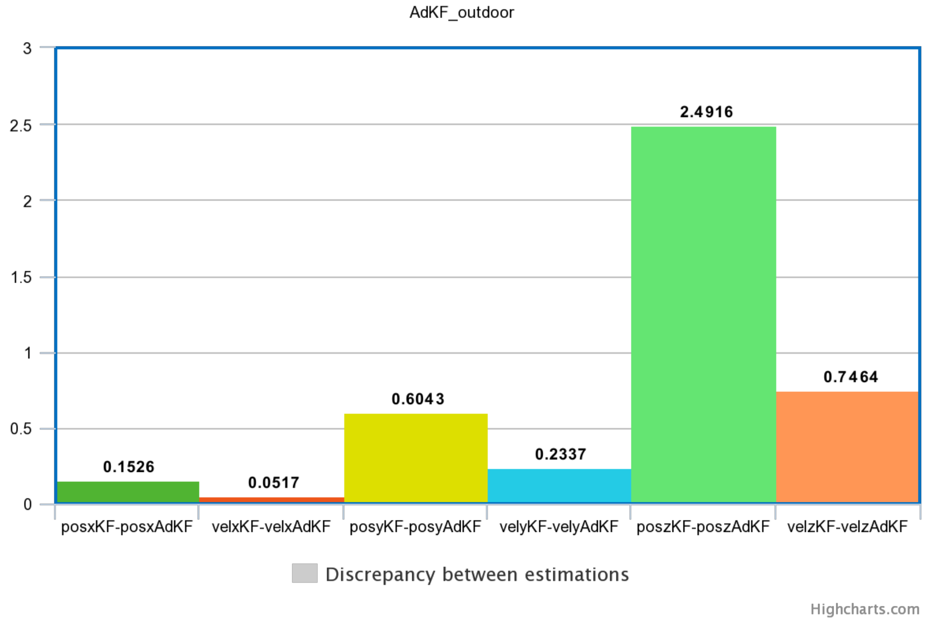


Figure 7.57: Discrepancy between estimations

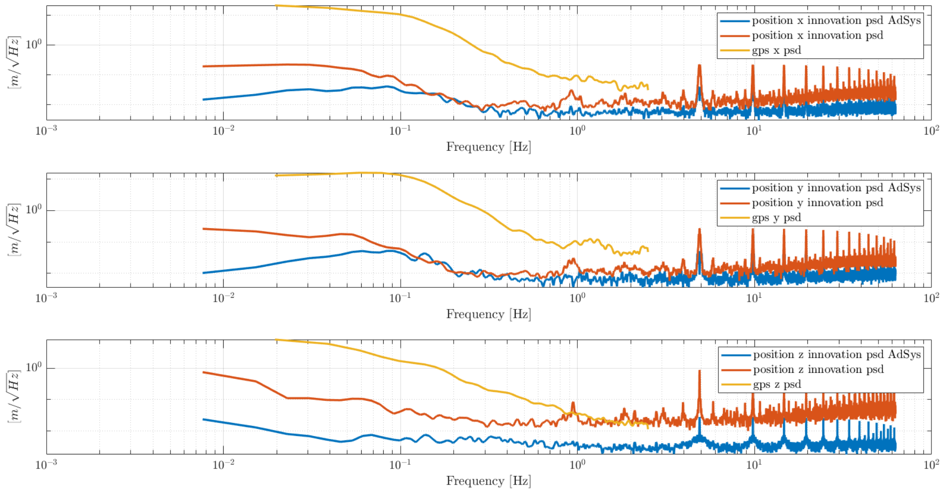


Figure 7.58: Innovation $PSDs$ for position

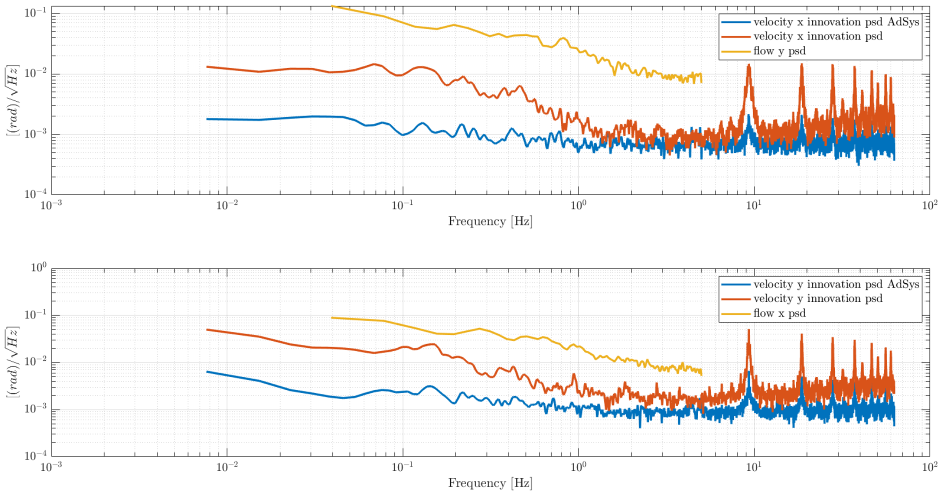


Figure 7.59: Innovation $PSDs$ for optical flow

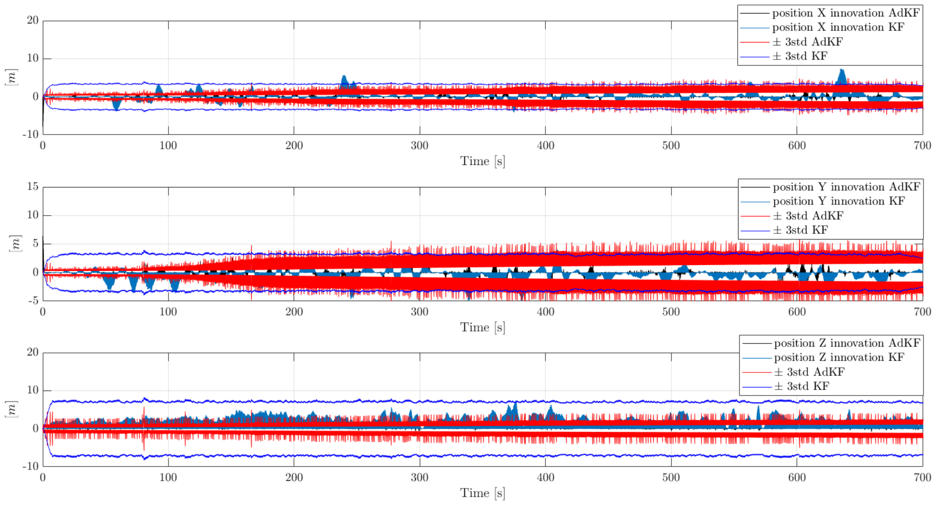


Figure 7.60: Position innovation

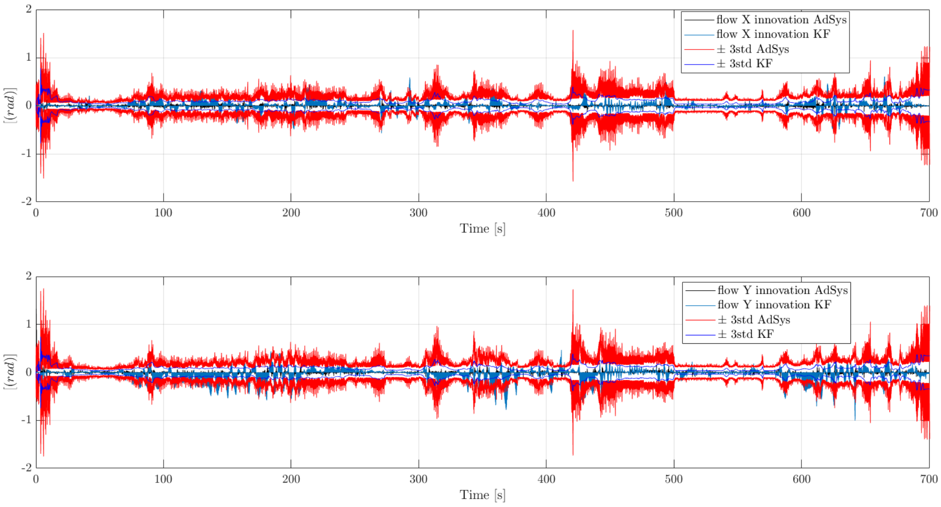


Figure 7.61: Optical flow innovation

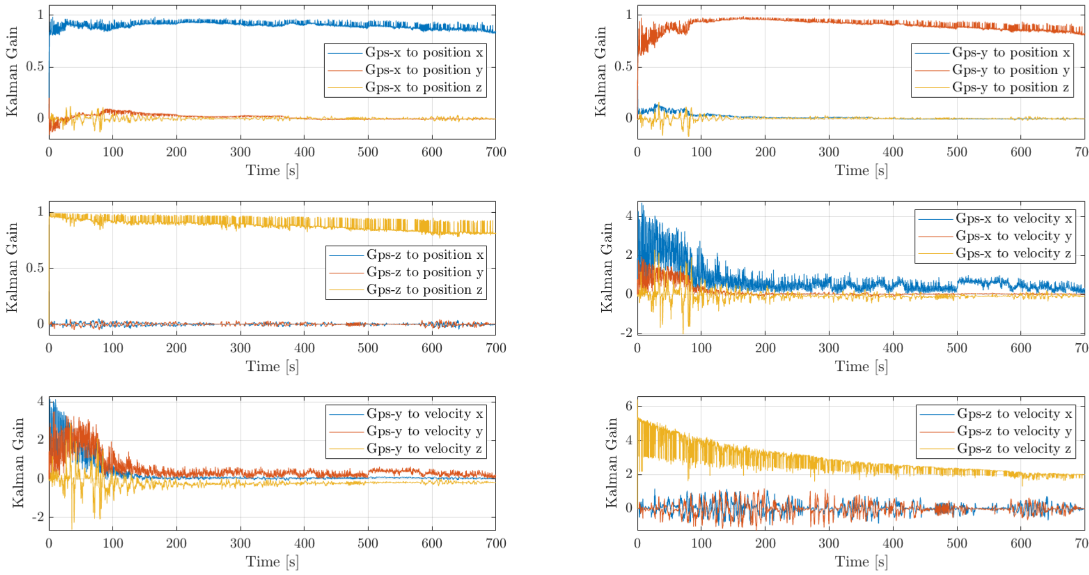


Figure 7.62: Kalman gain due to the position measurements

Figure 7.62 is noticeable a behaviour of the Kalman gain, in which the measurement along a certain axis gives a not negligible contribute to the estimation of the other axes. About the Kalman gains resulting by the f measurements is difficult to draw conclusions, since their time history is influenced by the Euler angles.

Considering all the aspects visited previously, the AdKF in outdoor flight works correctely and in comparison to the KF, presents more certain estimations.

7.2.2 Indoor flight

In this part the adaptive filter is used to perform the estimation of the UAV states, during a slow and a fast flight.

Slow flight

The adaptive algorithm when applied to the slow flight, the following estimations for the position are obtained. See Figures 7.64, 7.65 and 7.66 for p_x, p_y and p_z respectively. The adaptive Kalman filter adopted during the slow dynamics flight, at first glance seems to work coherently with the reference. This is true for all the three axes. Observing the plots of the position estimations, oscillations with small amplitudes are evident during all the time history. As said before this could be due to the fact that the MoCap sampling frequency is 100 Hz, while the Kalman filter runs at 125 Hz. These amplitudes are clear not only for the estimations provided by the KF but also for those given by the AdKF. The similar behaviour between the two approaches is due to the fact that the adaptive

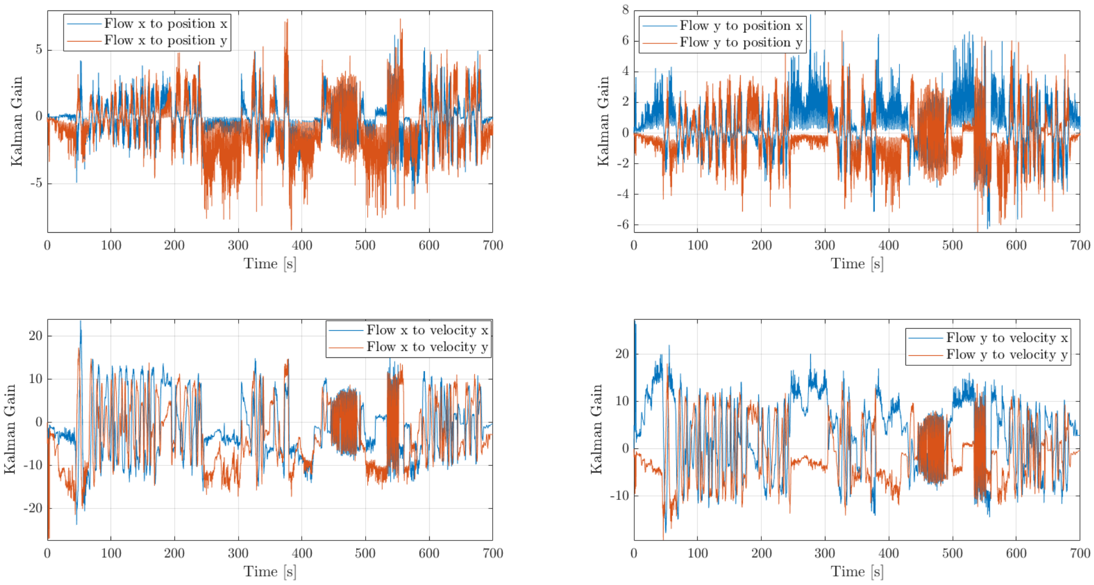


Figure 7.63: Kalman gain due to the optical flow measurements

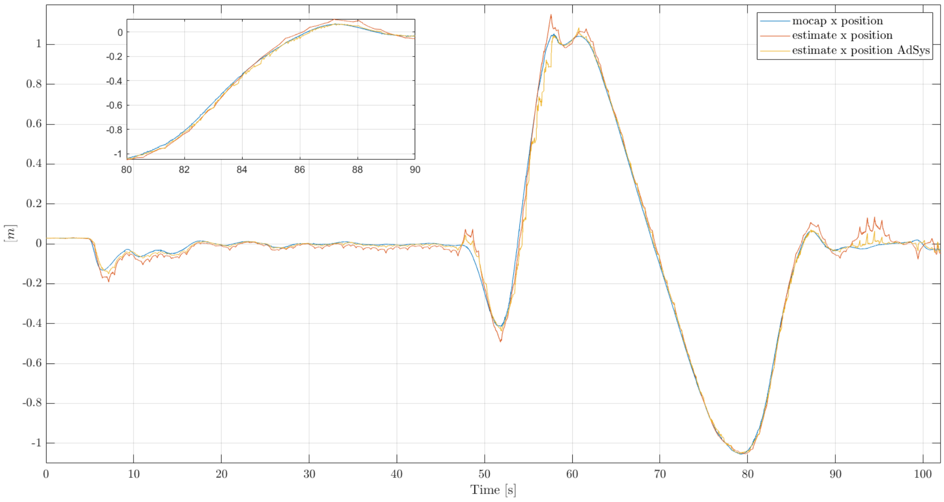


Figure 7.64: Position estimation along x-axis

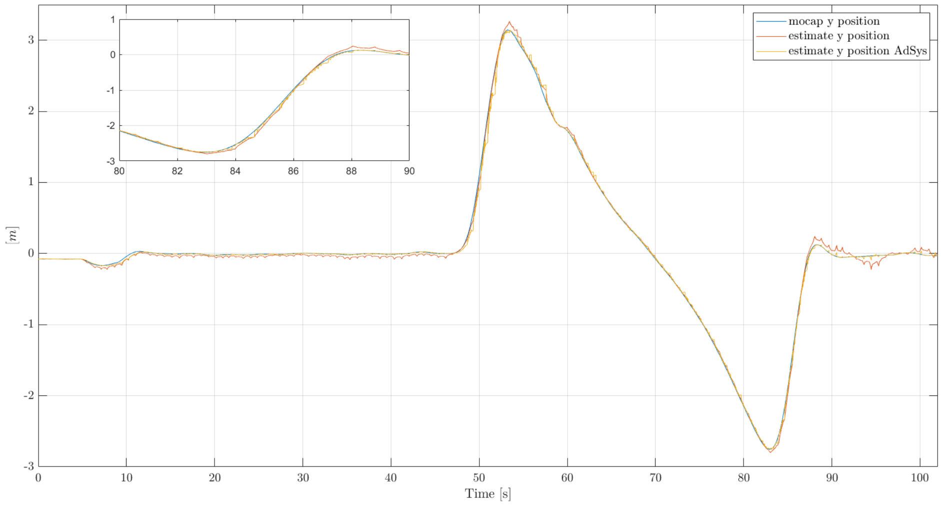


Figure 7.65: Position estimation along y-axis

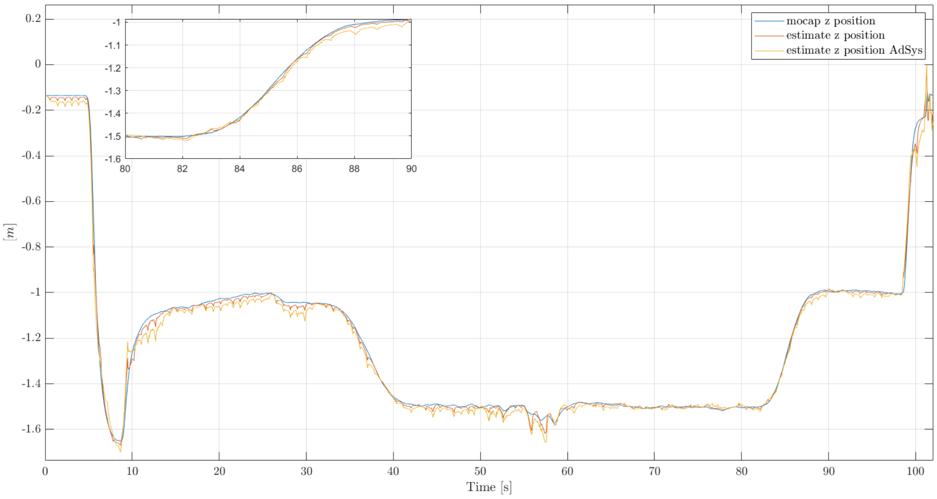


Figure 7.66: Position estimation along z-axis

algorithm becomes very similar to the algorithm implemented in the KF, when α has a value close to one, like in this case.

Looking at the values in the histogram in Figure 7.67 is noticeable an improvement of the estimates along the axes x and y, but a worsening along z. This does not mean that the AdKF is not a good observer, since the conclusions depend on the innovaion PSD behaviour. In Figures 7.68, 7.69 and 7.70 the estimations of v_x, v_y and v_z respectively are shown. Similarly to the position estimates, for the velocity estimations the oscillations due to the use of the MoCap, are noticeable.

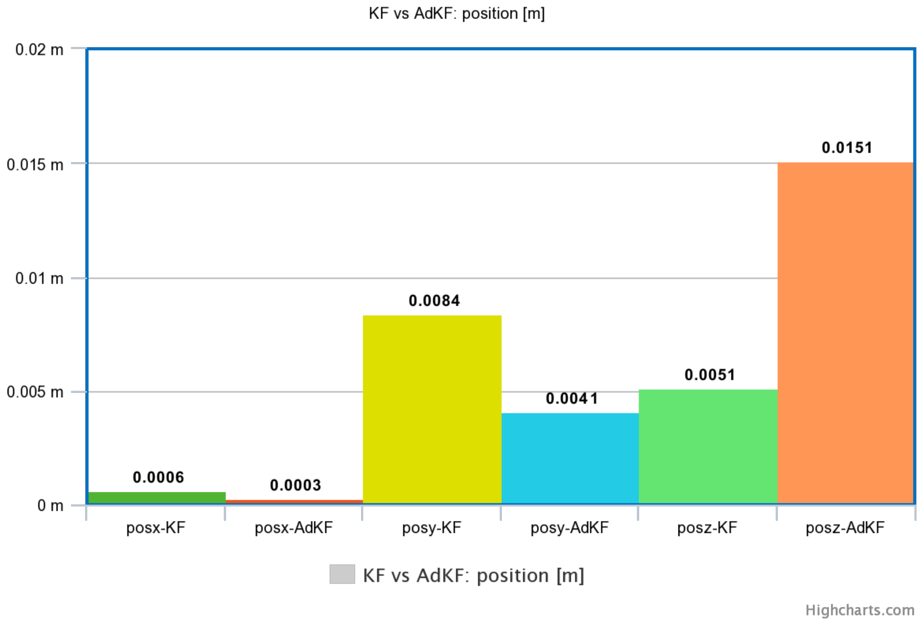


Figure 7.67: Comparison of discrepancies "estimation-measurement" for position

The discrepancies "estimation-measurement" correspondent to the estimations of the velocity, are collected in the histogram in Figure 7.71.

Meaningful values are those correspondent to the discrepancies between the various estimations (see Figure 7.72). As said before, in order to judge if the filter is working well, is necessary to observe the plots of the innovation PSD and time history. In Figure 7.73 the innovation $PSDs$ for the position, are shown. As said just before, the position estimations provided by the AdKF are better only along x and y, while along z, seems that the KF works better. This implies the inversion of the third plot in Figure 7.73 respect to the first two.

In Figure 7.74 the innovation $PSDs$ for the f are shown. The two filters present similar innovation $PSDs$, even if those correspondent to the AdKF have a smaller amplitude. This implies that the estimations of the velocity provided by the AdKF are expected to be more correct respect to those provided by the KF.

In Figures 7.75 and 7.76 the innovation time evolutions for position and velocity are plotted. A part for some time instants, the innovations stay bounded in

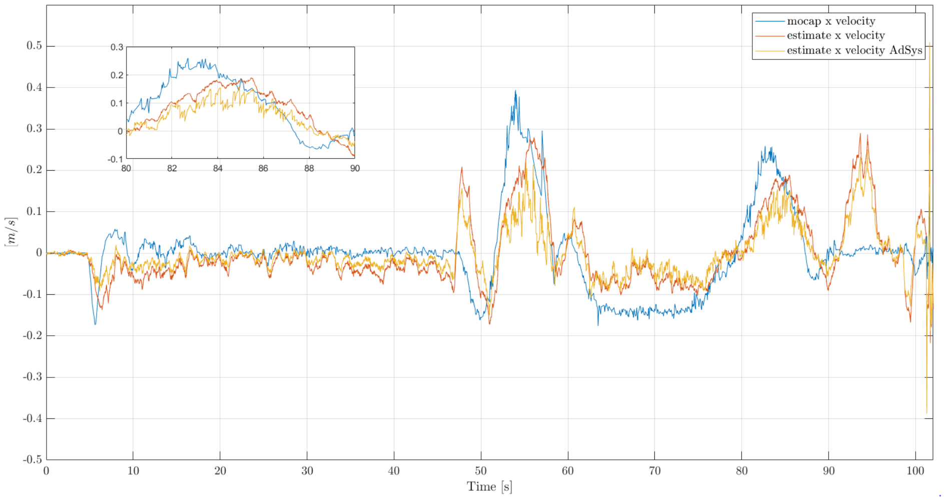


Figure 7.68: Velocity estimation along x-axis

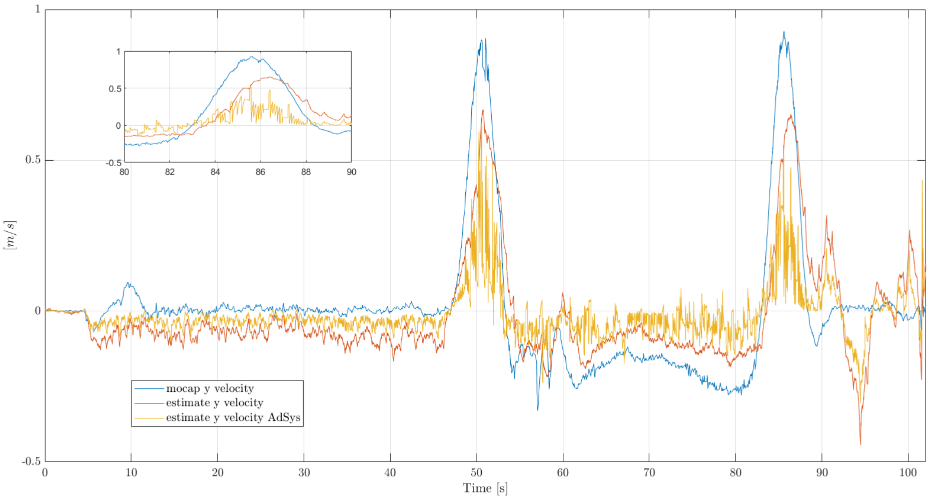


Figure 7.69: Velocity estimation along y-axis

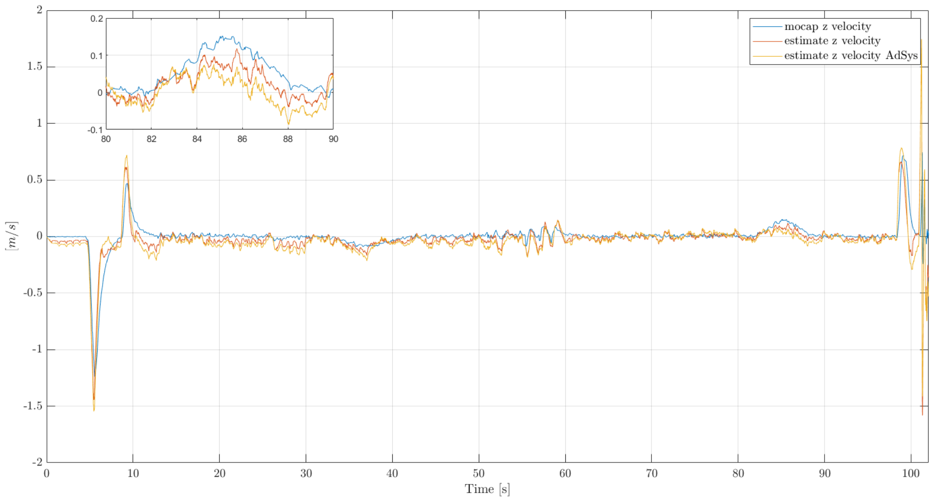


Figure 7.70: Velocity estimation along z-axis

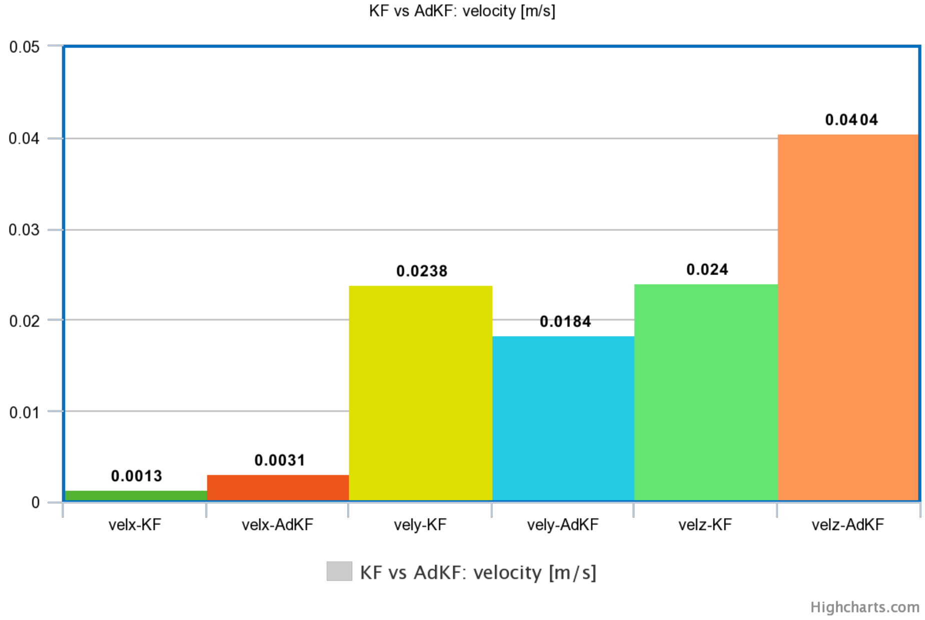


Figure 7.71: Comparison of discrepancies ”estimation-measurement” for velocity

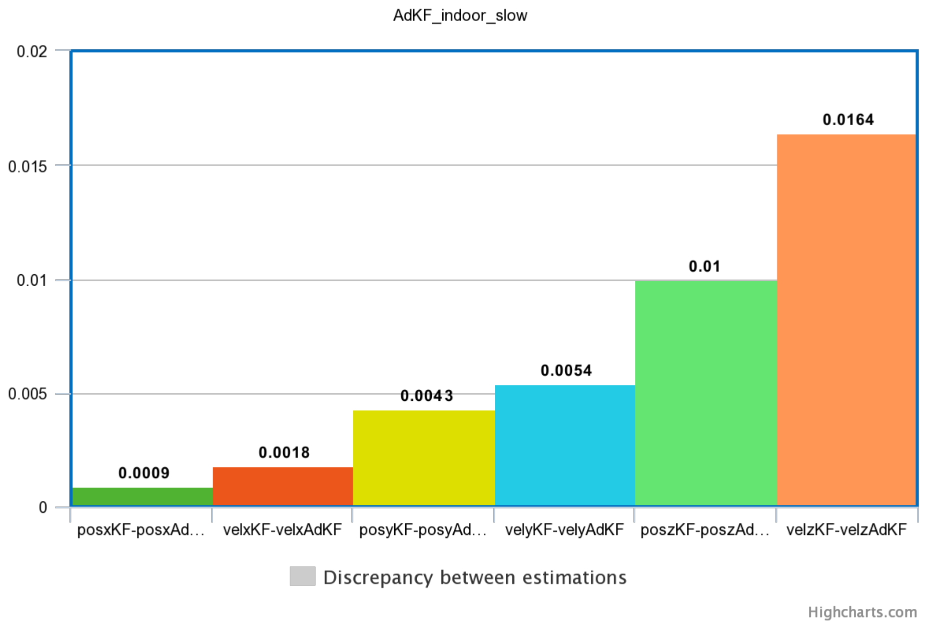


Figure 7.72: Discrepancy between estimations

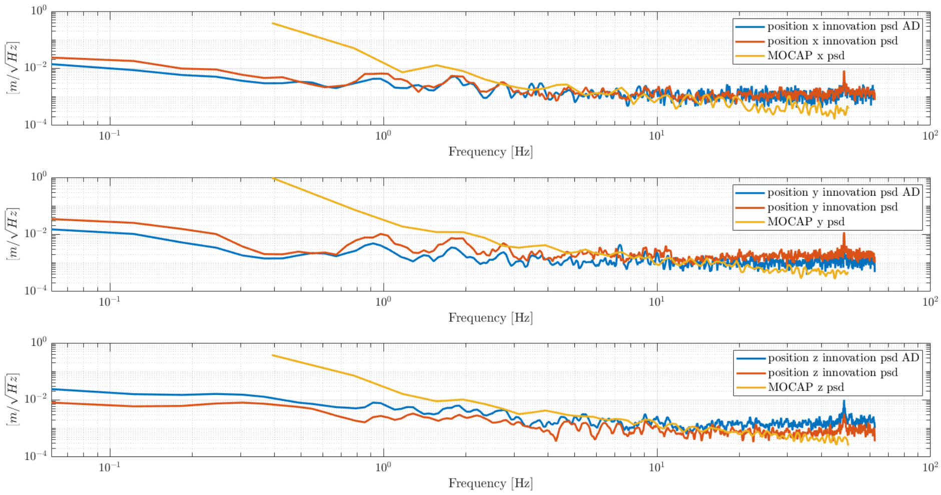


Figure 7.73: Innovation $PSDs$ for position

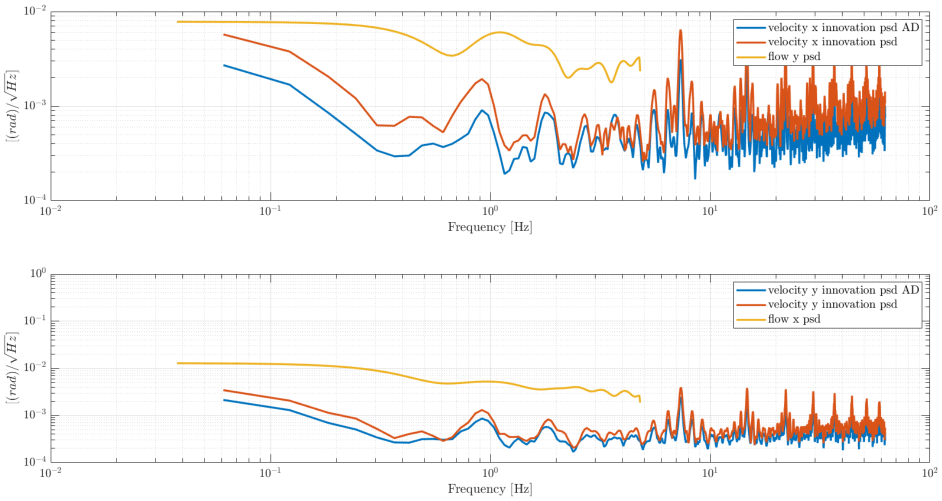


Figure 7.74: Innovation $PSDs$ for optical flow

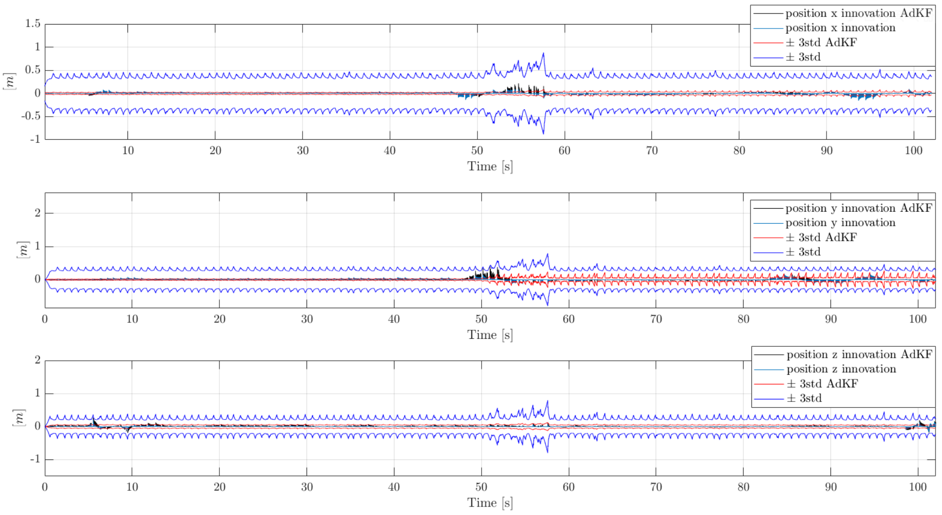


Figure 7.75: Position innovation

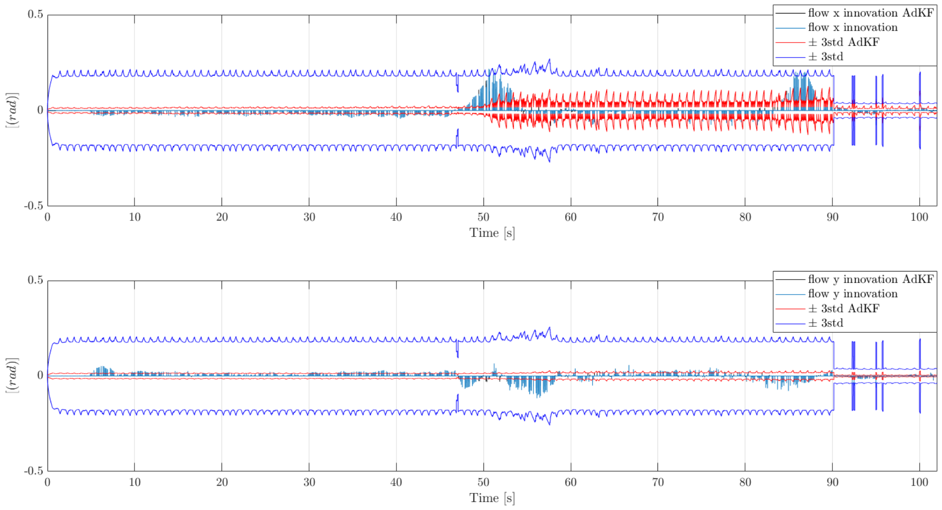


Figure 7.76: Optical flow innovation

the band of reference. Moreover the uncertainty band correspondent to AdKF is smaller than that of the KF.
Therefore considering that the innovation $PSDs$, of position and f, are practically flats, that their amplitude is less than that of the considered measurement and that the time history of the innovations is bounded in the reference band, then is possible to conclude that the AdKF is working well. And if also the KF is kept into account, then small differencies between the two approaches are noticeable, even if the AdKF presents more certain estimations.
In the following figures the Kalman gains deriving from the adaptive algorithm are plotted (Figures 7.77 and 7.78).

Keeping into account the results obtained above, seems that for indoor slow flights the two approaches guarantee results very similar but slightly better for the AdKF.

Fast flight

In this part the AdKF is applied to a fast flight for evaluating if it works correctely. The analysis starts observing the plots of the position estimations in Figures 7.79, 7.80 and 7.81 for p_x, p_y and p_z respectively. Considering the aforesaid figures, is evident that the estimations of the position are coherents if compared to the measurements of the MoCap. Observing the estimations along the three axes, an improvement using the AdKF is noticeable.
For the sake of getting an idea about the differencies between the estimates and the correspondent measurement, see Figure 7.82. Now is necessary to check if also for the velocities, the estimates provided by the filter are coherent; then in Figures 7.83, 7.84 and 7.85 the estimations of v_x, v_y and v_z are plotted.

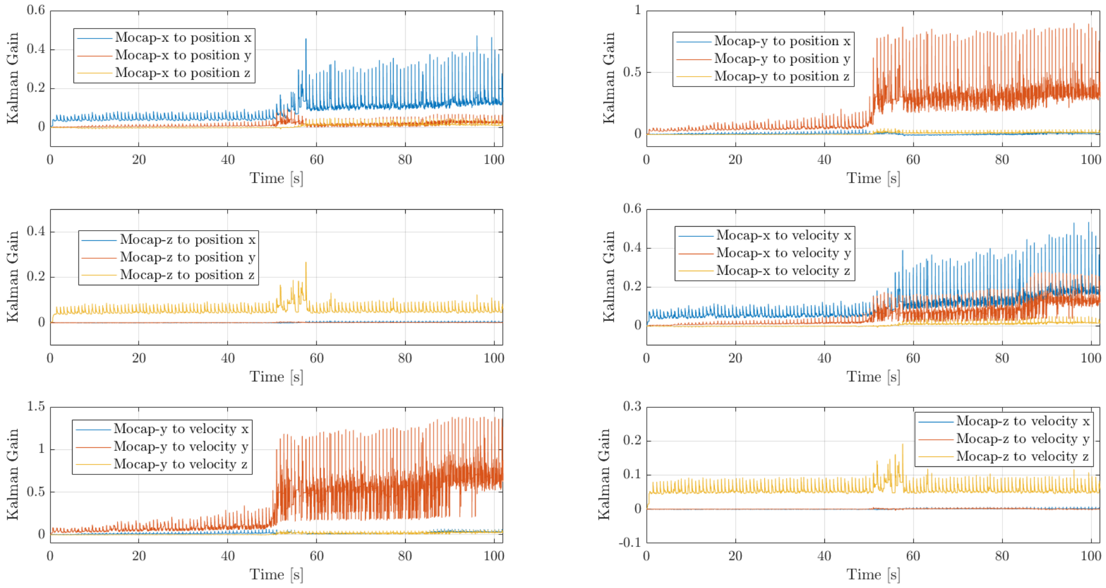


Figure 7.77: Kalman gain due to the position measurements

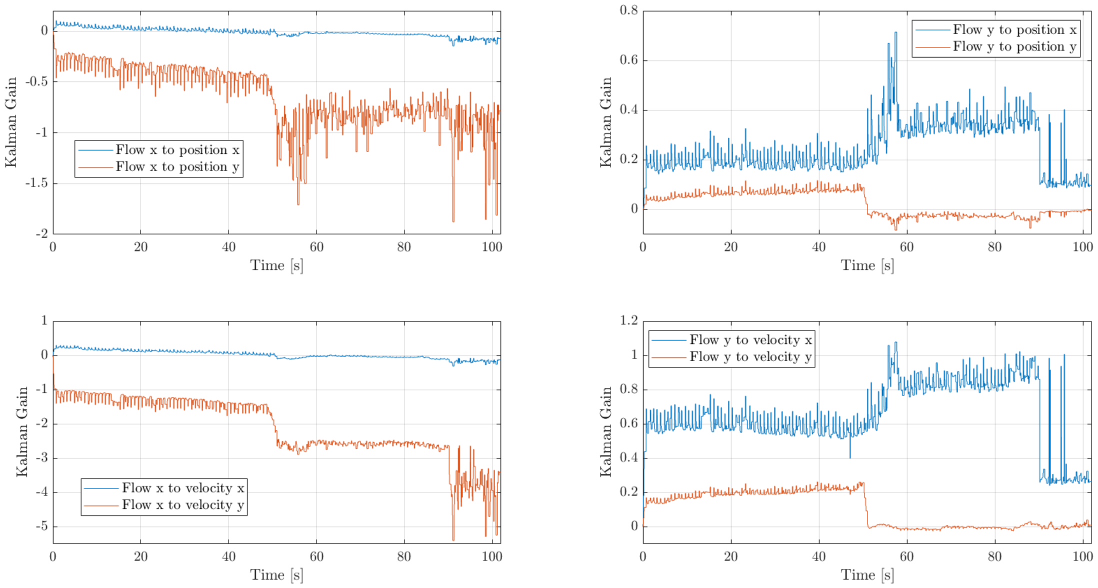


Figure 7.78: Kalman gain due to the optical flow measurements

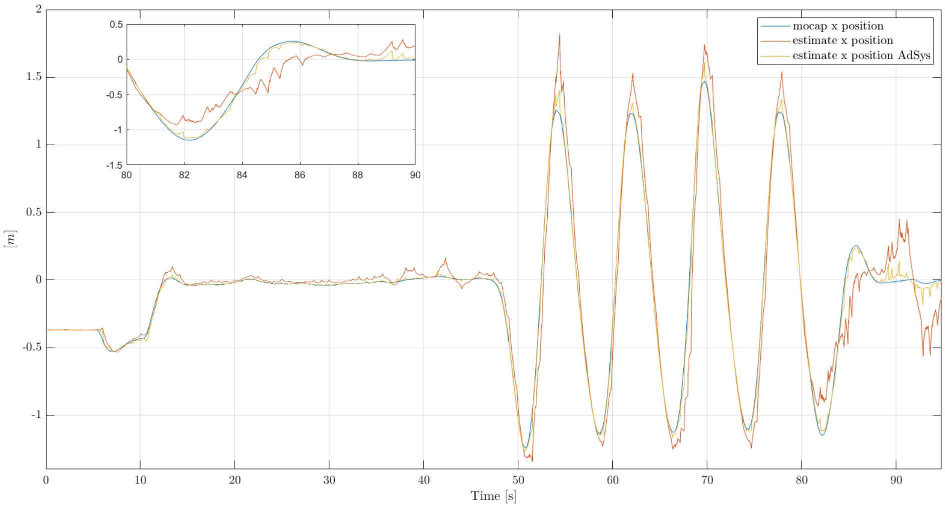


Figure 7.79: Position estimation along x-axis

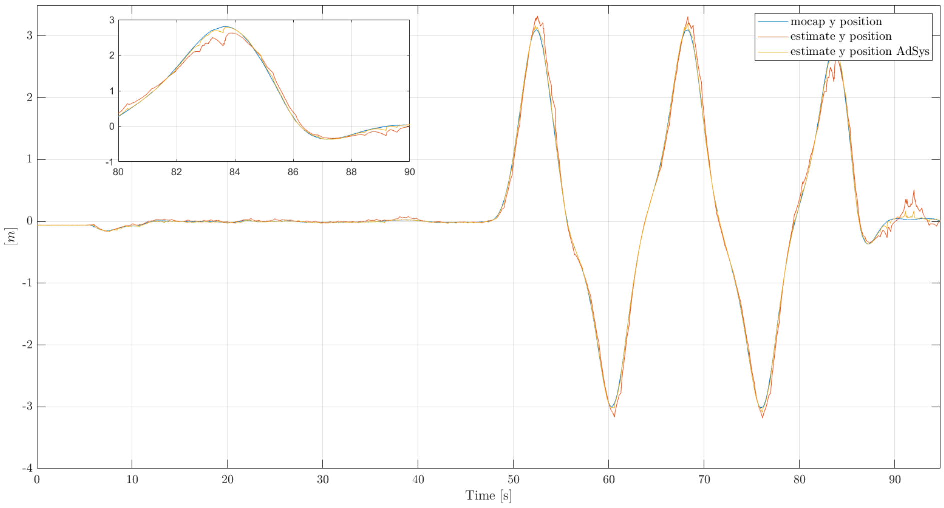


Figure 7.80: Position estimation along y-axis

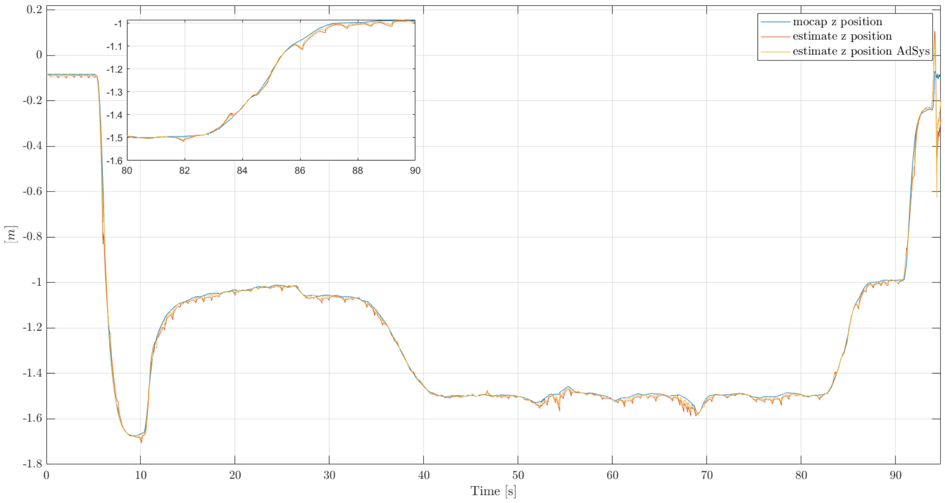


Figure 7.81: Position estimation along z-axis

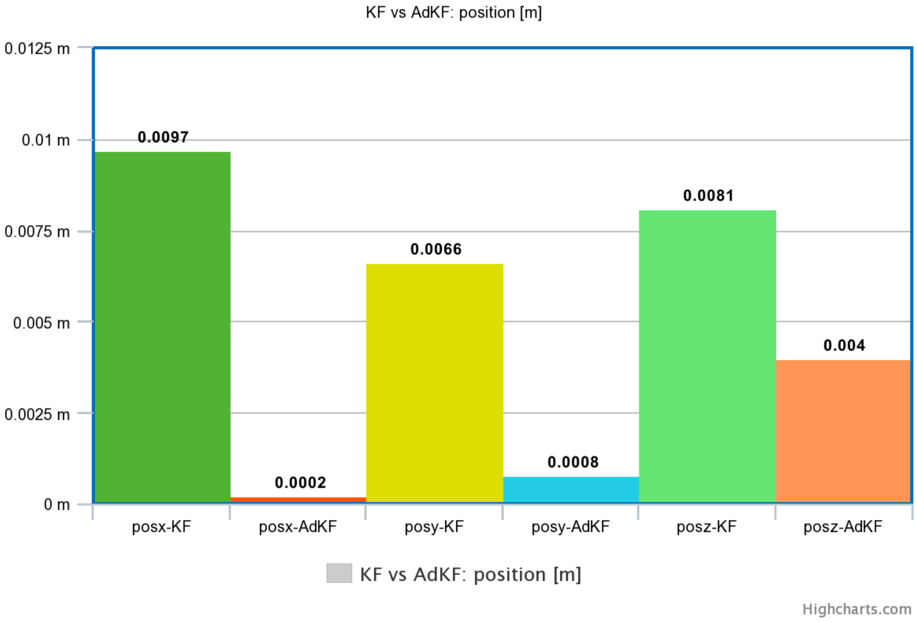


Figure 7.82: Comparison of discrepancies ”estimation-measurement” for position

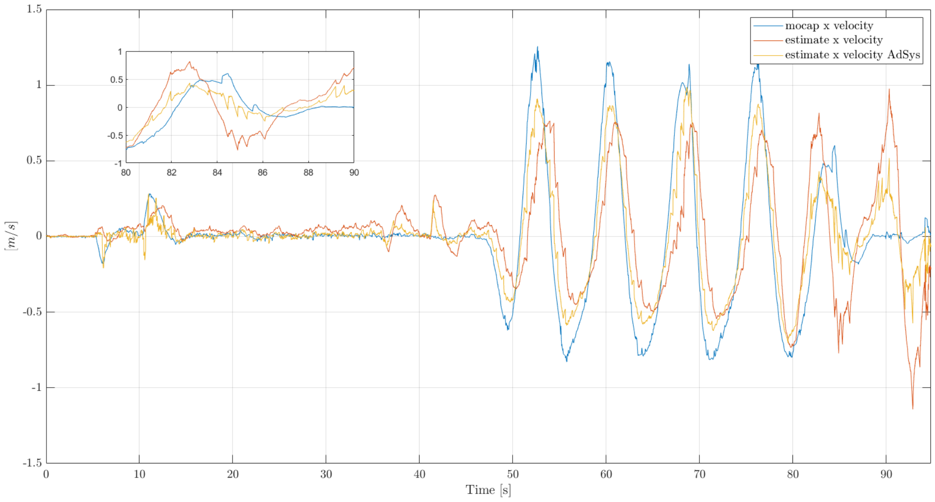


Figure 7.83: Velocity estimation along x-axis

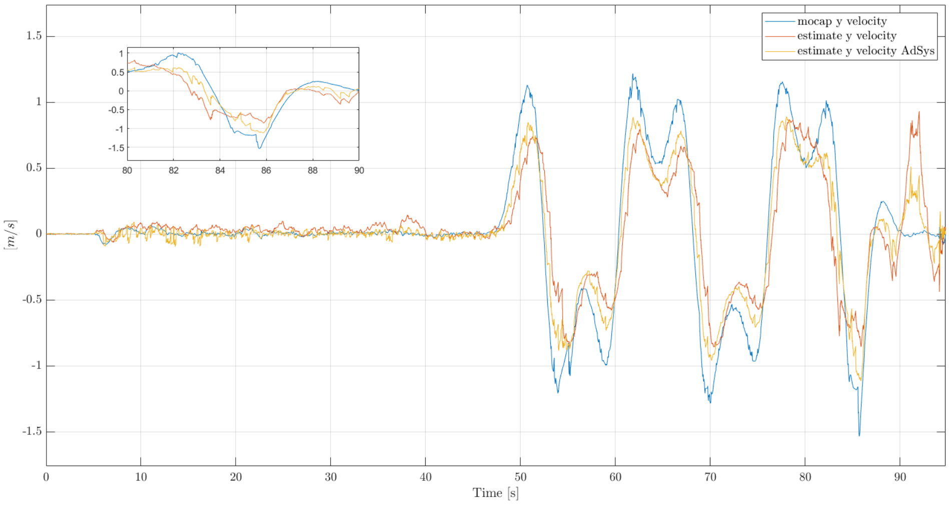


Figure 7.84: Velocity estimation along y-axis

Observing the Figures 7.83, 7.84 and 7.85, seems that using the AdKF respect to KF the estimations become slightly more coherent with the measurements. The discrepancies between each estimation and the correspondent measurement, are collected in the histogram in Figure 7.86, while the differences between the estimations are collected in 7.87. The adoption of the AdKF seems to return more satisfying results respect to the KF, but in order to be able to evaluate if the estimation process works correctely, the analysis of the innovations is necessary. The first analyzed figure is with the plots of the innovation $PSDs$ correspondent to the position (Figure 7.88). The innovation $PSDs$ for the position are flats, and those of the AdKF are smaller than those of the KF. Moreover the AdKF presents a trend of the $PSDs$ which is always smaller than the PSD of the considered measurement. Then the AdKF estimates adequately the UAV positions.

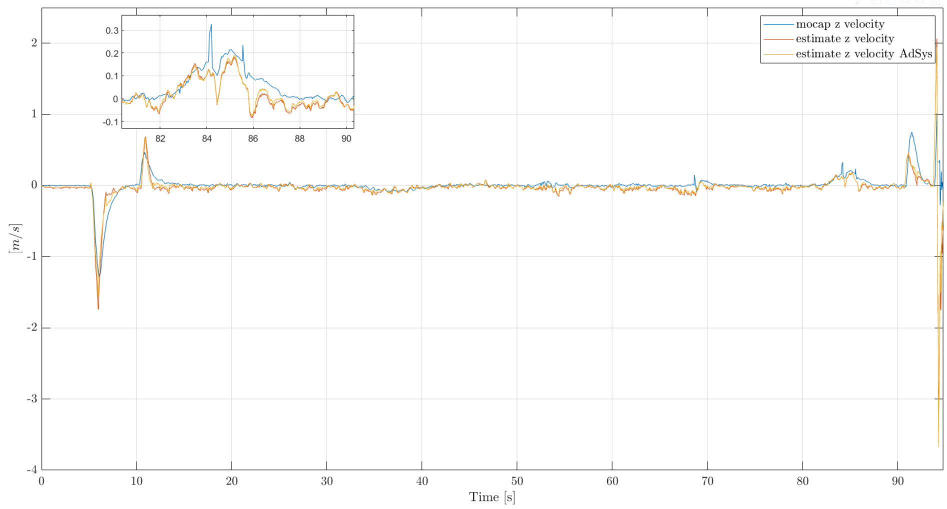


Figure 7.85: Velocity estimation along z-axis

About the f results see Figure 7.89. The innovation PSD correspondent to the AdKF, in Figure 7.89, reports a behaviour similar to that of the KF, but slightly improved. This guarantee that the estimates of the velocities, provided by the AdKF, are correct.

As said before, the other parameter used for evaluating the goodness of the filtering process, is the time evolution of the innovation. In Figures 7.90 and 7.91 is shown that in both the cases, the plus-minus band is very large; this means that the estimate contains a large uncertainty. Nevertheless the innovations stay bounded in the requested band, then the filter is working correctely.

In the end, to get an idea about the filtering process the plots of the Kalman gains are marked out (Figures 7.92 and 7.93).

Considering the results observed above, the AdKF provides good estimations of the UAV states, *i.e.* it works correctely. Respect to the estimations returned from

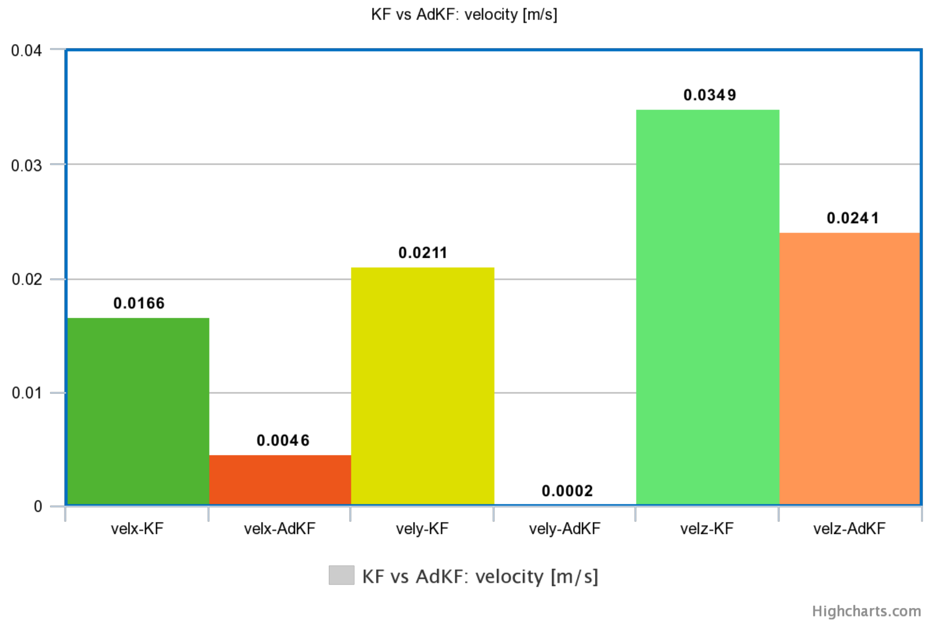


Figure 7.86: Comparison of discrepancies ”estimation-measurement” for velocity

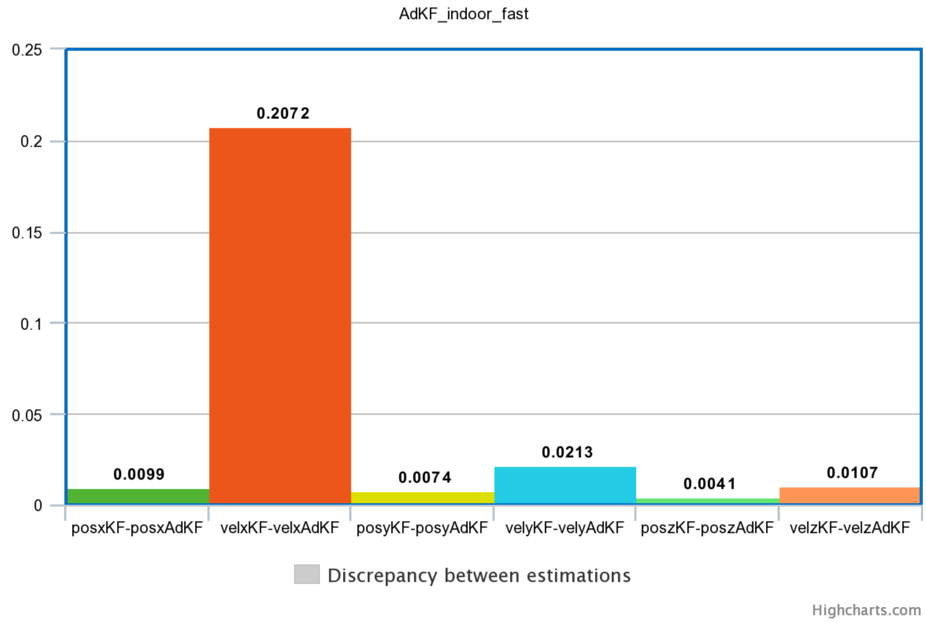


Figure 7.87: Discrepancy between estimations

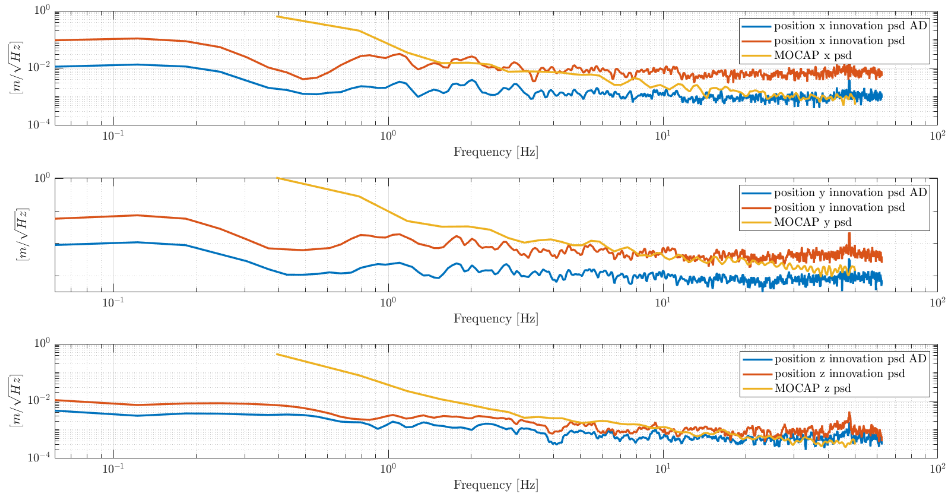


Figure 7.88: Innovation $PSDs$ for position

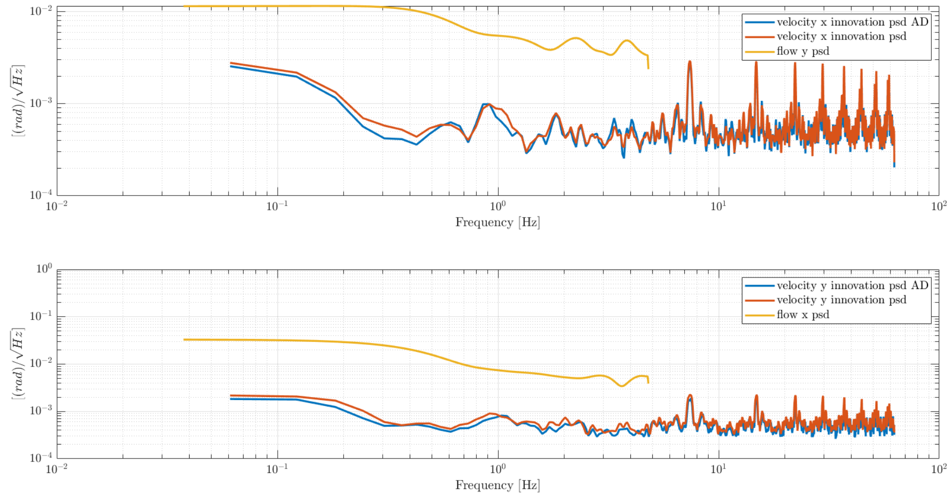


Figure 7.89: Innovation $PSDs$ for optical flow

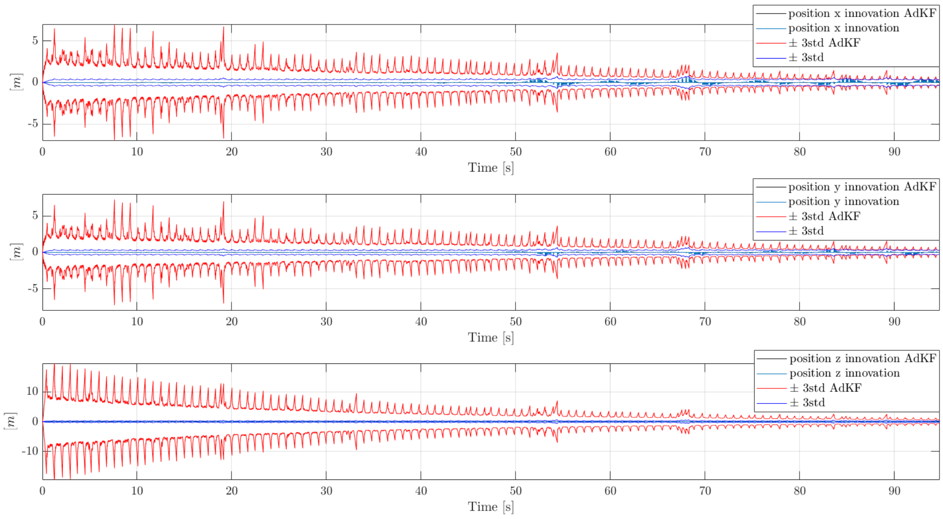


Figure 7.90: Position innovation

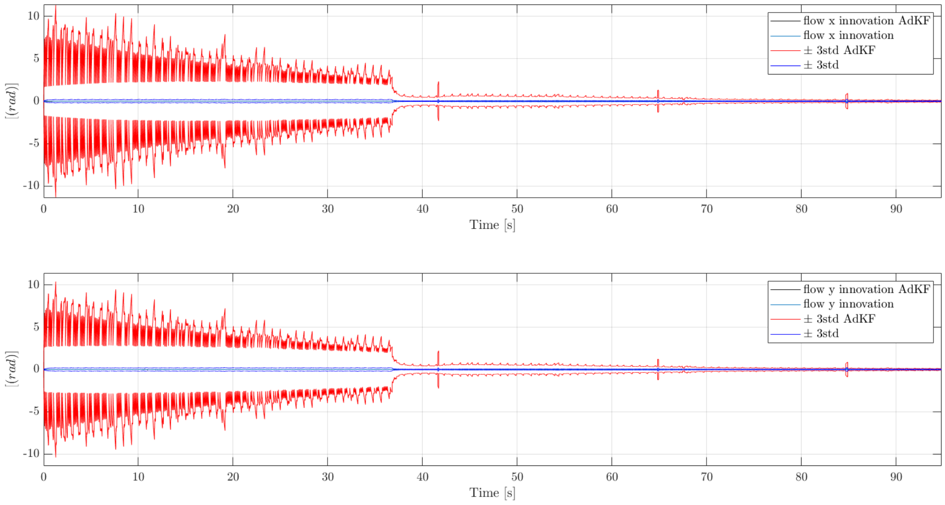


Figure 7.91: Optical flow innovation

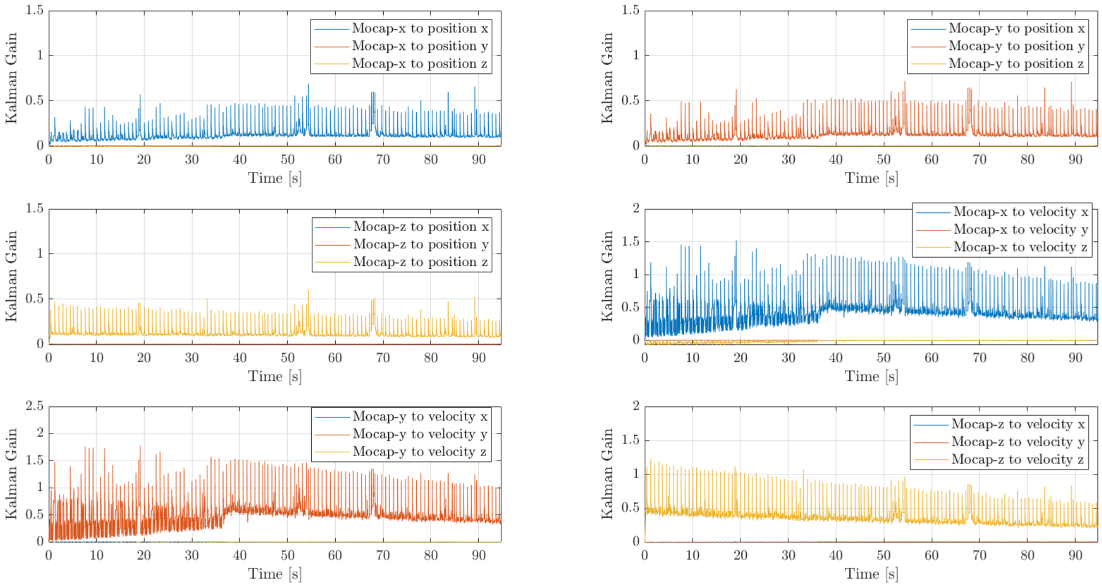


Figure 7.92: Kalman gain due to the position measurements

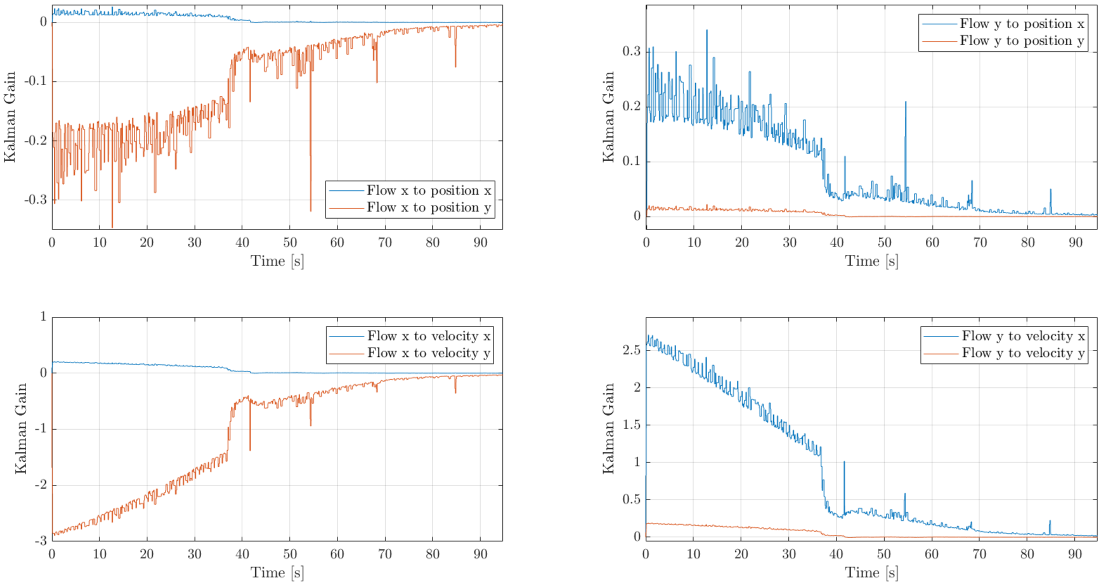


Figure 7.93: Kalman gain due to the optical flow measurements

the KF algorithm, those from the AdKF present an improvement in terms of the filtering process, since the *PSDs* are smaller than those of the KF; but the state estimations presents larger uncertainty.

7.3 Estimate results: augmented vs adaptive Kalman filter

In this part of the chapter the purpose is to gather the most important conclusions, seen before, and then to make a comparison between the two versions of the Kalman filter, the augmented and the adaptive one. For this purpose not all the UAV states will be used but only those which can better underline the results.
Since z is the axis along which the estimation is the most critical one, then the comparison between the filters will focus on that axis. About the axes x and y, as seen in the previous sections, the estimations by the AugKF and the AdKF are better than those by the KF, but actually the KF for those axes, return sufficient good results. Then the purpose is to find the filter which improve the estimations along z. Starting from the outdoor flight, let see the plots of the discrepancy between each estimation (KF, AugKF and AdKF) and the considered measurement, for position and velocity along z. The results for the position are in the plot in Figure 7.94. As it is possible to see the AugKF and the AdKF present an improvement respect to the use of the KF. In particular the estimation made by the AdKF presents a more constant discrepancy along time, while the one made by the AugKF presents oscillations. From the plot is deductible that the

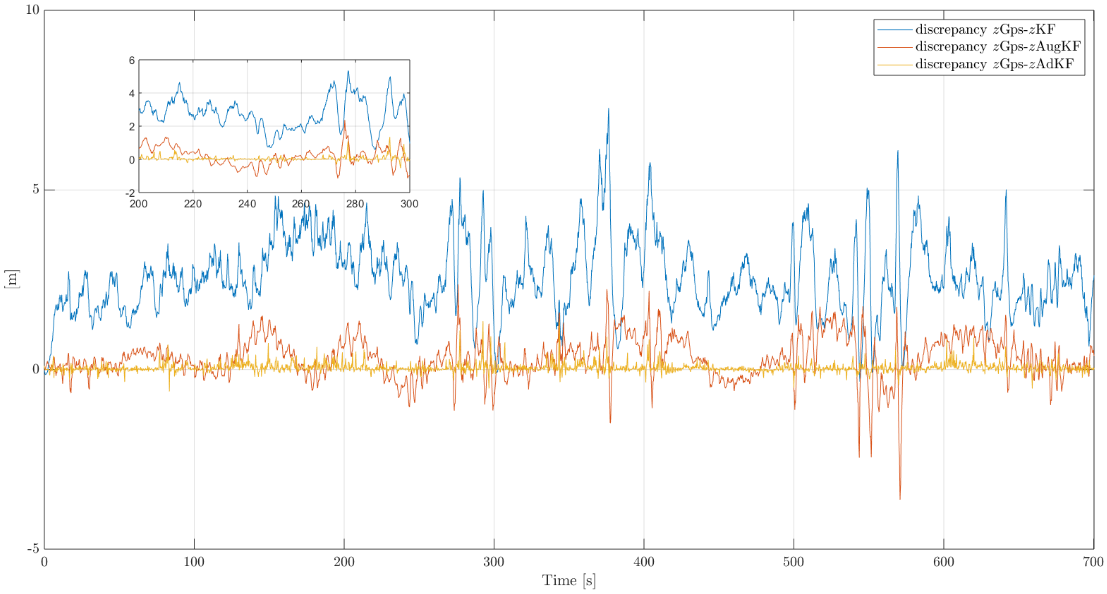


Figure 7.94: Comparison: KF vs AugKF vs AdKF for position

estimation provided by the AdKF is better along z. This does not mean that the augmented filter does not provide a good estimation. The same analysis is made for the velocity along the same axis, and it is in Figure 7.95. The conclusions for the velocity are exactly the same of the position; namely the estimation given by the AugKF presents oscillations which the estimation provided by the adaptive filter does not present. Furthermore both these filter versions return a better chasing of the considered measurement, respect to the KF estimation.

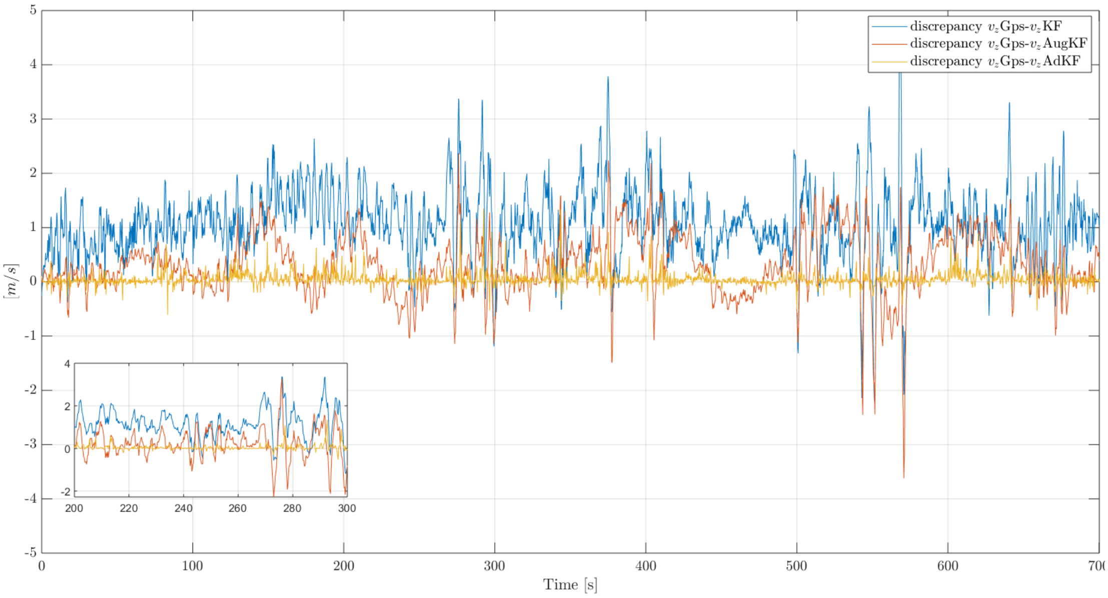


Figure 7.95: Comparison: KF vs AugKF vs AdKF for velocity

Moreover the innovation $PSDs$ correspondent to the position (Figure 7.96) are plotted, since they are a summary of the performances of the different filtering processes. In Figure 7.96 all the filters give a PSD which is almost flat and with a smaller amplitude than the PSD of the measurement considered, therefore each version could be used for the estimation of the states of interest; but as said before what matters is the axis z, then the choice will depend on this. For the axes x and y the AugKF gives estimations with smaller variances respect to the AdKF. On the contrary the estimation along the axis z provided by the AdKF presents a smaller PSD than the AugKF. About the innovation PSD correspondent to the f, in Figure 7.97, is observable a better behaviour of the AdKF for both the channels of the f.

Therefore keeping into account the results above, the AdKF is a better choice respect to the AugKF, when the UAV is let fly in outdoor enviroment.

Now the focus is to find the best filter, when the UAV is flying in an indoor enviroment, and subjected to trajectories with slow dynamics. As for the outdoor case, the choice of the filter must improve the estimate along z. The first plot

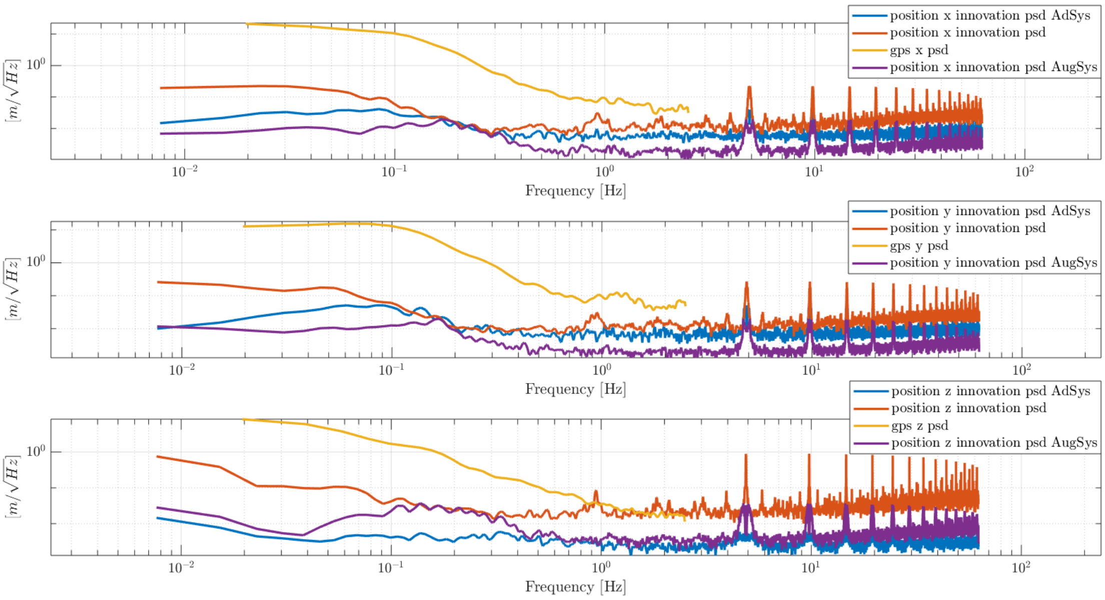


Figure 7.96: Comparison: KF vs AugKF vs AdKF innovation $PSDs$ for position

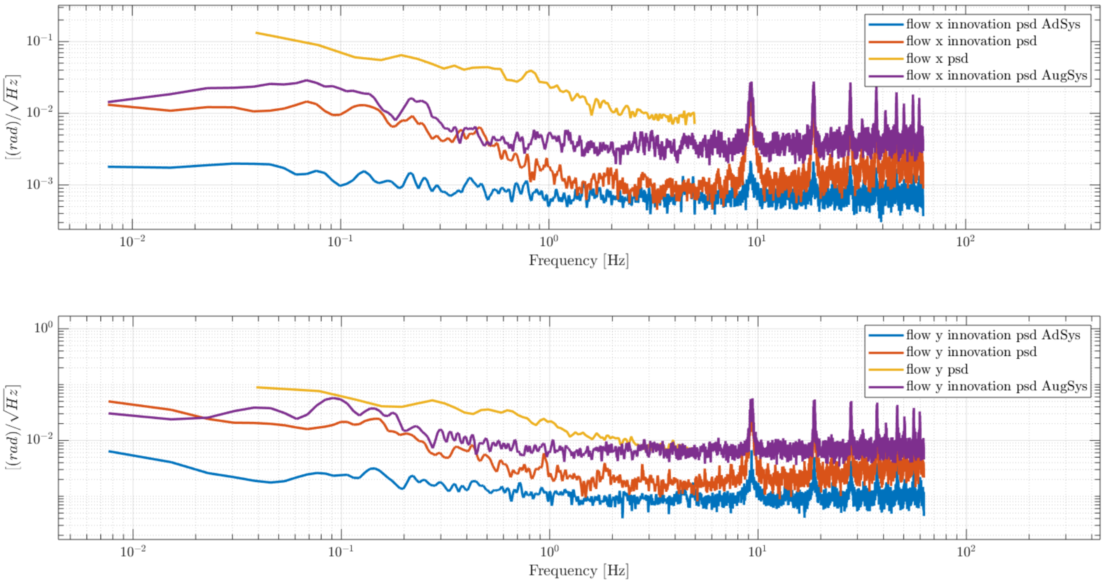


Figure 7.97: Comparison: KF vs AugKF vs AdKF innovation $PSDs$ for optical flow

observed is in Figure 7.98, in which the discrepancy between the estimates and the considered measurement is depicted. As seen in the outdoor test, the estimation by means of the AugKF presents some oscillations, while that given by the AdKF has a very small amplitude. Actually all the three estimates are not bad, but relying on the plot in 7.98, the AdKF seems to give the best performances. But as said before only the PSD of the innovation, can draw a conclusion like this. Returning to the discrepancies plot, in Figure 7.99 is shown what happen to the velocities. For the velocity, in 7.99, is observable a good behaviour of all the estimators; all the estimations present a small discrepancy.

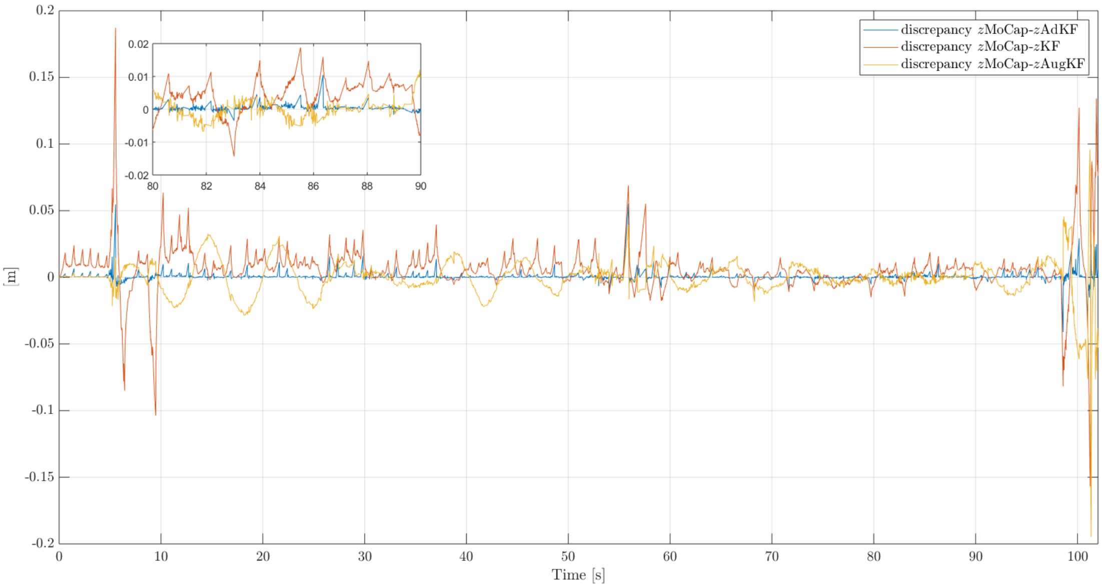


Figure 7.98: Comparison: KF vs AugKF vs AdKF for position

For the sake of being able to report a techincal conclusion, is necessary to observe the plots marked out in Figures 7.100 and 7.101. In 7.101 is evident that the adaptive and the augmented algorithms provide similar innovation $PSDs$, even if along z the AugKF gives smaller amplitude, then less uncertainty on the estimation value.

About the innovation $PSDs$ correspondent to the flow, is observable an almost similar trend of all the filters.

Then, considering the results just seen, the version of the Kalman filter which can guarantee a good estimation of position and velocity, also along the axis z, during a slow flight indoord, is the AugKF.

Finally an estimator for the fast indoor flight is chosen. Alike the two previous cases, the discriminants for making the right choice, are the estimations along the z-axis in NED. First of all the discrepancies are shown, to compare how much the estimations are coherent with the actual flight of the UAV; then the conclusion

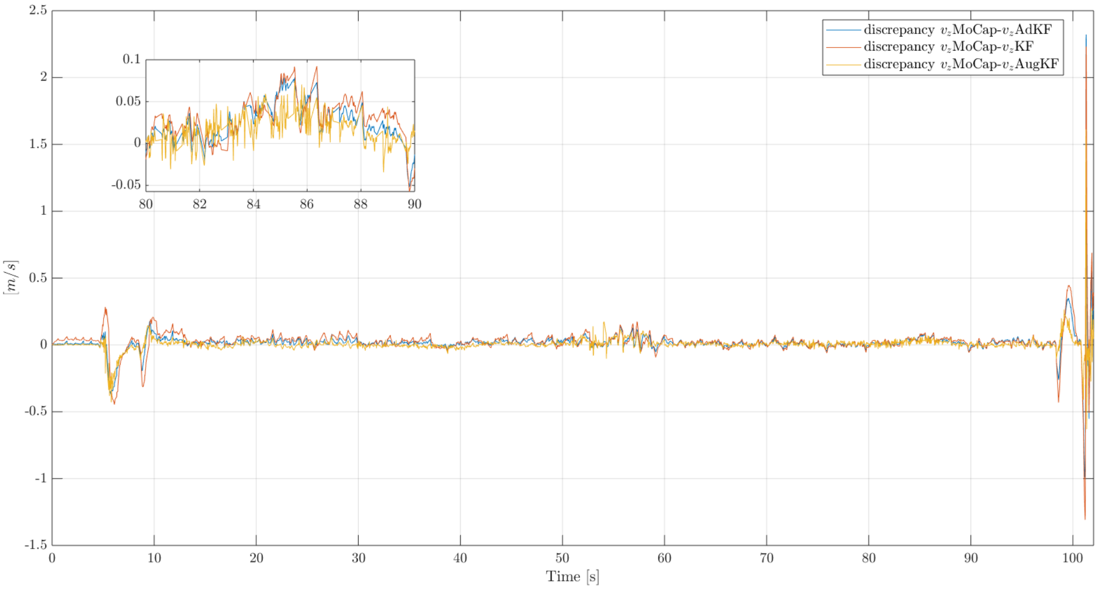


Figure 7.99: Comparison: KF vs AugKF vs AdKF for velocity

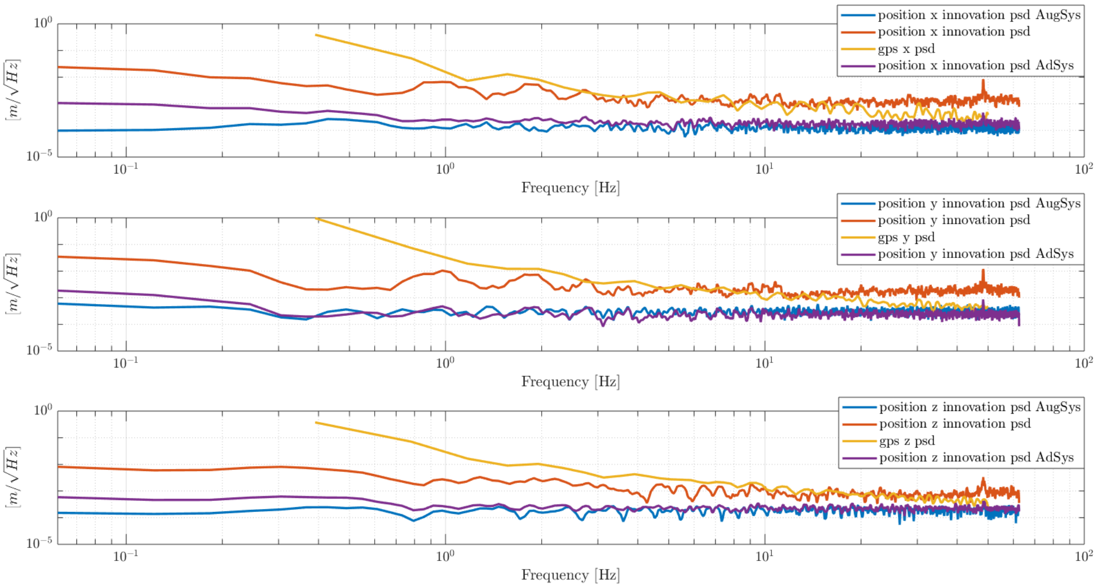


Figure 7.100: Comparison: KF vs AugKF vs AdKF innovation $PSDs$ for position

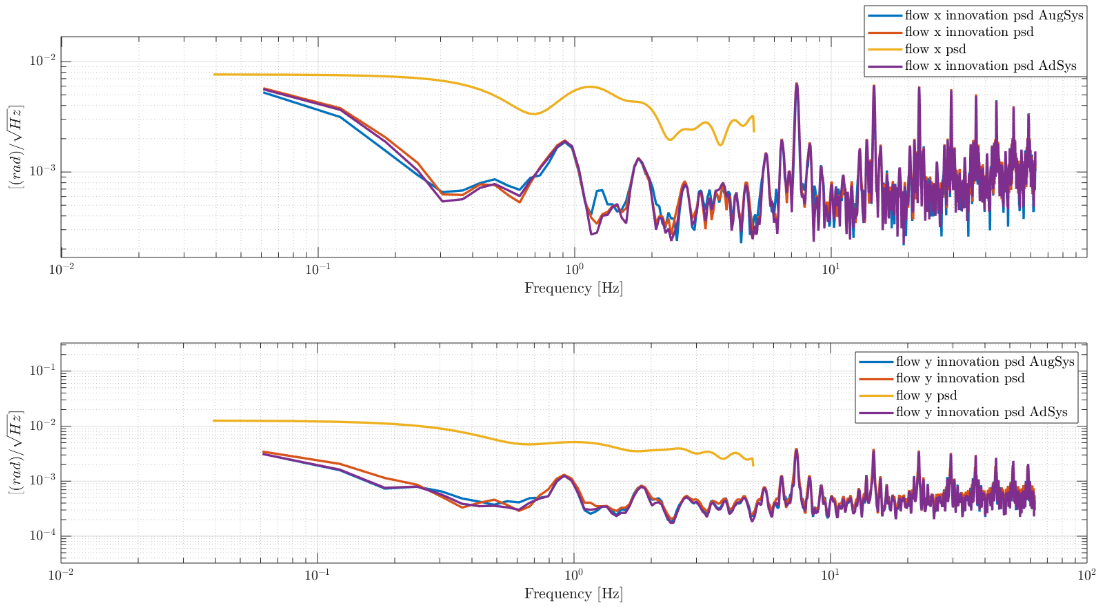


Figure 7.101: Comparison: KF vs AugKF vs AdKF innovation $PSDs$ for optical flow

will be based on the analysis of the innovation $PSDs$ of the states of interest. The first encountered plot is in 7.102, in which the differencies between the estimations and the measurement considered, are plotted. The AugKF presents the usual periodic behaviour shown before, while the AdKF reports a very small discrepancy. In general the three types of filter, give coherent trends. Similarly for the velocity in Figure 7.103. By observing the Figures 7.104 and 7.105 the same conclusions, for the same reasons of the slow flight, are drawn; that means the AugKF is the best choice in the aforesaid condition.

7.4 Conclusions

The purpose of this chapter has been to evaluate if the Kalman filter based on the algorithms shown in the previous chapters (KF, AugKF and AdFK), is or less a good estimator. To answer at this question, first of all each estimation has been compared to the available measurement, to understand if the filter is working coherentely with the actual movement of the UAV. The following passage has been to evaluate from a techincal point of view, if the filtering process is correct; to veify this, the innovation behaviour, has been analyzed, both in time domain (its time history) and in frequency domain (its PSD). On the basis of these two tests the correctness of the filter has been checked. The first good result obtained is that the three versions of the Kalman filter, have been able to filter correctely the states of interest; the second result obtained was a more detailed one: on the

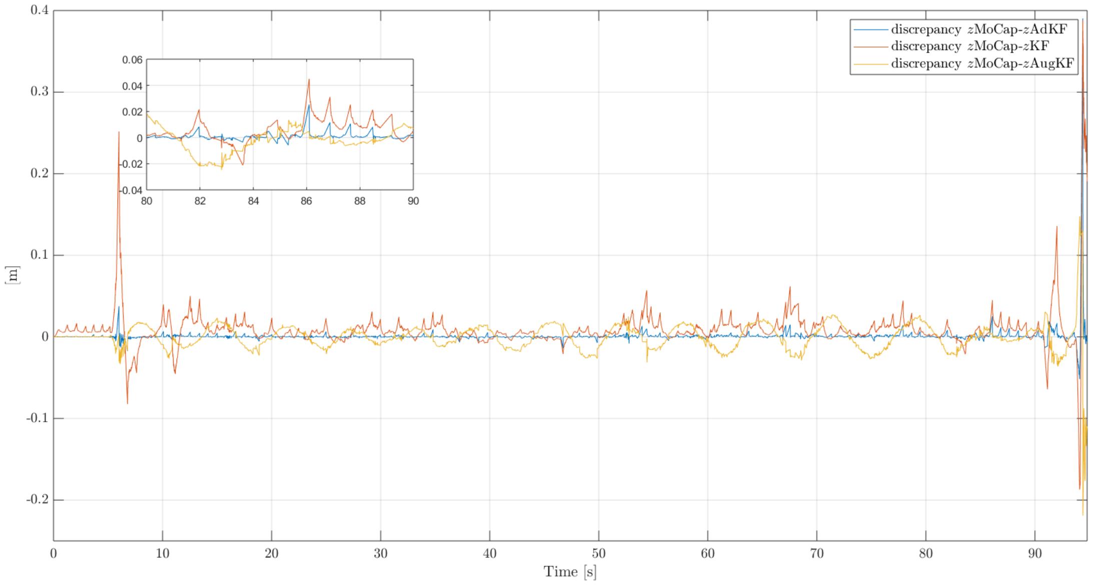


Figure 7.102: Comparison: KF vs AugKF vs AdKF for position

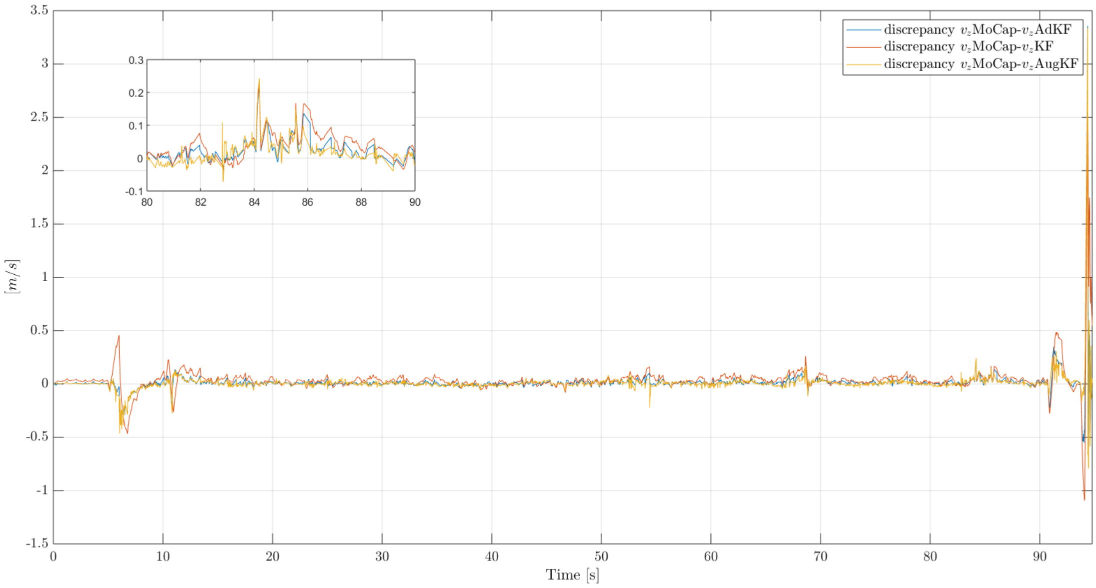


Figure 7.103: Comparison: KF vs AugKF vs AdKF for velocity

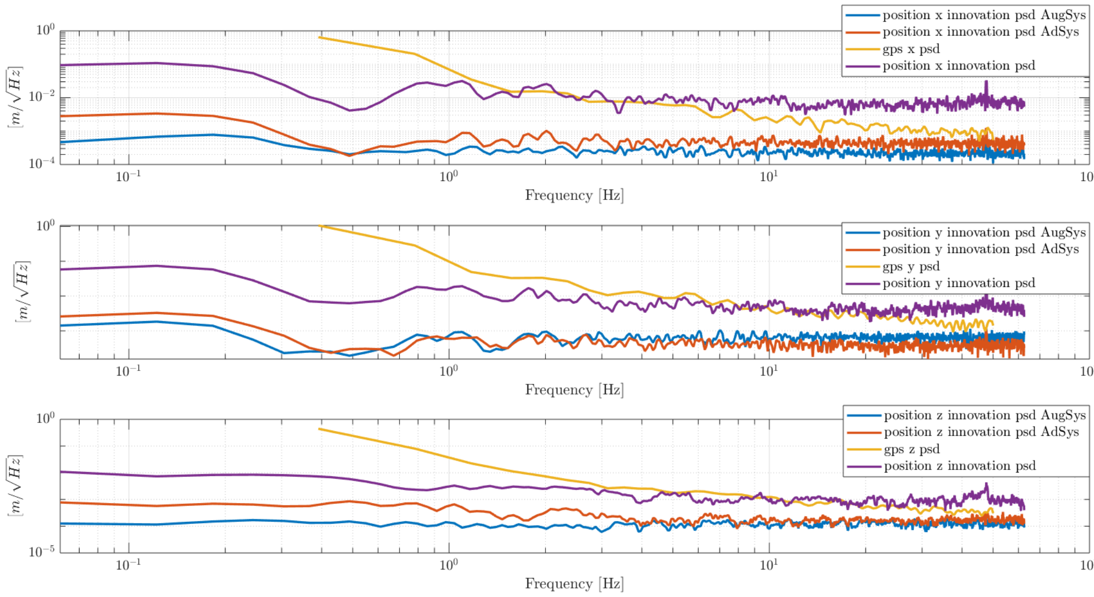


Figure 7.104: Comparison: KF vs AugKF vs AdKF innovation $PSDs$ for position

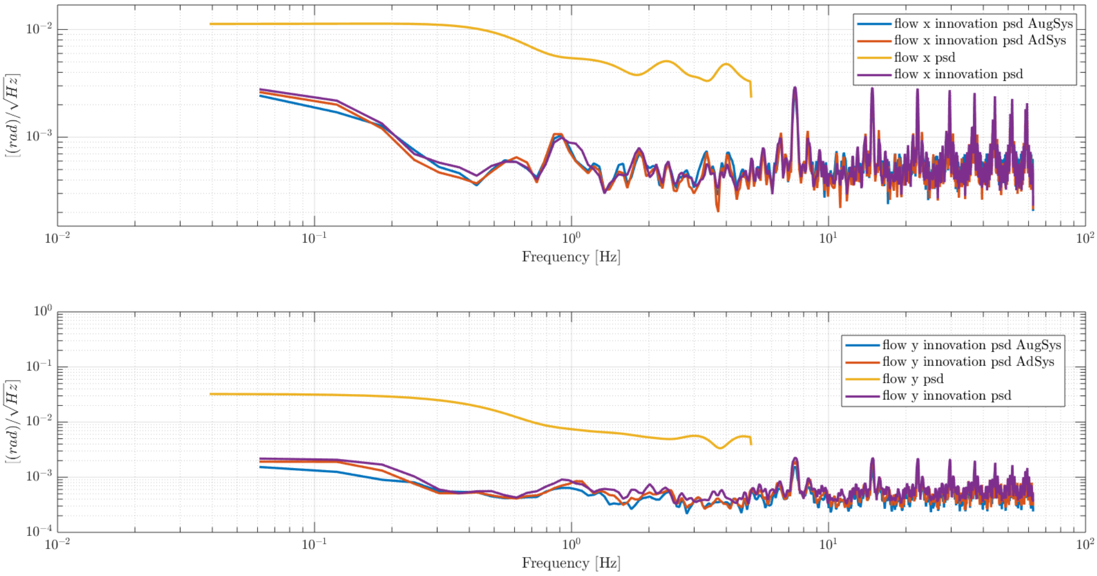


Figure 7.105: Comparison: KF vs AugKF vs AdKF innovation $PSDs$ for optical flow

basis of the flight condition, a filter rather than another has been chosen for the filtering process. In particular for the outdoor flight has been obtained that the UAV states, are better estimated by means of the adaptive Kalman filter; while for the indoor test, for both the slow and fast dynamics the best algorithm to exploit is the augmented one.

Conclusions

The aim of this thesis work is to construct the estimates of the UAV position and velocity, by means of a Kalman filter based on GPS and optical sensor measurements, to improve the pre-existing UAV navigation system. Actually three Kalman filters have been adopted with the purpose to model the noises provided by the sensors in three different ways.
The estimation process has been made by integrating a new optical flow sensor, called Px4Flow, on-board the UAV, which outputs have been identified by flight testing activity. Following the work previously made by the author Musacchio, the relation between the optical flow sensor output and the UAV velocity in the NED reference frame, has been sought, 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 implementation of the Kalman filter versions, has been made following three different pathes, based on the assumptions made to model the variances of the noises provided by the sensors. The assumptions are the following:

- noises modeled as white noises;
- noises modeled using shape filters to keep into account their dynamics;
- noises modeled using an adaptive approach.

It is rightful to explicit the fact that the estimation process is performed once the flight data are collected, in other words the construction of the states is not made in real time, but post-flight.
The flights have been organized in outdoor and indoor enviroment, and in fast and slow flights. This latter case has been executed to keep into account that the Kalman filter performes differently in dependence on the flight dynamics.
It has been observed that all the three approaches return reliable estimations results, so each case presents acceptable behaviour of the states time history. The correctness of the results has been evaluated analyzing the $PSDs$ of the innovation and the time history of the innovation. More precisely for the outdoor flight has been obtained that the UAV states, are better estimated by means of the adaptive Kalman filter; while for the indoor test, for both the slow and fast dynamics the

best algorithm to exploit is the augmented one. This does not mean that the others algorithms do not work, but that the best results are obtained by those particular filters.
In the end some considerations are made for future developments:

- for the augmented Kalman filter consider the correlation between the noise of the state equation and that of the output equation;
- perform more flights, for each previously seen case, to collect more data for a better awereness of the different filter versions performances;
- in the Kalman algorithm keep into account the problem of the jitter.

Bibliography

[1] Simone Musacchio. Optimal and robust UAV state estimation based on Gps and optical flow. 2017-2018.

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

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

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

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

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

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

[8] 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.

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