Development Process of a Smart UAV for Autonomous Target Detection

Ryan Bobby Tang Dan, Utsav Shah, Waseem Hussain Mechatronics Engineering
Vaughn College of Aeronautics and Technology, NY, USA, btd248@nyu.edu, utsav.shah@vaughn.edu, w927hussain@gmail.com

Mentor: Amir Elzawawy, Ph.D., Hossein Rahemi, Ph.D.
Vaughn College of Aeronautics and Technology, NY, USA, amir.elzawawy@vaughn.edu, hossein.rahemi@vaughn.edu

***Abstract*– The project outlines the development of an autonomous quadcopter that can be used in GPS or motion capture cameras denied environment where the quadcopter needs to localize itself. The quadcopter developed can detect targets using only onboard sensors and compute using low cost parts. Following a preset criteria, a medium sized S500 drone was modified to function using a Pixhawk 2.1 Cube flight controller for actuator control and an NVIDIA TK1 Developer board for the processing of raw sensory data, localization, and path planning.**

***Keywords*—Autonomous, Image Processing, Target Detection, Drone, Localization, Mobile Robotics, Aerial Robotics**

I. Introduction

The development of engineering and technology has advanced the technology and application of mobile robotics. To perform autonomous tasks, the robot must have all necessary components of the See-Think-Act cycle. This cycle consists of perception, cognition, and motion control. The perceptive layer of the mobile robot intakes all raw data from the onboard and external sensors of the real-world environment and extracts the necessary information. Once the necessary information has been processed, the central logic controller then processes the information creating an environment model and local map. Through this model and map, the robot can localize itself to the global reference and use cognition, begin path planning. Thus, creating the cognitive layer of the cycle. The last layer is the motion control which includes path execution through actuator commands. Once complete, the cycle repeats until the mission task is complete. [1]

In recent years, the application of autonomous robots has expanded with the incorporation of Unmanned Aerial Vehicles (UAVs). In the field of drones, there are two very distinct classes, fixed wings and multirotor. Drones in general are influenced heavily by size, range, equipment, and weight. A larger sized multirotor provides more stability in flight, but lack speed. On the other hand, smaller sized drones lack in stability, but have increased speed and agility. Depending on your application and goal for the autonomous UAV, the weight plays a major role on the range and efficiency of the robot. In general, the heavier the drone, the less flight time it will have. In addition, increased speed results in increased power draw from the motors, decreasing flight time. The most critical factor of weight is that increased battery life increases battery size, which causes an increase in total weight.

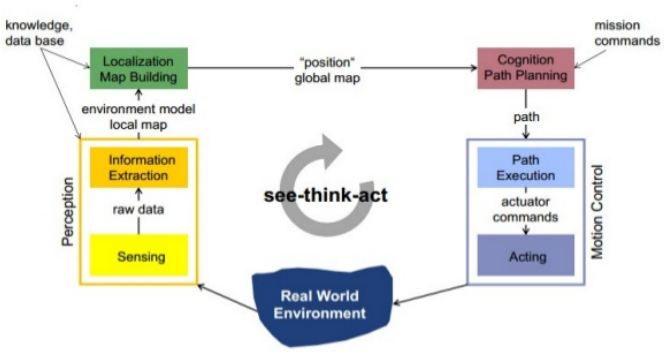

Figure 1. Autonomous Robot See-Think-Act Cycle

The application required for the drone plays an important role in the design and functionality of the quadcopter. An example of this is the equipment that will be onboard the quadcopter as well as the type of drone electronics that will be necessary to complete the mission. This includes the flight controller and motors necessary.

In current industry, the implementation of full autonomous capability has been the largest obstacle in the advancement of drone technology. Although complete autonomous has been achieved in an outdoor environment using a global positioning system (GPS) onboard the drone, very little progress has been made in autonomous flight in an indoor environment. In an indoor environment, the quadcopter must rely on either external or onboard sensors to navigate. Researchers have been able to use external sensors like motion capture camera in the environment and by using it raw perceptual data can be provided to the drone allowing it to perform the necessary maneuvers and actions.

In this paper, we present the development process of an autonomous quadcopter for target detection and obstacle avoidance. In the paper, Section II and III discusses the mechanical design criteria and final mechanical design for the

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2018.1.1.480
ISBN: 978-0-9993443-1-6
ISSN: 2414-6390

project. Section IV provides an in-depth analysis of all electronics components of the frame necessary for our application. In Sections V to VII we will discuss the autonomous criteria, movement, and target detection development process.

II. Design Criteria

A. Mechanical System

In the development of an autonomous quadcopter capable of maintaining a stable flight, several different requirements are needed for the mechanical frame. These requirements are:

1. The frame design must be rigid with minimal flexibility to the frame.
2. Vibrations created by the motors must be dampened to achieve more useful and reliable data from the onboard sensors and flight controller.
3. The frame must be medium in size with a cross shape design.
4. The drone must be symmetrical and balanced in weight.

The rigidity of the frame plays a major role in the mechanical design of a quadcopter to prevent the deformation that occurs as the motor increases speed creating torque. This is crucial as deformation prevents the quadcopter from being able to even take off. As motors increase in speed, the centripetal force creates large amounts of vibrations that can disorient the onboard sensors such as the inertial measurement unit (IMU). The cross-shape design was used to provide the medium sized drone more stability. In addition, another crucial requirement was to keep the design and electronic components housing symmetrical and balanced in weight to prevent certain motors from overdrawing power to keep the drone balanced in the air. This also works in keeping the center of gravity and the center point of the drone frame.

B. Electrical System

The electrical system of the autonomous aerial robot plays a major role in the functionality of the drone. Without the proper components, a drone will not be able to function at the level of accuracy and efficiency that is required for autonomous missions. Below are the requirements that were set for this project:

1. The motors must provide enough thrust to support the weight of all onboard equipment.
2. The flight controller must support autonomous capability.
3. The battery must be able to provide power to all the electronics for the duration of the mission.
4. Onboard sensors should be accurate enough to provide the necessary raw perceptual data.

To ensure that the drone can take off and remain stable enough to perform the autonomous mission, a crucial factor is the total thrust output created by the four brushless motors of the quadcopter. The general rule followed is that the total thrust of the four motors must be at least 2 times the weight of the UAV. Although many flight controllers exist on the market today, only a few can perform the autonomous navigation that is necessary for the project mission. The total drone weight and power draw of the motors influences the size and total capacity of the battery that is chosen. The battery must be able to provide enough power for the estimated duration of the mission. Lastly, to ensure accuracy and efficiency, the chosen onboard sensors must be able to quickly and accurately provide the necessary raw perceptual data with minimal delay.

III. Mechanical design

The design criteria set forth the requirements that the drone frame must be rigid with minimal flexibility, dampen vibration, and follow a cross shaped design. Using these criteria, it was decided that the best choice was to use a S500 Glass Fiber Quadcopter Frame with an integrated power distribution board (PDB). The frame measures 480 mm in length diagonally. In 3D printing a drone frame, it was discovered that each individual arm was not equal in weight which causes imbalance in the drone. This imbalance in the drone could cause a decrease in the overall efficiency of completing the mission tasks.

Figure II. S500 Glass Fiber Drone Frame

Although the main drone frame was large enough to hold the necessary electronic components, several adjustments were made. A mounting plate was 3D printed and mounted as a third level at the top of the frame. Before installation, the mounting plate was checked for symmetry and balance to prevent any imperfections that would alter the balance of the design. The mounting plate was then mounted to the frame using plastic standoffs to limit any additional weight as much as possible.

Figure III. Final Drone Build

IV. Electronic Components

A. Drone Components

Following the traditional drone design of a multirotor, the quadcopter motors chosen were Turnigy Multistar 2216 800KV direct current (DC) brushless motors. The brushless motors utilize a small circuit that coordinates the delivery of energy to the windings. When paired with 9-inch propellers, it was seen that each motor was able to produce 8,000 revolutions per minute (RPM) and generates a thrust of 771 grams. Through simple addition, the four motors combined will produce a thrust output of 3,084 grams with a total power consumption of 480 Watts and a current draw of 43.2 Amps. [2]

Furthermore, through the specifications chart of the motor seen

Figure IV. Multistar 2216 800KV Brushless Motor

below, it was discovered that when comparing our drone weight to the motor thrust output, the most efficient propeller size was a 9047 propeller. The 1047 propellers are 10 inches in length and travel 4.7 inches per revolution. Although larger propellers would be able to provide more thrust, the power and current consumption would not be optimal.

SPECS for Turnigy Multistar 2216-800Kv 14Pole Multi-Rotor Outrunner V2

Battery	Prop Size	RPM	Thrust (g)	Current (A)	Power
11.1V (3Cell)	12" Slow Flier	5,500	907	17	185
	11" thin Style	7,200	908	11.7	128
	11" Slow Flier	6,000	908	16.2	171
	10" Slow Flier	7,440	771	10.8	120
	9" Slow Flier	8,000	544	8.9	105
14.8V (4Cell)	11" thin Style	8,400	1,006	17	217
	11" Slow Flier	7,400	1,224	22	355
	10" Slow Flier	9,200	975	16.5	265
	9" Slow Flier	10,080	725	13.8	217

Figure V. Brushless Motor Specifications Chart

To provide full control of the speed of the motors and provide stability in flight and hovering, Turnigy Plush 25 amperage rated electronic speed controllers (ESCs) were chosen. The Turnigy ESCs can provide a continuous current of 25A to the ESCs with a burst current rating up to 35A. The speed controllers are 24x45x11mm in size weighing 22 grams, minimizing the total weight of the quadcopter while satisfying the 8.9-amp current draw from each motor.

Figure VI. Turnigy Plush 25A ESC

In the possibility of a manual kill switch or manual override is necessary during the autonomous mission, a Taranis FrSky X8R radio receiver was chosen. The receiver comes with a 16 channel SBUS port and 8 conventional channel outputs. Operating between 4 to 10 volts and 100 mA at 5 volts, the receiver can receive commands at a range of more than 1.5 kilometers. The total weight of the receiver is 16 grams keeping the total weight of the quadcopter as minimal as possible. The receiver is bound to a Taranis X9D Plus 2.4 GHz band transmitter. In addition to the receiver for manual override, a 3DR 9.15 MHz Telemetry Module was installed on the quadcopter. The telemetry module is a lightweight, small open source radio platform that can be used to communicate autonomous data between the flight controller and the computer used to monitor drone activity. [3]

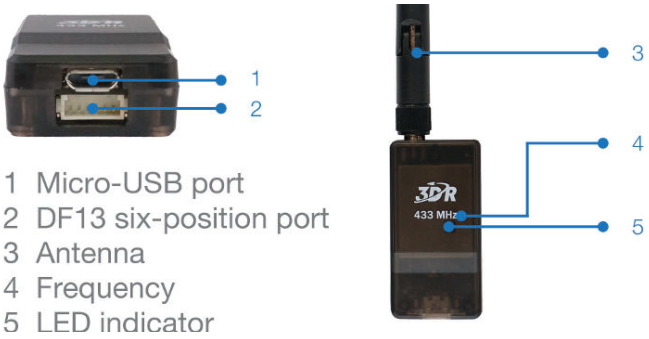

Figure VII. 3DR 9.15 MHz Telemetry Module

At the center of the entire autonomous drone system is the 3DR Pixhawk 2.1 Cube flight controller designed by the Pixhawk open source hardware community. The Pixhawk flight controller has a Cortex-M4F Core Microcontroller and a processing speed of 168 MHz, with 256 KB of RAM, 2 MB of flash memory, and a 32-bit STM32F103 failsafe co-processor.

The logic controller comes fully equipped with 5 UART, 1 I2C, 1 SPI, and 2 CAN serial connection ports for peripherals as well

Figure VIII. Pixhawk 2.1 Cube Flight Controller

as an integrated backup system for in-flight recovery and manual override. In addition, the controller has redundant power supply inputs, external safety switch, and a multi-tone, high power piezo audio indicator. The voltage rating of the Pixhawk is triple-redundant if three power sources are supplied for module input, servo rail input, and USB input. In normal operation, the maximum voltage ratings are 4.8V to 5.4V for the power module input, servo rail input, and USB power input. Due to the necessary precision and mission complexity, the Pixhawk 2.1 Cube flight controller is the most efficient choice. [4]

Although the flight controller can perform the autonomous navigation and command processing, it is unable to process all the raw sensory data from the camera and perform the necessary target detection, tracking, and path planning. The NVIDIA Jetson TK1 developer board is used to process the raw information and give the corresponding commands to the flight controller that then executes the commands to the actuators. The TK1 is equipped with several different serial port communications including I2C, UART, and USB. The NVIDIA developer board utilizes a 2.32 GHz ARM quad-core Cortex-A15 CPU with a Cortex-A15 battery saving shadow core, 16 GB of storage, 2GB DD3RL 933MHz RAM, and a 12 Volt DC barrel power jack. [5]

Figure IX. NVIDIA TK1 Developer Board

Aside from the necessary main components to create an autonomous aerial robot, several sensors are required to provide the raw perceptual data needed for the drone to localize itself with the environment and execute the proper path planning procedures. One such sensor is the Garmin Lidar Lite V3 laser rangefinder sensor. The sensor measures the distance between itself and an object by calculating the time delay that occurs between the transmission of an infrared laser signal and the reception of the reflected signal. At an operating power of 5V direct current, the sensor can detect up to 40 m. (131 ft.) with a 1 cm resolution and 2.5 cm. accuracy. This is done with a 905nm laser wavelength powered at 1.3 W. The laser pulse width is rated at 0.5μs (50% duty cycle) and 10-20 KHz pulse train repetition cycle. Thus, providing more than the required height required to perform indoor autonomous navigation. [6] Furthermore, the sensor is capable of I2C and PWM interface control allowing for connection to the Pixhawk flight controller.

Figure X. Garmin Lidar Lite V3

Another essential sensor used for the autonomous development of the multirotor is the PX4 Optical Flow sensor. The optical flow sensor utilizes a downward facing distance sensor and camera to estimate the robot's location regarding the local environment. The camera module utilizes a 3-axis gyroscope sensor, the lidar sensor and visible features to calculate the aircraft ground velocity and provide an estimation of the robot's location. As a result, both the lidar and PX4 FLOW optical flow sensor must work in unison to allow the robot to localize itself with the environment. [7] In addition to these sensors, a HD 1080P Mini Micro USB camera was used to provide real time information for the target recognition program. The camera chosen was a compact, lightweight camera to limit the overall weight of the drone. Furthermore, the size of the camera played a factor due to the size of the frame and the mounting location.

The final component of the aerial robot is the power supply system to power all onboard electronics. The battery chosen for the quadcopter was a Turnigy 5000 mAh 3 cell Lithium Polymer (LiPo) battery with an XT-60 connector. The battery consists of three power cells rated at 11.1 V with a constant cell discharge of 20 C. and a peak discharge of 30 C. In accounting for weight and flight time, the total weight of the battery was 360 grams which provided approximately 20 ~ 25 minutes of flight time. A power module was installed to provide an accurate reading of the battery and provide warnings if the battery has insufficient power to perform the tasks at hand. In addition, a 5-volt, 3-amp Battery Elimination Circuit (BEC) module was utilized to supply power to the Garmin Lidar Lite V3 sensor.

Figure XI. Turnigy 5000 mAh 3S LiPo Battery

V. Weight and Cost Analysis

A. Weight Analysis

The total weight of the final system is a crucial factor in the failure or success of an aerial robot. In Table 1 below, a weight analysis of the entire drone system was done to ensure that all chosen components did not exceed the thrust the motors are able to produce (3,578 g). It was seen that our total weight of the multirotor was 1,789 grams in weight satisfying the 2 rule to ensure the robot can fly. In keeping the total weight close to half the total thrust, the system is able to maximize the battery life and extend the mission duration.

Table I. Weight Analysis of the Drone

Item	Unit Weight (g)	Quantity	Total Weight (g)
Drone Frame	405	1	405
Multistar 2216 Motors	83	4	332
9047 Propellers	6	4	24
Turnigy 25A ESC	22	4	88
Turnigy 5000mAh Battery	360	1	360
3DR Telemetry Module	10	1	10
Pixhawk 2.0 Cube	19	1	19
NVIDIA TK1 Board	498	1	498
Garmin Lidar Lite V3	16	1	16
PX4 Optical Flow	5	1	5
USB Camera	25	1	25
Power Module	3	1	3
5V 3A BEC Module	3	1	3
Total Weight: 1,789 Grams			

B. Cost Analysis

Taking the total cost into account, the drone costed only $907.00. Most of the cost was the result of the Pixhawk 2.1 Cube flight controller, NVIDIA TK1 Developer boar, Garmin Lidar Lite, and PX4 FLOW Optical Flow. In eliminating those components, the cost of the basic drone necessities totaled to be $238.95. The Taranis X9D transmitter was not considered as it is possible to perform manual override and kill switch through the computer used to monitor the drone activity. In analyzing the cost, the multirotor built for this project remains relatively low cost.

Table II. Cost Analysis of the Drone

Item	Price	Quantity	Total
Drone Frame	$24.15	1	$24.15
Multistar 2216 Motors	$17.60	4	$70.40
9047 Propellers	$1.89	4	$7.56
Turnigy 25A ESC	$13.83	4	$55.32
Turnigy 5000mAh Battery	$26.55	1	$26.55
3DR Telemetry Module	$19.19	1	$19.19
Pixhawk 2.0 Cube	$238.00	1	$238.00
NVIDIA TK1 Board	$191.00	1	$191.00
Garmin Lidar Lite V3	$129.99	1	$129.99
PX4 Optical Flow	$109.96	1	$109.96
USB Camera	$14.99	1	$14.99
Power Module	$8.99	1	$8.99
5V 3A BEC Module	$12.99	1	$12.99
Total Weight: $907.00			

VI. Autonomous Movement

A. Flight Stack

Pixhawk is compatible to use with ArduPilot flight stack and PX4 flight stack. Both flight stacks have their pros and cons. ArduPilot firmware is tested by developers and then made available for the user to use and upload to their flight controller. They also have master firmware which is available for developers to work on but is not recommended for normal users. PX4 on the other hand is built and maintained for the developers. This firmware has more features compared to the ArduPilot and supports more advanced algorithms for use with companion computers. The builds are unstable compared to the ArduPilot flight stack and in PX4 firmware the users are the developers so, the developers test fly using the code and report any issues if any found. For application of, Autonomous indoor quadcopter, ArduPilot flight stack is used as it is managed by developers and the builds are more stable compared to PX4.

Figure XII. ArduPilot vs PX4 Flight Stack

B. MAVLink

MAVLink (Micro Aerial Vehicle Link) is a communication protocol used for communication with unmanned vehicles. It is designed as a header only marshaling library. The ArduPilot firmware supports MAVLink messages which is used to communicate with the unmanned vehicles to get data to the ground control station such as Mission Planner or QGroundControl. MAVLink is a binary telemetry protocol designed for resource-constrained systems and bandwidth-constrained links. Telemetry data streams are sent in a multicast design while protocol aspects that change the system configuration and require guaranteed delivery like the mission protocol or parameter protocol are point-to-point with retransmission. This makes the protocol to be reliable and useable for autonomous vehicles.

Figure XIII. MAVLink Communication Protocol

MAVLink has been deployed in two major versions - 1.0 and 2.0. MAVLink 1.0 was adopted by many developers for multiple applications. The transition from version 1.0 to version 2.0 is very smooth as in version 2.0 C/C++ and Python are backwards compatible. In the application for the autonomous quadcopter MAVLink 2.0 is being used. To use the MAVLink commands, MAVLink has a version handshake check which checks if the ArduPilot version on the quadcopter is version 2.0 or version and an acknowledge command is sent back to the companion computer. If the acknowledge command is not sent back means the ArduPilot version cannot send or receive messages until upgraded to the compatible version. This version check performed by a Ground Control station (GCS) is illustrated in the following figure[10]:

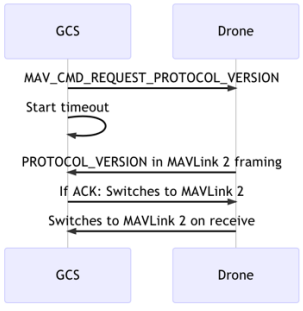

Figure XIV. Version Check by MAVLink

C. DroneKit-Python

DroneKit-Python is an app developed for onboard companion computers to send commands to the flight controller such as Pixhawk for maneuvering. Using onboard computer can increase the amount of stuff that can be done on the quadcopter as it can send commands based on time or location without user input. This allows the quadcopter to have autonomous flight capabilities which in turn can be very useful to eliminate human input errors. Companion computers can be very helpful for applications like - computer vision, path planning, or 3D modelling. All these tasks require intense computing power that the flight controller alone cannot provide. This is the reason why the companion computer is used alongside the flight controller. This way the flight controller can manage the flight computing for Proportional Integral and Derivative (PID) controls and Extended Kalman Filter (EKF) algorithms for a stable flight while the companion computer can focus on the other tasks that require higher computing power. DroneKit is run on Linux as the OS on the companion computer. DroneKit supports the previously mentioned MAVLink commands to communicate with the flight controller to make the drone autonomous. MAVLink message called "SET_POSITION_TARGET_LOCAL_NED"[9] for maneuvers. This message can be called to send position, velocity and acceleration commands in x, y, and z direction. The user can select from different coordinate frames to perform movements relative to the coordinate frame the drone had when it was turned on or based on North East Down (NED) frame. In this case, "MAV_FRAME_BODY_NED" was used as the coordinate frame, so that the movements are relative to the heading of the quadcopter when turned on.[8] A simple take off script is shown here.

```
#!/usr/bin/env python
from dronekit import connect, VehicleMode, LocationGlobal, LocationGlobalRelative, Rangefinder
from pymavlink import mavutil
import time
import math
vehicle = connect('/dev/ttyACM0', wait_ready=True)


def arm_and_takeoff(TargetAltitude):

    print "Arming motors"

    vehicle.mode    = VehicleMode("GUIDED")
    vehicle.armed   = True

    while not vehicle.armed:
        print " Waiting for arming..."
        time.sleep(1)

    print "Taking off!"
    vehicle.simple_takeoff(TargetAltitude) # Take off to target altitude

    while True:
        print " Altitude: ", vehicle.location.global_relative_frame.alt

        if vehicle.location.global_relative_frame.alt>=TargetAltitude*0.95:
            print "Reached target altitude"
            break
        time.sleep(1)

arm_and_takeoff(1)
```

Figure XV. Simple Take-off Script

VII. Target Detection

A. Single Board Computer

The single board computer used is NVIDIA TK1 which runs Linux4Tegra OS which is a version of Ubuntu with pre-configured drivers. The NVIDIA TK1's Kepler GPU with its 192 CUDA cores can perform significantly better compared to Raspberry Pi and Odroid XU4. Using the CUDA cores combined with OpenCV's optimized libraries for image processing makes target recognition very quick and accurate.

B. Target Recognition

Target detection for the drone is being handled by OpenCV computer vision libraries. The target detection algorithm uses features picked up from the ground to detect if the targets are present in the frame of the downward facing camera on the drone. The targets will be identified using recognition of different shapes present on the home base, drop-off, and pick-up location. The OpenCV libraries will be used to differentiate between the targets by using multiple shape detection. The target detection algorithm bases different shapes detection on the difference between features of the image. The algorithm finds the different contours present on the ground and tries to approximate the curve using the least vertices possible. This results in the algorithm finding contours and shape with relatively high accuracy. The algorithm is further improved by running a color filter on the image and actively looking for different colors. Coupled with the color filter and the shape detection, the drone can find the target accurately and can plot the center point on a grid based on the current frame. To aid in determining different targets, color filters are used to differentiate between the different targets. For the search algorithm, if the quadcopter does not find any target in the camera frame, it will keep searching for the targets. Once the target is found the drone will try to align the X and Y coordinates of the frame with the X and Y coordinates of the detected target to perform necessary task.

VIII. Applications

The application of target detection and tracking has become the fundamentals of all current drone applications. The indoor autonomous system can be applied to search and rescue operations inside buildings of a natural disaster that are too dangerous for human entry. Furthermore, in locations where GPS signal is unobtainable the multirotor will still be able to function and perform the necessary missions. Thus, aiding researchers in harsh environments.

IX. Conclusion

The paper presented the development process of a drone capable of autonomous navigation and target detection and tracking in an indoor environment without GPS. In doing so a low cost, efficient drone system was designed capable of efficiently finding targets with an autonomous flight to eliminate user input error and be more efficient. The use of MAVLink messages makes the communication more efficient and reliable when performing an autonomous flight. The use of NVIDIA TK1 makes the processing of the target recognition faster than other low-price companion computer. In conclusion, for a quadcopter to have these autonomous capabilities without having motion capture cameras or GPS requires multiple sensors for localization and perform the same tasks.

Acknowledgements

The authors would like to thank Vaughn College of Aeronautics and Technology and the Department Chair of Engineering and Technology, Dr. Hossein Rahemi, for their enthusiastic efforts to consistently support the Unmanned Aerial Vehicles Club. Furthermore, we would like to thank the UAV Club faculty advisor, Dr. Amir Elzawawy for his support and guidance.

References

1. https://www.ethz.ch/content/dam/ethz/special-interest/mavt/robotics-n-intelligent-systems/asl-dam/documents/lectures/autonomous_mobile_robots/spring2016, Accessed by January 2018
2. https://hobbyking.com/media/file/288905599X28019X33.pdf , Accessed January 2018.
3. http://ardupilot.org/copter/docs/common-sik-telemetry-radio.html, Accessed by February 2018
4. https://pixhawk.org/modules/pixhawk2 Accessed by February 2018
5. https://elinux.org/Jetson_TK1 Accessed by February 2018
6. https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf Accessed by February 2018
7. https://dev.px4.io/en/tutorials/optical_flow.html Accessed by February 2018
8. http://mavlink.org/messages/common Accessed by March 2018
9. http://python.dronekit.io/guide/copter/guided_mode.html#guided-mode-copter-velocity-control Accessed by March 2018
10. https://mavlink.io/en/mavlink_version.html Accessed March 2018