

TALLINN UNIVERSITY OF TECHNOLOGY

Faculty of Information Technology

Department of Computer Engineering

IAY70LT

Evgenia Udod 120042IASMM

LIDAR-BASED ENVIRONMENTAL PERCEPTION
SYSTEM FOR EXPERIMENTAL UNMANNED
AERIAL VEHICLE

Master Thesis

Thomas Hollstein, Ph.D., Professor

Uljana Reinsalu, Ph.D., Research Scientist

Karl Janson, M.Sc., Early Stage Researcher

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Evgenia Udod

Wednesday 25th May, 2016

Abstract

This thesis presents the design of an experimental autonomous unmanned aerial platform for implementing autonomous navigational capabilities. Most suitable environmental perception system for such platform is introduced, designed and built. A LIDAR-based environmental perception system uses one laser range-finder sensor and a motorized gimbal stabilization system to sense the environment in 3D, being undependable from the aerial vehicle's movements. Such system is capable of perceiving the environment in a range up to 40 meters, however longer-range (15 – 40 m) measurements may lead to inaccurate representation of an environment. With the point-cloud-like result from a LIDAR-based system, the XYZ coordinates of the target objects in the environment are calculated – based on the object's distance and the sensors location in the environment. With such dataset, it is possible to create a representation, using OctoMap model, based on octrees. Main issue of the LIDAR-based environmental perception system can be proper configuration: scanning granularity and scanning angle must be carefully considered, as the actual scanning-time is limited. It has been shown, that approach for such system is successful – results of the environmental scans are promising and further improvements may greatly add more flexibility to this system.

Annotatsioon

LIDAR-i-põhine ümbruskonna tajumise süsteem eksperimentaalsele mehitamata õhusõidukile

Käesoleva projekti eesmärgiks on ehitada autonoomne mehitama õhusõiduk, mis suudab ohutult navigeerida algpunktist lõpp-punkti. Autonoomse funktsionaalsuse eesmärgi saavutamiseks on kõigepealt uuritud erinevaid mehitamata õhusõidukite platvorme ja tööpõhimõtteid. Tulemusena ehitatakse eksperimentaalse õhusõiduki platvorm.

Käesoleva lõputöö eesmärgiks on uurida ümbruskonna tajumise süsteemide võimalusi eksperimentaalsele mehitamata õhusõidukile, mis aitab saavutada autonoomse funktsionaalsuse ja ohutu navigeerimise. Ümbruskonna tajumiseks on valitud avatud lähtekoodiga tõenäosuslik 3D-kaardistamise mudel OctoMap, mis põhineb Octree puustruktuuril.

Lõputöö eesmärgina on vaadeldud erinevaid lähenemisi, kuidas ümbruskonna tajumise süsteeme luuakse, milliseid mudeleid on realiseeritud ning millise põhimõttega süsteem sobiks käesolevasse töösse kõige enam. Antud süsteemiks on valitud LIDAR-i-põhine ümbruskonna tajumise süsteem, mis koosneb laser-kaugusmõõtja sensorist, motoriseeritud stabiliseerimise süsteemist ja juhtkontrollerist, mille abil on seda süsteemi võimalik juhtida. Selline süsteem on võimeline teostama mõõtmisi 3D-keskkonnas, võimaldades väljastada punktipilve tulemusi – sihtmärgi-objektide XYZ koordinaate antud keskkonnas. Sellist punktipilve võib kasutada ümbruskonna mudeli loomiseks ja lennu ajal taskistuste avastamiseks. Ümbruskonna mudelit on võimalik kasutada õhusõiduki ohutu teekonna planeerimiseks õhusõiduki autonoomsuse saavutamiseks.

On oluline õigesti valida ümbruskonna kaardistamise karakteristikud – kaardistamise detailsuse samm, mõõtmise kaugus ja mõõtmise ulatus, et süsteemi väljund oleks võimalikult täpne. Tulemused näitavad, et antud süsteem suudab edukalt ja täpselt ümbruskonda tajuda ning selle süsteemi väljundit võib kasutada edukalt ruumi kaardistamiseks. Süsteemi täiendamiseks on võimalik kasutada mitmeid lähenemisi, parendades süsteemi ümbruskonna kaardistamist kiiremaks ning süsteemi efektiivsemaks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 106 leheküljel, 6 peatükki, 58 joonist, 15 tabelit.

Table of Abbreviations and Terms

GPS	Global Positioning System
UAV	Unmanned Aerial Vehicle
MAV	Miniature Unmanned Aerial Vehicle
LIDAR	Light Detection and Ranging
IMU	Inertial Measurement Unit
CW	Clockwise
CCW	Counter Clockwise
BLDC	Brushless DC Electric Motor
SF	Slow-Fly
LIPO	Lithium Polymer
RPM	Revolutions Per Minute
UBEC	Universal Battery Elimination Circuit
ESC	Electronic Speed Controller
API	Application Programming Interface
UART	Universal Asynchronous Receiver/Transmitter
MAVLink	Micro Air Vehicle Link
GPLV3	GNU General Public License Version 3
RTOS	Real-Time Operating System
PID	Proportional-Integral-Derivative
RC	Radio Control
3D	Three-Dimensional
PCL	Point Cloud
IR	Infrared
US	Ultrasonic
2D	Two-Dimensional

ROS	Real-Time Operating System
ADC	Analog-To-Digital Converter
I2C	Inter-Integrated Circuit
PPM	Pulse Position Modulation
PPM-SUM	Sum Signal for Pulse Position Modulation
SPI	Serial Peripheral Interface
CAN	Controller Area Network
ARM	Advanced RISC Machine Architecture
MCU	Microcontroller Unit
I/O	Input/Output
PWM	Pulse-Width Modulation
RSSI	Received Signal Strength Indicator

Contents

1	Introduction	12
1.1	Problem Definition	13
1.2	Concept for the UAV	14
1.3	Thesis Organization	18
2	Configuration for Experimental UAV	20
2.1	Quad-rotor's Model	21
2.1.1	Quad-rotor's Dynamics	21
2.1.2	Steering	23
2.1.3	Frame	24
2.1.4	Motors and Propellers	27
2.2	Viability Estimate	28
2.3	Component Specifications	34
2.4	Avionic System	35
2.5	Setup Routines and Flight Results	38
2.5.1	Setup Routines	38
2.5.2	Flight Results	40
3	Environmental Mapping	45
4	Environmental Perception Approaches	49
4.1	Environmental Perception	49
4.2	Approaches	50
4.3	Possible Design	54
5	LIDAR-based Environmental Perception System	57
5.1	Best Solution for the Design	57
5.1.1	Different LIDAR Systems	57
5.1.2	Design Concept	59
5.2	Sensor Selection	60
5.2.1	OSLRF-01 Sensor	60
5.2.2	LIDAR-Lite v1 Sensor	62
5.2.3	Results	64
5.3	Gimbal System	65
5.3.1	Physical Design	66
5.3.2	Controller Board	68

5.4	Final Configuration	69
5.5	Environmental Scanning Concept	72
5.5.1	Scanning Granularity	72
5.5.2	Scanning Idea	77
5.5.3	Scanning Routines	78
5.6	Implementation	81
6	Conclusions and Future Work	83
6.1	Conclusion	83
6.2	Future Work	83
	References	84
	Appendix 1	91
	Appendix 2	92
	Appendix 3	94
	Appendix 4	98
	Appendix 5	100
	Appendix 6	106

List of Figures

1	<i>System Architecture Block Diagram.</i>	16
2	<i>Model of Quad-rotor and Perception System.</i>	21
3	<i>Model of Quad-rotor and Perception System.</i>	21
4	<i>H-typed Frame Configuration.</i>	22
5	<i>Quad-rotor's Axis Configuration.</i>	23
6	<i>Frame Design. First Design of Polystyrene Foam.</i>	24
7	<i>Frame Design. Second Design – 3D Printed Frame.</i>	25
8	<i>Frame Design. Second Design – 3D Printed Frame with Polystyrene Foam.</i>	25
9	<i>Frame Design. Third Design – Frame from Polystyrene Material.</i>	26
10	<i>Frame Design. Third Design – Frame from Polystyrene Material.</i>	26
11	<i>Frame Design. Final Frame.</i>	27
12	<i>Estimated results for final model with Turnigy 800 KV 12"x4.5".</i>	33
13	<i>Quad-rotor Model.</i>	34
14	<i>Avionic System. Avionic Software System Framework Block Diagram.</i>	36
15	<i>Avionic System. APM Configuration.</i>	38
16	<i>Flight results. Smooth flight. Altitude Changes.</i>	40
17	<i>Flight results. Rapid flight. Altitude Changes.</i>	41
18	<i>Flight results. Rapid flight. Roll and Pitch Axis Changes in Autotune Mode.</i>	41
19	<i>Flight Results. Guided Mode. Way-points and Real Trajectory.</i>	42
20	<i>Flight Results. Guided Mode. Battery Voltage Change.</i>	43
21	<i>Flight Results. Guided Mode. Current Consumption vs Flight Dynamics.</i>	43
22	<i>Flight Results. Guided Mode. Altitude Changes.</i>	44
23	<i>Example of Generated Point Cloud by a LIDAR System.</i>	45
24	<i>OctoMap Corridor visualization, resolution 10 cm [1].</i>	47
25	<i>OctoMap Corridor Visualization, Cube 20 cm [1].</i>	47
26	<i>Corridor Visualization, Cube 20 cm. Occupied; Free; Unknown Space [1].</i>	48
27	<i>Environmental Perception. Velodyne HDL-64E LIDAR Scan Example [2].</i>	56
28	<i>LIDAR Systems. Different Possible Commercial 2D Devices [3].</i>	57
29	<i>LIDAR Sensor. Test for 2D Dataset.</i>	58
30	<i>LIDAR-based Environmental Perception System's Concept.</i>	59
31	<i>OSLRF-01 LIDAR Sensor.</i>	61
32	<i>LIDAR-Lite v1 Sensor.</i>	62
33	<i>Design for Stabilizing System.</i>	67
34	<i>BaseCam SimpleBGC 32-bit Controller Board.</i>	68

35	<i>A LIDAR-based System's Configuration Block Diagram.</i>	69
36	<i>LIDAR-based Environmental Perception System's Model.</i>	70
37	<i>Real Model for LIDAR-based Environmental Perception System.</i>	70
38	<i>LIDAR Sensor. Projection of the Beam on the Surface.</i>	73
39	<i>Gimbal System. LIDAR's Sensor's Scanning Step – 1.5°, 1°, 0.5°.</i>	74
40	<i>Scanning Strategies. Angle and Distance (Top View).</i>	75
41	<i>Scanning Strategies. Angle and Distance (Side View).</i>	75
42	<i>Scanning Strategies. Optimal Scanning Distance.</i>	76
43	<i>Scanning Strategies. Full Scan – Plains.</i>	79
44	<i>Scanning Strategies. Wide Angle Full Scan – Circular Motion.</i>	79
45	<i>Scanning Strategies. Full Scan – Circular Motion.</i>	80
46	<i>Scanning Strategies. Real-Time Scan.</i>	80
47	<i>Scanning Strategies. Real-Time Scan [4].</i>	81
48	<i>Scanning Example. Testing a LIDAR-system in the Woods.</i>	82
49	<i>Scanning Example. Top View PCL of the Woods – Scanning Step 1°.</i>	82
50	<i>Full Flight Results. Guided Mode. Current Consumption vs Flight Dynamics.</i>	91
51	<i>Components for the Quad-rotor Model.</i>	92
52	<i>Estimate on Suitable Rotor Configuration [5].</i>	93
53	<i>OSLRF-01 Sensor: Outgoing Pulse and Reflected Signals from White Surface.</i>	100
54	<i>OSLRF-01 Sensor: Outgoing Pulse and Reflected Signals from Black Surface.</i>	101
55	<i>Real Model for LIDAR-based Environmental Perception System.</i>	102
56	<i>Gimbal System. IMU Configuration for XYZ Coordinate System.</i>	103
57	<i>Gimbal System. Configuration Profile.</i>	104
58	<i>OctoMap Corridor Visualization, Cube 20 cm. Occupied Space [1].</i>	106

List of Tables

1	<i>Estimate on Model's Dimensions.</i>	29
2	<i>Different Quad-rotor Possible Setups.</i>	31
3	<i>Configuration of Real Model. Weight vs Size.</i>	36
4	<i>LIDAR Systems. Velodyne 3D LIDAR: Puck LITE™ [6].</i>	58
5	<i>LIDAR Sensors Differences. LIDAR-Lite v1, OSLRF-01-01.</i>	63
6	<i>a LIDAR-system Model's Final Properties.</i>	71
7	<i>Laser Beam's (1.5°) Projection for Certain Distance.</i>	73
8	<i>Model Components: Motor Specifications.</i>	94
9	<i>Model Components: ESC Specifications.</i>	94
10	<i>Model Components: Propeller Specifications.</i>	95
11	<i>Model Components: Battery Specifications.</i>	95
12	<i>Model Components: Telemetry Specifications.</i>	96
13	<i>Model Components: Radio Control (RC) Specifications.</i>	96
14	<i>Model Components: 3DR Pixhawk Flight Control System.</i>	97
15	<i>BaseCam SimpleBGC 32-bit Specifications.</i>	105

1. Introduction

STATEMENT

THE ANALYSIS AND CONSTRUCTION OF AN EXPERIMENTAL UNMANNED AERIAL VEHICLE IS COLLABORATION WORK AND SECTIONS 1, 2, 3 WITH THEIR SUBSECTIONS (PG 12 – 48) ARE SHARED BETWEEN CURRENT THESIS AND [7].

Unmanned aerial vehicles offer wide application usage – these systems are usually autonomous and operate without human-intervention. Such-small scale vehicles can perform tasks as search, rescue and disaster response, inspection of dangerous places, data collection for geographical information systems, mapping and observation of indoor or outdoor areas, environmental monitoring, security surveillance and inspection. Fixed-wing unmanned vehicles, suitable for searching large areas, are successfully used for agricultural inspection. Various payload-based applications are also suitable – for delivery of low-mass supplies.

Today, different approaches are used for designing such applications from standalone vehicles to aerial vehicle swarms. Systems can be designed on global positioning system (GPS) control, which simplifies the localization problems, however, accuracy of the sensor must be considered. Such vehicles cannot fly in cities near large buildings as GPS reception is reduced. To achieve better accuracy with GPS, another approach of redundant design with sensor-fusion can be used – from 2.5 m to centimeter-level accuracy can be achieved [8]. On the other hand, systems can also be designed for indoor use, utilizing other sensors and approaches to achieve localization within the environment.

The UAV is an aircraft which is equipped with necessary data processing units, sensors, automatic control and communication systems. The UAV is capable of performing an autonomous flight mission without the intervention of a human pilot [9]. UAVs are used in various applications, therefore the final size, configuration, shape and payload capabilities are different. For building such a system, following core-parts are used: aerial vehicle, avionic system for flight control, ground control system and radio control for human intervention.

1.1. Problem Definition

This thesis is an experimental project for creating integrated subsystems in order to the achieve construction of a small-scale unmanned aerial vehicle (UAV). Such small-scale vehicles are also known as miniature unmanned aerial vehicles (MAV).

THE MAIN TASK FOR THIS PROJECT is to design autonomous flight mission capabilities for an experimental small-scale aircraft. The aerial vehicle is able to perform autonomous navigation and environmental mapping operations in sparse unknown environments, where obstacles are mostly fixed, mainly non-dynamic and well-shaped. There must be possibility for the aerial vehicle to navigate through the environment. The main task for the UAV is to autonomously navigate to a desired destination avoiding well-shaped and fixed obstacles based on light detection and ranging-based (LIDAR) perception system's probabilistic representation. The vehicle is able to carry out such missions in GPS-available areas of sparse old-grown forest, with low density of forest-floor, below treetops.

THE MAIN TASK FOR CURRENT THESIS is to provide such a vehicle with suitable perception system which would fit to the aerial vehicle. Possible different approaches for designing such systems are introduced. Best design solution, a LIDAR-based environmental perception system is described, considering a limited budget and the UAV's lifting limitations for final construction of such system. An actual system is built and system's controlling implementation ideas are proposed. Advantages and possible problematic topics of the actual design are addressed with ideas and concepts for further improvement.

[7] investigates path-planning algorithms that can be used for the navigational operation in this project. A most suitable algorithm for final implementation is proposed considering characteristics of the aerial vehicle system.

The construction of such system is a challenging task – selection of overall design such as frame construction, weight balance, type of aircraft and selection of flight system must be carefully considered. Flight dynamics of such vehicles are hard to model. The main criterion for this project is to build a compact aircraft, able to fly in narrow areas that the vehicle can pass with sufficient payload width and lift power to carry a LIDAR-based perception system. flight time is not restricted at this point, it will depend of the type of the vehicle and its configuration.

Multi-rotors are highly maneuverable and thus especially suited for challenging indoor operations [10], [11], [12], [13] and outdoor [14], [15], [16], [17] tasks. Maneuverability allows such systems to traverse in narrow and complex areas. Such vehicles are able to carry a wide range of scientific payloads for deploying UAV's for the task of an autonomous exploration.

For the design, a four-rotor configuration is selected, as it allows higher maneuverability in the targeted environments compared to fixed-wing aircrafts. The four-rotor configuration is more suitable than traditional heli- or tri-copter as quad-rotor can generate more lift power for the desired payload. A configuration with more than four rotors would need more power to operate which adds additional weight to the system and increases the frame's size. Additional components will also result in a more costly vehicle.

1.2. Concept for the UAV

Main task for the project is to introduce autonomy for the aerial vehicle. For this purpose, concept of the on-board computer system is introduced. This system would handle the tasks that would result in analyzing the environment and creation of suitable paths for the UAV to follow. Actual flight routines are handled within the quad-rotor system by the avionic subsystem.

Goal for the specific mission is to autonomously navigate from target way-point to a fixed destination traversing the route around the obstacles. It is assumed that sparse environment is traversable by the UAV, obstacles are well-shaped and mostly non-dynamic. If obstacles are discovered in the flight corridor, either static or dynamic, the UAV must react to this event and re-evaluate the path. The GPS-reception is assumed available.

To simplify the concept, it is assumed that the UAV flies always front side forward so the perception system can be locked on the front direction for simplifying the concept for scanning routines.

Map for the environment may be available beforehand. When mission engages, firstly, the UAV performs the scan in the direction of the goal. Scan is also performed when the map is available – so the data about the environment would be up-to-date. After the scan, path calculation is executed and suitable path transmitted to the avionic system for engaging the

flight routine. While flight routine is executing, the perceptual system scans for possible obstacles that may occur in the flight-corridor. If obstacle is located in the flight-corridor that is assumed to have been empty, the navigational system would get the trigger of possible threat. UAV system must stop the navigation and re-evaluate the path.

The reasonable scanning distance must be set beforehand, at what range the system will flag the possible threat when UAV is navigating. This greatly depends on a vehicle's speed, navigational route and the scanning routine's implementation. Possible approach for effective and fast response must be considered.

If the environmental representation is not complete, even after the first scan, the path calculation will find the path for the nearest point to the goal, where the scanning execution may again take place in the direction of the destination.

Aviation system is responsible for the navigational part. If the goal is not achieved and the power supply becomes exhausted, the UAV will engage the safe mode and land. If the environment is too dense that the UAV will not succeed to navigate to the goal, the UAV will land. In all cases, the UAV should label the status of the mission and when landed, the coordinates of the vehicle should be fixed and transmitted to the ground control application.

MAIN ARCHITECTURE BLOCKS

Figure 1 shows the four main architecture blocks of the UAV's flight control system. The system is designed to the following four separate: quad-rotor system, environmental perception system, on-board computer system and ground station control application.

The quad-rotor system includes the physical and software design for the vehicle – avionic system within the physical UAV handles the flight control routines. Perception system's idea is to collect data from the environment to create environmental representation. On-board computer system handles the autonomy for the UAV, it drives main task blocks – the environmental perception task, path calculation task and navigation task. If all tasks finish with successful result – creation of the flight plan based on the environment, it is possible to transmit data to avionic system that would handle the actual flight routines.

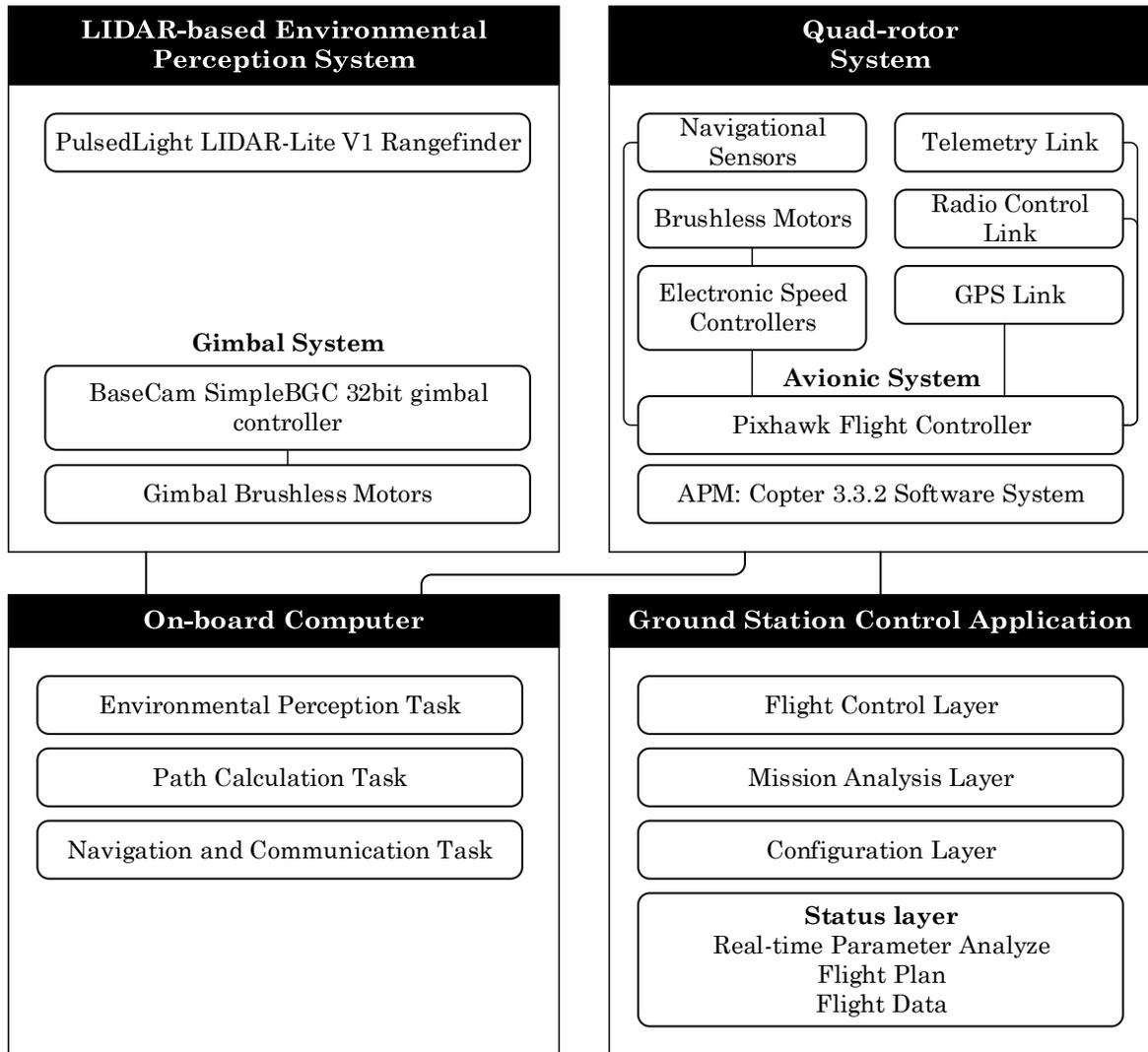


Figure 1. System Architecture Block Diagram.

QUAD-ROTOR SYSTEM

Quad-rotor system consists of physical components for aerial vehicle, (i) navigational sensors including an inertial measurement unit (IMU), (ii) global positioning system (GPS), (iii) magnetometer, (iv) barometer and (v) an avionic system. Main functionality of an avionic software [9] system is: (i) collection and analysis of in-flight information, (ii) execution automatic control laws: flight control algorithm execution; actuator control driving, (iii) communication with ground control station, and (iv) logging necessary in-flight data.

LIDAR-BASED ENVIRONMENTAL PERCEPTION SYSTEM

For environmental sensing, LIDAR-based system will be used (described in Section 4), which consists of a gimbal system and laser range-finder, device that measures distance from the observer to a target. Gimbal is used for stabilizing and directing sensor's beam to desired direction. Laser range-finder module works in various environmental conditions such as sunlight or light rain which has advantage over ultrasonic or infrared sensors as range-finder provides good measurement accuracy and longer distance sensing.

ON-BOARD COMPUTER SYSTEM

Main routines hosted on the on-board computer are (i) environmental perception task: scanning task to create map of environment and real-time safety scanning task while navigating, (ii) path calculation, and (iii) navigational task for quad-rotor.

Final selection of the on-board computer system depends on the final software design and how much processing power each task needs to finish processing in reasonable time for the aerial vehicle to operate. Time is restricted by the power supply, used on the UAV. Estimated time for UAV's is considered to be approximately 10 – 15 minutes. With usage of new power source, the time-restriction may be extended.

Perception task handles the scanning routines and representation of the environment. To represent the environment, OctoMap mapping model, based on octree structures will be used (described in Section 3), implementation is considered as a future work. To create a world-map of possible obstacles and free space, OctoMap tools will be used, input for the map is used from LIDAR-based system. Based on the map it is possible to navigate the environment and plan for best suitable trajectories for UAV to follow based on most suitable path-planning algorithm.

Environmental representation is used for executing path calculation. Most suitable path-planning algorithm will be implemented as the future work on OctoMap tree-structure to find most suitable and energy efficient trajectories for the quad-rotor. Path-planning algorithm, proposed specially for this project for is introduced in [7]. For finding the best possible path from source to destination, a cost based the energy model should be evaluated. Path-planning algorithm should take into consideration parameters such as (i) security measures

(width of the corridor of possible path), (ii) vehicle speed, (iii) path-length and (iv) time limitation for our model. Goal is to find an algorithm that provide most effective path for current configuration of the UAV. Usually, the most effective path is the shortest, depending on the speed of the vehicle. Path calculation task may return the GPS-based way-points as a possible route for navigational task.

Implementation of the navigation task will remain as future work. The idea of the task is to handle the navigational operations – destination point can be set for UAV and flight-plan according to path calculation task can be executed. Communications handling must be considered for the end-user to be able to see the status of the mission. Usage of additional radio systems may be considered for implementation of user-interface operations such as debug and navigational-link.

GROUND STATION CONTROL APPLICATION

Ground station control application implements communication between avionic system and ground station control system via wireless link. This system is preferable but not compulsory, purpose is to see in-flight data in real-time via an user interface. The application can be categorized as follows: (i) flight control layer, (ii) mission analysis layer, (iii) configuration layer and (iv) status layer. Flight control layer is responsible for sending desired executable flight control messages from ground station application to avionic system. Mission analysis layer handles generation and maintenance of desired flight trajectories and logging mission data. Configuration layer allows to read and edit UAV's flight and configuration parameters. Status layer allows the user to monitor in-flight data in real-time.

1.3. Thesis Organization

Main tasks are defined in Section 1. The focus for this project is on investigation, design and creation of an experimental autonomous unmanned aerial platform for implementing navigational capabilities to the UAV. For this particular thesis, the focus remains on the perceptual system which serves as a foundation for implementation navigational capabilities to the UAV. Main problems are addressed and the concept for the UAV is introduced.

Section 2 introduces the configuration for the UAV. Quad-rotor's model and flight dynamics

are described. Estimation on several models is carried out for construction of an effective final platform. The real model is built and flight tests are successfully carried out.

Section 3 investigates the environmental modeling possibilities that can be used in this project. Different models are investigated and final model for possible mapping operations is selected.

Section 4 explores the possibilities for achieving an environmental perception and possible solutions that can be used for this project. Related works and different approaches are introduced. Most suitable approach, a LIDAR-based environmental perception system is selected.

Section 5 presents real design for a LIDAR-based system in detail. Different systems are compared and final configuration is introduced. System is built and simple implementation for controlling the device is enclosed.

Final conclusions, discussion of the problems and possible improvements are presented in Section 6

2. Configuration for Experimental UAV

For the model to be effective – frame construction, weight balance, type of aircraft and selection of flight system must be carefully considered. Section 2.1 introduces quad-rotor's dynamics for selecting components like motors, propellers and frame.

Selection of actual avionic system is important (described in Section 2.4), as it includes flight controller with avionic software system, which is responsible for executing automatic control laws: flight control algorithm execution, actuator control driving, communicating with ground control station and logging necessary in-flight data.

Estimated models with possible suitable components are introduced (described in Section 2.2). To make rough evaluation on the models, viability calculation is made with ECalc [5] tool. It is essential to evaluate that all components, especially that rotors and propellers would suit a specific frame. This way would quad-rotor system result in working vehicle, being able to lift the frame with estimated payload. Section 2.2 introduces most suitable configurations for building the experimental quad-rotor system. Section 2.3 introduces the construction for the final quad-rotor.

Figure 51 [18], included to Appendix, introduces in detail generic components for the quad-rotor model. Flight results for a real model are introduced in Section 2.5.

Figure 2 and Figure 3 proposes the final model for the system, although, final design for suitable landing gear must be considered. Landing gear is used for safe take-off and landing, allowing to support the vehicle and the payload without damage. As selected flight controller has auxiliary outputs (described in Section 2.4), it is possible to implement motorized landing gear to a working system.

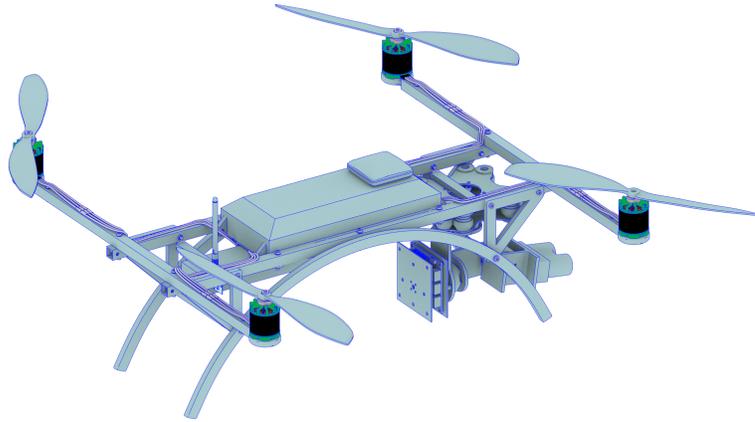


Figure 2. *Model of Quad-rotor and Perception System.*

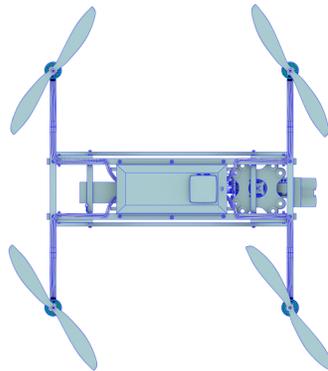


Figure 3. *Model of Quad-rotor and Perception System.*

2.1. Quad-rotor's Model

2.1.1. Quad-rotor's Dynamics

For the UAV model, a quad-rotor-typed air vehicle is constructed. Four rotors generate upwards lift, independent control of relative thrust to each rotor results in desired movement of the model. With a change of a speed of each rotor, possible desired turning force is achieved.

Rotors are aligned in shape of rectangle, two rotors turn in clockwise (CW) direction and the other two rotate in the opposite direction (CCW), as shown on Figure 4. The aerodynamic torque of the first rotors pair cancel out the torque created by the second pair which rotates in the opposite direction. This rotation configuration neutralizes rotors' tendency to make quad-rotor rotate so if all four rotors apply equal thrust, the quad-rotor will maintain it's direction.

Quad-rotor has four controllable degrees of freedom: altitude, yaw, pitch and roll. Each degree of freedom is controlled by changing speed of rotors. To maintain the overall balance and desired position, sophisticated control system must be used. Pixhawk avionic flight control system (described in Section 2.4) is used as a quad-rotor's control system.

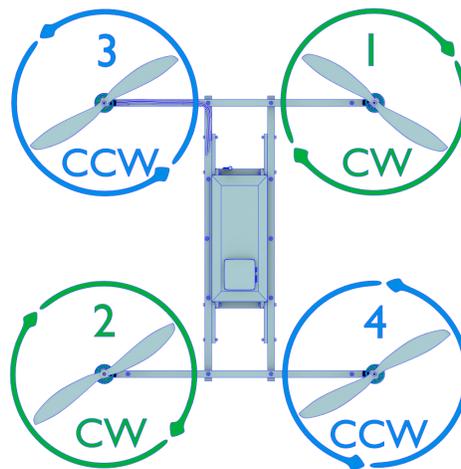


Figure 4. *H-typed Frame Configuration.*

To control position in any degrees of freedom, speed of rotors are changed. Different axis are shown on Figure 5.

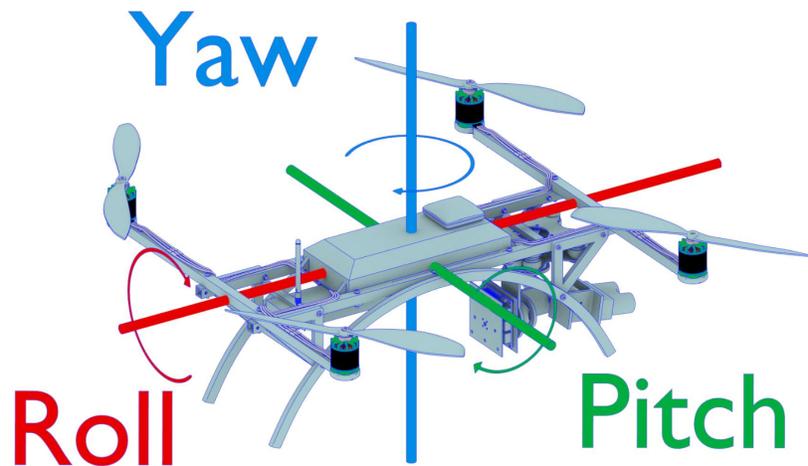


Figure 5. *Quad-rotor's Axis Configuration.*

2.1.2. Steering

For controlling direction in yaw axis, the control system slows down opposite pairs of rotors relative to the other pair. Rotation takes place as angular momentum of the two pairs of propellers will not be in balance. Quad-rotor can be rotated in both direction by changing rotor speed in different pair of rotors.

For controlling direction to roll and pitch axis, control system must change speed of two rotors on the side. One side of the model would have more thrust than other side, causing quad-rotor to tilt.

To gain altitude, lift force produced by the rotors must be increased – speed of rotation of all motors must be increased, resulting vehicle to gain altitude. Opposite operation must be engaged when vehicle should reduce the altitude – with decreasing the speed of the motors, lift force produced by the rotors is reduced, which results vehicle to decrease the altitude.

Axis-tilt position changes causes quad-rotor to move. With tilting, quad-rotor can move to different directions as lift force produced by the rotor is not directed downward resulting in pushing the quad-rotor. For this movement it is important for the control system to be able to maintain the altitude since less rotor thrust power is directed downward while tilting.

2.1.3. Frame

Frame is the structure that holds the quad-rotor and its components together. Frame has to be as light as possible. Most available materials are carbon fiber, aluminum or polyurethane foam. For the final model, hollow aluminum square rods are used for their relatively light weight, rigidity and affordability. However as damping effect for aluminum is not good, to reduce some vibrations to the frame, plastic motor attachments are used.

Frame consists of three-dimensional (3D) printed parts: bottom, top, motor attachments and aluminum rods. Frame is light enough for our settings. Design is expandable if needed, easy to repair and as light as possible.

For the quad-rotor model many frames were considered and tested. Figure 6 introduces the first design of the frame, polystyrene foam was used with square-size of 500 x 500 x 50 mm. 4 round placeholders for rotors were cut out, each with diameter of 180 mm. Two aluminum profile rods were attached for motor attachments. This frame was too soft and small for possible desired configuration.



Figure 6. *Frame Design. First Design of Polystyrene Foam.*

Figure 7 and Figure 8 introduce another design. Frames were constructed with 3D printed

parts. Neither frame did suit because of the high weight and elasticity, severe blow would harm the frame as the construction and materials are not strong enough.



Figure 7. *Frame Design. Second Design – 3D Printed Frame.*



Figure 8. *Frame Design. Second Design – 3D Printed Frame with Polystyrene Foam.*

To increase the rigidity of frame and to preserve the protective motor construction, design from polystyrene material with aluminum sub-frame was introduced. Unfortunately the weight of material was not suitable as the frame was too heavy for the final construction. Strong wind drag was noticed. The frame was relatively big, compared with previous ones as the safety rigs were thicker.



Figure 9. *Frame Design. Third Design – Frame from Polystyrene Material.*

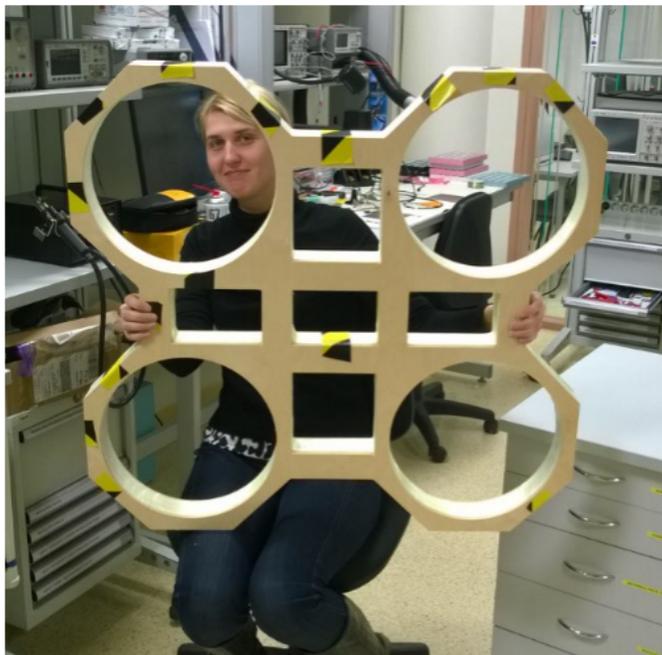


Figure 10. *Frame Design. Third Design – Frame from Polystyrene Material.*

Final design of the frame consisted of few printed parts and aluminum profiles as weight of the frame is important. This design is used for outside-flight only and suits the requirements

such as: being light enough, easy to repair and having simple design.



Figure 11. *Frame Design. Final Frame.*

2.1.4. Motors and Propellers

For heavy lift and for a slow and stable ride brushless direct current electric (BLDC) motors are used along with electronic speed controllers (ESCs) for each motor. Each motor is controlled separately by a speed controller via avionic system. Selection of the motor and propeller needs to match the overall model. Main characteristics considered for motor selection is KV-rating, which indicates the revolutions per minute for number of volts. For motor selection the final weight of the model must be estimated to create the required thrust to lift the quad-rotor.

General rule is to provide two times the thrust than weight of the quad-rotor. For larger quad-rotors that carry payloads, low KV-rated motors work better as they have more rotational momentum and maintain quad-rotor's stability.

Propellers are selected based on motor's characteristics. For selection of propellers, the length and pitch are important. Pitch is a parameter for traveled distance during a single propeller rotation. Higher pitch means slower rotation, but increase of the quad-rotor's speed with usage of more power. Lightness of overall model is very important as excessive weight reduces battery life and maneuverability. To have an optimal flight-time, it is important to

find balance between final selection of motors and propellers.

By increasing the propeller's size – more thrust can be generated, therefore more lift-force for quad-rotor is achieved. The bigger the propeller's diameter, efficiency at hovering will be increased, but control over the vehicle will be decreased. Although carbon fiber typed propellers are expensive in comparison to plastic ones, carbon fiber material is selected as it is significantly stronger, stiffer and more durable than plastic.

With the intention of having stable ride with heavy quad-rotor, low revolutions per minute (RPM) motors are selected. Based on motor selection, suitable carbon fiber slow-fly (SF) typed propellers are used. Slow-fly propellers [19], that usually generate higher amount of thrust in respect to regular propellers, have wide taper and broad flat blades, usually lower pitch. With correct propeller and motor selection, the quad-rotor will have enough power to lift the estimated payload.

For controlling motors, suitable ESCs are used. Selection of the components depends on a selection of motor and it's current consumption. Lithium-Polymer battery (LIPO) as power source is used. Configuration and type of a battery depends on motors – bigger capacity of battery results usually in longer flight-time, important aspect which also affects the flight-time, is the weight of the battery.

2.2. Viability Estimate

To find suitable components for the model, Ecalc [5] tool is used to make a estimate for the correct setup. Ecalc is used to simulate and evaluate electric motor driven systems for remote controlled models. This tool shows problematic areas when choosing components for the model. It is essential to evaluate that all components, especially that motors and propellers would suit a specific frame. With correct setup, the quad-rotor would be able to lift the frame with estimated payload, which would result in a working vehicle.

Larger the propeller for the frame is used, the larger the total disc area becomes, leading to more efficient quad-rotor hovering capability. However, larger propeller slows down effective control response. Final propeller selection is based on the final motor, for air vehicle dynamics are described in Section 2.1.4.

Table 1. *Estimate on Model's Dimensions.*

Parameter	Value	Unit
<i>General</i>		
Number of Rotors	4	-
Type of Rotor Configuration	flat	-
Frame Size	400	<i>mm</i>
Flight Controller Tilt Limit	No	-
<i>Estimate</i>		
Multi-rotor System	1200	<i>g</i>
Battery	350	<i>g</i>
Payload (LIDAR system)	300	<i>g</i>
Total Estimate Weight	1850	<i>g</i>

Many different combinations were considered for the same frame configuration. Best possible results based on estimated weight and frame type are shown in Table 2. For the motors, SunnySky A2212-980 (980 KV) and Turnigy Multistar 2216-800 (800 KV) were considered. Description for the Table 2 is included to Appendix, pg 98.

ESTIMATE ON MOTORS. Weight of the model and payload is essential (see Table 1). For the most optimal flight, based on our configuration – slow motors are proposed (estimated [est], KV, 770 - 1120). Figure 52 included to Appendix, proposes two-bladed 11" propellers, final selection depends on a specific motor. Approximate motor power is estimated between 275 - 480 watts, ESCs minimum 25 - 45 amperes should be used.

Usage of low-RPM motors were considered. Turnigy Multistar motor with 11"x4.7" and 12"x4.5" propellers gave the most promising estimation. Hover flight-time with usage of 5000 mAh 3-cell battery in all cases is around 11 minutes (est, min, 10"x4.5": 11.1; 11"x4.7": 11.4; 12"x4.5": 12.1). Enough lift power is generated in all cases, but with combination of SunnySky 980 KV rotors and 10"x4.5" propellers, estimated load to rotor is near to maximum (est, 154 W; max, 160 W), which might lead to possible motor failure.

ESTIMATE ON PROPELLERS. Two different slow-fly propellers can be used with Turnigy Multistar 800 KV motors for our model. With 11"x4.7" propellers, better estimated motor efficiency is achieved, but efficiency of both configurations are very similar. At total drive, when aerial vehicle with payload is operating in the air, estimated efficiency at hover is 11"x4.7": 75%, 12"x4.5": 73%. At total drive, estimated efficiency at maximum throttle is 11"x4.7": 72%, 12"x4.5": 68%. In both cases 11"x4.7" and 12"x4.5" propellers result in

similar efficiency.

THROTTLE INPUT. With longer propeller blades, better estimated throttle input at hover is achieved (est, %, 12"x4.5": 50; 11"x4.7": 57). With our specific configuration best throttle aim at hover estimate is less than 60%, the lower, the better. Below 50% throttle estimate is used for racer quad-rotors.

SPECIFIC THRUST. In both cases (est, g/W, 12"x4.5": 7.9; 11"x4.7": 7.5), specific thrust of propeller is efficient, results above 6 g/W considered as good efficiency. Specific thrust indicates grams of produced thrust with one watt of electric input on rotor. Thrust-weight ratio is better with configuration of 12"x4.5" (est, 12"x4.5": 1.9; 11"x4.7": 1.6) although this estimate is rough as for final model carbon fiber propellers are used. Calculator's estimation is based on APC manufacturer's SF-propellers, which have different *PConst* parameter value from final propeller. *PConst* represents how much power is absorbed by the propeller. Such parameter differs for brands, size and propeller's material and can affect the final flight performance. But as usually this information is not published by manufacturer, rough estimation is used in this case.

CURRENT CONSUMPTION. Estimated current per motor in both cases stays under 20 A, at hover, estimate is around 5 A (est, A, 12"x4.5": 5.6; 11"x4.7": 5.3). At full throttle current estimate per one motor is around 11 A (est, A, 12"x4.5": 10.2; 11"x4.7": 10.2). Based on estimation, 20 A ESCs can be used.

TEMPERATURE. With both setups, load on motor does not result in motor overheating. With maximum estimated load on a motor and good cooling presumed, temperature increase is minimal, around 11 degrees (est, °C, 12"x4.5": 11; 11"x4.7": 14). Temperatures of motor case over 80°C and higher might result in motor failure, estimated temperature changes stay far below maximum, which not be a limiting factor at summer time. Estimation is done for winter conditions – temperature starting from 0°C. It is not advisable to use LIPO batteries below 0°C.

RPM LIMITS FOR PROPELLERS. Both propellers fit within permissive RPM range. At hover, 11"x4.7" estimated RPM represent 58% of max RPM, 12"x4.5" estimated RPM represent 61% of max RPM on specific propeller. At full throttle RPM of propellers are estimated 83% versus 94%. 11"x4.7" propellers show better result with Turnigy Multistar 800 KV rotor as lower load on a propeller is preferred.

Table 2. Different Quad-rotor Possible Setups.

Parameter	Unit	SunnySky 10"x4.5"	Turnigy 11"x4.7"	Turnigy 12"x4.5"
<i>Battery LIPO 25/35C</i>				
Configuration		3S1P	3S1P	3S1P
Load	C	12.11	9.81	11.64
Total Capacity	mAh	5000	5000	5000
Minimum Flight Time	min	4.2	5.2	4.4
Mixed Flight Time	min	8.8	9.3	9.4
Hover Flight Time	min	11.1	11.4	12.1
<i>ESC</i>				
Current	A cont	20	20	20
	A max	25	25	25
<i>Motor at Maximum</i>				
		SunnySky (980 KV)	Turnigy (800 KV)	Turnigy (800 KV)
Current	A	15.14	12.27	14.55
Voltage	V	10.19	10.36	10.22
Estimated RPM	rpm	8239	6665	6253
RPM at Full Battery	rpm	12348	10080	10080
Maximum RPM for Propellers	rpm	10500	8000	6667
Maximum Motor Power	W	160	220	220
Electric Power	W	154.2	127.1	148.7
Mechanical Power	W	123.4	98.4	110.2
Efficiency	%	80	77.4	74.1
Estimated Temperature	degrees	13	11	14
<i>Motor at Hover</i>				
Propeller		Speed 10"x4.7"	SF 11"x4.7"	SF 12"x4.5"
Current	A	5.72	5.58	5.27
Voltage	V	10.76	10.76	10.78
Revolutions	rpm	5409	4615	4050
Throttle	%	51	57	50
Efficiency	%	80.1	76.7	74.3

Continued on next page

Table 2 – continued from previous page

Parameter	Unit	SunnySky 10"x4.5"	Turnigy 11"x4.7"	Turnigy 12"x4.5"
Specific Thrust	g/W	7.31	7.5	7.92
Estimated Temperature	degrees	5	5	5
<i>Total Drive</i>				
Model Estimate	g	1800	1800	1800
Estimated Maximum Thrust	g	3240	2880	3420
Estimated Thrust per Motor	g	810	720	855
Thrust-Weight	: 1	1.8	1.6	1.9
Current @ Hover	A	22.88	22.3	21.08
P(in) @ Hover	W	254	247.5	234
P(out) @ Hover	W	197.1	184.1	168.8
Thrust @ Hover	g	1857	1857	1853
Efficiency @ Hover	%	77.6	74.4	72.1
Current @ max	A	60.57	49.06	58.18
P(in) @ max	W	672.3	544.6	645.8
P(out) @ max	W	493.8	393.5	440.8
Efficiency @ max	%	73.5	72.3	68.3
<i>Multicopter</i>				
Additional Possible Payload	g	1008	710	1095
Maximum Tilt	degrees	50	44	52
Maximum Speed	km/h	39	28	29
Rate of Climb	m/s	5.4	3.6	4.1

FINAL RESULTS

Results for final model are based on Table 2. Both propellers generate enough thrust for the model with 800 KV rotor (estimated [est], g/W, 12"x4.5": 7.9; 11"x4.7": 7.5), which would result a working vehicle. With propellers of 11"x4.7", estimate for setup thrust-weight parameter is 1.6. With using 12"x4.5" propellers, estimate is 1.9. As propeller's P_{Const}

parameter value remains undocumented by manufacturer of possible propellers that are used, reliability of an estimation remains unknown.

Throttle input at hover is efficient in both cases, lower throttle is preferred (est, %, 12"x4.5": 50; 11"x4.7": 57). Current consumption per motor in both cases are similar, resulting in usage of same ESC of 20 A for motors. Temperature changes are similar if model has good cooling.

Based on estimation, combination of motors of 800 KV with 12"x4.7" propellers is preferred (see Figure 12). Although both setups are very similar, combination of selected motor and propellers result in the most efficient setup as thrust-weight ratio is very important metric for selecting of final components:

1. Total drive thrust has better estimate – maximum thrust of 3420 grams by 12"x4.5" propellers to be generated while 11"x4.7" would result in 2880 grams of thrust on model all-up-to 1800 grams (est, thrust, 12"x4.5": 1.9; 11"x4.7": 1.6).
2. Better throttle at hover is achieved – with longer propeller blades, better estimated throttle input at hover is achieved (est, %, 12"x4.5": 50; 11"x4.7": 57).
3. Longer hover time estimate – with longer propeller blades, longer estimated hover flight time is achieved (est, min, 12"x4.5": 12.1; 11"x4.7": 11.4).



Figure 12. *Estimated results for final model with Turnigy 800 KV 12"x4.5".*

2.3. Component Specifications

Real model is built on a estimated result of possible usable components. Estimated weight of 1850 grams resulted in a reduced model weight of 1732 grams, for specifications refer to Table 3. Real dimensions of the model are 390 x 365 mm. Figure 13 introduces the model: H-typed frame is used as deck-space is needed for mounting the payload. This design provides more available area for components than usual X-typed frame.

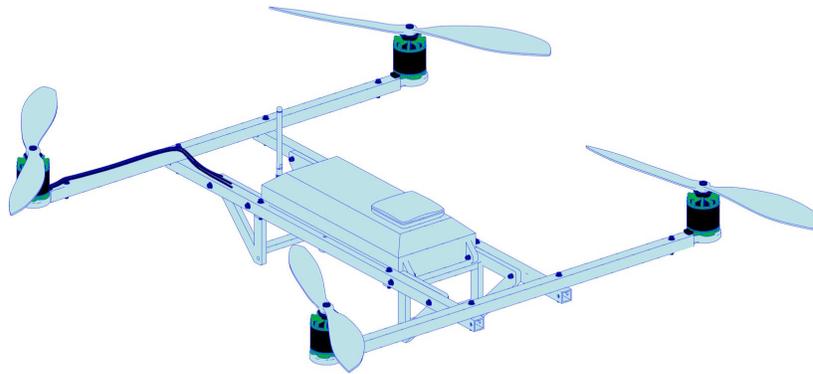


Figure 13. *Quad-rotor Model.*

MOTORS AND PROPELLERS. Four Turnigy Multistar 2216 800 KV Outrunner motors (see Table 8 included to the Appendix), are used with 12"x4.5" Carbon fiber SF propellers (see Table 10 included to Appendix). For testing purposes 11"x4.5" propellers are also used. Motors are controlled with Hobbywing 20A Skywalker Quattro uBEC 4-in-1 Brushless ESC (see Table 9 included to the Appendix). This controller is capable of handling continuous current consumption of 20 A for each motor, burst of 25 A, that is enough of source current the selected motors require. Model is compact and comfortable as wiring complexity is simplified.

POWER SUPPLY. Zippy Compact 5000 mAh 3S 25C LIPO is used (see Table 11 included to Appendix). Weight of 342 grams, this compact battery helps to reduce overall weight for a model. Battery's capacity is 5000 mAh resulting in maximum 12-minute hover flight time with payload on board.

AUTOPILOT SYSTEM. As an avionic system - 3DR Pixhawk flight controller is used with

APM: Copter 3.3.2 source configuration (see Table 14 and Section 2.4). 3DR uBlox GPS with Compass Kit is used for positioning.

RADIO SYSTEM. Three different radio modules are used for the model: telemetry link, manual radio control and radio system as a future work for debug and user interface information for navigational operations. Such radio systems should be considered in final stage autonomous navigation implementation for user-interface operations such as debug and navigational-link.

For telemetry link, SiK Telemetry Radio module v1 433 MHz is used (see Table 12 included to the Appendix). Telemetry application is used for primarily sending data from aircraft to ground station application [20]. It gives possibility to monitor vehicle's status while in operation. Basic and quick configurations of avionic software's parameters can be done via telemetry link. Updating, creating and loading autonomous missions to aircraft with simple point-and-click way-point entry on Google or other maps is possible via telemetry link and Mission Planner application software [21].

Manual control of the quad-rotor is achieved by using Taranis X9D Plus remote control unit [22] with X8R 16-channel Receiver, connected directly to flight controller's SBUS port (see Table 13 included to Appendix). As safety precaution, this radio system gives user full control over quad-rotor if it is necessary to take over autonomous control operations. Remote control also offers testing possibilities of flight dynamics while developing autonomous flight implementations. Adjustments to radio configurations are made, to ensure correct usage of radio devices according to general requirements for the use of radio transmission equipment in Estonia [23].

2.4. Avionic System

Avionic system consists of Pixhawk flight controller [24] and APM software [25] which coordinates all the hardware components and firmware on board in an appropriate sequence.

Figure 14 explains the framework of an avionic software system, where each task is shown as a block. Navigational data control task manages collection of sensor data. Motor control task generates appropriate motor control signal to drive the motors. Communication control ensures communication between avionic system and ground control system application. Data

Table 3. Configuration of Real Model. Weight vs Size.

Parameter	Value	Unit
<i>General</i>		
Number of Rotors	4	-
Type of Rotor Configuration	flat	-
Estimated Frame Size	60 x 400 x 400	mm
Real Frame Size	86 x 390 x 365	mm
Real Frame Size with Propellers	86 x 696 x 672	mm
<i>Estimate</i>		
Multi-rotor System	1200	g
Battery	350	g
Payload (LIDAR system)	300	g
Total Estimate Weight	1850	g
<i>Real Model</i>		
Multi-rotor System	1081	g
Battery	342.4	g
Payload (LIDAR system)	308	g
Total Real Weight	1732	g

logging provides log of in-flight data. Flight control implements the automatic flight control laws. Main control block manages all tasks.

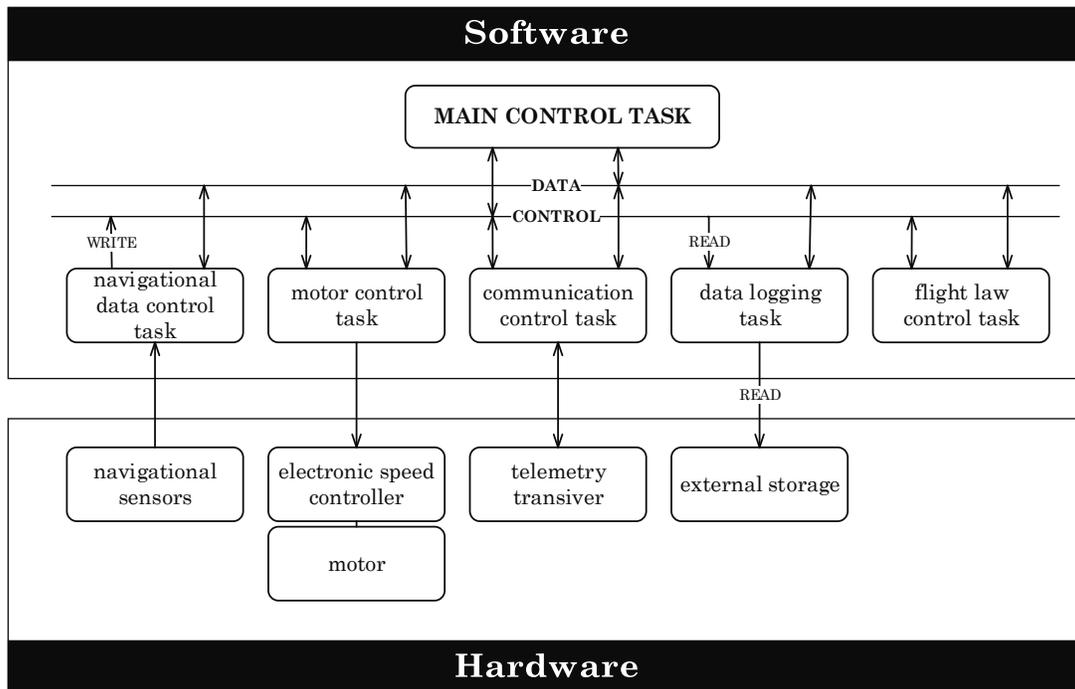


Figure 14. Avionic System. Avionic Software System Framework Block Diagram.

Important requirement for avionic system selection for research and development is an open software and hardware platform, widely active community and up-to-date software development. Pixhawk flight controller is mainly used for high-end research and amateur usage, can be considered as new but mature platform. Open-source and -hardware design allows researchers to adapt every detail as needed. More commercially open-source systems either only allow users to modify a part of the software or to replace the complete software stack [11], which requires researchers to completely rebuild the whole software stack. As Pixhawk is open system, it is a great testing and research environment for the UAVs.

Pixhawk project is a further evolution of the PX4 flight controller system [26]. Pixhawk is a single board controller having powerful 32-bit processor with an additional failsafe backup controller (see Table 14 to the Appendix). Board is equipped with I/O interfaces and advanced sensor profile: (i) 3 axis 16-bit gyroscope for determining orientation, (ii) 3 axis 14-bit accelerometer and compass for determining outside influences and compass heading, (iii) external sensor kit: magnetometer and GPS unit, (iv) barometric pressure sensor for determining altitude, (v) voltage and current sensing for battery condition determination.

System is optimized to provide control APM flight navigation software with high performance and capacity. For vehicle control, APM open-source flight-stack is used, licensed under GNU General Public License v3 (GPLv3) [27]. It is actively developed and has a large community. Flight software runs on NuttX real-time operating system, which features high performance, flexibility and reliability for controlling any autonomous vehicle [28].

Basic structure [29] of the flight software stack and the configuration is shown on Figure 15. APM flight-stack is responsible for state estimation and flight control. "PX4Firmware" is base middle-ware and driver layer for Pixhawk board, licenced under BSD [30]. APM flight-stack interfaces through "Hardware Abstraction Layer", which makes APM software portable for Pixhawk board. "Libraries" block represents structure of essential libraries such as core, sensor and other libraries for APM flight-stack [25].

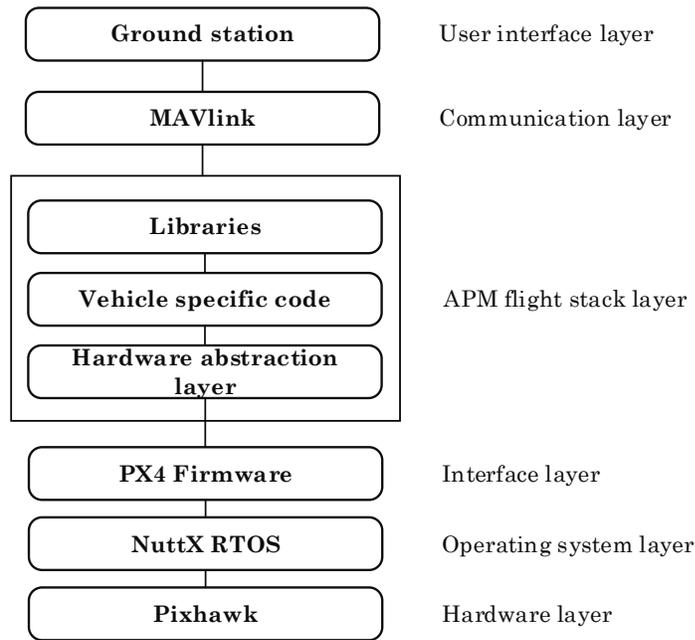


Figure 15. *Avionic System. APM Configuration.*

2.5. Setup Routines and Flight Results

2.5.1. Setup Routines

Setup for a quad-rotor includes proper assembly of the system as well as installation of the ground control application. Such software system is used as the configuration utility and the dynamic control application for an autonomous vehicle. Proper setup and configuration of quad-rotor is done after full assembly, which is included in [31].

The ground control application [32] allows to (i) load the firmware into the flight controller; (ii) configure the vehicle for an optimum performance; (iii) create, save and load autonomous missions to the flight controller; (iv) download and analyze mission logs created by the flight controller's firmware; (v) use the telemetry link to monitor vehicle's status while in operation and (vi) record, view and analyze telemetry logs.

After the UAV assembly, firmware must be loaded to the flight controller and mandatory initial setup must be completed to achieve best performance for the vehicle's operation. Initial

setup requires setting up the frame configuration for mapping the motors for the software. Proper flight modes are configured supporting different types of flight capabilities, mostly used modes are [33] "Stabilize", "Altitude Hold", "Loiter", "Return-to-Launch", "Autotune", "Follow-Me", "Guided Mode" etc.

For advanced configuration, the autotune functionality helps to automatically configure control loop feedback mechanism. Proportional–Integral–Derivative controller (PID) gains – these parameters are used in the stabilization-algorithm for the flight controller that provides vehicle’s highest control-response without significant overshoot. Autotune functionality can be triggered manually while vehicle is operating in the air.

Failsafe mechanisms [34] are set up for certain events, that are triggered for failure of the devices or software (radio, battery, ground control application) or not acceptable flight-behavior (flying out of the permitted geo-fence area). The flight software supports return-to-launch or landing functionality in cases where contact between the RC transmitter and the flight controller’s receiver is lost. Failsafe for battery can be set up to trigger return-to-launch or landing functionality when battery voltage has crossed below configurable threshold. Geo-fence failsafe ensures that vehicle will remain in desired area, if manually flying too far away from the allowed area, failsafe will be triggered forcing the vehicle to return-to-launch or land.

Pre-arm safety check is enabled for safety routines. These checks will prevent vehicle from arming if any problems are discovered – including missed calibration, configuration or a bad sensor. Pre-arm safety includes control of the device failures or miscalibrations of devices such as RC, magnetometer, GPS, accelerometer or gyroscope [35].

Calibration of the sensors including (i) magnetometer; (ii) RC device; (iii) accelerometer; (iv) ESCs, must be done before the first flight. ESCs are responsible for spinning the motors at the speed requested by the flight controller. ESCs need to be calibrated so the minimum and maximum control values from the flight controller will be recorded [36]. Before each flight, it is recommended to perform the calibration of the magnetometer, other devices should be calibrated optionally.

2.5.2. Flight Results

SMOOTH FLIGHT.

With the configuration of 1732 grams, quad-rotor is capable of smooth flight with payload for around 11.5 minutes, as seen from Figure 16. The flight was conducted in the strong winds using "Altitude Hold" mode, holding the desired height of 1.2 meters, using barometer sensor readings. Readings may vary in the different weather conditions, altitude change was between 1.0 - 1.5 meters, with peaks of 2; 0.8 meters.

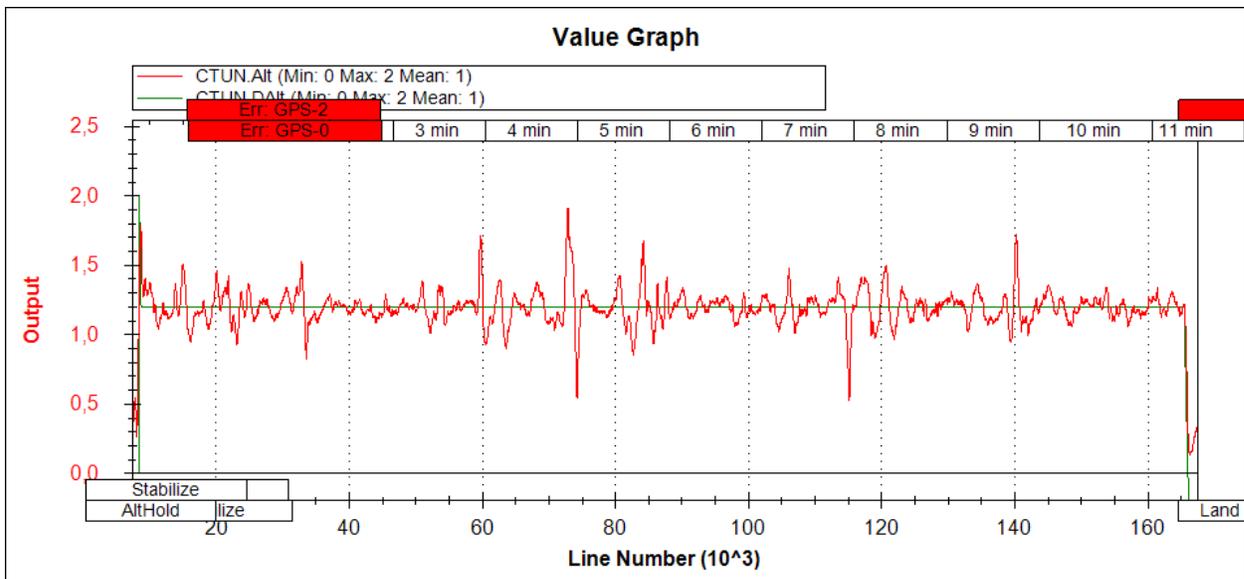


Figure 16. *Flight results. Smooth flight. Altitude Changes.*

RAPID FLIGHT.

Figure 17 and Figure 18 shows that the flight time was 9.5 minutes. Flight was engaged in moderate¹ weather conditions in "Autotune" mode, which is used for the vehicle's advanced configuration. "Autotune" should be engaged in light air² or light breeze³, without wind gusts. Figure 17 shows altitude of the vehicle. Although the barometer sensor's readings (blue line) are sensitive due to moderate weather conditions, the quad-rotor holds its desired altitude. Desired altitude was set to 3.5 meters (red line) resulting in minimal changes of calculated

¹Moderate breeze – 6-7 m/s. Wind gusts – 7-9 m/s.

²Light air – 0-2 m/s

³Light breeze – 2-3 m/s

GUIDED MODE.

The flight was conducted in the "Guided Mode" to invoke the automatic mission execution. The mission consisted of the ten earlier pre-loaded independent way-points, planned to hold the altitude of 7 meters with the 5-second delays between every way-point. The execution time of the flight-mission was 3 minutes. Figure 19 represents the way-point data and the real passed trajectory by the quad-rotor. Way-point precision was set to 1 meter in the mission configuration, so all way-points were taken accurately.

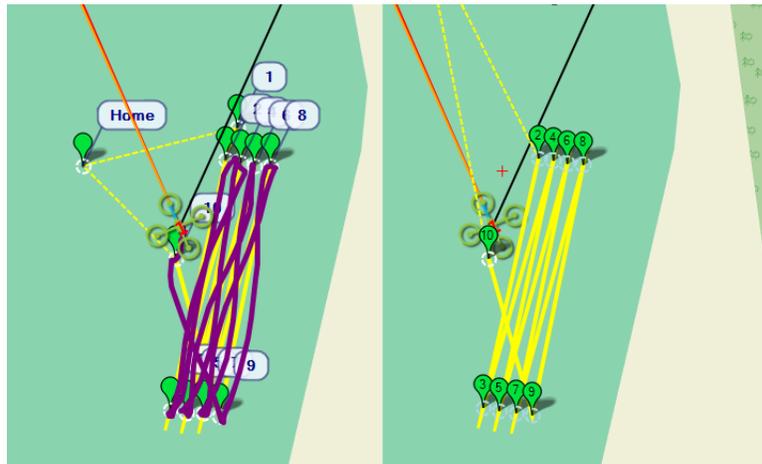


Figure 19. *Flight Results. Guided Mode. Way-points and Real Trajectory.*

The setup for the maximum flight-speed between the way-points is set to 500 cm/s (5 m/s, 18 km/h) as an internal parameter. Such flight-speed is successfully carried out by the quad-rotor (see Figures 20, 22 and 21). Figure 20 shows also the battery voltage drop. During three minutes of the mission execution, battery voltage dropped from 12.4 V to 10.9 V. Red line represents battery voltage, green line shows horizontal speed of quad-rotor.

During the flight, current consumption for the whole system was between 22 A – 25 A. After reaching the certain way-point, 5-second delay was invoked. Figure 21 contains a small segment of Figure 50 – the speed remains around 0 m/s during a 5-second delay (orange line). After the delay, the quad-copter reaches target-speed of 500 cm/s (5 m/s, 18 km/h) with a constant acceleration (purple line). During the acceleration, slight current consumption raise can be noticed (see Figure 21 and Figure 50 included to Appendix). Yellow bar represents an area, where the constant acceleration takes place. During this period, current consumption rises around 2.5 A, such trend takes place upon every acceleration of the UAV.

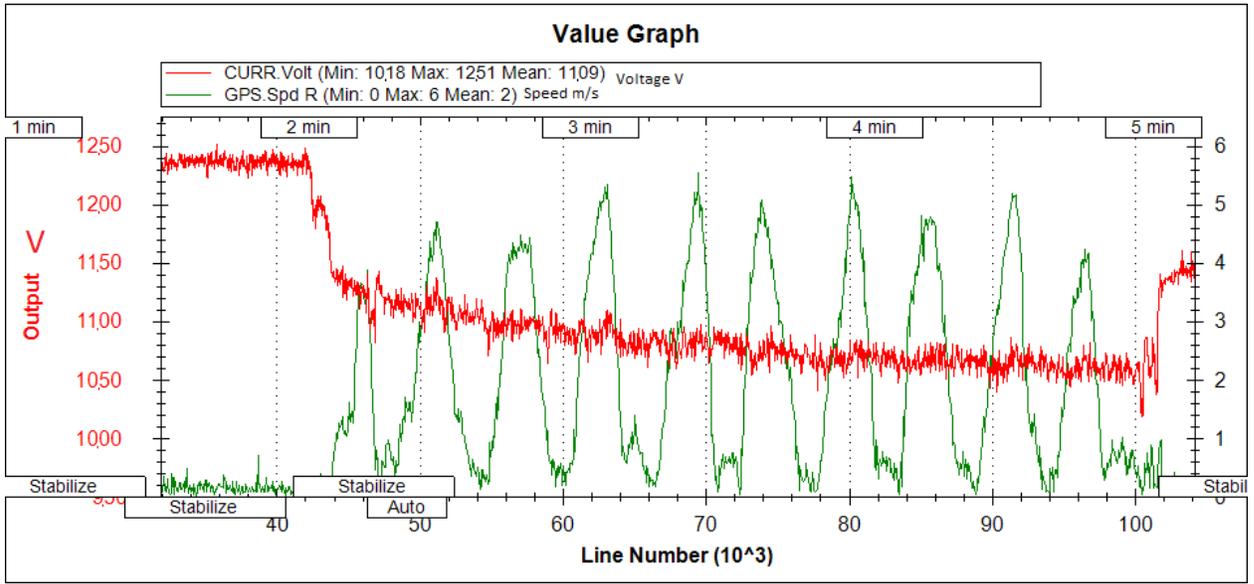


Figure 20. *Flight Results. Guided Mode. Battery Voltage Change.*

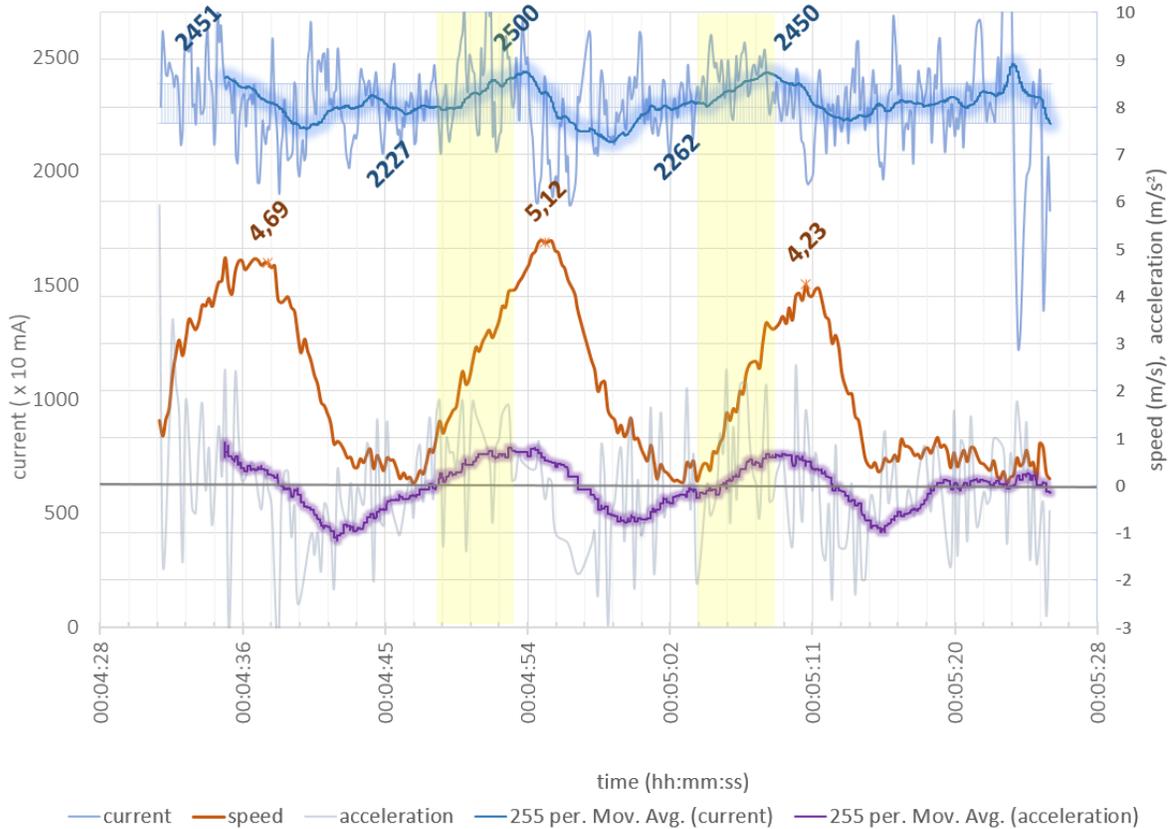


Figure 21. *Flight Results. Guided Mode. Current Consumption vs Flight Dynamics.*

Figure 22 shows the altitude. Green line represents desired altitude that was set beforehand. Blue line shows barometer's sensor-readings, which may vary in weather conditions, red line represents real altitude of the vehicle. The real altitude is calculated fusing barometric sensor's data and GPS readings. Real altitude differs slightly from the desired altitude. Changes in altitude can occur due to the constant speed changes (purple line).

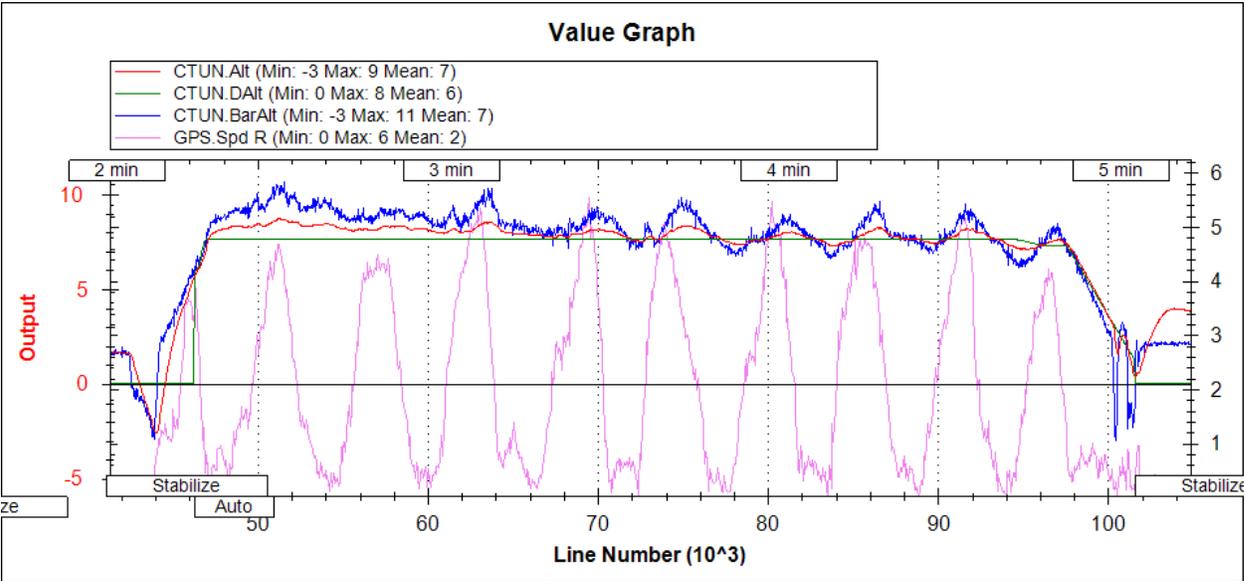


Figure 22. *Flight Results. Guided Mode. Altitude Changes.*

3. Environmental Mapping

For an autonomous behavior, the mapping and localization of the UAV in the 3D space is an important component. With a LIDAR-based approach, the UAV senses the environment by taking 3D-range measurements (see Figure 23). Storing the raw measurements is not reasonable, therefore suitable modeling framework to represent the environment needs to be considered. Localization in the environment is an important issue, which should not be underestimated. In this project, GPS-signal is expected for localization implementation.

Based on a specific modeling framework, the map of the space is created. Such map is considered the central component for an autonomous operations as it will be used for path-planning and navigational operations. Such map has to be effective and efficient in respect to access and size, so the large outdoor environments could be mapped.



Figure 23. *Example of Generated Point Cloud by a LIDAR System.*

For our configuration, following requirements are set: we would use a probabilistic representation for modeling the occupied, free and unknown space in addition with optimum runtime and memory usage. The modeling framework has to be capable of transforming environmental readings to an environmental map. We are looking for an implementation from open-source project that could be refined to specific needs for this project.

ENVIRONMENTAL MAPPING MODELS

Although several environmental modeling concepts are available, lack of finalized, working and successful implementations leads to barrier of using such concepts.

POINT CLOUD MODELS. Point clouds (PCL) are very precise and are proved to be used in static environments, however these models are not memory efficient. Large raw point cloud data, that is stored without organizing nor segmenting readings into structures, introduce modeling and computational problems. Several measurements for the same space-segment can exist with such approach. With the growing amount of raw readings, the representation of the model increases with no upper-bound. Without structuring and processing the data, it is impossible to differentiate between obstacle-free space. Such models have no information about unknown nor free areas.

ELEVATION MODELS. These structures store the height information in each cell of a discrete grid of the surface. These models provide maps of discretization the space in vertical dimension, not the actual volumetric representation. Whereas the elevation maps provide a compact representation, they lack the ability to represent vertical structures on multiple levels [37]. Upper surface of the environmental space for a specific height is stored on such maps and useful for the navigational tasks for ground [38] mobile vehicles. These models do not have full distinction between free and unknown space, they may have large memory consumption, particularly outdoors. With 3D precise dataset, precision will be lost as the map of the surface does not represent an actual space.

OCTREE MODELS. Probabilistic octree-based models avoid one of the main shortcomings of the fixed grid-map structures. Octree structure can be used as a multi-resolution representation – structure contains multi-node elements for obtaining coarser subdivision. OctoMap [39] is an octree-based framework, that is able to address large point clouds and integrate measurements into memory efficient volumetric occupancy map. Octree structure represents spatial subdivision in 3D, represented as a voxel (cubic volume). Structure is divided into eight substructures until minimum cube size is reached, which defines the resolution of a structure. This structure can be decreased at any level, allowing to have a coarser subdivision for obtaining another resolution. Different resolutions, as can be introduced on Figure 24 and Figure 25, rendered with OctoVis tool from publicly available model [39], [1].

OctoMap modeling framework is an implementation for the octree model, which uses tree-

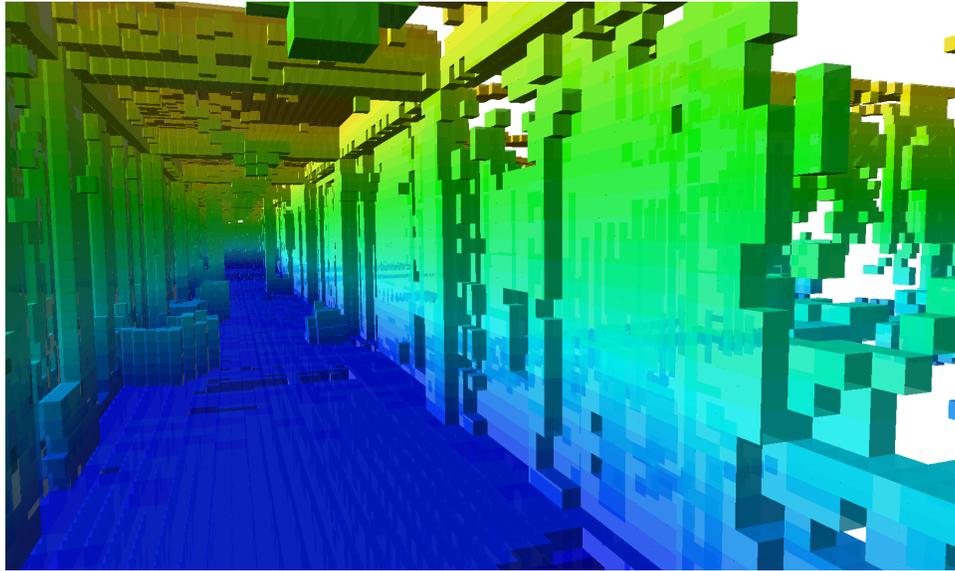


Figure 24. *OctoMap Corridor visualization, resolution 10 cm [1].*

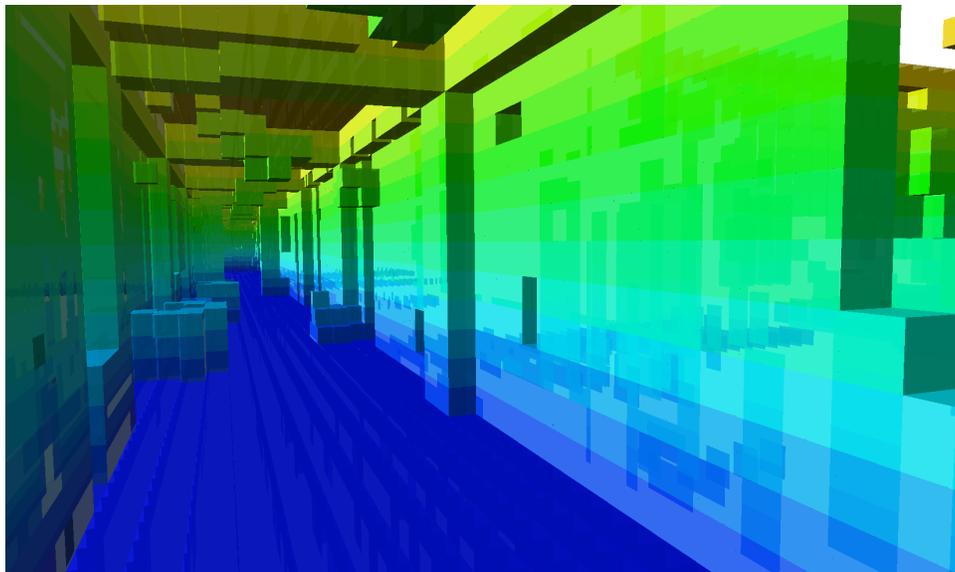


Figure 25. *OctoMap Corridor Visualization, Cube 20 cm [1].*

based representation for modeling the environment. This approach uses probabilistic occupancy estimation to ensure updatability and cope with sensor noise. Method provides compactness on the resulting models as boolean occupancy states are used. If certain space is considered occupied, the node in a structure is initialized. Unoccupied volumes in the space are also represented. Figure 58 included to Appendix, represents OctoMap visualization for an occupied space and Figure 26 shows occupied, free and unknown areas.

For the compact implementation structures are maintained as follows: if all substructures

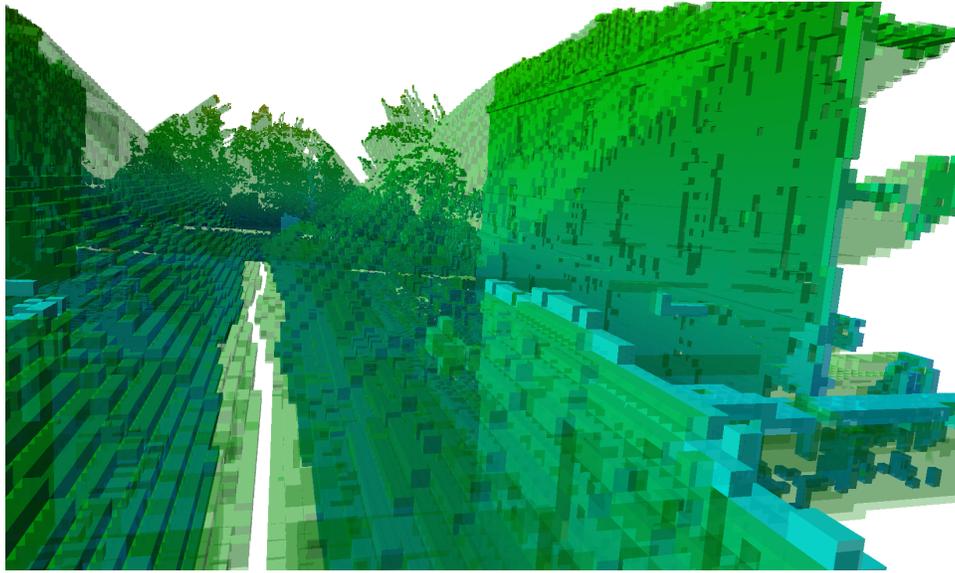


Figure 26. *Corridor Visualization, Cube 20 cm. Occupied; Free; Unknown Space [1].*

of a node have the same state of probability parameter, eg. occupied or free, the structure is pruned. OctoMap ensures that confidence of the map remains bounded and the model itself can adapt changes in the environment quickly. Compression is not completely lossless in terms of full probabilities, as structures with close thresholds ($[0; 1]$) are considered as a stable in high confidence and pruned. In between the thresholds, full probabilities are preserved. In static environments all voxels will converge to a stable state after a sufficient number of applied measurements. In such cases, as all substructures get the same occupancy state, they are pruned from the main node. When new measurements are applied that would conflict the corresponding inner-node, the sub-structures are accordingly regenerated and updated.

OctoMap framework is available as an open-source BSD-licensed C++ library and has been successfully applied to several robotics projects. There are publicly available real-world datasets that can be used for simulation and testing while implementation process takes place. OctoMap approach is able to update the environmental representation efficiently keeping the memory requirements at a minimum.

4. Environmental Perception Approaches

4.1. Environmental Perception

Small-scale vehicles can be used in narrow outdoor and indoor environments, performing different tasks [14] such as search and rescue, environmental monitoring, security surveillance and inspection. Today, different approaches are used for designing such applications from standalone vehicles to the aerial vehicle swarms. However, there are technical challenges for designing such vehicles, including physical design, perception capabilities, actuation, control and navigation. All topics and challenges are fused into one project which results an integrated system.

Reliable and accurate enough environmental perception is one of the main preconditions for the UAV's operations, awareness of the environment and localization within the surroundings are important topics. For the UAV being capable of autonomously operating in complex and challenging environments, it is important to address the focus on topics of the perception and vehicle's control. More precisely, usually the UAVs themselves are not autonomous [40] – autonomy is implemented on a subsystem that involves with collecting and processing the perceptual data for providing the UAV suitable navigational trajectory to follow.

The UAVs offer wide application usage and design possibilities, demanding different approaches as it comes to the environmental perception. Unmanned ground vehicles compared with the aerial vehicles require fundamentally different approach as the main restriction includes the weight and payload that vehicles can lift, therefore the whole perceptual system and on-board devices limit the design-capabilities. Since limitation rises from the UAV's physical capabilities, including on-board power for actuation, sensing and computation operations, the physical design determines and sets the restrictions to permissible payload, which dictates on-board perceptual sensing and computational capabilities.

While vision sensors are appealing [14] – they require computing power to extract meaningful information for navigational operations. Creation of the environmental representation serves as an input for the navigational control. Input can be either fixed or adaptive, allowing to detect dynamic obstacles as well as fixed ones.

For autonomous navigation, 3D localization and mapping is required. Localization determines the UAV location in the environment in respect to some reference location. Mapping is the process that creates the representation of the environment using sensors. As the exteroceptive sensors go smaller, the UAV-related research has progressed over time, favoring development of the autonomous UAVs. Multi-rotor copters have several advantages compared with fixed-wing UAVs – they are able to take off and land vertically, also hover on the spot. This capability allows such vehicles to easily work in a small or narrow [14] indoor environments, able to traverse through small spaces.

Area of interest is perception system for the MAV, which is capable of retaining enough payload to perform complex tasks. Many different platforms usually provide a good stabilization and way-point navigation functionalities, but effective obstacle detection and collision avoidance [17] remains as an open question. Environmental mapping and obstacle detection with collision avoidance for moving vehicles has been topic of interest for wide number of research, using different technologies such as external motion capture, monocular or stereo vision and LIDAR.

4.2. Approaches

Different technologies offer wide set of design possibilities for perceptual representation. Depending on the goal of the UAV, various approaches and technologies may be used for developing such systems.

INFRARED AND ULTRASONIC SENSORS

It greatly depends on the environment which sensors are reasonable to use. Infrared (IR) sensors may have difficulties working in direct sunlight or poor weather conditions such as heavy smoke or fog [41]. Usage of this type sensors may be problematic even indoors due to the reflection issues which may result possible inaccurate measurements. Since the light reflects differently from different surfaces and colors, the measurements may differ even if the range is same.

Ultrasonic (US) sensors cannot detect sound-absorbing surfaces properly, which makes them unreliable to detecting people as obstacles accurately. Another problem arises with "ghost-echoes" – surfaces may reflect sound in unexpected pattern.

The IR and US sensors are used for short-range applications, limiting usage of these sensors on the vehicles that operate fast velocities. In [41], the researchers used 18 IR and 14 US sensors for obstacle detection and collision-avoidance functionality on their UAV for redundant 360° coverage. Situation-awareness for fixed obstacle avoidance was created by fusing the measurements. Successful test showed that the UAV is capable of avoiding collisions with objects such as walls and people while distance was controlled towards them. It was stressed that usage of only US sensors is not enough as previous works of authors, using only the US sensors for avoiding collision, failed to detect persons reliably. Although implementation shows successful results in [42], for the UAV with only 4 US sensors used for obstacle detection, US sensors may fail to detect persons reliably. Particularly, it is not clear how "dead-zone" problem is solved as US sensor's cover-span is not wide enough, leaving possible blind-spots for the UAV, which may result in ignoring possible harmful obstacles.

LIDAR-BASED SYSTEMS

LIDAR, Light Detection and Ranging, is a remote sensing method that uses light in the form of a pulsed laser to measure ranges, offering precise, three-dimensional information about the shape of the surface-characteristics.

[15] is investigating creating perception of moving obstacles in 3D environment. For this approach, Velodyne HDL-64E LIDAR-system [2] is used for ground vehicle. Such systems provide enormous amount of data, as a single scan of the Velodyne HDL-64E LIDAR consists of approximately of 100 000 points. To sense the environment, 3D point cloud output is provided, represented further into an occupancy grid representation. This approach gives very precise environmental information, however due to the costly processing operations, very high power consumption and weight of the perception system, this approach is suitable only for robust ground vehicles.

In [16], for creation of the environmental representation and localization, both computer-vision and LIDAR-system approaches are used along with intermittent GPS reception. LIDAR is used for short-range operations while the vision-system utilizes the course of point-of-interest exploration and generates full two-dimensional (2D) global map. A LIDAR-based system is used for detecting dynamic obstacles on the 3D grid-model-based using a suitable cost-model for detecting obstacles that are closest to the UAV.

Fully autonomous indoor exploration is achieved in [10] by using 2D laser range-finder sys-

tem as a main sensing device. Aerial system provides enough computational resources to perform high-level tasks such as localization in space, environmental mapping and obstacle detection. Based on a global occupancy grid-map, where grid is categorized into free, occupied and unknown cells, exploration tasks generate most reasonable 2D-trajectories. Exploration algorithm is implemented to avoid obstacles towards the goal, unknown spaces are also avoided. Although 2D range-scanner is considered enough for localization of free-standing obstacles, the full 3D-environmental information is not presented, therefore it may be not possible to sense all important obstacles. As a future work, 3D path-planning is considered with improved planning and tracking abilities to find more efficient obstacle-free trajectories.

COMPUTER VISION

Challenges for designing environmental perception systems are addressed in [43] for MAV vehicles. Perceptual models for utilizing range sensors such as ultrasound and infrared may not work in complex environments to ensure full autonomy for the aerial vehicle. Range-finder scanners provide relative distance to target, but the weight of the systems may serve as a restriction. Challenges for vision systems are described – expensive computation possibilities, fusion of camera information with inertial data in respect to the base system, efficient reconstruction of 3D environment based on specific model, adaption to possible flight dynamics, vibrations for image processing and performance in real-time.

DEPTH CAMERA

In [44], RGB-D camera is used, providing color images and dense depth-maps for creating environmental awareness. An approach combines 3D depth-information with images resulting dense 3D-environmental representation. Localization is accomplished from visual features and match against previous images, resulting 3D point cloud. Before producing data into occupancy grid representation, point cloud must be generated, from extraction of visual features – matching against previous images. Although results allow this approach for on-line operations, the mapping result can contain erroneous edges as depth-data can be inconsistent with color images which may lead to poor autonomous utilization. Since the visual data-processing may be a costly operation, this approach may not be preferred option for MAV systems that appeal energy-efficient approaches. Furthermore, such sensors are sensitive to lighting conditions such as sunlight, preventing outdoor operations for the aerial vehicle. Depth-information may not be as precise to detect dynamic obstacles near the vehicle [45].

STEREO VISION

In [12], occupancy grid representation of possible obstacles from stereo vision is created while navigational function rapidly computes optimal paths, improving the trajectory for the UAV. Way-points are pre-defined on the map, path planner avoids the obstacles. Usually mission-critical algorithm implementations have been done off-board, because stereo image processing is very computation heavy operation. Such vehicles may not be able to avoid obstacles at close range while in flight. Problems occur when connection is lost between off-board and on-board computer. This issue is particularly important for indoor implementations.

The MAV [11], further development of [12] and [13], which is geared towards global localization and autonomous exploration of unknown environments using stereo vision, provides depth-map for obstacle detection. The platform is able to run vision-based flight control and optimized stereo-vision-based obstacle detection on on-board computer on MAV. System integrates a computing board on MAV, that is powerful enough to handle costly image processing and flight control operations.

Researches ([11], [12], [13]) are focused on [46] for efficient design on computer vision algorithms for state estimation, environmental mapping and trajectory planning operations. Addressed topics include the need of significant computational resources. Aim is to explore unknown environments on different MAV platforms with refined routines for computer vision and environmental perception. Incremental occupancy grid-map serves as an input for navigational tasks. It is stressed that autonomous exploration needs efficient routines to create a perception map, remaining computational resources have to be carefully managed. Therefore, suitable and efficient trajectory planning and navigational algorithms can be executed for full autonomy. Improvements to previous design, including [11], [12] and [13] are carried out to keep map size even smaller than what OctoMap framework is offering.

Although implementation is done for a ground vehicle [47], lacking fundamental restrictions as the MAV systems do, the perceptual model of 3D occupancy grid is used to achieve autonomy for navigational and obstacle avoidance tasks. Probabilistic incremental approach is used to represent the environment as the vehicle moves, allowing to successfully implement dynamic obstacle avoidance routines.

On the other hand, in [17], using fixed-wing vehicle, a dynamic obstacle detection, plan-

ning and feedback control is performed in real-time with promising results. Optimized fast stereo-vision algorithm is used for detecting obstacles without overburdening an on-board computer. From stereo images – local 3D maps of the environment are created, however as vehicle moves too fast, at approximately 50 km/h, old information on the map is discarded every 4 seconds. This approach allows fast processing and trajectory planning capabilities. Today, this implementation is the fastest aerial vehicle that is capable of performing obstacle avoidance in complex environments at high speeds.

MONOCULAR VISION

Monocular on-board camera is used for navigational routines in [48] for achieving a localization and mapping functionality, however, the dynamic obstacle detection for achieving full autonomy is not realized nor considered. Inexpensive lower-quality consumer monocular camera is used, presenting image processing challenges like noise and accuracy. Sensor is used for performing off-board localization and mapping implementations for estimating position and direction of the UAV. Due to lack of depth information from monocular camera, map-scale observability problems occur. For optimizing the implementation, it is possible to use additionally other sensors or different image-processing approaches to refine the result for the project. Authors do not specify the practices used for mapping the environment.

Similarly, [49] uses monocular camera approach for localization and mapping operations. Incremental depth-map of the environment is created with image processing. The map is processed and optimally re-arranged to aerial vehicle swarms, which explore the desired area without obstacle avoidance functionality. Depth-maps are not detailed, rather general shape of an environment is reconstructed. In unprepared outdoor environments, the level of details and features on these elevation maps is considered largely sufficient for scope of the research. Challenges for implementation improvements may mainly be related with computational power limitations.

4.3. Possible Design

There are many possibilities and different approaches to create environmental awareness system to achieve efficient implementation for our goal (described in Section 4.2). As the goal is to accomplish autonomous unmanned aerial vehicle for exploration tasks, system demands accurate, precise and energy efficient representation of the environment. Many approaches

and technologies were considered.

Range sensors, such as US and IR sensors can be used, however, they lack precision for building an accurate 3D occupational grid-map representation. Since it is not possible to ensure an accurate representation of 3D environment, usage of such technologies as US or IR sensors, are insufficient in our case. For accurate representation of the environment with using occupancy grid model (described in Section 3), it is reasonable to use camera-based or laser range-finder-based systems, as different successful approaches are introduced in Section 4.2.

Although camera-based solutions require significant computational resources for precise image reconstructions or advanced image processing operations, such approaches have been preferred in MAV research since small-scale LIDAR-systems were not available yet – being heavy, too big or expensive for the MAVs. Cameras measure light reflected from an object to the sensor. Images typically display a visual image of the environment similar to what the human eye experiences [50]. Unlike LIDAR, camera images usually do not measure distance in three dimensions. For creating an image information for 3D – depth camera systems or stereo vision techniques are necessary. Such approaches require significant computational resources and effective image processing methods [43], [11], [12], [13], [46], [47]. Monocular cameras also can be used to represent the environment, but output as depth-maps is not usually detailed and requires costly image-processing resources, usually general shape of environment is reconstructed [49]. When applying extensive image processing algorithms [51], dense 3D reconstruction can be achieved for the environment, but energy efficiency must be considered.

LIDAR's advantages are strong – according to [3], it is the best technology now for reliable navigation, localization and obstacle avoidance for robotic vehicles. The LIDAR systems offer possibility to produce datasets with high accuracy and greater density than traditional mapping (see Figure 27). Such systems allow researches to reconstruct the environments with accuracy, precision and flexibility. LIDARs are used to produce more high-resolution maps, precise digital elevation models for use in geographic information systems, to assist in emergency response operations, and in many other applications [52]. Although this technology provides detailed perception, such approach is not always suitable for all kind of UAV applications – final design must be carefully considered.

In recent years, the LIDAR systems and sensors have been becoming more popular and avail-



Figure 27. *Environmental Perception. Velodyne HDL-64E LIDAR Scan Example [2].*

able on the market. This sensor technology allows usage in the projects that require compact, low-power and high-performance distance measurement, such as drones, robots or unmanned vehicles. Sensors are becoming more available, compact, energy-efficient and relatively low-cost, while, on the contrary, full LIDAR-systems still remain mostly unaffordable. Today, LIDAR sensors are independent from environmental factors by means of spectral filtering ([50], [53], [54]), making it possible to work in indoor and outdoor environments, in sunlight and even light rain.

For implementing an accurate environmental perception to this project, most suitable solution is to use a LIDAR-system. With such system as accurate dataset for an environmental representation can be created. As the standalone sensor is more affordable than a fully-built system, building own low-cost device it is better choice for final implementation. Mechanism for movement-control of the laser sensor should be used, thus allowing the data collection independent from the MAV's movements and position in respect to horizontal plain.

5. LIDAR-based Environmental Perception System

Fully payload-equipped autonomous MAVs are fragile and expensive. Limited sensor-suite that MAV can lift and fast movements of the aerial vehicle make a challenge to fully automate autonomous navigational operations for such platform.

A LIDAR-based environmental perception system allows to model the environment without expensive data processing [17]. Such system consists of a LIDAR sensor, stabilizing construction and control-system to direct the sensor in desired direction. Accurate dataset of the environment can be translated into accurate environmental representation (described in Section 3) if sensor's capabilities allow.

5.1. Best Solution for the Design

5.1.1. Different LIDAR Systems

Fully-built LIDAR-based systems that could suit for the MAV platform are costly, too heavy or operate only in 2-dimensional position. Possible design for our aerial vehicle requires payload that operates in 3D (described in Section 4.3), would weight <300 g so the MAV could lift the whole system (see Table 3).

Features	 Scanse™	 robo peak	 HOKUYO	 SICK
Model	Sweep™	RPLIDAR	URG-04LX	TiM561
Price	\$249	\$398.90	\$1115	\$2000
Update Rate	500Hz	2000Hz	10,000Hz	16363Hz
Recommended Scan Rate	2.5Hz	5Hz	10Hz	15Hz
Range Indoors	50m	5m	5.6m	10m
Range Outdoors	40m	1m	4.01m	8m
Range Accuracy	1-2% of distance	0.2% of distance	1% of distance	±6cm
Range Resolution	1cm	1cm	1mm	1mm
Field of view	360deg	360deg	240deg	270deg
Power	1W@5V	1.2W@5V	2.5W@5V	3W@10-28V
Weight	120g	170g	160g	250g
Interface	UART/USB	UART/USB	RS232/USB	Ethernet/USB

Figure 28. LIDAR Systems. Different Possible Commercial 2D Devices [3].

Possible ready-made designs, such as [55], [56] (not in production yet, *LIDAR-Lite v2 sen-*

Table 4. *LIDAR Systems. Velodyne 3D LIDAR: Puck LITE™ [6].*

Features	Velodyne® LiDAR	Unit
Price	8000	\$
Rotation Rate	5 – 20	Hz
Operation Range	100	m
Accuracy	3	cm
Resolution (Vertical)	2	deg
Resolution (Horizontal)	360	deg
Laser channels	16	
Power	8W@9V	
Weight	590	g
Interface	Web	
Output	300 000	points p/s

sor), [57], [58], could fit some restrictions for the MAV, but these devices are too costly or heavy, not yet available or operate only in 2D (see Figure 28 and Table 4).

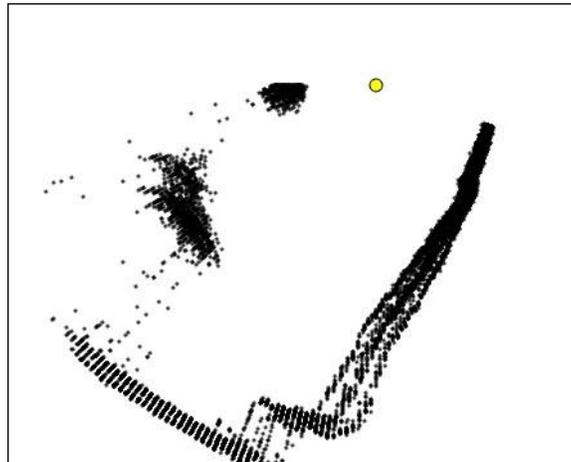


Figure 29. *LIDAR Sensor. Test for 2D Dataset.*

Systems, that operate in 2D, can produce datasets as in Figure 29. Surrounding targets such as walls and even windows, with other obstacles, are demonstrated as the black points, but the dataset is provided only from one plain. Yellow point represents the location of a LIDAR sensor.

Most common design for a LIDAR-based systems consist of one or multiple laser sensors, that spin around the yaw axis and take readings rapidly, resulting an accurate 2D or 3D dataset from the environment. Affordable available commercial devices (see Figure 28), can capture

the environment in 2D with rotation field of view 240° – 360° . On the other hand, [6] offers 360° with 3D dataset, but is little too heavy and costly for the system.

If a 2D LIDAR-based system would be used, which result 2D dataset, environmental mapping would require sophisticated MAV-depended movement in order to ensure possibility for 3D reconstruction of space. As 3D model is used, devices with 2D output datasets are avoided (described in Section 3).

5.1.2. Design Concept

Need for sensor and reasonable design for a LIDAR-based device remain as the main focus, as fully suitable commercial LIDAR-based devices are not available yet for the MAV. Target is to use one LIDAR sensor with motorized stabilization system for moving the sensor to desired direction. Such design would give an advantage to be undependable from the MAV's physical movements. Such gimbal system should provide precise information about sensor's location in the environment with respect of aerial vehicle's horizontal plain.

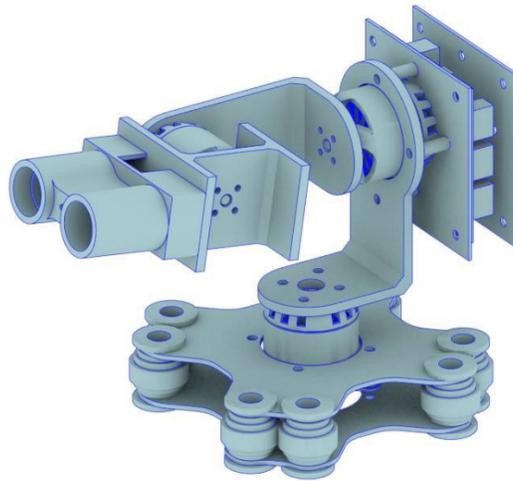


Figure 30. *LIDAR-based Environmental Perception System's Concept.*

Concept model for a LIDAR-based system is introduced in the Figure 30 [59]. The system would need one LIDAR sensor (described in Section 5.2 for details), and motorized stabilizing system (described in Section 5.3 for details), which would be able to conduct accurate movements for motion in pitch- and yaw-direction. Design idea resembles with fully rotatable systems [56], [55], but instead of capturing environment in 2D, pitch axis can be moved

in respect to horizontal plain to create 3D representation of the environment. Similar output like in [6] is achieved, but with different time-scale and accuracy, as design uses only 1 sensor instead of 16.

Motion control, stabilization, speed and smoothness of the system is important, brushless motor stabilization system is preferred. With stabilizing support, system does not depend on UAV's frame physical position in respect to horizontal plain, the sensor is moved to desired direction for scanning.

5.2. Sensor Selection

The technology allows to use smaller, cheaper and more efficient components while still achieving comparable or better performance than existing technologies, empowering design-flexibility at a low cost. As fully-built LIDAR-systems are not yet available, the main goal is to find most suitable sensor for the device.

From LIDAR sensor, short laser pulse is fired to travel from the sensor to an object and back, calculating the distance. As position of the sensor is known, possible XYZ coordinates of the surface can be calculated. Optical ranging sensors use principle of "Time-of-Fight" to calculate distance to a target, calculating time to travel from sensor to a target's surface and back based on speed of light.

Two suitable sensors were available for testing and possible usage – open-source laser rangefinder sensor OSLRF-01 [53] and LIDAR-Lite v1 [54] sensor. Following sections will introduce both sensor's capabilities and final selection will be done based on the sensors' characteristics for the MAV.

5.2.1. OSLRF-01 Sensor

Figure 31 introduces the OSLRF-01 sensor which is an open-source laser-finder sensor, allowing to detect surfaces in a range on 0.5m – 9m. The device interfaces with ADC channels of a microcontroller and includes a laser, detector optics and time-sampling circuits. These components work together to create signals that is possible to analyze, amplified and



Figure 31. *OSLRF-01 LIDAR Sensor.*

slowed down onto a manageable timebase. The output signals from the OSLRF-01 include the outgoing laser pulse, based on synchronization signal, the return signal and various timing references [53].

Laser Product is designated as Class 1M, laser is safe for all conditions of use except when used with optical aids [60].

The real-time span of the timer in the OSLRF-01 is 122 ns, which equates to a target distance of 18.33 m at the speed of light. Design expands timebase of a period to 20 ms, which allows a interfacing device, microcontroller, read the output signals. Once these signals have been captured using analog-to-digital converter (ADC) conversion, the digital representation can be analyzed to calculate the distance between the device and the target surface.

Measurement quality depends on how good the signal-processing design is. Each signal-processing algorithm embodies a timing strategy that has benefits and limitations depending upon the final application. Proposed strategy is to define the threshold for signal analysis, but as signals' shapes are different for different surfaces, the results may lead to unreliable measurements (see Figure 53 and Figure 54, both included to Appendix).

OSLRF-01 operates from 12 V power source, consuming less than 100 mA power. There are two digital (0 – 3.3 V) synchronization outputs, that can be used to manage ADC conversions for calculating distance measurements. Two analog outputs, outgoing pulse and return signal are used to analyze the distance between device and the target in one synchronization period.



Figure 32. *LIDAR-Lite v1 Sensor.*

5.2.2. LIDAR-Lite v1 Sensor

Figure 32 introduces the LIDAR-Lite v1, it is compact, high-performance and affordable device, which is easy to interface with and gives accurate distance measurements up to 40 m. Advantage is a long range detection, precision of 1 cm with 2.5 cm accuracy, allowing to create high resolution environmental image applications. Optics' and filters' technology allows LIDAR-Lite v1 to operate outdoors in the sunlight. Sensor has an acquisition time of 0.02 seconds or less and can be interfaced via Inter-Integrated Circuit (I2C) or Pulse-Width Modulation (PWM).

When measurements take place in bright light, against non-reflective target or against the target which is far, the transmitter beam has to be more powerful or collimated [61]. In order to reliably detect the returning signal in such cases, detector itself needs to be sensitive. Conventional optics' technologies use very sensitive detectors and precise clocks, which makes such devices more expensive.

The key components within the system are signal-processing algorithms and a system-architecture. For the LIDAR-Lite v1 device, a novel signal-processing is applied to perform a signature match between transmitted and received signal-pulses. It uses a signature-matching technique called signal-correlation to compare the original signal against the returned signal [62]. This approach allows to recognize and process very weak returning signal, accurately calculate distance without having directly measure it real-time [61]. The technology allows to use smaller, cheaper and more efficient components while still achieving comparable performance with existing technologies, therefore application design flexibility is provided at a low cost [63].

LIDAR-Lite v1 uses an edge emitting, 905 nm, single stripe laser [63]. This laser product is

designated as Class 1, safe during all procedures of operation, however operating the sensor without its optics or housing or making modifications to the housing can result in direct exposure to laser radiation and the risk of permanent eye damage [60].

Beam width of the LIDAR-Lite v1 is 1.5°, allowing long-range performance, thus this needs to be considered when measuring longer-range surfaces. The topic is described in Section 5.5.1.

LIDAR-Lite v1 device suits for applications which demand low power consumption as it operates from 5 V power source, consuming less than 100 mA peak power when measuring and less than 10 mA when idle.

New version of the sensor is available, LIDAR-Lite v2, which offers more flexibility as the distance measurement rate has been greatly increased. The signal-processing algorithms also have been improved.

Table 5 shows main characteristics for each sensor.

Table 5. *LIDAR Sensors Differences. LIDAR-Lite v1, OSLRF-01-01.*

Parameter	LIDAR-Lite v1	OSLRF-01	Unit
Production	Nov 2014	Feb 2014	-
Weight	26.5	57	g
Range	0 – 40	0.5 – 9	m
Resolution	1	Adjustable	cm
Precision	± 2.5	–	cm
Update Rate	50	3 – 50	Hz
Outputs Interfaces	PWM, I2C	Analog Signal Output	-
Power Supply Voltage	5	12	V
Current Consumption	<100	<100	mA
Dimensions	20 x 48 x 40	27 x 56 x 65	mm
Operating Temperature	N/A ... + 70°C	- 20°C ... + 60°C	deg
Safety	Class 1	Class 1M	-
Wavelength	905	850	nm
Total Laser Power Peak	1.3	< 14	W

5.2.3. Results

Both sensors are promising and could possibly suit for the system. LIDAR-Lite v1, however has more advantages over OSLRF-01 device – being more compact and over two times lighter, which is important in respect of system design. Power consumption for both devices stay low – LIDAR-Lite v1 is consuming less than 100 mA peak power at 5V when measuring and less than 10 mA when idle. OSLRF-01’s energy consumption stays around 100 mA at 12 V while constantly measuring. In respect of required power, LIDAR-Lite v1 is more energy efficient, as the source voltage is lower (5 V vs 12 V).

LIDAR-Lite v1 advantage is a long range detection, precision of 1 cm with 2.5 cm accuracy, giving accurate distance measurements up to 40 meters. Measurement distance of OSLRF-01 is in range of 0.5 to 9 meters. Longer-range detection possibilities give more flexibility to the MAV. Therefore, LIDAR-Lite v1 is preferred. On the other hand, as the LIDAR-Lite’s v1 laser beam’s width is 1.5°, too long distance measurements may give imprecisions. This means, when measuring longer distances with LIDAR-Lite v1, precision of the map needs to be taken into consideration.

Advanced signal-processing algorithms and routines are implemented into LIDAR-Lite v1, device is interfaced with I2C and PWM, while OSLRF-01 lacks any signal-processing. OSLRF-01 has two analog outputs (0 – 3.3 V) that are used for calculating distance measurements – quality of the measurement depends greatly on signal-processing algorithms and techniques. It is possible to calculate precise distance measurements from OSLRF-01 of ± 3 cm – testing result of [64]. Real test-results, using [65], were not accurate enough, giving inaccurate measurements from 10 – 37 cm.

In particularly, calculated measurements for darker target surfaces were not accurate with the implementation. Comparison of returning signals from OSLRF-01-sensor for different surfaces can be seen on Figure 54 and Figure 53. Different-colored targets have different height and width of return signal, making it difficult to use simple signal-processing approaches. Signals for darker surfaces in range of 6 – 9 m are very similar, therefore simpler signal-processing approaches give inaccurate measurements.

It was not reasonable to use simple fixed threshold approach for OSLRF-01, where threshold is set at a fixed height of the return signal. As this device operates from 0.5 meters, it is not possible to get accurate results in range 0 – 0.5 m. On the other hand, LIDAR-Lite

v1 gave precise results with, specially while testing the device in shorter-ranges from 0 to 15 m. Advantage of OSLRF-01 is opportunity to improve algorithms to get precise calculated measurements while LIDAR-Lite v1 returns the real distance. LIDAR-Lite v1 sensor's signal-processing algorithms are sophisticated, giving accurate final results. In addition, implementing signal-processing onto on-board system, if using OSLRF-01, will increase the need for extra computational resources.

While LIDAR-Lite v1 is suitable for the configuration, both sensors remain relatively slow as measurement rate remains at 50 Hz. PulsedLight3D has designed new LIDAR-Lite v2 sensor of measurement rate up to 500 Hz, that may be more useful for the system. With the implementation of a new signal-processing architecture, LIDAR-Lite v2 can operate at measurement speeds of up to 500 readings per second offering greater resolution for scanning applications [66]. Improved newer version of LIDAR-Lite v3 sensors have been announced in March 2016, but with no detailed information.

5.3. Gimbal System

Gimbal is motorized stabilizing system for keeping mounted device leveled and stable. Gimbal compensates for undesirable movement for the mount that requires precise positioning irrespective of the movement in the surrounding frame of reference [67]. Stabilizing is accomplished by directing energy to the gimbal's motors in response repositioning data from the gyroscopic sensor, which is on the primary mount. Sensor registers any repositioning that must be compensated [67].

Gimbal systems are different in design, robustness, size, price-range and application usage. Such systems are widely designed and used in film, television and aerial photography for achieving smooth and stable pictures or video output. Main characteristics of such systems are (i) compact and lightweight design, (ii) easy installation, (iii) high precision and (iv) stability. Brushless motors are preferred [68] as motion control, stabilization, speed and smoothness of the system is important – such motors offer instantaneous response and no backlash in the gear.

Gimbal systems are usually mounted to moving vehicles. Systems that are designed specially for aerial vehicles, tolerate high vibrations that may be caused by propellers, motors and aerial vehicle's frame. Usually vibration isolation mount with high elastic anti-vibration damping

bushings is included.

Gimbal systems do not depend on the UAV's frame physical position, the motion control is achieved with gimbal's controller that drives the motors to adjust desired position of the mounted device. Gimbal system consists of the frame, vibration isolation mount, motors, controller and inertial measurement unit. Final robustness and size of the design depends on the application. Regardless of the actual design and final application usage – smooth, stable and precise movement is the goal to achieve for such systems.

Such motorized system consists of the physical frame, motors and inertial measurement unit, but one important aspect is the main controller and it's software that will drive the motors and run the stabilization routines.

Engineering behind controller boards is complex, some controllers have better design and firmware which results system to be more tolerant to vibrations and is generally more stable, faster and more accurate. Final performance of the system depends on the controller, sensors and the software design. More expensive controllers offer better performance of the system, capability to drive larger motors, having more accurate motion and pose-information and running stabilization loop faster than the competing makers. Advanced features may be offered like possibility of usage of dual IMU sensors for enhanced stability, different operating modes, documentation, customization, advanced auto-tuning functionality for stabilization algorithms, vibration analysis, noise tolerance, error detection/correction from IMU sensors, and integration with serial application programming interface (API) for external control methods.

5.3.1. Physical Design

The system should be balanced on all axes when the mounted device is attached, so the load on the motors and motor wear-out would remain minimum, resulting power consumption to be effective. As such systems are usually designed for camera applications – possible ready-made design is not available that would suit for the perception system. Connection links are designed specially for mounting and holding specific camera units – when the camera is mounted, system remains balanced.

For current design, 3-axis gimbal with the anti-vibration mount will be used. Estimated

weight of the frame is important, as overall weight of the payload is limited to ~300 g (see Table 1). For the construction of the frame, it is important to fit within the estimated weight of the payload. 3-axis gimbals are considered – compact, strong and low-weight constructions are preferred.



Figure 33. *Design for Stabilizing System.*

For the construction of the frame, 3D-printed parts are used with ready-made parts of compact, low-cost, low-weight and robust Arris CM3000 gimbal system (see Figure 33). LIDAR sensor is mounted to the front side of the 3D-printed connection link. Frame consists of the carbon fiber mounts, three BLDC motors and a vibration isolation mount with eight high-elastic anti-vibration damping bushings. BLDC motors' parameters are not described in the datasheet of the initial gimbal, but this system is capable of driving payload heavier than the LIDAR sensor. Arris CM3000 is designed for GoPro cameras, connection links are re-designed for mounting the LIDAR sensor – so the system remain level after attaching the sensor.

System can be powered from external battery, for testing purposes Turnigy "Nano-tech" LIPO 370 mAh 3S1P battery is used. It is possible to power the system from the MAV's power source.

5.3.2. Controller Board

For the gimbal's system control – BaseCam SimpleBGC 32-bit controller is used (see Table 15 included to Appendix). Highly reliable controller and the software is mostly used for 3-axis camera stabilizing systems. 32-bit MCU ARM Cortex M4 72MHz is used for complex calculations that offers fast response for the system.

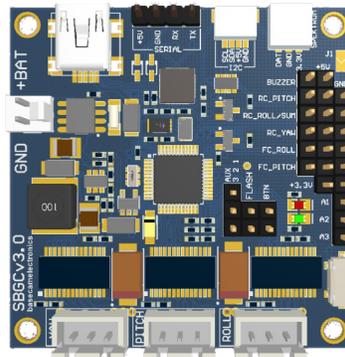


Figure 34. *BaseCam SimpleBGC 32-bit Controller Board.*

There are several advantages over cheap gimbal controllers – reliable functionality and stabilizing operations, possibility to manage the system and configuration on different platforms, available API, possible to control over universal asynchronous receiver/transmitter protocol (UART), auto-tune functionality for stabilizing algorithm, adaptive PID-algorithm for preventing vibrations and support for various configurations of the gimbal. Controller supports usage of two IMU sensors. Figure 34 shows the physical design of the board.

For control operations, sophisticated and optimized control algorithms are used to reduce CPU power allowing to implement various functionality on the same platforms for future improvements. Controller offers support for wide-range of external control protocols. Battery monitoring functionality is included, which prevents battery's over-discharging. Controller's software processes inclination angle data from inertial measurement unit – gyroscopes and accelerometers, mounted on the connection link of the platform. As usually configuration specifies desired tilting angles of the system – PID controller's stabilization routine calculates the compensation and motors are moved to correct position [67].

Although project is not considered as an open-source – drawings of mechanics, test data, solutions and samples are offered as well as newer firmware versions with improved functionality.

5.4. Final Configuration

LIDAR-based environmental perception system model consists of one LIDAR sensor (described in Section 5.2), gimbal system (described in Section 5.3): actual construction and the controller that is responsible for driving the stabilization system. External control of the system is managed by the on-board computer system (described in Section 1.2).

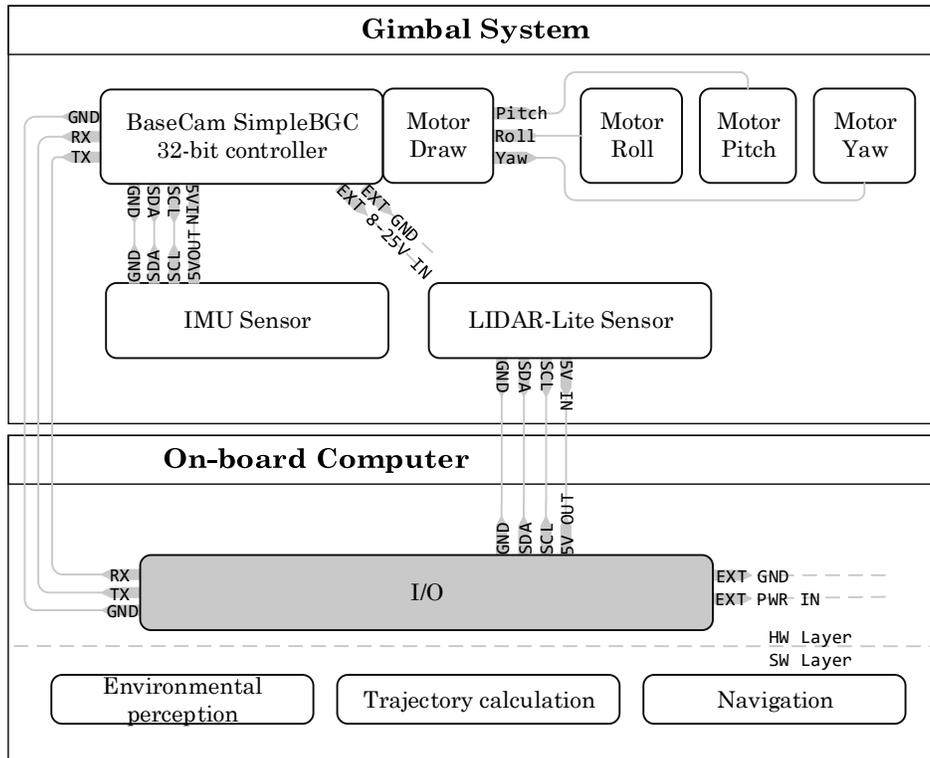


Figure 35. A LIDAR-based System's Configuration Block Diagram.

Figure 35 introduces the configuration of the system. For testing purposes Arduino Due Board is used, 84 MHz Atmel SAM3X8E ARM Cortex-M3 CPU on board. Main advantage is great number on input/output (I/O) ports: 54 digital I/O pins, of which 12 provide PWM output. Gimbal system is controlled over serial communication, open serial API's functionality is used. For testing purposes small-scale, low-weight external power is used – 3S1P LIPO battery Turnigy Nano-Tech 850 mAh, weight 71,1 g.

Scanning accuracy of the system is adjustable and depends on the real scanning routine's implementation. Sensor's resolution is 1 cm with accuracy of ± 2.5 cm. Control granularity for gimbal control for BaseCam SimpleBGC controller is minimum 0.02 degrees [69]. Maximum tilting angles for each motors are physically limited – system's movements should be

driven with caution because of possible wire-twisting. Pitch mount can move up to 45° upwards and to 90° downwards. Both roll and yaw motors can move freely 360° – wire twisting must be considered, as for the final model wiring may slightly differ. Speed for the motor is adjustable, maximum speed can be set to 100 deg/s. For high-speed configurations, depending on weight of the payload, it is recommended to make adjustments to configuration for the controller to avoid vibrations and jerks under follow-control and overshoot of target.



Figure 36. *LIDAR-based Environmental Perception System's Model.*

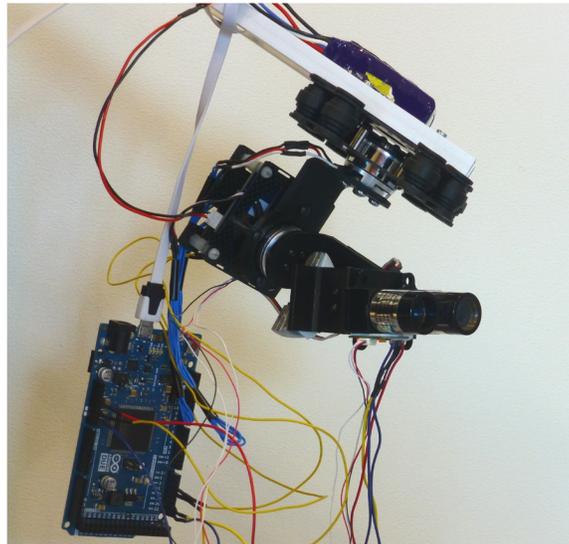


Figure 37. *Real Model for LIDAR-based Environmental Perception System.*

Figure 36 introduces the CAD-model for the system. Figure 37 and Figure 55 included to Appendix, introduce the real implementation of the system. Table 6 describes full physical

properties for the final system. Figure 57, included to Appendix, shows the parameters for the configuration of the gimbal's system.

Table 6. *a LIDAR-system Model's Final Properties.*

Parameter	Value	Unit
<i>Power Source</i>	3SP1 LIPO Battery - Separate power (or) - Direct supply from MAV's battery	
<i>Frame</i>	Custom-made using with commercial parts - Anti-vibration mount, 8 damping bushings - Carbon fiber mounts: back-side, up-side - 3D-printed connection links - Balancing attachments for front mount - 3 brushless motors	
<i>Maximum Tilting Angles</i>		
Pitch Motor	135°	deg
Roll Motor	360°	deg
Yaw Motor	360°	deg
<i>Weight</i>		
Real Weight	302.1	g
Frame	200	g
LIDAR-Lite v1fac Sensor	26.5	g
Gimbal Controller	27	g
Balancing Attachments	43	g
Wiring	5.6	g
<i>Sensor</i>	PulsedLight3D LIDAR-Lite v1	
Measurement rate	50	Hz
Resolution/Precision	1/±2.5	cm
Operating Range	0 – 40	m
Used Interface	I2C	
Power Supply Voltage	5	V
<i>Controller</i>	BaseCam SimpleBGC 32-bit	
Firmware	2.56b7	
Accuracy	0.02	deg/step
Maximum Speed	100	deg/s
Used External Control	UART (API)	
Axes	3-axis Control	
Power Supply Voltage	8 – 25	V

5.5. Environmental Scanning Concept

5.5.1. Scanning Granularity

Sweeping systems, such as [6] that are able to produce large dataset in 3D during short time, are efficient but as the cost of these systems remains very high, they are not considered due to the budget. Current system is able to provide 3D dataset, but with different time and precision scale as only one sensor is used. Sensor's resolution/precision is $1 \text{ cm} \pm 2.5 \text{ cm}$, while gimbal system can be moved with 0.02° resolution at fast speeds up to 100 deg/s . Therefore, for representing the environment, it is very important to perform the scanning of the environment with reasonable configuration: time restrictions, granularity of the scan, actuator-friendly movement to prevent mechanical wear-out and precision of the map (described in Section 3) must be considered.

For the MAV's scanning goal – it is possible to propose and implement many different environmental scanning techniques with different configurations as the system does not sense the whole environment at once. For environmental scan and creation of the global map – full scan can be made with fine granularity. For real-time safety scanning, time and expected precision are the main important aspects. While the gimbal system is capable of executing much faster movements, the sensor's measurement rate limits the results – it is not possible to acquire measurements faster than sensor provides. This means, within the time-limit, scan needs to be completed with reasonable precision. System's configuration – scanning-speed and granularity should be carefully considered.

Sensor's beam is 1.5° wide – complications may occur when the measurements will be taken for longer-range distances. As the beam is projected as a circle on the target-surface for a certain distance, not a small point, the measurements for longer distances may not be as reliable as would be expected (see Table 7). If small obstacle is located and captured by the projected beam at 40 meters, the whole block could be considered as a scanned occupied voxel. Possible solution is to move closer again and re-scan the area to update the map.

Table 7. *Laser Beam's (1.5°) Projection for Certain Distance.*

Distance (m)	Beam Radius (m)	Beam Radius (mm)	Area (m ²)	Area (m ²)
1	0.01	13	0.0005	538
2	0.03	26	0.0022	2150
3	0.04	39	0.0048	4838
4	0.05	52	0.0086	8601
5	0.07	65	0.01	13439
10	0.13	131	0.05	53755
15	0.20	196	0.12	120948
20	0.26	262	0.22	215019
25	0.33	327	0.34	335967
30	0.39	393	0.48	483792
35	0.46	458	0.66	658494
40	0.52	523	0.86	860074

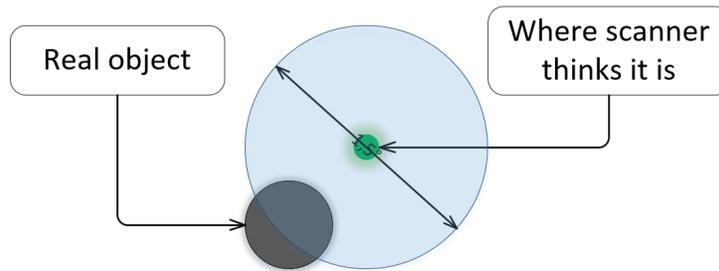


Figure 38. *LIDAR Sensor. Projection of the Beam on the Surface.*

Figure 38 introduces another problem that could rise with scanning longer-range distances – misplacement of an actual obstacle on the map. Final representation of the environment will have fixed granularity [7] of the voxels starting from 1-meter size, but they must still be placed to the correct location on the map. While the laser beam is projected as a circle on a surface, the actual measurement will be considered from the center of the projection. Although the obstacle will be sensed, this needs to be taken into consideration. The solution is to update the map at closer ranges, which would correct the representation with much reliable information.

For longer-range distances, finer scanning-granularity will not have better effect as the beam's projection increases with distance. For example – for distance of 30 meters, the area of the projected circle on the surface will increase to 0.48 m², having radius of 40 cm (see Table 7).

For shorter ranges (0 – 10 m), while the beam’s projection on the surface remains relatively small, it is possible to achieve good results.

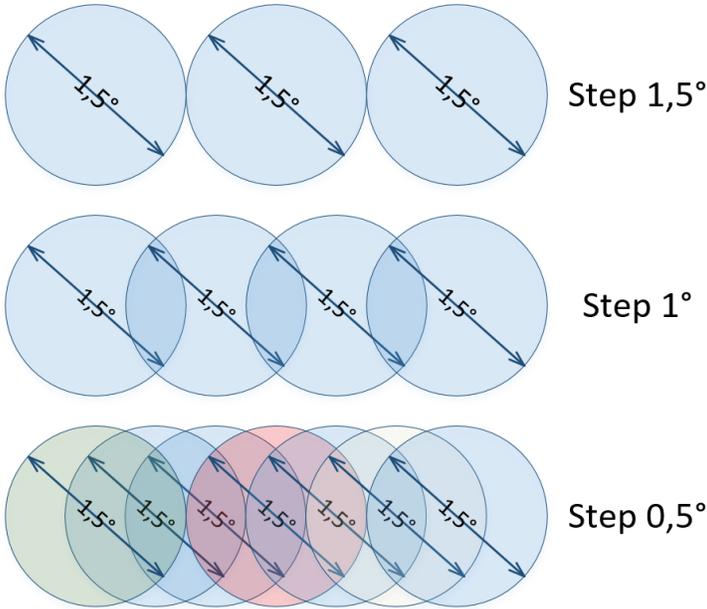


Figure 39. *Gimbal System. LIDAR’s Sensor’s Scanning Step – 1.5°, 1°, 0.5°.*

Figure 39 represents visually possible overlap for evaluating the granularity of the scan – at which gimbal scanning-granularities it is reasonable to operate. Gimbal system’s scanning granularity of 1.5° is not very suitable for closer ranges, some objects may not be scanned, therefore not sensed. Granularity of 1° and even 0.5° may give very good results with detecting objects at closer ranges.

If time is very limited, the granularity of 1° may be used, rough estimate for the scan for one plain with yaw angle (β , see Figure 40) of 50° would take approximately 1 second. Interfacing communication speeds and data acquiring time must be considered. 1° should give good results because of the overlapping areas. If time is not so critical for the scanning operations, the scan could be made with finer granularity, 0.5° will take twice as much time as 1°, but for the shorter-range measurements, result will improve in respect to 1° step. To decrease scanning time – resolutions between 0.5° and 1° are suitable.

For the real-time safety scan, while the vehicle is moving, it is very important to configure the most efficient setup, as the main aspect is the speed of the vehicle. Real-time scan must be fast, effective, actuator-friendly and should be engaged for scanning for the vehicle’s moving direction. While the vehicle moves fast, the perceptual system may not be able to detect all

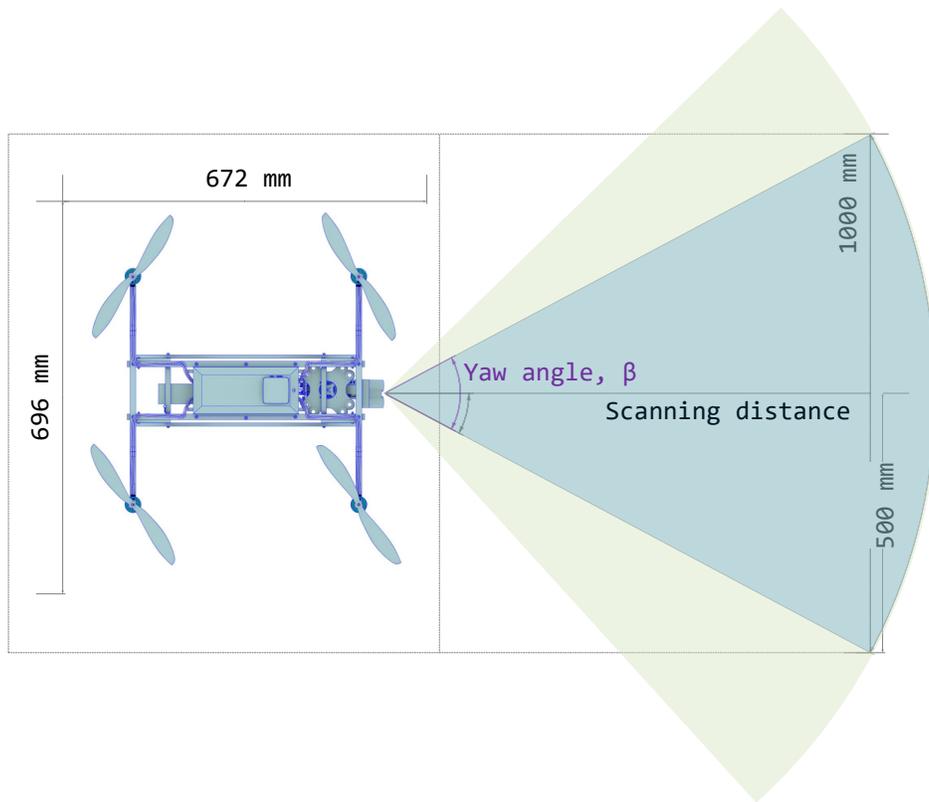


Figure 40. *Scanning Strategies. Angle and Distance (Top View).*

obstacles that may occur on the path. Furthermore, reaction time for the vehicle, scanning distance and scanning angle must be carefully selected so the UAV will have time to react to possible threats and obstructions.

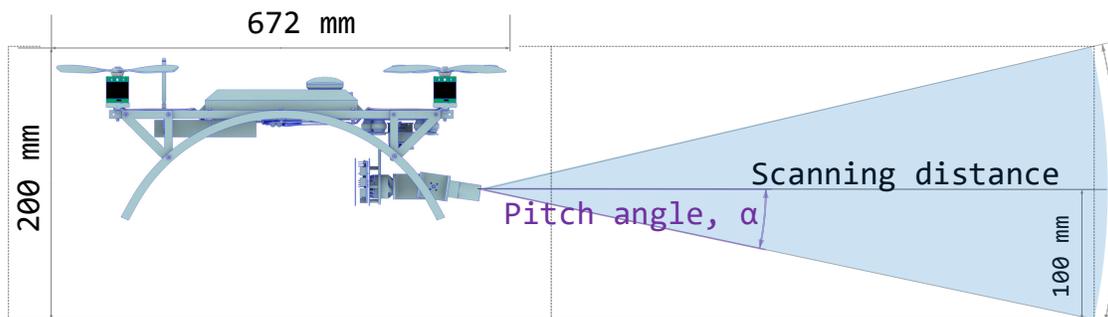


Figure 41. *Scanning Strategies. Angle and Distance (Side View).*

It is likely that dynamic obstacles that may hit the vehicle on the side, may remain out of the field of view. Figure 40 and Figure 41 show that some areas remain out of the field-of-view, scanning angles should be carefully selected. As scan is done in 3D, both scanning angles

must be considered – in vertical (pitch angle, α) and horizontal (yaw angle, β) direction. All obstacles may not be registered by such system if narrow scanning angle is selected. To solve this problem, wider scan of the area may be executed, but this affects directly the scanning time. If to use newer version of the sensor, which is able to take measurements at 500 Hz (described in Section 5.2), this problem could be avoided as the scanning speeds can be increased for the same scanning-granularity. With sensor’s increased measurement rate, the final implementation would be more flexible. Another solution to improve the real-time scanning outcome is to use sensor fusion. A monocular camera can be used to improve the functionality for detecting the obstacles. This idea will remain as an outlook for the project.

Optimum scanning region for real-time scan depends of the final navigational functionality implementation and final definition of the maximum speed and the acceleration of the MAV. Test-flights were made in an open area without any obstacles (described in Section 2.5) – maximum speed on the vehicle was set to 5 m/s, while maximum acceleration was set to 1 m/s². As the main goal of the MAV is to operate in the environments, where the obstacles are well formed and mainly not dynamic, such speeds may not be reasonable as the vehicle must navigate around the obstacles and avoid the paths with closely located obstacles. Reasonable speed, depending on the environment and final navigational routines, could be set to 1 m/s, from which it is possible to evaluate suitable configuration for the optimum scanning region (see Figure 42).

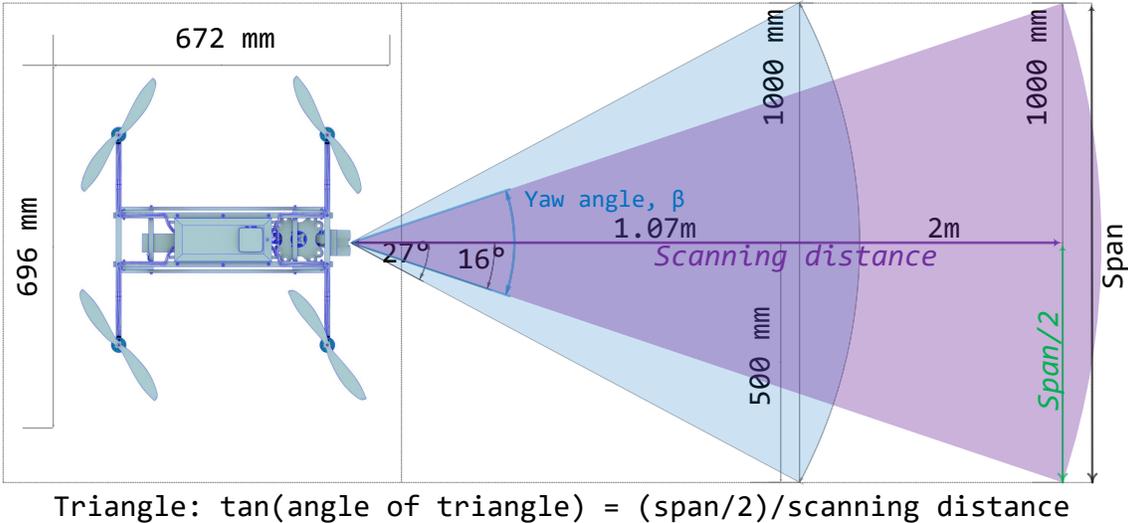


Figure 42. Scanning Strategies. Optimal Scanning Distance.

Figure 42 shows, that with the speed of 1 m/s and the acceleration of 1 m/s², it can be

reasonable to scan ahead for minimum 1 or 2 meters. At such range and speeds, there will be time to react to the possible obstacles. According to Table 7, distance measurement of 1 – 2 meters can give quite precise results as the projected beam on the surface remains small. On the other hand, the scanning angle must be considered, if it is wide-enough to detect all obstacles. As time is the limitation, the fastest scan could be preferred, but with a narrow angle, some obstacle may remain unseen. Another parameter is the granularity, 1° is enough for shorter ranges, specially for real-time scan, this granularity may help to decrease the scanning time.

5.5.2. Scanning Idea

Important aspects that should be considered for the implementation is the granularity of the scan, introduced above, precision of the scan, scan type, the MAV's movement and the gimbal system's location mounted on the MAV. Localization within the map is also important. It is possible to propose and implement many different environmental scanning techniques with different configurations. Main scan types for the MAV system are: environmental scan for creating the representation of the environment and real-time scan while the MAV is navigating to avoid the possible obstacles.

For the real-time scan, the scanning distance, width of the scan and granularity must be considered. It is important to have an efficient scanning routine and the direction the scanning must take place. While it is assumed that the MAV moves always front-frame forward, the scanning system can correct its position in respect to the UAV's main frame while scanning operation is in progress. The scanning measurements for the real-time scan are used locally – for triggering a possible obstacle threat to the MAV in a certain distance. Gimbal's controller supports "Follow Mode" without any additional sensors needed, therefore it is possible to move the mount in respect of the MAV's direction changes. If the concept requires MAV-movement to any side and any direction, then sophisticated direction changing approach must be considered. In such case, the "Follow Mode" cannot be used.

While the real-time scanning-routine's idea is simplified, position information within the environment is needed for the full scan. This data can be acquired by the on-board computer system from the flight controller. Data from GPS sensor and magnetometer can be acquired.

When mission engages the first scan – the gimbal system can lock the position from GPS signal and heading from magnetometer. This ensures that the vehicle moves from locked

position, the offset of the new coordinate and offset of the heading can be calculated so that resulting data would remain correct for building the environmental map. Such approach ensures that calculated point cloud will be correct in reference of initial position.

Implementation of the OctoMap sets the input for the real map – map can be updated with point clouds or by separate ray-insertion operations. Information about distance and coordinates must be calculated. If the map is available beforehand, then the first scan can be added to the map for the up-date. Current position of the MAV must be located on the map. To reconstruct the map, only the location of the root node needs to be known. The concept for storing GPS coordinates to the map file needs to be investigated as a future work. As OctoMap is open-source, it is possible to make adjustments to the libraries for root-node to additionally support user-based coordinate input. Simpler solution would be storing the GPS information when the map is created within the on-board computer.

Gimbal mounting position is important aspect to take into account as the initial coordinate is set in respect of the sensor's location. Current design mounts the gimbal system to the front and batteries to the back side for counter balance.

5.5.3. Scanning Routines

When implementing the scanning routines, efficiency must be considered. The efficiency of the system depends on both – physical design and the actuator's control logic. Correct frame and suitable motors must be selected that handle the payload. The system must be balanced – stabilization quality strongly depends on balance quality. When the gimbal is not well-balanced, motors will suffer from mechanical friction.

Efficient scanning routines pro-long the lifespan for the actuators that move the gimbal system. Mechanical bearings determine the lifetime of a BLDC motor. Extreme changes in velocity, rapid changes in motor's direction and heavy load on the motor must be avoided. MAV's movement should also be efficient, avoiding the sharp movements helps to minimize negative impact on the gimbal system as it reacts instantly on motion changes of the MAV. Heavy load can cause motor excessive temperature increase that causes bearing-failures and may damage the insulating coating on the stator wires.

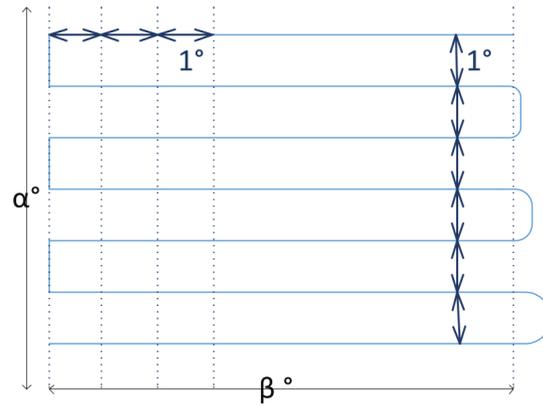


Figure 43. *Scanning Strategies. Full Scan – Plains.*

To make scanning more efficient, smooth movements of the motors can be implemented. Rapid and too fast movement and sharp changes in different directions should be avoided. Of course, the scanning must be fast to capture the 3D environment within time limitation, but different movement strategies can be used. For full scan (see Figure 43), simplest scanning technique is to move across the plain and change the angle when the whole plain is scanned. α and β denote the pitch and yaw angles of the possible scan. As this can be the simplest implementation, it is not efficient enough as the movement will not remain continuous. Left side of the Figure 43 shows rapid changes in movement that are not desirable in respect of smooth movement. To make this scan more efficient, it is reasonable to change the plains smoothly as possible, even decrease the speed when plain changes, as it is shown on the right side of the Figure 43.

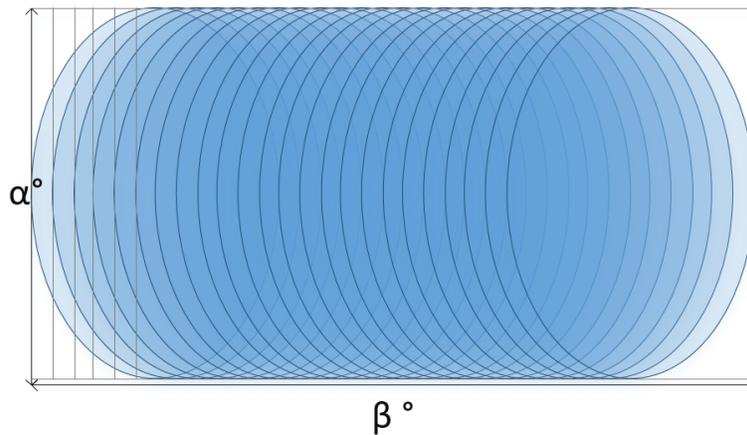


Figure 44. *Scanning Strategies. Wide Angle Full Scan – Circular Motion.*

Figure 44 and Figure 45 show more efficient scanning technique for the full scan. Circular motions with the offset can be efficient as the rapid movement is avoided, the movement

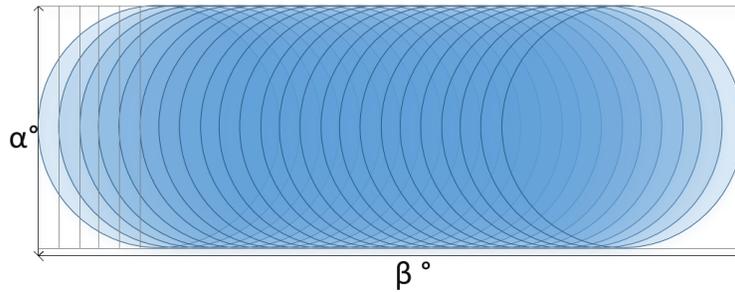


Figure 45. *Scanning Strategies. Full Scan – Circular Motion.*

remains smooth and continuous. Such technique helps also to overlap scanning areas which may improve the accuracy map for the system.

Same idea could be implemented for the real-time scan. Unfortunately, the scanning-time is the limitation as the MAV vehicle is moving. Probably the full circular scan is not suitable, as the scan will take some time to finish. The MAV is moving while real-time scan is engaged, it is essential to get obstacles' information as fast as possible for the MAV's moving direction. Therefore, lemniscate-like curved shapes for scanning may be useful. Different approaches can be used, movements similar to the curve with possible offsets can be used. This movement remains smooth and continuous. Figure 46 and Figure 47 both show ideas of possible actuator movement implementation.

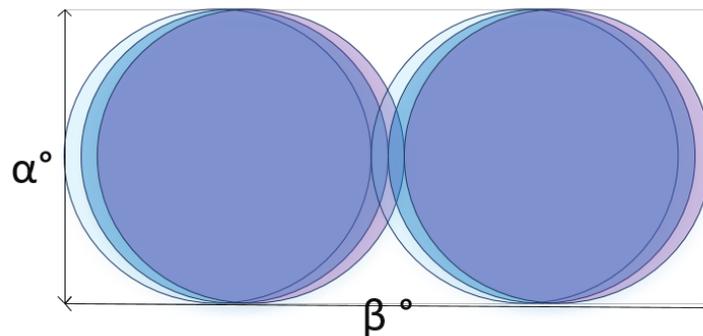


Figure 46. *Scanning Strategies. Real-Time Scan.*

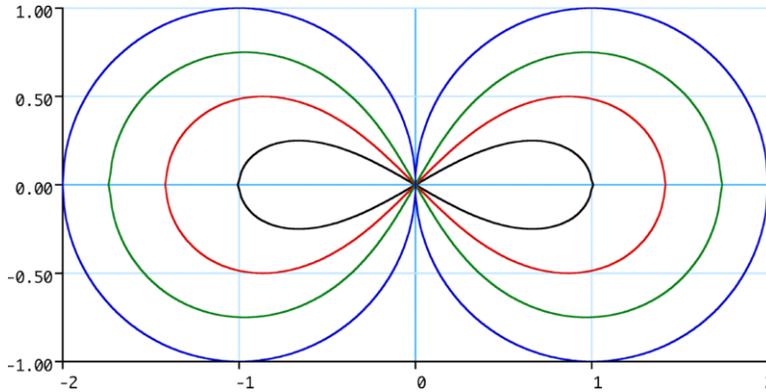


Figure 47. *Scanning Strategies. Real-Time Scan [4].*

5.6. Implementation

The files for a simple implementation and test are enclosed to [31].

Figure 56, included to the Appendix, shows the coordinate system for the gimbal's IMU sensor. IMU sensor is mounted on the primary mount of the gimbal, under the LIDAR-Lite v1 sensor. Knowing the angles α (pitch), β (yaw) and measurement distance (d), the coordinates from target object's surfaces can be calculated using Equations 1, 2, 3.

Point-cloud-like output can be used as an input for the OctoMap modeling framework. Final implementation will dictate the correct setup for a scanning routine. For testing purposes, the scanning routine is presented on Arduino Due platform, but may be ported to any platform. BaseCam SimpleBGC 32-bit controller can be interfaced with serial connection, documentation for API's functionality can be found in [69].

Scanning routine includes interfacing with BaseCam SimpleBGC 32-bit controller and LIDAR-Lite v1 sensor. The gimbal's controller is responsible for the movements of the 3-axis system. The "Follow Mode" (see Section 5.5.2), is tested and can be activated in the implementation during the scanning operations. Measurements from the LIDAR-Lite v1 sensor can be acquired via an I2C interface. Final output is transformed into 3D point cloud, which can be visualized with PCL visualization tools.

As a proof-of-concept, Figure 49 shows the testing output that the LIDAR-based system is capable of providing (real environment is shown on Figure 48). Based on this example, it is possible to distinguish tree objects from the point cloud, making it feasible to use the

LIDAR-based system for the mapping operations for the UAV for future works.

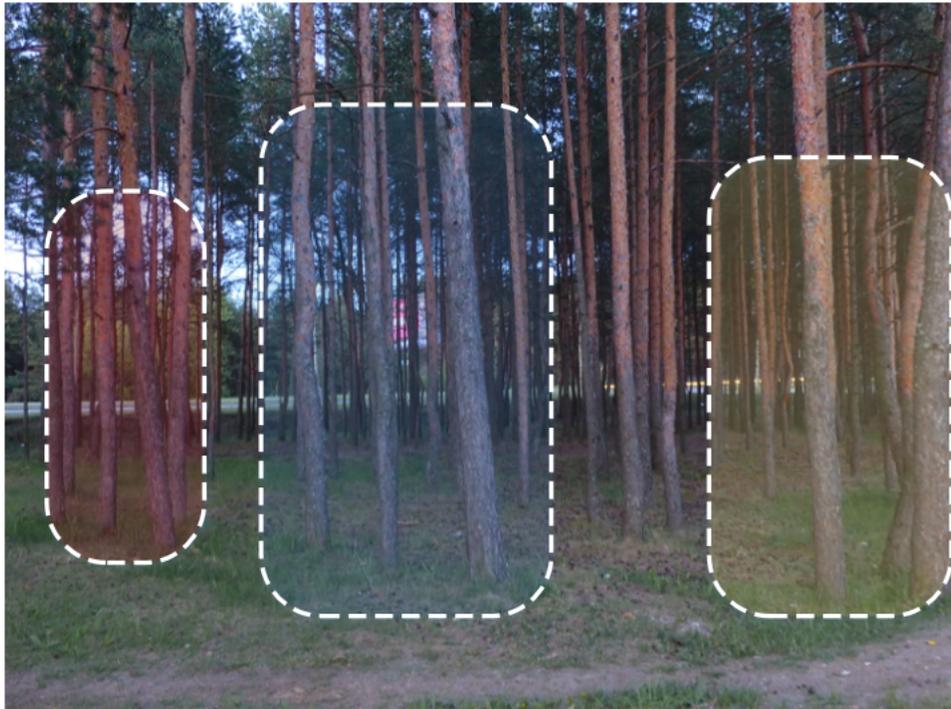


Figure 48. *Scanning Example. Testing a LIDAR-system in the Woods.*

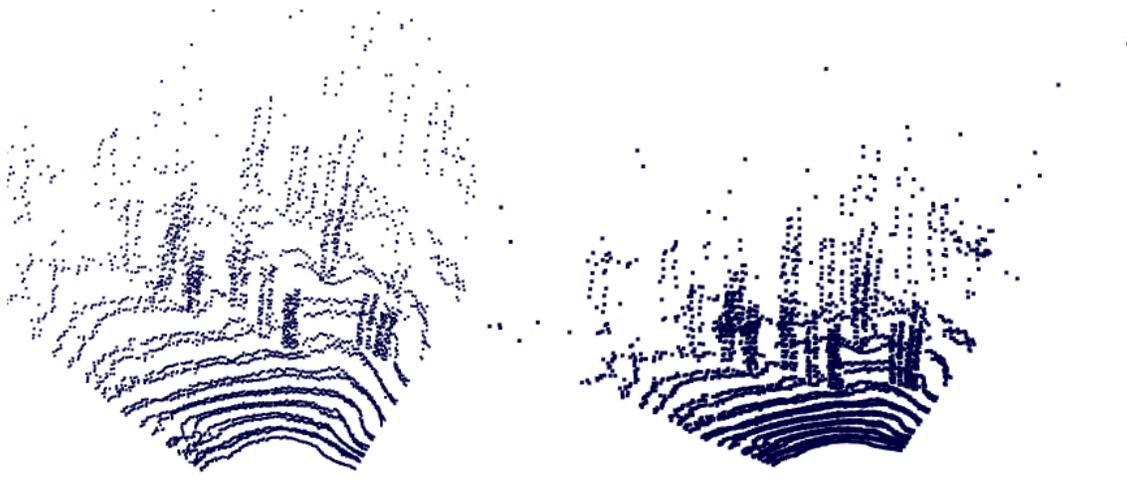


Figure 49. *Scanning Example. Top View PCL of the Woods – Scanning Step 1°.*

6. Conclusions and Future Work

6.1. Conclusion

An experimental autonomous unmanned aerial platform was created that would serve as a base for the implementation of the autonomous navigational capabilities. The main task for this thesis was to investigate the perceptual systems for the UAV. Different possible design solutions were compared and a LIDAR-based environmental perception system approach was considered – using one laser range-finder sensor and gimbal system for moving the sensor in desired directions, undependable from the UAV's movements.

Due to sensor's low measurement rate, the final implementation for the environmental scanning operations must be very effective – scanning granularity and the angle of a scan must be carefully considered. Scans, that involve longer-range measurements, may create imprecise maps – map should be updated at closer ranges for creating an accurate environmental maps.

Newer version of lidar sensor would add more flexibility and better performance to the system as measurement rate increases up to 10 times, reducing greatly scanning time and offering more precise representation of the environment while vehicle is moving.

The concept of an actuator-friendly and energy-efficient movement was introduced to extend the gimbal's operating time and prevent wear-out of the actuators.

6.2. Future Work

An efficient implementation must be accomplished. Main focus should remain on the improvements and efficiency of the system. For optimization of the system, newer, more efficient sensor can be used that would add more flexibility.

Sensor fusion could also be considered, a monocular camera could be used with a LIDAR-based system to improve the functionality for detecting obstacles in real-time. Computationally cheap implementation for an edge detection can be deployed that would not affect overall efficiency of the system.

References

- [1] Albert-Ludwigs-Universität Freiburg. Autonome Intelligente Systeme . OctoMap 3D scan dataset". (20.01.2015). [Online]. Available: <http://ais.informatik.uni-freiburg.de/projects/datasets/octomap/>
- [2] Velodyne: HDL-64E. (08.05.2015). [Online]. Available: <http://www.velodynelidar.com/hdl-64e.html>
- [3] E.Ackerman. Sweep Is a 250-dollar LIDAR With Range of 40 Meters That Works Outdoors. IEEE Spectrum. Robot Sensors and Actuators. (Published - 06.04.2016). [Online]. Available: <http://spectrum.ieee.org/automaton/robotics/robotics-hardware/sweep-lidar-for-robots-and-drones>
- [4] Lemniscate of Booth (PNG). (11.05.2015). [Online]. Available: https://commons.wikimedia.org/wiki/File:Lemniscate_of_Booth.png
- [5] Ecalc, the RC Calculator. (02.01.2016). [Online]. Available: <http://www.ecalc.ch/>
- [6] Velodyne Puck LITE TM LIDAR Device: Datasheet. (15.05.2015). [Online]. Available: http://velodynelidar.com/docs/datasheet/63-9286_Rev-B_Puck%20LITE_Spec%20Sheet_Web.pdf
- [7] L. Kurs, "Trajectory Planning for Experimental Unmanned Aerial Vehicle," Master's thesis, Tallinn University of Technology, May 2016, (Eksperimentaalse mehitamata õhusõiduki trajektoori planeerimine).
- [8] DJU Matrice 600. GNSS centimeter-level accuracy with redundant sensors.(20.01.2015). [Online]. Available: <http://www.dji.com/product/matrice600>
- [9] C. Guowei, B. M. Chen, and T. H. Lee, *Unmanned Rotorcraft Systems*. Springer, 2011.
- [10] M. Gäbel, T. Krüger, and U. Bestmann, Eds., *Design of a Quadrotor System for an Autonomous Indoor Exploration*. Delft University of Technology, 2014.
- [11] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight using Onboard Computer Vision," *Autonomous Robots*, vol. 33, no. 1, pp. 21–39, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10514-012-9281-4>

- [12] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “Autonomous Obstacle Avoidance and Maneuvering on a Vision-guided MAV using On-board Processing,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 2472–2477.
- [13] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “PIXHAWK: A System for Autonomous Flight using Onboard Computer Vision,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 2992–2997.
- [14] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier, “Vision-Controlled Micro Flying Robots: From System Design to Autonomous Navigation and Mapping in GPS-Denied Environments,” *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 26–40, Sept 2014.
- [15] A. Azim and O. Aycard, “Detection, classification and tracking of moving objects in a 3D environment,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 802–807.
- [16] S. Scherer, J. Rehder, S. Achar, H. Cover, A. Chambers, S. Nuske, and S. Singh, “River Mapping from a Flying Robot: State Estimation, River Detection, and Obstacle Mapping,” *Autonomous Robots*, vol. 33, no. 1, pp. 189–214, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10514-012-9293-0>
- [17] A. Barry, “High-Speed Autonomous Obstacle Avoidance with Pushbroom Stereo,” Ph.D. dissertation, Massachusetts Institute of Technology, 2 2016.
- [18] Pixhawk Infographic, Jethro Hazelhurst. (05.01.2014). [Online]. Available: <http://diydrones.com/profiles/blogs/pixhawk-infographic>
- [19] RC Hobby, FAQ Propellers. (17.03.2016). [Online]. Available: <http://www.headsuprc-info.com/propeller-faq.html>
- [20] SiK Radio — Advanced Configuration. (27.10.2015). [Online]. Available: <http://ardupilot.org/copter/docs/common-3dr-radio-advanced-configuration-and-technical-information.html>
- [21] APM. Mission Planner. (22.11.2015). [Online]. Available: <http://ardupilot.org/planner/index.html#home>

- [22] FRSKY Radio System. Taranis X9D Plus. (01.01.2015). [Online]. Available: http://www.frsky-rc.com/product/pro.php?pro_id=137
- [23] Raadiosaateseadmete kasutamise üldised nõuded lähitoimeseadmete klassile. (27.10.2015). [Online]. Available: <https://www.riigiteataja.ee/akt/23174>
- [24] Pixhawk project. (27.10.2015). [Online]. Available: pixhawk.org
- [25] ArduPilot Programming (Core) Libraries. (01.01.2016). [Online]. Available: <http://ardupilot.org/dev/docs/apmcopter-programming-libraries.html>
- [26] Autopilot Hardware Options: Pixhawk Overview. (06.04.2016). [Online]. Available: <http://ardupilot.org/copter/docs/common-pixhawk-overview.html>
- [27] APM Copter Source Code Licence (GPLv3). (01.05.2015). [Online]. Available: <http://ardupilot.org/dev/docs/license-gplv3.html>
- [28] A Gas Powered Quadcopter. (05.04.2016). [Online]. Available: <https://hackaday.io/project/1230-goliath-a-gas-powered-quadcopter/log/12723-firmware-and-flight-stacks>
- [29] ArduPilot: Basic Structure. (01.01.2016). [Online]. Available: <http://ardupilot.org/dev/docs/learning-ardupilot-introduction.html>
- [30] PX4 Source Code. PX4 core libraries and flight stack (01.05.2015). [Online]. Available: https://pixhawk.org/firmware/source_code
- [31] Repository for "A LIDAR-based Environmental Perception System for An Experimental Unmanned Aerial Vehicle". (25.05.2016). [Online]. Available: <https://github.com/DroneProjectTUT/>
- [32] Mission Planner Overview. (06.04.2016). [Online]. Available: <http://ardupilot.org/planner/docs/mission-planner-overview.html>
- [33] APM: Flight Modes. (06.03.2016). [Online]. Available: <http://ardupilot.org/copter/docs/flight-modes.html>
- [34] Failsafe Topics and Setup for APM Flight Software. (06.04.2016). [Online]. Available: <http://ardupilot.org/copter/docs/failsafe-landing-page.html>
- [35] Pre-arm Safety Checks. (06.03.2016). [Online]. Available: http://ardupilot.org/copter/docs/prearm_safety_check.html

- [36] ESC Calibration. (06.04.2016). [Online]. Available: <http://ardupilot.org/copter/docs/esc-calibration.html>
- [37] R. Triebel, P. Pfaff, and W. Burgard, "Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 2276–2282.
- [38] P. Pfaff and W. Burgard, *Field and Service Robotics: Results of the 5th International Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, ch. An Efficient Extension of Elevation Maps for Outdoor Terrain Mapping, pp. 195–206. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-33453-8_17
- [39] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an Efficient Probabilistic 3D Mapping Framework based on Octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s10514-012-9321-0>
- [40] N. Michael, D. Scaramuzza, and V. Kumar, "Special Issue on micro-UAV Perception and Control," *Autonomous Robots*, vol. 33, no. 1, pp. 1–3, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10514-012-9295-y>
- [41] N. Gageik, P. Benz, and S. Montenegro, "Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.
- [42] S. Bhardwaj, A. Warbhe, and B. R. Kumar, "Sensor System Implementation for Unmanned Aerial Vehicles," *Indian Journal of Science and Technology*, vol. 3, pp. 7–11, January 2015.
- [43] G. Cai, F. Lin, B. M. Chen, and T. H. Lee, "Development of Fully Functional Miniature Unmanned Rotorcraft Systems," in *Proceedings of the 29th Chinese Control Conference*, July 2010, pp. 32–40.
- [44] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGBD SLAM Systems," in *In International Conference on Intelligent Robot Systems (IROS)*, 2012.
- [45] Microsoft Kinect. Vision for Machines. (03.05.2015). [Online]. Available: http://www.roborealm.com/help/Microsoft_Kinect.php

- [46] L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Autonomous Visual Mapping and Exploration With a Micro Aerial Vehicle," *Journal of Field Robotics*, vol. 31, no. 4, pp. 654–675, 2014. [Online]. Available: <http://dx.doi.org/10.1002/rob.21520>
- [47] R. L. Klaser, F. S. Osório, and D. Wolf, "Vision-Based Autonomous Navigation with a Probabilistic Occupancy Map on Unstructured Scenarios," in *Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol), 2014 Joint Conference on*, Oct 2014, pp. 146–150.
- [48] V. Ghadiok, J. Goldin, and W. Ren, "On The Design and Development of Attitude Stabilization, Vision-based Navigation, and Aerial Gripping for a Low-cost Quadrotor," *Autonomous Robots*, vol. 33, no. 1, pp. 41–68, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10514-012-9286-z>
- [49] L. Doitsidis, S. Weiss, A. Renzaglia, M. W. Achtelik, E. Kosmatopoulos, R. Siegwart, and D. Scaramuzza, "Optimal Surveillance Coverage for Teams of Micro Aerial Vehicles in GPS-denied Environments using Onboard Vision," *Autonomous Robots*, vol. 33, no. 1, pp. 173–188, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s10514-012-9292-1>
- [50] Velodyne: Difference Between Camera and LIDAR Systems. (08.05.2015). [Online]. Available: <http://velodynelidar.com/faq.html>
- [51] J. Engel and J. Sturm and D. Cremers, "Semi-Dense Visual Odometry for a Monocular Camera," Sydney, Australia, December 2013.
- [52] National Ocean Service: LIDAR. (10.05.2015). [Online]. Available: <http://oceanservice.noaa.gov/facts/lidar.html>
- [53] Lightware OSLRF-01 (9m) LIDAR Sensor. (20.01.2015). [Online]. Available: <http://www.lightware.co.za/shop/en/proximity-sensors/24-oslrf-01.html>
- [54] LidarLite v1 "Silver Label" LIDAR Sensor. (06.04.2015). [Online]. Available: <https://github.com/PulsedLight3D/LIDAR-Lite-Documentation/blob/master/Docs/LIDAR-Lite-v1-docs.pdf>
- [55] RPLIDAR: 360 °LIDAR Development Kit. (15.05.2015). [Online]. Available: <http://www.slamtec.com/en/lidar/a1>
- [56] Scanse: Sweep LIDAR kit. (15.05.2015). [Online]. Available: <http://scanse.io>

- [57] Scanning range finder (SOKUIKI sensor, Hokuyo). (15.05.2015). [Online]. Available: https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html
- [58] SiCK LIDAR: Tim560. (15.05.2015). [Online]. Available: <https://www.sick.com/de/en/detection-and-ranging-solutions/2d-laser-scanners/tim5xx/tim561-2050101/p/p369446>
- [59] “LIDAR-based Perception System’s Concept Figure (Blender),” Auth: L. Kurs, E. Mubarakšin.
- [60] “ANSI Z136.1 Standard,” Laser Institute of America. American National Standard.
- [61] Project LidarLite by PulsedLight. Fundraiser. (01.03.2014). [Online]. Available: <https://www.dragoninnovation.com/projects/32-lidar-lite-by-pulsedlight>
- [62] S. Higgins. 3D SPAR. A LiDAR Unit for Under 100-dollar . (09.07.2015). [Online]. Available: <http://www.spar3d.com/blogs/the-other-dimension/vol13no27-lidar-unit-your-kids-can-afford/>
- [63] LidarLite v1 "Silver Label" LIDAR Sensor Datasheet. (06.04.2015). [Online]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/LIDAR-Lite-Laser-Datasheet.pdf>
- [64] Open source, TOF laser rangefinder. Test Results. (01.02.2014). [Online]. Available: <http://forum.arduino.cc/index.php?PHPSESSID=jhn9qvspn8t4g11tgvcpn6hd87&topic=213774.0>
- [65] Open Source Laser RangeFinder OSLRF01 version of Tim Eckels NewPing Library. (23.03.2014). [Online]. Available: <https://github.com/michaeljball/NewPingOSLRF>
- [66] LidarLite v2 Sensor. (01.05.2016). [Online]. Available: <https://www.sparkfun.com/products/retired/13680>
- [67] BaseCam SimpleBGC 32-bit Information. (10.05.2014). [Online]. Available: <http://www.basecamelectronics.com/simplebgc32bit/>
- [68] “Brushless DC Motor Fundamentals Application Guide,” MPS AN047 Rev 1.0 5/7/2014.
- [69] BaseCam SimpleBGC Serial API 2.5. (13.05.2016). [Online]. Available: <http://www.basecamelectronics.com/serialapi/>

- [70] Pixhawk's Main MCU. STM32F427 Cortex-M4 MCU. (01.05.2016). [Online]. Available: http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f4-series/stm32f427-437.html?querycriteria=productId=LN1789
- [71] Pixhawk's failsafe MCU. STM32F103 Cortex-M3 MCU. (01.05.2016). [Online]. Available: http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f1-series/stm32f103.html?querycriteria=productId=LN1565
- [72] ECALC. The most reliable RC Calculator. How to use Calculator. (02.01.2016). [Online]. Available: <http://www.ecalc.ch/calcinclde/help/xcoptercalchelp.htm>
- [73] RC Model Aviation Knowledge Base. (20.03.2016). [Online]. Available: <http://2bfly.com/knowledgebase/powerplants/propellers/types/>
- [74] Suggested RPM limits for APC propellers. (20.03.2016). [Online]. Available: <https://www.apcprop.com/Articles.asp?ID=255>
- [75] Thrust-to-weight ratio. (20.03.2016). [Online]. Available: <http://www.grc.nasa.gov/WWW/K-12/airplane/fwrat.html>

Appendix 1 - Flight Dynamics of the UAV

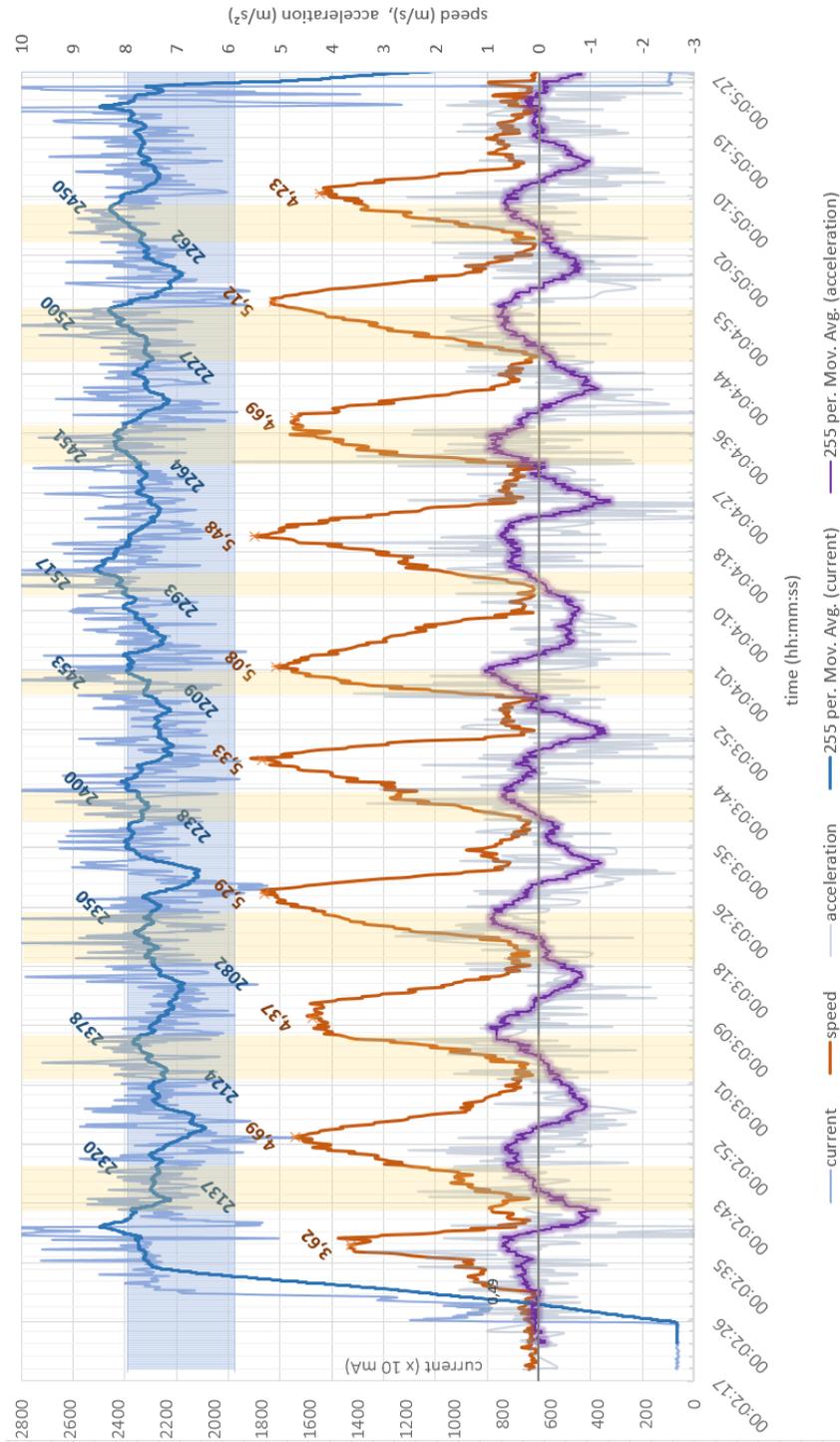


Figure 50. Full Flight Results. Guided Mode. Current Consumption vs Flight Dynamics.

Appendix 2 - General Components for the UAV

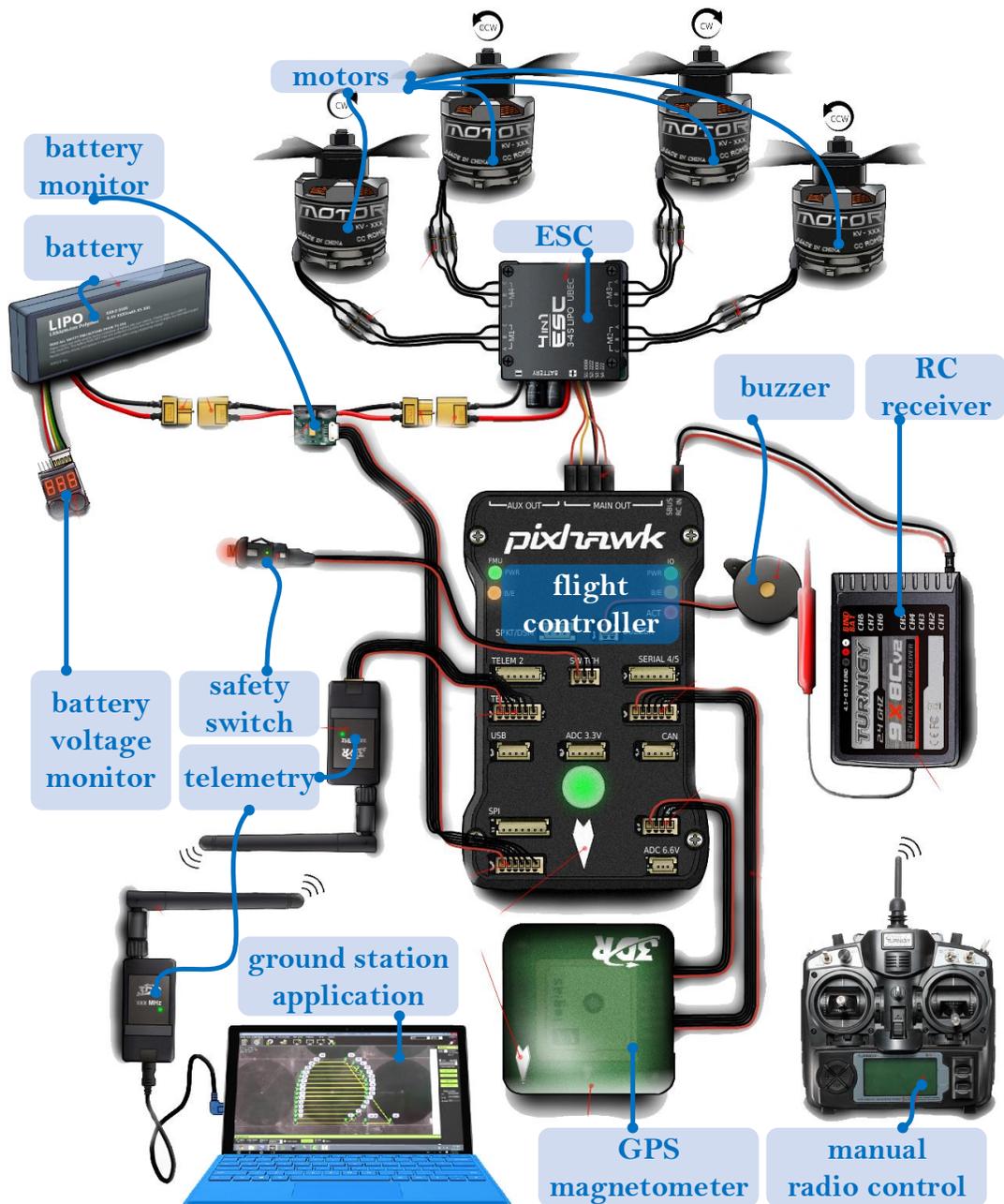


Figure 51. Components for the Quad-rotor Model.

Prop-Kv-Wizard

All-up Weight:	<input type="text" value="1800"/>	g	
# of Rotors:	<input type="text" value="4"/>	flat	<input type="button" value="v"/>
Frame Size:	<input type="text" value="400"/>	mm	
Battery - Rated Voltage:	<input type="text" value="12.6"/>	V	
Propeller - Diameter:	<input type="text" value="11"/>	inch	max. 11"
Propeller - Pitch:	<input type="text" value="4.7"/>	inch	max. 7.3"
Propeller - # Blades:	<input type="text" value="2"/>		
	<input type="button" value="calculate"/>		
recommended KV:	770 ... 1120 rpm/V		<input type="button" value="Find..."/>
min. Motor Power:	275...480 W+		
min. ESC size:	25...45 A+		

Figure 52. *Estimate on Suitable Rotor Configuration [5].*

Appendix 3 - Component Specifications for the UAV

Table 8. *Model Components: Motor Specifications.*

Parameter	Value	Unit
Model	Turnigy Multistar 2216 Outrunner	-
<i>KV</i>	800	<i>rpm/V</i>
Poles	14	-
Weight	83	<i>g</i>
Maximum Current	20	<i>A</i>
Idle Current	0.5	<i>A</i>
Maximum Voltage	12	<i>V</i>
Maximum Power	222	<i>W</i>
Connector	3.5mm bullet-typed	-
<i>Dimensions</i>		
Lenght	34	<i>mm</i>
Diameter	28	<i>mm</i>

Table 9. *Model Components: ESC Specifications.*

Parameter	Value	Unit
Model	Hobbywing 20A Skywalker Quattro universal battery elimination circuit (uBEC) 4-in-1 Brushless ESC	-
Continous Current (per 1 output)	20	<i>A</i>
Burst Current (per 1 output)	25	<i>A</i>
BEC Output	5; 3	<i>V; A</i>
Battery Cell	2S-4S, 7.4V-14.8V	-
Size	70, 62, 11	<i>mm, mm, mm</i>
Weight	112	<i>g</i>

Table 10. *Model Components: Propeller Specifications.*

Parameter	Value	Unit
Model	Carbon Fiber Propeller 12"x4.5" Black	-
Material	Carbon Fiber	-
Pitch	4.5	<i>inch</i>
Diameter	12	<i>inch</i>
Weight	15	<i>g</i>
Shaft Diameter	6	<i>mm</i>
Hub Thickness	8	<i>mm</i>
Number of Blades	2	

Table 11. *Model Components: Battery Specifications.*

Parameter	Value	Unit
Model	Zippy Compact 5000mAh 3S 25C LIPO Pack	-
Capacity	5000	<i>mAh</i>
Voltage	11.1	<i>V</i>
Cells	3	-
Continuous Discharge Rate	25C	-
Burst Discharge Rate	35C	-
Weight	354	<i>g</i>
Balance Plug	JST-XH	-
Discharge Plug	XT60	-
<i>Dimensions</i>		
Length	162	<i>mm</i>
Height	21	<i>mm</i>
Width	46	<i>mm</i>

Table 12. *Model Components: Telemetry Specifications.*

Parameter	Value	Unit
Model	RCTimer Radio Telemetry Kit 433 MHz	-
Supply Power	5	<i>V</i>
Operating Frequency	433	<i>MHz</i>
Receiver Sensitivity	-121	<i>dBm</i>
Maximum Transmit Power	100	<i>mW</i>
Configured Transmit Power [23]	10	<i>mW</i>
Air Data Rate	250	<i>kbps</i>
Standard Interface	UART	-
UART Baud Rate	57600	<i>bps</i>
Used Firmware	SIK Telemetry Radio 1.9	-
Data Protocol	MAVLink Protocol	-
Weight	15	<i>g</i>

Table 13. *Model Components: Radio Control (RC) Specifications.*

Parameter	Value	Unit
<i>Transmitter</i>		
Model	Taranis X9D Plus	-
Operating Frequency	2.4	<i>GHz</i>
Number of Channels	16	-
Operating Voltage Range	6 - 15	<i>V</i>
Maximum Operating Current	260	<i>mA</i>
Maximum Transmitting Power	100	<i>mW</i>
Configured Transmitting Power	10	<i>mW</i>
<i>Receiver</i>		
Model	X8R 16ch Receiver	-
Operating Frequency	2.4	<i>GHz</i>
Number of Channels	16	-
Operating Voltage Range	4 - 10	<i>V</i>
Operating Current	100 (@5V)	<i>mA</i>
Operating Range	up to 1.5	<i>km</i>

Table 14. *Model Components: 3DR Pixhawk Flight Control System.*

Parameter	Description
<i>3DRPixhawk Processor</i> Main Processor [70] Failsafe Processor [71]	32 bit STM32F427 Cortex M4 core with floating point unit Operating Frequency: 180 Mhz, Flash Memory: 2MB, RAM 256 KB 32 bit STM32F103C8T6 failsafe co-processor, own power supply Operating Frequency: 72 Mhz
<i>Sensors</i>	<ul style="list-style-type: none"> - ST Micro L3GD20H 16 bit gyroscope - ST Micro LSM303D 14 bit accelerometer / magnetometer - Invensense MPU 6000 3-axis accelerometer/gyroscope - MEAS MS5611 barometer
<i>Interfases</i>	<ul style="list-style-type: none"> - 5x UART, 1x high-power capable, 2x with HW flow control - 1x CAN with internal transceiver - 1x CAN on expansion connector - Spektrum DSM / DSM2 / DSM-X® Satellite compatible input - Futaba S.BUS® compatible input and output - PPM-SUM signal input - RSSI input - I2C, SPI, 2x ADC inputs - Internal micro-USB port and external micro-USB port extension
<i>External</i>	- 3DR uBlox GPS with Compass Kit 5Hz
<i>PowerModule</i>	- 3DR power module with XT60 connectors

Appendix 4 - Description for Table 2 [72]

■ Battery

- Configuration: Setup for configuration, number of cells.
- Load: Actual discharge rate in relation to the capacity.
- Total Capacity: Used setup of capacity of the battery.
- Minimum Flight Time: Expected minimum flight time, based on maximum throttle of maximum discharge % of battery and is independent of the weight.
- Mixed Flight Time: Based on all-up weight when moving, result on max. discharge % of battery, base is geometric mean value of current difference from hover to maximum throttle.
- Hover Flight Time: Expected flight time based on all-up weight when hovering only on max. discharge % of battery.

■ Motor at Maximum

- Current: Maximum estimated current draw per rotor.
- Voltage: Maximum estimated voltage per rotor at maximum current.
- Estimated RPM: Maximum revolutions for rotor at full throttle.
- RPM at Full Battery(calculated): Revolutions for rotor at 100% full battery, at 3S1P, 12.6 V.
- Maximum RPM for Propellers (calculated): Maximum estimated revolutions for specific propeller, common RPM limits from manufacturers. Estimate taken: Graupner SF CF propellers 88000 RPM/diameter [73], APC Multi-rotor Speed propellers 105000 RPM/diameter [74].
- Maximum Motor Power: Maximum load on a specific rotor.
- Electric Power: Maximum electric input power.
- Mechanical Power: Maximum mechanical output power or shaft power.
- Efficiency: Efficiency at maximum ampere draw.
- Estimated Temperature: Temperature of the rotor case. Temperatures over 80C result in rotor failure.

■ Motor at Hover (for each rotor)

- Propeller: Propeller type for setup.
- Current: Estimated current for hovering. The hover current should be close to the optimal current.
- Voltage: Rotor voltage for hovering.

- Revolutions: Rotor revolutions at hover.
 - Throttle: Stick position to hover in manual mode as input signal. Indication of power signal to rotor at hover, to aim for 50-60%. Under 50% used for racer vehicles.
 - Efficiency: Rotor efficiency at hovering.
 - Specific Thrust: How many gram of thrust is produced with one watt of electric input power at the rotor.
 - Estimated Temperature: Predicted rotor temperature - subject to the motor cooling.
- Total Drive
 - Model Estimate: Total estimate on the weight of the model.
 - Estimated Maximum Thrust: Total estimate for full thrust on a model.
 - Estimated Thrust per Rotor: Total estimate for full thrust per rotor.
 - Thrust-Weight: Dimensionless ratio of thrust to weight of a rocket, jet engine, propeller engine, or a vehicle propelled by such an engine that indicates the performance of the engine or vehicle. [75]. Flying below 1.2 is almost impossible.
 - Current @ Hover: Total current consumption for the 4 motors when hovering.
 - P(in) @ Hover: Electric input power at battery when hovering.
 - P(out) @ Hover: Mechanical output power or shaft power when hovering.
 - Thrust @ Hover: Calculated thrust for hovering, based on specific thrust.
 - Efficiency @ Hover: Total efficiency when hovering.
 - Current @ max: Sum of all motors at full thrust.
 - P(in) @ max: Electric input power at battery at full thrust.
 - P(out) @ max: Mechanical output power or shaft power at full thrust.
 - Efficiency @ max: Total efficiency at full thrust.
- Multicopter
 - Additional Possible Payload: Maximum additional payload possible to hover with 80% throttle to guarantee maneuverability.
 - Maximum Tilt: Theoretically maximum possible tilt of the copter to maintain level flight (neglecting down force due tilt).
 - Maximum Speed: Theoretically maximum attainable forward speed in flight at max. tilt and throttle (neglecting copter aerodynamic drag and down force due tilt)
 - Rate of Climb: Estimated maximum achievable rate of climb (neglecting copter aerodynamic drag).

Appendix 5 - LIDAR-based Perception System

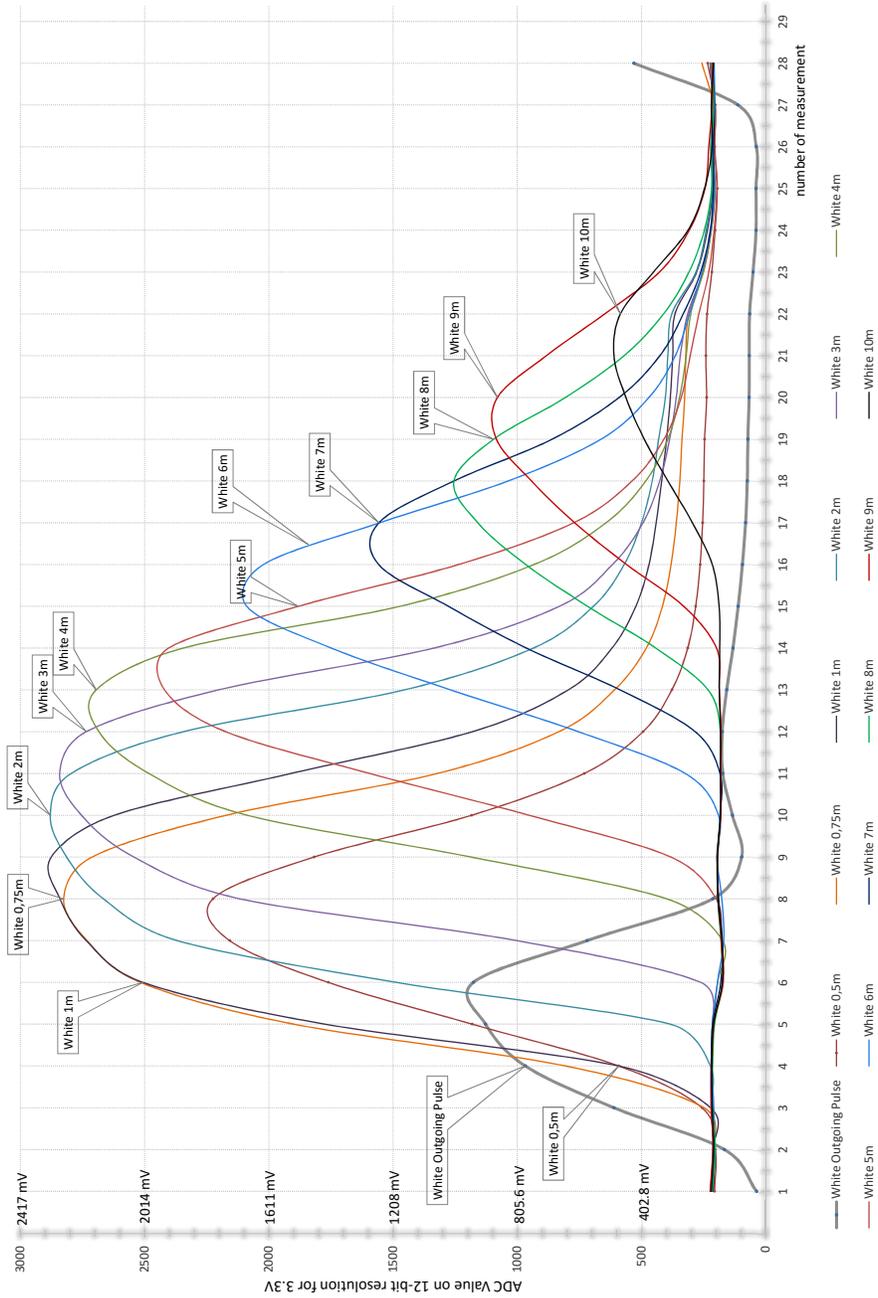


Figure 53. OSLRF-01 Sensor: Outgoing Pulse and Reflected Signals from White Surface.

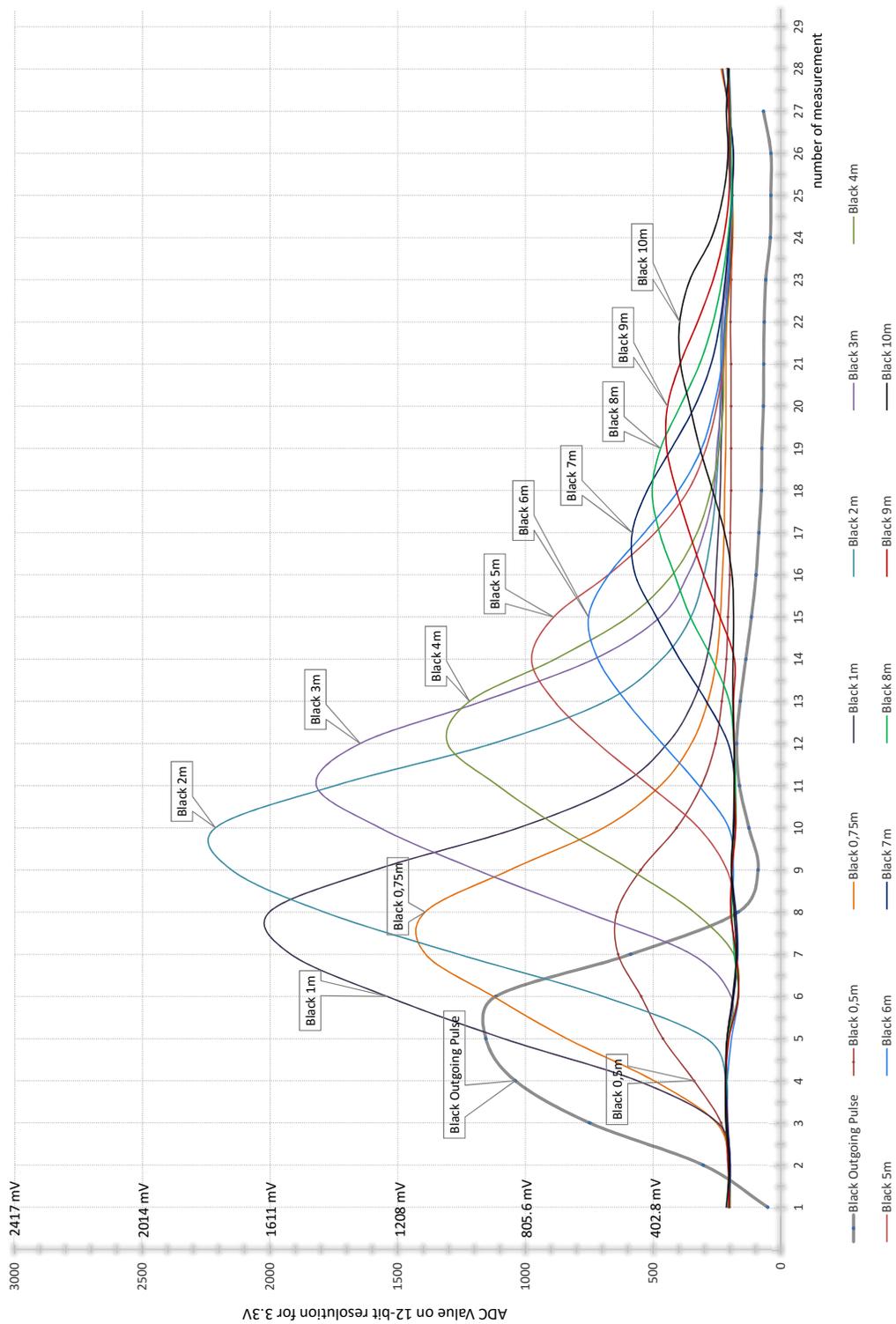


Figure 54. OSLRF-01 Sensor: Outgoing Pulse and Reflected Signals from Black Surface.

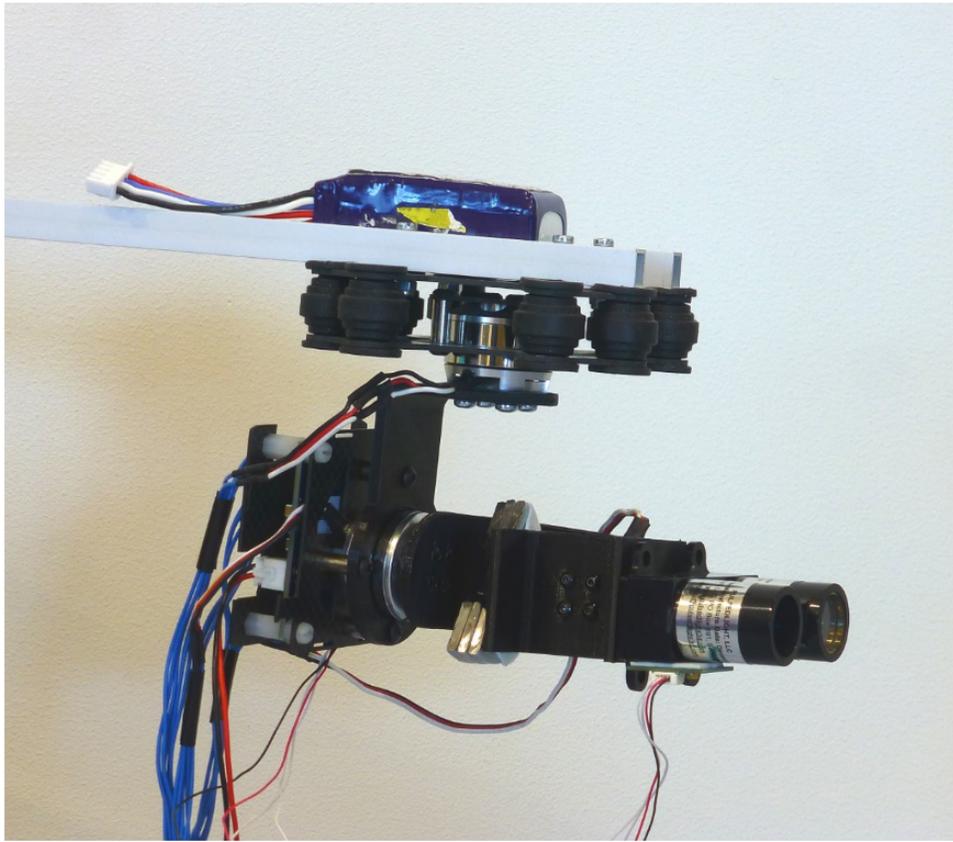


Figure 55. *Real Model for LIDAR-based Environmental Perception System.*

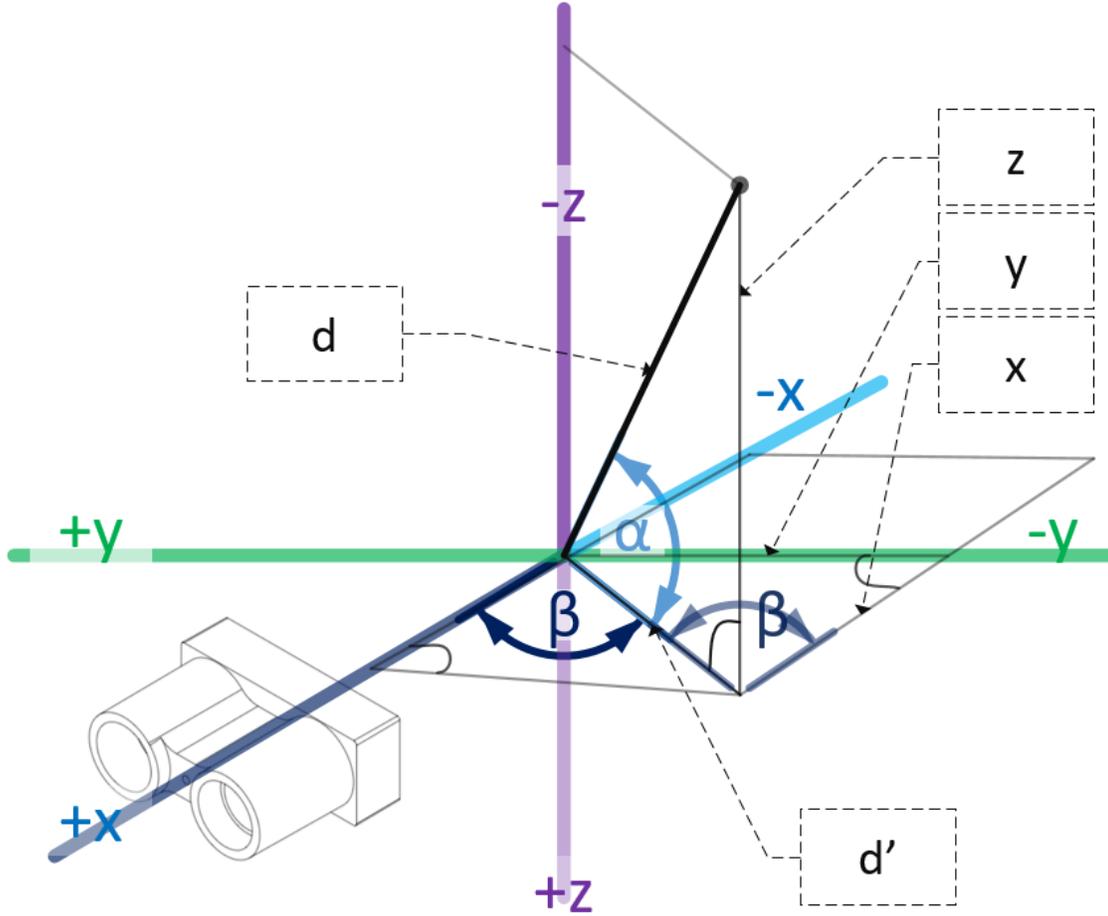


Figure 56. Gimbal System. IMU Configuration for XYZ Coordinate System.

Firstly we will find the z . As $\sin(\alpha) = \frac{z}{d}$ then it is possible to conclude:

$$z = d \cdot \sin(\alpha) \quad (1)$$

As $\sin(\beta) = \frac{y}{d'}$. Also, $\cos(\alpha) = \frac{d'}{d}$, therefore $d' = \cos(\alpha) \cdot d$. Is it possible to conclude:

$$y = \sin(\beta) \cdot \cos(\alpha) \cdot d \quad (2)$$

Lastly, the x . The $\cos(\beta) = \frac{x}{d'}$, from previous $\cos(\beta) = \frac{x}{\cos(\alpha) \cdot d}$

$$x = \cos(\beta) \cdot \cos(\alpha) \cdot d \quad (3)$$

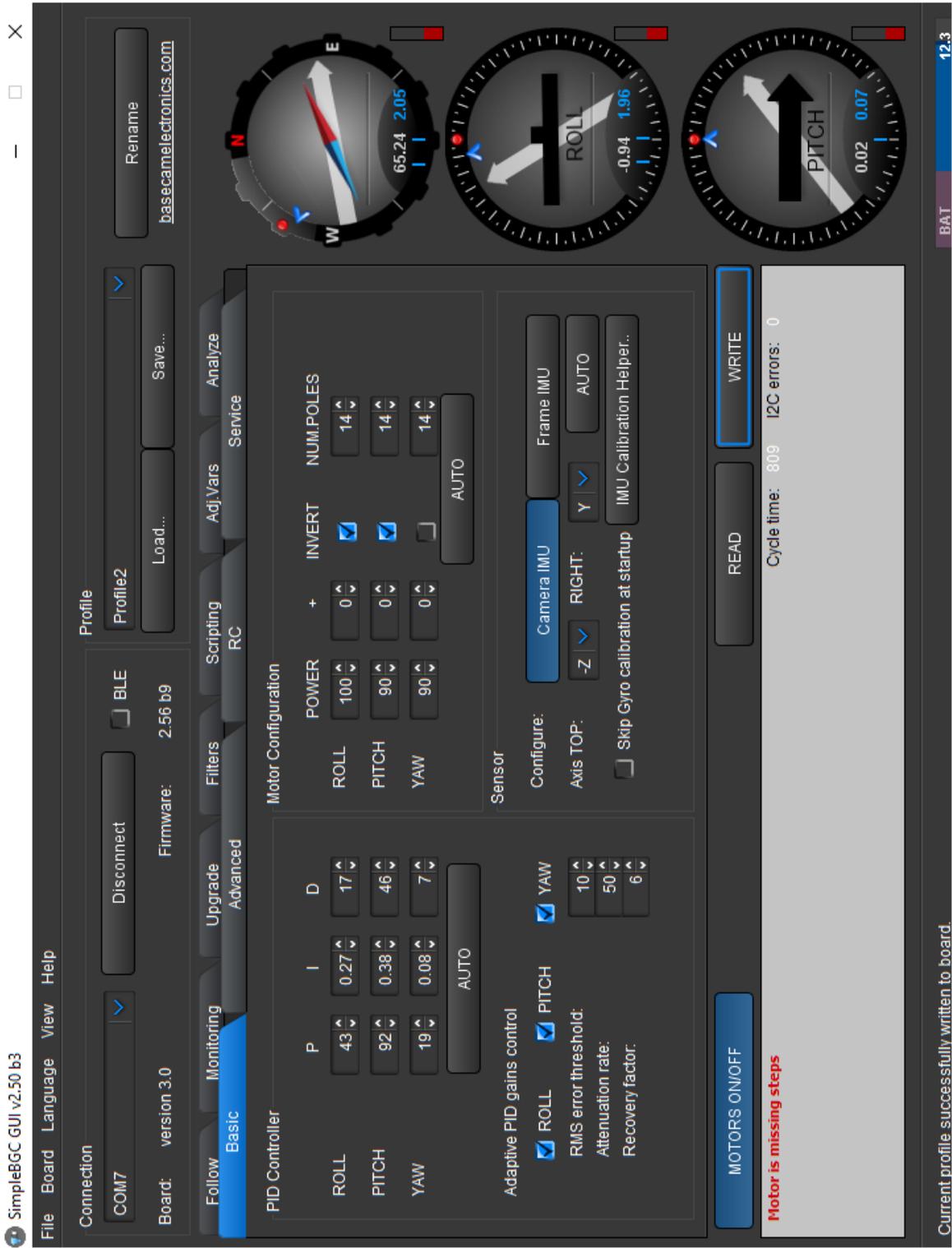


Figure 57. Gimbal System. Configuration Profile.

Table 15. *BaseCam SimpleBGC 32-bit Specifications.*

Parameter	Value
Size	50 x 50 mm
Processor	32-bit MCU ARM Cortex M4 72MHz
Power Supply Voltage	8 – 25 V (3S – 5S LIPO)
Maximum Motor Current	1.5 A, per motor
Axes Control	Yaw, Pitch, Roll
Features	<ul style="list-style-type: none"> - Open Serial API - GUI Software Support on Different Platforms - Automatic Functionality for Tuning PID-Parameters - Dual IMU Support - Optimized Control Algorithm Implementation - Follow-Mode Functionality - Adaptive PID-Algorithm for Preventing Vibrations - Support and Continous Improvement
External Control	<ul style="list-style-type: none"> - Futaba - Spektrum - PWM - Sum-PPM - UART

Appendix 6 - Environmental Mapping Model

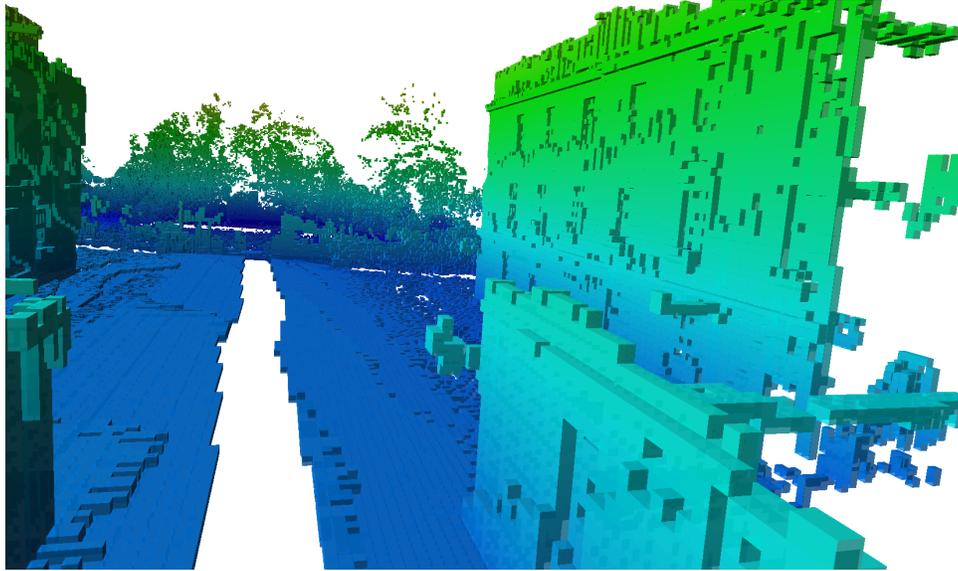


Figure 58. *OctoMap Corridor Visualization, Cube 20 cm. Occupied Space [1].*