AUTO TAKEOFF AND PRECISION LANDING USING INTEGRATED GPS/INS/OPTICAL FLOW SOLUTION

by
Mohammad Khaled Salameh Al-Sharman

A Thesis Presented to the Faculty of the
American University of Sharjah
College of Engineering
in Partial Fulfilment
of the Requirements
for the Degree of

Master of Science in
Mechatronics Engineering

Sharjah, United Arab Emirates
January 2015

Approval Signatures

We, the undersigned, approve the Master's Thesis of Mohammad Khaled Al-Sharman.

Thesis Title: Auto Takeoff and Precision Landing using Integrated GPS/INS/Optical Flow Solution.

Signature **Date of Signature**

___________________________ _______________
Dr. Mohammad Amin Al-Jarrah
Professor, Department of Mechanical Engineering
Thesis Advisor

___________________________ _______________
Dr. Mamoun Abdelhafez
Associate Professor, Department of Mechanical Engineering
Thesis Co-Advisor

___________________________ _______________
Dr. Mohammad Nazzal
Visiting Associate Professor, Department of Mechanical Engineering
Thesis Committee Member

___________________________ _______________
Dr. Shayok Mukhopadhyay
Assistant Professor, Department of Electrical Engineering
Thesis Committee Member

___________________________ _______________
Dr. Mamoun Abdelhafez
Director, Mechatronics Graduate Program

___________________________ _______________
Dr. Mohamed El-Tarhuni
Associate Dean, College of Engineering

___________________________ _______________
Dr. Leland Blank
Dean, College of Engineering

___________________________ _______________
Dr. Khaled Assaleh
Director of Graduate Studies

To Mum and Dad

*“I would thank you from the bottom of my heart,
but for you my heart has no bottom”*

Abstract

Auto takeoff and landing has been considered as the most challenging part in performing a flight with a high degree of autonomy. Hence, many researchers have addressed the problem of developing a precise auto takeoff and landing system for unmanned miniature helicopters. The work on enhancing autonomous takeoff and landing for unmanned aerial vehicles can be categorized into two groups. The first group works on designing a robust control algorithm in which the controller performs auto takeoff and auto landing. The second group focuses on utilizing sensors with high accuracy to get accurate state measurements. As a result, the performance of both the estimator and the controller would be improved. The present research addresses the use of optical flow sensors to augment the Global Positioning System/ Inertial measurement unit (GPS/INS) solution in the terminal phases of the flight (i.e., takeoff and landing). The GPS/INS unit has an internal estimator to estimate the vehicle's state vector which is not accurate enough to perform precision landing. The GPS/INS estimated position using Commercial-Off-The-Shelf COTS components is inaccurate with a few meters error, which is called the radius of uncertainty (ROU). To perform a precise landing, an optical flow sensor is used to augment the GPS/INS readout while performing the takeoff and landing phases of the flight. In this research, we use a sensor fusion algorithm between the optical flow sensor measurements of the location of a predefined pattern within the ROU of the GPS/INS and the GPS/INS location measurements. This estimator is used to output the helicopter's position and velocity during takeoff and landing. The proposed estimator has succeeded in performing an auto takeoff with a maximum error of *0.26 m* and a precise landing with a maximum error of 0.27 m in Z-position. The novelty in this study is in the use of GPS/INS/Optical Flow fusion algorithm to perform a precise auto takeoff and landing for a small-scale helicopter. In addition, an accurate model for the OF sensor is used for developing the control laws for autonomous takeoff and landing for Vertical takeoff and landing vehicles (VTOL).

Search Terms: *Autonomous landing, auto-takeoff, rotary wing UAV, helicopter, optical flow sensor, state estimation, and sensor fusion.*

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SAD	:	Sum of Absolute Differences	
BMA	:	Block Matching Approach	
MAV	:	Micro Aerial Vehicle	
FSBMA	:	Full Search Block Matching Approach	
a	:	Main rotor lift curve slope	5.5 [rad^{-1}]
a_{tr}	:	Tail rotor lift curve slope	5.5 [rad^{-1}]
C	:	Rotor blade chord	[m]
UAV	:	Unmanned Aerial Vehicle	
GPS	:	Global Positioning System	
INS	:	Inertial Navigation System	
ROU	:	Radius of uncertainty	
OF	:	Optical Flow	
VTOL	:	Vertical Takeoff and Landing	
CCPM	:	Cyclic Collective Pitch Mixing	
CG	:	Center of Gravity	
COTS	:	Commercial-Off-The-Shelf	
C_T	:	Coefficient of thrust	[•]
fx, fy, fz	:	Helicopter forces in the X, Y and Z directions, respectively	[N]
G	:	Gravitational acceleration	9.81 [m/s^2]
H_{tr}	:	Vertical length from tail rotor hub to helicopter CG	0.0779 [m]
$I_{xx}, I_{yy},$	:	Helicopter Moments of Inertia, respectively	[0.1 0.4 0.3]

I_{zz}			[kg.m^2]
L, M, N	:	Helicopter Roll, Pitch, and Yaw moments , respectively	[N m]
L_b, M_a	:	Roll Moment and Pitch Moment Derivative , respectively	[0 0] [Nm/rad]
L_{tr}	:	Horizontal length from helicopter CG to tail rotor hub	1.068 [m]
M	:	Mass of helicopter	7.3 [kg]
N	:	Number of blades	2 blades
p, q, r	:	Helicopter body roll, pitch, and yaw rates, respectively	[rad/sec]
R_{mr}	:	Main Rotor radius	0.89[m]
R_{tr}	:	Tail Rotor radius	0.16 [m]
S	:	Rotor Solidity Nc/(πR)	
T_{mr}	:	Main rotor thrust	[N]
u, v, w	:	Helicopter airspeed in the x, y, and z directions, respectively	[m/s]
u_{col}	:	Collective lever input	[rad]
u_{lat}	:	Lateral stick input	[rad]
u_{lon}	:	Longitudinal stick input	[rad]
u_{ped}	:	Rudder pedal input	[rad]
u_{tr}, v_{tr}, w_{tr}	:	Tail rotor hub airspeed in the x, y, and z directions, respectively	[m/s]
α	:	Angle of attack.	[rad]
β_{1c} , β_{1s}	:	Longitudinal and Lateral Main Rotor Flap Angles, respectively	[rad]
δ_{ped}	:	Collective pitch-angle of tail rotor blade	[rad]

θ_o	:	Collective pitch-angle of main rotor blade	[rad]
λ_{mr}	:	Inflow ratio for main rotor	[•]
μ	:	Advance ratio.	[•]
μ_z	:	Advance ratio along the z axis	[•]

Chapter 1: Introduction

1.1 Background

In the last decade, unmanned aerial vehicles (UAV) have proved to be ideal platforms for various practical purposes such as surveillance, aerial photography, and other military and civil applications. Unmanned vertical takeoff and landing (VTOL) vehicles are finding important applications in civil as well as for homeland security in addition to their ever-growing military applications. Therefore, auto takeoff and precision landing is important to these vehicle's successful use for civil applications. Autonomous takeoff and landing has been recognized as the main focus of attention during recent years. In this thesis, an autonomous takeoff and precision landing system is developed using optical flow sensor information.

1.2 Literature Review

Research in the literature primarily concentrates on the landing issue, which has been extensively studied because it is the most difficult phase of achieving an autonomous flight. The landing technique has been done on moving platforms [1] [2] [3] and on a fixed target [4] [5] [6]. Recently, the vision sensors have been brought to the field of performing autonomous landing missions. The present visual landing systems can be classified into two groups: the first group uses a vision in order to detect a suitable place for the landing [4] while the second group is for landing on a predefined target [5] [6] [7].

A vision-based landing algorithm for an autonomous helicopter was made in [5], in which the location and orientation of the landing target are determined by using image moment descriptors. An accurate target detection is obtained by the vision and an associated GPS for the helicopter navigation. In GPS-denied areas or once the GPS is lost, this approach will not be accurate anymore. Therefore, vision information is combined with inertial sensing to perform the landing in GPS-denied areas [6]. In [8], the vision system comprises of a camera fixed on a pan/tilt unit (PTU), in addition a special helipad with five different sizes of circles which was designed to achieve fast recognition. The position and the attitude are estimated using the images captured by the camera. An auto takeoff, hovering, and landing were achieved using low cost inboard hardware, where a Wii remote infrared camera and a T-shaped pattern made of four infrared LED indicators placed at the landing place. The camera was mounted at the center of the quadcopter frame looking down to ensure that the quadcopter was above the landing place.

Landing for an unmanned aerial vehicle based on a multiple view algorithm in which state estimates were estimated using vision based algorithm the proposed algorithm [7]. The vision-based controller sends the commands to the helicopter to hover over the landing target. Once the camera detects the target, the estimated (x, y) position of the landing target will be the desired hover point. [9] investigates a full 6 DOF pose estimation of an MAV using on-board monocular vision solutions. A special landing pad was designed to be recognized by the object recognition algorithm. The landing pad contains a letter (H) bordered by a circle. From the projection of the known circle, the 3D position, and roll and pitch angles are estimated, and the yaw angle is estimated from the projection of the letter “H”. In [10], a 3D vision based over-ground height estimation approach for landing control of an unmanned helicopter was presented. The height was estimated by a plane fitting method, which is an insensitive to change in attitude of the helicopter. A two-stage landing controller was designed, where the first stage is the descending phase and the second is the landing phase which starts at a certain height around 2 meters.

The problem of landing a helicopter in unknown areas was resolved in [11] using vision based terrain recovery. The combined concepts of computer vision were used to find a safe landing area avoiding trees and other obstacles, and navigate the vehicle to the landing site. The landing structure comprises of a low-level vision to recover a ground elevation map and a high-level navigation system for requiring way-points to the desired landing target. In [12], a vision system was based on off-the-shelf hardware and provides real time estimates at 30 Hz of the position and orientation of the UAV relative to the landing pad. An image processing segmentation, feature point extraction, and camera control algorithm were used to estimate the helicopter states and land the helicopter correctly.

A landing and navigation algorithm using natural landmarks was proposed in [4], where the UAV was able to navigate from an initial to a final position in a partially known environment. The remote user defines the target areas from aerial or satellite images to decide either the waypoints of the navigation trajectory or the landing area. The natural landmarks are founded by a feature-based image-matching algorithm, which sends feedback to an onboard control system for autonomous navigation and landing. In [13], the design and the implementation of control algorithm for the takeoff and landing for a prototype coaxial unmanned helicopter was investigated. Ground forces feedback was used to stabilize the longitudinal forces, while the attitude and drifting feedbacks were used to balance for the helicopter's longitudinal and lateral movements during takeoff and landing.

The design of an autopilot for a tethered helicopter for autonomous landing on a moving ship platform was presented in [1]. The tether was used for the safety and ease of landing. Two control approaches were designed to achieve the task: the first approach is the position control and the second is the attitude–altitude control. The outer loop is the position control, which is slower than the inner attitude-altitude loop. To predict the position of the target and track the trajectory of the helicopter, a Kalman filter was used in [2]. The landing distance is given, and discrete updates of the trajectory are performed to track the path and land on the target using an object recognition algorithm.

The design of an autopilot for autonomous landing of a helicopter on a rocking ship was investigated in [3]. The deck was modeled to behave in a sinusoidal motion, and a tether was used to support the target tracking mission. A time-scale controller was designed to ensure landing of the helicopter on the ship, in which the angle between the cable and the helicopter is used to achieve the landing on the moving ship platform. Due to the inaccuracy of the GPS readings, the optical flow approaches were used to stabilize the position of the quadrotor just before reaching the ground. To control the height of the quadrotor, two types of height controller were designed and tested in [14]. An automatic landing was achieved and an altitude controller was designed using the information coming from the ultrasonic sensor [15].

In [16], optical flow Sensors are deemed as efficient and robust navigational sensors to suite for MAVs. These were used to avoid collisions and to measure the altitude during the landing stage. In [17], an optical flow mouse sensor was used for height approximation and terrain navigation. In [18], a nonlinear controller for hovering flight and landing control on a moving platform for a (VTOL) Unmanned Aerial Vehicle (UAV) was designed and implemented using optical flow hardware. The divergent optical flow was used as feedback information to regulate the automatic vertical landing on a moving platform. In [19], the optical mouse sensor was proposed to be considered as an incremental rotary encoder. The major sensitivity issues of using an optical flow are the sensitivity to the changes in height and accumulative error in the measured motion in random paths. Those issues were solved by maintaining the height and pointing to the same surface.

In [20], [21] and [22], the problem of the ground effect on the main rotor of the helicopter is addressed. In [22], the ground effect on the main rotor while the helicopter is in forward flight was estimated approximately. A theoretical analysis was done to show how the ground effect becomes significant while the helicopter approaches the ground in different wind speeds. In [20], the tip vortex geometry and performance of the main rotor of the helicopter were examined in both zones, while the helicopter went in and out of the ground effect. It was observed that the tip vortices were decreased very close to the ground plane. In [23], a comparison was done between the out ground effect and in the ground effect wake of a hovering rotor. The changing in the airflow structure beneath the rotor while the helicopter was hovering at (Out Ground Effect) OGE and at IGE (In Ground Effect) was noticed and plotted. An empirical formula was used in [21] to compensate for the ground effect thrust of the main rotor of a coaxial small RC helicopter.

The autonomous flight of unmanned aerial vehicles has been getting much more attention in the last few years. The three basic phases of any autonomous flight of a rotorcraft are the takeoff, hovering and the final phase is the landing phase. Autonomous landing on a predefined target has been considered a challenge, in terms of finding a good enough place to perform the landing [4], and the landing technique itself [16] [6], and [24].

Optical flow sensors have been employed in unmanned systems. Low cost optical flow sensors have gained acceptance for their applications in ground mobile robots [25] flying robots UAV. Optical flow sensor have been used in unmanned quadrotor [26-

30], and in the fixed wing [18]. Most of the existing research on using optical flow sensors in unmanned aerial vehicle has been done with quadrotors and fixed wing [17]. Not much research on exploiting the optical flow sensors by a single rotor helicopter. In [24], optical flow regulation algorithms are applied to perform autonomous aircraft missions such as taking off, landing, hovering, and cruising. An elementary motion detector (EMD) is fixed beneath a rotorcraft optical flow of the ground image. In [13], the optical flow and ultrasonic sensors are used to stabilize the small scale helicopter while performing perch and stare maneuvers.

Much research attention has been concentrated on developing high-accuracy sensor fusion algorithms for use in autonomous systems [31] [32] [33]. These fusion algorithms have to take into account practical systems' specifications and limitations. These include small computational complexity and robustness to common sensor problems such as accounting for measurement delays, missed measurement epochs, and uncertain observations [31] [32]. In addition to vehicle navigation, sensor fusion algorithms are also essential for target tracking applications [33]. Small-scale helicopters are considered a viable candidate for many different applications such as surveillance missions, power line inspection, wildlife monitoring, and several military applications. Recently, researchers have focused on the design of autonomous control systems to control helicopters in different missions [34][4]. In order for any control law to work, an accurate knowledge of the different states of the vehicle is needed. In this study, the model of a Maxi Joker 3 helicopter is utilized.

Precise state estimation is essential for controlling the helicopter autonomously. Nevertheless, it is hard to obtain accurate values for different helicopter states because of the large drifts, possible measurement bias, and immense noise of the onboard sensors [35] [36] Sensors with such errors are commonly used in vertical take-off and landing (VTOL) of unmanned aerial vehicles (UAV) because of their low weight, small size, and low power consumption. By fusing the measurements of different sensors, an accurate estimate can be obtained [37-41].

In [42], a data fusion between the kinematic optical flow sensing and the GPS/INS was used to estimate the position and the velocity of an object revolving in three dimensional space. In [43] [44], state estimation using a Kalman filter was done using the dynamic model of the helicopter platform, and in [43]. We were able to estimate the attitude and flapping angles of the helicopter with high accuracy and minimal errors not exceeding 0.3° in longitudinal flapping angles and 0.1° in lateral flapping the experimental validation was introduced.

In [45], an IMU/GPS system was designed and implemented using a low-cost COTS components. And to maximize the accuracy of the proposed IMU/GPS system, the IMU data is fused with GPS measurements using a loosely-coupled kalman filter. In [46], the formation flight instrumentation system (FFIS) was designed to increase the precision of the relative position and velocity between two moving aircraft. Each aircraft was equipped with (GPS) receiver, and an inertial navigation sensor (INS) unit combined with a wireless system to share measurements between aircrafts. And an extended Kalman filter was used to obtain an accurate relative state estimates. In [47] [48], an intelligent fusion between the GPS and IMU was developed. An adaptive Neuro-Fuzzy Inference System (ANFIS) was proposed for fusing the data coming from the GPS and IMU and estimating the velocity, and the position states.

Table 1 below summarizes the most relevant previous work. It includes the main sensors and platforms used to perform the auto takeoff and landing, and results of these investigations.

Table 1: The summarized literature review

Reference	Platform	VTOL Sensors	Results	Remarks
[5]	AVATAR Helicopter	▪ Novatel RT -20 DGPS. ▪ CCD camera. ▪ Ultrasonic.	Landing with 47 cm average position error.	▪ Very expensive solution. ▪ Computationally heavy.
[6]	Yamaha RMAX helicopter	▪ CCD camera mounted on an off the shelf ▪ Gyros & accelerometers.	Landing with 43 cm average position error (54 cm maximum position error).	▪ Computationally heavy (two PC104 stacks with 700 MHz processors). ▪ ▪ Works with a high payload helicopters.
[15]	Small scale helicopter	▪ IMU ▪ Range sensor	Auto takeoff and landing with 0.5 m altitude error.	▪ Ultra sonic sensor is not that accurate at high altitudes.
[8]	ASCTec Hummingbird Quadrotor	▪ IMU ▪ Infrared Camera. ▪ IR LEDs placed in T-shape.	50 indoor VTOL flight tests were conducted, the standard deviation is less than 3 cm in each position axis.	▪ The solution will not work properly in outdoor environments. ▪ The camera should be close enough to the pattern to detect the IR spots.
[9]	MAV Quadrotor	▪ On-board Monocular vision solution. ▪ IMU.	Pose estimation based on a single image of the landing pad.	▪ Special landing pad need to be design to get an accurate estimate the yaw angle. ▪ Very light weight sensor should be used due to the small payload.
[14]	Mini Quadrotor	▪ Optical flow sensor. ▪ GPS. ▪ IMU. ▪ Sonar sensor.	Position stabilization for the Quadrotor at the final meter before touching ground.	▪ No estimator was designed.
[29]	Cheetah Quadrotor	▪ PX4FLOW Sensor. ▪ IMU.	Hovering and outdoor flight trajectory.	▪ It shows a better results comparing to the previously designed optical flow sensors. ▪ The output flow are auto-compensated for the 3D rotations.

1.3 Problem Statement

In this research, we are addressing autonomous takeoff and precision landing using sensor fusion between a GPS/INS navigation solution and optical flow sensor readings in the terminal phase of the landing process. Precision landing using GPS/INS COTS is not possible due to the position error estimates. Therefore, enhanced position estimates are needed for the terminal phase of landing the helicopter on a prescribed area. Two major causes describe the challenges in achieving precision landing which are addressed in the present work. The first is the ground effect which is described next. This problem is solved by using a ground effect model to obtain accurate thrust compensation while the helicopter is in the ground zone. The second cause is the GPS position inaccuracies. The precision of the GPS in stabilizing the position of the rotorcraft is low, and the problem becomes significant when the rotorcraft is close to the ground because the GPS quality at low altitudes is usually worse compared with the GPS quality while flying. Those reasons tell us to use a better position estimation technique especially for the last few meters in landing, in which the GPS readings become unreliable. This technique is to measure the movements of the helicopter by a more precise sensor. The use of the optical flow sensor will solve the problem and avoid the inaccuracies coming from the GPS sensor. This investigation develops a precise landing algorithm for landing a helicopter on a predefined marked target with a known GPS location. Figure 1 illustrates the proposed precision landing technique based on the GPS/INS and the optical flow sensor fusion algorithm used at in the final stages of the landing phase of the flight.

To achieve an autonomous landing within a certain radius R2, a GPS/INS based landing is sufficient. While a precise landing within a smaller radius R1 is not possible using GPS/INS. Therefore, there is a need to use other more precise sensors such as optical flow mouse sensors. Hence, this investigation addresses the problem of precision landing using an optical flow algorithm.

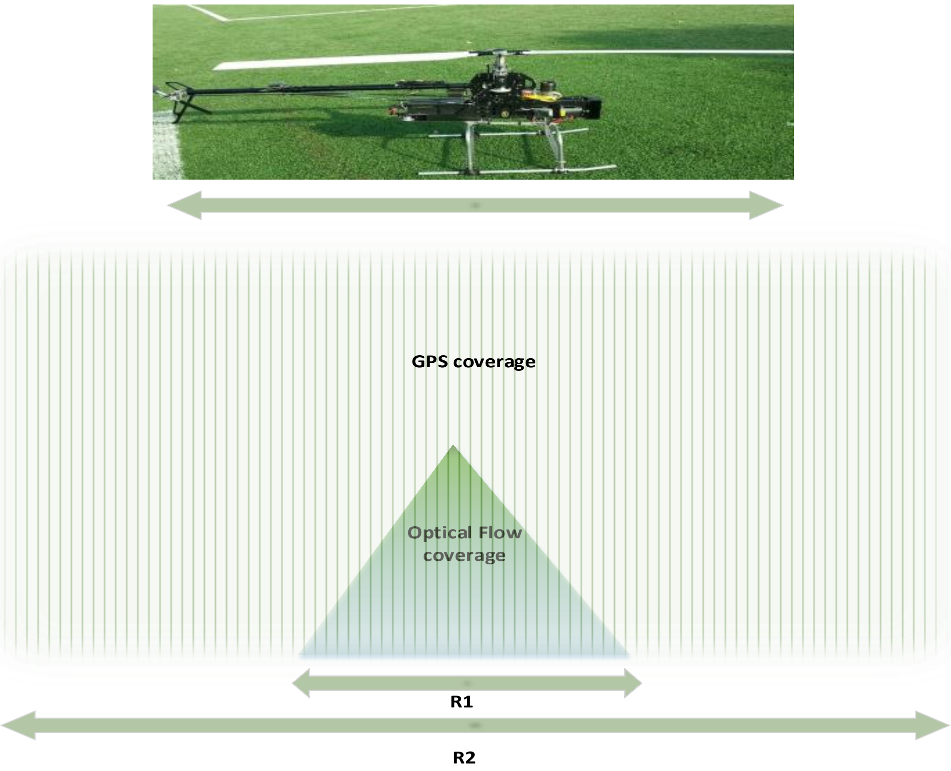

Figure 1: Precision landing position error by using different sensors

1.4 Thesis Contribution

The main contributions of this research work can be summarized as follows:

- Obtain a robust dynamic models for the optical flow sensor. These models are based on a set of real time experiments conducted under simulated landing conditions. Also, validate these obtained models by comparing the output of these models with the actual reading of the optical flow sensor.
- Develop a sensor fusion algorithm between the GPS/INS sensor and the optical flow sensor to land the vehicle at the desired landing touch down location.
- Apply the proposed algorithm on the Maxi Joker 3 helicopter and perform an accurate auto takeoff and precise landing.

1.5 Thesis Outline

This thesis is organized as follows:

In Chapter 1, we start with an introduction to the use of unmanned aerial vehicle, and we cite the literature that relevant to the subject. It also includes the problem statement and the major contributions of this study.

Chapter 2 presents the background of unmanned aerial helicopters. It describes the helicopter and its controls. It introduces the optical flow state of art, algorithms and an introduction about the smart optical flow sensor used.

In Chapter 3, the mathematical approaches for computing the optical flow were introduced. The motion field equations were addressed. Also, the basic concepts are of the optical flow sensing are thoroughly explained.

In Chapter 4, the top-down modeling style was used to represent the mathematical model of the Maxi Joker 3 helicopter.

In Chapter 5, we present the experimental design of modeling the optical flow sensor, and the performed validation tests. The obtained optical flow sensor models are listed and the validation results are meticulously described.

Chapter 6 illustrates the sensor fusion algorithm to estimate the position and the velocity of the helicopter. Then, the algorithm is integrated into autopilot, and auto takeoff and landing tests were performed to prove the use of the proposed algorithm for performing the auto landing.

In Chapter 7, the achieved work is concluded and the planned work is summarized.

Chapter 2: Background

This chapter presents the background for the research presented in this thesis. Section 2.1 shows the advantages of using aerial helicopters and their applications. In Section 2.2, the main components of RC helicopters are illustrated and the helicopter control inputs are explained as well. Section 2.3 introduces the state of art optical flow sensor. The most used optical flow algorithms are described in Section 2.4. In Section 2.5, an introduction to the PX4FLOW smart sensor used in the present investigation is introduced.

2.1 Unmanned aerial helicopters

Lately, research on unmanned aerial helicopters has increased drastically because of the unique features that the helicopter has. Rotary wing UAVs are more advantageous than fixed wing UAVs in the sense of performing vertical takeoff and landing, hovering performance, low speed cruising, and they have the ability to operate close to the ground. However, they have a complicated mechanical system, and unstable dynamics. This creates a need to design a precise dynamic model, and a robust controller to ensure stable and safe operation of the helicopter.

2.2 Unmanned aerial RC helicopters

This section is about illustrating the basic model of the RC helicopter, the electrical and mechanical subsystems, and the helicopter controls. The rotor hub (1), main rotor (2), swash plate (3), tail servo (4), tail boom (5), tail rotor (6), and landing skid (7) are the main parts of the helicopter and its control as demonstrated in Figure 2. The movements of the helicopter can be controlled by four inputs: collective, longitudinal cyclic and lateral cyclic, and the rudder. Figure 2 depicts the main RC helicopter parts and its controls.

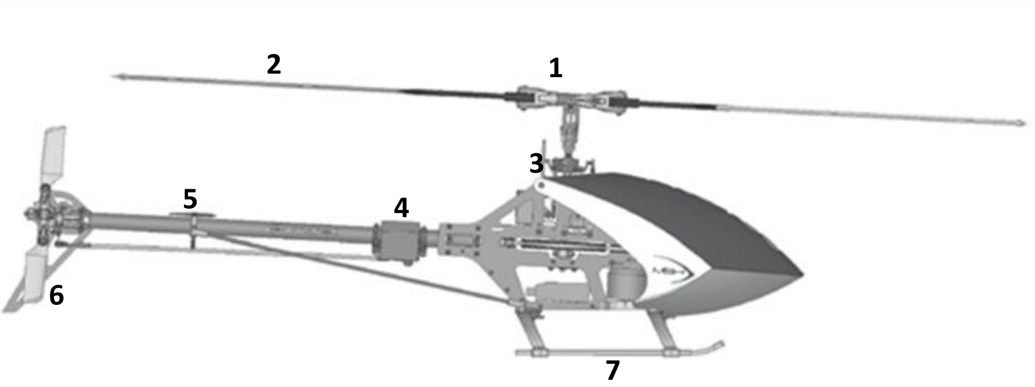

Figure 2: RC Helicopter Main Components [26]

The lift force that is needed to keep the helicopter airborne is produced by the main rotor. By tilting the main rotor, movements in the longitudinal and the lateral directions are generated. This can be done through the swash plate assembly, which is linked to the 120 CCPM servos configuration. And, the heading direction can be controlled by spinning the tail rotor and encountering the torque reaction created by the main rotor.

In this thesis, we consider the auto takeoff and landing of the Maxi Joker 3 RC helicopter shown in Figure 3.

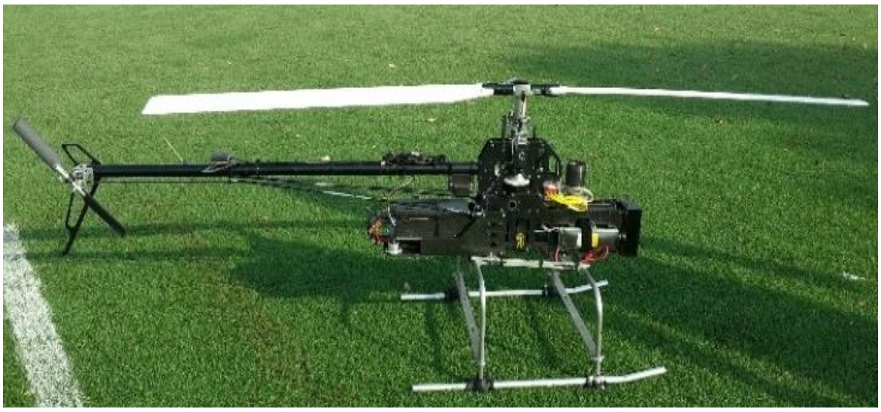

Figure 3: Maxi joker 3 helicopter

Maxi Joker is a 3-axis gyro as an electronic stabilizer. Using the 3-axis gyro adds to the stability of the helicopter and makes it easy to control. Figure 4 shows the response of the helicopter where there is no 3-axis gyro used. Figure 5 depicts the advantage behind the use of the 3-axis gyro. Obviously, when utilizing the 3-axis gyro there is no tilting of the rotor, as the rotor part stays horizontal and the helicopter's body swings,which gives more stability to the flight and the system becomes easier to control.

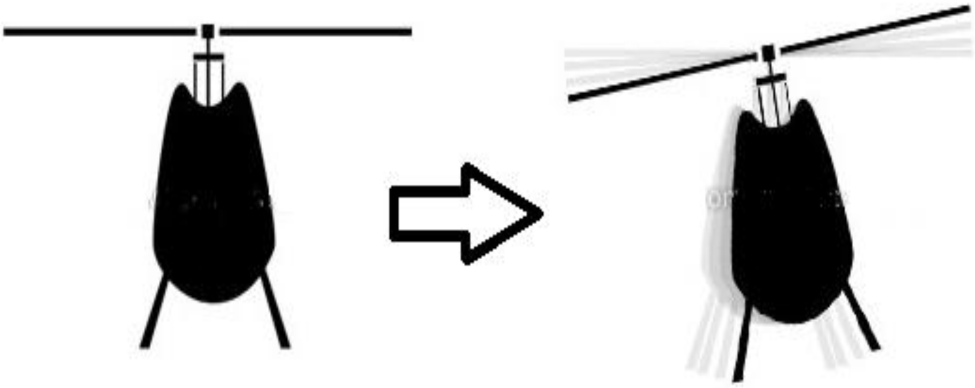

Figure 4: RC flybarless helicopter response without using (3-axis gyro) [49]

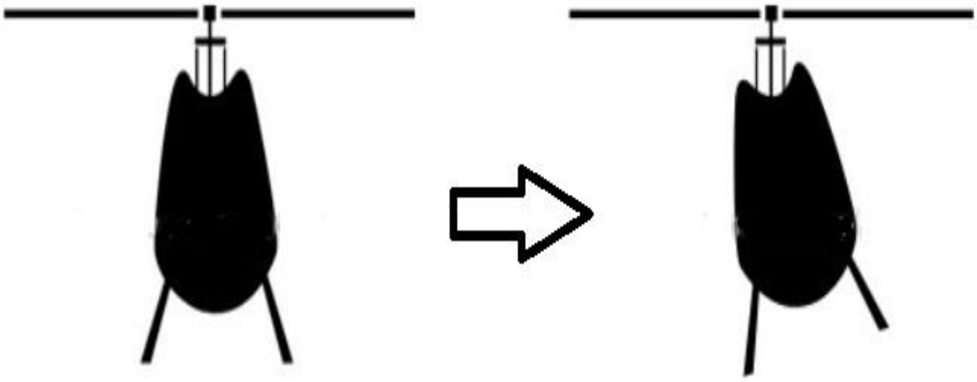

Figure 5: Flybarless helicopter response with using 3-axis gyro stabilizer [49]

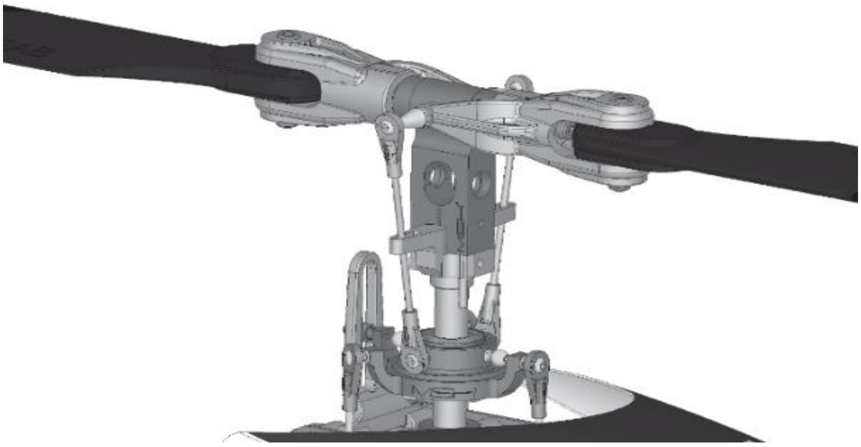

Figure 6 : Flybarless Helicopter Rotor Head [49]

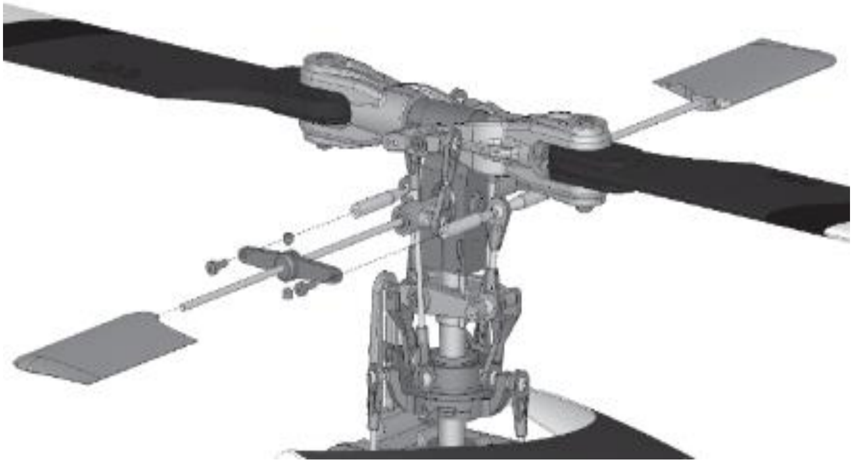

Figure 7: Flybar Helicopter Rotor Head [49]

2.3 State-of-the-Art Optical Flow

An ADNS-3080 optical flow sensor [50] is used in controlling the hovering of a micro quadrotor. The same optical flow sensor is utilized in [51] to obtain information on the velocity and the position of the quadrotor, especially in places where there are no GPS signals. An ADNS-2620 mouse optical flow sensor is used with an IMU to estimate quadrotor's velocity [27] using an extended Kalman filter. In [42], the position and the velocity of an object evolving in three dimensional space are estimated using a Kalman filter. A sensor fusion algorithm is designed between the GPS/INS sensor and the optical flow sensor. The optical flow sensor is used to provide the object's velocity while the position is measured by the GPS sensor. The heading of a small fixed pitch quadrotor is estimated as well in [28] by applying optical flow algorithms on the images captured by a camera looking downward. An optical flow sensor is used to control the motion of the quadrotor in [30], a vision based PID controller is designed based on the readings obtained from the optical flow sensor. Optical flow sensor is used in [18] to control the landing of the quadrotor on a moving platform. The quadrotor is equipped with a camera and an IMU to stabilize the corresponding motion of the vehicle with the moving platform and to control the auto vertical takeoff and landing of the quadrotor.

Optical flow sensors are sensitive to light and require a special light conditions to perform well, in many indoor applications or low-light conditions. To solve this, a special type of cameras which based on a visioning CMOS image is introduced in [29].. CMOS cameras can perform properly in outdoor, indoor and low-light conditions.

Honegger and Meier presented an open source and open hardware of an optical flow sensor design. In [29], a hover control experiment of a quadrotor is conducted indoors using the PX4FLOW smart camera which has an optical flow sensor and an ultrasonic sensor placed on the same PX4FLOW kit. The optical flow sensor is a 2D sensor, which means that the sensor's reading gets affected by the 3D movements such as the roll, and pitch of the vehicle rotations. Therefore, the ultrasonic sensor is used to measure the vertical distance to the ground and by knowing the vertical distance, the optical flow measured could be compensated for the 3D rotations and the OF readings will be corrected as well. The PX4FLOW smart camera performs well in both indoor and outdoor environments.

Since that the PX4FLOW sensor is more advantageous than the other mentioned optical flow sensors, and its measurements are compensated for the 3D rotations, there is no need to do further computations. In this study the PX4FLOW smart sensor is used to perform the auto takeoff and landing process for the helicopter on a distinguished pattern.

2.4 Optical flow algorithms

Various ways of calculating the optical flow of a scene are introduced in [52] and [53]. The optical flow can be computed via basic methods:

2.4.1 Feature-based algorithms

In this algorithm, at least two consecutive image frames have to be captured. This algorithm computes the optical flow by picking certain features in the first image and locating the same features in the consecutive image. Once the features are matched in the two images, the discrepancy between the two images can be calculated and the flow in the image can be computed too.

2.4.2 Gradient-based algorithms

This algorithm computes the gradient intensities which are higher at the location of the edges where the pixel intensity value changes the most (i.e., from light to dark). The gradient is computed by solving for the derivative in the, and y directions of the image.

Gradient based methods of computing optical flow are discussed in Horn and Schunk [54], Lucas and Kanade [55], Srinvasan [56].

2.4.3 Block matching algorithms

Block matching algorithms (BMA) are considered rigorous algorithms which can be utilized to detect moving objects, and for motion estimation purposes. In block matching, the image frame is subdivided into non-overlapping blocks, and the blocks in the current frame are matched to the blocks in the reference frame. To figure out the block which has the best matching among the candidate blocks in the reference frame, specific matching criteria are used. From which one can tell whether the block has moved or not. If the matched block has a different location compared with its location in the reference frame, this means that the block has moved [57] .

Feature-based optical flow computing algorithms are known as complex algorithms [53]. The process of matching between two specific consecutive images is hard, and it requires many computations to figure out the differences between the two image frames. Also, these algorithms are not accurate enough.

In this study, we are more interested in the block matching algorithms to compute the optical flow. In Chapter 3, the (BMA) algorithm will be illustrated.

2.5 PX4FLOW Optical Flow Sensor

Recently, the use of computer mouse optical sensors has increased considerably. Optical flow sensors are used in many engineering applications. They can be used in applications containing vision feedback control system. The smart PX4FLOW kit as shown in figure 8 uses an embedded high performance processor ARM Cortex M4 is to process the frames captured by the CMOS vision sensor. The processor performs optical flow processing at 250 frames per second and the resolution of the captured image frames is 64x64 pixels. The sensor focal length is 16 mm, and M12 lenses are used with a 21° FOV containing an IR-block covered with an Aptina MT9V034 imager. An ultrasonic sensor is used to measure the distance toward the scene. Knowing the distance to the scene helps with converting the optical flow sensor velocity readings into the metric scale.

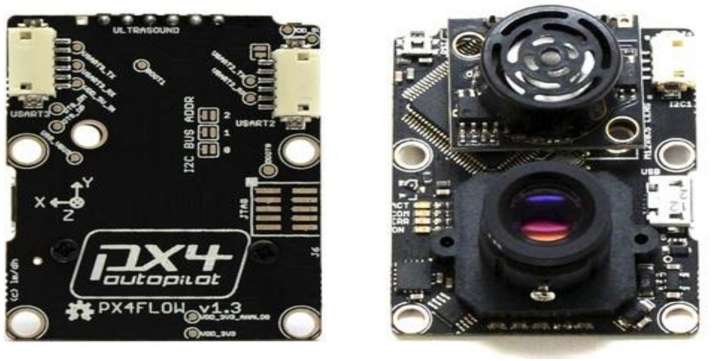

Figure 8: PX4FLOW smart optical sensor [29]

Chapter 3: Motion Field and Optical Flow Sensing

This chapter presents the theoretical side of optical flow. The geometrical approach is illustrated in Section 3.1. Section 3.2 presents the basic equations of the motion field, and the optical flow main concepts are also explained in Section 3.3.

3.1 Geometrical Approach

This approach illustrates how the actual displacement (dx, dy) of an object can be computed through the displacement readings of an optical flow sensor. In [58], the pinhole model is used to generate the actual displacements of objects. In this model, the point of an object in Cartesian coordinates (x, y, z) is projected on the focal plane. This plane, as shown below in Figure 9, which is located between the object scene and the center of projection.

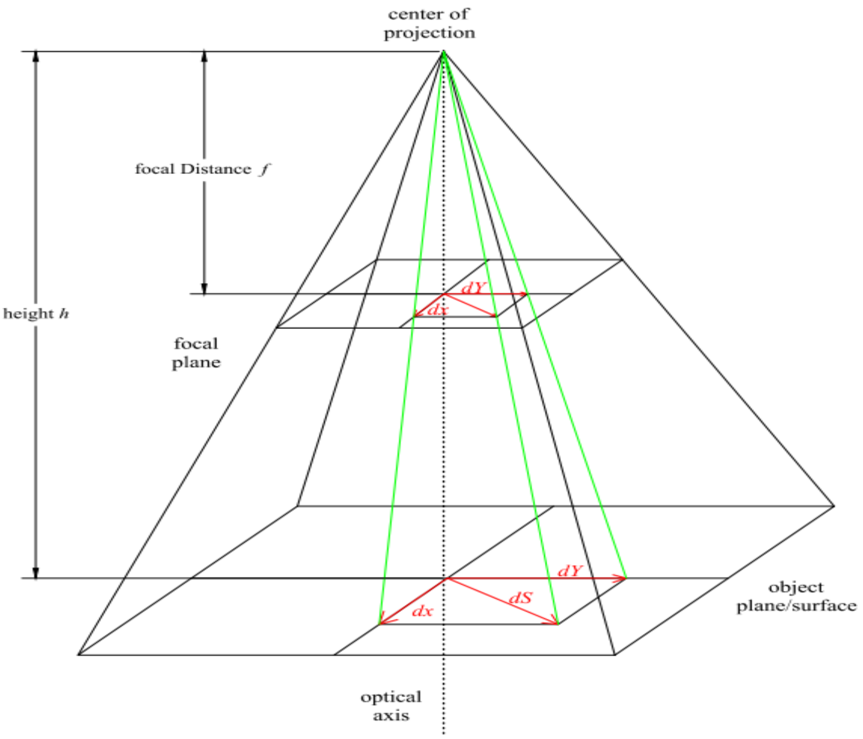

Figure 9: Illustration of the OF Geometry

Since the point gets projected on a focal plane, its coordinates will be changed to be in (u, v) which are the focal plane coordinates. In addition, a lens is used to focus the image of the scene on the sensor's image plane. Because the optical flow sensor has a lens used to focus the image displacement on the image plane of the sensor, the lens law is applied:

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{b} \tag{1}$$

$$h = b + g \tag{2}$$

where g denotes the distance between an object point and the lens plane, b is the distance between the image plane and the lens plane and f is the focal length of the lens. The summation of b and g yields the total distance h between the object plane and the image plane.

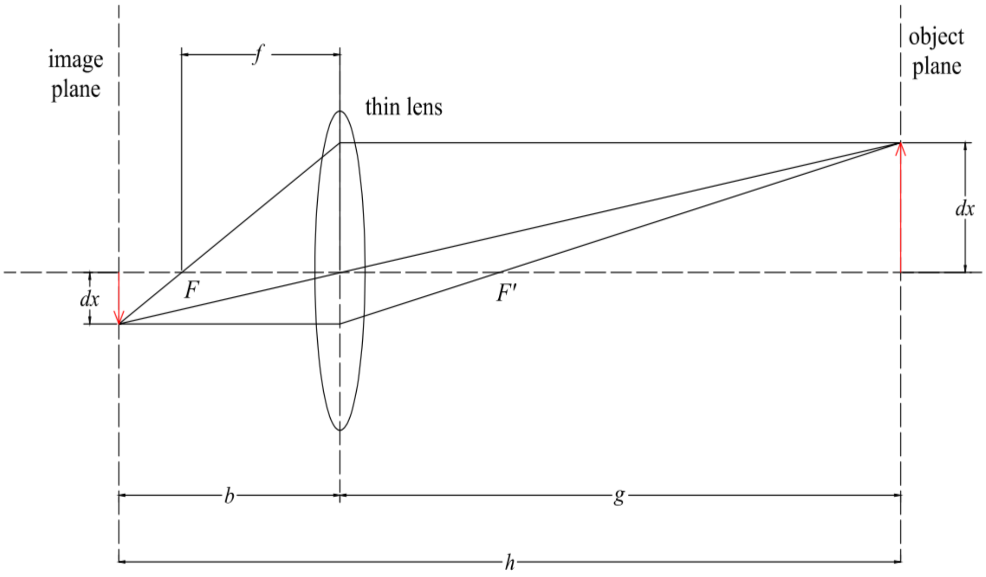

Figure 10: Object displacement projection on the image plane with lens focusing

Solving equation (1) for b shows that at low heights the image distance is very sensitive to small changes in g, while at high heights the image distance almost converges to f.

The actual displacements are computed as follows:

$$dX = res\,.\ g.\ \frac{dx}{b} \tag{3}$$

$$dY = res\,.\ g.\ \frac{dy}{b} \tag{4}$$

where res is the sensor resolution in $(cm/count)$. Ideally, the focal distance of the lens used is known. g and b are determined by solving equations (1) and (2). Then, the actual displacement can be computed.

3.2 Motion field equations

The motion field is created by projecting the 3D velocity field on the image plane. Assume $\mathbf{P} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}]^{\mathbf{T}}$ is a moving point in a 3D camera reference frame as shown in Figure 11.

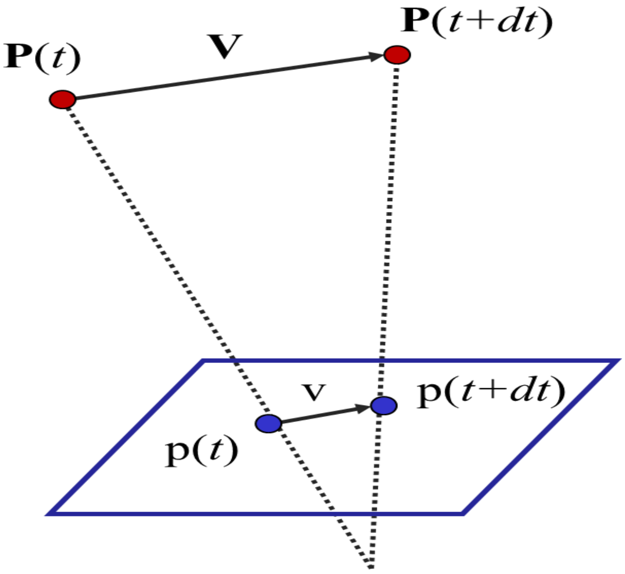

Figure 11: The projection of point moving in the reference frame on the image plane

Assume that the optical axis is the **Z**-axis of the reference camera frame, and f denotes the focal length and that the center of the projection is located exactly at the origin. Then, the projected coordinates in pixel of **P** on the image plane are obtained via

$$p = f\frac{\mathbf{P}}{\mathrm{Z}} \tag{5}$$

The distance of the image plane to the origin is equal to the focal length f so the third coordinate of p is the focal length $p = [\mathrm{x}, \mathrm{y}, \mathrm{f}]^{\mathrm{T}}$. The relative motion between the camera and a point in the image plane can be computed as follows:

$$\mathbf{V} = -\mathbf{T} - \omega \times \mathbf{P} \tag{6}$$

where **T** denotes the translational velocity and ω is the angular velocity. The image velocity v can be obtained by deriving (5) with respect to time.

$$\mathbf{v} = \mathrm{f}\frac{(\mathrm{Z}\mathbf{V} - \mathrm{V_Z}\mathbf{P})}{\mathrm{Z}^2} \tag{7}$$

By substituting in (6) the motion field equations components in x and y are computed through (8) and (9) as follows:

$$\mathrm{v_x} = \frac{(\mathrm{T_z x} - \mathrm{T_x f})}{Z} - \omega_\mathrm{y}\mathrm{f} + \omega_\mathrm{z}\mathrm{y} + \frac{(\omega_\mathrm{x}\mathrm{xy} - \omega_\mathrm{y}\mathrm{x}^2\,)}{\mathrm{f}} \tag{8}$$

$$\mathrm{v_y} = \frac{(\mathrm{T_z y} - \mathrm{T_y f})}{Z} - \omega_\mathrm{x}\mathrm{f} + \omega_\mathrm{z}\mathrm{x} + \frac{(\omega_\mathrm{x}\mathrm{y}^2\, - \omega_\mathrm{y}\mathrm{xy})}{\mathrm{f}} \tag{9}$$

where $\mathrm{T_x}$, $\mathrm{T_y}$, and $\mathrm{T_z}$ denote the translational velocity components in each axis. And ω_x , ω_y , and ω_z are the angular velocity components in around each axis.

Obviously, the motion field equations consist of a translational component and rotational component of motion. The translational velocity components in (8) and (9) are dependent on the depth of the scene Z while the rotational velocities are not. This means that the rotational components of the motion fields have no information concerning the depth of the scene [29].

3.3 Optical Flow Sensing

Optical flow sensing is a purely bio-inspired technique. Insects make big use of vision, and in particular they use optic flow to navigate, take off and land. Srinivasan [59] noticed that honeybees perform their landing by keeping the optic flow on the landing spot constant. By mimicking the honeybees’ performance, Green [16] [60] designed a prototype called CQAR Closed Quarter Aerial Robot that was capable of performing autonomous missions like, takeoff, landing and collision avoidance using an optical flow microsensor for providing the depth information. Applying the optic flow sensing technique to a flybarless single rotor helicopter simulates how an insect flies and lands on a specific landing point. Using a single camera as a sensor to measure the optical flow of the scene has been introduced in many published work [10] [51] [11]. By capturing a two consecutive image frames, and studying the pixel movements we can tell that the camera went through either a translation or a rotation movement.

Optic flow is basically the apparent visual motion seen by a sensor which travels over a certain scene. This motion is created either by the motion of the scene (SCMO) or the motion of the sensor throughout the scene (MCSO). And there are two other cases. The first includes a moving camera and a moving object (MCMO), and the second is the (SCSO) when the camera and the scene are stationary. In this thesis, we are interested in performing the landing and takeoff over a stationary object (MCSO). In this research, we assume that the optical flow sensor is placed beneath the helicopter, facing the ground and moving exactly with the helicopter. And the object is stationary.

Since the optical flow is the apparent visual motion of the object, objects that are close to the camera seem to move faster than objects which are relatively far (as shown in Figure 12). In Figure 12 (a) the pixel movements is faster than its movement in 12 (b).

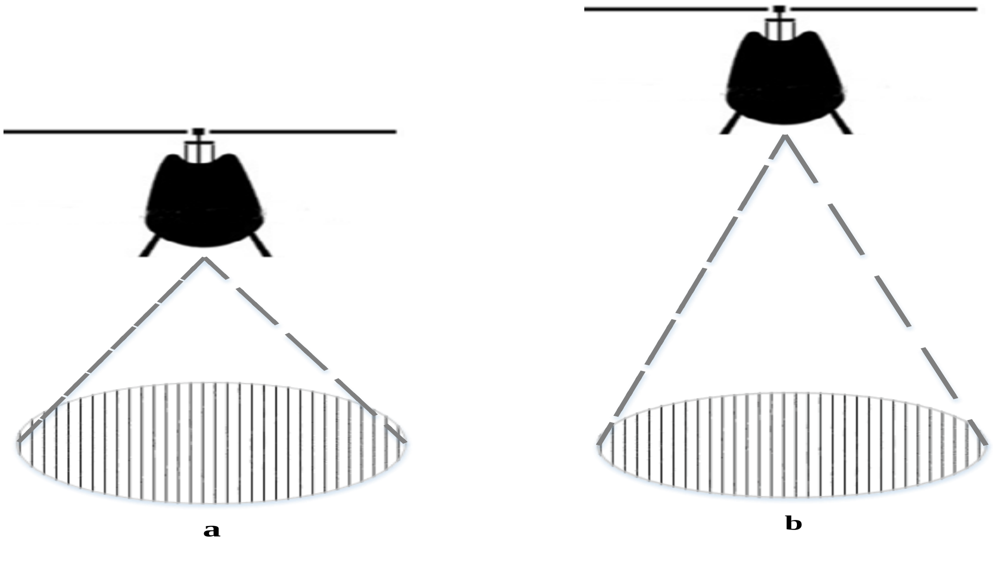

Figure 12: Optical flow values at different altitudes

Figure 13 illustrates how the helicopter reacts to the 3D movements. Figure 13 (a) describes the helicopter when no rotation is applied, while in 13 (b) the helicopter is rolled by a small degrees. Clearly, in (a) the pattern fills the view and the measured optical flow of the image will be accurate. While the pattern is partially seen in 13 (b), due to this rotation the calculated flow value will be inaccurate.

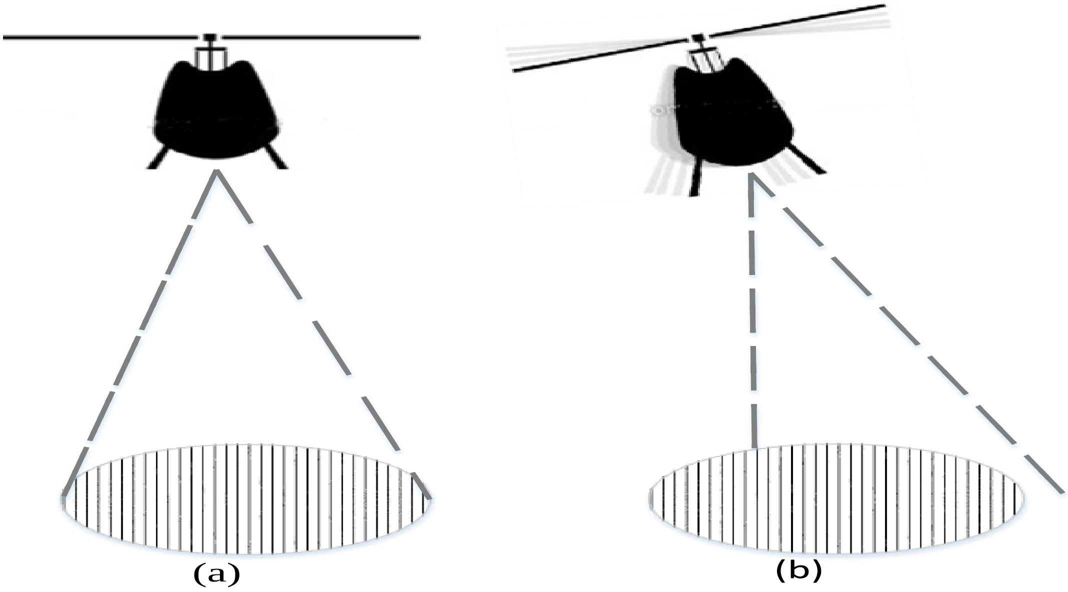

Figure 13: The effect of the 3D rotations on the optical flow values

Chapter 4: Helicopter Mathematical Modeling

4.1 Background

The non-linear model of the helicopter is presented in this chapter. A flybarless nonlinear model for the Maxi Joker 3 is introduced. A top-down modelling approach is used to model the Maxi Joker 3 helicopter. As shown in figure 14, the flybareless helicopter model consists of four main blocks. The commanded signals (u_{lat}, u_{lon}, $u_{col}$and u_{ped}) to the flapping and thrust equations are generated from the first block which represents the actuator dynamics. In the second block, the thrust equations the thrust for both rotors is computed and the flapping angles of the main rotor are computed. The force and the torque equations are derived in the third block. The forces and the torques are used to generate the rigid body equations of the helicopter. The position and the velocities of the helicopter are related to the earth frame. Figure 14 represents the overall dynamic model of the mentioned helicopter. A Matlab Simulink environment is used to simulate the helicopter model.

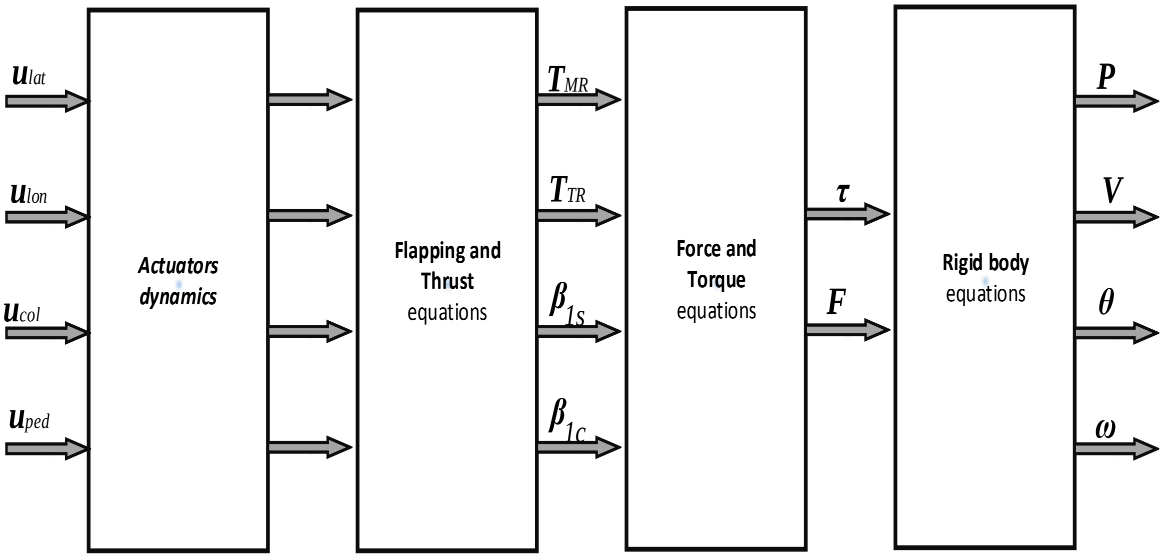

Figure 14: Top-down modelling approach of a flybarless helicopter

There are many reference frames, and we discuss how we can go from a specific frame to another frame freely using transformation matrices. This is discussed in the following sections.

4.2 Reference Frames

To describe the position and orientation of the helicopter, two coordinate frames are used: the body frame and the inertial frame. The body frame (BF) has its origin in the center of gravity (CG) of the helicopter. This right-handed frame is shown in Figure 18. As seen from the figure, the BF axes are denoted as $x^b, y^b,$ and z^b. On the other hand, the earth frame (EF) is an inertial frame with axes denoted as $x^e, y^e,$ and z^e. The position and velocity of the helicopter are described using the EF. The EF is positioned at a fixed point on the Earth’s surface, usually at the location of the monitoring ground station (see Figure 15) [43].

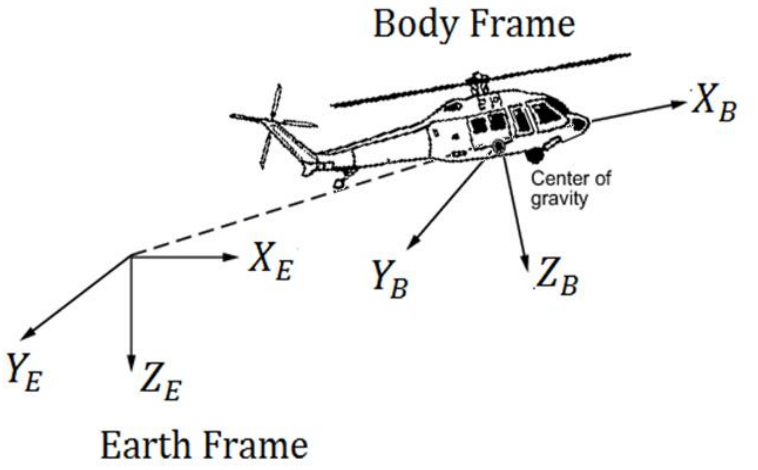

Figure 15: Reference frames

4.3 Matrix Transformation

The forces and moments in the BF are transferred to the EF using a rotation matrix. The orthonormal rotation matrix R_b^e is used to change the representation of a vector from the BF to the EF and vice versa. The matrix is obtained by successive rotations about the x, y, and z-axes by the roll, pitch, and yaw Euler angles ϕ, θ, and ψ, respectively [49].

$$R_b^e = \begin{bmatrix} c(\theta)\,s(\psi) & c(\psi)\,s(\theta)\,s(\phi) - s(\psi)\,c(\phi) & c(\psi)\,s(\theta)\,c(\phi) + s(\psi)\,s(\phi) \\ c(\theta)\,s(\psi) & s(\psi)\,s(\theta)\,s(\phi) + c(\psi)\,c(\phi) & s(\psi)\,s(\theta)\,c(\phi) - c(\psi)\,s(\phi) \\ -s(\theta) & c(\theta)\,s(\phi) & c(\theta)\,c(\phi) \end{bmatrix} \tag{10}$$

The main function of the rotation matrix is that it can be used to convert the position vector (X^b) in BF to its corresponding vector (X^e) in EF (see Eq. (11)).

$$\mathbf{X}^{\mathrm{e}} = R_b^e \, . \, \mathbf{X}^{\mathrm{b}} \tag{11}$$

The body angular rates are expressed about the BF, and they are denoted as $\boldsymbol{\omega}^{\mathbf{b}} = [p \;\; q \;\; r]^{\mathrm{T}}$. The angular velocities of the BF with respect to the EF are called the Euler rates, $\dot{\boldsymbol{\Theta}} = [\dot{\phi} \;\; \dot{\theta} \;\; \dot{\psi}]^{\mathrm{T}}$ vector represents the Euler rates.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} =$$

$$\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} \tag{12}$$

$$+ \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

By reversing the transformation, the Euler rates can be obtained as follows

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta)\sin(\phi) & \tan(\theta)\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sec(\theta)\sin(\phi) & \sec(\theta)\cos(\phi) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{13}$$

4.4 Rigid Body Equations

The rigid body equations of motion are listed as follows [49]:

$$\dot{V} = \frac{F}{m} - \omega \times V \tag{14}$$

$$\dot{\omega} = I^{-1}(M - \omega \times I\omega) \tag{15}$$

where F denotes the net force acting on the helicopter (N), ω represents the body angular velocity (rad/sec), V is the velocity in the BF (m/sec), M is the moment about the CG (N.m), and I is the moment of inertia matrix (kg. m^2).

$$V = \begin{bmatrix} u \\ v \\ w \end{bmatrix},$$

$$F = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix},$$

$$\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{16}$$

$$M = \begin{bmatrix} L \\ M \\ N \end{bmatrix},$$

$$\text{And } I = \begin{bmatrix} Ixx & Ixy & Ixz \\ Iyx & Iyy & Iyz \\ Izx & Izy & Izz \end{bmatrix}$$

4.5 Forces and Moments

With respect to the helicopter CG, the forces and moments are derived and expressed in the BF as illustrated in Figure 16. The thrust direction is presumed to be perpendicular to the tip path plane (TPP). The small effect of the vertical fin and the horizontal stabilizer on the dynamics of the helicopter at hover is neglected.

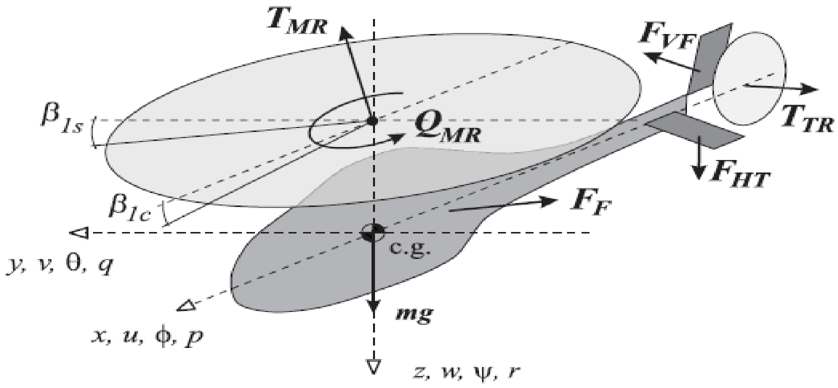

Figure 16: Moments and forces acting on the helicopter [61]

4.5.1 Forces

The force *F* acting on the helicopter in the BF has three components along the three axes f_x, f_y and f_z. These forces are made up of the following components:

- F_{MR} : The force generated by the main rotor thrust, and it is acting in the center of the main rotor disc.
- F_{TR} : The resulting force generated by the tail rotor thrust, and it is acting on the center of the tail rotor disc.
- $F_{\ g}$: The resulting force generated by the gravitational acceleration, and it is acting in the CG.
- F_{fus} : The resulting drag force generated by the fuselage, and it is acting in the CG.

The main rotor thrust can be defined as a function of the flapping angles which are β_{1s} and β_{1c} in the lateral and longitudinal ordinations, respectively.

The main rotor forces are listed as follows:

$$f_{x,MR} = -\,T_{MR}.\sin(\beta_{1c})\cos(\beta_{1c}) \tag{17}$$

$$f_{y,MR} = \quad T_{MR}.\sin(\beta_{1s})\cos(\beta_{1s}) \tag{18}$$

$$f_{z,MR} = -\,T_{MR}.\cos(\beta_{1s}).\cos(\beta_{1c}) \tag{19}$$

The force of the tail rotor has a component along the y-axis:

$$f_{y,TR} = \ -T_{TR} \tag{20}$$

The force due to the gravitational acceleration has three components along the three axes. The force can be written as follows:

$$f_{x,g} = -\ \sin(\theta)\,.\,m.\,g \tag{21}$$

$$f_{y,g} = \sin(\phi)\,.\cos(\theta).\,m.\,g \tag{22}$$

$$f_{z,g} = \cos(\phi)\,.\cos(\theta).\,m.\,g \tag{23}$$

The induced velocity for the main rotor thrust while performing a hover flight is introduced assuming that the inflow is steady and uniform. The inflow while the helicopter is hovering is shown below in Figure 17.

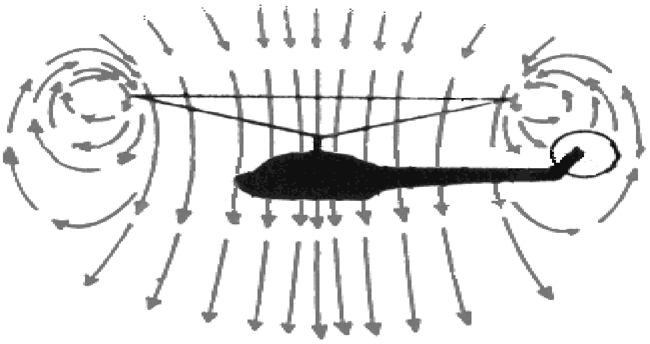

Figure 17: Inflow of the helicopter in hover mode

By applying the momentum theory, the induced velocity at a hover conditions can be found using the equation below [61]. For the Maxi Joker 3 helicopter the induced velocity at hover is 3.41 m/sec:

$$v_{ihover} = \sqrt{\frac{mg}{2\pi\rho {R_{mr}}^2}} = 3.41\,m/sec \tag{24}$$

The drag forces which are generated by induced flow oppose the direction of the helicopter movement. Then, the drag forces of the fuselage in the X, Y and Z axes are computed by:

$$f_{x,fus} = -\,S_x^{fus}\,\frac{1}{2}\,\rho\ v_{ihover}\ u \tag{25}$$

$$f_{y,fus} = -\,S_y^{fus}\frac{1}{2}\,\rho\,v_{ihover}\,v \tag{26}$$

$$f_{z,fus} = -\,S_z^{fus}\frac{1}{2}\,\rho(\,w_a + v_{ihover})V_\infty \tag{27}$$

where the vehicle cross sectional areas are computed to be as follows $S_x^{fus} = 0.0662$, $S_y^{fus} = 0.0872$, and $S_z^{fus} = 0.0993\ m^2$ [49].

4.5.2 Moments

The main sources of the moments acting on the helicopter are as follows

τ_{MR} : The moment generated by the main rootor.

τ_{TR} : The moment produced by the tail rotor.

τ_D : The drag moment on the main rotor.

The drag moment of the tail rotor is ignored because of its small effect compared to other moments. The helicopter moments are expressed in the BF along the 3-axes which are L, M and N respectively. The moments in the positive clockwise direction are considered positive. The dominant moments of the main rotor are the moments produced by the rotor flapping. In Figure 18, the moments resulting from the lateral flapping. The restraint in the blade attachment is expressed as a linear torsional spring with a constant stiffness coefficient K_β [61].

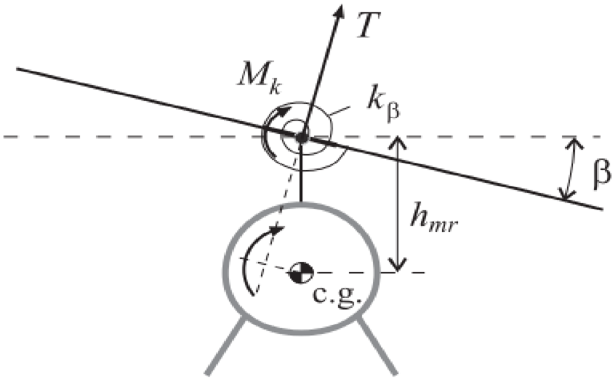

Figure 18: Dominant main rotor moments acting on the fuselage [61] .

The main rotor moments consist of three components:

$$L_{MR} = f_{y,MR}.h_{mr} - f_{z,MR}.y_{mr} + K_{\beta}.\beta_{1s} - L_b\tau_{fs}p \tag{28}$$

$$M_{MR} = f_{x,MR}.h_{mr} + f_{z,MR}.l_{mr} + K_{\beta}.\beta_{1c} - M_a\tau_{fc}q \tag{29}$$

$$N_{MR} = -\rho(\Omega R)^2\frac{bcR^2}{s}C_Q^{MR} \tag{30}$$

where the distance from CG to the main rotor is h_{mr}, h_{tr} is the distance from the CG to the tail rotor as shown in Figure 19, l_{mr} the distance from CG to the main rotor, l_{tr} is the distance from CG to the tail rotor and y_m is the distance from CG to the main rotor (see figure 20).

Since we are using the flybarless helicopter, the terms which represent the flybar dynamics (L_b , M_a) will be equal to zero. Then, the total torque coefficient is represented as:

$$C_Q^{MR} = C_T\sqrt{\frac{C_T}{2} + \frac{c_d s}{8}} \tag{31}$$

$$s = \frac{Nc}{\pi R} \tag{32}$$

where s is the rotor solidity which is the total blade area to the area of the disk and the drag coefficient c_d= 0.024.

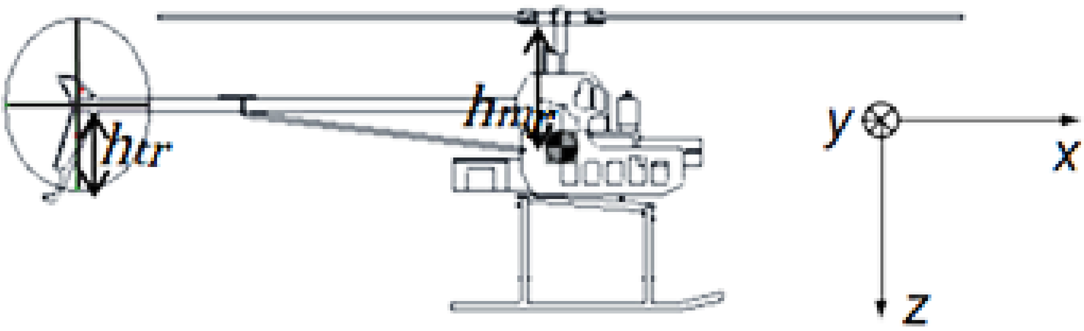

Figure 19: helicopter from side view [49]

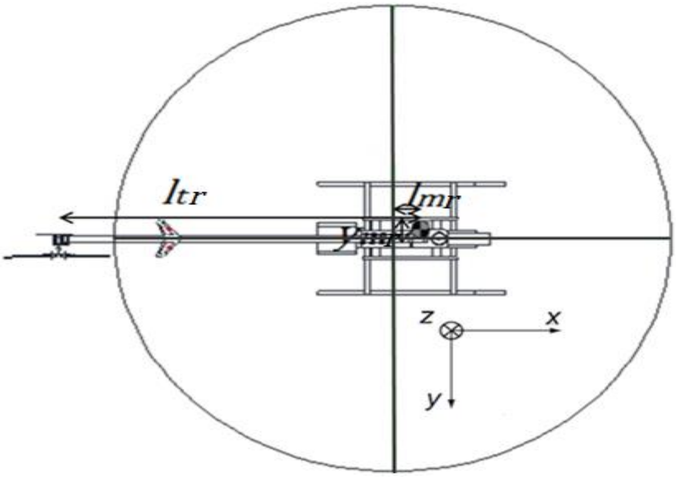

Figure 20: helicopter from top view [49]

The tail rotor contributes in creating moment along 2 axes:

$$L_{TR} = f_{y,TR}.\, h_{tr} \tag{33}$$

$$N_{TR} = -f_{y,TR}.\, l_{tr} \tag{34}$$

4.6 Flapping and Thrust Equations

The flapping and the thrust equations are presented in this section

4.6.1 Main Rotor Thrust and inflow Equations

It is shown in Eq. (35) that the thrust of the main rotor is directly related to the commanded collective and indirectly related to the induced velocity v_i and translatory velocities (u, v, w). The main rotor thrust is given in [62] as follows:

$$T_{MR} = \left[\theta_o\left(\frac{1}{3}+\frac{\mu^2}{2}\right)-\left(\frac{\mu_z+\lambda_i}{2}\right)\right]\frac{a.s}{2}\rho(\Omega R)^2 A_d \tag{35}$$

The rotor inflow, λ_i, has a direct influence on the thrust which adds the complexity of calculating the thrust. The ratio between the helicopter's translational velocity and the main rotor tip speed which is called the advance ratio (μ) can be found as in Eq. (36). λ_i is the inflow ratio for the main rotor, and a is the main rotor lift curve slope. μ_z is the ratio of vertical velocity to main rotor tip speed.

Padfield reveals that the rotor thrust is related proportionally to the rotor inflow. Also, the thrust increases whenever the collective is commanded, which is expressed by θ_o. The main rotor thrust equation is solved iteratively until T_{MR} and v_i are converged using the Newton-Raphson iteration technique explained in [63].

$$\mu = \frac{\sqrt{u_a^2 + v_a^2}}{\Omega R} \tag{36}$$

$$\mu_z = \frac{w_a}{\Omega R} \tag{37}$$

According to the momentum theory, the hover inflow λ_{hover} can be obtained as follows:

$$v_{ihover} = \sqrt{\frac{T}{2\rho A_d}} \tag{38}$$

While hovering the helicopter thrust is equal to the helicopter weight. The weight of the Maxi Joker 3 helicopter platform is 71.613 N. The area of the main rotor disk is computed to be $A_d = 2.4885\ m^2$. By substituting the the thrust and the area of the rotor disk in Eq. (38) the hover inflow is $v_{ihover} = 3.41\ m/s$, and normalizing the rotor tip speed (ΩR) yields to $\lambda_{ihover} = 2.533 \times 10^{-4}$. The thrust coefficient at hover can be found as:

$$C_T = \frac{T_{MR}}{\rho(\Omega R)^2 A_d} \tag{39}$$

To find the hover trim of the main rotor collective, we solve Eq. (35) for θ_o and this leads to

$$\theta_0^{hov} = 3\left[\frac{2C_T^{hov}}{a.s} + \frac{\lambda_i^{hov}}{2}\right] = 0.0713 \text{ rad } (4.086°) \tag{40}$$

The main rotor collective varies with a range from -3° to +10 with no collective input is applied and that generates the trim main rotor collective of 4.086° .

4.6.2 Tail Rotor Thrust and Inflow Equations

The full tail rotor thrust is given by:

$$T_{TR} = \left[\delta_{ped}\left(\frac{1}{3} + \frac{\mu_{tr}^2}{2}\right) - \left(\frac{\mu_{ztr} + \lambda_i}{2}\right)\right]\frac{a.s}{2}\rho(\Omega_{tr}R_{tr})^2 A_{dtr} \qquad (41)$$

As increasing the rotor collective command δ_{ped} the tail rotor inflow velocity increases and this yield to an increment in the tail rotor thrust. The tail rotor thrust equation is solved iteratively until the tail rotor inflow is converged T_{TR} using the same iteration technique. In [61], the rotor inflow is normalized and given by the equation below

$$\mu_{ztr} = \frac{v_{tr}}{\Omega_{tr}R_{tr}} \qquad (42)$$

The tail rotor hub speed in the y direction is mentioned in [61] as follows:

$$v_{tr} = v_a - l_{tr}r - h_{tr}|p| \qquad (43)$$

where v_a is the resultant airspeed of the vehicle in the y direction subtracted for wind air speed.

The tail rotor hub airspeed in the z direction is given as:

$$w_{tr} = w_a - l_{tr}q - K_\lambda V_{imr} \qquad (44)$$

The advance ratio of the tail rotor is given in [43] as:

$$\mu_{tr} = \frac{\sqrt{u_{tr}^2 + w_{tr}^2}}{\Omega_{tr}R_{tr}} \qquad (45)$$

From Eq. (30) the torque generated by the main rotor is $N_{MR} = 7.9331 N.m$. Now, to find the tail rotor force needed to cancel the main rotor torque effect, we divide N_{MR} by the tail arm $l_{tr} = 1.068\ m$. This gives:

$$f_{y,TR(hover)} = \ 7.4279 \, N$$

This force needs to be positive all the time to encounter the positive yaw produced by the main rotor.

$$f_{y,TR} = \rho(\Omega_{tr}R_{tr})^2(\pi R_{tr}^2)C_{T(tr)}f_T \tag{46}$$

where the coefficient of the tail rotor thrust is:

$$C_{T(tr)} = \frac{T_{TR}}{\rho(\Omega_{tr}R_{tr})^2(\pi R_{tr}^2)} \tag{47}$$

$$f_{y,TR} = T_{TR}f_T \tag{48}$$

where f_T is the blockage factor which is ignored for the tractor type of tail rotors, so the equation will be as follows:

$$f_{y,TR} = T_{TR} \tag{49}$$

Solving for the for the thrust coefficient at hover gives:

$$C_{T(tr)}^{hov} = 6.356 \ \times 10^{-3}$$

The tail rotor thrust at hover can be expressed by the relation [61] :$T_{TR} = 2\rho A_{dtr}v_{itr}^2$. and solving for the induced velocity at hover yields to:

$$v_{i(tr)}^{hov} = 6.1382 \, m/s$$

$$\lambda_i^{hov} = \frac{v_{i(tr)}^{hov}}{\Omega_{tr}R_{tr}} = 56.361 \times 10^{-3}$$

And, by solving Eq. (47) the trim for the tail rotor can be found as:

$$\delta_{ped}^{hov} = 3\left[\frac{2C_{T(tr)}^{hov}}{a.s} + \frac{\lambda_i^{hov}}{2}\right] \tag{50}$$

$$\delta_{ped}^{hov} = 0.143\ (8.2°)$$

The tail rotor collective varies with a range of -3° to +10° with no collective input is applied that generates the trim tail rotor collective of 8.2°.

4.6.3 Main rotor Flapping Dynamics

In [64], Mettler derived the main rotor flapping dynamics. The main rotor flapping dynamics were modeled as a second order overdamped response. The same model is used in this study. The longitudinal and lateral flap angle commands, β_{1c}^{c} and β_{1s}^{c} , are proportional to the inputs u_{long} and u_{lat}. The linear flapping dynamics are obtained as:

$$\begin{bmatrix}\dot{\beta}_{1c}\\ \dot{\beta}_{1s}\end{bmatrix} = \begin{bmatrix}-\dfrac{1}{\tau_{fc}} & 0\\ 0 & -\dfrac{1}{\tau_{fs}}\end{bmatrix}\begin{bmatrix}\beta_{1c}\\ \beta_{1s}\end{bmatrix} + \begin{bmatrix}0 & -1 & \dfrac{A_{\delta_{lon}}}{\tau_{fc}} & 0\\ -1 & 0 & 0 & \dfrac{B_{\delta_{lat}}}{\tau_{fs}}\end{bmatrix}\begin{bmatrix}p\\ q\\ \beta_{1c}^{c}\\ \beta_{1s}^{c}\end{bmatrix} \tag{51}$$

The main rotor time constants τ_{fc} and τ_{fs} are, which can be determined in the system identification. At hovering conditions, the value of $\tau_{fc} = 0.113$ sec, $\tau_{fs} = 0.101$ sec, $A_{\delta_{lon}} = 4.2$, and $B_{\delta_{lat}} = 4.2$ [49].

4.7 Actuator Models

In [61][43], the servo was modelled to be:

$$H_{servo}(s) = \frac{s/T_z + 1}{s/T_p + 1}\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{52}$$

where $\omega_n = 36$ rad/sec, and $\zeta = 0.5$, $T_z = 104$ sec, $T_p = 33$sec. The tail servo was estimated by a second order system with the undamped natural frequency of 7 Hz and the 0.6 damping ratio. The rotational movement of the servo is transferred through a special mechanical linkage, which controls the helicopter surfaces.

The swash plate is controlled by three servos which are connected together using a special configuration called 120° cyclic collective pitch mixing (CCPM) servo configuration. The Maxi Joker 3 has three servos controlling the swash plate where they are linked together using a 120° cyclic collective pitch mixing (CCPM) servo configuration (see Figure 21).

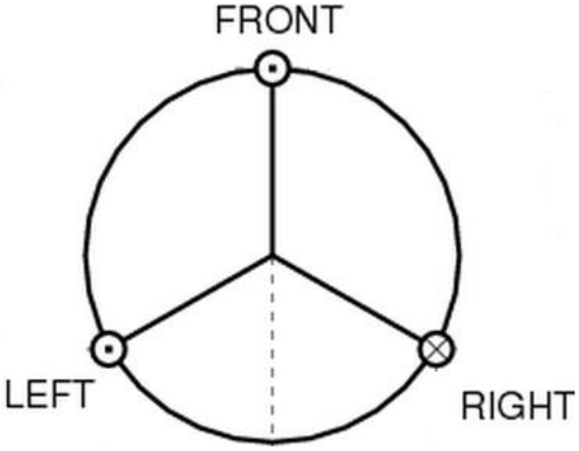

Figure 21: Servos mixing configuration

The servo mixing is represented by using a set of liner equations to calculate the servo positions with respect to the control signal (u_{col}, u_{lat}, u_{long}) as follows [49] :

$$\begin{bmatrix} servo_{front} \\ servo_{right} \\ servo_{left} \end{bmatrix} = \begin{bmatrix} 1 & \cos(120°) & \sin(0°) \\ 1 & \cos(-120°) & \sin(-120°) \\ 1 & \cos(120°) & \sin(120°) \end{bmatrix} \begin{bmatrix} u_{col} \\ u_{lat} \\ u_{long} \end{bmatrix} \tag{53}$$

4.8 Ground effect (GE) model

At altitudes that are comparable with the main rotor diameter, the ground has an effect on the airflow under the helicopter. The ground prevents the airflow from being established uniformly, so the velocity of the induced flow will be reduced. This leads to a reduction in the induced drag and an increment in the vertical lift vector as a result.

$$\frac{T_{IGE}}{T_{OGE}} = \frac{1}{\left(1 - \left(\frac{1}{16}\right)\left(\frac{R}{z}\right)^2 \left(\frac{1}{1 + \left(\frac{Vi}{vi}\right)^2}\right)\right)} \tag{54}$$

Thrust in the ground effect is given by equation Eq. 54, this equation illustrates the ratio between the thrust in both scenarios in ground and out ground effect zones. Where R is the rotor radius, Z is the height that the helicopter is hovering at, Vi represents the rotor speed, and vi is the induced velocity.

At heights of one half the rotor diameter the thrust is increased by 7%. At rotor heights above one rotor diameter, the thrust increase is small and decreases to zero at heights are above 1.25 of the rotor diameter.

Chapter 5: Optical flow sensor modelling

In this chapter, the problem of modeling the optical flow sensor will be addressed. Chapter 5 is divided into five sections. In section 5.1, the theoretical approach is described. The experimental setup for modeling the PX4FLOW optical sensor is illustrated in section 5.2. The optical flow sensor model is described in section 5.3. Section 5.4 describes the sensor model validation process. In section 5.5, we discuss the results of the validated model of the optical flow sensor.

5.1 Theoretical Approach

In this study we assume that the helicopter hovers for a while over a predefined landing spot before it starts its landing process. Our objective is to perform an autonomous precise landing, which guarantees a minimal positional error in the landing process. The proposed landing algorithm utilizes optical flow sensing to provide the translatory velocities of the helicopter in both axes x and y while the helicopter approaches the landing spot.

The helicopter at hover performs a pendulous motion with small velocity, and to simulate this motion, we thought of using a pendulum with a finite length and an optical flow sensor attached to the lower end of the pendulum. As the pendulum moves, there is an accurate encoder attached to the upper end of the pendulum measuring the angular position of the pendulum. On the other hand, the optical flow sensor measures the translatory velocities of the pendulum.

Figure 22 illustrates the geometrical design of the experiment. A special pad is designed and placed on the ground in the field of the PX4FLOW view. A pendulum with l length was manufactured. The optical flow sensor is attached to the lower end of the pendulum and the encoder is attached to the upper end of the pendulum.

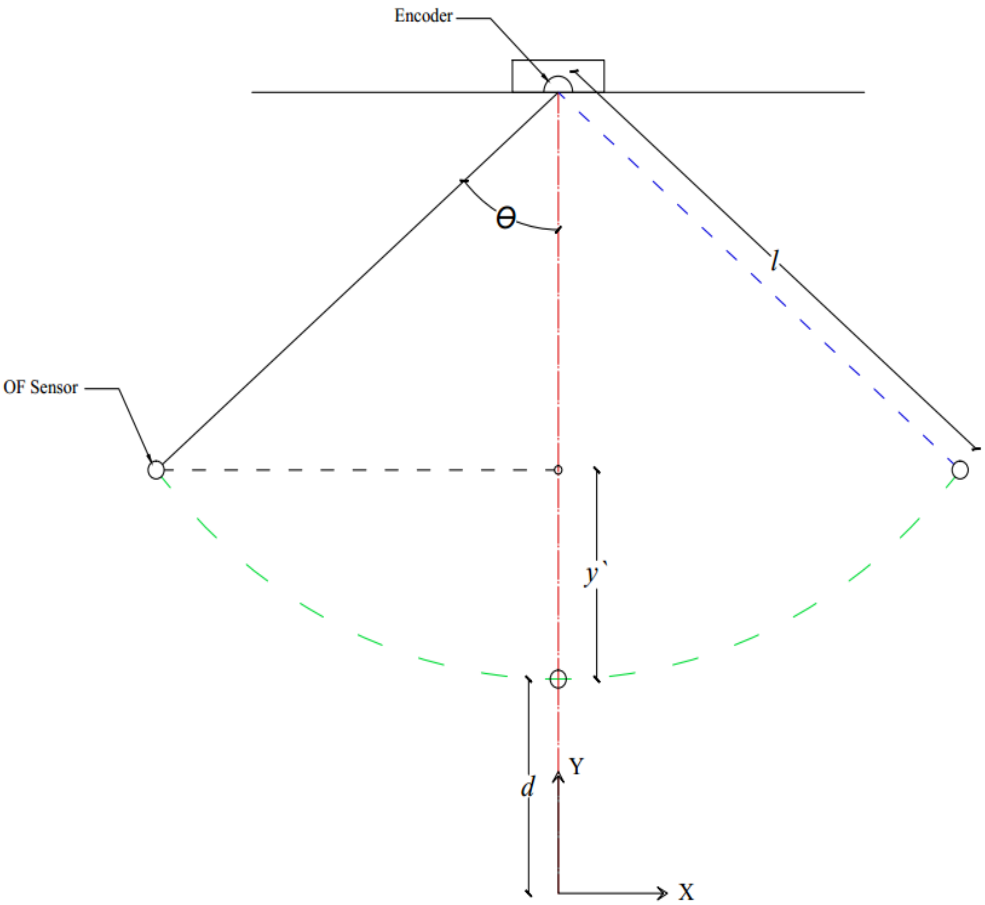

Figure 22: Geometrical design of the modeling experiment

The equations below (55-58) show the mathematical analysis of the pendulum displacements and velocities in both direction.

Equation (55) represents the displacement in the x-axis:

$$x = l.sin\theta \tag{55}$$

where l is the length of the pendulum, and θ is the angle from vertical.

The velocity in the x-axis is:

$$v_x = l.cos\theta.\dot{\theta} \tag{56}$$

Meanwhile, the displacement in the y-direction is:

$$y = d + l - l.cos\theta \tag{57}$$

where d is the distance from the lower end of the pendulum (the position of the optical flow sensor) to the ground. The velocity in y-direction is given by:

$$v_y = l.sin\theta.\dot{\theta} \tag{58}$$

Since an accurate encoder with high resolution is used, the measurements coming from the encoder are considered as the truth velocities of the pendulum. Comparing the measurements of the optical flow sensor to the truth velocities coming from the encoder we are able to tell how accurate the optical flow sensor is during the pendulum motion.

5.2 The Experimental Design

The pendulum test stand is designed to perform the experiment of modeling the optical flow sensor as shown in Figure 23. Clearly, the PX4FLOW sensor is attached at the lower end of the pendulum pointing downwards. An incremental encoder with 10000 ppr maximum resolution is attached to the upper end of the pendulum to measure the precise velocity of the pendulum while moving.

A pad of black horizontal lines as shown in Figure 24 is placed underneath the optical flow sensor allowing the sensor to measure the small velocities precisely.

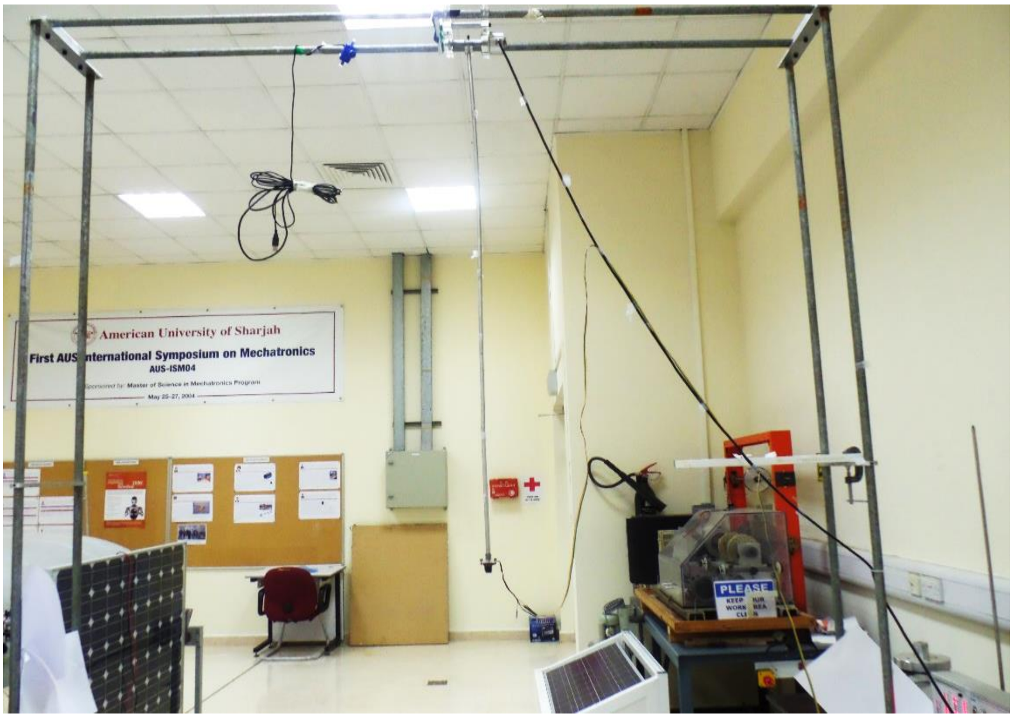
American University of Sharjah
First AUS International Symposium on Mechatronics
AUS-ISM04
May 25-27, 2004
PLEASE

Figure 23: Pendulum test stand.

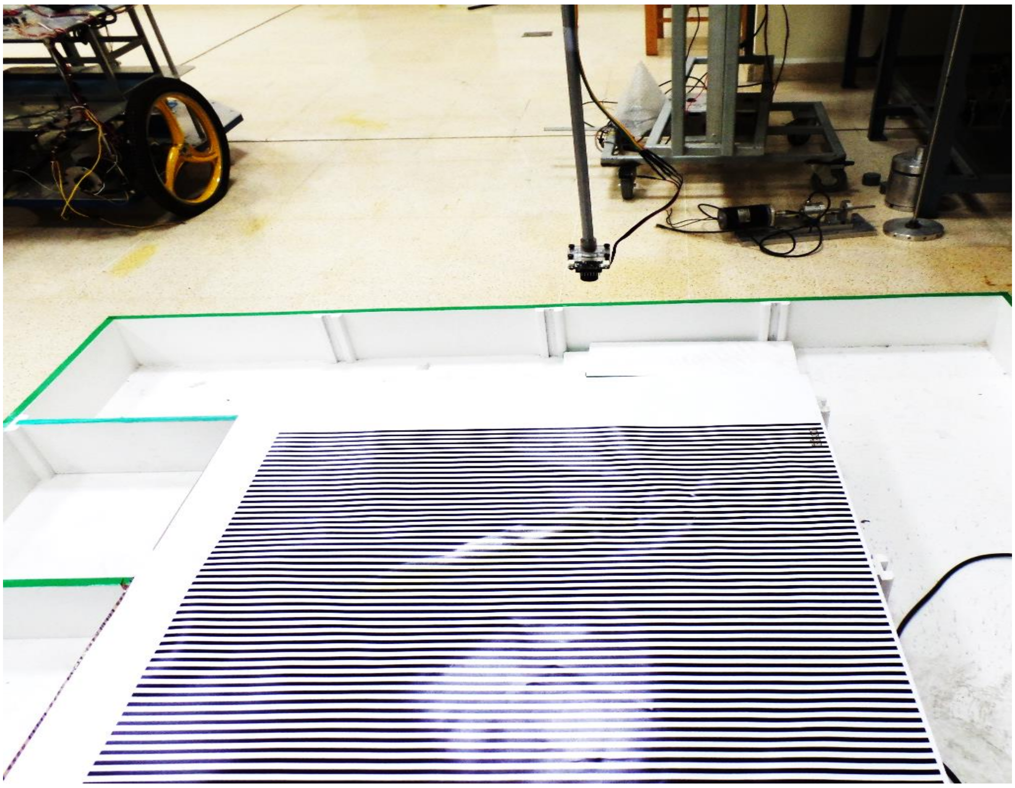

Figure 24: Landing pad

In our experiment, the optical flow sensor was interfaced serially to the microcontroller, and the data were read using a special protocol called the MAVlink protocol at a 115200 baud rate. The optical flow measurements were sent through another serial port to the dspace system. A Baumer encoder [65] was also read serially through the dspace (DS1104 R&D Controller Board) (See Figure 25).

As mentioned earlier, the encoder measures the pendulum's position θ and the velocity $\dot{\theta}$. By knowing the the position and the angular velocity of the encoder, we were able to compute the translatory velocity based on equations (56) and (58). The compensated optical flows in (m/sec) in both directions x and y are measured and sent to the Dspace unit with the Real-Time Interface (RTI). We ran our function models on the DS1104 R&D Controller Board and monitored the sensors readings in a real time environment.

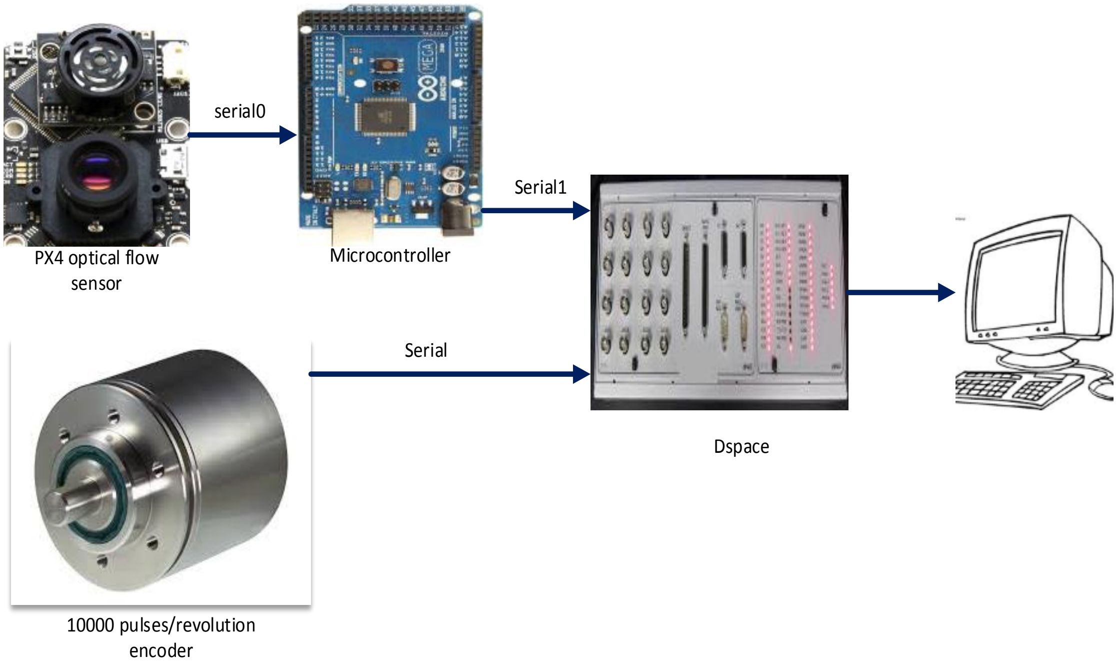

Figure 25: Real time optical flow sensor modeling experiment

In Figure 26 the raw measurements of both sensors are plotted before adding an average filters to smooth the data and reduce the amount of undesired spikes. We know that the optical flow sensor measures the apparent motion of the image, and the apparent motion is affected by varying the light conditions which makes the optical flow sensor readings quite sensitive. Figures 27 and 28 show the light influence on the optical flow sensor measurement. We applied an extra intense light source to test the PX4FLOW sensor at high light conditions. Figure 27 shows how the optical flow sensor becomes unreliable when an extra light source is applied. Contrarily, we turned off a number of the light sources to run the expeiment under low light conditions. As seen in Figure 28, the PX4FLOW performs well under low light.

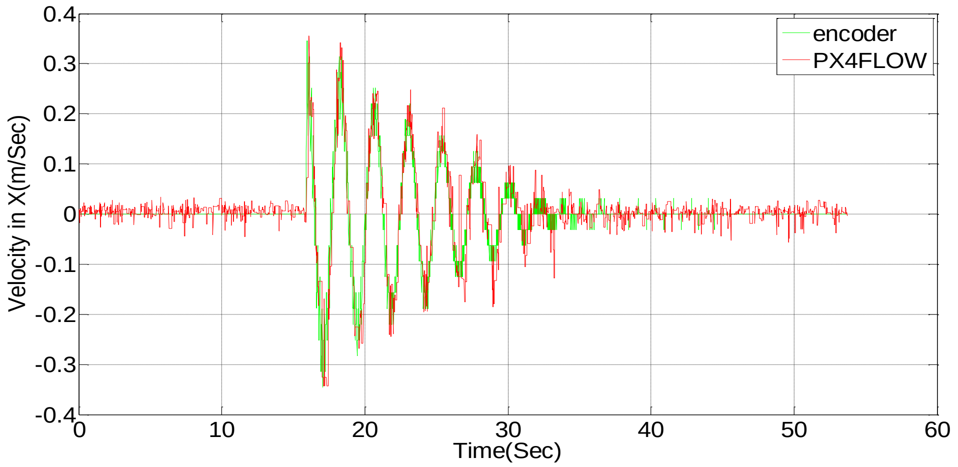

Figure 26: Raw data from both PX4FLOW and the encoder

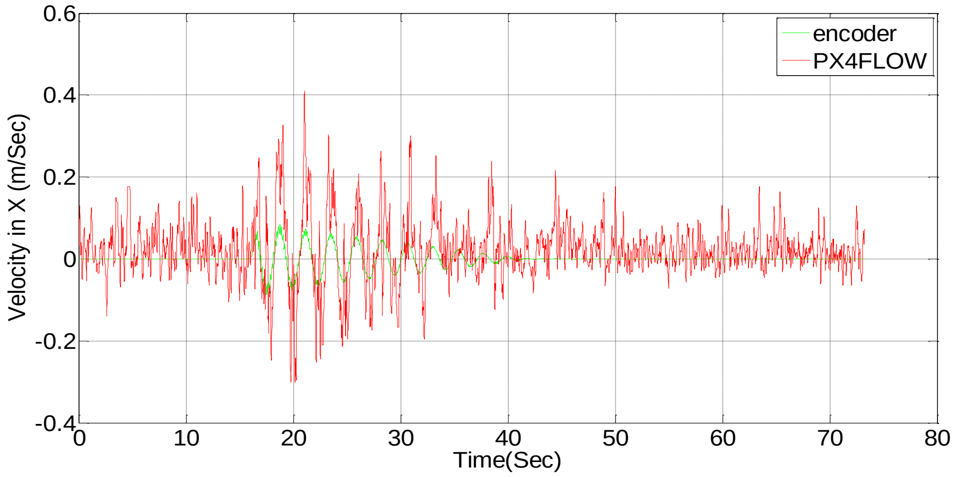

Figure 27: The influence of intense light applied on the pattern

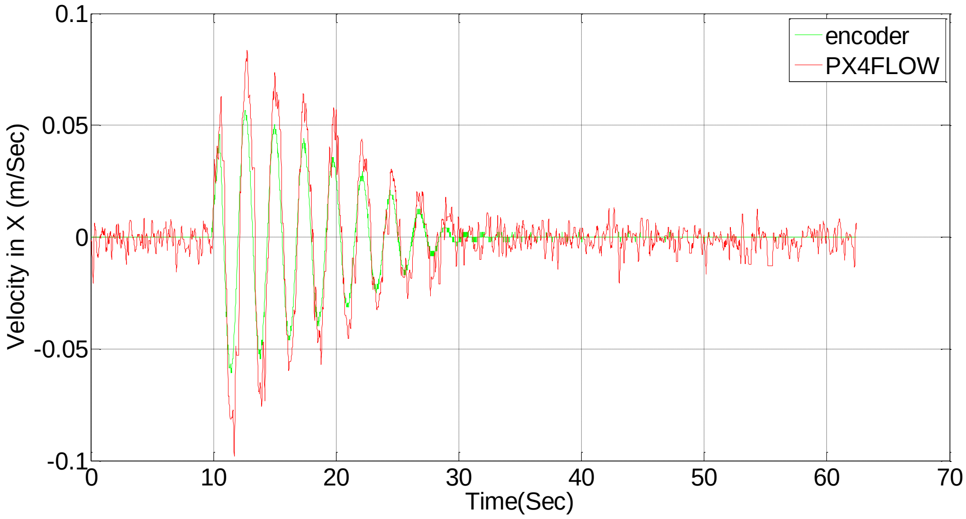

Figure 28: PX4FLOW readings under low light conditions

5.3 Optical Flow Sensor Modeling

In this section, a model for the optical flow sensor is derived. A system identification process was done to obtain the most representative models for the PX4FLOW sensor. The encoder readings of angular velocities were converted to get the translational velocities. The system identification MATLAB toolbox (*ident*) was used, as shown in Figure 29. We assigned the encoder measurements as the input to the optical flow model we wanted to identify and the optical flow sensor data is the output of the model.

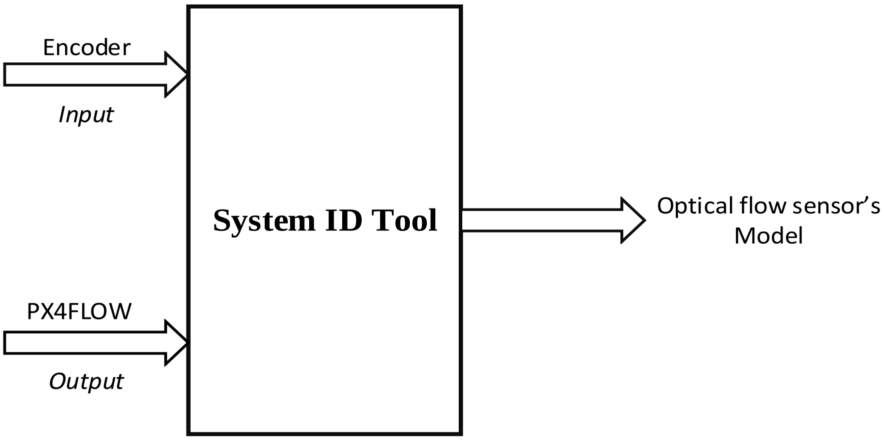

Figure 29: System ID process for modelling the optical flow sensor

In order to obtain an accurate optical flow sensor model and robust while the helicopter approaches the ground, the optical flow sensor is tested at different heights.

The length of the pendulum is adjustable to allow the optical flow sensor to be tested at different altitudes. We tested the optical flow sensor at four different altitudes near the ground to figure out at which altitude the optical flow measurement matches the encoder measurement perfectly. In all conducted tests the procedure was the same. First we made sure that the pattern was being seen clearly by adjusting the optical flow thread, and monitoring the (QGroundControl) software to watch the image being seen by the optical flow sensor. Second, we applied a small manual horizontal force on the rested pendulum which forced the pendulum to move for a certain time before going back to rest. Third, the motion profile of the pendulum was recorded as shown in Figures 30-33. The optical flow sensor was tested at four several heights, at $60, 80, 100$ and 150 cm. The system identification algorithms were applied on the obtained data to generate the possible optical flow sensor models. The four different altitudes tests are as follows:

- At $\boldsymbol{h = 60\ cm.}$

Many tests have been performed on the PX4FLOW sensor at this height, Figure 30 shows significant matching between the PX4FLOW sensor and the encoder. Table 2 shows the transfer functions which can be considered the most representative models at this specific height. According to the System ID tool we can conclude that the obtained models match the optical flow sensor's characteristics for the same inputs at the mentioned height perfectly. As shown in the table above, the obtained models match the optical flow with high percentage of fitting accuracy.

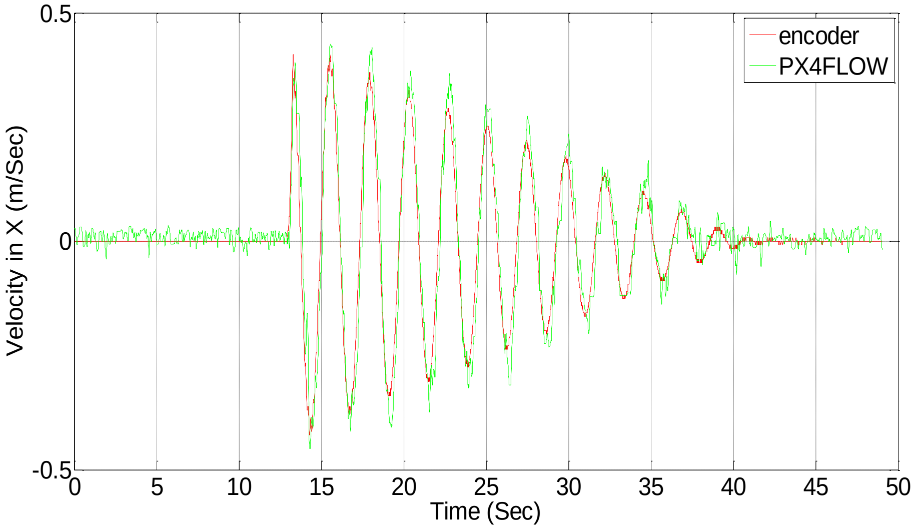

Figure 30: Optical flow vs encoder at 60 cm height

- **At $\boldsymbol{h = 80\ cm}$**

After performing numerous tests at this height, Figure 31 depicts the performance of the PX4FLOW. Obviously, the PX4FLOW measures the translatory velocity correctly at this height. Due to its sensitivity, the optical flow sensor reads a small velocity while the pendulum is at rest. This is because of the apparent motion of the image which could never be equal zero.

Table 3 contains the identified transfer functions and their corresponding percentage of matching. The characteristics of the identified second order models match the optical flow sensor characteristics with about 80% accuracy.

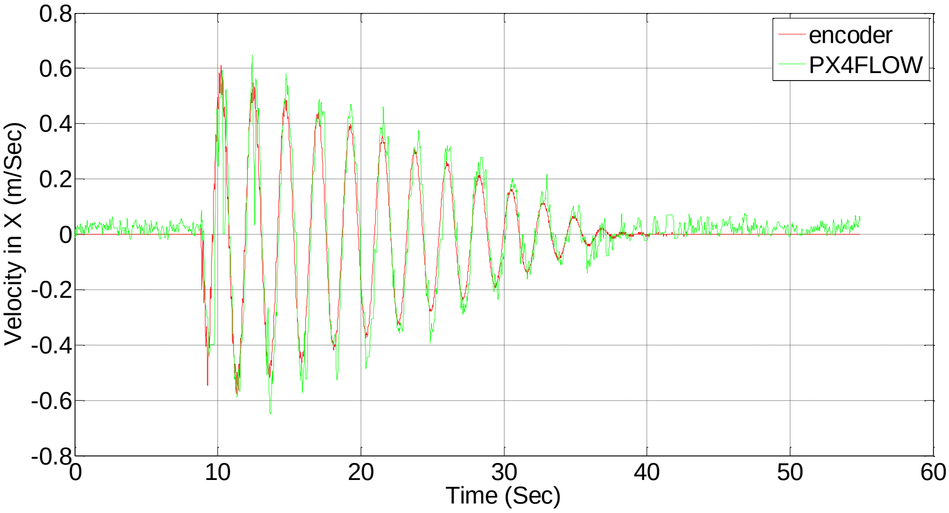

Figure 31: Optical flow vs encoder at 80 cm

- **At $h = 100\ cm$**

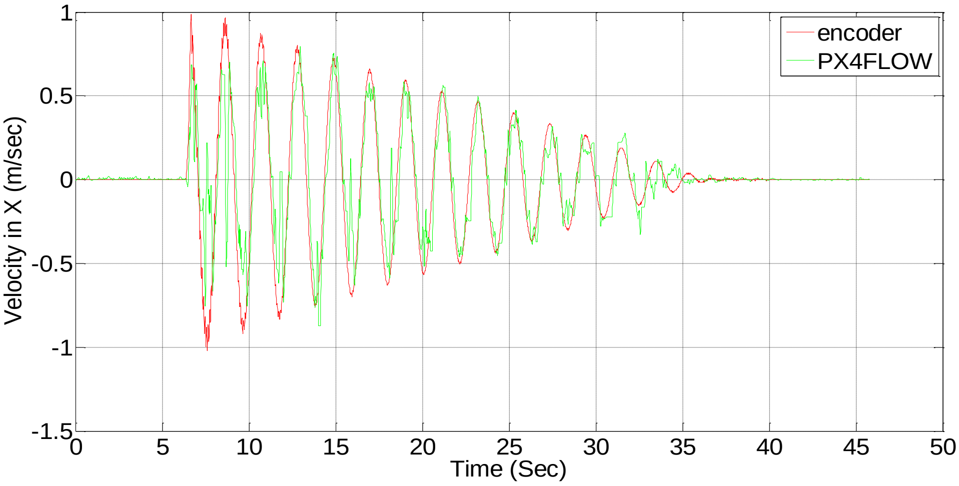

Figure 32: Optical flow vs encoder at 1m height

Since we are getting further from the ground, the optical flow sensor still performs well at 1 m height. Figure 32 describes the translational velocity measured by both sensors. Table 4 lists the acquired models. It is clear that the percentage of fitting is getting decreased as we reduce the pendulum length to increase the height.

- **At $h = 150\ cm$**

As shown in Figure 33, the performance of the PX4FLOW sensor degrades as we move up. The optical flow sensor has failed in recognizing the small velocities of the pendulum which are below 0.4 m/sec this height.

A new set of transfer functions has been obtained, and because of the degradation in the performance of the optical flow sensor at high positions, these transfer functions slightly match the optical flow sensors's output. Table 5 depicts the models and the corresponding matching accuracy.

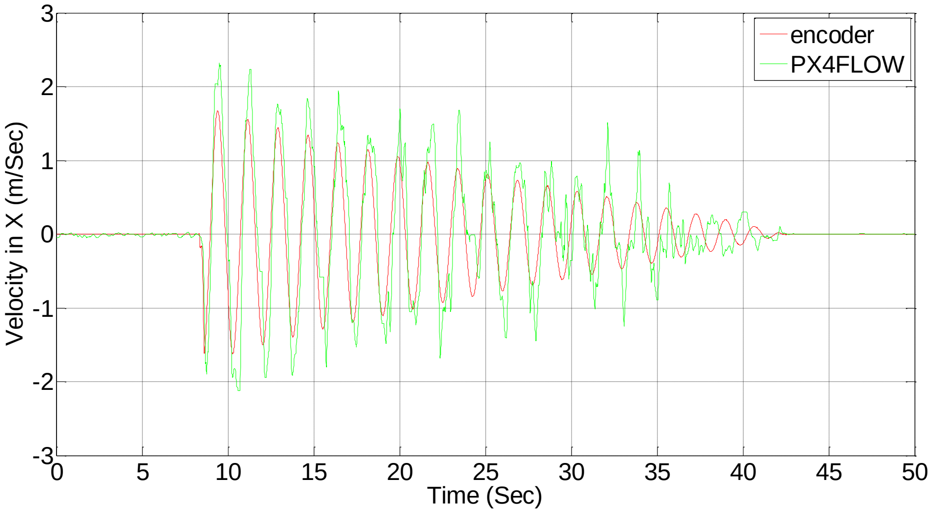

Figure 33: Optical flow vs encoder at 1.5 m height

Table 2: Obtained transfer functions at *60 cm*

Obtained Models	**Best fit**	**Function representation**
(1)	**87.70%**	$\frac{\mathbf{260.30}}{\mathbf{s^2 + 17.73s + 250}}$
(2)	**86.63%**	$\frac{\mathbf{196.90}}{\mathbf{s^2 + 14.83s + 191.90}}$
(3)	**84.22%**	$\frac{\mathbf{355.70}}{\mathbf{s^2 + 36.28\,s + 333.30}}$
(4)	**83.00%**	$\frac{\mathbf{298.60}}{\mathbf{s^2 + 33.24\,s + 277.10}}$

(5)	**84.22%**	$\frac{312.50}{s^2+31.84s+289.90}$
(6)	**83.26%**	$\frac{699.35}{s^2+56.84s+644.40}$
(7)	**84.60%**	$\frac{1127}{s^2+97.873s+1206}$
(8)	**84.00%**	$\frac{721.35}{s^2+57.10s+645}$

Table 3: Transfer functions at *80 cm*

Obtained Model	***Best fit***	***Function representation***
(1)	**79.58%**	$\frac{549.70}{s^2+45.19s+521.97}$
(2)	**78%**	$\frac{309}{s^2+25.68s+297.26}$
(3)	**75%**	$\frac{355.70}{s^2+43.59\,s+475.06}$
(4)	**78%**	$\frac{302}{s^2+33.24\,s+277.1}$
(5)	**73.14%**	$\frac{194}{s^2+27.19s+184.9}$
(6)	**73.60%**	$\frac{694.35}{s^2+56.84s+644.4}$
(7)	**77.56%**	$\frac{373.47}{s^2+36s+325.80}$
(8)	**78.10%**	$\frac{559.69}{s^2+46s+528.9}$

Table 4: Identified transfer functions at *1m*

Obtained Model	***Best fit***	***Function representation***
(1)	**64%**	$\frac{\mathbf{395.22}}{\mathbf{s^2 + 41.9s + 440}}$
(2)	**72%**	$\frac{\mathbf{224.50}}{\mathbf{s^2 + 30.89s + 237.80}}$
(3)	**65.80%**	$\frac{\mathbf{702.30}}{\mathbf{s^2 + 59\,s + 756.28}}$
(4)	**68.58%**	$\frac{\mathbf{465}}{\mathbf{s^2 + 45.7\,s + 522.4}}$
(5)	**67.37%**	$\frac{\mathbf{268.7}}{\mathbf{s^2 + 35s + 307}}$
(6)	**61.59%**	$\frac{\mathbf{1269}}{\mathbf{s^2 + 195.80s + 1359}}$
(7)	**62.51%**	$\frac{\mathbf{751}}{\mathbf{s^2 + 81s + 898.9}}$

Table 5: Identified transfer functions at *1.5m*

Obtained Model	***Best fit***	**Obtained transfer functions**
(1)	***57.67%***	$\frac{\mathbf{1299.40}}{\mathbf{s^2 + 87.34s + 1119.28}}$
(2)	***57.20%***	$\frac{\mathbf{226}}{\mathbf{s^2 + 26.63s + 177.30}}$
(3)	***51%***	$\frac{\mathbf{314.15}}{\mathbf{s^2 + 32.35\,s + 261.66}}$
(4)	***53%***	$\frac{\mathbf{807.16}}{\mathbf{s^2 + 66.50\,s + 619}}$
(5)	***53%***	$\frac{\mathbf{255.50}}{\mathbf{s^2 + 42.90s + 199.80}}$

(6)	***51.44%***	$\frac{\mathbf{209.58}}{\mathbf{s^2 + 38.70s + 169.75}}$
(7)	***57%***	$\frac{\mathbf{205.64}}{\mathbf{s^2 + 26.40s + 174.80}}$

As shown above that all the identified models are second order models. The reason of having a second order models is that the optical flow equation itself is a second order in the special coordinates and the pixel position equations are second order. Moreover, the experimental setup change in the position dynamics of the pixels is a second order.

5.4 Model Validation

After acquiring the models that have the best matching with the data coming from the optical flow sensor, the obtained models are validated at different heights. As observed in section 5.3, the performance of the optical flow sensor differs from one height to another, so the OF models have to be validated at different heights. Several validation tests have been done and the validation results are shown. The validation was done by performing a set of real time experiments at each height, as shown in Figure 34. Three velocities are plotted: the first one comes from the optical flow sensor which is read serially, the second velocity is the translational velocity which is computed from getting the angular position of the encoder, and the third velocity is the OF model velocity. The input to the OF transfer function is the encoder velocity and the output is the OF model velocity.

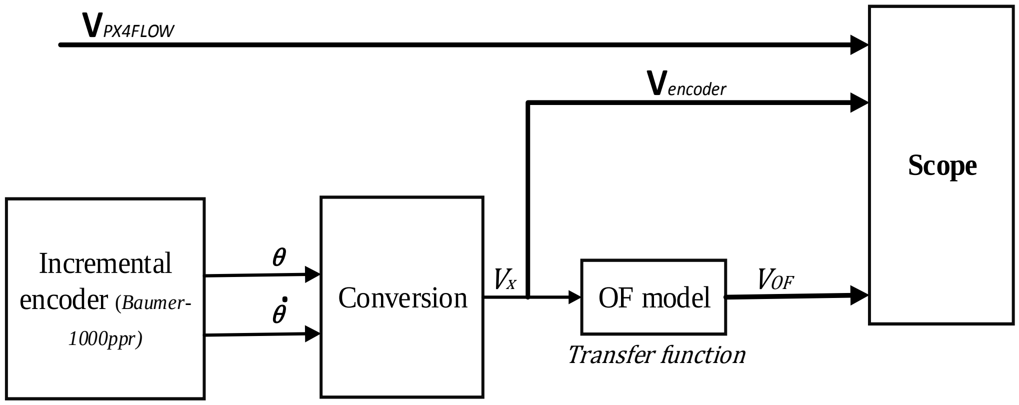

Figure 34: Block diagram of optical flow sensor model validation

A validation experiment was performed to test each model at various heights. Eight transfer functions were engaged in the validation experiment and these functions were validated at four different heights. The model validation results show that the obtained optical flow models are accurate and they can be utilized as OF sensor models. Figures 35- 42 illustrate the validation results for the selected models at 60 cm height.

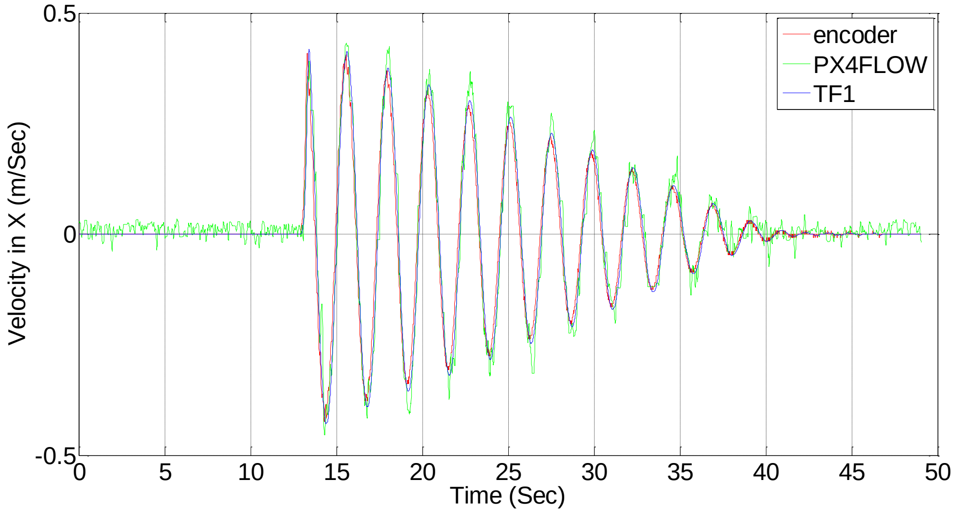

Figure 35: Validation of transfer function 1at 60 cm height

Figure 35 shows the significant matching between the OF sensor reading and the first OF model (TF1) reading. The mean error between the OF reading and transfer function reading is 0.0048 m/sec and the difference between both standard deviations is 0.007.

The second model (TF2) performs well as depicted in Figure 36. The mean error between the OF reading and transfer function reading is 0.0054 and the difference between both standard deviation is 0.0055.

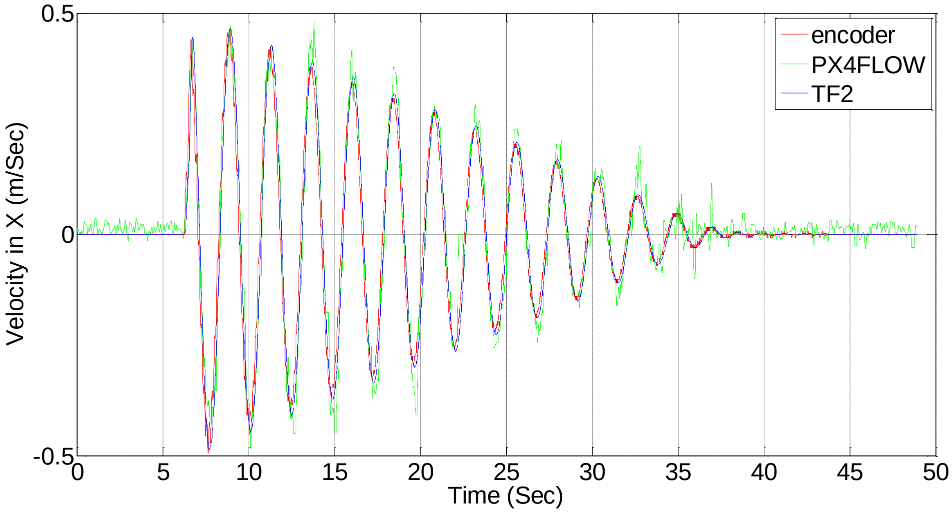

Figure 36: Validation of transfer function 2 at 60 cm height

Figure 37 describes the accuracy of the third model how accurate is the third model (TF3). An excellent matching between the OF sensor measurement and the sensor model reading is shown. The mean error between the OF reading and transfer function reading is *0.0046* m/sec and the standard deviation is *0.0100*.

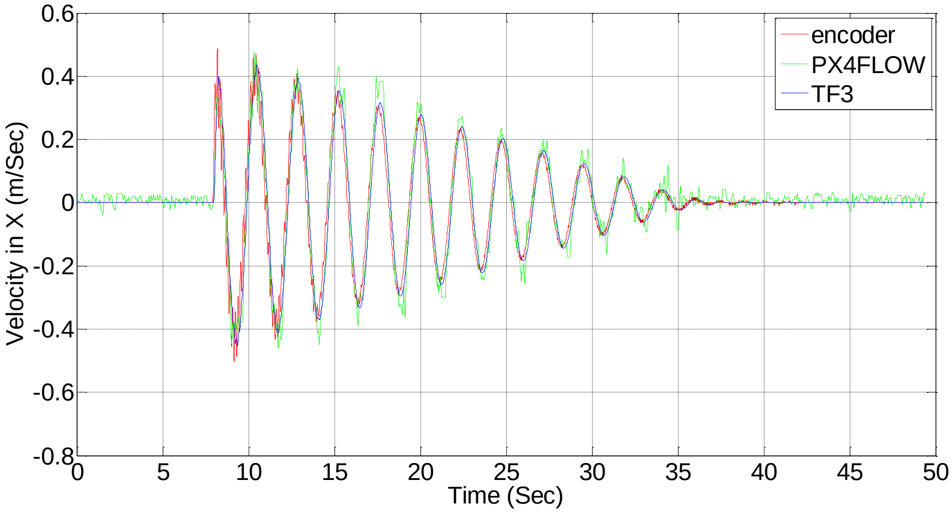

Figure 37: Validation of transfer function 3 at 60 cm height

Figure 38 illustrates significant matching between the OF sensor reading and the fourth OF model (TF4) reading. The mean error between the OF reading and transfer function reading is 0.0048 m/sec and the difference between both standard deviations is 0.007 m/sec.

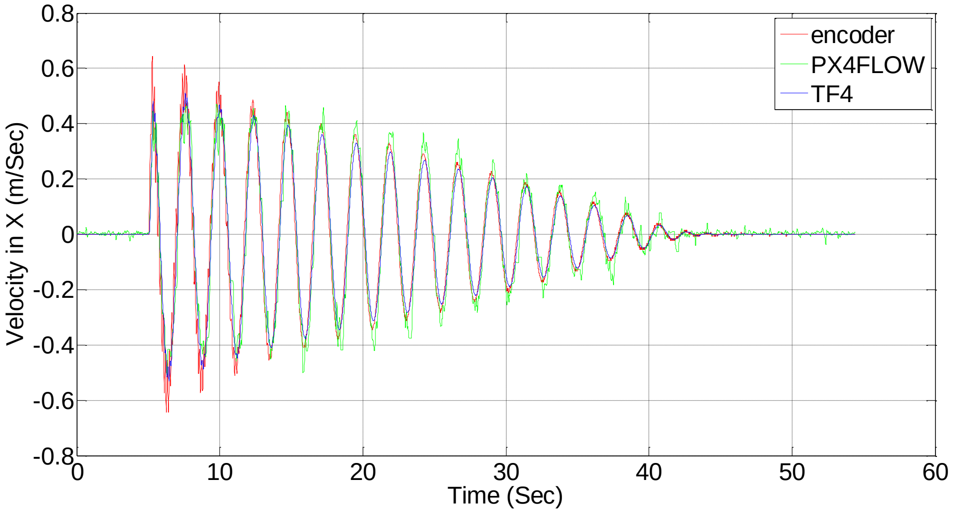

Figure 38: Validation of transfer function 4 at 60 cm height

The performance of the fifth model matches the performance of the real optical flow sensor. Figure 39 describes the response of the fifth model to the encoder input and shows significant matching between the real and the modelled OF sensor measurement. The mean error between the OF reading and the transfer function reading is 0.00042 and the standard deviation is 0.0027.

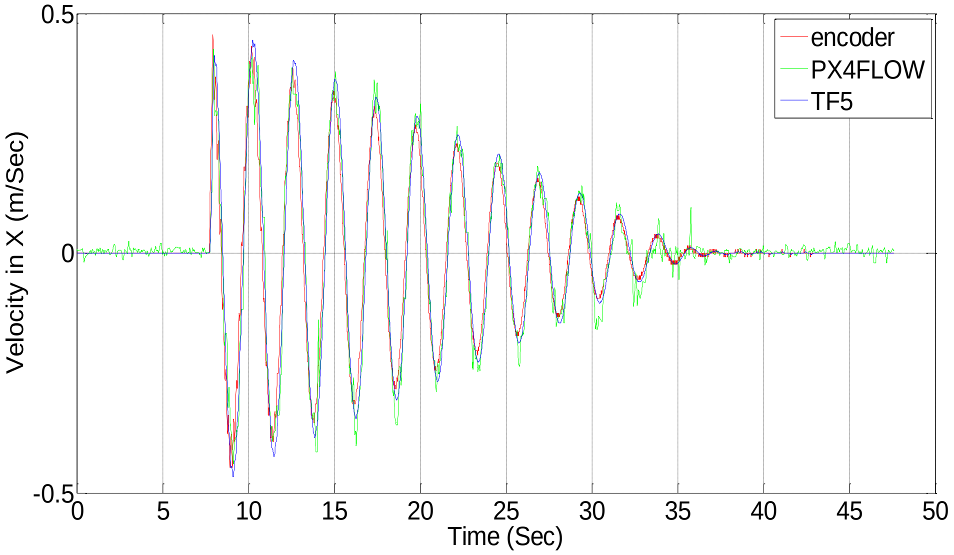

Figure 39: Validation of transfer function 5 at 60 cm height

Figure 40 clarifies the experiment done to validate the sixth model. The mean error between the OF reading and transfer function reading is *0.0029* m/sec and the difference between both standard deviations is *0.0214* m/sec.

The seventh model matches the real sensor reading perfectly as shown in Figure 41. The mean error between the OF reading and the transfer function reading is 0.0018 m/sec and the standard deviation is 0.0011 m/sec.

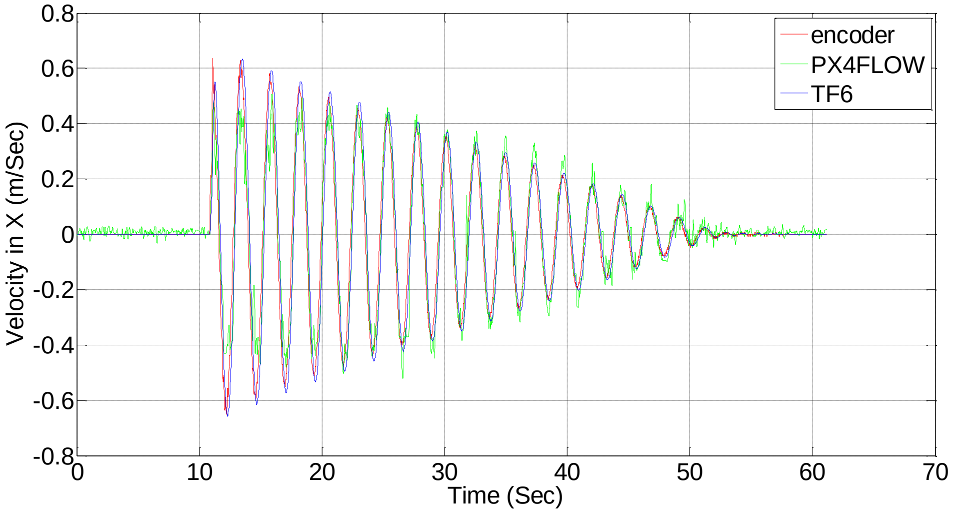

Figure 40: Validation of transfer function 6 at 60 cm height

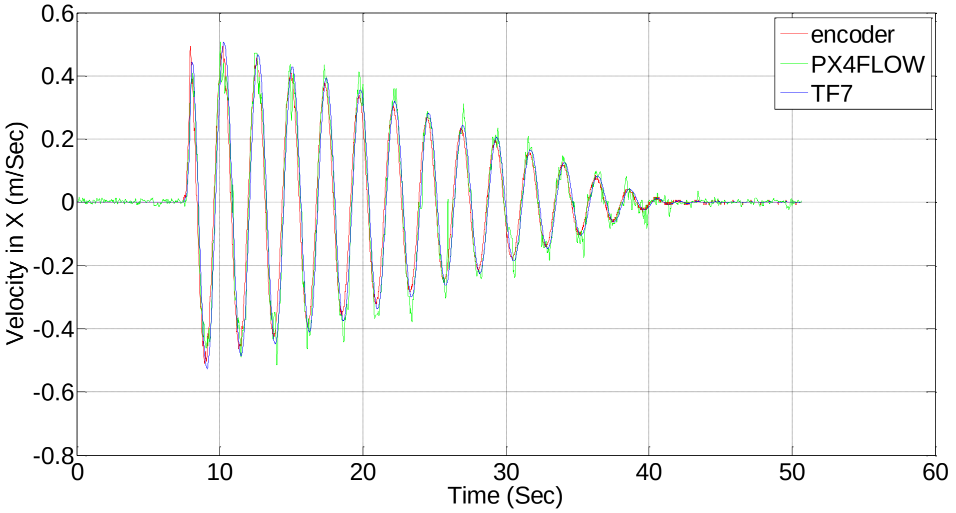

Figure 41: Validation of transfer function 7 at 60 cm height

In Figure 42, the mean error between the OF reading and transfer function reading is *0.0012* and the standard deviation is 0.0057. Hence, as per the previous validation tests, we can conclude that all models showed high accuracy in matching the real signal of the optical flow sensor at 60 cm.

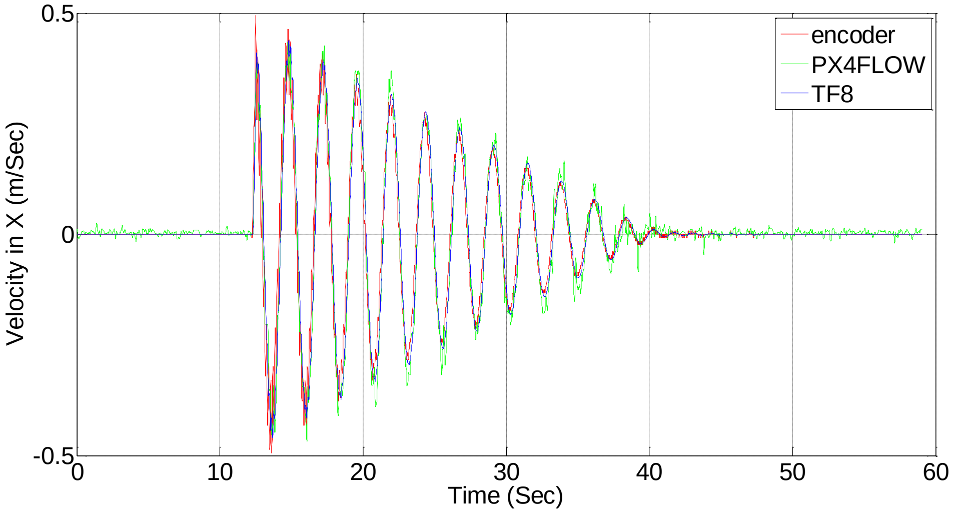

Figure 42: Validation of transfer function 8 at 60 cm height

After all models were validated at *60 cm*, we moved to validate the optical flow sensor at higher heights. We adjusted the pendulum length to keep the distance between the pattern and the PX4FLOW equal to *80 cm*. Figures 43-50 display the validation tests for all models at *80 cm* height.

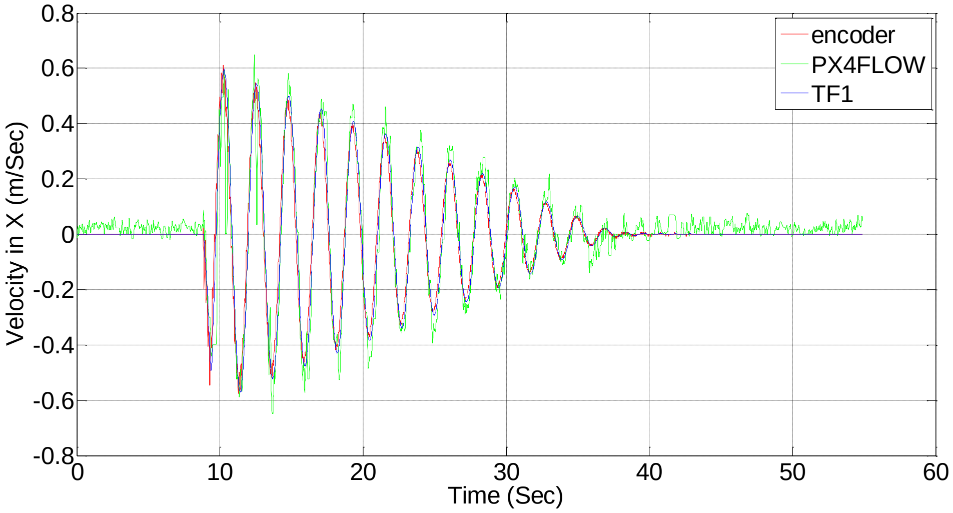

Figure 43: Validation of transfer function 1 at 80 cm height

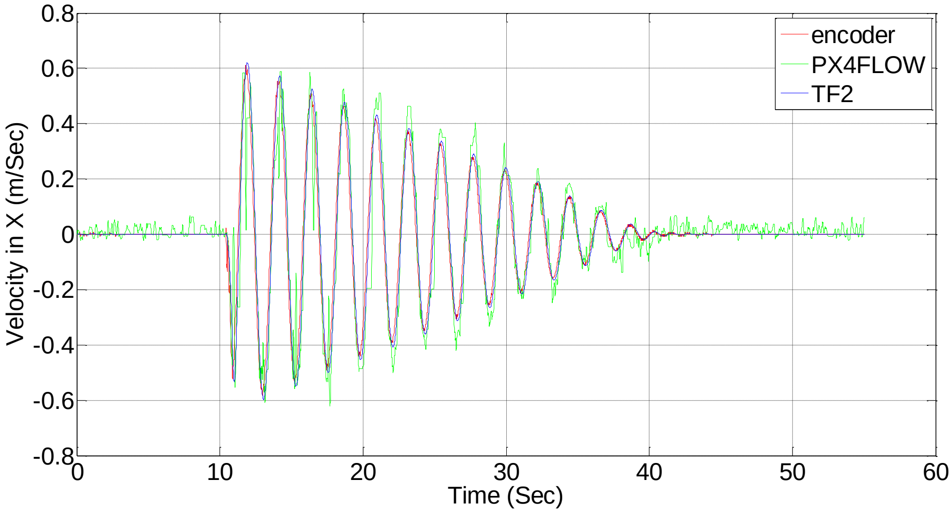

Figure 44: Validation of transfer function 2 at 80 cm height

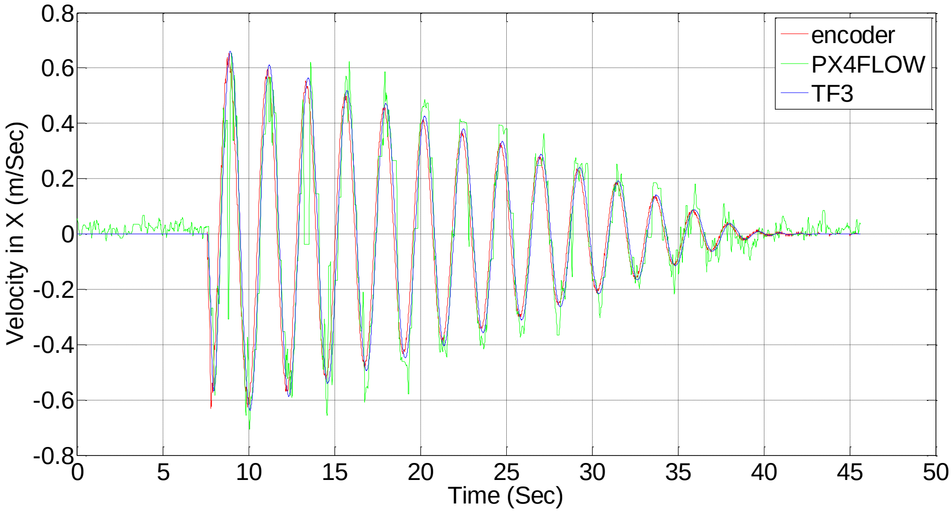

Figure 45: Validation of transfer function 3 at 80cm height

The first three models shown in Figures 43-45 show the great matching between the OF sensor measurement and the measurement of the modelled sensor. The mean errors between the OF reading and transfer function reading are *0.0048* cm, *0.0054* cm, and *0.0046* cm, respectively. And the standard deviations are 0.0070 cm, 0.0055 cm, and 0.0100 cm, respectively.

Figures 46-48 depict very good matching between the models and the real measurements of the OF sensor. The mean errors between the OF reading and transfer function reading are *0.00071 cm*, *0.00042 cm*, and *0.0029 cm.* The standard deviation is 0.0182 *cm*, 0.0027 *cm* and 0.0214 *cm*, respectively.

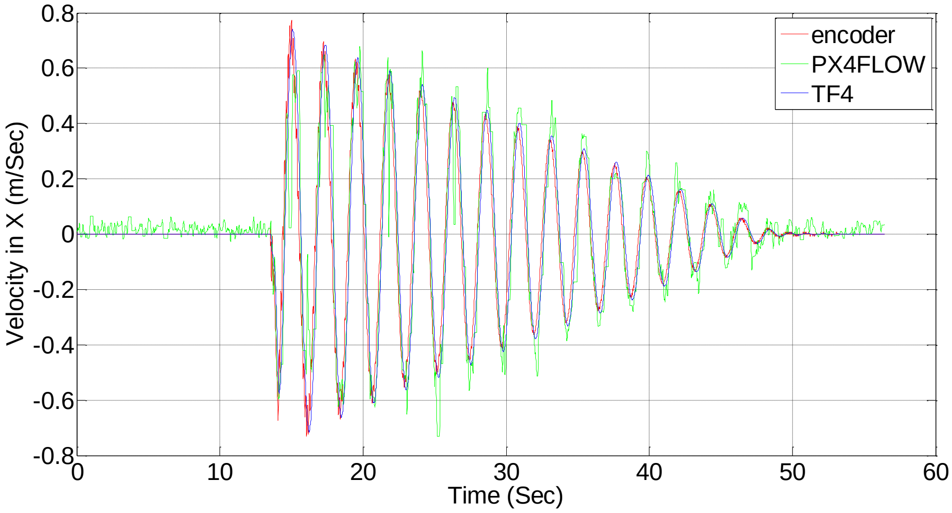

Figure 46: Validation of transfer function 4 at 80cm height

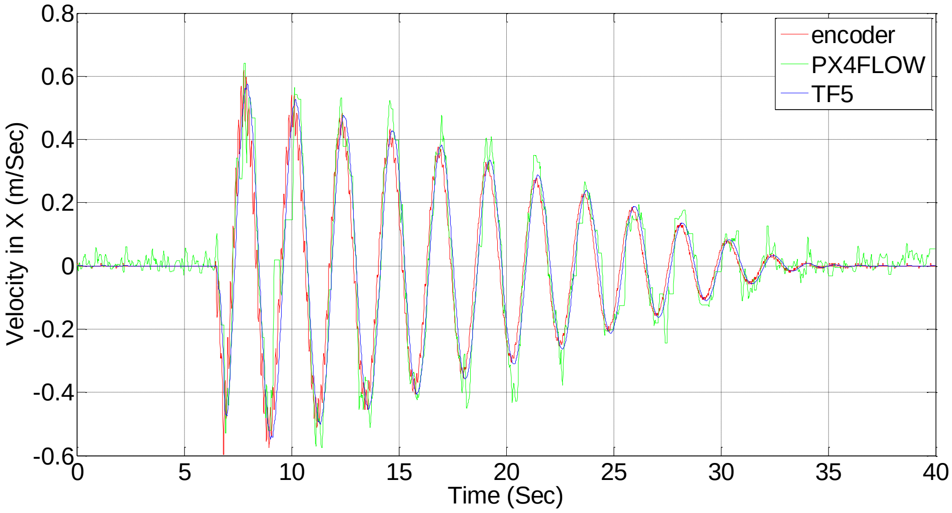

Figure 47: Validation of transfer function 5 at 80cm height

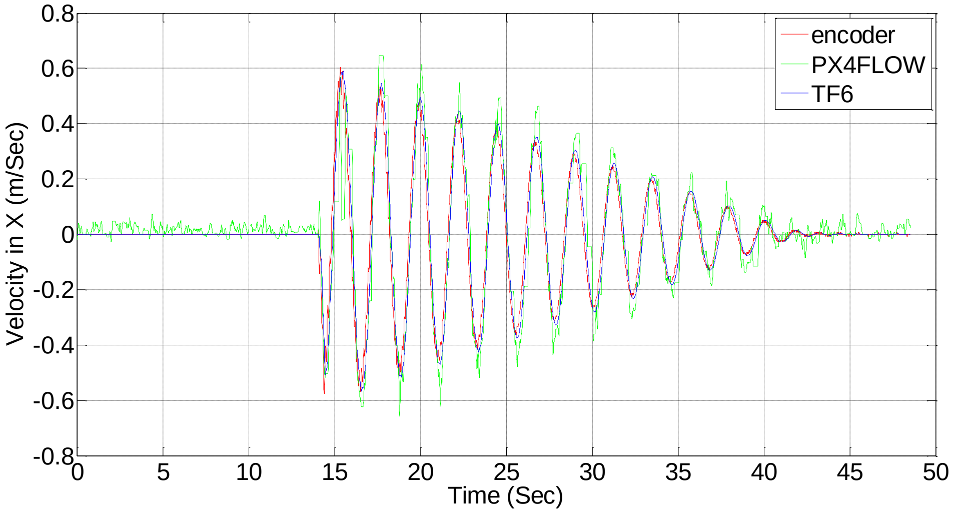

Figure 48: Validation of transfer function 6 at 80 cm height

The performance of the seventh model matches the real OF in Figure 49 where the mean error is about *0.0018* and the difference in both standard deviation is too small *0.0011*. In Figure 50, the eighth transfer function performs well compared to the real sensor. The mean error is $0.0012\ m/sec$ and the difference in both standard deviations is quite small $0.0057m/sec$. Figures from 51-58 display are the validation tests results for all identified models at *1m* height.

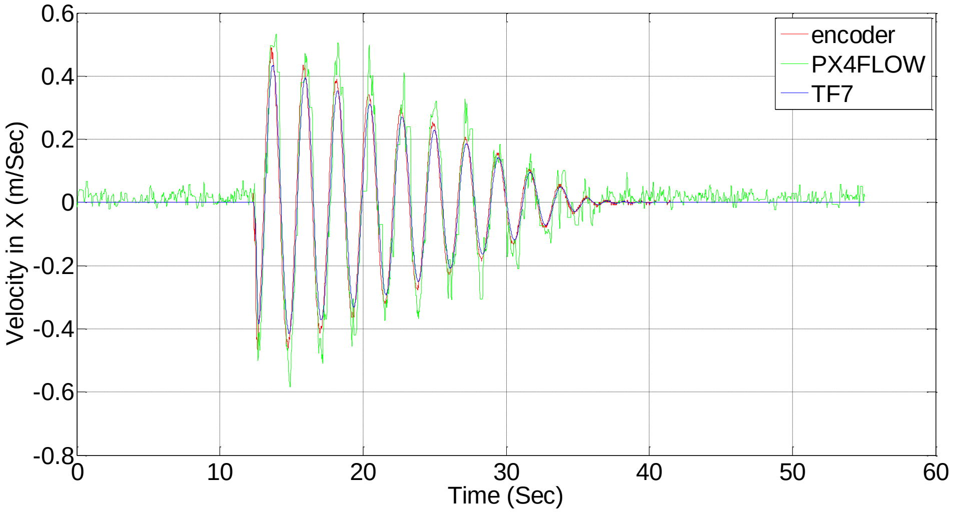

Figure 49: Validation of transfer function 7 at 80 cm height

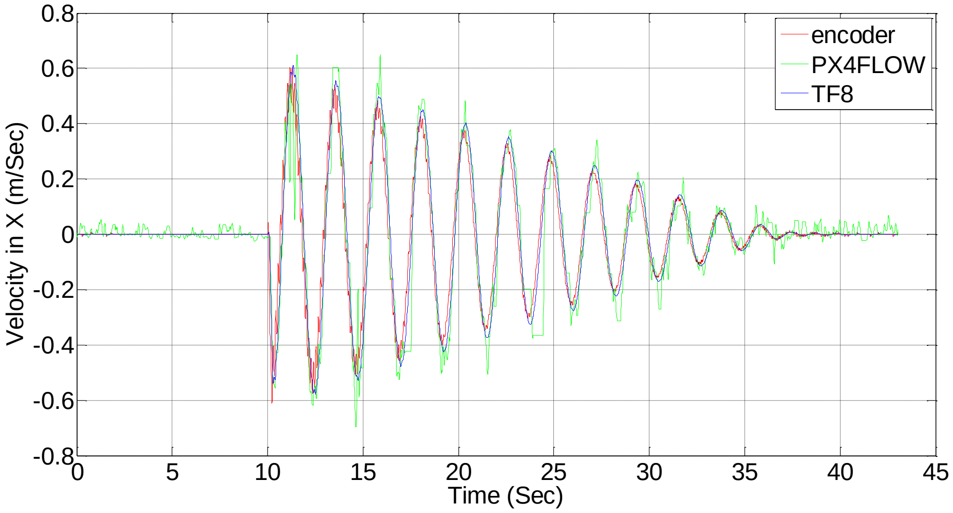

Figure 50: Validation of transfer function 8 at 80cm height

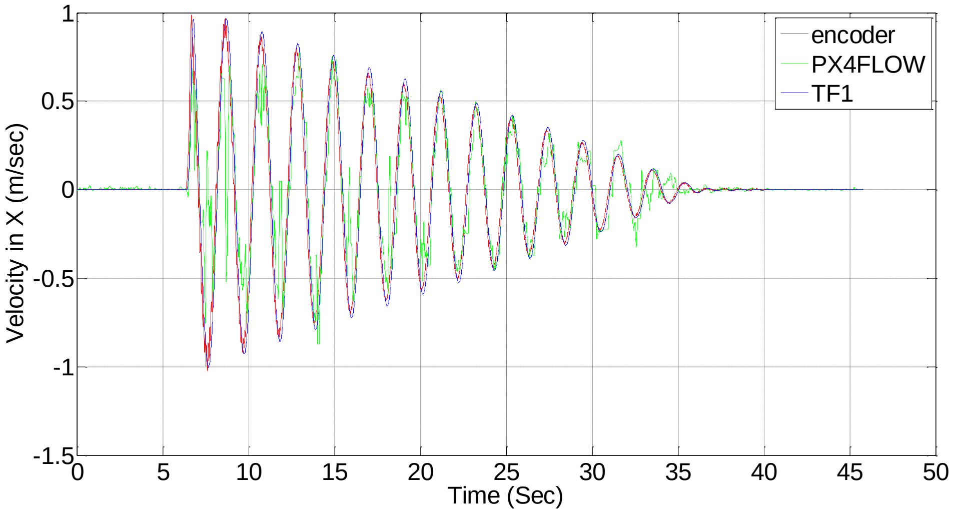

Figure 51: Validation of transfer function 1 at 1m height

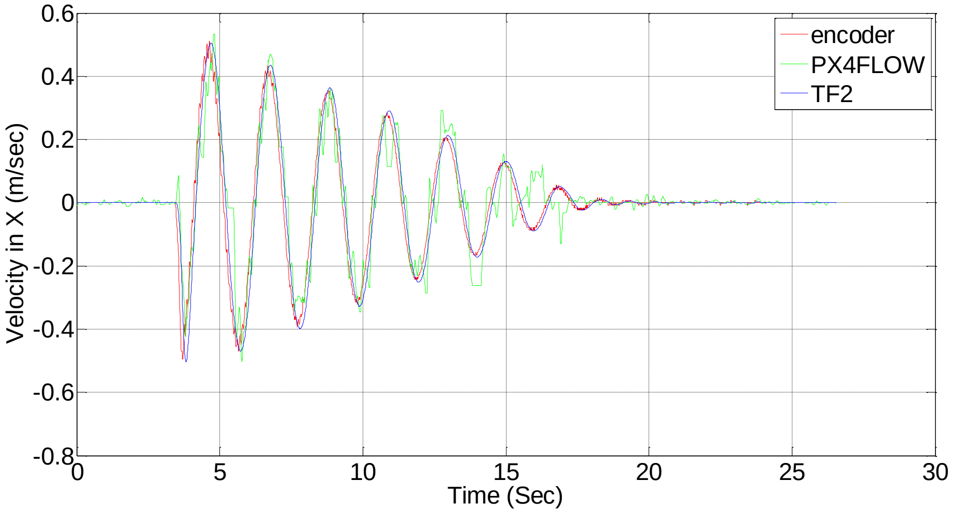

Figure 52: Validation of transfer function 2 at 1m height

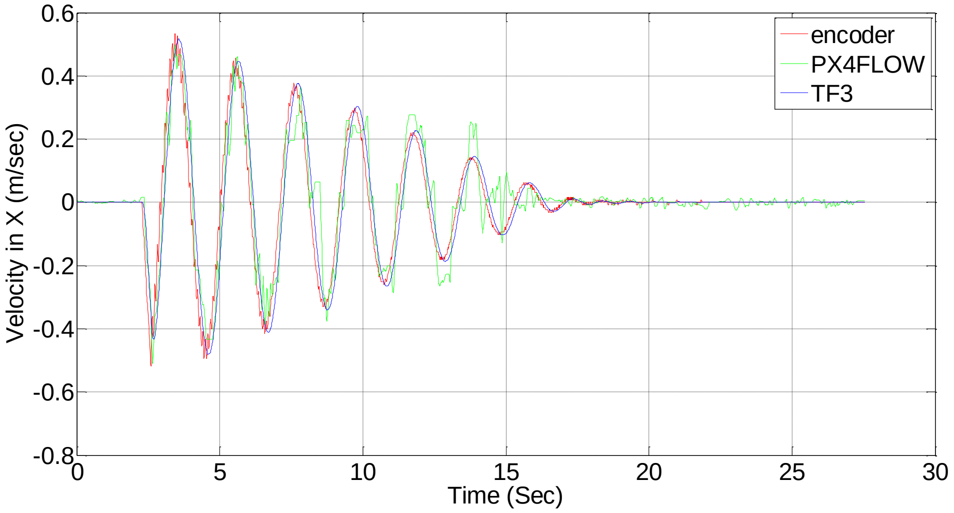

Figure 53: Validation of transfer function 3 at 1m height

The first three models show good performance compared to the real OF sensor. The first model matches the real sensor with a mean error of *0.0105*. The difference between both standard deviations is 0.0822. The second model works well compared with the real sensor: the mean error is *0.0055* m/sec and the difference between both standard deviations is 0.0165. Finally, the third model has a mean error is about *0.0080* m/sec and the difference between both standard deviations is 0.0167.The fourth model (TF4) performs well as depicted in Figure 54. The mean error between the OF reading and transfer function reading is 0.0107 m/sec and the difference between both standard deviations is 0.0474. TF5 as described in 58 matches the PX4FLOW reading with mean error equal 0.0047 and the difference in standard deviation being 0.0563 m/sec.

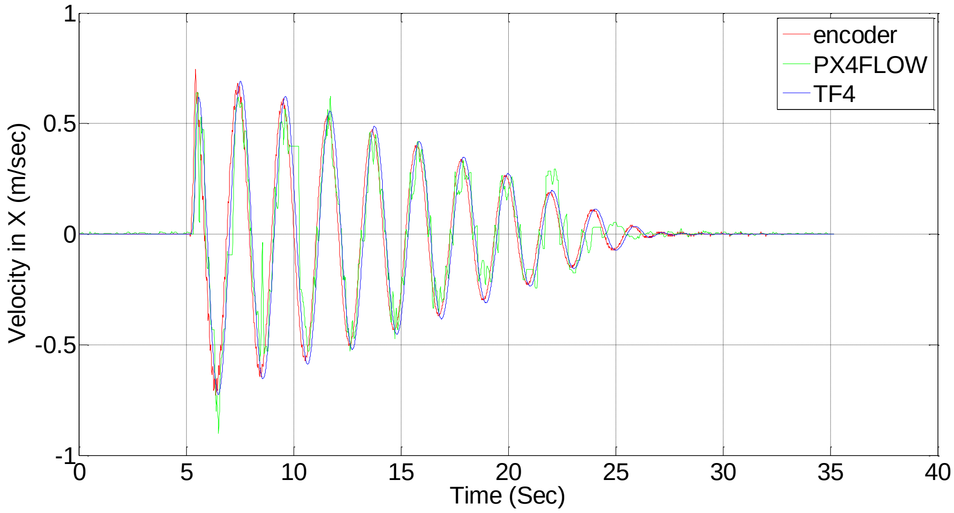

Figure 54: Validation of transfer function 4 at 1m height

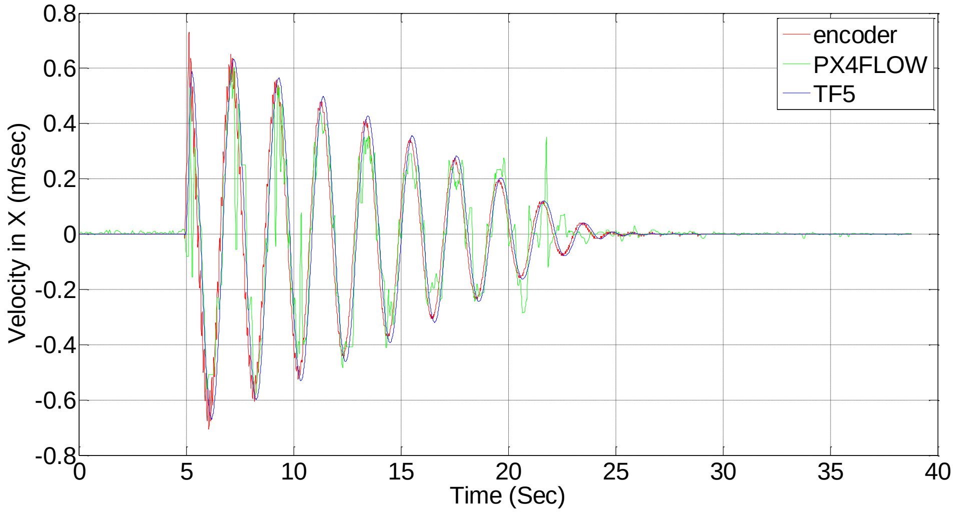

Figure 55: Validation of transfer function 5 at 1m height

TF6 as depicted in Figure 56, has a mean error between the OF reading and transfer function reading of 0.0176. The difference between both standard deviations is 0.0498.

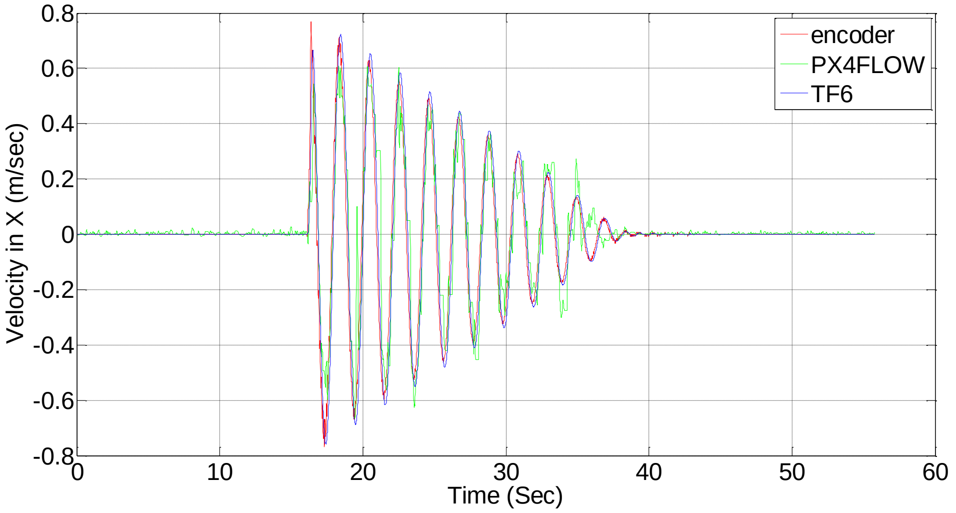

Figure 56: Validation of transfer function 6 at 1m height

TF7 as depicted in Figure 57, has a mean error between the OF reading and transfer function reading of 0.0063. The difference between both standard deviations is 0.0082 m/sec.

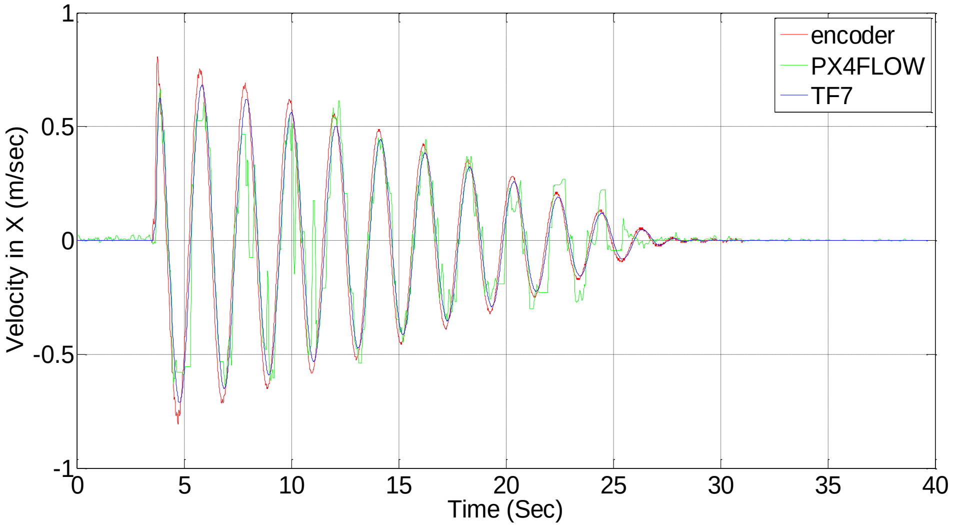

Figure 57: Validation of transfer function 7at 1m height

TF8 as depicted in Figure 58, has a mean error between the OF reading and transfer function reading of 0.0071. The difference between both standard deviations is 0.0499.

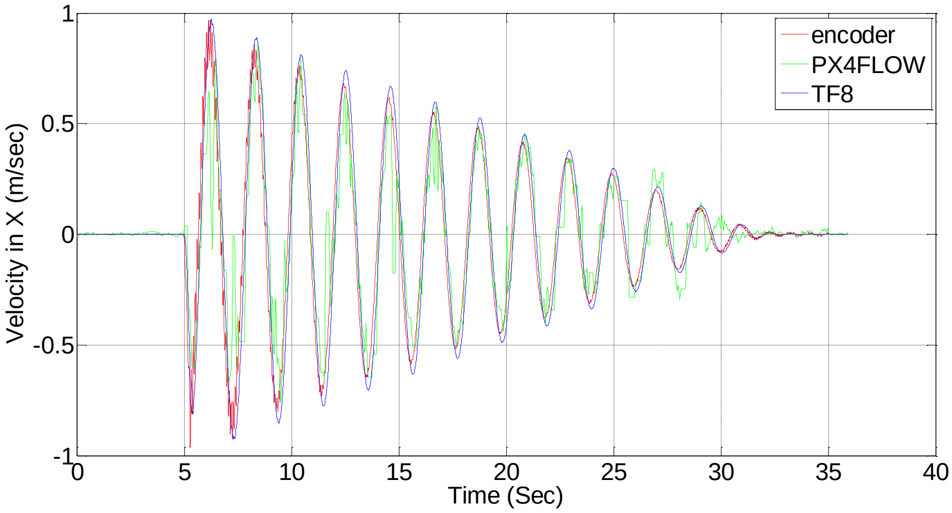

Figure 58: Validation of transfer function 8 at 1m height

At *1.5m* the performance of the optical flow sensor degrades. Figure 59 shows the validation for the first model, where the mean error between the OF reading and transfer function reading is $0.0183\ m/sec$ and the standard deviation is $0.1478\ m/sec$. Figure 60 illustrates the validation for the second model, where the mean error is between the OF reading and transfer function reading $0.0396\ m/sec$ and the standard deviation is $0.1000\ m/sec$. In Figure 61, the third model is validated, and the mean error between the OF reading and transfer function reading is *0.0243* and the standard deviation is *0.1644.*

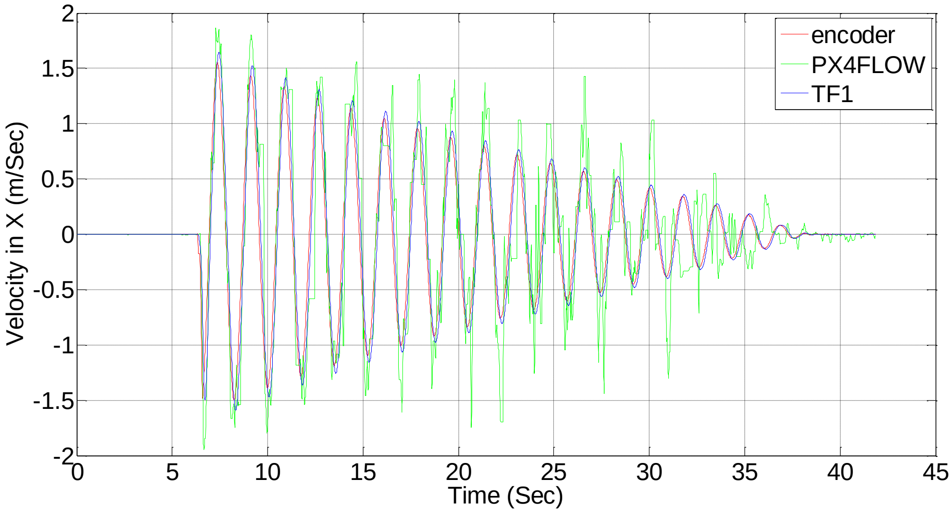

Figure 59: Validation of transfer function 1 at 1.5m height

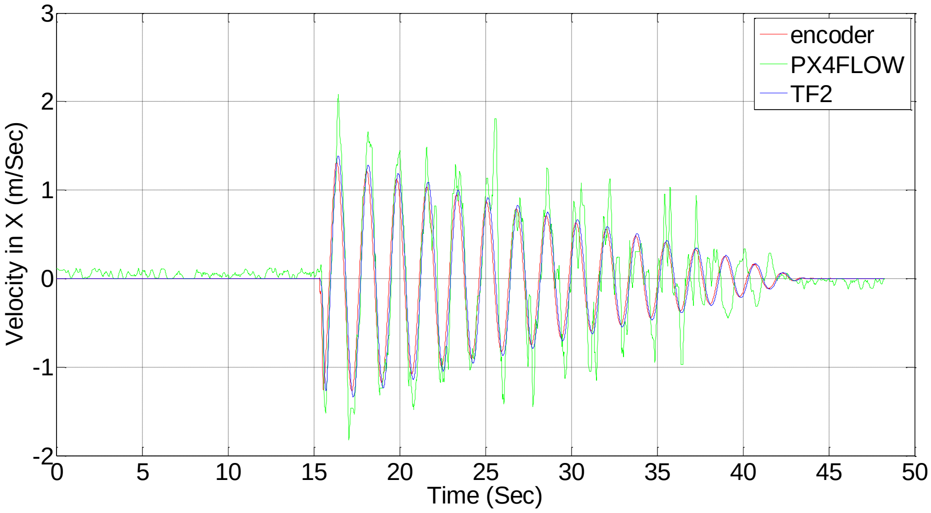

Figure 60: Validation of transfer function 2 at 1.5m height

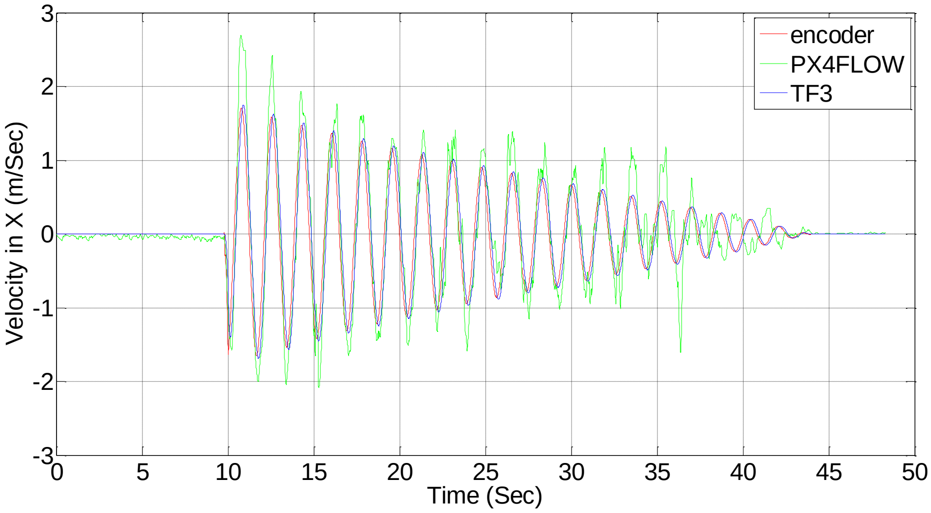

Figure 61: Validation of transfer function 3 at 1.5m height

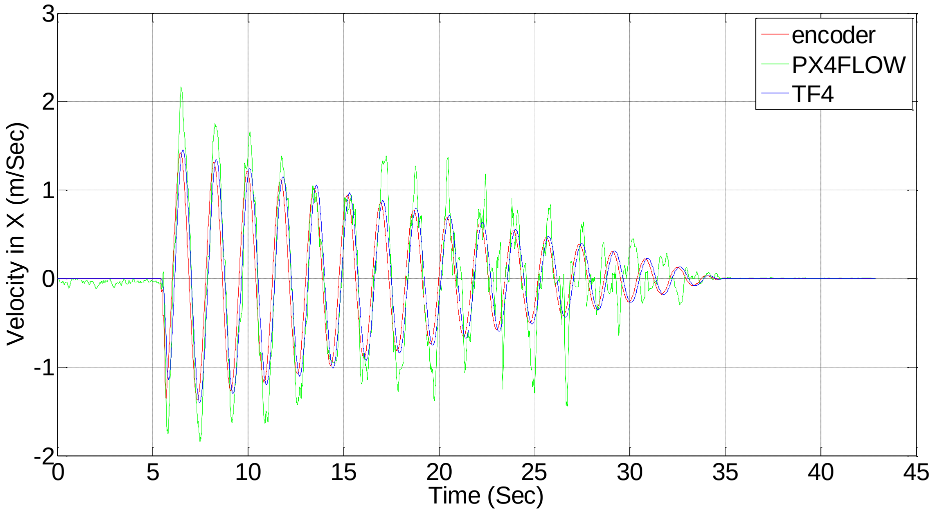

Figure 62: Validation of transfer function 4 at 1.5m height

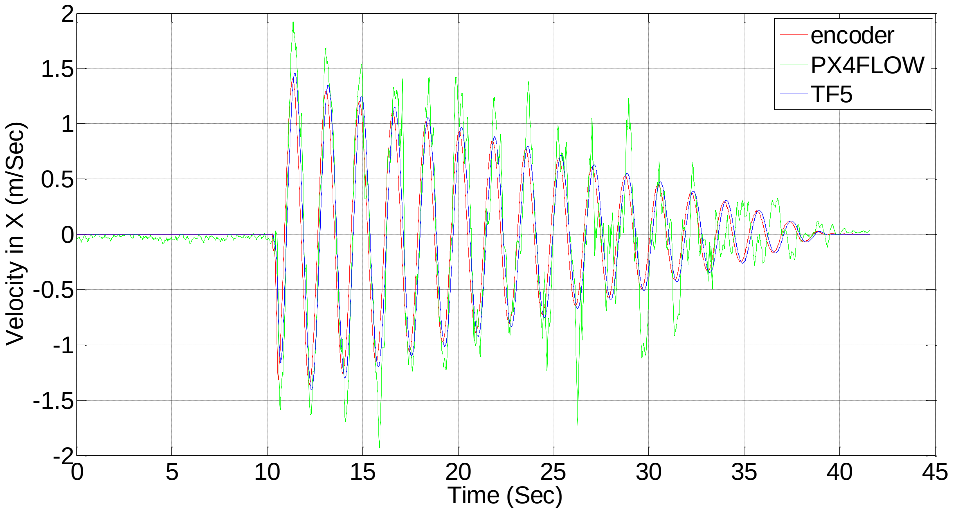

Figure 63: Validation of transfer function 5 at 1.5m height

The fourth and the fifth models are validated at this height with 0.0212 m/sec and 0.0139 m/sec mean error respectively. And they have a standard deviation of around 0.1405 m/sec and 0.1289 m/sec.

Figure 64 describes the validation for the sixth model. The mean error between the OF reading and transfer function reading is *0.0046* and the standard deviation is *0.1255* m/sec. Figure 65 shows the validation for the seventh model. The mean error between the OF reading and transfer function reading is *0.009* and the standard deviation is *0.1456*. In Figure 66, the last model is validated. The mean error between the OF reading and transfer function reading is *0.0197* m/sec and the standard deviation is *0.1059*.

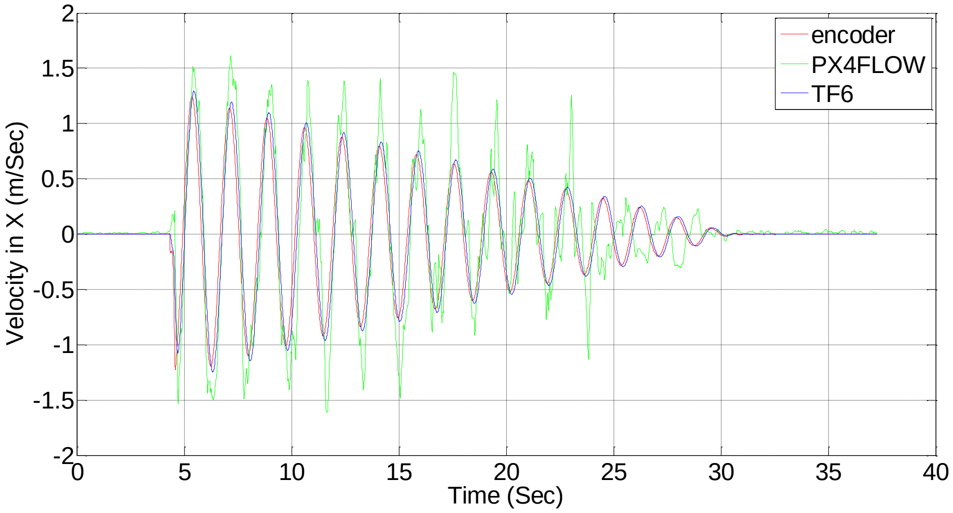

Figure 64: Validation of transfer function 6 at 1.5m height

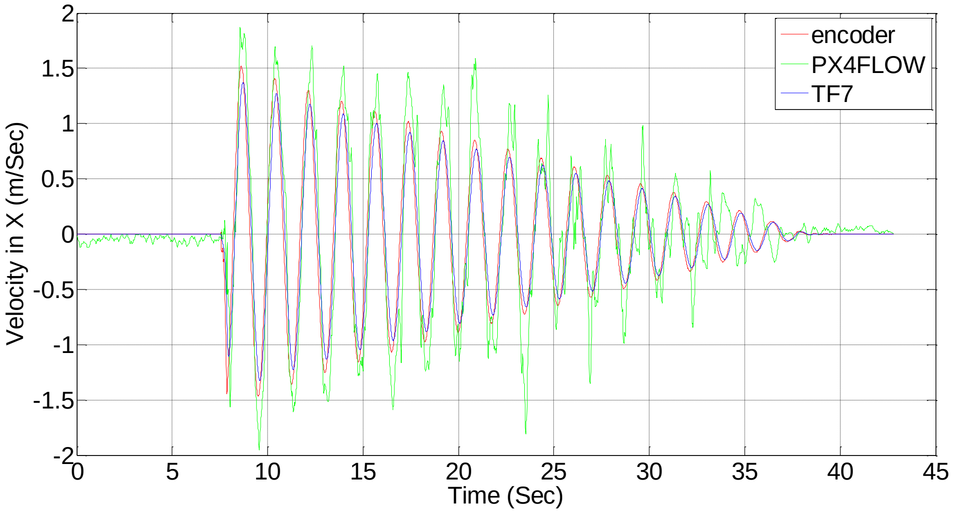

Figure 65: Validation of transfer function 7 at 1.5m height

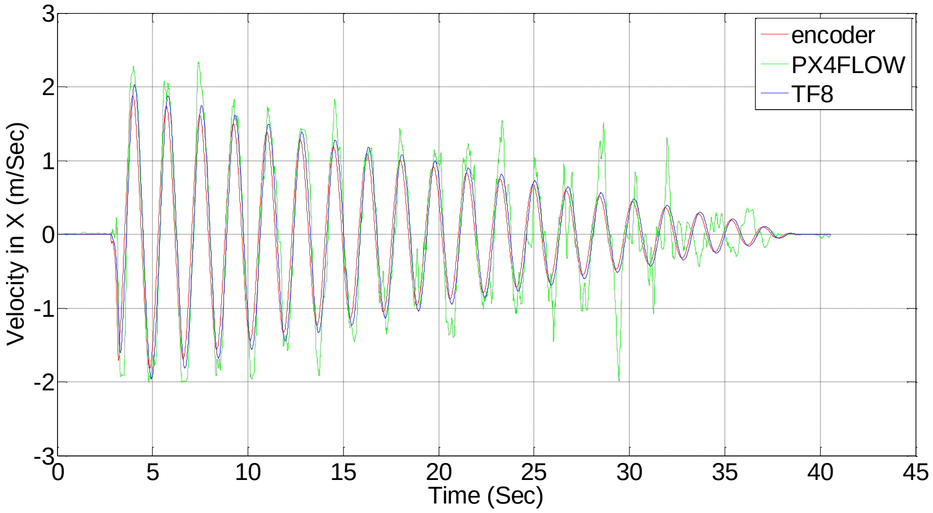

Figure 66: Validation of transfer function 8 at 1.5m height

5.5 Discussion

As explained in the previous sections, the optical flow model has been studied extensively. A robust model for the optical flow sensor has been identified and a precise validation process was performed to validate the optical flow sensor models. The identified models match the real data of the optical flow sensor accurately at heights below *2m*, and the validation results match the real signal of the optical flow sensor perfectly at these heights as well. Table 6 below lists the mean errors and the differences in the standard deviation for each model at 60 cm height. The small errors are shown in Table 6 display how accurate these models are at this specific height. The average mean error is *0.0027 m/Sec.* At 80 cm, Table 7 below illustrates the mean error and the difference in the standard deviation for each model. The errors are larger compared to those at 60 cm. The average mean error is *0.0051 m/Sec.*

Obviously, Table 8 illustrates the mean error and the differences in the standard deviations for each model. The errors are larger compared to those at lower heights. The average mean error is *0.0088 m/Sec.*

Table 6: Mean errors and standard deviations error of optical flow models at 60 cm.

OF Model	*Mean error (ME) (m/Sec)*	*Difference in Standard Deviation (SDE) (m/Sec)*
(1)	*0.0048*	*0.0070*
(2)	*0.0054*	*0.0055*
(3)	*0.0046*	*0.0100*
(4)	*0.00071*	*0.0182*
(5)	*0.00042*	*0.0027*
(6)	*0.0029*	*0.0214*
(7)	*0.0018*	*0.0011*
(8)	*0.0012*	*0.0057*

Table 7: Mean errors and standard deviation error of optical flow models at 80 cm

OF Model	**Mean error (ME) (m/Sec)**	**Difference in Standard Deviation (SDE)(m/Sec)**
(1)	*0.0061*	*0.0218*
(2)	*0.0038*	*0.0178*
(3)	*0.0039*	*0.0246*
(4)	*0.0054*	*0.0087*
(5)	*0.0041*	*0.0076*
(6)	*0.0046*	*0.0186*
(7)	*0.0061*	*0.0427*

(8)	*0.0071*	*0.0136*

Table 8: Mean errors and standard deviations error of optical flow models at 1m

OF Model	***Mean error (ME) (m/Sec)***	***Difference in Standard Deviation (SDE)(m/Sec)***
(1)	*0.0105*	*0.0822*
(2)	*0.0055*	*0.0165*
(3)	*0.0080*	*0.0167*
(4)	*0.0107*	*0.0474*
(5)	*0.0047*	*0.0563*
(6)	0.0176	0.0498
(7)	*0.0063*	*0.0082*
(8)	*0.0071*	*0.0499*

Table 9 shows the mean error and the transfer function for each transfer function at 1.5 m. The average mean error is the largest among all average mean errors. The modelled OF sensor reading differs by 0.0189 m/Sec from the real OF sensor reading.

Table 9: Mean errors and standard deviation errors of optical flow models at 1.5 m

OF Model	***Mean error (ME) (m/Sec)***	***Difference in Standard Deviation (SDE)(m/Sec)***
(1)	*0.0183*	*0.1478*
(2)	*0.0396*	*0.1000*
(3)	*0.0243*	*0.1644*

(4)	*0.0212*	*0.1405*
(5)	*0.0139*	*0.1289*
(6)	*0.0046*	*0.1255*
(7)	*0.0099*	*0.1456*
(8)	*0.0197*	*0.1059*

To conclude, all the obtained models prove their accuracy and worth using as trusted OF models for any applications utilizing optical flow sensing. At low altitudes the OF sensor performs very well and the optical flow sensor is reliable at low altitudes. According to the aforementioned statistics, it is unambiguous that the quality of the OF sensing decreases as the height increases.

Chapter6: Sensor Fusion Algorithm

Chapter 6 illustrates the sensor fusion algorithm and the state estimation needed to perform a precise auto takeoff and landing. In Section 6.1, a shortened background on the data fusion is mentioned. Section 6.2 presents the data fusion algorithm between the already available GPS/INS unit and the optical flow sensor. In Section 6.3 the results of the sensor fusion algorithm are presented. The conclusion is introduced in Section 6.4.

6.1 Background

The already available COTS GPS/INS units provide an estimate of the position with few meters error, via the internal estimator (most likely Kalman estimator). The major source for this estimation error is that the GPS/INS signal is affected by the surrounding environment; like the quality of the GPS receiver, the existence of any object interferes with the satellite signals and the number of the satellites that are involved. This error of estimation would not be sufficient for achieving high accurate missions like auto takeoff and landing for small scale helicopters. Hence, there is a need for extra accurate sensors to provide more accurate position estimation. In this study, a sensor fusion between the optical flow sensor and the GPS/INS is clarified. Since the Kalman estimator showed very accurate estimates [43] the Kalman fusion is utilized in this thesis.

6.2 Fusion Algorithm Design

In this section, a sensor fusion between the optical flow sensor and the GPS/INS solution is explained. The sensor fusion was done to estimate the position and the velocity of the helicopter, in order to obtain more accurate estimates for the position and the velocity. Figure 67 illustrates the fusion process in which two different sensors are involved. The OF sensor measures the velocity, and the position measurements are obtained from the GPS/INS unit. A linear estimator is used to estimate the position and the velocity states of the helicopter.

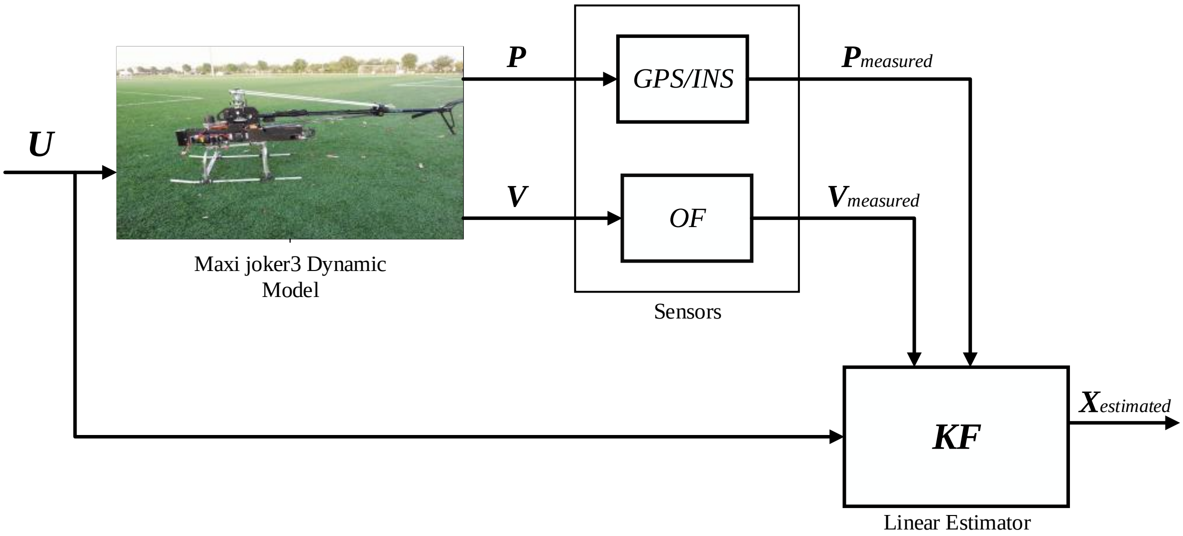

Figure 67: The sensor fusion block diagram

To design a linear estimator, a linear system model is needed. Therefore, a linearization process was performed. The linearization was calculated at a near hovering point where the attitude angles are constant and the attitude rates and body velocities are equal to zero. The linearized model is given in equation (59) as:

$$\begin{aligned} \dot{X} &= AX + BU + w \\ Y &= CX + DU + v \end{aligned} \tag{59}$$

where X is the state vector, U is the input vector, A, B, C, D are the state-space matrices, Y is the output vector, w is the dynamic noise, and v is the measurement noise. The latter two are assumed as Gaussian white noise processes.

The state vector is reduced to have only the position and the velocity states as shown in the equation below:

$$X_{reduced} = \begin{bmatrix} x \\ y \\ z \\ u \\ v \\ w \end{bmatrix} \tag{60}$$

And the input is given as:

$$U = \begin{bmatrix} u_{lon} \\ u_{lat} \\ u_{col} \\ u_{ped} \end{bmatrix} \tag{61}$$

where u_{col}, u_{ped} are the commanding angles from the throttle and the tail servo motors, respectively. The dynamics matrix, $A,$ is computed to be:

$$A_{red} = \begin{bmatrix} 0 & 0 & 0 & 1.0000 & -0.0008 & -0.0016 \\ 0 & 0 & 0 & 0.0007 & 0.9948 & -0.1018 \\ 0 & 0 & 0 & 0.0017 & 0.1018 & 0.9948 \\ 0 & 0 & 0 & -0.0190 & 0.1250 & 0.0699 \\ 0 & 0 & 0 & -0.1256 & -0.0268 & 0.5550 \\ 0 & 0 & 0 & -0.0687 & -0.5378 & -3.2826 \end{bmatrix} \tag{62}$$

The input matrix B is similarly calculated as:

$$B_{red} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0374 & 0 \\ 0 & 0 & 1.5680 & -17.3314 \\ 0 & 0 & -294.6074 & 0 \end{bmatrix} \tag{63}$$

The C matrix is a 6×6 unity diagonal matrix, and the D matrix is a 6×4 zero matrix.

The Kalman filter is a linear state estimator that uses the linear dynamic model described (see equations (59-63)). The Kalman filter uses the system’s control inputs and the data from the measurements to estimate the helicopter’s states [66].

The first step in designing a state estimator is to discretize the continuous time helicopter dynamics. The discretization was performed at the system’s sampling time, which is 0.001 seconds.

The measurement noise covariance matrix R_k is a diagonal matrix given as:

$$R_k = \begin{bmatrix} 0.2236 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2236 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.2236 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{66} \end{bmatrix} \qquad (64)$$

The process noise covariance matrix Q_k is also a diagonal matrix, represented as:

$$Q_k = \begin{bmatrix} 0.0022 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0022 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0022 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & q_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & q_{66} \end{bmatrix} \qquad (65)$$

Choosing the proper covariance matrices is a very difficult process, because it is a trial and error based process. The variances for both matrices were chosen meticulously to give the most accurate estimate. We know that the optical flow sensor is more accurate compared to the GPS/INS, therefore, the variances for the velocity states measured by the OF sensor are less than the position states measured by the GPS/INS.

Four major stages comprise the Kalman estimator design. The first stage includes fetching the model parameters like the system matrices and the covariance matrices and the sampling time. While the second stage concerns assigning an initial values to the estimates and the covariance. The third stage is the prediction stage in which the priori state estimates and priori covariance estimates are computed. The last stage is to update the estimates.

The implemented Kalman filter is illustrated in Figure 68.

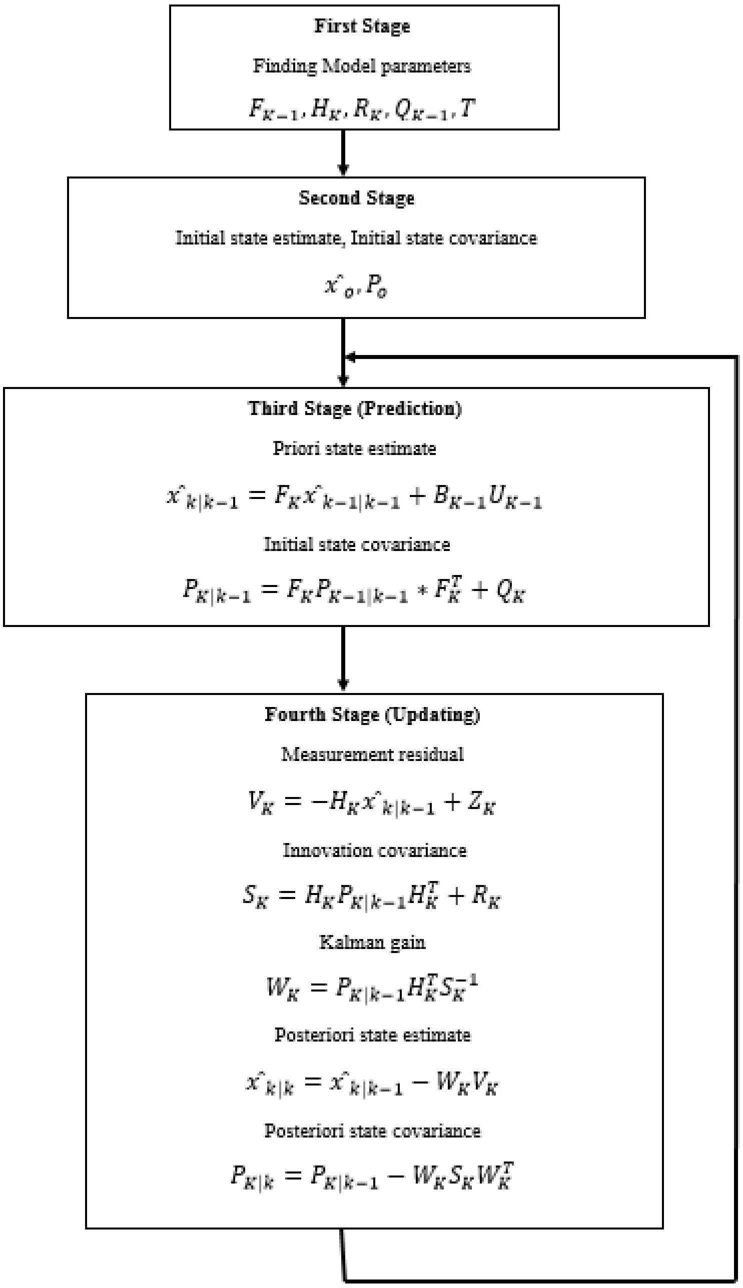
First Stage
Finding Model parameters
$F_{K-1}, H_K, R_K, Q_{K-1}, T$
Second Stage
Initial state estimate, Initial state covariance
$\hat{x}_o, P_o$
Third Stage (Prediction)
Priori state estimate
$\hat{x}_{k|k-1} = F_K \hat{x}_{k-1|k-1} + B_{K-1} U_{K-1}$
Initial state covariance
$P_{K|k-1} = F_K P_{K-1|k-1} * F_K^T + Q_K$
Fourth Stage (Updating)
Measurement residual
$V_K = -H_K \hat{x}_{k|k-1} + Z_K$
Innovation covariance
$S_K = H_K P_{K|k-1} H_K^T + R_K$
Kalman gain
$W_K = P_{K|k-1} H_K^T S_K^{-1}$
Posteriori state estimate
$\hat{x}_{k|k} = \hat{x}_{k|k-1} - W_K V_K$
Posteriori state covariance
$P_{K|k} = P_{K|k-1} - W_K S_K W_K^T$

Figure 68: Flowchart for Kalman filter design [66]

6.3 Results

To test the performance of the proposed sensor fusion method, a simulation environment was used. Simulated data were acquired based on a linear dynamic model for the Maxi Joker 3 helicopter platform. The position is measured by the GPS/INS unit and the velocity states are measured by the OF sensor. The Kalman filter uses the modeled measurements to obtain an estimate of the position and velocity states.

Two tests were made to visualize the estimator performance during the takeoff and landing. The control commands were sent in the body frame expression. The first test is the takeoff test in which we command the helicopter to start climbing until it reaches -3m in the Z-body frame with -1 m/s slope. In the second test, we test the estimator behavior during the landing. In the landing test, we command the helicopter to start descending until it reaches the ground with slope slower than the takeoff a slope which is about 0.3 m/s. The helicopter took almost 10 seconds to reach the ground which is reasonable and safe.

6.3.1 Takeoff Test

The position and the velocity states are estimated during the takeoff test in this section, and the states are estimated accurately with a small amount of error. The position and the velocity are controlled and estimated in the helicopter body frame (see Figure 18), and the Z-axis is assumed to be positive down. The mean error between the Kalman signal and the truth signal is computed for each state.

- **Position estimation results**

Figures 69-71 show the estimation performance for the position states measured by the GPS/INS unit. Although, the GPS/INS signal is noisy and has a lot of distortion, the Kalman estimator succeeds in estimating the position states perfectly. The X position estimation is shown in the Figure 69 below, It can be seen below that the estimator rejects the undesired noise and estimate the X position with small mean error about *0.0133 m* between the estimated signal and the truth signal.

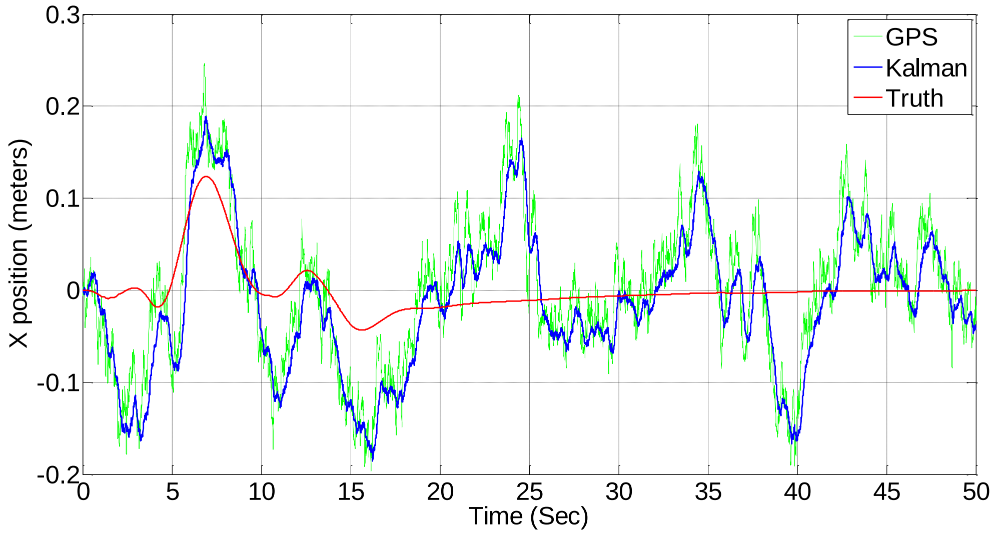

Figure 69: X position estimation

The estimation of the *Y* position is described in the Figure 70. As shown below, the undesired noise is reduced and the *Y* position is estimated with a small mean error around *0.0161 m*.

As we perform an auto takeoff test, the estimation of the *Z* position is very important too. As illustrated in Figure 71, we can comprehend that the estimator behaves well while the helicopter is taking off. The helicopter is commanded to climb to -3 m in the body Z-axis with a slope equals to -1 m/sec velocity in the body frame. After 10 seconds, a PID controller is used to control the helicopter's position. The *Z* position is estimated with a small mean of error about *0.0235 m*.

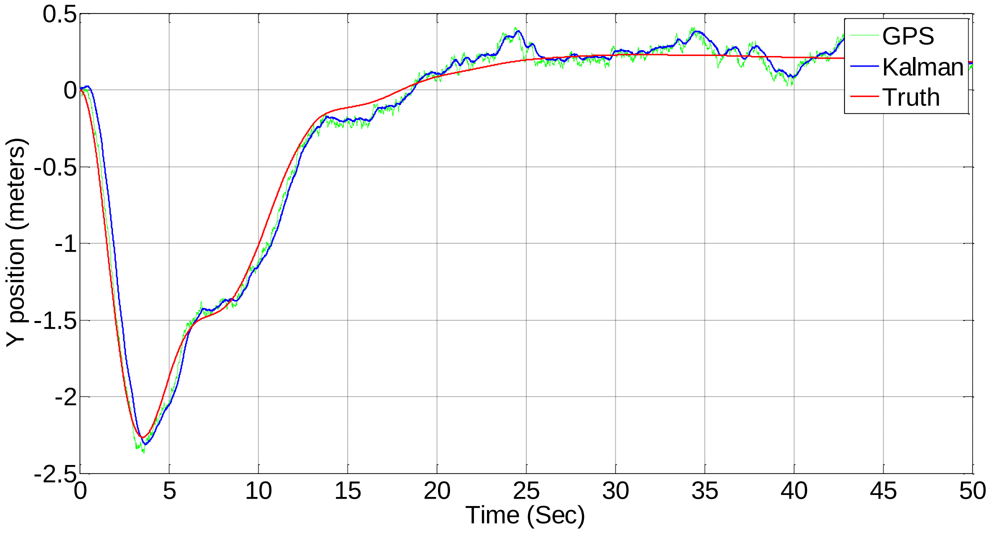

Figure 70: Y position estimation

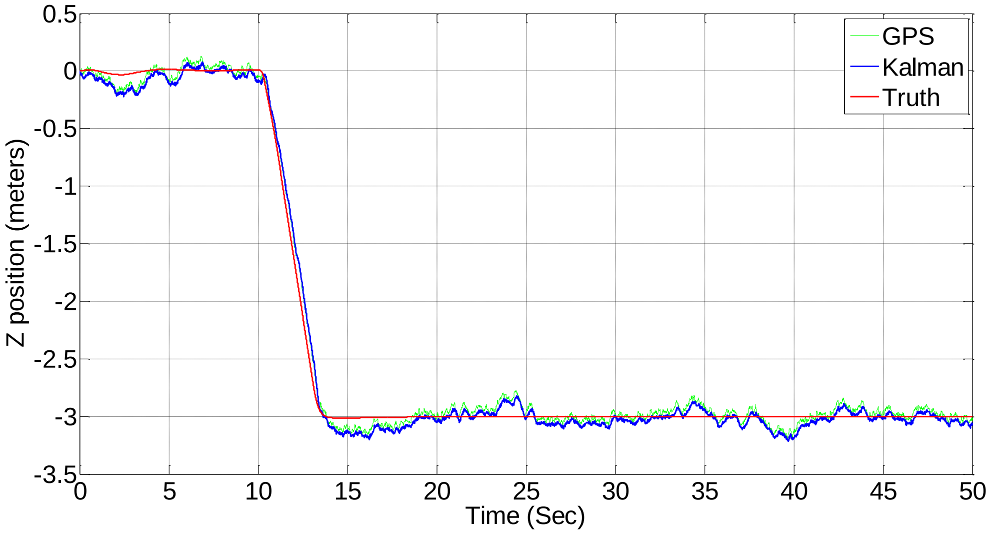

Figure 71: Z position estimation

The estimation of the translational velocities in the body frame is shown in the Figures 72-77 below. In this algorithm the velocities are measured by the optical flow sensor, as a result the estimates of the velocities are more accurate.

- **Velocity estimation**

Figures 72-77 show the estimation performance for the velocity states measured by the OF sensor. The OF measurements are more precise compared to the GPS/INS measurements. As a result of this, the performance of the estimator is better in estimating the velocity states than its performance in estimating the position states.

The *X* velocity estimation is shown in the Figures 72-73 below. It is clear that the estimator rejects the additive noise and estimates the *X* velocity with a small error about *0.0016 m/sec* between the estimated signal and the truth signal. Figure 76 is a zoomed portion of the *X* velocity estimation.

The estimation of the y velocity is illustrated in Figures 74-75. The y velocity is estimated with a small mean error of about *0.0068 m/sec*. Figure 75 shows the zoomed y velocity estimation.

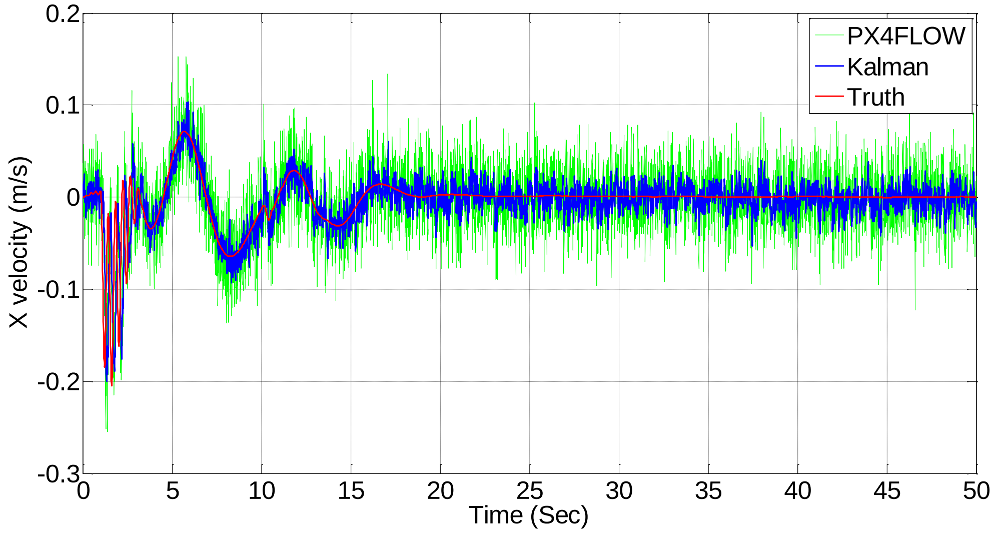

Figure 72: X velocity estimation

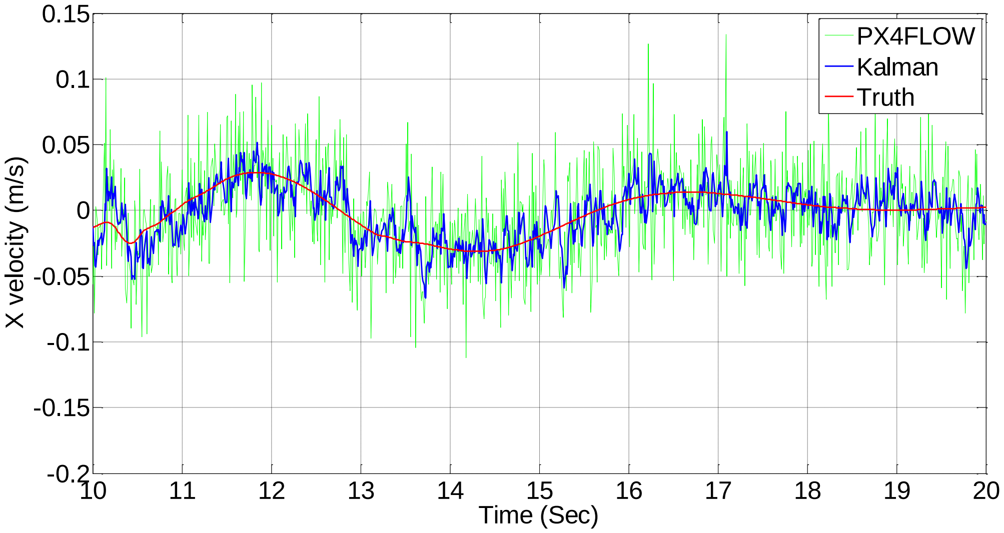

Figure 73: X velocity estimation (zoomed)

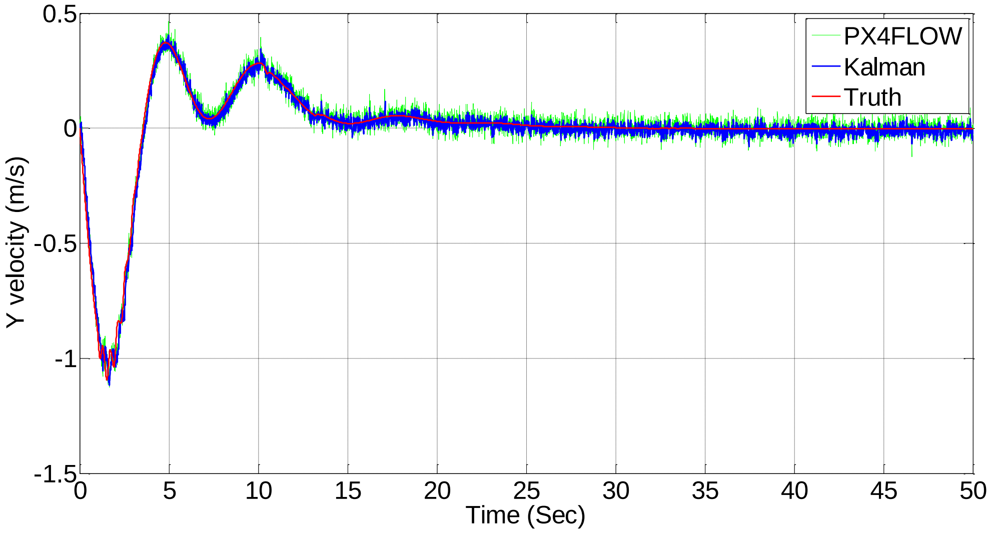

Figure 74: Y velocity estimation

The Z velocity is precisely estimated also, as shown in Figures 76-77. The *Z* velocity is about *-1 m/Sec* during takeoff and goes back to zero when the helicopter reaches the desired Z which is - 3m in our case. The Z velocity is estimated with small mean error of about *0.0112 m/sec*. Figure 77 is a zoomed figure showing the z velocity estimation.

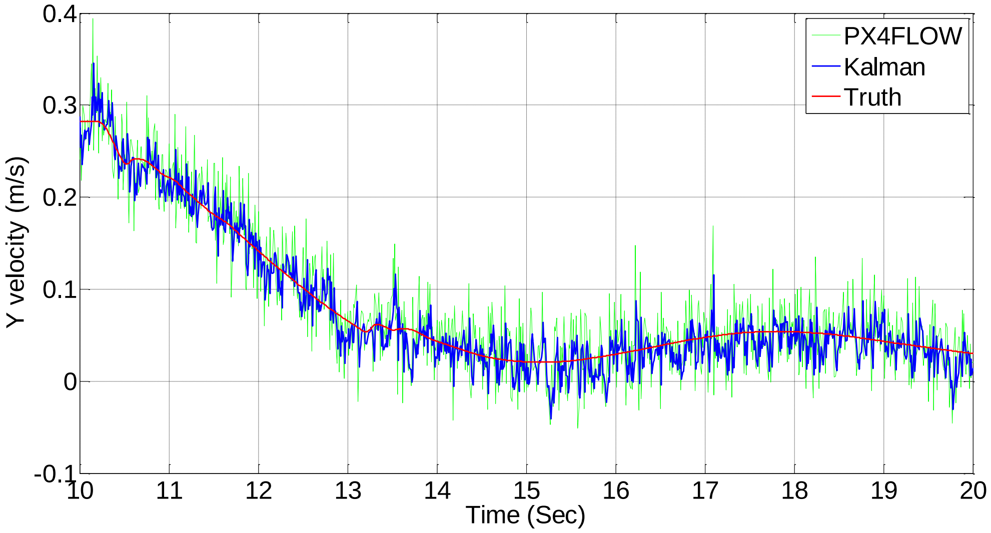

Figure 75: Y velocity estimation (zoomed)

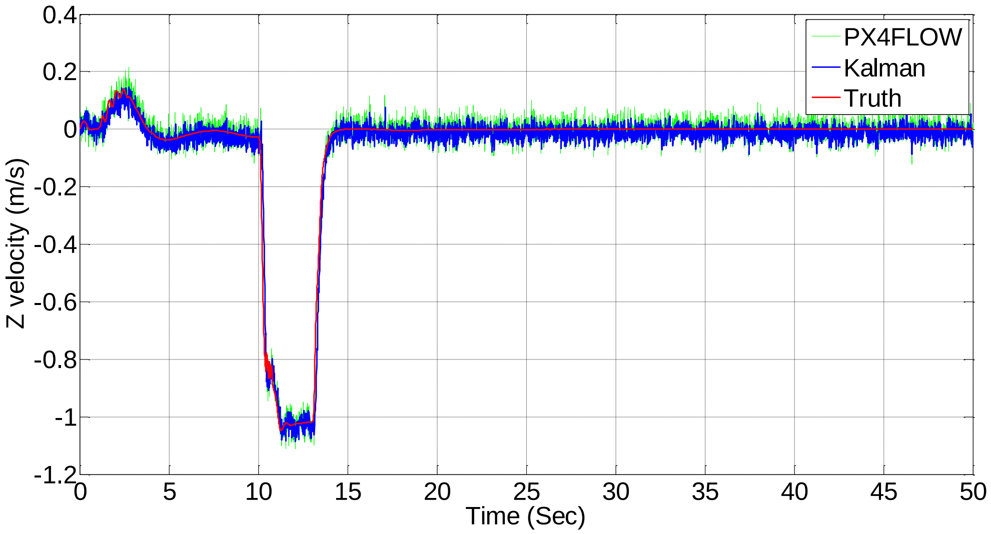

Figure 76: Z velocity estimation

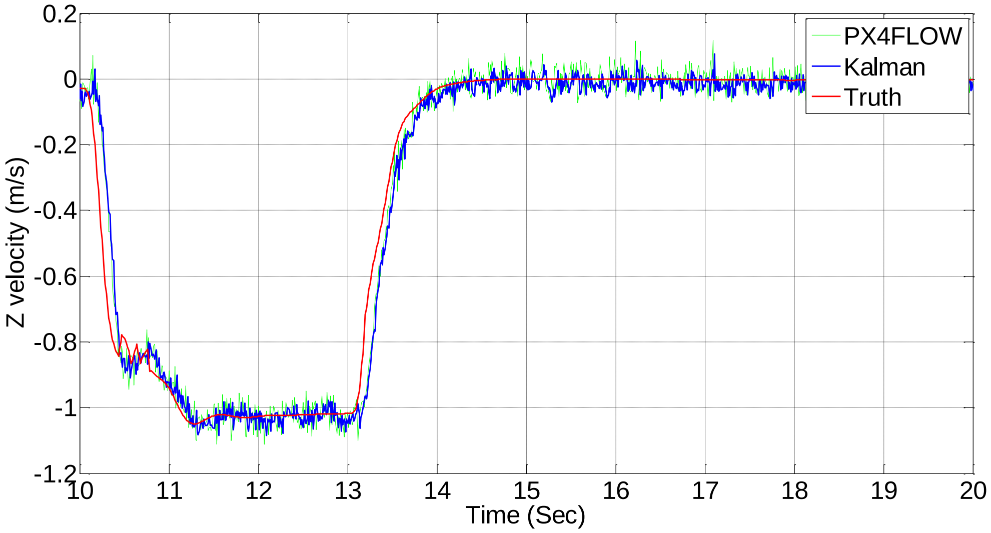

Figure 77: Z velocity estimation (zoomed)

6.3.2 Landing test

The position and velocity states are estimated accurately with a small amount of error. The mean error between the estimated signal and the truth signal is computed for each state during the landing. The X position estimation is described in Figure 79. The X position is estimated with a small mean error of about *0.0133 m* between the estimated signal and the truth signal.

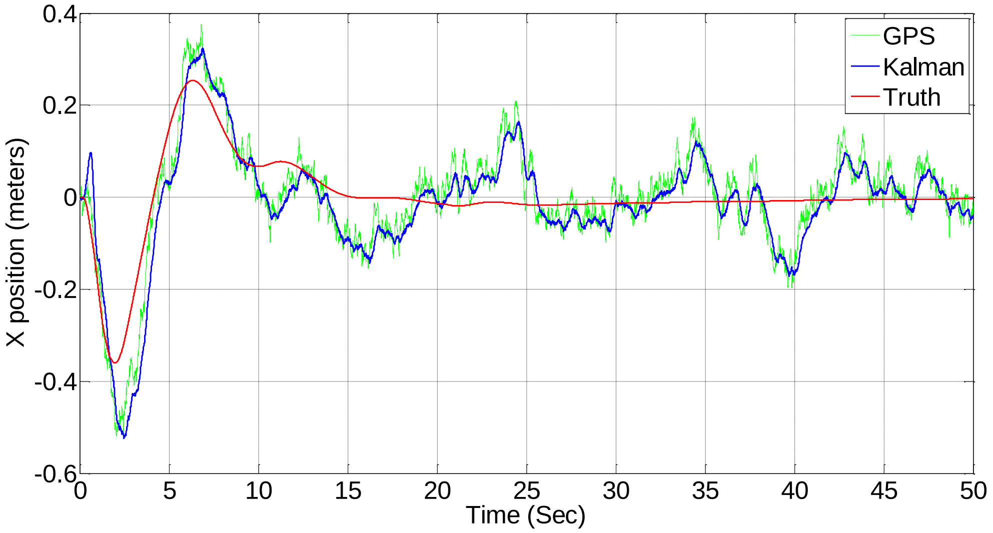

Figure 78: X position estimation

The estimation algorithm behaves well in estimating the *Y* position as shown in Figure 79. The *Y* position is estimated with a small mean error of about *0.0165 m* between the estimated signal and the real signal.

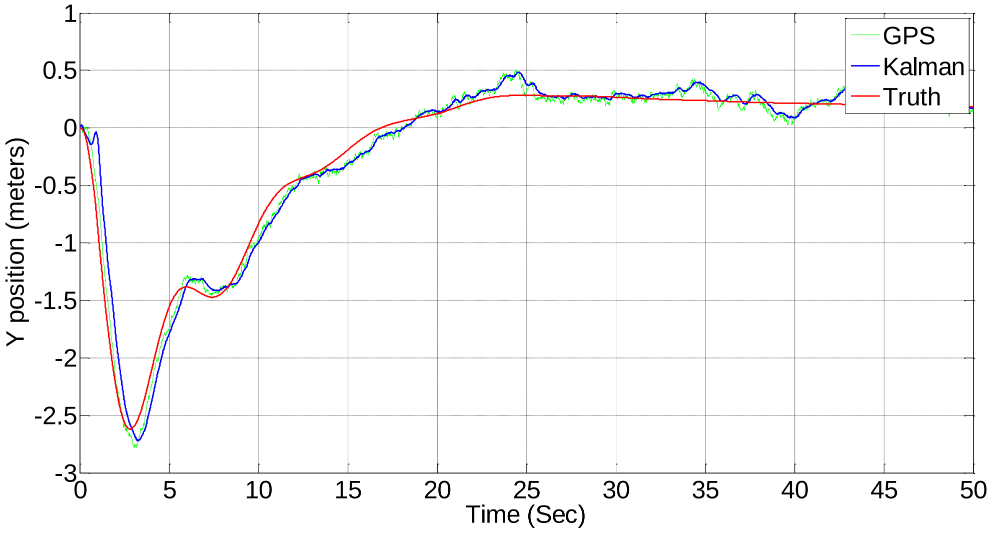

Figure 79: Y position estimation

The *Z* position estimation is depicted in Figure 80. The *Z* position is estimated with a small mean error of about 0.0401 m between the estimated signal and the truth signal.

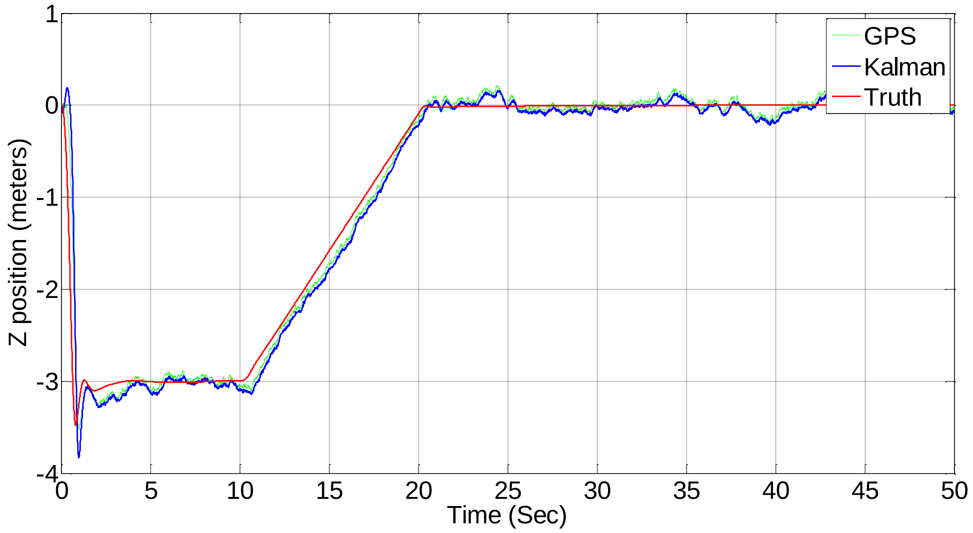

Figure 80: Z position estimation

Figures 81-82 show the *X* velocity estimation. It is obvious that the estimator rejects the additive noise and estimates the *X* velocity with a small error of about *0.0016 m/sec* between the estimated signal and the actual signal. Figure 82 is a zoomed portion of the velocity estimation.

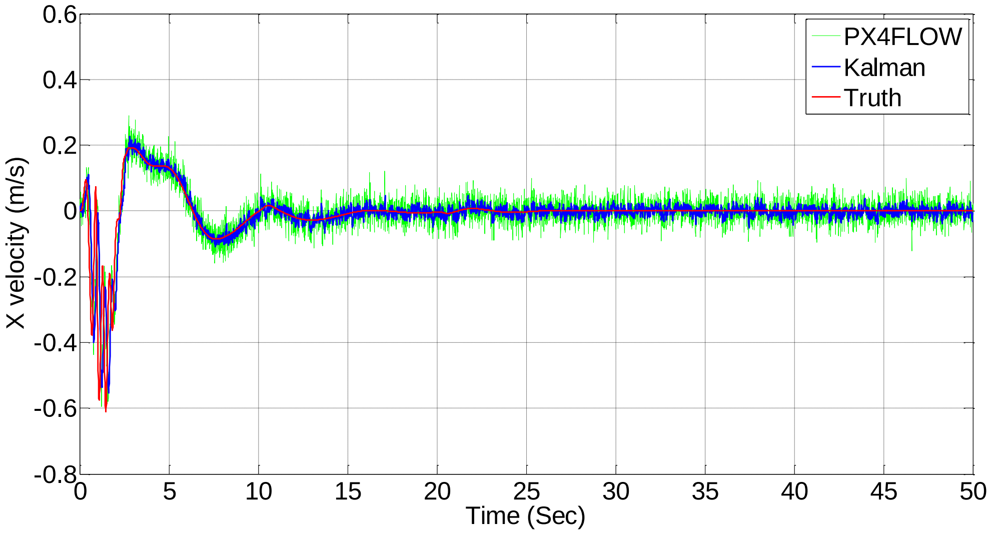

Figure 81: X velocity estimation

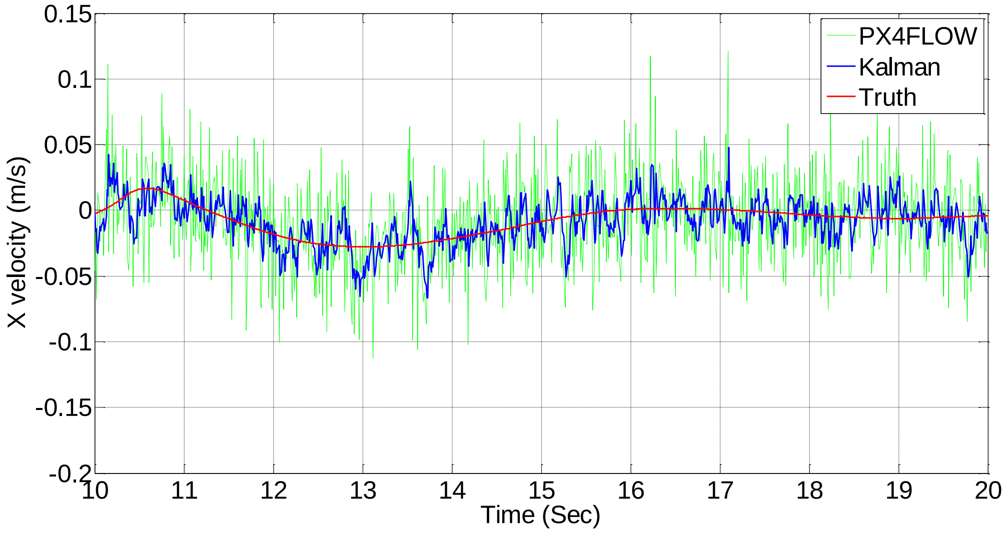

Figure 82: X velocity estimation (zoomed)

The Y velocity estimation is shown in the Figures 83-84. It is clear that the estimator rejects the additive noise and estimates the Y velocity with a small error of about *0.0069 m/sec* between the estimated signal and the truth signal. Figure 84 is a zoomed portion of the Y velocity estimation.

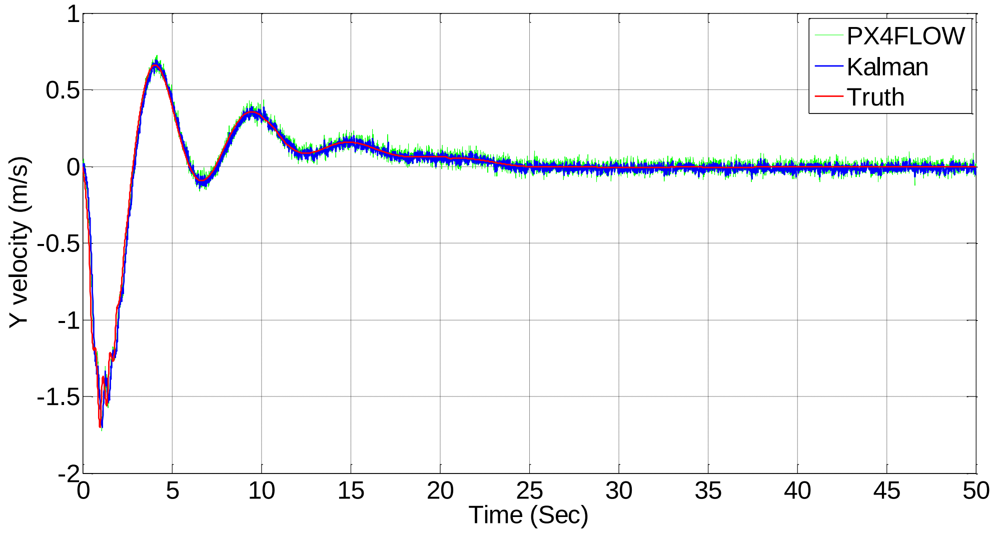

Figure 83: Y velocity estimation

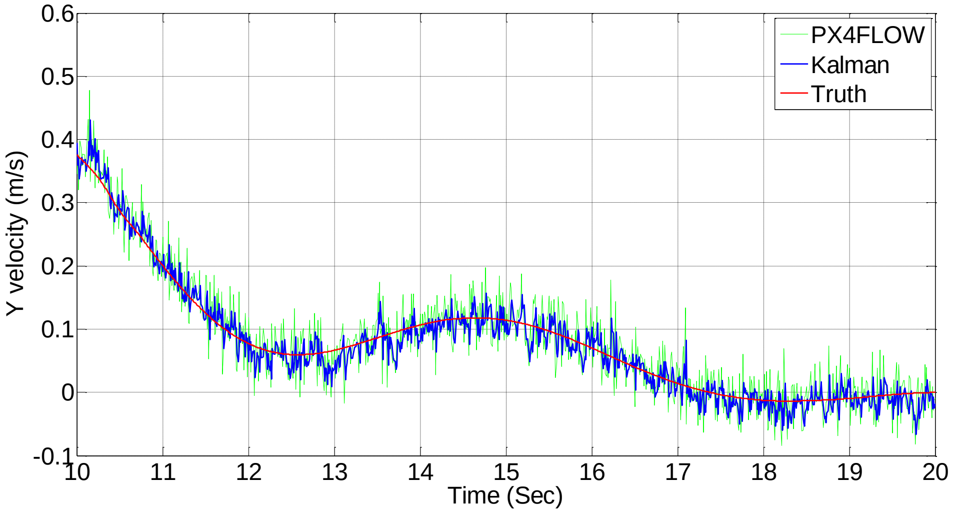

Figure 84: Y velocity estimation (zoomed)

The *Z* velocity estimation is shown in the Figures 85-86 below. It is clear that the estimator rejects the additive noise and estimates the *Z* velocity with a small error about *0.0117 m/sec* between the estimated signal and the truth signal. Figure 86 is a zoomed portion of the *Z* velocity estimation.

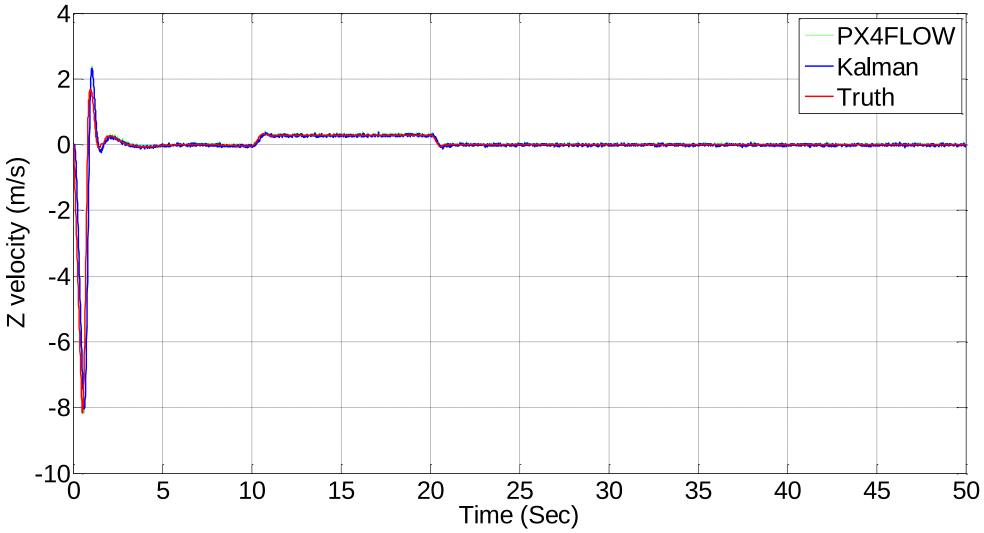

Figure 85: Z velocity estimation

As shown in Figure 86, the *Z* velocity is about *0.3 m/Sec* as commanded during the landing and goes back to zero when the helicopter reaches the ground.

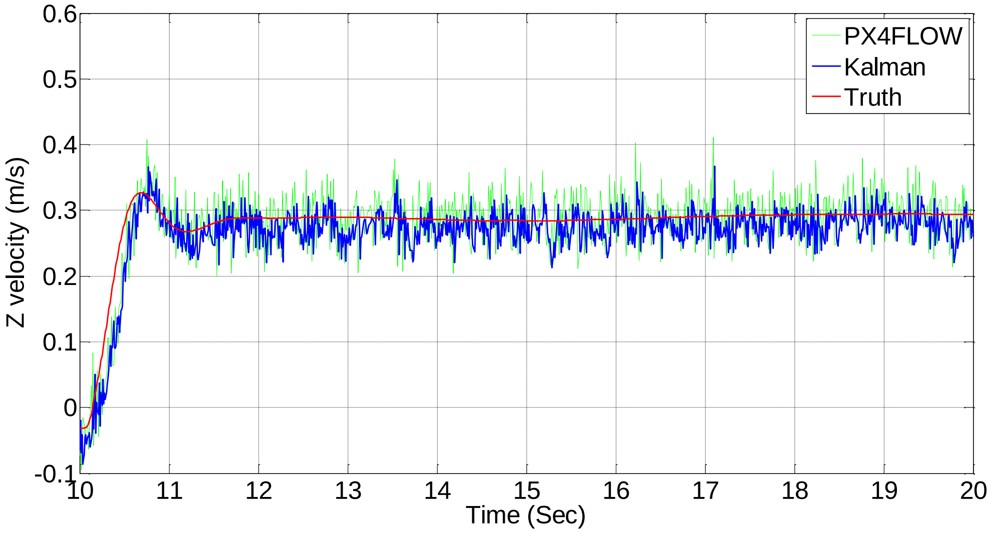

Figure 86: Z velocity estimation (zoomed)

6.4 Discussion

A sensor fusion algorithm was designed and tested to perform an auto takeoff and landing in a small scale helicopter. As shown in section 6.3, the helicopter took off and landed with small positional errors. The designed sensor fusion algorithm proved that the OF sensor measurements can be fused with GPS/INS to obtain much more precise estimates of the position and velocity of an unmanned aerial helicopter.

Tables 10-11 show the statistics of the designed estimation algorithm during the takeoff test. The position states are estimated precisely during the takeoff test. The mean error and the difference in the standard deviation are computed between the estimated and the actual data for the mentioned helicopter states. Obviously, the velocity states have a smaller mean errors compared to the mean errors of the position states. And, the most convincing interpretation of this is that the OF sensor is more precise than the GPS/INS.

Table 10: Position states estimates during takeoff

Helicopter positon states	***Mean error (ME) (m)***	***Difference in Standard Deviation (SDE)(m)***
$\boldsymbol{x}$	*0.0133*	*0.0653*
$\boldsymbol{y}$	*0.0161*	*0.1044*
$\boldsymbol{z}$	*0.0235*	*0.0834*
$\boldsymbol{maximum\ difference_{take\ off} = 0.2583\ m}$		

Table 11: Velocity states estimates during takeoff

Helicopter velocity state	***Mean error (ME) (m/Sec)***	***Difference in Standard Deviation (SDE)(m/Sec)***
$\boldsymbol{u}$	*0.0016*	*0.0224*
$\boldsymbol{v}$	*0.0068*	*0.0279*
$\boldsymbol{w}$	*0.0112*	*0.0346*

As shown in Table 10, the maximum difference between the estimator signal and the actual is about 25.83 cm, which means that the helicopter performs an auto take off with accurate position estimation using the OF sensor information.

Tables 12-13 present the statistics of the designed estimation algorithm during the landing test. Clearly, the mean errors of the position states in both tests are similar except for the Z state. In the landing test the estimator performed better because of the helicopter landing is slower than its takeoff.

Table 12: Position states estimates during landing

Helicopter positon states	*Mean error (ME) (m)*	*Difference in Standard Deviation (SDE)(m)*
$\boldsymbol{x}$	*0.0133*	*0.0750*
$\boldsymbol{y}$	*0.0165*	*0.1380*
$\boldsymbol{z}$	*0.0401*	*0.1994*
$\boldsymbol{maximum\ difference_{landing} = 0.2723\ m}$		

As shown in Table 12, the maximum difference between the estimator signal and the actual is about 13 cm, which means that the helicopter performs an auto landing with position stabilization using the OF sensor information.

Table 13 illustrates the mean errors of the velocity states while the helicopter performs the landing. The mean errors of the velocity in both tests are similar except for the Z velocity state.

Table 13: Velocity states estimation during landing

Helicopter velocity state	*Mean error (ME) (m/Sec)*	*Difference in Standard Deviation (SDE)(m/Sec)*
$\boldsymbol{u}$	*0.0016*	*0.0457*
$\boldsymbol{v}$	*0.0069*	*0.0446*
$\boldsymbol{w}$	*0.0117*	*0.2939*

Based on the previous statistics, the sensor fusion algorithm has obtained accurate estimates during the auto takeoff and landing tests. The estimator has estimated the position and the velocity components in the Z-axis during the takeoff test better than while the helicopter is approaching the ground. This could be due to the fact that the helicopter’s ascending velocity is faster than its descending velocity, which prevents the measurement noise from being accumulated while takeoff. And because of the slow rate of descending, the noise was accumulating more and as a result the measurements are noisier in the landing test.

It is known that GPS/INS based landings are not very accurate, and the internal estimator has a few meters position inaccuracies (Radius of Uncertainty ROU). So by applying the proposed estimation algorithm, we can conclude that the ROU error would be ten times less. Figure 87 illustrates the ROU reduction. The GPS ROU is about few meters while the sensor fusion ROU is a few centimeters.

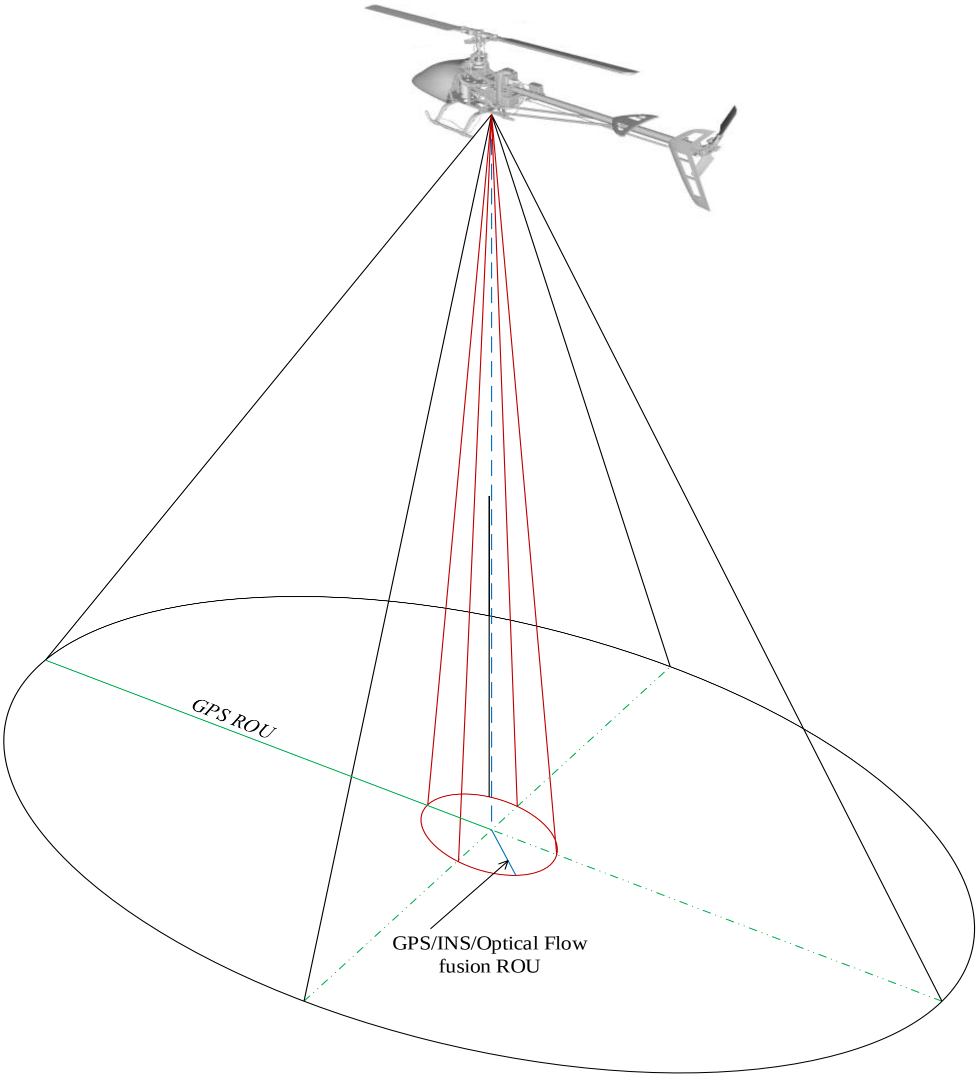

Figure 87: Position error reduction

Chapter 7: Conclusion and Future Work

In this research, we have addressed the problem of auto takeoff and landing of a small scale flybarless helicopter using optical flow sensor information. Two major challenging problems have been solved. The first problem is that the optical flow (OF) sensor model. And the second problem is how to apply the sensor fusion algorithm between the optical flow sensor and the GPS/INS sensor to perform an autonomous takeoff and landing. The first problem was solved by identifying a robust set of OF models, and validating the identified models at different heights. The second problem was solved by designing a sensor fusion algorithm between the OF sensor and the GPS/INS unit. The optical flow sensor was modelled with 87% accuracy of fitting between the identified model and the real OF measurements. The introduced models are tested at different altitudes to ensure their correctness to the altitude variations. The models confirm their usefulness and robustness to different speed conditions.

A sensor fusion algorithm between the GPS/INS sensor and the optical flow sensor was designed to estimate the velocity and the position of the Maxi Joker 3 helicopter. A linear Kalman estimator was used to obtain accurate states estimates. The algorithm was verified by performing a couple of tests. The first test was to ensure the estimator accuracy while the helicopter is taking off and the second test is while landing. The presented results have proven the importance of using the OF sensor and demonstrated the effectiveness of the proposed fusion algorithm. The helicopter has performed a precise take off with maximum error in position not exceeding $0.26m$ and $0.27m$ maximum position error for the landing.

According to the simulation results, the results are promising and the fusion worth implementing on a hardware. It would be interesting to apply the proposed algorithm using the real helicopter platform and conduct a real auto takeoff and landing on a predefined target. Our future work would be in applying different fusion algorithms such as an intelligent fusion algorithm and performing a real auto takeoff and landing flight on a distinguished pattern from various altitudes.

References

[1] S.-R. Oh, K. Pathak, S. K. Agrawal, H. R. Pota and M. Garratt, "Approaches for a Tether-Guided Landing of an Autonomous Helicopter," *IEEE Transactions on Robotics,* vol. 22, no. 3, pp. 536 - 544, 2006.

[2] S. Saripalli and G. Sukhatme, "Landing on a Moving T arget using an Autonomous Helicopter," *In Proceedings of the International Conference onField and Service Robotics,* Mt Fuji, Japan, 2003.

[3] S.-R. Oh, K. Pathak, S. Agrawal, H. R. Pota and M. Garrett, "Autonomous Helicopter Landing on a Moving Platform Using a Tether," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[4] A. Cesetti, E. Frontoni and A. Mancini, "A Vision-Based Guidance System for UAV Navigation and Safe Landing using Natural Landmarks," *Journal of Intelligent & Robotic Systems,* vol. 57, no. 1-4, pp. 233-257, 2010.

[5] S. Saripalli, J. Montgomery and G. Sukhatme, "Visually-Guided Landing of an Unmanned Aerial Vehicle," *IEEE Transactions on Robotics and Automation,* vol. 19, no. 3, pp. 371-381, Jun 2003.

[6] T. Merz, S. Duranti and G. Conte, "Autonomous Landing of an Unmanned Helicopter based on Vision and Inertial Sensing," in *Proceedings of the 9th International Symposium on Experimental Robotics (ISER-2004),* Singapore, 2006.

[7] O. Shakernia, C. Sharp, R. Vidal, H. Shim, Y. Ma and S. Sastry, "Multiple View Motion Estimation and Control for Landing an Unmanned Aerial Vehicle," in *IEEE International Conference on Robotics and Automation*, Washington, DC, 2002.

[8] K. E. Wenzel and A. Zell, "Automatic Take Off, Hovering and Landing Control for Miniature Helicopters with Low-Cost Onboard Hardware," in *Proceedings of the AMS'09, Autonome Mobile Systeme*, Karlsruhe, Germany, December 3-4 2009.

[9] S. Yang, S. Scherer and A. Zell, "An Onboard Monocular Vision System for Autonomous Takeoff,Hovering and Landing of a Micro Aerial Vehicle," *Journal of Intelligent & Robotic Systems,* vol. 69, no. 4, pp. 499-515, 2013.

[10] Z. Yu, K. Nonami, J. Shin and D. Cele, "3D Vision Based Landing Control of a Small Scale Autonomous Helicopter," *International Journal of Advanced Robotic Systems,* vol. 4, no. 1, pp. 51-56, 2007.

[11] M. Meingast, C. Geyer and S. Sastry, "Vision Based Terrain Recovery for Landing Unmanned Aerial Vehicles," in *In Decision and Control, CDC. 43rd IEEE Conference on*, 2004.

[12] C. Sharp, O. Shakernia and S. S. Sastry, "A Vision System for Landing an Unmanned Aerial Vehicle," in *IEEE International Conference on Robotics and Automation*, Seoul, Korea, 2001.

[13] T. Danko, A. Kellas and P. Oh, "Robotic Rotorcraft and Perch-and-Stare: Sensing Landing Zones and Handling Obscurants," in *IEEE International Conference on Advanced Robotics*, Seattle, WA, 2005.

[14] M. Ax, S. Thamke, L. Kuhnert, J. Schlemper and K.-D. Kuhner, "Optical Position Stabilization of an UAV for Autonomous Landing," in *7th German Conference on Robotics*, 2012.

[15] S. Suzuki, "Automatic Take-Off and Landing Control for Small Unmanned Helicopter," *In Robot Intelligence Technology and Applications 2012,* vol. 208, pp. 155-167, 2013.

[16] W. E. Green, P. Y. Oh, K. Sevcik and G. Barrows, "Autonomous Landing for Indoor Flying Robots Using Optic Flow," in *Proceedings of IMECE'03 2003 ASME International Mechanical Engineering Congress,* Washington, D.C., 2003.

[17] S. R. Griffiths, "Remote Terrain Navigation for Unmanned Air Vehicles," in *Master thesis, Brigham Young University*, 2006.

[18] B. H´eriss´e, T. Hamel, R. Mahony and F.-X. Russotto, "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow," *IEEE Transactions on Robotics,* vol. 28, no. 1, pp. 77- 89, 2012.

[19] M. Tresanchez, T. Pallejà, M. Teixidó and J. Palacín, "The optical mouse sensor as an incremental rotary encoder," *Sensors and Actuators A,* vol. 155, p. 73–81, 2009.

[20] J. S. Light, "Tip vortex geometry of a hovering helicopter rotor in ground effect.," *Journal of the American Helicopter Society,* vol. 38, no. 2, pp. 34-42, 1993.

[21] K. Nonaka and H. Sugizaki, "Integral Sliding Mode Altitude Control for a Small Model Helicopter with Ground Effect Compensation," in *American Control Conference (ACC),* San Francisco, CA, USA, 2011.

[22] I. C. Cheeseman and W. E. Bennett, "The Effect of the Ground on a Helicopter Rotor in Forward Flight," A.R.C Technical Report R & M No.3021, 1957.

[23] N. D. Nathan and R. B. Green, "Measurements of a Rotor Flow in Ground Effect and Visualisation of the Brown-out Phenomenon," in *American Helicopter Society 64th Annual Forum*, Montreal, Canada, 2008.

[24] F. Ruffier and N. Franceschini, "Optic Flow Regulation: The Key to Aircraft Automatic Guidance," *Robotics and Autonomous Systems,* vol. 50, no. 4, pp. 177-194, 2005.

[25] D. K. Sorensen, "On-line Optical Flow Feedback for Mobile Robot Localization/Navigation," in *Master Thesis Texas A&M University*, 2003.

[26] H. Lim, H. Lee and H. J. Kim, "Onboard Flight Control of a Micro Quadrotor Using Single Strapdown Optical Flow Sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, 2012.

[27] K. Alexis, C. Papachristos, G. Nikolakopoulos and A. Tzes, "Model Predictive Quadrotor Indoor Position Control," in *19th Mediterranean Conference on Control and Automation,* Aquis Corfu Holiday Palace, Corfu, Greece, 2011.

[28] J. Stowers, A. Bainbridge-Smith, M. Hayes and S. Mills, "Optical Flow for Heading Estimation of a Quadrotor Helicopter," *International Journal of Micro Air Vehicles* , vol. 1, no. 4, pp. 229-239, 2009.

[29] D. Honegger, L. Meier, P. Tanskanen and M. Pollefeys, "An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications," in *IEEE International Conference on Robotics and Automation (ICRA),* 2013.

[30] M. B. Hurd, "Control of a Quadcopter Aerial Robot Using Optic Flow Sensing," in *Master Thesis University of Nevada* , 2013.

[31] M. A. Jaradat and M. F. Abdel-Hafez, "Enhanced, Delay Dependent, Intelligent Fusion for INS/GPS Navigation System," *IEEE Sensors Journal,* vol. 14, no. 5, pp. 1545 - 1554., 2014.

[32] J. Ma and S. Sun, "Centralized Fusion Estimators for Multisensor Systems With Random Sensor Delays, Multiple Packet Dropouts and Uncertain Observations," *IEEE Sensors Journal,* vol. 13, no. 4, pp. 1228-1235, 2013.

[33] A. Dallil, M. Oussalah and A. Ouldali, "Sensor Fusion and Target Tracking Using Evidential Data Association," *IEEE Sensors Journal,* vol. 13, no. 1, pp. 285-293, 2013.

[34] J. Shin, K. Nonami, D. Fujiwara and K. Hazawa, "Model-based optimal attitude and positioning control of small-scale unmanned helicopter," *Robotica,* vol. 23, no. 1, p. 51–63, 2005.

[35] M. F. Abdel-Hafez, "The Autocovariance Least Squares Technique for GPS Measurement Noise Estimation," *IEEE Transactions on Vehicular Technology,* vol. 59, no. 2, pp. 574-588, 2010.

[36] M. F. Abdel-hafez, "Detection of Bias in GPS Satellites' Measurements: A Probability Ratio Test Formulation," *IEEE Transactions on Control Systems*, vol. 22, no. 3, pp. 1166-1173, 2014.

[37] Y.-L. Wu, T.-M. Wang, J. H. Liang, C.-L. Wang and C. Zhang, "Attitude estimation for small helicopter using extended kalman filter," in *Proceedings of the IEEE Conference on Robotics,Automation and Mechatronics,* Chengdu,China, 2008.

[38] T. Gao, Z. Gong, J. Lue, W. Ding and W. Feng, "An Attitude Determination System For A Small Unamnned heliocopter Using low-cost

sensors," in *IEEE International Conference on Robotics and Biomimetics*, 2006.

[39] M. Jun, S. I. Roumeliotis and G. S. Sukhatume, "State Estimation Via Sensor Modeling for Helicopter Control Using An Indirect Kalman Filter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999.

[40] M. Barczyk and A. . F. Lynch, " Integration of a Triaxial Magnetometer into a Helicopter UAV GPS-Aided INS," *IEEE transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2947- 2960, 2012.

[41] H.-K. Tenn, S.-S. Jan and F.-B. Hsiao, "Pitch and Roll Attitude Estimation of a small-scaled helicopter Using Single Antenna GPS with Gyroscopes," *GPS Solutions*, vol. 13, no. 3, pp. 209-220, 2009.

[42] D. A. Mercado, G. Flores, P. Castillo, J. Escareno and R. Lozano, "GPS/INS/Optic Flow Data Fusion for Position and Velocity estimation," in *International Conference on Unmanned Aircraft Sysems (ICUAS)*, Atlanta,GA, 2013.

[43] M. Al-Sharman, M. F. Abdel-Hafez and M. Al-Omari, "Attitude and Flapping Angles Estimation for a Small-Scale Flybarless Helicopter Using a Kalman Filter," *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2114 - 2122, 2014.

[44] M. Al-Sharman, M. Abdel-Hafez and M. Al-Omari, "State Estimation for a Small Scale Flybar-less Helicopter," in *2nd International Conference on System-Integrated Intelligence: Challenges for Product* , Bremen, Germany, 2014.

[45] L. R. Sahawneh, M. A. Al-Jarrah, K. Assaleh and M. F. Abdel-Hafez, "Real-Time Implementation of GPS Aided Low-Cost Strapdown Inertial Navigation System," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 527-544, 2011.

[46] W. R. Williamson, M. F. Abdel-Hafez, I. Rhee, E.-J. Song, J. D. Wolfe, D. F. Chichka and J. L. Speyer, , "An Instrumentation System Applied to Formation Flight," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 1, pp. 75 - 85, 2007.

[47] K. Saadeddin, M. F. Abdel-Hafez, M. A. Jaradat and M. A. Jarrah, "Performance Enhancement of low-cost, High-accuracy, State Estimation for Vehicle Collision Prevention System using ANFIS," *Mechanical Systems and Signal* , vol. 41, no. 1-2, pp. 239-253, 2013.

[48] K. Saadeddin, M. F. Abdel-Hafez, M. A. Jaradat and M. A. Jarrah, "Optimization of Intelligent Approach for Low-Cost INS/GPS Navigation System," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 325-348, 2014.

[49] A. Alshoubaki, "Autopilot Design and Commercial Autopilot Evaluation Using Flybarless Helicopter," in *Master Thesis American university of sharjah*, Sharjah, 2014.

[50] AVAGO Technology, " Data Sheet of ADNS-3080 High-Performance Optical Mouse Sensor".

[51] S. Lange, N. Sunderhau and P. Protzel, "A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments," in *IEEE International Conference on Advanced Robotics, ICAR* 2009.

[52] S. Barnard and W. Thombson, "Disparity Analysis of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 2, no. 4, pp. 333-340, 1980.

[53] R. Jain, R. Kasturi and B. G. Schunck, Machine Vision, McGraw-Hill, 1995.

[54] B. K. Horn and B. G. Schunck, "Determining Optical Flow," Artificial intelligence laboratory MIT, 1981.

[55] B. D. Lucas, "Generalized Image Matching by the Method of Differences," in *PhD Thesis Carnegie Mellon University*, 1984.

[56] M. V. Srinivasan and R. L. Gregory, "How Bees Exploit Optic flow: Behavioural Experiments and neural Models [and discussion]," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences,* vol. 337, no. 1281, pp. 253-259, 1992.

[57] N. S. Love and C. Kamath, "An Empirical Study of Block Matching Techniques for the Detection of Moving Objects," Lawrence Livermore National Laboratory, 2006.

[58] D. Flubacher, "Characterisation of an Optical Flow Sensor for Off-road Robot Application," in *Bachelor-Thesis Swiss Federal Institute of Technology Zurich (ETH*, 2008.

[59] M. V. Srinivasan, J. S. Chahl , K. Weber, S. Venkatesh, M. G. Nagle and S. W. Zhang, "Robot Navigation Inspired by Principles of Insect Vision," *Robotics and Autonomous Systems,* vol. 26, no. 2, pp. 203-216, 1999.

[60] W. E. GREEN and P. Y. OH, "Optic-Flow-Based Collision Avoidance Applications Using a Hybrid MAV," *IEEE Robotics & Automation Magazine,* vol. 15, no. 1, pp. 96-103, 2008.

[61] V. Gavrilets, "Autonomous Aerobatic Maneuvering of Miniature," in *PhD Thesis Massacheutes Institute of Technology* , 2003.

[62] G. D. Padfield, Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling, AIAA, 1996.

[63] R. E. Miranda, "Development of a Nonlinear 6-Degree of Freedom Miniature Rotary-Wing Unmanned Aerial Vehicle Software Model and PID Flight Path Controller Using Mathworks Simulink Simulation Environment," in *Master Thesis Naval Postgraduate School*, California, 2009.

[64] V. Gavrilets, B. Mettler and E. Feron , Dynamic Model for a Miniature Aerobatic Helicopter, AIAA, 2001.

[65] "Baumer encoders," Baumer, [Online]. Available: http://www.baumer.com/us-en/products/rotary-encoders-angle-measurement/absolute-encoders/.

[66] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software, New York: John Wiley &Sons, 2001.

Vita

Mohammad K. S. Al-Sharman was born in Irbid, Jordan. He studied and graduated from Al-Mazar Secondary School. He received his B.S in Mechatronics engineering from Tafila Technical University in Jordan in August 2011. In September 2012, Mr. Al-Sharman joined the Mechatronics graduate program and started his Master's in the science of Mechatronics engineering at the American University of Sharjah. He is a graduate research assistant at the American University of Sharjah. He is involved in designing and benchmarking the MAZARI autopilot. His research interests are in sensor fusion, state estimation, control theory, Unmanned Aerial Vehicles and Mechatronics design.