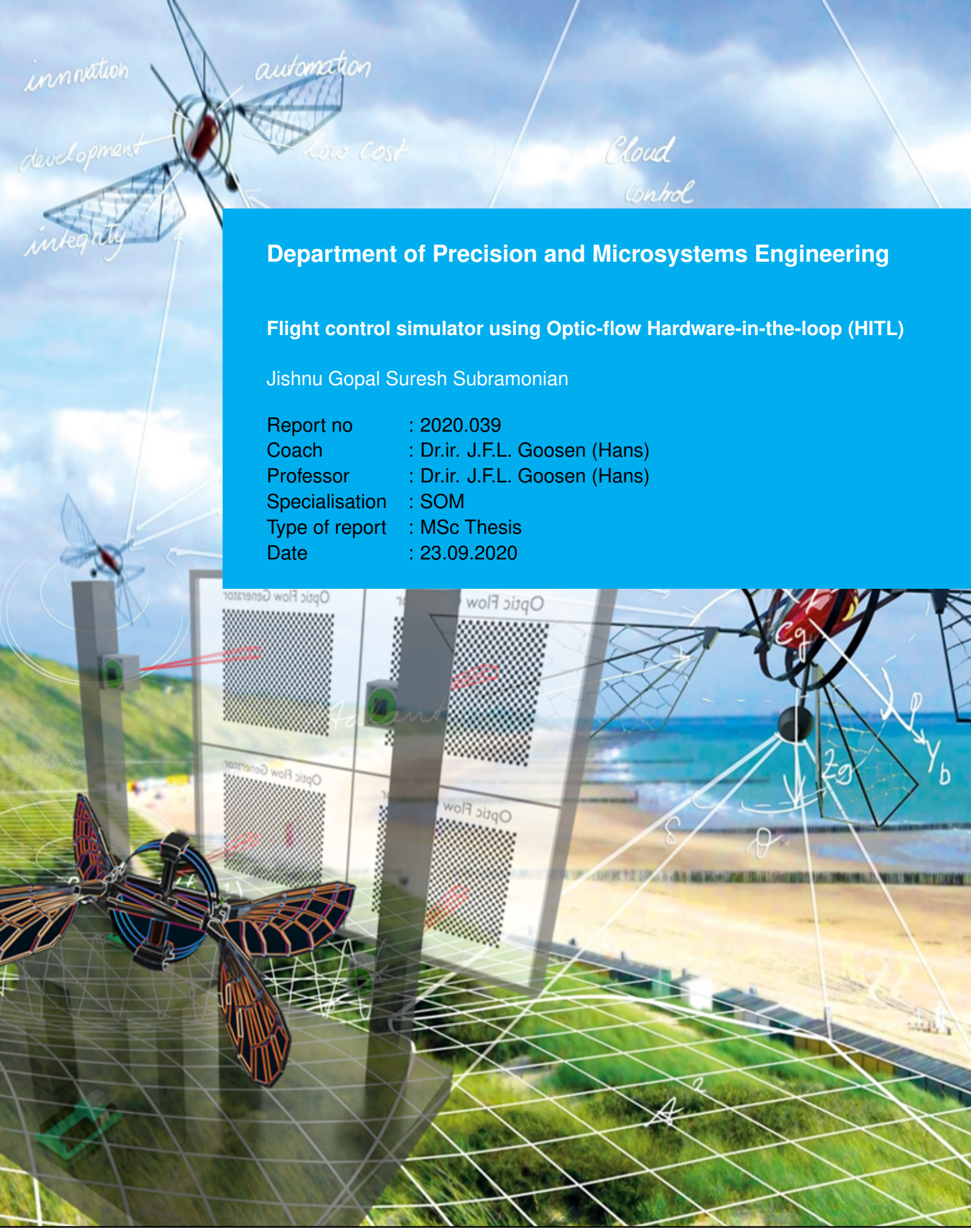


Department of Precision and Microsystems Engineering

Flight control simulator using Optic-flow Hardware-in-the-loop (HITL)

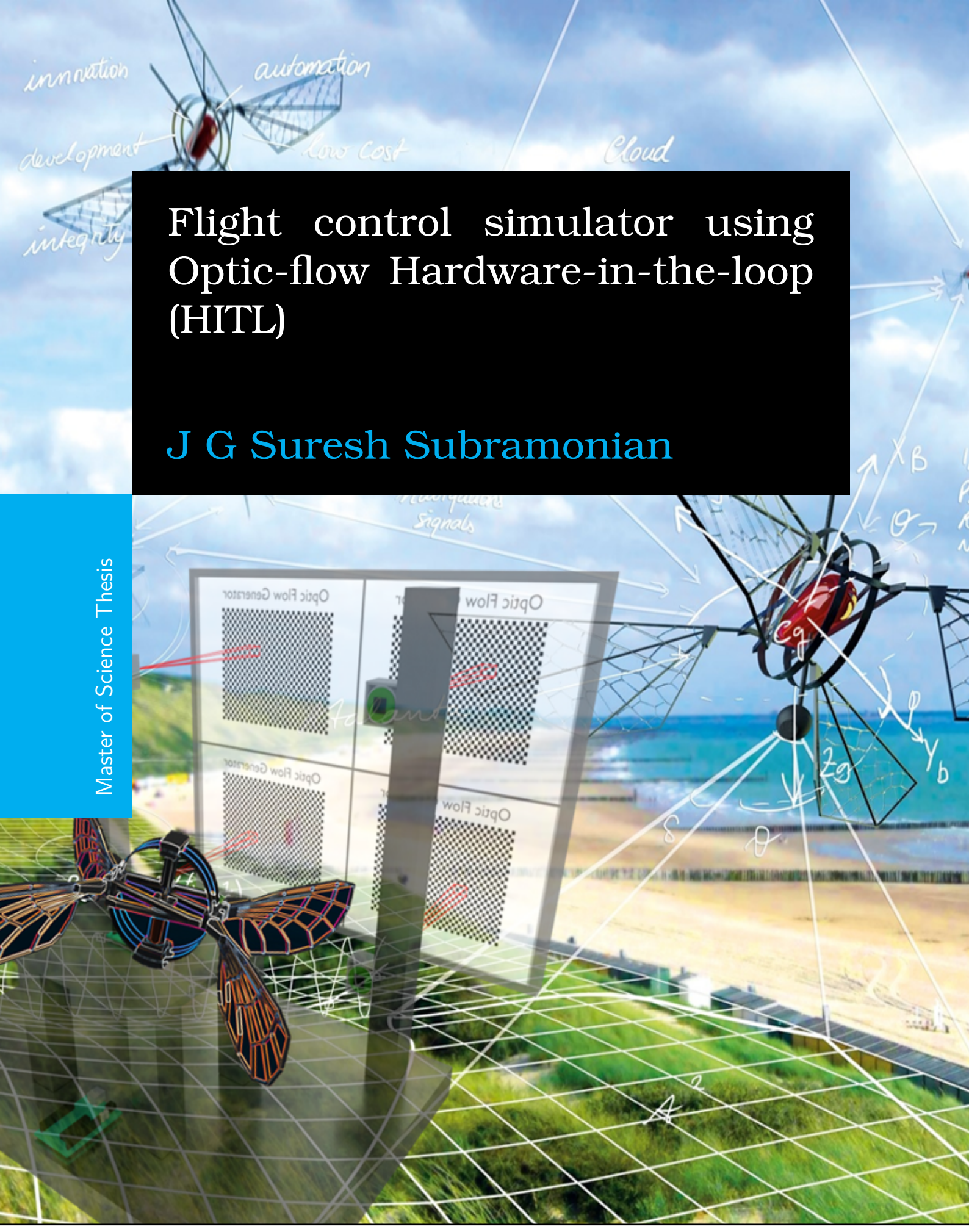
Jishnu Gopal Suresh Subramonian

Report no : 2020.039
Coach : Dr.ir. J.F.L. Goosen (Hans)
Professor : Dr.ir. J.F.L. Goosen (Hans)
Specialisation : SOM
Type of report : MSc Thesis
Date : 23.09.2020

Flight control simulator using Optic-flow Hardware-in-the-loop (HITL)

J G Suresh Subramonian

Master of Science Thesis



TU Delft Delft University of Technology

Precision and Microsystems Engineering

Flight control simulator using Optic-flow Hardware-in-the-loop (HITL)

Master of Science Thesis

J G Suresh Subramonian

September 23, 2020

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Flight control simulator using Optic-flow Hardware-in-the-loop (HITL)

Master of Science Thesis Report

In partial fulfillment of the requirements for the degree of
Master of Science in Mechanical Engineering

by

Jishnu Gopal Suresh Subramonian

to be defended publicly on 23 September 2020

Precision and Microsystems Engineering (PME)
Faculty of Mechanical, Maritime and Materials Engineering (3mE)
Delft University of Technology
The Netherlands

Thesis Committee:

Dr.ir. J.F.L. Goosen (Hans)	Faculty of Mechanical, Maritime & Materials Engineering
Prof. Dr. G.C.H.E. (Guido) de Croon	Faculty of Aerospace Engineering
Dr.ir. Chris Verhoeven	Faculty of Electrical Engineering, Mathematics & Computer Science (EEMCS)

Abstract

The most simplistic acts in nature can be incredibly complex to replicate. Among which, insects in general, are profound micro-machines with evolution as the backbone for their most optimized features. This study focuses on developing an efficient sensing and processing strategy for autonomous flight maneuvers in Atalanta, a bio-inspired robotic-fly that uses unconventional flapping wing propulsion for flights.

Visual guidance has proven a very significant role in the fields of autonomous robot navigation. Optical-flow-based solutions provide map-less navigation strategies, especially for miniaturized robots that possess stringent size, weight, and processing (SWaP) constraints. The primary objective focuses on the evaluation of a compact optical mouse sensor (ADNS 9800) for use as an optical-flow sensor in autonomous robots for indoor applications. The sensor is programmed for optic-flow detection and enhanced with suitable optics to improve the existing focal range from 3 mm to a few meters. A customizable MATLAB GUI is developed to create a digital sensing environment for the sensor. After necessary modifications, the sensor has proven to be sensitive under 70% brightness and contrast in the digital optical-flow environment. Despite the limited resolution (30×30 pixels) and field-of-view ($2 - 3^o$), the sensor is identified and proven to perform well as an optical-flow sensor. Combinations of these sensors are used to develop a Hardware-in-the-loop (HITL) flight simulator to understand pure optic-flow based controls-strategies. Results indicate that the flight simulator works as intended for indoor situations within an error tolerance in Optic-flow of 4.83% from the actual path.

Keywords - *Robotics, MAVs, Optic flow, mobile robots, vision, bio-inspiration, visual navigation, HITL, controls, optical-mouse sensor, flight-simulator.*

Table of Contents

List of Figures

List of Tables

Preface

This document is a part of my research work for my Master of Science graduation Thesis in the Department of High-Tech Engineering at the faculty of Mechanical Engineering, Delft University of Technology. The Atalanta project has been handled for several years in the Department of Precision and Microsystems Engineering by Dr.ir, Hans Goosen. His contribution has developed the research to a great extent and inspired many students to work on various subsystems. Fascinated by robotics and nature, I chose to approach him as my supervisor for my internship and thesis.

During my course work in the first year, I experienced a demo of FWMAV Delfly on air by MAV Labs, guest lectured by Prof. Hans. That was the first time I came across bio-inspired robotics. My natural flair towards robotics propelled me to choose related course work and pursue an internship where I mostly worked in vision sensing and control of manipulators. From the learnings I gained during the internship, I wanted to extend them further on similar grounds for vision-related robotics. Hans was instrumental in helping me narrow down to the appropriate focus area on further discussions. The major objective of my literature study aimed to develop an optical flow-related sensing strategy for the controls of the autonomous flying of the Atalanta. The current thesis research focuses on achieving this end-goal.

Acknowledgements

Foremost, I would like to thank Technische Universiteit Delft, 3mE – Department of High-Tech Engineering and RoboValley – RoboHouse Fieldlab for helping me pursue a journey towards my intended focus area Robotics. I would express my sincere and hearty gratitude towards Professor Hans Goosen, my academic Mentor for being a great pillar of support and encouragement for my studies in the field of Robotics for Internship and Thesis, and for helping me develop as a better engineer and individual. His timely inputs and expert guidance helped me adhere to the topic, yet thinking outside. I would like to thank Daan Van Vrede for being hugely supportive in clearing my doubts related to thesis. The work done on the mouse-sensor by BEP 2014 batch was helpful to understand the concepts. My colleagues at RoboHouse helped with honing my programming and robotic skills. I thank David Scheper, who has been highly supportive in creating relevant professional Blender-illustrations for my thesis. I am deeply obliged to express my sincere gratitude towards my parents Dr. B Suresh Subramonian and Mrs. M Neela, for their constant ever-green moral support. It has been a wonderful journey for the past two years at Delft, mainly because of the fruitful memories gathered with my friends here. Also, I would like to thank Eveline Matroos, our course-coordinator, our go-to person at the department for being so reachable. Lastly, I would like to thank everyone who has been instrumental in helping to maintain a calm situation, in-general, and amidst the COVID crisis, especially the university staff.

Delft University of Technology
September 23, 2020

J G Suresh Subramonian

"A good scientist is a person with original ideas. A good engineer is a person who makes a design that works with as few original ideas as possible. There are no prima donnas in engineering"

— *Freeman Dyson*

Glossary

List of Acronyms

OF - Optical Flow
FWMAV - Flapping Wing Micro Aerial Vehicle
MAV - Micro Aerial Vehicle
fps - Frames per second
FoV - Field of View
UAV - Unmanned Aerial Vehicles
SWaP - Size, Weight and Processing
SPI - Serial Peripheral Interface
MEMS - Micro-Electro-Mechanical Systems
GPS - Global positioning System
IMU - Inertial Measurement Unit
OF^{y}_{X+} - OF sensed by X+ sensor in Y direction

Chapter 1

Introduction

Initially, a more general perspective of Unmanned Aerial Systems is explained in brief further narrowing down to the expected focus area being unconventional Flapping wing Micro Aerial Vehicles (FWMAVs). A quick birdseye view of the supporting features enabling their bio-inspired nature and general advantages are described. The section concludes with understanding Atalanta, a flapping wing robotic-fly that was developed at TU Delft and the gaps to make it autonomous.

Nature does incredible things that seem effortless but are a-way beyond normal understanding, among which insects are one such extraordinary creation. This research focuses on understanding their sensing behaviors for efficient flights. For example, butterflies apart from being splendid, behave like self-propelled flowers that possess an immense challenge to reverse engineer and replicate as autonomous working mechanisms. A famous saying by Otto Lilienthal, *'To invent the airplane is nothing. To build one is something. To fly is everything'* reminds the meticulous difficulty involved in making mechanisms fly. The new generation fight-controllers for micro-aerial vehicles have well-equipped drones to explore unearthed potentials. Substantial effort has gone into the research and development of miniaturizing and enhancing the stability-cum-performance of Unmanned aerial systems.

However, down-sizing an airplane adhering to its core functionalities involves a thorough understanding of the complete system consisting of propulsion aerodynamics and the art of balancing the power to weight ratio. Unmanned-systems were primarily large and could fly a few thousand kilometers, and in contrast, mini-drones can hover and navigate in confined space [10]. Drones vary in their classification based on their mission and platform wherein most demanding areas are the small scale unmanned-systems that insist on the development of sensor-fusion technologies, optimized batteries, and smart navigation systems [1]. The following section supports to gain a broader perspective of the Unmanned systems and the various classifications, finally funneling down to the Flapping Wing Micro Aerial Vehicles (FWMAVs). This study focuses on one such Micro Aerial Vehicle called the Atalanta and enabling autonomous flight using optical flow.

1-1 Unmanned Aerial Vehicles - UAVs

The absence of an on-board human pilot in an aerial vehicle classifies it to be Unmanned-System. Usually, there may exist a ground-based controller and a system of communication among them.

The initial usage of drones can be recalled for being uses as the practice targets for training military professionals in the early 1900s [11]. A drone can be defined as a flying robot that can be remotely controlled or flown autonomously through software-controlled flight plans in their embedded systems, working in concurrence with built-in GPS and sensor data [12]. Depending on the intended task of a UAV, it can be classified into six functional categories where our focus area lies in Research and Development - to improve UAVs. Though drones were mostly used in military applications, they have now divulged into the commercial segment for various applications such as surveillance, photography, and swarm robotics. They are also becoming a vital part of the farming sector for ensuring crop health and door-delivery purposes. The general classification of the Drones can be seen in Figure 1-1 which is mainly separated based on the wingspan and weight. The other classifications for unmanned-systems can be based on the type of propulsion and intended purpose. In the ascending size order, unmanned-systems can be listed as starting from SD (Star Dust), PAV (Pico Aerial Vehicle), NAV (Nano Aerial Vehicle), MAV (Micro Aerial Vehicle), μUAV and UAV.

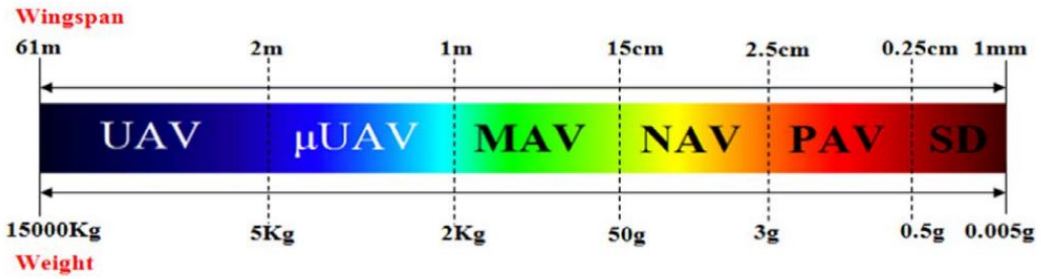


Figure 1-1: Classification of Aerial Unmanned Systems [1]

Miniature aerial vehicles usually range from micro-scale up to a man-portable system. The range of the aerial vehicle varies over a large scale from few grams to kilograms [13] and spanning across from few mm to 60 m [1]. Micro aerial vehicles are class of miniature UAVs that have size restrictions and tend to be autonomous. The outcomes of various ongoing research have increased the demand for more intelligence on-board. Bio-inspired MAVs use intelligent adaptive mechanisms and sensor fusion technology to enable flight operation.

1-2 Flapping Wing Micro Aerial Vehicle (FWMAV)

The current study focuses exclusively on this type of MAVs that were inspired by small birds like hummingbirds or huge insects like honeybees/dragonflies [14] that propel by periodic flapping wings generating its uplift and thrust unlike the regular high-speed motors used in drones. The underlying flight principles are slightly different, wherein insects almost produce a horizontal stroke plane while flapping their wings with large amplitudes at a faster rate.

In contrast, large birds flap their wings in vertical stroke plane while cruising and at a lower frequency with smaller amplitudes [15]. It becomes essential to understand the differences and similarities in insect and bird flights that help in designing various flapping-wing aerial vehicles. One main challenge possessed for the scientists was to understand the aerodynamics at such low Reynolds number and yet generating sufficient thrust for lift. This information provides significant insights into the control architecture.

1-2-1 Advantages

Conventional drones possess few limitations, that pave paths for introducing other systems such as FWMAVs. The fixed propellers in drones generate maximum upward thrust for the lift, but they limit the performance by sacrificing on agility and maneuverability. The alternative to combat the problem and modify the lift components of the blade during flight is implementing the Active Pitching of the propeller blades. Yet, drones possess potential threats like crashing on humans and creating devastating effects. Various influencing factors contribute to this harmful situation like accidental out of control (Figure 1-2), hacker influenced threats or hardware/software malfunction within the drone. In contrast without sacrificing the core functionality of the drone, bio-inspiration aspects indicate that the flapping wing presents unique maneuverability and ensures miniaturizing becomes more reliable. Such exquisite capabilities of insects in hovering and agile maneuvers have led to the bio-inspired MAVs for application in confined spaces, and surveillance purposes.

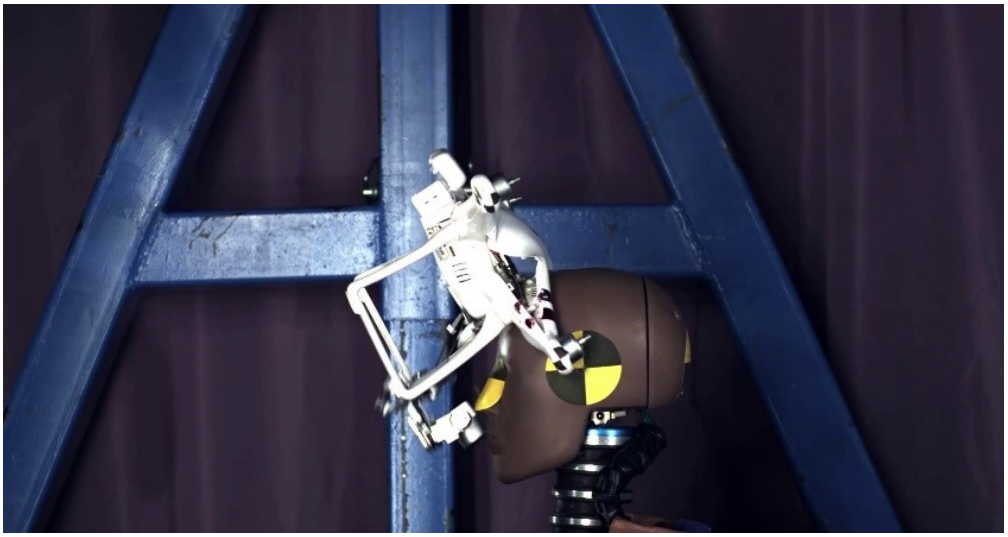


Figure 1-2: Indoor-scenario drone crash indicating potential damage to humans [2]

1-3 Atalanta

Moving in the direction towards building a fully autonomous FWMAV, the project was named after the Vanessa Atalanta butterfly. The project primarily had two objectives, to maintain a low total mass that requires lightweight components, and operate on low power consumption. The body of the MAV was designed with four wings and actuated by a compliant ring structure that resonates to reach the state of flapping wing frequency of 28.5 Hz [3]. With a wingspan of 100 mm and a body mass of 4 grams, Atalanta was aiming to hover and achieve (slow) indoor flight. The physical representation of Atalanta can be seen in Figure 1-3 with the explained description. Earlier the studies focused on the making use of conventional technologies like linear actuators, gears electric motors and compliant links [16] [17] [18] [19] and miniaturizing them. Among the various flying maneuvers, the most demanding task involves the hovering action. This may seem the most simple and basic flying state, but the most challenging task as it is highly demanding in terms of energy perspective. A sensing system is required to understand the external environment and to control the FWMAV that needs to be light and reliable.

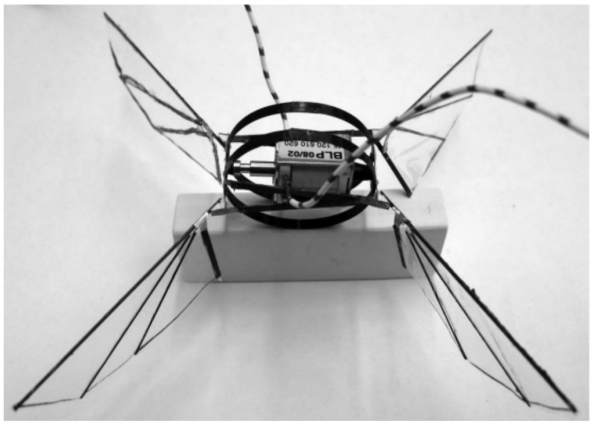


Figure 1-3: MAV Design of Atalanta project [3]

1-3-1 Application

There exist a wide range of applications for an autonomous FWMAV like Atalanta. The prime focus lies in the fields of swarm robotics, where a group of robots interact in a specific behavior. Atalanta is most suited for indoor-scenarios to undertake different types of measurements. Apart from the regular measurements like temperature, humidity, and others, an interesting application could be related to detecting a specific chemical presence. This can be very useful in warehouses or airports to target and surround a plausible suspect. An underlying advantage of FWMAVs is the potentiality to use them safely in a densely human-interacting space. Similarly, custom-made applications can be developed on the successful development of autonomous FWMAVs.

Chapter 2

Present Study - Focus Area and Expected outcomes

This section begins with understanding the general architecture of an FWMAV and arriving at the research questions. A summarised description of the literature findings is presented here to understand the gaps and challenges. Bio-inspiration has proven to provide more insights into arriving at the research goal. The final section deals with providing more insights into the thesis objective and approach

2-1 General Architecture - FWMAV

On completing the literature survey and in the process, a generic pattern of research was observed that comprised mainly focusing on five major areas that were Wing configuration, Sensing, Computing, Actuation, and Flight control as seen in Figure 2-1. The highlighted

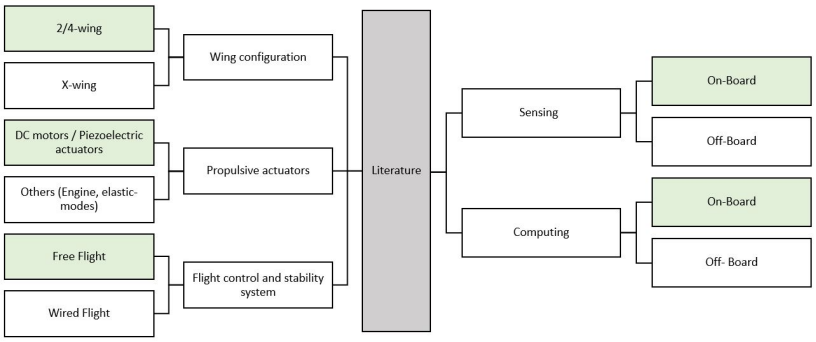


Figure 2-1: Proposed Major Classification of FWMAV according to Literature survey

green area indicates relevance to Atalanta. These areas were further analyzed by understanding the existing state-of-the-art inventions to make Atalanta autonomous. Especially the past-work done by Selvan [20] and Daan [21] helped proficiently in understanding the grey areas on time.

2-1-1 Related works - State of the Art

Enormous developments have been continually occurring in developing autonomous FWMAVs. Around 51 FWMAVs including the Atalanta have been developed across the world [22]. The general classification is based on their sensing and processing capabilities that are On-board or Off-board, and also if they are Indoor or Outdoor autonomous FWMAVs. Table 2-1 shows that only 14 FWMAVs amidst 51 are capable of free or wired flight that have complete On-board characteristics similar to real-time flying insects. Also, conclusions show that no off-board sensing and on-board computing trends have been tried so far for many practical reasons.

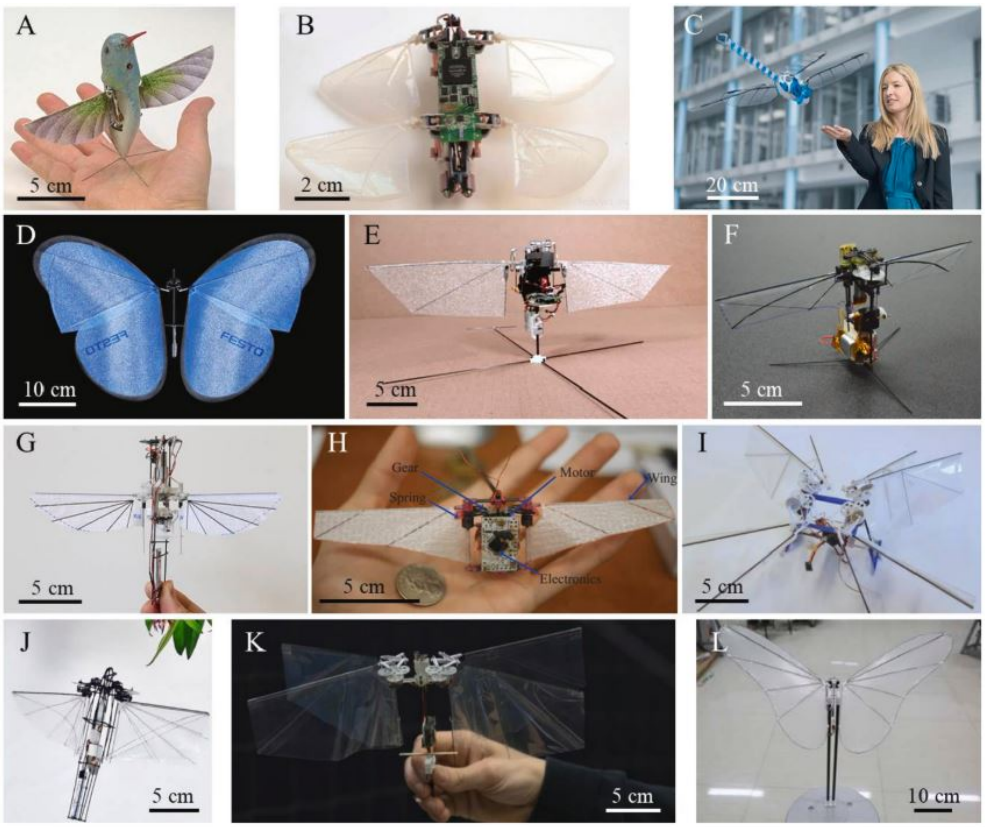


Figure 2-2: Recent Developments in tail-less FWMAV (A) Nano Hummingbird - AeroVironment Inc. (B) TechJect Dragonfly - TechJect Inc. (C) BionicOpter - Festo AG & Co. KG. (D) iMotionButterflies - Festo AG & Co. KG. (E) Robotic Hummingbird - Texas A&M University. (F) KUBeetle - Konkuk University. (G) Colibri robot - Universite Libre de Bruxelles. (H) Robotic Hummingbird - Purdue University. (I) Quadthopter - Delft University of Technology. (J) NUS-Robobird - National University of Singapore. (K) DelFly Nimble - Delft University of Technology. (L) Butterfly-type Ornithopter - Beihang University - [4]

Various MAVs have played a vital role in understanding the autonomous nature listed in Table 2-1. Delfly nimble [23] is a fully-autonomous FWMAV that weighs 28.2 grams and has four wings with a span of 33 cm and operating at 17 Hz that can perform extremely agile maneuvers and hover. The tailed predecessor Delfly II can fly up to a kilometer at 11 kmph (max 25 kmph) and carry payload up to 4 grams that was used by the stereo vision cameras. In the earlier version named Delfly Explorer [24], a novel onboard Stereo vision algorithm designed for flapping wing motion and tuned for the operating frame-rate made autonomous flights possible including collision avoidance.

SNo	Name	Year	Mass	Wings	Span	f	FC
1	Nano Hummingbird [25]	2007-11	19	2	165	30	Free
2	KU Beetle [26]	2009*	21.4	2	160	30.5	Free
3	Purdue Hummingbird [27]	2011*	12	2	170	30-40	Wired
4	TechJect [28]	2012	25	4	150	-	Free
5	Daedal Flapper [29]	2012	12	2	-	-	-
6	Colibri [30]	2012*	22	2	210	22	Free
7	Bionic Opter [31]	2013	175	4	630	15-20	Free
8	Robotic Hummingbird [32]	2015*	62	2	305	22	Free
9	Quad-thopter [33]	2018	37.9	8	280	15	Free
10	Delfly [34]	2018	19.7	4	280	15	Free
11	NUS-Robird [35]	2018	31	4	220	13.3	Free
12	Delfly Nimble [23]	2018	28.2	4	330	17	Free
13	Butterfly ornithopter [36]	2018	38.6	2	648	14	Free
14	RoboBee [37]	2007*	0.06-1	2	25-30	110	Wired

Table 2-1: Recent developments in FWMAVs relevant to Atalanta
Note: *Mass in grams, Wings - Number, Span - Wing Span in mm, f : frequency in Hz, FC: Flight Capability, ACM: Attitude Control Mechanism, * - in-progress until 2020-Jan*

Most of the successful models have proven to use unconventional approaches that are common in nature. Understanding this aspect better helps us to gain a holistic perspective in developing an autonomous MAV.

2-2 Bio-Inspiration in MAVs

'The notion of a biological principle is taken in a broad meaning, ranging from individual biological features like the anatomy of perceptive organs, models of information processing or behaviors, to the evolutionary process at the level of the species.'

-Zufferey Jean Christophe

It was observed earlier that most of the advanced technologies, draw inspiration from nature to develop an FWMAV as seen in Figure 2-1-1. This section verges more into the aspects of bio-inspired engineering. Once the classical approaches of robotics cease to satisfy the MAV needs, inspiration drawn from nature provides many inquisitive solutions. Insects like house-flies have been performing lightweight, highly efficient flight maneuvers that have developed and evolved over the ages. Biological inspiration from the flies have pushed the boundaries of innovation further, as they do not use GPS for their flights. Bio-inspiration in the current context pertains to retaining the core functionality of vision-intelligence that needs reverse-engineering for the re-creation of the crux idea. The visual-intelligence for flies mainly indicates using low-resolution with very-high processing power and wide FoV (Field-of-view). Also, the body hair interacting with air serves as speed sensors enabling this combination to perform complex maneuvers in elegant ways [38]. The daunting challenge lies in the manufacturing domain to make these subsystems via MEMS Technology (Micro Electro Mechanical Systems).

2-2-1 Optical-flow in Insects

Insects are exoskeleton-invertebrates that belong to one among the 4 class of arthropods in the phyla kingdom. These possess compound eyes for the vision that are a build-up of repeated units called *ommatidia* that have their receptors and lens. Diptera (flies) and Hymenoptera (bees) are the most successful arthropods [5] and hence chosen for inspiration. Many complicated tasks such are controlling flight speed, measure self-motion, track-substance, and distance-traveled estimation are executed by flies by using visual feedback [39]. From Figure 2-3, an overall idea of fly's vision and control systems can be gathered. The technical aspects of these are looked into in the subsequent optical flow Section 3.

The vision-system of flies bear a variety of difficulties such as immobile fixed-focus optics, closely positioned eyes (implying poor estimation of spatial and distance information), and much inferior Field-of-View (in the range of 5^o inter-ommatidial angle which is 60 times lesser than humans) [5]. Above all, flies overcome these hurdles with their impeccable temporal sensitivity that ranges up to 200 Hz that is almost 10 times more than Humans and the juxtaposition-arrangement of divergent ommatidia to gain larger Field-of-View, operating well in synchronous with the reflex system. All these shreds of evidence motivate to develop a sensing system with these characteristics to support light weight MAVs. Also, the research indicates absence of clear boundaries for sensing and conrtol operations.

Optic flow information from the insects contains both speed and direction. The surroundings closer to sensing seem to move faster than the objects at a distance. This feature enables us to develop interesting control phenomenons for object collision avoidance. As the vector flow becomes more rapid due to the approaching target, the trajectory can be re-planned by using

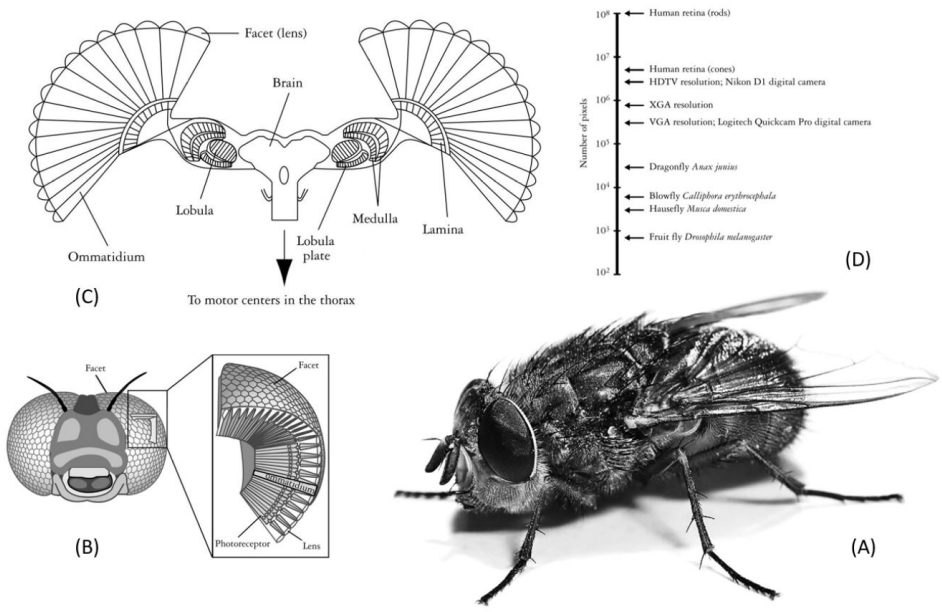


Figure 2-3: House Fly - Bio-inspiration. (A) **Blowfly Calliphora** - Highly capable flying insect (B) Compound eyes made of repeated *ommatadia* units with hexagonal tiny lenses aka Facets at the top and transparent cone-like crystalline structure arranged in honeycomb-like radial pattern. (C) **Central nervous system of flies** - Photoreceptor signals are transmitted to lamina that accentuate temporal changes; and mapping of visual inputs from retina to neurons are maintained via medulla. Finally the signals from Lobula Plate (made of wide-field motion sensitive tangential neurons) actuate the wings (via thoracic ganglia and optic lobe) (D) Pixels range comparison of various Vision Systems, Flying insects pixel resolution are much below silicon imagers [5]

relevant controls and executed by manipulating the thrust at each wing. A general overview of various flying conditions can be observed in Figure 2-4. The morphological stature of insects shows passive stabilization to some extent as the wings are positioned slightly above the center-of-gravity. The attitude control is implemented by adhering with the horizon, always maintaining a brighter upper side than the ground, in outdoor daylight conditions [5]. Optical flow data can provide multiple information including the velocity and proximity to objects. This motivates us to explore deep insights into available methodologies to sense them. The hardware proves an essential role in sensing and the various sensors are analysed for using in Atalanta.

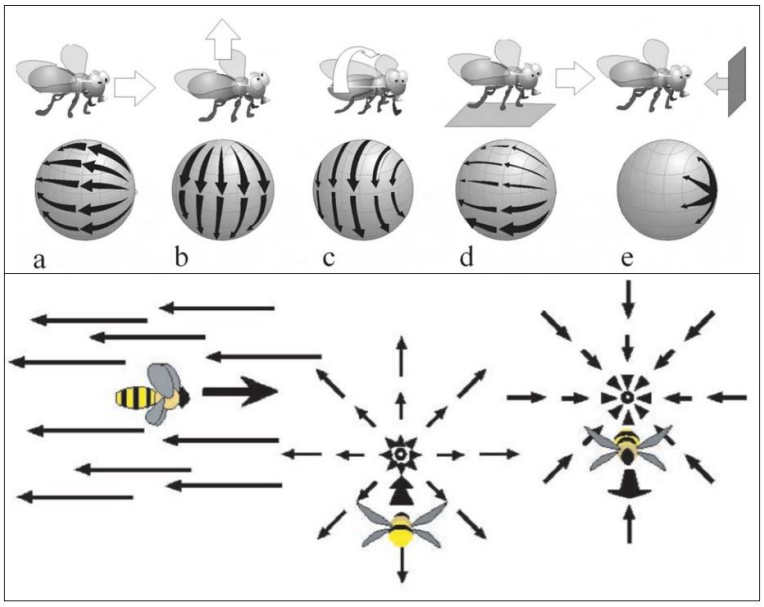


Figure 2-4: Optical Flow perceived by insects; **Top**: the lines represent the field vector of motion in different scenarios starting from forward, upward, left roll, speed difference due to surface and outward flow respectively [6]; **Bottom**: Panoramic optic flow from perspective projection camera, the flow pattern linearly varies from backward at sides, top and bottom to expansion and contraction at front and rear [7]

2-3 Hardware for Sensing

Unlike heavy active sensors such as laser scanners or lidars, the vision-related hardware remains to be the most suitable solution for sensing. These systems are usually light, passive, consume much lesser energy, and provide maximum information of the environment including larger FoV (Field of view) at once.

The optical flow sensors are mainly classified into three major categories seen in Figure 2-5. The first category consists of conventional cameras, which capture the live motion in a sequence of frames. In the past, many researchers have used a combination of cameras to extract the optical flow data from the images [40] [41]. Lucas-Kanade [42] and Horn-Shunk [43] strategies from the past have largely contributed to obtaining the OF data from images. The need for a separate image-processing environment makes the system bulky and more demanding in terms of power and efficiency. Also, Section 2-2-1, shows the need for high capture rates (200Hz). In contrast, conventional cameras possess limited frame capture rates around 30-40 frames per second.

The next category of sensors that are used mostly in the field of robotics can are custom made as per the specific requirement. These sensors are in-house developed using MEMS technology and mainly built based on the application requirement. Few examples include the fabrication of analog VLSI technology in millimeters scale for optical flow sensing [9] and imitation of insect compound eyes [44]. These sensors contain much greater processing power, operate a wide range of light, and consume very low power. As seen earlier these sensors mostly require processing for OF extraction.

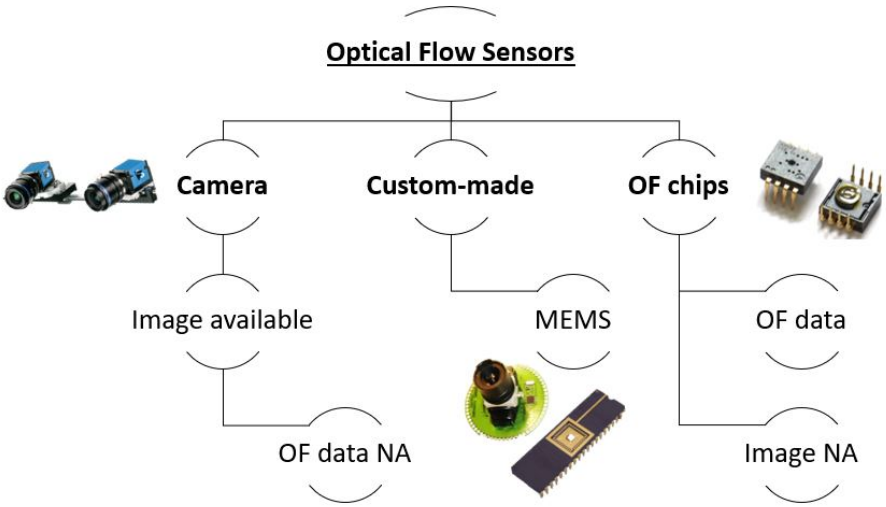


Figure 2-5: Classification of Optical Flow sensors

Finally, there exists a class of sensors that directly produces the OF vectors from the sensed environment. The system usually contains an image acquisition system to obtain images, and digital signal-processing unit to extract the OF data. Few known examples of these include optical mouse sensors and other commercial OF sensors such as px4Flow as seen in Table. 2-2. The dedicated OF sensors have a larger focal range and operate at a much higher resolution. This consumes more power and processing that makes the package heavy. The untapped potential from a common mouse sensor can be unconventionally used to detect the optic flow. The Optical-mouse sensor is light weighing and power-efficient with a higher frame capture rate up to 12kHz. The challenge here is to improve the optical range from a few millimeters to accommodate indoor scenarios. They are also known to operate efficiently based on the illuminance available.

Sno	Sensor	Dimension	Mass	Voltage	FPS	Inference
1	ADNS 9800	Ø 32.5 x 5	2	3.3/5	12000	200cpi
2	PMW 3360	28 x 21 x 5	-	3.3/5	12000	100cpi
3	3901U	14 x 11 x 5	0.6	3.3-6	66	Baud rate 19.2k
4	PMW 390/1	Ø 18 x 10	∼1	1.8	100	-
5	VL 53L1	Ø 18 x 8	-	∼2	50	Combination
6	FPV RC	Ø 13 x 20	∼1	2.3/5	25	Baud rate 19.2k
7	UP LC302	Ø 22 x 14	>5	3/5	-	Medium 2k
8	Px4	45.4 x 32	∼5-10	5	250	752x480 Pixels
9	HereFLow	-	-	5	-	Lidar-4PMW3901
10	Px4Flow	Ø 33 x 38	-	5	400	Sonar-px4 921.6k
11	OpenMv	45 x 36 x 29	17	5	-	-
12	Adns 3080	22 x 9 x 8	-	3.3	6400	-

Table 2-2: Compilation of Various Sensors used to detect Optic flow **Note:** Dimension in mm, Mass in grams, Voltage in Volts, FPS - Frames Per Second, cpi - counts per inch

The foremost criterion considered in the search for relevant sensors was the monolithic existence of sensing and processing unit, higher capture rates, and weighing less. Considering the various sensors, the optical mouse sensor ADNS 9800 [45] was chosen mainly due to a very high processing rate, light-weight, programmable feature, and good documentation.

Optical Mouse as OF Sensor

The mechanical mouse discovered earlier uses a spherical ball to find the relative motion and thereby alter the pointer-position on screen. As the ball moves, the motion sensors/encoders detect the current position and enable them to navigate on the screen. Since the introduction of optical-mouse, the need for mechanical components and reliance on a surface used was drastically reduced. Optical mouse captures the sequence of images below the surface and tracks the contrast features in subsequent images that are processed for optic-flow data thereby enabling to know the displacement in X and Y coordinates. Laser mouse became more popular due to the higher wavelength of the light used to observe the image and hence better accuracy.

In the past, since 2004 various researchers have used optical mouse in different fields like robotics, positioning and measurement applications as seen below,

- **2004** *Thakoor et al* [7]: Use of ADNS 2051 16×16 pixels and used in combination with various sensors for UAVs in Mars exploration where air is thin and gravity is low.
- **2007** *Griffith et al* [46]: Three Agilent ADNS-2610 with 18×18 pixels is used as complementary distance sensors navigation in mountainous canyons combined with inertial measurement unit (IMU) and GPS.
- **2012** *Goosen* [6]: This paper represents a system of 6-optical mouse sensors combining with simple controls for actuating four wings.

  - **2014** *Selvan* [20] [47]: The optical mouse sensor ADNS 9800 was used to detect optical flow under extreme lighting conditions and simulation was executed based on the control law.

  - **2018** *Daan*[21]: Flight control achieved for quad-copter setup in linear and hovering conditions.

  The following studies use ADNS 9800 in different fields within the operational focal range of 2-3mm,

- **2016** *Alvarez et al* [48]: Used in stage-positioning measurement system for microscopic application. The system is made reliable by using optical magnification, periodic error rectification and compensation for undesired dynamics.
- **2016** *Liu et al* [49]: Used for rotational-speed measurement from a motor by measuring the air-gap change and using Fourier transformation to calculate the speed.
- **2019** *Wendler et al* [50]: This study uses ADNS 9800 to measure the velocity of water in flooding situations. Initially the sensor characteristics are analyzed and to ultimately detect the flow velocity.

2-4 Research Question

A research question focuses on the exclusive problem(s) aimed to be solved in the thesis within the given time frame. There exist multiple challenges to be solved for Atalanta robotic-fly to make it autonomous. Daunting research has been on-going for almost a decade and significant progress has been accomplished. Among the various contributing factors to enable flight, the current study focuses on solving sensing and control related issues. The final realization of smooth flights in Atalanta could be achieved with the advancement in manufacturing technologies. The most influencing factors in terms of proofs-of-concept are looked into this research. In the past various approaches have been followed to initiate vision sensing and controls using pure optic-flow data. Most of these approaches focused on using complete or partial simulation to predict the flight behaviors in Atalanta. It has been concluded that the ADNS 9800 optical mouse sensor used is highly dependent on extreme lighting for sensing. The previous sections infer that the optical-mouse sensor remains as the potential option to be used. It is believed that the sensor could deliver better results than intended.

Hence this study continues to explore the possibilities to enhance the sensitivity of the sensor using a different approach and to analyze the controls developed by Goosen. [6].

The following questions are analyzed in the research

- ***Can ADNS 9800 be used as a optical flow sensor in indoor application?***
- ***Is there a method to analyze the pure Optic-Flow based control strategies using real-time data?***

2-4-1 Objective and Approach

The research question is answered in the upcoming sections. Initially, the major objective of the study focuses on developing a reliable SWaP satisfying sensor for indoor applications. Since Optical flow is the crucial data being handled in the research, an in-depth understanding of the concept and the relation with sensing parameters are explored in Section 3.

To answer the first research question, ADNS 9800 optical mouse sensor was evaluated for use in robotic applications in Section 4. The initial focal range of the sensor was increased from a few mm to accommodate the necessary distance. Later the modified sensor was used to detect optic flow in ambient indoor conditions. An innovative approach is handled to use a digital screen to detect the optic flow. This enables quick modifications in experimental setup for different scenarios. Since the sensor can be varied to use different frame rates to capture data, it was matched with the computer refresh rate that is around 60 Hz. Finally, a Hardware-in-the-loop setup was developed using multiple sensors for the simulation of controls. A brief comparison with the Software in the loop setup was done using the ROS-Gazebo environment. The signals are identified to be controllable and stable. The details related to this are discussed in Section 5. This general setup can be used for various applications to verify the control strategies.

Chapter 3

Optic Flow - Theory and Sensing

This chapter focuses on explaining the concepts of Optic-Flow in general and relating it to the current context. Initially, the type of information obtained from optical flow is briefed and followed by understanding the dependency of motion-type. The basic concepts for optic-flow involving transnational and rotational motion of a ideal sensor is explained.

In 1925, Helmholtz described optical-flow from an exciting experience of an observer moving in the woods:

"Suppose, for instance, that a person is standing in a thick woods, where it is impossible for him to distinguish, except vaguely and roughly, in the mass of foliage and branches all around him what belongs to one tree and what to another, or how far apart the separate trees are, etc But the moment he begins to move forward, everything disentangles itself, and immediately he gets and apperception of the material contents of the woods and their relations to each other on space, just as of he was looking at a good stereoscopic view of it.

Objects that are at rest by wayside, appear to glide past us in our field of view in the opposite direction to that in which we are advancing. More distant objects do the same way, only more slowly, while very remote bodies like the stars maintain their permanent positions in the field of view. Evidently, under these circumstances, the apparent angular velocities of objects in the field of view will be inversely proportional to their real distances away; and consequently, safe conclusions can be drawn as to the real distance of the body."

– ***Helmholtz*** [51]

3-1 Optic Flow

The context described by Helmholtz reveals the richness of information encoded in image motion (Section 3) that laid the foundation for Optical-flow and its related concepts. It can be explained as the pattern of apparent motion of features (surfaces, objects, or edges) caused by the relative motion of the observer and the scene. The object's distance from the observer, can be entangled and estimated knowing the self-motion of the observer. The optic flow data consist of the linear and angular velocities with the distance information from the obstacle.

3-1-1 Optical flow sensors - Information Obtained

The visual representation of the optical flow data viewed by the observer containing vectorial data is seen in Figure 3-1. The system can be viewed as a spherical camera with an optical system. Three-dimensional data from the surrounding environments are projected onto the two-dimensional surface plane of the camera. These static points on the image plane move in a specific direction with magnitude when the observer moves in spatial arrangement. These vectors shall be defined as the optical flow vectors which reveal the roll, pitch, or yaw motion depending on the observer motion. An example of the roll motion observed by the spherical camera can be observed in Figure 3-1. These vectors move in the opposite direction to the movement of the observer. Also, the larger vectors are present near the Field-of-view, and its magnitude decreased as the view shrinks. In general, optical flow sensors possess operating thresholds in terms of visual contrast and aperture for detection of flow from the surroundings. The amount of ambient light available impacts the contrast to a certain extent and often the right combination results in efficient sensing. On the processing aspect complex algorithms are more consuming and the variation in actual and predicted optical flow exists. These factors hinder the performance and result in imperfections in an actual spherical sensor.

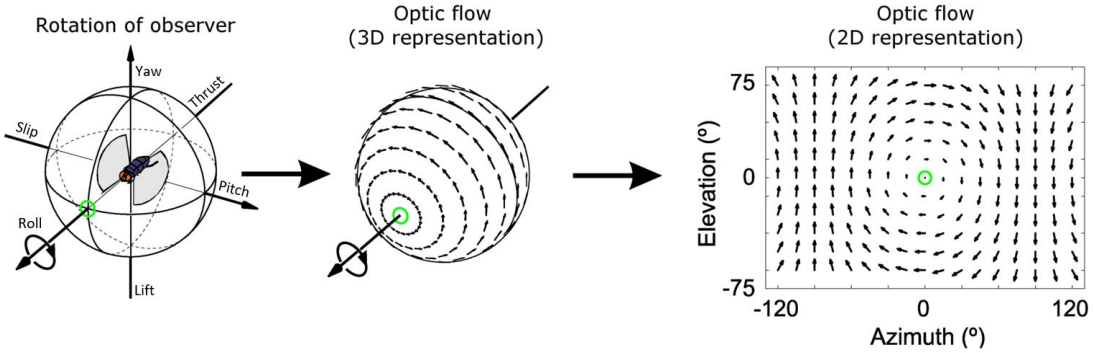


Figure 3-1: Visual representation of Optic-Flow Vectors (direction and magnitude) as experienced by a fly (observer) while moving. Left: image depicts the possible motion of observer around all the axes including the lift direction. Center: image shows Optical-flow vectors occurring due to pure rolling motion. Right: visualization of Center image in two-dimensional view by bisecting the Yaw-Pitch plane [8]

3-2 Ideal Optic flow sensor

An ideal sensor can be considered to be perfectly spherical and has an infinite spatial resolution. All the interventions caused by external and internal conditions such as luminance, sensitivity, and image-processing are neglected. This ensures a proper understanding of the actual scenario and to know how the optical flow data can help the control system. Currently, it is assumed that the optical-flow, is purely yielded by the motion of the observer in a stationary environment.

The upcoming sessions delve more into the mathematics involved during the translation and rotation motion of the observer/camera. The following sections deal with vectors and their projection. A brief explanation of vectors and the required concepts are quickly refreshed using the basic configuration as described in Figure 3-2. Consider two vector l and m separated by an angle α, where the projection of vector l on m is indicated by n and points in the same direction as m.

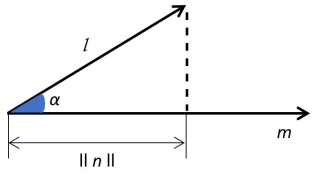


Figure 3-2: Illustration of two vectors at an angle α with projection of the vector *l* on *m*

The scalar product also known as dot product for the two vectors l and m can be defined as,

$$\vec{l}.\vec{m} = \|\vec{l}\| \, \|\vec{m}\| \, cos(\alpha) \tag{3-1}$$

Where the cosine compound can be expressed as,

$$cos(\alpha) = \frac{\|\vec{n}\|}{\|\vec{l}\|} \tag{3-2}$$

On substituting and rearranging, this simplifies the dot product to,

$$\vec{l}.\vec{m} = \|\vec{l}\| \, \|\vec{m}\| \, \frac{\|\vec{n}\|}{\|\vec{l}\|}$$

$$\|\vec{n}\| = \frac{\vec{l}.\vec{m}}{\|\vec{m}\|} \tag{3-3}$$

The projected vector n can be derived by multiplying the unit vector along direction of m,

$$\vec{n} = \|\vec{n}\| \, \frac{\vec{m}}{\|\vec{m}\|}$$

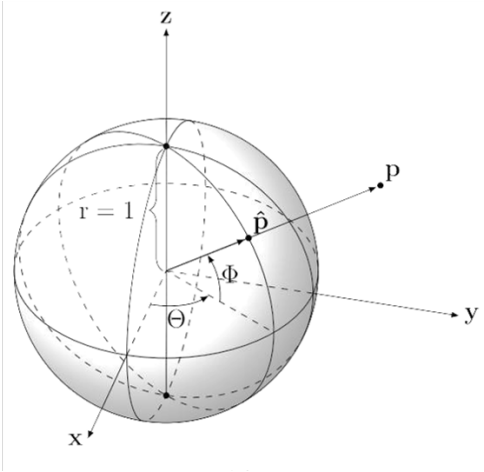


Figure 3-3: Illustration of spherical sensor with unit radius, (a) projection of a arbitrary point p on sphere is denoted by p̂ and can be described by θ, ϕ and unit radius [9]

$$\vec{n} = \frac{\vec{l}.\vec{m}}{\|\vec{m}\|^2}\vec{m} \tag{3-4}$$

Comparing to the current context, these derived equations for the projection of vectors enable estimation of optical flow vectors for a spherical sensor. A formal description of the configuration of this sensor can be observed in Figure 3-3. The sphere of the radius (r) is assumed to be unity and any point (p) in the environment of the sensor can be projected on the sphere. This point p as shown in 3-3 can be described using azimuth angle θ and elevation ϕ. A unit vector in the direction of p can be described using p̂ whose length is unity at r = 1 represented by the equation 3-5.

$$\hat{p} = \frac{1}{\|p\|}\, p \tag{3-5}$$

3-2-1 Translation motion - Optical flow

This section deals with considering the translational motion of the spherical sensor with an origin located cartesian coordinate system. A linear motion of the sensor along Z-axis is being indicated by T in Figure 3-4, where the arbitrary point on the sensor-environment p_1 shifts to p_2 moved by the same distance T, but in opposite direction. The relative optic flow on the sensors surface sphere shifts from $\hat{p}_1$ to $\hat{p}_2$ and by the normal distance $\hat{T}$. The equations 3-6 and 3-7 describe the relation between the points in environment and on the sensors surface respectively. The projection of the translation (T) in the sensor's environment is denoted by the $\hat{\mathrm{T}}$ on sensors surface.

$$\vec{p_2} = \vec{p_1} - \vec{T} \tag{3-6}$$

$$\hat{T} = \hat{p_1} - \hat{p_2} \tag{3-7}$$

The translation of the arbitrary point on the sensor surface is denoted by Eq. 3-7, The precision of the sensor surface influences the estimation of the trajectory of the projected point. The optical flow vector field can be defined as the velocity field whose value changes at each point.

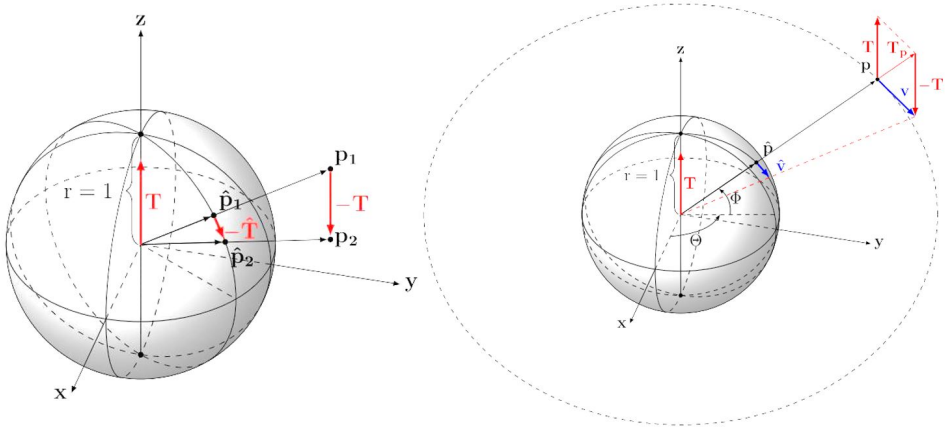


Figure 3-4: Left: a linear transnational (T) motion along z direction inducing optical flow in opposite direction $\hat{\mathrm{T}}$ on the unity sphere. Right: Illustration of the the optical flow vector $\hat{\mathrm{v}}$ perpendicular to the projected point $\hat{\mathrm{p}}$, including T_p - projection of T on $\hat{\mathrm{p}}$

The velocity of each projected point is perpendicular to the surface as seen in Figure 3-4. The point p moves at velocity v, similarly, the point $\hat{\mathrm{p}}$ on sensors surface induces a velocity $\hat{\mathrm{v}}$. This can be calculated from the original point p and translation T.

$$\hat{p_1} = \frac{\vec{p_1}}{\|\vec{p_1}\|} \tag{3-8}$$

$$\hat{p_2} = \frac{\vec{p_2}}{\|\vec{p_2}\|} = \frac{\vec{p_1} - \vec{T}}{\|\vec{p_1} - \vec{T}\|} \tag{3-9}$$

Also, v and $\hat{v}$ are $\perp$ to p and $\hat{p}$, which implies the velocity v can be derived by projecting T on $\hat{p}$ as seen in Figure 3-5

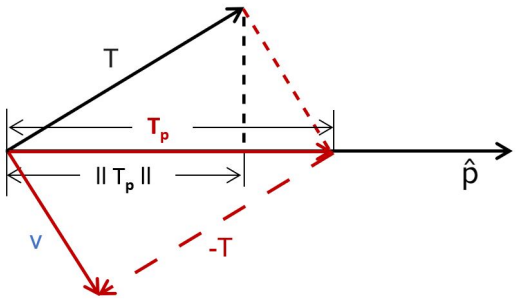


Figure 3-5: Vectorial representation: Projection of T on $\hat{p}$ denoted by T_p where v $\perp$ $\hat{p}$

$$\vec{v} = \vec{T_p} - \vec{T} \tag{3-10}$$

Similar configuration is observed earlier from Eq. 3-4, where $l = \mathrm{T}$, $m = \hat{\mathrm{p}}$ and $n = T_p$ given by,

$$\vec{T_p} = \frac{\vec{T}.\hat{p}}{\|\hat{p}\|^2}\,\hat{p} \tag{3-11}$$

Now substituting Eq. 3-11 in Eq. 3-10 with $\|\hat{\mathrm{p}}\| = 1$

$$\vec{v} = (\vec{T}.\hat{p})\hat{p} - \vec{T}$$

$$\hat{v} = \frac{1}{\|p\|}\,[\,(\vec{T}.\hat{p})\,\hat{p}\; -\; \vec{T})] \tag{3-12}$$

The general equation for the optical flow velocity on the sensor surface is given by the Eq. 3-12 in case of a pure translational motion.

3-2-2 Rotation motion - Optical flow

This section deals with optical flow caused by the pure rotational motion of the sensor. The system functions based on the polar coordinate system, wherein any arbitrary point p on the sensor environment can be depicted using azimuth angle θ and elevation angle ϕ. This can be visualized as seen in Figure 3-6 where R indicates rotation orthogonal to the plane of rotation in the Cartesian coordinate system. The rotation along the x-axis induces motion in the sensor's environment, thereby the surface and causes an optical flow vector $\hat{v}$. This can be expressed as the cross-product of R and the projected point $\hat{p}$.

$$\hat{v} = -\vec{R} \times \vec{P} \tag{3-13}$$

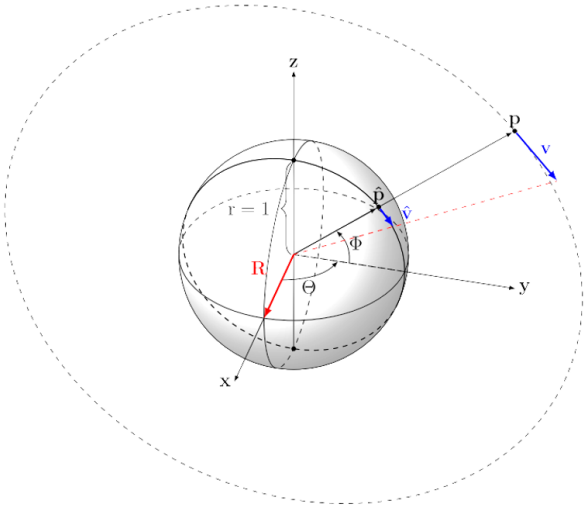


Figure 3-6: Influence of pure rotation of sensor on angular velocity

Chapter 4

Paper I - Sensor performance Evaluation

Most of the theory relating with optic-flow have been seen before. This chapter focuses more on the evaluation of hardware involved for optic-flow detection. This section focuses more in answering the first research question i.e. can ADNS 9800 optical mouse sensor be used a as optical flow sensor?

It explains the working of ADNS 9800 optical mouse sensor and gaps and challenges in using them in current scenario. Few necessary modification were implemented to achieve required functioning and the sensor was evaluated based on various parameters that explicated in results.

Evaluation of Optical flow sensors for Indoor robotics application and Flight simulator

J.G. Suresh Subramonian*, J.F.L Goosen**
Department of Precision and Microsystems Engineering
Delft University of Technology
2628 CD Delft, The Netherlands
*j.g.sureshsubramonian@student.tudelft.nl, **j.f.l.goosen@tudelft.nl

***Abstract*—**
Visual guidance has proven a very significant role in the fields of autonomous flight controls. Optical-flow based solutions provide map-less navigation strategies, especially for miniaturized robots that possess stringent size, weight, and processing (SWaP) constraints. The current study focuses on the evaluation of the compact ADNS 9800 (optical mouse sensor) for use as an optical flow sensor in autonomous robots for indoor applications. Initially, the sensor programmed to detect optic flow and enhanced with suitable optics to improve the existing focal range from 3mm to a few meters. A customizable Matlab GUI is programmed to generate contrasting optic-flow patterns that are detected from the digital computer screen by the sensor. After necessary modifications, the sensor data has proven to be stable and controllable. The sensitivity has improved below 70% brightness and contrast in the digital optical-flow environment. A brief comparison with analog input is executed. The results are supportive to build a Hardware-in-the-loop flight simulator.

Keywords: Optic-flow, mobile robots, visual navigation, HITL, optics, and controls.

I. INTRODUCTION

Autonomous navigation in robots can be considered as the process of charting a safe trajectory from the current state to the final destination [1]. Optical flow signals are one of the widely accepted means of achieving map-less flight controls.

Optic-flow can be perceived as the visual motion of objects, surfaces, and edges as the observer moves relative to them (or) defined as the distribution pattern of apparent velocities of brightness in an image [2]. The optic-flow is obtained as vector field data that contains the direction and magnitude. Since it contains information regarding self-motion and the surrounding environment it can play a vital role in navigation. In general, a sensor captures the images as frames and are interpreted by a Digital Signal Processing module [3]. There exist a spectrum of sensors to realize these signals viz., discrete electronic circuits, digital cameras with processing [4], single-chip analog motion processors, and optical mouse chips [5]. Usually, the signal processing of the sensed image data for optical flow extraction is computationally expensive. The current study focuses on developing a vision sensing system based on optical flow signals for the controls of miniaturized robots. Hence among these various sensors available, optical mouse chips are selected as the area of interest considering the Size Weight and Processing (SWaP) constraints. From the past, shreds of evidence show that optical mouse sensors have been used in various robotics applications such as odometry, navigation, encoders and to estimate acceleration [5].

The current work investigates the prospects of using the ADNS 9800 optical mouse sensor as an optical-flow sensor to stabilize a Micro Aerial Vehicle. This work mainly pertains to support the sensing task for MAV controls developed by Goosen [6]. The FWMAV in the study consist of three pairs of optical mouse sensors placed on the six Cartesian co-ordinates. The proposed control-logic uses simple arithmetic operators to drive the control actuators and achieve robot stabilization. The study aims to extract the optic flow vectors from the sensor. In its base application, the mouse sensor operates for a focal range of 3mm, and the primary objective aims to extend this range to an indoor scenario and understand the effects. The secondary objective aims to develop an experimental setup to study the behavior of the sensor in the digital environment under various conditions. These results help us in deciding the possibility to develop a flight-simulator for MAV applications that use optical flow. To achieve this, a compact experimental setup is developed using the sensor to detect the known optic flow on a PC monitor. The optic flow is generated by a MATLAB program developed to test various linear motions of known velocities.

An overview of the remaining paper is organized in the following pattern, wherein Section II exposes the basic working principle of ADNS 9800 followed by explaining the necessary experimental setup required for sensor evaluation. Section IV helps to understand the theory behind sensed data. Finally, the crucial results obtained are discussed that include linearity and sensitivity for the sensor. Also, the current bandwidth of working conditions relating to the minimum brightness and contrast required has been substantially improved and discussed.

II. OPTICAL FLOW SENSOR

A. Optical Flow sensors - Major Classification

This study focuses on identifying an optical flow sensor that resembles that of a flying-insects eye. Despite their much lower image-resolution, they achieve significant efficiency in visual sensing due to the very high temporal frequency in

processing (around 200Hz) [4]. This enables flying insects to be highly receptive to even minor changes in the visual field (especially to detect optic flow). Therefore, the key requirements for the ideal sensor were considered to be low weight and power consumption with a very high processing rate. The optical flow sensors are mainly classified into three major categories.

1) Camera: The first category consists of conventional cameras, which capture the live motion in a sequence of frames. In the past, many researchers have used cameras to extract the optical flow data from the images [7] [8]. Lucas-Kanade [9] and Horn-Shunk [2] strategies have contributed to obtaining the OF data from images. The need for a separate image-processing environment makes the system bulky and more demanding in terms of power and efficiency. The conventional cameras possess limited frame capture rates around 30-40 frames per second which hinders the usefulness of Optical flow extraction.

2) Custom-made: The second category of sensors that are used mostly in the field of robotics can are custom made as per the specific requirement. These sensors are in-house developed using MEMS technology and mainly built based on the application requirement. Examples include the fabrication of analog VLSI technology in millimeters scale for optical flow sensing [10] and imitation of insect compound eyes [11]. These sensors possess much greater processing power, operate in a wide range of light conditions, and consume very low power. The major drawback of these sensors is the expensive nature and time-consuming process.

3) OF Chips: The final category that interests us the most is the dedicated OF sensor chips that directly produces the OF vectors. The system usually contains an image acquisition system to obtain images, and a digital signal-processing unit to extract the OF data. Known examples of these include optical mouse sensor and commercial px4Flow sensors used in drones. Interestingly the mouse sensor is light and power-efficient with a much higher frame capture rate up to 12kHz. The challenge here is to improve the focal range from a few millimeters and to extract useful optic-flow information in an indoor scenario.

B. *ADNS 9800 - Optical mouse sensor*

ADNS 9800 optical mouse sensor was selected as the OF sensor as it mostly agreed with the intended requirements from Section II-A and has good documentation. The standard ADNS 9800 gaming mouse sensor is in a 16-pin integrated single chip-on-board package which includes the sensor and vertical-cavity surface-emitting laser (VCSEL) with advanced programmable features for frame rate and resolution. The sensor is capable of detecting high-speed motions up to 3.81 m/s, which is slightly more than the forward cruising speed (3m/s) of autonomous FWMAV Delfy nimble [12]. This enables the sensor to operate safely in an indoor scenario up to the detectable speed. The presence of 16-bits motion data registers enables quick maneuvers of the MAV for accelerations up to 30g. Enhanced programmability of the sensor helps to attune it as per the needs that provide OF/output resolution up to

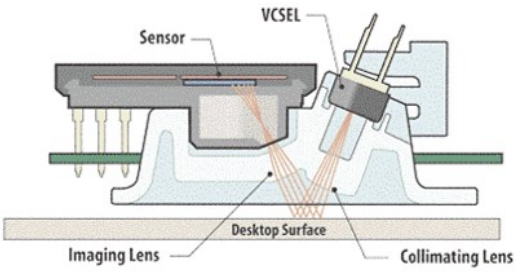


Fig. 1: Working Principle of mouse sensor (ADNS 9800) [13]

8200 cpi (counts-per-inch). All these features help in better detection of the optic-flow from the surrounding environment.

1) Overview of complete system: The working principle of the basic sensor setup is explained in Fig. 1. It shows us that VCSEL illuminates the surface via a collimating lens that narrows the laser light. This bright surface is focused by the imaging lens onto the photo-diode array (PDA). This array (pixels) produces differential current output based on the level of illumination they receive. Now the image can be retrieved from the PDA and compared against different frames to track the optic flow. ADNS 9800 can be programmed to capture images at the rates up to 12kHz with the major limiting factor being exposure time and threshold illumination required by the PDAs to detect the features. The data-sheet provides methods to extract the surface-quality (SQUAL) of the image captured that indicates the number of features detected [14]. Fig. 2 depicts an illustration of the raw data obtained at consecutive timestamps and how the displacement vector is analyzed.

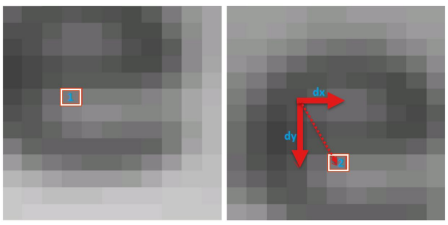


Fig. 2: Optic flow extraction in consecutive images

2) ADNS 9800 - Working explained: The single Chip-on-board (COB) package of the ADNS 9800 sensor uses a C8051F347 micro-controller that features an on-board universal serial bus 2.0. It performs at 25 MHz and uses a powerful 8051 core with 32 kB Flash, 2.25 kB RAM. The underlying principle of the sensor lies in measuring the change in position by optical image acquisition in a sequential order followed by mathematical analysis of the pixel displacement for direction and magnitude. Surface images are captured by the Image Acquisition System (IAS) on the 30×30 pixel arrays using the relevant optics and illumination system. A Digital Signal Processor (DSP) is involved in the processing of these images to identify the relative planar displacements δx and δy

corresponding to the direction and distance of motion. These signals are later communicated via the 4 wire Serial Peripheral Interface (SPI). An advantage of SPI communication is that data can be transferred without any interruption. They interact in a master (micro-controller) slave (sensor) relationship where a single master can have multiple slaves [15]. The current supplier Avago has made necessary connections in the single-chip sensor for establishing the SPI interface through the 4-channels MOSI (Master Out Slave In), MISO (Master In Slave Out), SS, and SC (Table I).

TABLE I: ADNS 9800 Sensor SPI Protocol

SS	Slave Select	MOT	Motion
MO	MOSI	DG	Digital Ground
SC	Serial Clock	AG	Analog Ground
MI	MISO	VI	Supply Voltage

C. Challenges

The primary concern using the ADNS sensor lies in the extraction of optical flow data by programming via Arduino-IDE. By extending the existing focal range from a few millimeters to the required distance of PC monitor using suitable optics. The sensor is conditioned to work in very bright conditions. Adapting the sensing for ambient lighting conditions and identifying the critical parameters influencing the senor needs to be studied. In order to achieve this, a test setup with a reliable and controllable sensing environment is required to be developed. The approach and findings are discussed in the upcoming sections.

III. Experimental setup for Evaluation

An experimental setup is developed to extract the optic-flow information from the ADNS sensor. The primary objective of the setup focuses on sensing the optical flow from a digital environment such as a computer screen. This unconventional approach provides an opportunity to use interactive software with customizable design parameters in a simulated optical flow environment. Various trials based on the different design parameters (pattern - size, contrast, speed, etc.) can be quickly analyzed without the need to manually change the setup. Another intention of the setup heads in the direction to build a flight simulator using pairs of ADNS 9800 sensor. Hence it becomes vital to analyze the most sensitive zone for the sensor.

An overall representation of the setup can be observed in Fig. 3 and Fig. 4 wherein the sensor is placed at a fixed distance of 320 mm from the PC monitor. The Optic flow generator is a Matlab-GUI developed to create controlled optical-flow. The details of the developed software are discussed in detail in Section III-C. Arduino Uno is used as a master micro-controller in the setup to monitor the data-acquisition.

A. Sensor optics

This section describes the optics involved in enabling the sensor to improve the focal range and to identify the influencing parameters in the formation of an image.

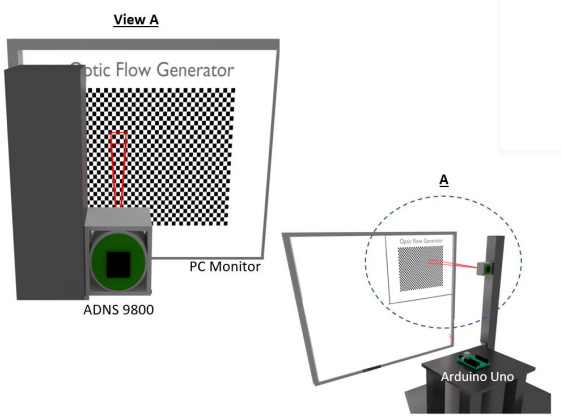


Fig. 3: Initial Setup for Optical Flow detection

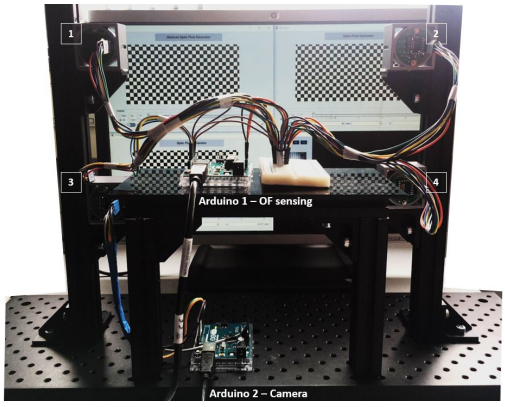


Fig. 4: Complete experimental setup for OF detection

The original operation of the sensor system primarily consists of three parts that are the camera, lens, and light source. The following sequence of processes occurs to obtain the image on the sensor. A light source illuminates the surface, and the reflected light generates enough particles to create the camera image. The lens system ensures the formation of a sharp image for the given distance. Now the camera consisting of PDA (Section II-B) divides the image amongst (30×30) 900 pixels. This image is processed further by the Digital Signal Processor (DSP) to analyze the displacement in each frame.

1) Influencing parameters: The overall idea of the sensor system and interaction with optics is shown in Fig. 5 wherein the X-Y plane is parallel to the image plane with the center axis passing through the origins. A checkerboard pattern is used as an image in the current scenario that emits the light particles via lens into the sensor (O). The surface area denoted as dA is at a distance ρ on the image for a solid angle ϕ from the sensor (O).

The radiation received by the sensor can be computed as the radiosity (E_o) received at the point from a diffused hemispherical surface area [A_o]. On equating the irradiation

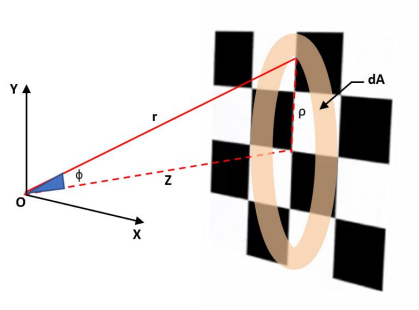


Fig. 5: Field-of-view visualization from Sensing environment
.

from the surface dA as seen in Eq. 3 leads us to the conclusion that the amount of light received at point O is independent of the distance. However, it depends on other parameters like FoV (ϕ), the pattern used, and reflectivity of the surface. The sensor has limitations on this front as the minimum amount of light required for proper functioning is around 80 W/m^2 [16] [17].

$$P_o = \int E_{surf} da_{surf} \tag{1}$$

$$da_{surf} = 2\pi\rho r d\phi = 2\pi r^2 \sin\phi d\phi$$

$$E_o = \int \frac{E_{surf}}{A_o} da_{\text{surf}} = E_{surf} \int_0^{\phi} \frac{2\pi r^2}{2\pi r^2} \sin\phi d\phi \tag{2}$$

$$E_o = E_{surf} \int_0^{\phi} \sin\phi d\phi \tag{3}$$

2) Optical System: The experimental setup is primarily being developed to analyze the sensor and to build a flight simulator. As the system mostly operates in a fixed distance, the optics involved is straight-forward and simple. The sensor is identified to be used on MAV/FWMAV for optical-flow detection [6]. Thus a general approach for developing the lens system is briefly described in this section. A good image in general requires precise focus, aperture, brightness, and depth of field for its formation. Humans possess adaptable optics to adjust these as per need. Insects like house-flies cannot vary these parameters and have fixed optics. A similar approach is used to develop a light-weighting optical system to best suit an indoor environment to sense the optical flow.

The working principle of a biconvex thin lens system is shown in Fig. 6 (a) can be calculated using the Eq. 4, where *s* is the object distance and *v* is the image distance. The depth of field can be regulated by adjusting the aperture *d* of the lens with fixed focal length *f*. An optimal focus exists for this system, so the image becomes blurry when moved closer or farther away. Identification of the farthest S_F and nearest S_N

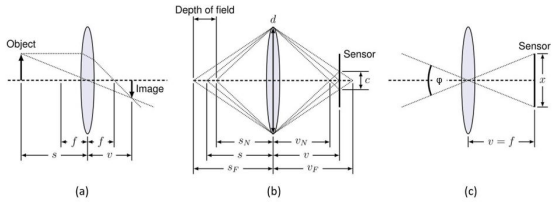


Fig. 6: Sensor Optics for thin lens system.
(a) Real image formation - lens with focal length f projects an object at distance s. (b) Circle of confusion c - optical spot created by light rays not coming to perfect focus. Depth of field marks the difference in max S_F and min S_N object distances for acceptable c. (c) Field of View (ϕ) - Depends on f and pixel/sensor (x) size [10]
.

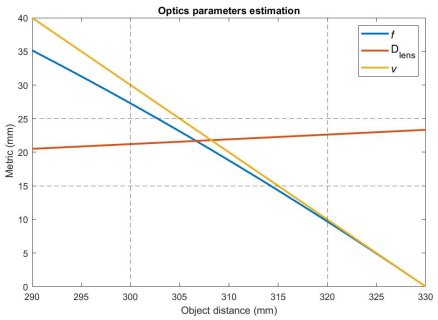


Fig. 7: Optical system parameters estimation f

object distance from Fig. 6 (b) helps us to know the Depth of Field (DoF) that still produces acceptable sharp images to understand the system better.

$$\frac{1}{s} + \frac{1}{v} = \frac{1}{f} \tag{4}$$

The Depth-of-Field also behaviorally characterizes the robot's responsiveness to sense the optic-flow and to generate signals for the controls. Another important factor affecting the vision system of the sensor is the Field-of-view (ϕ) as depicted in Fig. 6 (c) with the mathematical representation given in Eq. 5. This angle ϕ can be defined on either side of the lens.

$$\phi = 2arctan\left(\frac{x}{2f}\right) = 2arctan\left(\frac{\rho}{s}\right) \tag{5}$$

$$E_o = \left(\frac{D_{lens}}{D_{sens}}\right)^2 \int_0^{\phi} \sin\phi d\phi \tag{6}$$

$$D_{lens} = D_{sens} \sqrt{\frac{E_o}{E_{surf}} \frac{1}{1 - cos\phi}} \tag{7}$$

From Eq. 5, the solid angle contributing to FoV (ϕ) can be estimated for the chosen range (around 320 mm) from the sensor to monitor the screen. This range can be varied

depending on the required application. Also, the corresponding D_{lens} can be calculated using Eq. 7 for a object height (ρ) and fixed D_{sens} that represents the diameter of sensor opening (aperture). The Fig. 7 is plotted with three parameters (focal length, image distance, and lens diameter) with the object distance. The box enclosed by the dotted lines represents the most feasible zone considering standard available lenses. It can be inferred that the lens with f-20 mm and diameter D_{lens} 20 mm are chosen for the sensor system. A suitable casing is designed to prevent any alignment and adjustment issues in the lens system.

B. Information extraction from ADNS Sensor

Our next objective in the process aims to extract the optic flow data from the sensor via a suitable micro-controller. It has already been discussed in Section II-B2 that the SPI protocol with Arduino UNO microcontroller is suitable for communication. It is used as a master to control the ADNS mouse sensor that behaves as a slave in the architecture. Fig. 8 represents the connections made with the sensor as shown in Table II.

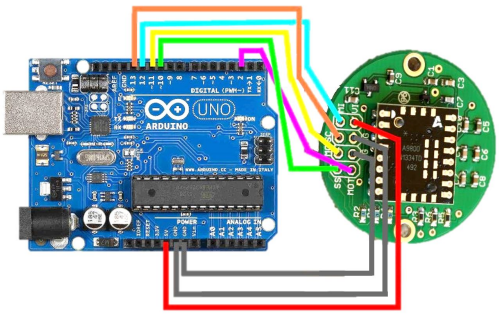


Fig. 8: ADNS 9800 Arduino UNO connection

TABLE II: ADNS 9800 - Arduino Uno Circuit connection

SS	10	MOT	2
MOSI	11	DG	GND
SCLK	13	AG	GND
MISO	12	VI	5V

The detailed explanation for communication relating to read and write operations can be referred from Appendix A. ADNS 9800 sensor is powered by 5V via the Arduino from a laptop through the USB A-B cable. This cable is responsible for both powering the system and for serial communication. To obtain the live image data, a secondary Arduino micro-controller is used. The circuit-configuration remains the same in both scenarios due to the similar master-slave SPI interaction. However, the frame-capture algorithm programmed in Arduino IDE for the image acquisition needs to be uploaded via COM port. Depending on the requirement, the system can be made to behave as a camera or to detect optic-flow. The algorithm containing the mandatory start-up and OF extraction procedure is uploaded via COM port. The sensed data is logged via Processing software for further analysis.

C. Optic Flow Generator - MATLAB GUI

Once the sensor is prepared to be used as the optical flow sensor, the vital aspect now depends on developing a customizable sensing environment. In general, two approaches exist to proceeding further, either to move the sensor or to move the surroundings. Since our predominant interest lies in obtaining the optic-flow data, moving the external environment fixing the sensor is favored as it reduces the vibration and kinematic disturbances affecting the sensor. A standard approach for developing a controlled environment setup involves moving a screen with a high contrast-pattern at known distance and velocity. The sensor can later be calibrated against the known velocity of the moving-screen. The distance moved by the screen is not very flexible to adjust, the pattern cannot be easily modified, only linear motions can be detected, and also the system is not portable. To address the shortcomings from the standard setup, a new compact system was developed using a Matlab GUI interface on a computer monitor. A visual representation of the software-based digital environment can be seen in Fig. 9 with the merits and challenges discussed,

- Advantages:
  - Flexibility to make changes in Flow pattern
  - Inclusion of rotational and linear motions
  - Closed-loop configurations for simulations
  - Longer data-acquisition logging ability
  - Variation in velocity and directions
  - No mechanical moving parts required
  - Can be used in combinations
- Challenges:
  - Limited frame-rate
  - Prediction of exact velocity

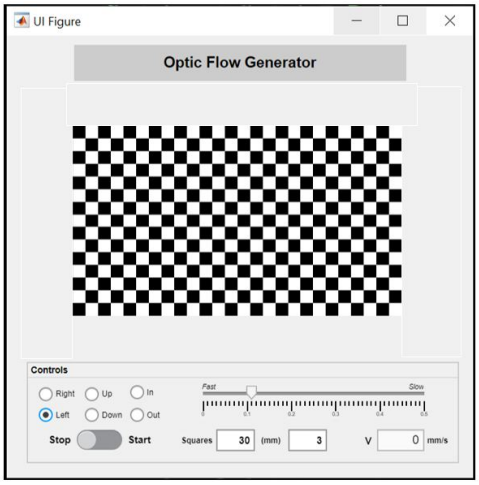


Fig. 9: MATLAB Optic Flow Generator

The Optic-flow Generator Matlab GUI is programmed to use a standard checkerboard pattern. The pattern is internally computed as a matrix and converted to a grey-scale image with only black and white colors. The core principle of creating

motion pertains to moving a small window within the large image that generates the flow. The monitor containing the GUI tab is aligned with the ADNS sensor for experimenting with various strategies. It contains interactive features to perform various tasks and is explained as per the same labels in Fig. 9. Similarly, a random matrix can be generated that results in a QR pattern in Fig.10.

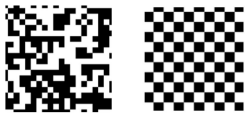


Fig. 10: Optic Flow Patterns (a) QR (b) Checkersboard

- **Start/Stop**: This function behaves as a Start, Stop, and Reset button in the GUI. It performs the tasks based on the position of the switch
- **Direction**: To move in all four directions (Up, Down, Right, Left) to virtually imitate the environment of the sensor in the intended direction.
- **Zoom**: This feature helps to visualize the sensors that are present in the linear direction of motion. In the case of forwarding motion, the front plane is zoomed-in whilst the rear plane is zoomed out.
- **Squares**: The datasheet shows the sensor functionality is affected by the texture of the surface used. This parameter helps to vary the texture within a constant surface area. Increasing the window size (squares) accommodates more patterns for the same area and vice versa reducing this. There exist an optimum size and can be found via window sweep operation.
- **Speed control**: The motion of the optic-flow can be increased or decreased for the desired speed. The program by default runs at max-speed and can be reduced/controlled using the delay function.
- **mm**: This parameter plays a vital role in knowing the actual velocity of the Optical flow. The size/aspect ratio of the monitor determines the velocity of the pattern on the screen. The squares tend to move faster on a bigger screen since the run time remains constant but the distance to cover is more. It is a calibration factor and estimated by measuring the size of squares on the sensing screen and averaging them.
- **v (mm/s)**: This feature estimates the actual speed of the moving pattern and displays it. The run-time for a single step is identified from the program and the corresponding distance/squares moved is estimated from the mm parameter. The ratio with distance and time results in the speed estimation. This factor is logged in as an excel file and used to calibrate the ADNS sensor.
- **Multiple tabs**: The process is scalable, meaning the number of sensing points can be varied as per the need. The stand-alone feature allows access to multiple programs at the same time.

This session concludes with the development of a complete system including the hardware, software, and sensing environment. The controlled surroundings help in better understanding and evaluation of the sensor.

IV. Optic Flow

The final objective aims to detect the optic-flow from the digital environment. A brief description of optic-flow is described in Introduction (Section I). Bio-inspiration from flying insects suggest that amidst poor visual resolution, they achieve efficient flight behaviors due to the higher processing power [18]. This context was taken as inspiration to proceed ahead using the ADNS 9800 sensor.

The general equation for detecting the optic flow is given by the Eq. 8 where v and ω represent the linear and angular velocities of the robot. Similarly, r indicates the normal distance between the sensor and moving-environment while θ indicates the angle between the moving-environment and direction of robot locomotion. The Optic flow is represented in *rad/s*.

$$\vec{OF} = \frac{v}{r}\sin\theta + \frac{\omega}{\beta} \tag{8}$$

In the current setup, the sensor is placed parallel to the screen. This shows the angle between the sensor and the moving environment is perpendicular (sin θ = 1). It can be seen from the Eq. 12 that the X+ sensor consists of two flow vectors in Z and Y direction. These components are mutually perpendicular and each component consists of linear and angular velocity components. Depending on the flow, a resultant vector can be derived that gives the overall predicted motion. In a bigger picture, when multiple sensors are used in robots, the final vector can be computed from the individual resultants depending on their orientation. To avoid complications, only linear velocity components (ω = 0 *rad/s*) are analyzed in the study. The velocity components are negative because optic flow and velocity are opposite in direction.

$$\vec{\mathbf{OF}}_{Sensor}^{Direction} \tag{9}$$

The general notation is seen in Eq. 9 is followed in the rest of the study where the $\vec{OF}$ is the optic flow vector. The subscript indicates the sensor and the superscript shows the corresponding sensing direction.

$$OF_{X+}^{y} = \frac{-v_y}{r_1}\sin\theta - \frac{\omega_z}{\beta} \tag{10}$$

$$OF_{X+}^{z} = \frac{-v_z}{r_1}\sin\theta + \frac{\omega_y}{\beta} \tag{11}$$

The resultant vector for the sensor can be expressed as,

$$\vec{OF_{X+}} = [OF_{X+}^{y}, OF_{X+}^{z}] \tag{12}$$

The experiment focuses on detecting a known velocity profile from the developed GUI at a fixed distance. The sensor output processed in counts per second (cps) is calibrated with

a Γ (*rad*) factor and analyzed for actual velocity. The raw data needs a calibration factor to be used in unknown scenarios to convert the optical flow to velocity.

The sensed and actual velocity (*m/s*) need to be equal,

$$V_{adns} = V_{gui} \tag{13}$$

The final optical-flow vector $\vec{OF_{adns}}$ is estimated as a product of the calibration factor Γ and raw sensor output OP_{adns} indicated by

$$OF_{adns}(rad/s) = \Gamma(rad) \times OP_{adns}(s^{-1})$$

$$\vec{OF_{adns}} = \frac{V_{adns} \times sin\theta}{r}$$

$$OP_{adns} = \frac{V_{adns} \times sin\theta}{\Gamma \times r} \tag{14}$$

Finally, solving the equations with $\theta = \frac{\pi}{2}$ results in,

$$\Gamma = \frac{V_{gui}}{OP_{adns} \times r} \tag{15}$$

V. Results and Discussion

This section discusses various results relating to understanding the sensor that is obtained from the developed Hardware-in-the-loop (HITL) setup.

The PC screen used is an HP 24" monitor with a resolution of 1680 x 1050 and operates at 60 Hz. The sensor is operated at 2000 Hz that is the lowest operating capacity that can be programmed via Arduino UNO. This also ensures sufficient light entry for image formation. Finally, the filtering effect is implemented by setting the baud rate to 19200 that prints the data at rates of 60 Hz. Thereby the data acquisition rate from the Arduino micro-controller has been matched with the computer screen refresh rate to avoid latency and noise issues.

The following general sign conventions are adhered to the remaining experiments wherein the Right and Up motion set in GUI represents a positive output and the other motions result in a negative output. The position of the sensor was fixed and the focal was adjusted and remained unchanged. The circle of confusion, meaning the distance of the lens moved for a blurry and clear image within focus was found to be around 2-3 mm. The distance between the PC screen and sensor was fixed at 320 mm. No external lightings were used in the indoor environment apart from the available natural lighting.

A. Modifications on Sensor

Initially, after a few trials, the sensor was analyzed to be more receptive to light and inferred the need for a few modifications to enhance the performance. In the initial sensor casing, an offset existed between the sensor aperture and the physical optical lens. Also, the aperture opening was at an angle, to support the entry of laser-reflected rays from the surface (original intended purpose - Fig. 1). The optical centers

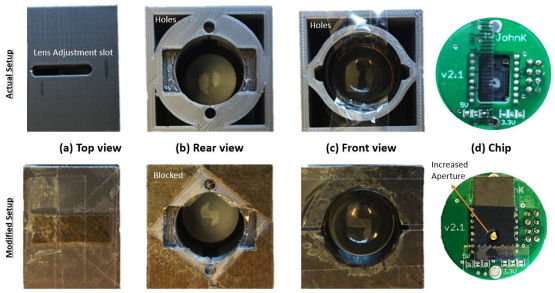


Fig. 11: Comparison of ADNS Modified vs Original sensor

were deviated thereby resulting in partial light entry and blurry image formation. Also, the existing aperture of 0.25 mm was insufficient for allowing enough light particles into the PDA. To rectify these issues, the sensor was modified to be in line with the optical axis, and the aperture size was increased to 2 mm. It was observed that increasing the aperture resulted in too bright an image and inclusion of unnecessary noise in the image. To counterfeit this effect, all the possible reflected luminescence were sealed using a black enclosure as seen in Fig. 11 that made the sensor much more efficient to capture optical flow data.

1) ADNS image capture: Once the sensor is adapted to perform better, it becomes important to know what the sensor visualizes. The actual zone contributing to the optical flow was initially identified by blocking the GUI interface on PC until it influenced the optic flow output. This rough estimation enabled us to have insights into the field-of-view (ϕ). The sensor is programmed to capture a live image via Arduino IDE. A GUI is developed using the Processing software to visualize images captured by the sensor. Initially, a 15 $\times$ 15 image with limited features is obtained. The blur and focus were corrected by aligning the focal axis of the sensor and lens. Finally, the 30 $\times$ 30 pixels image containing all the 900 pixels can be Fig. 12.

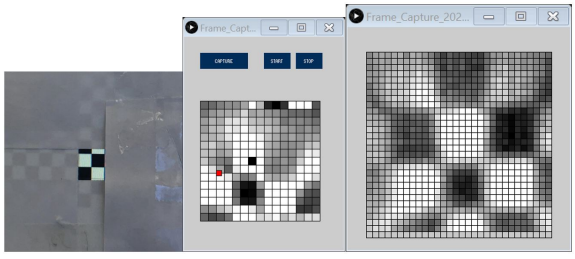


Fig. 12: Field of View Identification from Live Data

2) Impact zone - FoV: Once the more tedious task of obtaining the image is completed, more insights about the sensor can be inferred. The FoV can be estimated using the image data. 900 pixels in total form the image consisting of a specific number of checkerboard squares on them. Since the size of the square is a known variable, it is equated directly

for ρ which gives us ϕ. Eq. 5 is used to obtain the view of the sensor in a horizontal, vertical, and diagonal direction as seen in Table III. The image is square and contains an equal field of view in the horizontal and vertical direction. The pictorial representation of this can be seen in Fig. 13.

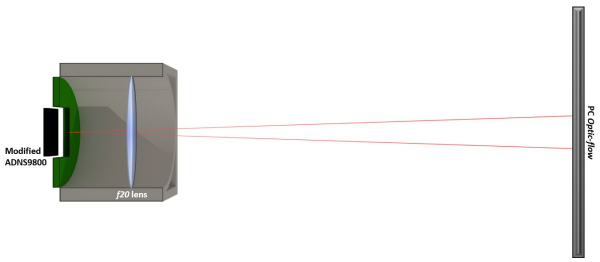


Fig. 13: Experimental setup showing FoV

TABLE III: FoV estimation for r at 320 mm

Fov	ρ (mm)	Angle (deg)	Angle (rad)
Horizontal	7.5	2.6852	0.0457
Vertical	7.5	2.6852	0.0457
Diagonal	10.6	3.7968	0.0663

B. Optic Flow related Inference

The general properties of the sensor setup have been discussed earlier. This section deals with the raw Optic flow data acquired from the sensor under various conditions. The output is analyzed in counts per second (cps).

1) Raw OF Data: The foremost criterion required to affirm the vectorial characteristics of sensed optic flow data is to have a change in both direction and magnitude. Numerous trials have been performed to confirm this considering the consistency in the relation between speed and sensor data. The figure below shows the output from a single sensor facing X+ direction with the data in the Y and Z direction. The output from the ADNS sensor subjected to varying velocity and random direction change in Optic-flow Generator can be seen in Fig. 14. The optic-flow obtained in Z and Y directions can be along either the positive or negative axis that gives the angle. The variations in the detected signal occur mainly due to the discrepancy in the data acquisition rate. The sensor operates at a much higher frequency, the set baud rate in the Arduino micro-controller filters the data acquired. There exist a vital possibility for the sensor to detect a blank screen in between the refresh rate of the monitor. It can also be observed that the data increases in steps of unit cps. The sensor frame capture rate can be tweaked more and also improving the smoothness in the optic-flow pattern contribute to more clean data. These factors are considered and rectified for upcoming sessions. The dotted lines represent the moving average estimated from the raw sensor.

2) Optimal Feature Identification: The sensor datasheet indicates ADNS 9800 is sensitive to the contrast pattern used to detect change. This variation is subtle but present when using the stock lens [14]. The surface quality (SQUAL) value shows the quality of the image acquired for analysis. Studies have proven ADNS 9800 to be more sensitive to checkerboards pattern [19].

Identification of the optimum feature for sensing helps in achieving the maximum optic flow for a given velocity. This feature can be described as the window-size indicating the number of squares in the checkerboard pattern. For example, W20 indicates there exist 20 squares within the length of the Matlab-GUI checkerboard display area. The velocity of the moving pattern on GUI is set constant for all the trials executed with different window sizes from W20 to W90. The corresponding optic flow data are compiled and plotted for the window sweep operation. This procedure is repeated with two sensors (X+ and X-) for affirmation of optimal feature size

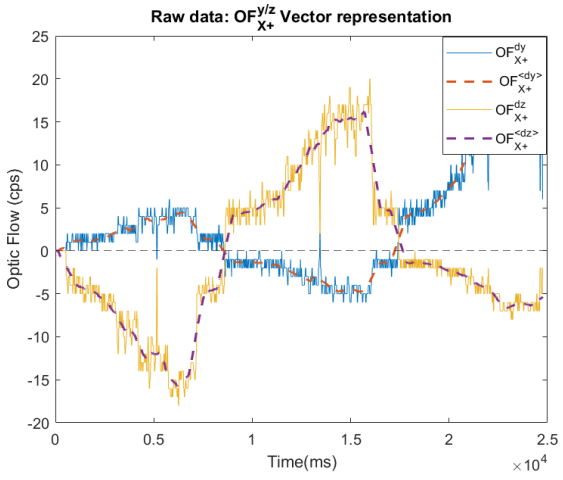


Fig. 14: Raw Optic Flow data (OF_{X+}) from ADNS 9800 sensor and moving average ($OF^{<avg>}_{X+}$)

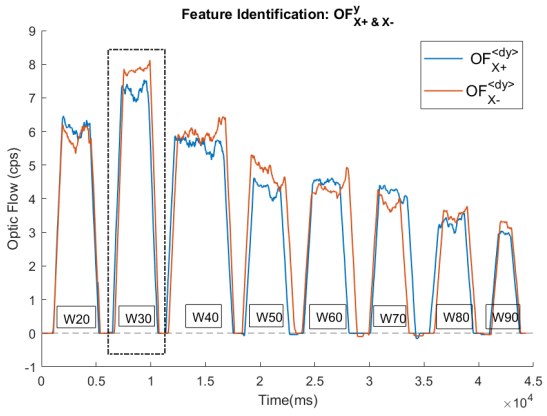


Fig. 15: Optimal Feature Identification for Window-sweep from W20 - W90

and data can be observed in Fig. 15. Tuning this parameter becomes vital in improving the sensing efficiency. The W20 marker indicates that 20 squares fit in the GUI window. The experimental analysis is stopped beyond the W90 value as a decreasing trend was observed. It was concluded that W30 was chosen as the apt fit for the current situation as it had the maximum optical flow among others for the constant optical flow. With the known FoV (ϕ), it becomes possible to calculate the visible region on the computer screen. The estimated visible height complies with the visualized live data and suggests that the contrast pattern of W30 with 3×3 black and white squares influence the sensor at 320 mm. This optimal window size varies with distance from the sensor as the FoV changes. The SQUAL value estimated was around an acceptable range of 50.

3) Performance with Brightness: An interesting feature to analyze the sensor is the dependency of the lighting involved. The sensor was originally intended to be used with a laser light source, that provides extreme brightness in a confined space to capture the changes. A major objective relating to the research focuses on enhancing the threshold required for sensing. In an indoor scenario, laser or highly luminous lights cannot be used. The modified sensor aims to solve this issue.

A comparative analysis with the dependency of brightness and contrast on the original and modified sensor is analyzed in Fig. 16. The plot explains the impact of lighting on the original and modified sensors. Similar to the feature identification analysis, the velocity is maintained constant through the experiment. The brightness and contrast on the PC are maintained at 100 %, for the constant speed of the optic flow on the Optic-flow Generator GUI. Initially, the brightness is reduced until the sensor output ceases. A similar procedure is repeated while maintaining the brightness at 100 % and reducing the contrast. The performance is substantially improved during sub nominal conditions and can be seen in Fig. 16. The dependence on optic-flow output is more on the contrast of the screen than the brightness at lower operating conditions. When the contrast is lowered below 40%, the sensor-output drops to a large extent. Under a similar situation, though the brightness is low, the optic-flow output is relatively higher than reducing the contrast. It can be inferred that the sensor is more prone to contrast change than brightness. Overall, it can be observed that prior modifications implemented on the sensor make it highly sensitive to detect the optic flow. However, care must also be ensured to filter the ambient optical noise as suggested in Fig. 11 for better sensitivity.

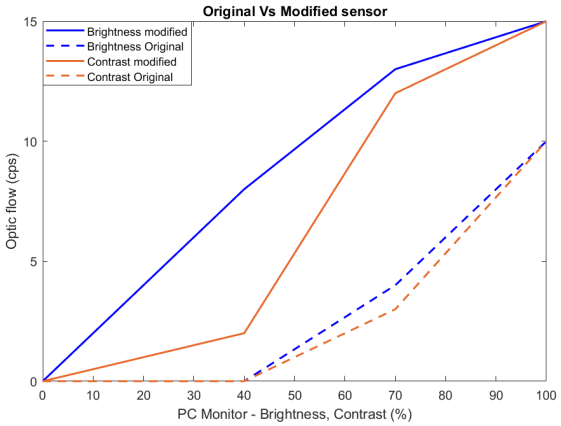


Fig. 16: Optic-flow - Brightness & Contrast dependence

4) Linearity trend: The subsequent objective of the sensor is aimed to detect the velocity in intended directions as seen in section IV. Currently, all the parameters are focused on raw sensor data. The standard variations from the mean profile account to 1-3 cps. Calibration is required to convert from Optic flow (cps) to velocity (mm/s). This data can be further processed for position identification. Linearity is analyzed, for the sensed optic flow and on-screen velocity as this is necessary to have smooth calibration. Fig. 17 represents the scattered data against various trials at unequal intervals. It was found that for the fixed distance, the data trend is linear as per the Eq. 16 where OP_{adns} is sensed data and v_{gui} is the actual on-screen velocity from GUI interface. The calibration factor Γ was estimated as 0.0121 *rad* from Eq.15 and OF was calculated as follows,

$$OP_{adns} = 0.2588 \times v_{gui} \tag{16}$$
$$OF_{adns} = \Gamma \times OP_{adns}$$

Hence, the optic flow can be calculated from the known velocity. Similarly, the velocity of a moving object can be predicted from the known sensor output.

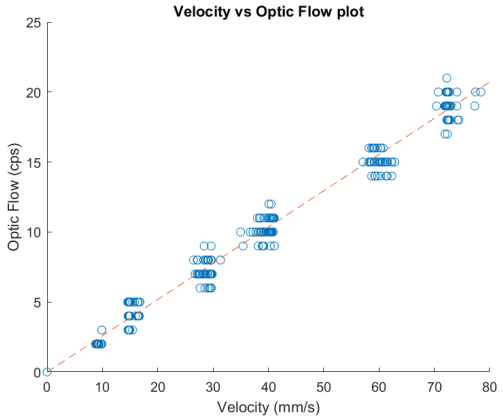


Fig. 17: Linear relation: OP_{adns} vs V_{gui}

5) Digital vs Analog comparison: Until now, the sensor performance evaluation has been executed using a digital sensing environment. The digital screen has a fixed refresh rate and the sensor is currently matched to adhere to this. A comparison between the digital and analog sensing environment would provide better insights. To verify the optical flow data

Fig. 18: Simple analog setup for Optic flow detection

acquired from the digital environment, an analog test setup was developed. Most of the design parameters were replicated from the digital environment. The setup can be viewed in Fig. 18 where the computer monitor is replaced using a rotating setup with a QR/checkerboard contrast pattern. The distance between sensor and environment, optic-flow pattern, FoV, and data-acquisition rates are maintained the same in both the conditions.

A direct comparison of the signal acquired from the Digital and Analog environment for a similar velocity profile, from the same sensor has been plotted in Fig. 19. The velocity profile of analog-setup is varied to match the required optic-flow obtained from the digital environment for comparison. Hence an average deviation exists in the output obtained from the analog and digital sensing environment as the velocity profiles are different. The plot shows that the variation observed in Analog-environment is more than the Digital environment. This may be due to various possible reasons. The optic-flow output reports the change count in each direction since the last report. An important factor suspected for more deviations in the analog-environment is the continuously varying input that is subjected to notice more changes. Also, the curvature of the rotating-pattern influences the variation as there is a loss of around 1 mm at the edges in the field-of-view. The discrete digital environment acts piece-wise continuous, with the presence of appearing-blank screens in between the refresh rates a monitor.

The sensor captures the contrast of the black and white image on the monitor, there exist zones with flat-line and suspected due to the synchronous-nature of the blank screen and black image in checkerboard. It can be concluded that the data obtained from the digital environment can be used, considering the sensor is sensitive to different velocities and does not vary much within a constant velocity.

The results found in Fig. 20 show the comparison between the original and modified sensors viewing at the same source for constant velocity. A repeating trend is observed in the data acquired due to the periodic motion of the pattern used. The dotted lines in the plot indicate the peak and lows occur during the same time. A moving average of the data is plotted since the focus here lies more in understanding the increment from the original sensor. The output has almost doubled indicating positive inference that supports the sensor can be useful in a real indoor environment. A deviation of 46mm/s in predicted-velocity of the rotating analog pattern using the Γ calculated

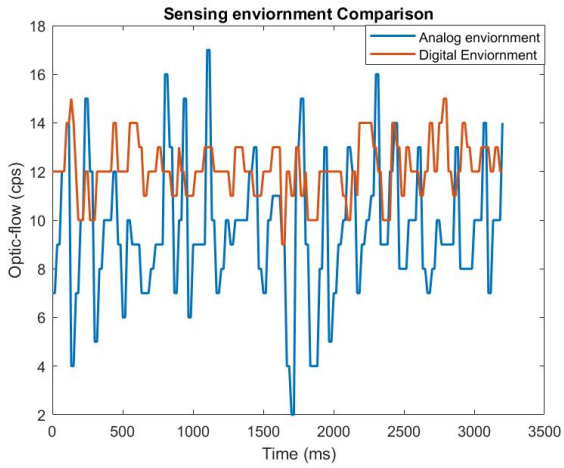


Fig. 19: Comparison of Analog and Digital Environment

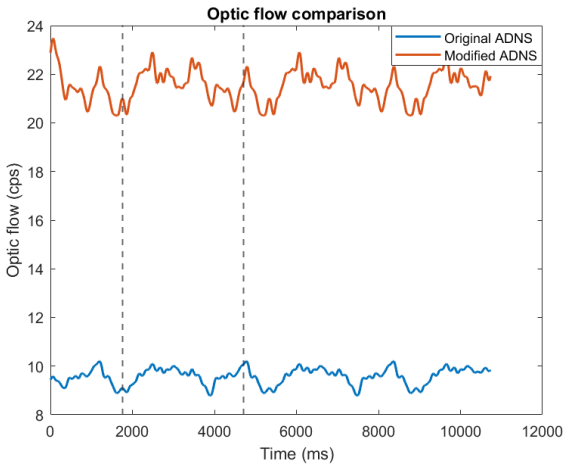


Fig. 20: Original vs Modified sensor - Analog environment

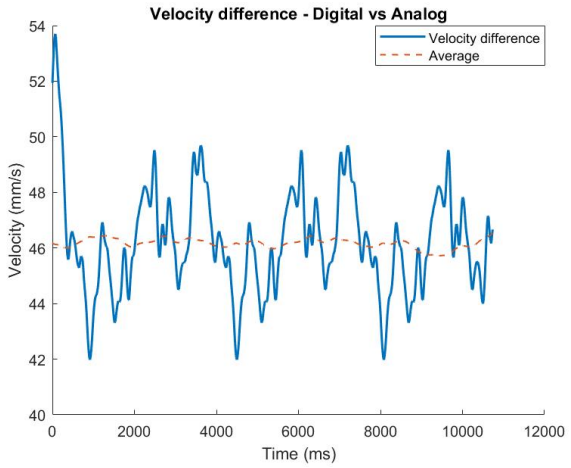


Fig. 21: Difference in predicted Analog-setup velocity using $\Gamma_{digital}$ as 0.0121

(Eq. 15) from the digital environment can be seen in Fig 21. This suggests that the Γ needs to be re-calibrated for an analog setup as it depends on various factors including lighting, the reflectivity of paper, and contrast of pattern used.

VI. Conclusion

The results compiled lead us to the conclusions that the ADNS 9800 optical-mouse sensor works fine as an optical flow sensor in the current scenario satisfying the SWaP constraints. Initially, the sensor required increasing the aperture by 1.75mm coupled with suitable optics. The sensor has been proven sufficient to be used for indoor conditions under normal lighting. The optic-flow sensor output varies more depending on the contrast of the pattern used. Also, the linear behavior within bearable deviations, for the set parameters proves reliability. The comparison with the analog movement of flow pattern shows that the velocity deduced can be reliable with prior calibration. The raw-signal analyzed from the digital-environment, sensed by the ADNS 9800 indicates that the sensor provides stable signals for controls. The current experiment also suggests the possibility to build a Hardware-in-the-loop flight simulator for controls and path prediction applications.

References

[1] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 53(3):263–296, 2008.

[2] Bl Horn and KP Berthold. Schunck. determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[3] Rico Moeckel, Roger Jaeggi, and Shih Chii Liu. Steering with an aVLSI motion detection chip. *Proceedings - IEEE International Symposium on Circuits and Systems*, (May):1036–1039, 2008.

[4] Jean-Christophe Zufferey, Adam Klaptocz, Antoine Beyeler, Jean-Daniel Nicoud, and Dario Floreano. A 10-gram microflyer for vision-based indoor navigation. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3267–3272. IEEE, 2006.

[5] Alejandro Alvarez-Aguirre, Gihin Mok, S. Hassan HosseinNia, and Jo Spronck. Performance improvement of optical mouse sensors: Application in a precision planar stage. *2016 International Conference on Manipulation, Automation and Robotics at Small Scales, MARSS 2016*, 2016.

[6] J. F.L. Goosen. Design aspects of a bio-inspired flying sensor node. *Proceedings of IEEE Sensors*, pages 1–4, 2012.

[7] Kimberly McGuire, Guido De Croon, Christophe De Wagter, Karl Tuyls, and Hilbert Kappen. Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. *IEEE Robotics and Automation Letters*, 2(2):1070–1076, 2017.

[8] Richard JD Moore, Saul Thurrowgood, Daniel Bland, Dean Soccol, and Mandyam V Srinivasan. A fast and adaptive method for estimating uav attitude from the visual horizon. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4935–4940. IEEE, 2011.

[9] Bruce D. Lucas and Takeo Kanade. Iterative Image Registration Technique With an Application To Stereo Vision. 2:674–679, 1981.

[10] Rico Mockel. Bio-Inspired Optical Flow Vision Sensors for Visual Guidance of Autonomous Robots. *PhD Thesis*, (20736), 2012.

[11] Fabien Expert and Franck Ruffier. Flying over uneven moving terrain based on optic-flow cues without any need for reference frames or accelerometers. *Bioinspiration & biomimetics*, 10(2):026003, 2015.

[12] Matěj Karásek, Florian T Muijres, Christophe De Wagter, Bart DW Remes, and Guido CHE de Croon. A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns. *Science*, 361(6407):1089–1094, 2018.

[13] Cheng Liu, Yanan Xu, Ji-Gou Liu, Hui Sun, and Ralph Kennel. Rotational speed measurement based on avago adns-9800 laser mouse sensor. In *PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, pages 1–5. VDE, 2016.

[14] ADNS9800, howpublished = https://datasheet.octopart.com/adns-9800-avago-datasheet-10666463.pdf, note = Accessed: 2020-05-04.

[15] T. Praveen Blessington, B. Bhanu Murthy, G. V. Ganesh, and T. S. R. Prasad. Optimal implementation of uart-spi interface in soc. In *2012 International Conference on Devices, Circuits and Systems (ICDCS)*, pages 673–677, 2012.

[16] Abhimanyu Selvan. Simulation of optic flow based flight control for a flapping wing micro aerial vehicle. page 70, 2014.

[17] Microsystems Engineering and L Goosen. Optic Flow karakteristiek van een optische muis sensor. (groep 130), 2014.

[18] Mandyam V Srinivasan, Shaowu Zhang, and Javaan S Chahl. Landing strategies in honeybees, and possible applications to autonomous airborne vehicles. *The Biological Bulletin*, 200(2):216–221, 2001.

[19] Frank Wendler, Nivedita Pagar, and Olfa Kanoun. Evaluation of ccd-based human device interface sensor for the velocity measurement in floodwater. In *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, pages 254–256. IEEE, 2019.

[20] Feedback Control Systems. The Massachusetts Institute of Technology. *Nature*, 167(4247):470–471, 1951.

Appendix

Appendix A
Communication protocol

The MOT channel pings a signal when motion is detected. Since continuous measurement of data is needed, it is not in use. Several standard protocols such as the startup and frame capture are mandatory to use the ADNS sensor. These configure the required parameters into the sensor. The master selects the slave by lowering the voltage in the set SS line (Table I). This corresponds to the Nil Chip Select (NCS) signal that indicates if the connection is authenticated. The other 2 channels MOSI and MISO are responsible for the data communication between the master and slave. In the current context, the ADNS sensor is the slave, and the micro-controller - Arduino UNO is the master. This proposes that data from the

ADNS sensor to Arduino UNO is transferred via MISO and the vice-versa is done through MOSI. A better understanding

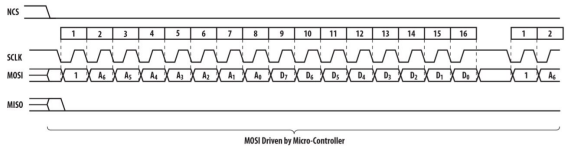


Fig. 22: ADNS9800 Write operation [14]

of the Read and Write operation of the ADNS and micro-controller is explained in Fig. 22 that enables us to send and receive data in a specific sequence. The Write operation can be defined as the data transferring from master to the slave and is always initiated by micro-controller (Arduino UNO) and consist of 2 bytes. The data direction is indicated by Most Significant Bit (MSB) as "1" followed by the 7 bits containing the address (A_0 - A_6 [14]) in the first byte. Subsequently, the sensed data (D_0 - D_7 [14]) is transferred by the second byte. ADNS sensor reads MOSI on the rising edge of the Serial Clock. This information is vital, as programming for the right MSB mode on micro-controller is vital for efficient data transfer. The SCLK frequency controls the speed of data transfer which is set to a few MHz via clock divider.

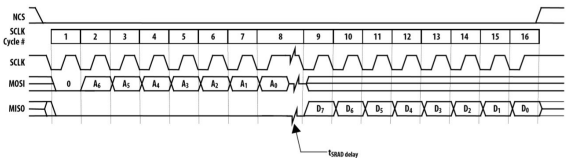


Fig. 23: ADNS9800 Read operation [14]

The Read operation completes the return communication from the ADNS sensor to the micro-controller. Similarly, it has 2 bytes and is always initiated by the master. The first byte contains the address and sent by MOSI with MSB as "0" to indicate Read operation. The second byte contains the sensor outputs and is sent over the MISO line. An important note from the datasheet [14] shows that sensor outputs MISO bits on the Falling edge of SCLK and samples MOSI bits on every rising edge. The t_{SRAD} is the address-data delay i.e. measured from rising SCLK for the last bit of the address byte to falling SCLK for the first bit of data being read.

APPENDIX B
SUPPORTING DATA

A. Controls and Stability

This section briefly discussed the stability criterion for the sensed signal. The sensed data from the simulator have been analyzed with a suitable plant-identification toolbox in Matlab using a step-input. The optic-flow data from ADNS 9800 sensor subjected to a step-input for a constant velocity has been analyzed. A relevant PID controller for the identified plant has been developed.

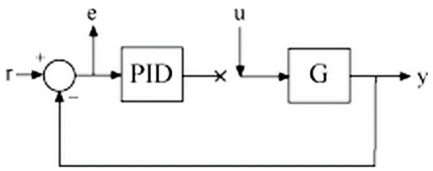


Fig. 24: System representation

1) Signal Identification: The initial objective of the analysis focuses on understanding the type of system using the sensed optical-flow sensor data. The step-response of the optical-flow sensed data with appropriate delay details are fed into the system. Various plant types were tried to fit with the acquired data. Finally, the plant with single-pole was chosen due to the best fit represented by the Eq. 17. The identified-plant and corresponding step-response plot can be seen in Fig. 25 and Fig. 26.

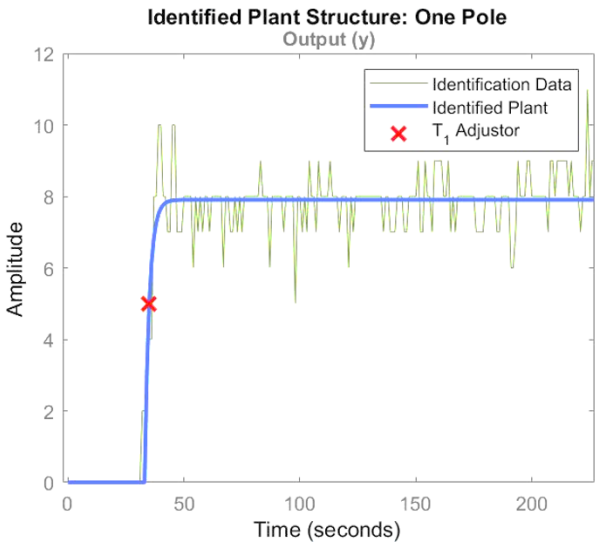


Fig. 25: Plant Identification for step-response

Process model with transfer function: The K_P value was found to be 8.315 and the T_{p1} to be 3.1323.

$$G(s) = \frac{K_p}{1 + Tp_1 \times s} \tag{17}$$

The plot 25 represents the step response identified from the sensor signal. The amplitude is around 8 indicating the obtained optic-flow for the set pattern velocity.

2) Control plots: A PID controller was designed for the plant and tuned to adhere to the following parameters. This briefly explains the possibilities to use current-setup reliable experiments. An in-depth analysis can be performed for improving the performance. It can be inferred from the Table. IV the system is stable. Bode-plots shows the frequency response of a system with the useful gain and phase margin of a system.

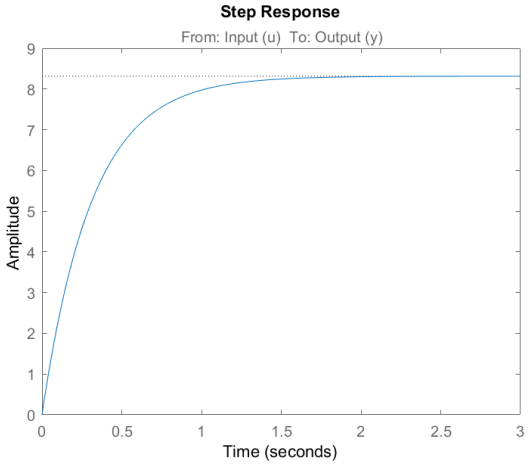


Fig. 26: Step response plot

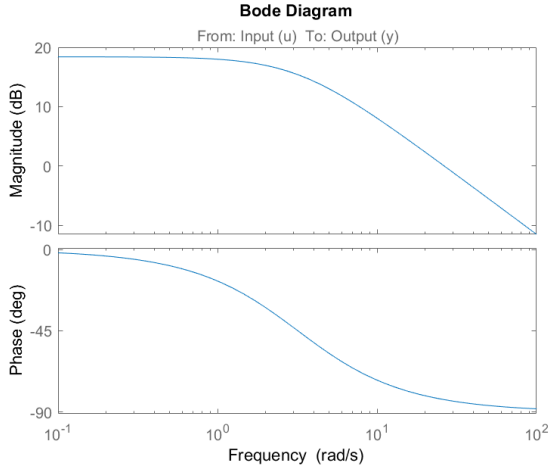


Fig. 27: Bode-Plot

3) Stability: The stability of the system can be verified with the Pole-zero plot.The transfer function of dynamic systems such as sensors can be plotted in a complex plane of poles and zeros. The presence of a pole on the left side of the s-plane indicates stability [20]. It can be seen in Fig. 28 that the pole lies in the negative real axis at a value of -3.19. This confirms that the signal is stable.

These supporting graph represent the stability of the system with the 0 dB crossing over at 26.2 rad/s at phase of -83. The phase lies within the controllable zone.

TABLE IV: Parameters for PID controller

Control Parameters	
K_p	0.3528
K_i	1.3959
K_d	0
Performance and Robustness	
Rise time	0.2303s
Settling time	0.305s
Overshoot	1.7%
Peak	1.02
Gain Margin	Inf dB @ NaN rad/s
Phase Margin	86 deg @ 9.61 rad/s
Closed-loop stability	Stable

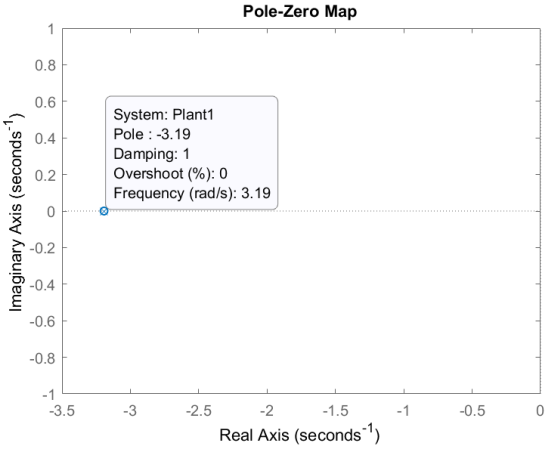


Fig. 28: Pole-zero plot

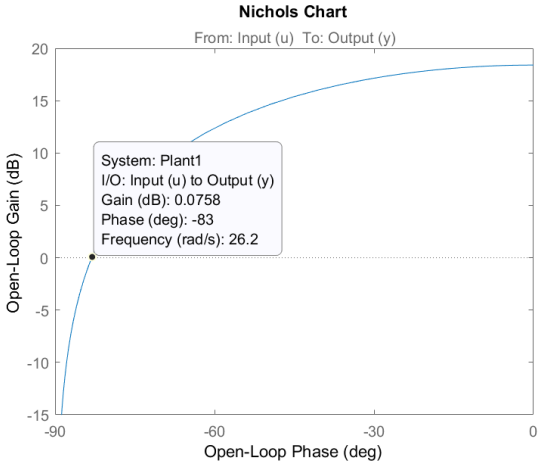


Fig. 29: Nicholas Plot

Chapter 5

Paper II - HITL Flight Simulator

The previous paper discussed more about a single sensor and its performance. It was concluded that the sensor seemed sufficient to be used as a optical flow sensor. This paper discusses more in using combinations of these sensors to develop a Hardware-in-the-loop flight simulator. The major inferences relating to the simulator are discussed in this section. More insights on using the simulator are discussed. Finally the trajectory path analysed from the sensed data for controls.

Development of HITL flight simulator for controls using Optic flow

J.G. Suresh Subramonian*, J.F.L Goosen**
Department of Precision and Microsystems Engineering
Delft University of Technology
2628 CD Delft, The Netherlands
*j.g.sureshsubramonian@student.tudelft.nl, **j.f.l.goosen@tudelft.nl

***Abstract*— The current study demonstrates the possibilities of using two-dimensional motion detection data for trajectory prediction. The ADNS 9800 optical mouse sensor is being used to sense the optical-flow data to predict the intended-path. Despite the limited resolution (30×30 pixels) and field-of-view ($2 - 3^o$), the sensor is identified and proven to perform well as an optical-flow sensor. Combinations of these sensors are used to develop a Hardware-in-the-loop (HITL) flight simulator using a digital environment. This experimental setup provides the possibility to initiate flight controls and understand control-strategies. Results indicate that the flight simulator works as intended for indoor situations within an error tolerance in Optic-flow of 4.83 % from the actual path.**

Keywords: Optic flow, HITL, SITL, mobile robots, visual navigation, controls.

I. INTRODUCTION

Visual optical-flow motion is a combination of linear and angular velocities with distance information from the obstacle. This vital data from a single parameter make them very interesting to handle and process in the fields of autonomous navigation. Optical flow can be analyzed in a dense manner using Färneback [1], Horn-Schunck [2] or by sparsely identifying the features proposed by Shi [3] over time using the commonly known Lucas-Kanade method [4]. It is a common phenomenon used by flying insects to steer and perform complex maneuvers. Various studies have indicated that extremely limited vision from compound eyes of insects is sufficient to perform difficult tasks like take-off and landing operations [5]. This helps us in realizing two major inferences in detecting optical flow data. The first being, absence for the complicated high-resolution systems followed with the need for a very high processing rate of images [6]. These core values are considered as motivation to develop a Hardware-in-the-loop flight simulator for FWMAV robot application as seen in Atalanta [7]. In general, optic-flow is detected by cameras, custom-made sensors, and mouse chips [8]. From the literature, it can be inferred that the use of optical mouse sensors such as the ADNS 9800 is sufficient for detecting the flow from a digital environment [9]. The sensor posses convincing technical characteristics in terms of size, weight, and pocessing domains.

This context focuses on developing a flight simulator using four sensors to predict the trajectory and to understand the controllability of the signal. The motivation behind developing a flight simulator for the MAV (Micro Aerial Vehicle) application concentrates more on understanding pure optic-flow based controls. Most of the kinematic factors of the robot are not considered here as the influencing parameter for controls only depends on optical-flow. The digital environment with a Matlab GUI supports the generation of various path profiles. Comparing the actual path with the sensed data gives the deviation from the actual position. On further identifying the plant for the system, it can suitably be tuned using the PID controller. The supporting parameters can be studied for sensitivity and stability. Sensors positioned in the HITL simulator can be considered as replicating the original position on MAV and accordingly the data can be fed into the system.

The outline of the paper can be split into three sections. The initial objective focuses on developing a reliable experimental setup to support a minimum of four sensors. The data obtained from the experiment is processed to predict the relative position based on pure optic-flow based control strategies [7]. Finally, the necessary control-signals for the stable flight of MAV are analyzed. A quick comparison with Software-in-the-loop (SITL) using the ROS-Gazebo simulation environment is demonstrated.

II. REQUIRED BACKGROUND INFORMATION

This section begins by explaining the prior knowledge required to accomplish the end objective. A quick understanding of the ADNS 9800 optical mouse sensor is gathered. Later, the MAV configuration required for building the setup and its related Optical-flow derivations is examined. This section concludes with understanding the GUI developed for the HITL simulator.

A. Sensor - ADNS 9800

The ADNS 9800 optical mouse sensor comes as a single Chip-on-Board (CoB) package with 16-pins and includes a vertical-cavity surface-emitting laser (VCSEL). The advanced programmable features enable us to control the frame rate and resolution in X and Y directions. The basic principle of the sensor can be explained as follows, it captures the image using an Image Acquisition System (IAS), this is analyzed by a Digital Signal Processor (DSP) which identifies the

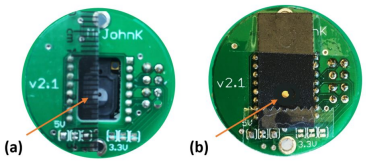


Fig. 1: Aperture modification: (a) Original sensor - 0.25mm vs (b) Modified sensor - 2mm

change in two directions on the image. Finally, the data is obtained in counts per second indicating the shift occurred in each frame in two-dimensions. It is sensitive to direction and displacement since it is a vector. This optical flow information is used to predict the distance from walls and velocity. Due to a limited resolution of 900 pixels, the processor enables to capture frames at rates up to 12 kHz. Since optic-flow is more interested in knowing the change in each frame, the frame rate outweighs the resolution factor. This concept is mainly inspired from flying insects [6] considering the SWaP criterion. The communication between the sensor and master microcontroller happens over the 4-way SPI protocol.

To enhance the focal range and to make it more sensitive to ambient lighting, the aperture was increased from 0.25 mm to 2 mm (Fig. 1) and coupled with $f20$ lens. To counterfeit the effects of ambient optical noise, the casing was sealed at possible openings as seen in Fig. 2.

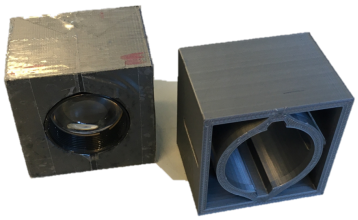


Fig. 2: Sensor packaging: (Left) Modified vs (Right) Original

B. General MAV configuration

The sensing is accomplished on the robot depending on the position and orientation of sensor placement. The general possible configuration along with the associated optic flow resultants are discussed here.

1) Robot orientation: To build an optic flow simulator, knowing the robot orientation with the surrounding environment becomes essential. The configuration followed can be seen in Fig. 3 where the MAV uses four-optical sensors in the Cartesian coordinates. In general, the robot requires six sensors to view in all the directions for flight controls and stabilization. The sensors *Y+* and *Y-* direction are into and out of the environment, it is not illustrated in the representation. Mostly during hovering, all the six optical flow data contribute to the controls. Unlike normal drones, Flapping wing MAV possesses

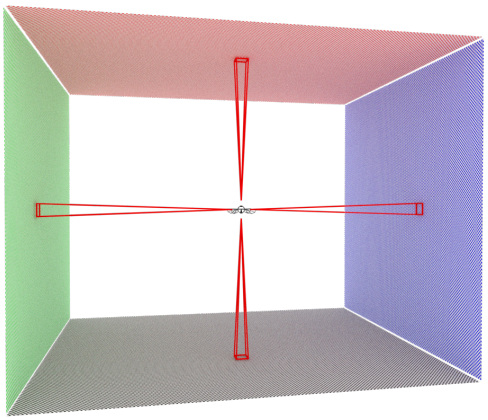


Fig. 3: Robot orientation in Indoor environment. *Direction nomenclature: Walls - X+(Green) & Ceiling - Z+(Red), Y+ (into the room)*

slight fluctuations in their state due to the innate kinematics. This can be exploited to analyze if the fluctuation can provide enough optical flow data for controls. During a linear motion in one direction, the sensor is more sensitive to the changes in the direction perpendicular to the motion. Theoretically, nil optic-flow output is expected in the direction of motion due to constantly expanding and contracting environment. Ideally, there exist fluctuation in the motion, that generates optic-flow data in the direction of motion as-well.

The pictorial representation of the aerial-robot in an indoor environment with ceiling, floor, and walls can be seen in Fig. 3. This environment is developed using checkerbord pattern for providing better contrast. The forward motion is depicted along the *Y* direction (into the room in Fig. 3). The walls influence the sensors facing in the X-direction, similarly the ceiling and floor influence the sensors in the Z-direction. This directional configuration is followed in the rest of the study.

C. Related Optical Flow

On knowing the orientation, the proceeding step involves deriving the optical flow equations for the individual sensors. The general equation for detecting the optic flow is given by the Eq. 1 where v and ω represent the linear and angular velocities of the robot/sensor. Similarly, r indicates the normal distance between the camera/sensor and moving-environment while θ indicates the angle between the moving-environment and direction of bot locomotion.

$$\vec{OF} = \frac{v}{r}\sin\theta + \frac{\omega}{\beta} \tag{1}$$

Considering the general configuration of the MAV as observed in Fig. 3, the equations 3-8 have been derived [10]. Each sensor consists of two scalar flow components that are mutually perpendicular, the sensor X+ can have flow in either *Z* or *Y* direction. This yields one resultant vector, similarly there exist six resultant vectors for all the sensors. Solving

these equations results in obtaining the final trajectory of the robot.

$$\vec{\mathbf{OF}}_{Sensor}^{Direction} \tag{2}$$

The general notation seen in Eq. 2 is followed in the rest of the study where $\vec{OF}$ is the optic flow vector. The subscript indicates the sensor and the superscript shows the corresponding sensing direction.

$$\vec{OF_{X+}} = \frac{-v_y}{r_1}\sin\theta - \frac{\omega_z}{\beta} + \frac{-v_z}{r_1}\sin\theta + \frac{\omega_y}{\beta} \tag{3}$$

$$\vec{OF_{X+}} = [OF_{X+}^{y}, OF_{X+}^{z}]$$

The final optical-flow vector for the X+ sensor is expressed in the Eq. 4, where the first two terms represent the linear and angular velocity relating to the Y direction. Fig. 4 shows the sensor (X+) denoted by green box at the origin translating in the Y direction at velocity V_y. Parallel to the Y-axis, a set of discrete points at distance r_1 shows the optical flow direction. In Fig 4(b) the red dot indicates nil flow as the sensor stops in the final position. This inference shows that optical flow and velocity are opposite in direction, hence in Eq. 4, V_y is negative.

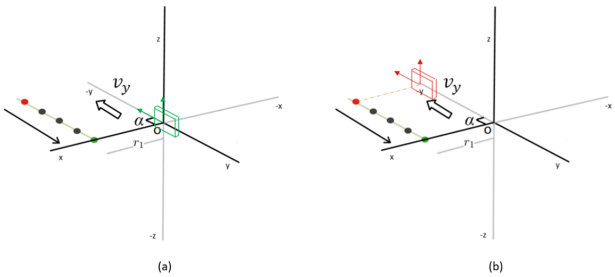


Fig. 4: Linear motion of Sensor in Y direction
(a) Initial Position (b) Final Position

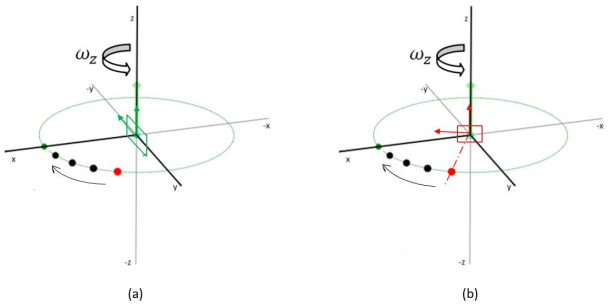


Fig. 5: Angular motion of Sensor along Z axis
(a) Initial Position (b) Final Position

The angular velocity as shown in Fig. 5 represents the sensor at origin and rotating along Z-direction, this induces an optic flow as seen by the dotted points. The factor β is used as a scaling factor to normalize the angular velocity. Similarly, for the X+ sensor linear motion along Z-direction and angular rotation along Y direction exist. Finally, these components combine to form four terms in Eq. 3.

The following equations were derived considering the similar directional sign conventions [10],

$$\vec{OF_{X-}} = [\frac{-v_y}{r_2}\sin\theta + \frac{\omega_z}{\beta}, \frac{-v_z}{r_2}\sin\theta - \frac{\omega_y}{\beta}] \tag{4}$$

$$\vec{OF_{Y+}} = [\frac{-v_x}{r_3}\sin\theta + \frac{\omega_z}{\beta}, \frac{-v_z}{r_3}\sin\theta - \frac{\omega_x}{\beta}] \tag{5}$$

$$\vec{OF_{Y-}} = [\frac{-v_x}{r_4}\sin\theta - \frac{\omega_z}{\beta}, \frac{-v_z}{r_4}\sin\theta + \frac{\omega_x}{\beta}] \tag{6}$$

$$\vec{OF_{Z+}} = [\frac{-v_x}{r_5}\sin\theta - \frac{\omega_y}{\beta}, \frac{-v_y}{r_5}\sin\theta + \frac{\omega_x}{\beta}] \tag{7}$$

$$\vec{OF_{Z-}} = [\frac{-v_x}{r_6}\sin\theta + \frac{\omega_y}{\beta}, \frac{-v_y}{r_6}\sin\theta - \frac{\omega_x}{\beta}] \tag{8}$$

D. Optic Flow Generator - MATLAB GUI

This section briefly discusses the Matlab software developed to be used as optical flow data input for simulation. A standard approach for developing a controlled environment involves moving a screen with a high contrast-pattern at known distance and velocity. The sensor can later be calibrated against the known velocity of the moving-screen. The distance moved by the screen is not very flexible to adjust, the pattern cannot be easily modified, only linear motions can be detected, and also the system is not portable. A new compact system was developed using a Matlab GUI interface on a computer monitor to address the shortcomings of the standard mechanical setup.

A visual representation of the stand-alone software is shown in Fig. 6 with a standard checkerboard pattern. The GUI consist of four checkerboard-screens, as the inference from

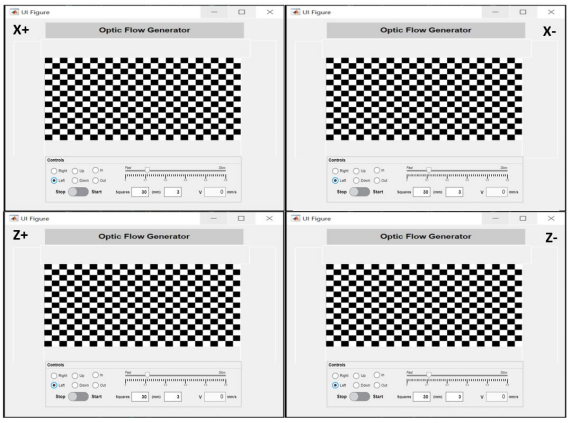


Fig. 6: Matlab GUI for four sensors

Section II-B indicates using four sensors. The pattern is internally computed as a matrix and converted to a grey-scale image with only black and white colors. Similarly, a random matrix can be generated that results in a QR pattern. The core principle of creating motion pertains to moving a small window within the large image that generates the flow. The monitor consisting of four GUI tabs aligned with concurrent sensors for experimenting with various strategies. It contains interactive features to perform various tasks,

The *Start/Stop* function controls the Start, Stop, and Reset function in the GUI. *Direction* commands help to move in four directions (Up, Down, Right, Left) to virtually imitate the sensor environment in different directions. To facilitate vision observed by the MAV in the moving direction, the zoom feature helps to visualize this. In the case of forwarding motion, the front plane is zoomed-in whilst the rear plane is zoomed out. The feature on the pattern can be modified using *Square* function. The datasheet shows the sensor functionality is affected by the texture of the surface used. This parameter helps to vary the texture within a constant surface area. Increasing the window size accommodates more patterns for the same area and vice versa reducing this. There exist an optimum size which can be found via window sweep operation. The velocity of the optic-flow can be increased or decreased for the desired speed. The program by default runs at max-speed and can be reduced/controlled using the delay function. This parameter plays a vital role in knowing the exact velocity of the pattern. The size/aspect ratio of the monitor determines the velocity of the pattern on the screen. The squares tend to move faster on a bigger screen since the run time remains constant but the distance to cover is more. It is a calibration factor and estimated by measuring the mean size of squares on the sensing screen. The *v (mm/s)* feature estimates the exact speed of the moving pattern and displays it. The runtime for a single step is identified from the program and the corresponding distance/squares moved is estimated from the mm parameter. This factor is logged in as an excel file and used to calibrate the ADNS sensor. The process is scalable, meaning the number of sensors used can be varied as per the need. The stand-alone feature allows access to multiple programs at the same time.

III. Experimental Setup

A. Motivation

To control an aerial vehicle with a flight-simulator, the Hardware-in-the-loop experimental setup requires the ability to sense optic-flow data from a minimum of four directions (walls, ceiling, and floor). An overall view of the system developed can be seen in Fig. 7 and Fig 8. The setup contains four ADNS sensors depicting its position as seen in Fig. 3. The three-dimensional existence of an indoor environment is unwrapped on a computer screen and placed adjacent to each other. The sensors are placed on X^+, X^-, Z^+ and Z^- directions of the MAV to sense the optical flow generated on the PC screen via the Matlab program. Two Arduino microcontrollers are used in the setup, wherein one of them is used

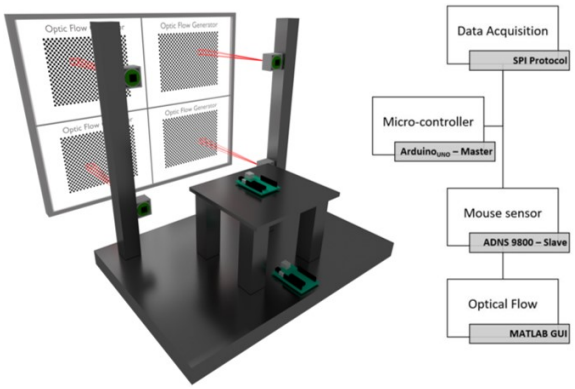


Fig. 7: Experimental setup with data-flow informationt

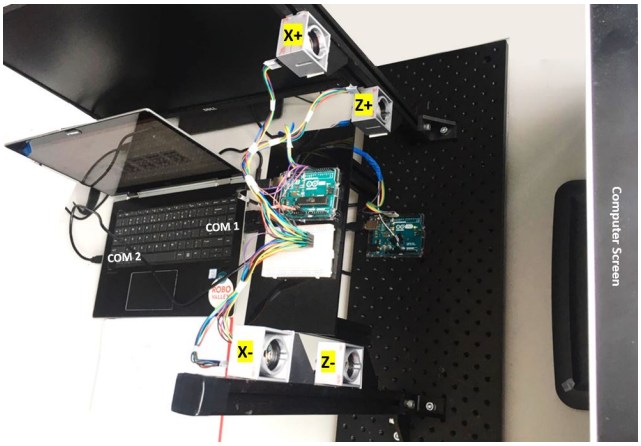


Fig. 8: Complete HITL Flight Simulator Setup Setup

for optic flow detection and the other for viewing the live image. The schematic flow diagram in Fig. 7 explains the master-slave interaction and Arduino micro-controllers role in data transfer and acquisition.

B. Information extraction from ADNS Sensor

Our next objective in the process aims to extract the optic flow data from the sensor and via a suitable micro-controller. Arduino-UNO is used as a master to control the ADNS 9800 sensors that behave as slaves in the architecture. Table I represents the connections made with the sensor using MISO, MOSI, SCLK, and SS.

TABLE I: ADNS 9800 - Arduino Uno Circuit connection

SS	10	MOT	2
MOSI	11	DG	GND
SCLK	13	AG	GND
MISO	12	VI	5V

This procedure is repeated across the other sensors. All the sensors are primarily powered by the Laptop via Arduino through the USB A-B cable. This cable is responsible for both powering the system and for serial communication. It can be

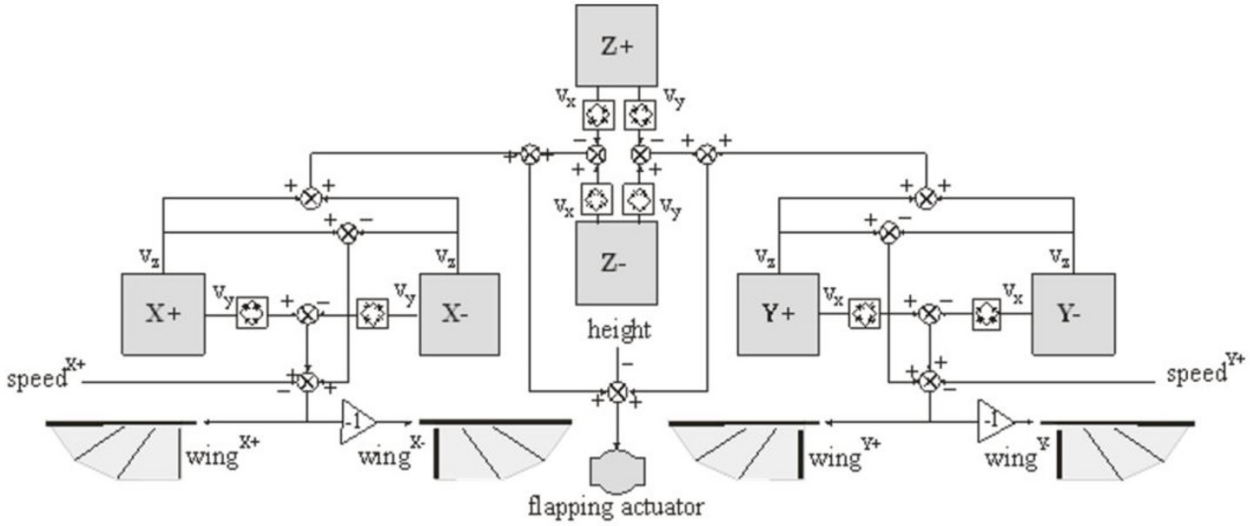


Fig. 9: Control logics of MAV using pure optic-flow data [7]

observed in Fig. 10 that two Arduino microcontrollers are used. The circuit remains the same in both scenarios whilst different programs are uploaded for specific tasks. Depending on the requirement, the system can be made to behave as a camera or to detect optic-flow. The code is programmed in Arduino programming language with .ino extension and is uploaded into via COM port. The sensed data is later saved in a local drive using the Processing software, exclusively for post-processing.

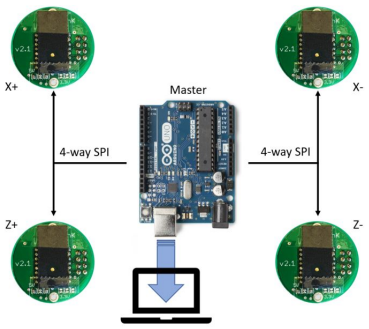


Fig. 10: ADNS 9800 - Arduino UNO communication schematic

IV. Controls for MAV

The motivation to develop a flight simulator aims to determine if the sensed signal is usable for controls and to predict the trajectory. To understand the interaction between optic-flow and controls of an autonomous MAV, pure optic-flow based controls [7] was preferred due to the simple architecture. Various control-approaches can be tested with the current setup. Fig. 9 explains the control logic that uses the inputs from six optical-flow sensors. As seen in the schematic, the combinations of sensed data are evaluated using simple arithmetic functions to propose commands to control the MAV. The lift production occurring in the system can be varied thereby supporting the steering operation in the intended direction. Take-off or landing operation depends on the power supplied to each actuator/wing. The lift production for ascent or descent operation directly depends on the power consumption. In case of a collision scenario, adhering to the current controls, the MAV automatically steers away. Since the object in proximity produces much higher optic-flow, to maintain equilibrium, the MAV steers in a direction away from the obstacle. Hovering is considered to be the most challenging task since there exists almost negligible optic-flow data during this period. The flight-simulator can be used to simulate a hovering condition by inducing a periodic motion with a small amplitude in the Matlab GUI.

The architecture as seen in IV of MAV with the sensor position enables detection in all the zones and reduces the robot blind-spots. Distinctions can be made on the type of motions based on the sensed optic flow. Individual actuation of the wings/rotors in the MAV help in steering and to fly in any direction.

This section explains the basic flight maneuvers of a MAV for stabilization proposed by Goosen [7]. To maintain an upright orientation, the optic-flow from X and Y sensors in Z direction are considered. The concept can be clearly explained using the equations below, a net positive value obtained in Eq 9 suggest that optic-flow is more on the X+ side of the MAV, inferring it is closer to the wall. This can be countered by supplying more power to the X+ actuator/wing of MAV. The control-signal to monitor the power supplied to the X+ actuator is indicated by C^{act}_{X+}. For a net negative value obtained for the control-signal C^{act}_{X+}, the power supplied to the X+ actuator will be reduced, helping it navigate towards the X+ direction.

Similar logic can be applied for the sensors in the Y direction given by Eq. 10.

In the equations below, OF^z_{X+} and OF^z_{Y+} represents the flow detected by X+ and Y+ sensor in Z direction,

$$C^{act}_{X+} = OF^z_{X+} - OF^z_{X-} \tag{9}$$

$$C^{act}_{Y+} = OF^z_{Y+} - OF^z_{Y-} \tag{10}$$

OF^z denotes the overall optic-flow sensed in the Z direction by the X and Y sensors,

$$OF^z = OF^z_{X+} + OF^z_{X-} + OF^z_{Y+} + OF^z_{Y-} \tag{11}$$

Altitude stabilization can be implemented by balancing the flow in Z sensors. The difference in X and Y components of Z+ and Z- sensors helps to predict a relative altitude given by OF^{alt} in equation below,

$$OF^{alt} = (OF^x_{Z-} - OF^x_{Z+}) + (OF^y_{Z-} - OF^y_{Z+}) \tag{12}$$

The difference between actual height and overall flow in the Z direction with relative altitude facilitates to predict the power required for the MAV. The control-signal C^{MAV} indicates the power required for the flight. α is the conversion factor used for dimensional equality,

$$C^{MAV} = (OF^z - \alpha.height) + OF^{alt} \tag{13}$$

Implementing these control-logics adhering with prior direction nomenclature solves the issue of obstacle avoidance. This can later be tuned with a PID for better performance.

V. Results and Discussions

This section mainly discusses the results obtained from the Hardware-in-the-loop (HITL) flight simulator experimental setup. The main focus relies on predicting the trajectory and to analyze the controllability of the sensed signal.

The following set of parameters were unchanged during the experiment to maintain consistency in the results compiled. All four sensors were used from the same supplier coupled with a $f20$ lens. The PC screen used is HP 24" with a resolution of 1680 $\times$ 1050 and operates at 60 Hz. The data acquisition rate from the Arduino UNO micro-controller has been matched with computer operating frequency to improve the data points obtained. The distance between the PC screen and sensor was unchanged and maintained at 320 mm in the developed flight simulator. No external lighting was used in the indoor environment apart from the available ambient lighting.

The logics discussed in Section IV are used to predict the path. The only required information to control the MAV relates to the optic-flow from surrounding directions. Velocity can be predicted from the sensed optical flow.

A. Translational Motion

To realize the complete controls developed by Goosen [7] and Selvan [10], a simple linear forward motion is considered for the analysis. The flight simulator is initiated to replicate a linear motion occurring in the Y-direction as seen in Fig. 13. To achieve this, the simulator needs to induce linear optic flow in the sensors facing in X and Z direction. A translational motion requires pitching along the X-axis. This means the thrust on actuator Y+ needs to be more than the counterpart thereby enabling a forward motion (Fig. 13).

The optic flow and velocity of the moving robot are in the opposite direction. Hence during linear motion along the Y direction, the X+ sensor is subjected to a linear *left* motion on the corresponding screen on the flight-simulator monitor. Similarly, the other motions for the sensors are configured as shown in Table II. The control inputs with the sensor orientation are shown in Fig. 11.

The raw data acquisitions from all the four sensors are plotted in Fig. 12. Each sensor produces two vector components resulting in eight individual plots. It can be seen that since a linear translational motion is intended without any direction or velocity change, apart from motion along Y-direction, it replicates a rectilinear motion. It can be seen that due to the constant velocity set for the checkerboard pattern, the optical flow detected from all the sensors eprimarily in Y-direction and outputs at 10 cps (counts per second). Plots in the first row from Fig. 12 show the vector-flow in y-component from all the sensors and the second row almost operates with zero-flow (almost 0 cps). This data is used to predict the trajectory and required moment for intended motion.

TABLE II: Simulator inputs for Linear-motion along Y

Sensor	**Flow Generator - Motion**			
	X+	**X-**	**Z+**	**Z-**
X+	Right			
X-		Left		
Z+			Down	
Z-				Up

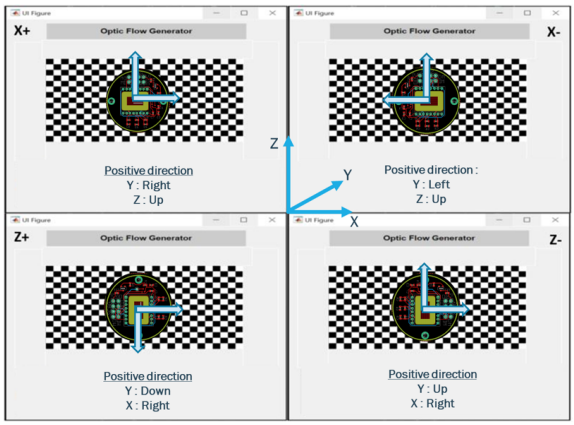


Fig. 11: Sensor orientation with GUI interface for Controls

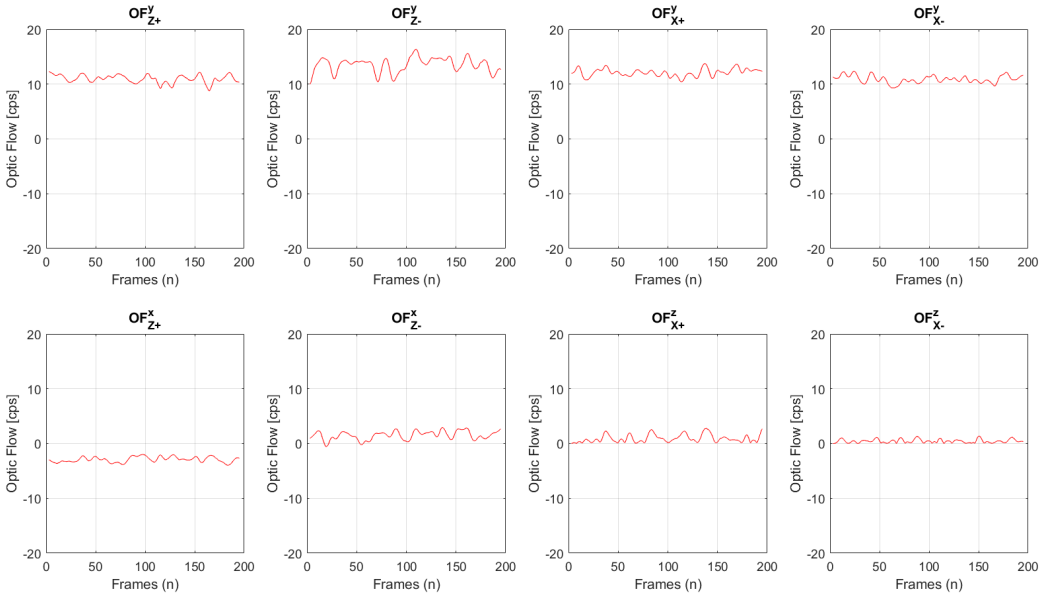


Fig. 12: Combined data collected from all the 4 sensor in X and Z direction

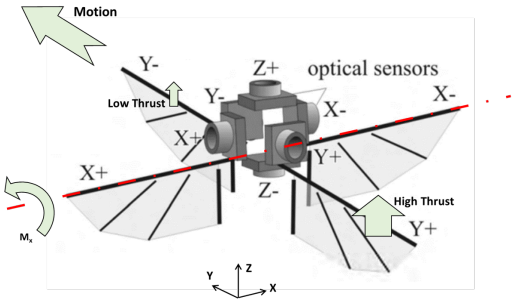


Fig. 13: MAV nomenclature indicating translational motion along Y-direction pitching along x axis (Moment M_x)

1) Trajectory: The complete trajectory predicted from the sensed data can be visualized in Fig. 14. The plot represents the spatial position of the robot in a two-dimensional X-Z plane plotted for each captured frame along time. This is indicated by the blue path in the figure, the actual planned-path is indicated in green. In an ideal environment, the position should have been in the middle i.e., (x,z) at (0,0), since the optical-flow is equal on all the four sides for the MAV. The minor deviation exists due to the non-zero optical flow vectors in the x-direction from Z+ sensor and Z- sensors (from Fig. 12). Also, the slightly higher optical flow in Y-direction contributes to this variation.

Initially the raw Optic flow data is acquired, the resultant is found, considering them as two perpendicular vectors,

$$\vec{OF_{X+}} = \sqrt{\vec{OF^{y}_{X+}}^{\,2} + \vec{OF^{z}_{X+}}^{\,2}} \tag{14}$$

The angle between the z and y component of the X+ sensor can be computed as follows

$$\theta_{X+} = tan^{-1}\left(\frac{\vec{OF^{z}_{X+}}}{\vec{OF^{y}_{X+}}}\right)$$

Similarly the resultant and angle are analyzed,

$$\vec{OF_{X-}} = \sqrt{\vec{OF^{y}_{X-}}^{\,2} + \vec{OF^{z}_{X-}}^{\,2}} \tag{15}$$

$$\theta_{X-} = tan^{-1}\left(\frac{\vec{OF^{x}_{X-}}}{\vec{OF^{y}_{X-}}}\right)$$

Finally the resultant for the X+ and X- sensors are computed with ϕ being the angle between the individual resultants.

$$\vec{OF_X} = \sqrt{\vec{OF_{X+}}^{\,2} + \vec{OF_{X-}}^{\,2} + 2\vec{OF_{X+}}\vec{OF_{X-}}cos\phi} \tag{16}$$

$$\phi = \Delta(\theta_{X+}, \theta_{X-})$$

The relative position of the MAV in x and z directions are given by,

$$\begin{bmatrix}\delta x\\ \delta z\end{bmatrix} = \begin{bmatrix}mod(\vec{OF_{X+}} - \vec{OF_{X-}})\\ mod(\vec{OF_{Z+}} - \vec{OF_{Z-}})\end{bmatrix} \tag{17}$$

A similar procedure is repeated, to find the resultant vector

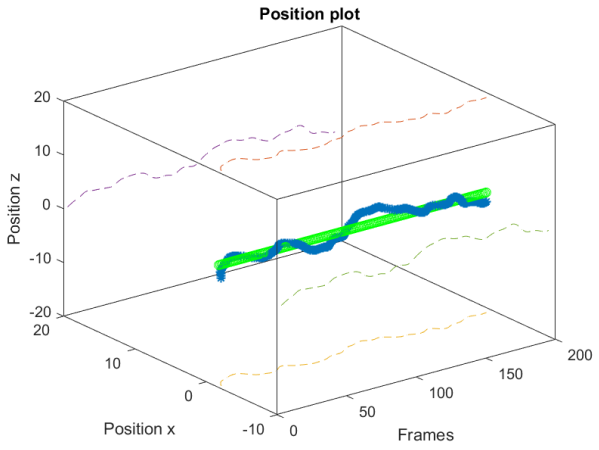


Fig. 14: Trajectory from sensed optic-flow HITL Setup (Green: planned path, Blue: sensed path)

for the sensor in the Z-direction. The Eq. 17 represents the difference in the resultant optical-flow vectors in Eq. 14 and Eq. 15. The relative position (from optic-flow) of the MAV is plotted in the X-Z plane,

There exists a deviation from the actual path that is charted as an error plot in Fig. 15. The Δx and Δz indicate the variation in the paths. This variation exists due to various possible reasons, some of the major reasons being the flickering screen of the digital monitor, slightly rough flow from the sensing environment, and ambient light available. The error can be estimated using the conventional method as per Eq. 18, and the average deviation for all the sensors yield a value of 4.83 %. The individual errors are estimated and can be seen in Table III,

$$e = \frac{\Delta OF_{X+}}{OF_{X+}} \times 100 \tag{18}$$

TABLE III: Error analysis

Sensor	Error (%)	Sensor	Error (%)
OF_{X+}	3.68	OF_{Z+}	4.83
OF_{X-}	4.06	OF_{Z-}	6.73

2) Control Signals: Optic-flow data acquired from the simulator can be used to predict the type of motion steered by the MAV. A set of control signals (M_x, M_y and ΔF) can be estimated based on pure optic-flow to understand the actuation required in terms of moment and force along X and Y directions [10]. The following control equations are required to stabilize the MAV. Since the motion occurs along the Y-direction, the moment is applied across the X-axis of the MAV (Fig. 13). The control parameters are estimated and are plotted in Fig. 16. It reveals the existence of a notable moment control-signal M_x required for pitching. Also, the control-signal required for rolling shows a negligible M_y value. The altitude-stabilization control signal is determined by the following equations. As maintaining altitude is considered the default state, the force required to stabilize the altitude is almost zero from the Eq. 21 as seen in Fig. 16. Table IV shows the contributing factor that corresponds to the amplification factor associated with the control signal estimation.

$$\begin{aligned} M_y = {} & a_{rot}(OF^z_{X+} - OF^z_{X-}) \\ & + a_{displr}(OF^x_{Y+} + OF^x_{Y-} + OF^x_{Z+} + OF^x_{Z-}) \\ & + a_{aavoid}(|OF^y_{X+}| - |OF^y_{X-}|) \end{aligned} \tag{19}$$

$$\begin{aligned} M_x = {} & a_{rot}(OF^z_{Y+} - OF^z_{Y-}) \\ & + a_{displr}(OF^y_{X+} + OF^y_{X-} + OF^y_{Z+} + OF^y_{Z-}) \\ & + a_{aavoid}(|OF^x_{Y+}| - |OF^x_{Y-}|) \end{aligned} \tag{20}$$

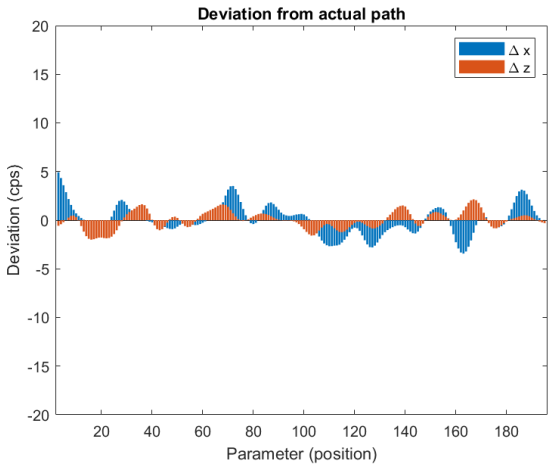


Fig. 15: Deviation in OF data from actual data

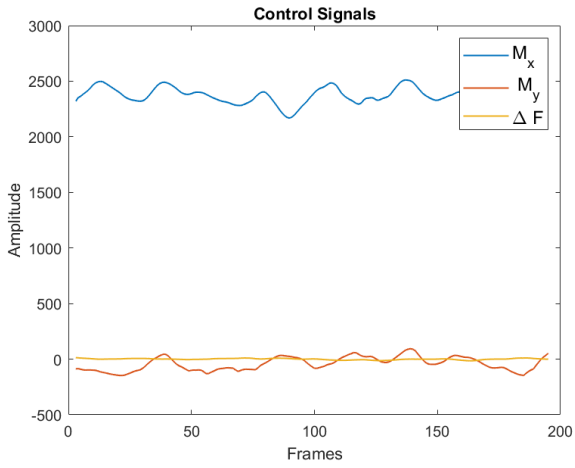


Fig. 16: Control Signals for MAV Stabilization

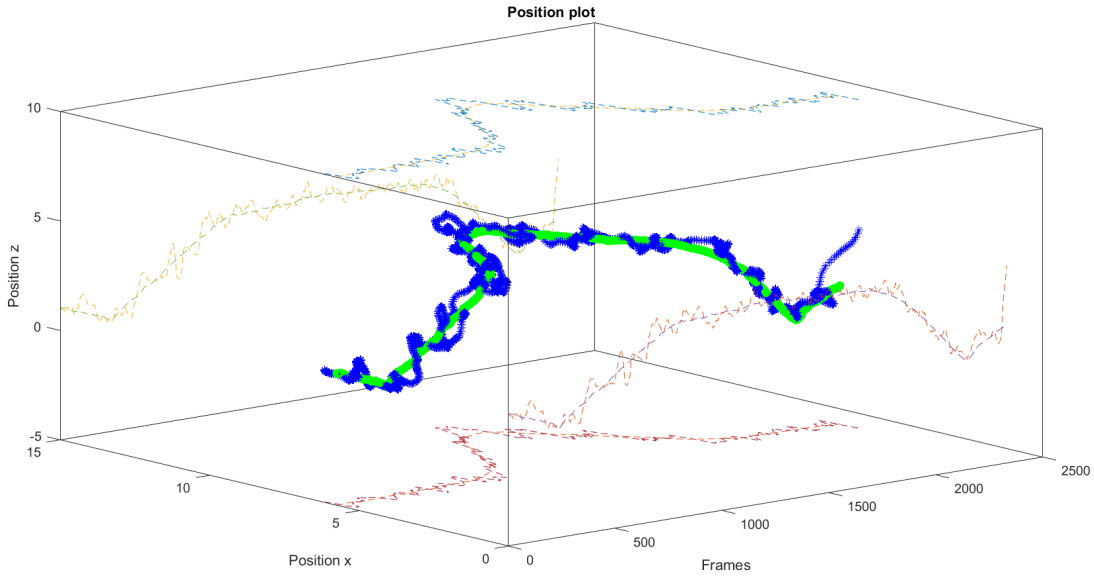


Fig. 17: Trajectory from sensed optic-flow HITL Setup (Green: planned path, Blue: sensed path)

$$\Delta F = a_{displr}(OF^z_{Y+} + OF^z_{Y-} + OF^z_{X+} + OF^z_{X-}) + a_{aavoid}(|OF^x_{Z+}| - |OF^x_{Z-}|) + a_{aavoid}(|OF^y_{Z+}| - |OF^y_{Z-}|) \quad (21)$$

TABLE IV: Amplification parameters

Parameters	a_{arot}	a_{aavoid}	$a_{adisplr}$	$a_{adisplt}$
Value	50	4	50	4

B. Random motion

Linear motion analysis of the MAV was explained earlier. This experiment aims to predict a more random path of the MAV from the given input in the optical-flow environment. In a real-world scenario, the MAV changes its position in both directions which varies the Field of View depending on the distance moved. This means the number of features (squares in the checkerboard pattern) detected varies. This criterion can be included in the current setup by varying the window size. An assumption is made here such that the distance moved by MAV is within the sensitive window size (W20-W40). In this zone, the sensor output does not vary much for constant velocity, given the features sensed vary [9].

In an open-loop configuration, the resultant of optic flow data from four sensors are compiled to locate the position in the X-Z plane. The resultant at each frame is compiled and plotted according to the Eq. 17. To find the deviation of the sensed path from the originally intended path, they both are plotted on the same Fig. 17. The projection of the planned and sensed path can be observed in the planes to have a better idea of the deviations. All the analysis performed for the linear

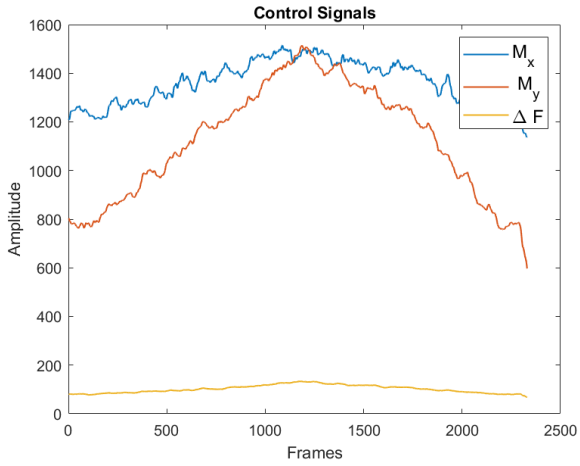


Fig. 18: Control signals predicted from sensed OF data for random path

motion earlier is performed. The most contributing factors for flight-stabilization are the control signals shown in Fig. 18 [10]. The major difference is the complex trajectory generated from the HITL simulator where all the types of motion with varying velocity and direction are performed. Hence the ascent and descent characteristics are observed as well.

C. Insights from ROS Gazebo

A brief comparison of the existing Hardware-in-the-loop (HITL) setup with Software-in-the-loop (SITL) architecture in

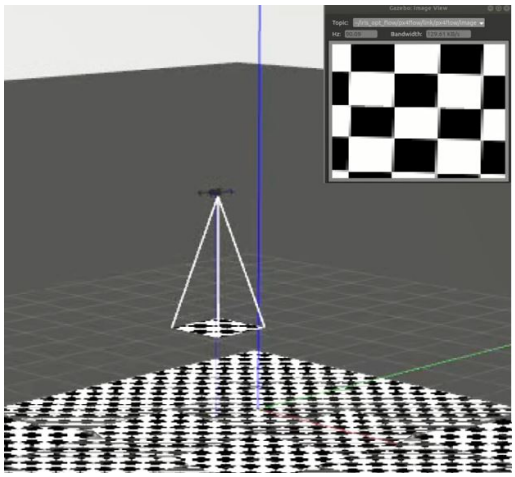


Fig. 19: PixHawk drone simulated in ROS-Gazebo environment with optic flow sensor data

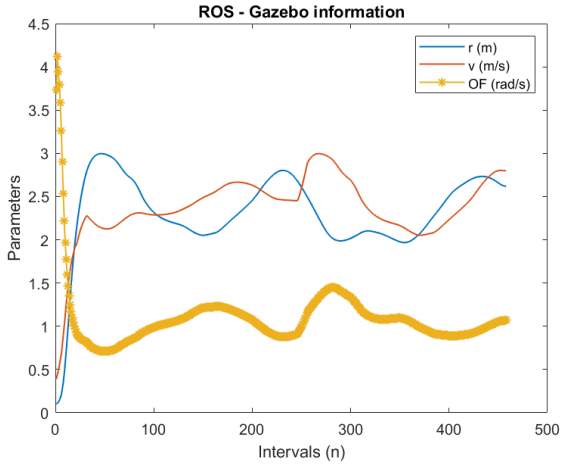


Fig. 20: Optic flow estimation from ROS data

the ROS Gazebo environment is made to gain better insights into the real world scenario. HITL runs on the actual hardware using firmware, whereas SITL functions in a simulated environment on a development computer that uses specific firmware for the environment. A pictorial representation of the ROZ-Gazebo environment can be seen in Fig. 19.

The purpose of this comparison aims to gain a wider perspective for comparing the optic-flow data from ADNS 9800 with an optical flow sensor. To complete this process, PX4 flight control software for drones that provided flexible tools to control was chosen. The Iris drone was selected among the available options as it supported easy accommodation of the standard px4-optical-flow sensor to be integrated. This drone-sensor setup was simulated in the ROS-gazebo environment. A standard checkerboard pattern was used to replicate the scenario from the HITL setup. However, the drone used various sensor inputs for its operation, which is not the case for us. Hence Data is collected from translational motion generated in the field. The compiled results obtained from SITL can be seen in Fig. 20. As the major contributing factor for the optic flow depends on the velocities and distance from the obstacle (floor in this case), only the related results are discussed. The optic flow is estimated from the altimeter and the velocity of the moving drone. Since is drone was moving at a height of 2 m above the ground at around velocity of 2 m/s, almost 1 rad/s value was obtained. The image captured by the px4 optical flow sensor remains with undistorted resolution until at least 5 m above the ground level.

VI. Conclusion

The results imply the HITL simulator predicts the trajectory within deviations of around 5% from the actual path. The variations in the predicted trajectory can be reduced by optimizing for a smoother pattern flow in the Matlab-GUI. The estimated control-signals comply with the actual scenario from the flight-simulator. The setup can be adapted for various trajectories. In this aspect, different control logics can be tested for post-processing. The intended purpose for the simulator is hence proven.

Future recommendations would suggest to include six optical-flow sensors to consider the zooming in and out the background during translation motion. Ideally, the constant expansion of the environment would produce zero optic flow data, as the mutually perpendicular vectors are equal and opposite in direction. This logic can be used for landing and takeoff operations. Also, feeding the data from Arduino directly into the Matlab environment would provide closed loop scenario for live-path. This would also enable a real-time PID tuning setup for the MAV. The HITL setup is suspected for good use in simulating ground-vehicle motion that use optic-flow.

References

[1] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[2] Bl Horn and KP Berthold. Schunck. determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[3] Jianbo Shi et al. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE, 1994.

[4] Bruce D. Lucas and Takeo Kanade. Iterative Image Registration Technique With an Application To Stereo Vision. 2:674–679, 1981.

[5] Mandyam V Srinivasan, Shaowu Zhang, and Javaan S Chahl. Landing strategies in honeybees, and possible applications to autonomous airborne vehicles. *The Biological Bulletin*, 200(2):216–221, 2001.

[6] Jean-Christophe Zufferey, Adam Klaptocz, Antoine Beyeler, Jean-Daniel Nicoud, and Dario Floreano. A 10-gram microflyer for vision-based indoor navigation. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3267–3272. IEEE, 2006.

[7] J. F.L. Goosen. Design aspects of a bio-inspired flying sensor node. *Proceedings of IEEE Sensors*, pages 1–4, 2012.

[8] Light Controlcollision, Avoidance For, Quadcopterflapping Wing, Using Only, and Optical Flow. F LIGHT CONTROL AND COLLISION AVOIDANCE FOR. 2018.

[9] J G Suresh Subramonian and J F L Goosen. Evaluation of Optical flow sensors for Indoor robotics application and Flight simulator.

[10] Abhimanyu Selvan. Simulation of optic flow based flight control for a flapping wing micro aerial vehicle. page 70, 2014.

Chapter 6

Conclusions and Recommendations

This section deals with the results and observations from earlier sections. The individual inferences about the experimental-analysis concerning with the ADNS sensor and the flight-simulator setup are discussed earlier in the previous chapters. An overall gist of the major findings in a condensed form is discussed here. Also, the second-half concludes with relevant recommendations.

The primary objective of the study focuses on understanding the possibility of using a suitable optical-flow sensor satisfying SWaP (Size, Weight, and Processing) for indoor MAV applications. The feasibility study was accomplished with the chosen ADNS-9800 optical mouse sensor. The results are discussed in two major aspects relating to the evaluation of individual sensor and the combination of sensors used to develop the HITL Simulator. The parameters considered for the evaluation varies in each scenario, and are discussed here.

6-1 Discussion for Sensor evaluation - ADNS9800

This section contains the vital information gathered from the ADNS 9800 optical mouse sensor for the feasibility to be used as an optical flow sensor. Among the various sensor parameters analyzed, the ability to detect the change in the direction and speed are the most important features considered.

- The sensor initially required slight modifications for better functioning

  - To adapt the sensing at ambient conditions, the aperture was increased from 0.25mm to 2mm.

  - Necessary sealing was made on the sensor-encasing to counter the excess optical noise.

- The output was verified as an optical-flow vector, as there existed a change in magnitude and direction by varying the velocity and direction of the sensing environment.
- The most sensitive feature for the sensor to maximize the output for a constant velocity was found to be W30 in the developed Matlab GUI.
- The operating zone was improved beyond 40% from initial conditions, in terms of contrast and brightness. Considerable dependance on contrast and brightness remains.
- A linear operating zone was identified for sensed optic-flow with actual velocity as per the relation $OF_{adns} = 0.2588 \times v_{gui}$.
- Comparisons with actual analog generated optical-flow shows a very similar trend,

6-1-1 Conclusions

The results compiled lead us to conclusions in using the optical mouse sensor as an Optic-flow sensor. Initially, the sensor required increasing the aperture by 1.75mm coupled with suitable optics. The sensor has been proven sufficient to be used for indoor conditions under normal lighting. Also, the linear behavior for the set parameters proves reliable. The comparison with the analog movement of flow pattern proves that the velocity deducted is within a error of 90mm/s. This can be reduced by calibrating the Γ for analog environment. The current experiment helps us to proceed further with building a Hardware-in-the-loop flight simulator for controls and path prediction applications.

6-2 Discussions related to the HITL Flight simulator

This context focuses more on developing a flight simulator using four sensors to predict the trajectory and to understand the controllability of the signal. The motivation behind developing a flight simulator for the MAV application concentrates more on understanding pure optic-flow based controls. Most of the kinematic factors of the robot are not considered here as the influencing parameter for controls only depends on optical-flow. The digital environment with Matlab GUI supports the generation of various path profiles. Comparing

the actual path with the sensed data gives the deviation from the actual position. On further identifying the plant for the system, it can suitably be tuned using the PID controller. The supporting parameters can be studied for sensitivity and stability. Sensors positioned in the HITL simulator can be considered as replicating the original position on MAV and accordingly the data can be fed into the system.

6-2-1 Conclusions

The results imply the HITL simulator predicts the trajectory with acceptable deviations (4-5%) from the actual path. Also, the control-signals estimated, comply with the actual scenario. The deviations in the predicted trajectory can be reduced by optimizing for a smoother pattern flow in the Matlab-GUI. The setup can be varied for various trajectories. Also, different control logics can be tested for post-processing. The intended purpose for the simulator is hence proven.

Overall Concept

The crux of the working principle involving Atalanta is developed with vision sensing system using optic-flow. The fundamental requirement for sensing optic flow requires the movement of images with detectable contrast. For giving a better idea, a Blender model with comparable technical specifications in terms of FoV, resolution and contrast pattern (checkerboard) was developed using all the six sensors (Section 4).

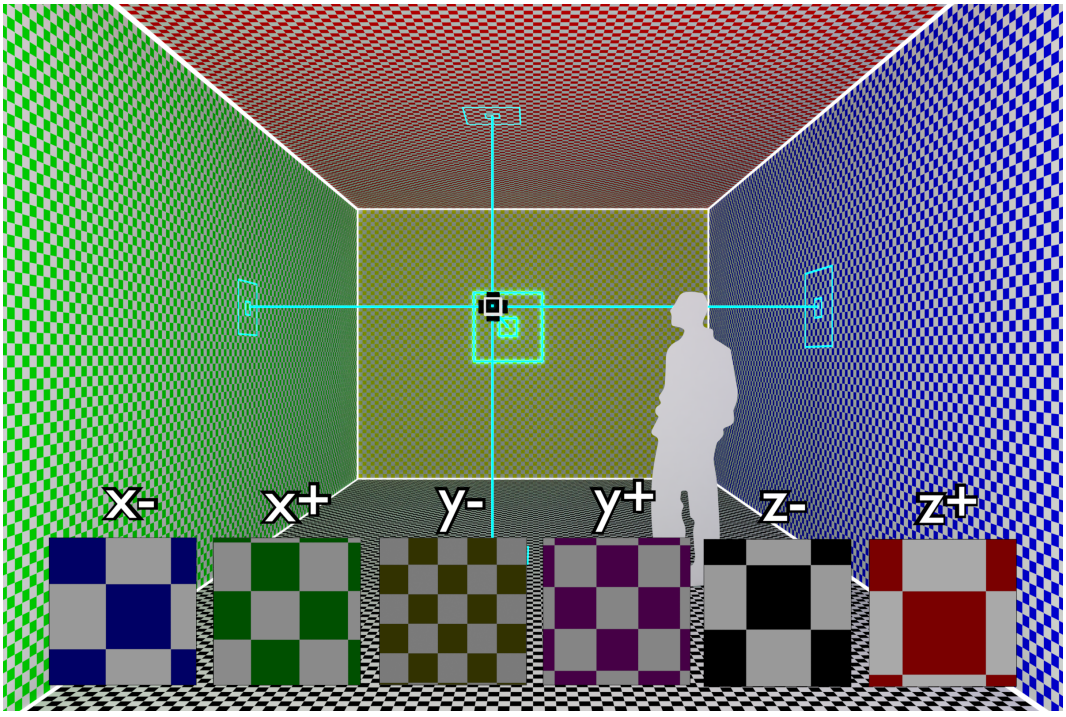


Figure 6-1: Complete system of Atalanta in indoor environment before Obstacle
Direction nomenclature: X+(Green), Y+(Pink - Not visible), Z+(Red)

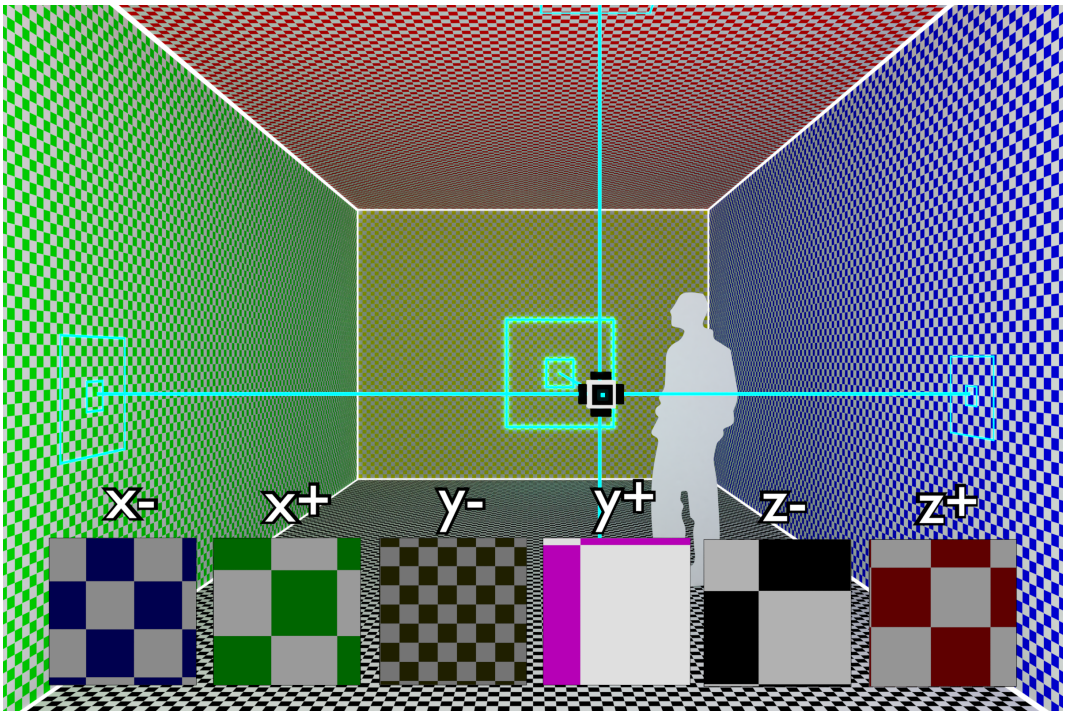


Figure 6-2: Complete system of Atalanta in indoor environment after Obstacle
Direction nomenclature: X+(Green), Y+(Pink - Not visible), Z+(Red)

A representation of this model in an indoor scenario can be seen in Figure 6-1 and Figure 6-2.The sensor outputs can be seen in bottom with the FoV (around 2-3^o) of ADNS 9800. The checkerboard-colors are used for illustration purposes in distinguishing the individual sensor outputs. The real-time sensor output from the illustration containing data from six directions can be visualised. More squares (features) in the sensor output indicate the aerial-robot is away from the surface (Y-, Y+, and X+). The Y- sensor output is dimmed due to the less availability of light due to the farther distance. Similarly the other sensors are relatively closer to the wall. To understand the concept of obstacle avoidance, a human is introduced in the indoor scenario. As soon as the presence is detected in Figure 6-1, a sudden increase in the OF occurs that steers the robot away (Moment along Y, more flapping in X- wing). This phenomenon can be observed in Figure 6-2. These concepts help in better understanding of the MAV-control.

6-3 Recommendations

The current study has focused more on developing a flight simulator for pure Optic-flow based controls of autonomous MAVs. The study has been developed as a proof-of-concept for the simulator to analyze the flight path. ADNS 9800 is the hardware used in the loop for the detection of optic flow. Usually, autonomous robots use multiple sensor inputs for the controlling of a robot. In the current context, a more simple SWaP satisfying optic-flow based control strategy is attempted to be solved. There exist improvements that can be implemented for the betterment of the system. There can be enhancements made on the

software and hardware aspects. The current study relates more to a generic approach to building the vision sensing system. The resolution of the sensed data is very minimal and cannot be used to identify a specific person or object. However, to make the solutions more specific to Atalanta, significant changes/improvements can be performed.

6-3-1 Hardware

ADNS 9800

To accommodate the sensor in the Atalanta robotic fly, the sensor is still heavy. It is believed the sensor weight can be substantially reduced by removing the unwanted components. The CMOS chip is the major contributing component and others are supporting material. This can enable the accommodation of all the six sensors within reasonable constraints. The major weight reduction can be conceived by replacing the existing glass lens with other materials like fiber or film sheets as well. Investigation in this direction helps to build a more light weight system.

Experimental setup

The HITL setup can be upgraded using a larger number of sensors. An alternative would be to change the position of the sensor in the MAV. Currently, the sensors are placed in cartesian coordinates. The basic concept that each sensor can give two directions vectors indicates the bare minimum requirement of four sensors to know the complete environment. Different approaches can be considered in the positioning of the sensors as seen in Figure 6-3. The red dots in the Figure indicate the presence of sensors in the corners while the green flats indicate the sensors placed in a 45 chamfer position. Care must be taken to ensure the blind-spots during the design and development phase. All these may improve the sensing, but more processing of the data may be required. A good trade-off can be made to investigate the possibilities.

6-3-2 Software

Future recommendations mainly include using six-optical flow sensors. Also, the loop can be closed by feeding the data from Arduino-Uno directly into the Matlab environment. The program is easy to modify and implement changes.

- The sensor is subject to linear motion, it is possible to test rotational motion.
- Six optical flow sensors can be used for a complete simulation of the control logic.
- Inclusion of zooming in and out motion for optic-flow detection in the MAV translation direction.
- The Matlab GUI can be optimized more for a smooth flow to have more linear behavior.
- The testing experimented with ROS in the uni-directional sensor can be simulated using multiple sensors.

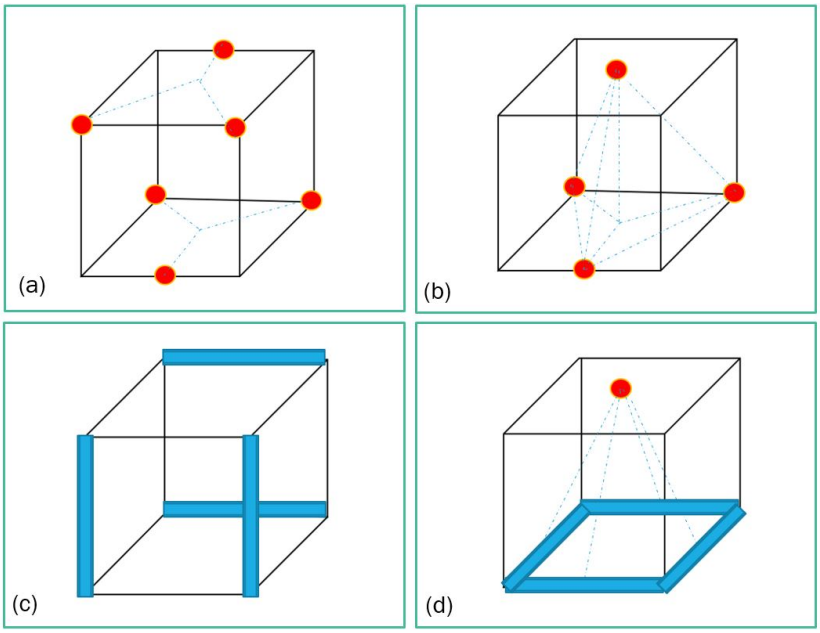


Figure 6-3: Various possible configurations for sensor placement on the MAV

The constants used in the prediction of control signals are referred from Literature. It can be useful to understand these better and optimize the values as per the need.

In conclusion, the final objective for verifying this model can be tested using an in-house developed MAV/Drone. This can be a basic drone with a simple architecture coupled with the six ADNS 9800 optical mouse sensors. The presence of an on-board micro-controller like Arduino-UNO would facilitate SPI communication between the master (Arduino UNO) and slave (ADNS 9800). Also, they permit analysis of ADNS 9800 data and to send commands for controlling the individual motors, thereby the complete robot (MAV / FWMAV). It can be initially tested in a scaled checkerboard-environment and further investigation can be made depending on the results. The HITL setup is suspected for good use in other autonomous-robotics application that use optic-flow as feedback for their controls.

Bibliography

[1] M. Hassanalian and A. Abdelkefi, "Design, manufacturing, and flight testing of a fixed wing micro air vehicle with zimmerman planform," *Meccanica*, vol. 52, pp. 1265–1282, Apr 2017.

[2] "impact, howpublished = https://ww.dronedj.com/2019/08/26/dji-phantom-mavic-pro-crashes-head, note = Accessed: 2020-01-17."

[3] C. T. Bolsman, *Flapping wing actuation using resonant compliant mechanisms*. 2010.

[4] H. V. Phan and H. C. Park, "Insect-inspired, tailless, hover-capable flapping-wing robots: Recent progress, challenges, and future directions," *Progress in Aerospace Sciences*, vol. 111, no. March, p. 100573, 2019.

[5] J. C. Zufferey, *Bio-inspired flying robots: Experimental synthesis of autonomous indoor flyers*. 2008.

[6] J. F. Goosen, "Design aspects of a bio-inspired flying sensor node," *Proceedings of IEEE Sensors*, pp. 1–4, 2012.

[7] S. Thakoor, J. M. Morookian, J. Chahl, B. Hine, and S. Zornetzer, "Bees: Exploring mars with bioinspired technologies," *Computer*, vol. 37, no. 9, pp. 38–47, 2004.

[8] S. J. Huston and H. G. Krapp, "Visuomotor transformation in the fly gaze stabilization system," *PLoS biology*, vol. 6, no. 7, 2008.

[9] R. Mockel, "Bio-Inspired Optical Flow Vision Sensors for Visual Guidance of Autonomous Robots," *PhD Thesis*, no. 20736, 2012.

[10] A. Cavoukian, *Privacy and drones: Unmanned aerial vehicles*. Information and Privacy Commissioner of Ontario, Canada Ontario, 2012.

[11] "History of Drones, howpublished = https://understandingempire.wordpress.com/, note = Accessed: 2020-01-07."

[12] "Drones, howpublished = https://internetofthingsagenda.techtarget.com/definition/drone, note = Accessed: 2020-01-07."

[13] "Weight Specification, howpublished = https://www.faa.gov/regulations_policies/rulemaking/recently_published/media/2120-aj60_nprm_2-15-2015_joint_signature.pdf, note = Accessed: 2020-01-07."

[14] W. Shyy, Y. Lian, J. Tang, D. Viieru, and H. Liu, *Aerodynamics of low Reynolds number flyers*, vol. 22. Cambridge University Press, 2007.

[15] D. E. Alexander, *Nature's flyers: birds, insects, and the biomechanics of flight*. JHU Press, 2002.

[16] V. Malolan, M. Dineshkumar, and V. Baskar, "Design and development of flapping wing micro air vehicle," in *42nd AIAA Aerospace Sciences Meeting and Exhibit*, p. 40, 2004.

[17] M. Keennon and J. Grasmeyer, "Development of two mavs and vision of the future of mav design," in *AIAA International Air and Space Symposium and Exposition: The Next 100 Years*, p. 2901, 2003.

[18] C. Galiński and R. Żbikowski, "Insect-like flapping wing mechanism based on a double spherical scotch yoke," *Journal of the Royal Society Interface*, vol. 2, no. 3, pp. 223–235, 2005.

[19] C. Galiński and R. Żbikowski, "Materials challenges in the design of an insect-like flapping wing mechanism based on a four-bar linkage," *Materials & design*, vol. 28, no. 3, pp. 783–796, 2007.

[20] A. Selvan, "Simulation of optic flow based flight control for a flapping wing micro aerial vehicle," p. 70, 2014.

[21] L. Controlcollision, A. For, Q. Wing, U. Only, and O. Flow, "F LIGHT CONTROL AND COLLISION AVOIDANCE FOR," 2018.

[22] M. Kiani, B. Davis, F. P. Quevedo, N. Cabezut, S. Hince, M. Balta, and H. Taha, "A new bio-inspired flying concept: The quadflapper," *AIAA Scitech 2019 Forum*, no. January, pp. 1–12, 2019.

[23] M. Karásek, F. T. Muijres, C. De Wagter, B. D. Remes, and G. C. de Croon, "A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns," *Science*, vol. 361, no. 6407, pp. 1089–1094, 2018.

[24] B. Y. F. V. a. N. Breugel, W. Regan, and H. O. D. Lipson, "Demonstration of a Passively Stable, Untethered Flapping-Hovering Micro-Air Vehicle," *Machine Design*, no. December, pp. 68–74, 2008.

[25] M. Keennon, K. Klingebiel, H. Won, and A. Andriukov, "Development of the nano hummingbird: A tailless flapping Wing Micro Air Vehicle," *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, no. January, pp. 1–24, 2012.

[26] H. V. Phan, T. Kang, and H. C. Park, "Design and stable flight of a 21 g insect-like tailless flapping wing micro air vehicle with angular rates feedback control," *Bioinspiration & biomimetics*, vol. 12, no. 3, p. 036006, 2017.

[27] F. Fei, Z. Tu, Y. Yang, J. Zhang, and X. Deng, "Flappy hummingbird: An open source dynamic simulation of flapping wing robots and animals," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 9223–9229, 2019.

[28] J. Ratti and G. Vachtsevanos, "Inventing a biologically inspired, energy efficient micro aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 437–455, 2012.

[29] G. Gremillion, P. Samuel, and J. S. Humbert, "Yaw feedback control of a bio-inspired flapping wing vehicle," in *Micro-and Nanotechnology Sensors, Systems, and Applications IV*, vol. 8373, p. 83731H, International Society for Optics and Photonics, 2012.

[30] M. Karasek, A. Hua, Y. Nan, M. Lalami, and A. Preumont, "Pitch and roll control mechanism for a hovering flapping wing mav," *International Journal of Micro Air Vehicles*, vol. 6, no. 4, pp. 253–264, 2014.

[31] I. Dragonfly, "BionicOpter," *FESTO_Projects*, 2013.

[32] D. Coleman, M. Benedict, V. Hrishikeshavan, and I. Chopra, "Design, development and flight-testing of a robotic hummingbird," in *71st annual forum of the American Helicopter Society*, American Helicopter Soc. Virginia Beach, VA, 2015.

[33] C. De Wagter, M. Karásek, and G. de Croon, "Quad-thopter: Tailless flapping wing robot with four pairs of wings," *International Journal of Micro Air Vehicles*, vol. 10, no. 3, pp. 244–253, 2018.

[34] G. De Croon, K. De Clercq, R. Ruijsink, B. Remes, and C. De Wagter, "Design, aerodynamics, and vision-based control of the delfly," *International Journal of Micro Air Vehicles*, vol. 1, no. 2, pp. 71–97, 2009.

[35] Q.-V. Nguyen and W. L. Chan, "Development and flight performance of a biologically-inspired tailless flapping-wing micro air vehicle with wing stroke plane modulation," *Bioinspiration & biomimetics*, vol. 14, no. 1, p. 016015, 2018.

[36] X. Chi, S. Wang, Y. Zhang, X. Wang, and Q. Guo, "A tailless butterfly-type ornithopter with low aspect ratio wings," 2018.

[37] S. B. Fuller, M. Karpelson, A. Censi, K. Y. Ma, and R. J. Wood, "Controlling free flight of a robotic fly using an onboard vision sensor inspired by insect ocelli," *Journal of the Royal Society Interface*, vol. 11, no. 97, 2014.

[38] L. F. Tammero and M. H. Dickinson, "The influence of visual landscape on the free flight behavior of the fruit fly drosophila melanogaster," *Journal of Experimental Biology*, vol. 205, no. 3, pp. 327–343, 2002.

[39] M. V. Srinivasan, "Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics," *Physiological Reviews*, vol. 91, no. 2, pp. 413–460, 2011.

[40] K. McGuire, G. De Croon, C. De Wagter, K. Tuyls, and H. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1070–1076, 2017.

[41] R. J. Moore, S. Thurrowgood, D. Bland, D. Soccol, and M. V. Srinivasan, "A fast and adaptive method for estimating uav attitude from the visual horizon," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4935–4940, IEEE, 2011.

[42] B. D. Lucas and T. Kanade, "Iterative Image Registration Technique With an Application To Stereo Vision.," vol. 2, pp. 674–679, 1981.

[43] B. Horn and K. Berthold, "Schunck. determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[44] S. Viollet, S. Godiot, R. Leitel, W. Buss, P. Breugnon, M. Menouni, R. Juston, F. Expert, F. Colonnier, G. l'Eplattenier, *et al.*, "Hardware architecture and cutting-edge assembly process of a tiny curved compound eye," *Sensors*, vol. 14, no. 11, pp. 21702–21721, 2014.

[45] Avago, "ADNS-9800 Data sheet," 2008.

[46] S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, and R. Beard, "Obstacle and terrain avoidance for miniature aerial vehicles," in *Advances in Unmanned Aerial Vehicles*, pp. 213–244, Springer, 2007.

[47] M. Engineering, "Optic Flow characteristic of a optical mouse sensor," no. group 130, 2014.

[48] A. Alvarez-Aguirre, G. Mok, S. H. HosseinNia, and J. Spronck, "Performance improvement of optical mouse sensors: Application in a precision planar stage," *2016 International Conference on Manipulation, Automation and Robotics at Small Scales, MARSS 2016*, 2016.

[49] C. Liu, Y. Xu, J.-G. Liu, H. Sun, and R. Kennel, "Rotational speed measurement based on avago adns-9800 laser mouse sensor," in *PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, pp. 1–5, VDE, 2016.

[50] F. Wendler, N. Pagar, and O. Kanoun, "Evaluation of ccd-based human device interface sensor for the velocity measurement in floodwater," in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 254–256, IEEE, 2019.

[51] H. v. Helmholtz *et al.*, "Physiological optics," *Optical Society of America*, vol. 3, p. 318, 1925.