Fiducial Pattern Tracking Drone

Team 59 - Angelos Guan, Muhammad Umer Belagam, Alexander Amiel
ECE 445 Project Proposal - Spring 2021
TA: Chaitanya Sindagi

1. Introduction

1.1 objective

Many drones use a controller, like a game controller, to fly it. This is usually enabled with one's phone to keep track of location and to be used as a camera to see where the drone is looking. These drones are limited by the range of electromagnetic radiation and susceptible to interface noise. What happens if the phone dies? If someone gets a call/notification, this can exit the camera mode from the phone/controller. What if the drone is locked onto a person, but it doesn't know which one to focus on? The problem with many of the “best” drones also reside in the kind of phone you have. They require the user to have an app downloaded from their smartphone to be used as a camera, and used as a way to make the drone do specific actions. These drones are also relatively expensive and many people don't want to spend the capital on buying this device especially if they aren't going to use it that often.

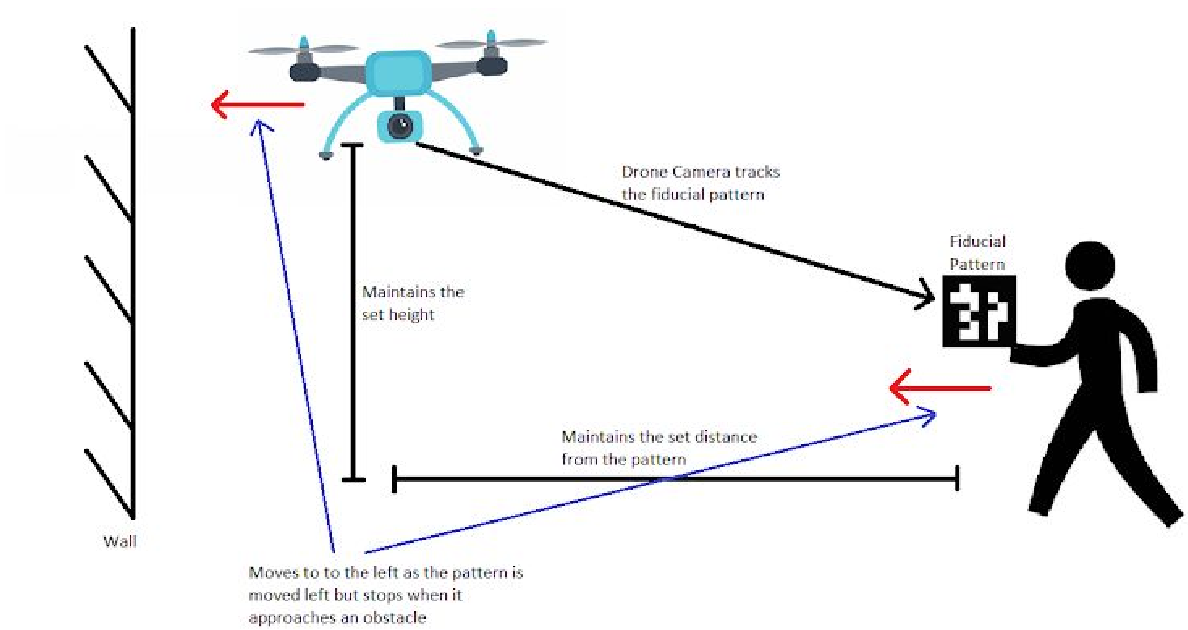
Our goal is to bring a drone capable of detecting a certain fiducial pattern to pilot a certain way. Our system is almost entirely hands free with the fact that you only have to raise a sheet of paper that has a fiducial pattern on it. A fiducial pattern is like a QR code pattern except it is not as complex yet it still has many sequences that can be generated. We will use a raspberry pi connected to a camera module to recognize the pattern, and communicate data to the stm32 microcontroller to signal the drone to do a specific action based on the fiducial pattern.

1.2 Background

DJI's spark and Samsung have come up with alternate ways to pilot a drone through the use of hand gestures. MIT's artificial intelligence lab uses a muscular-control system to pilot the drone. These also present many limitations in piloting a drone. The hand gestured controlled drones present a problem in where it can detect someone else's hand, and the drone wouldn't function the way it was supposed to. Detecting a gesture also results in many errors in piloting a drone. In MIT's perspective, what if the person piloting the drone has some sort of muscular breakdown or “accidentally” squeezes too hard or too soft?

Our solution presents a way for the drone to recognize a specific pattern and act in a way it is told to. If a pattern isn't recognized, it would signal the user through a buzzer and LED. This helps solve the problem in a hands free approach to pilot a drone. Since the fiducial patterns are all distinct, this is a good replacement for hand gestures.

1.3 Physical Design



1.4 High-level requirements list

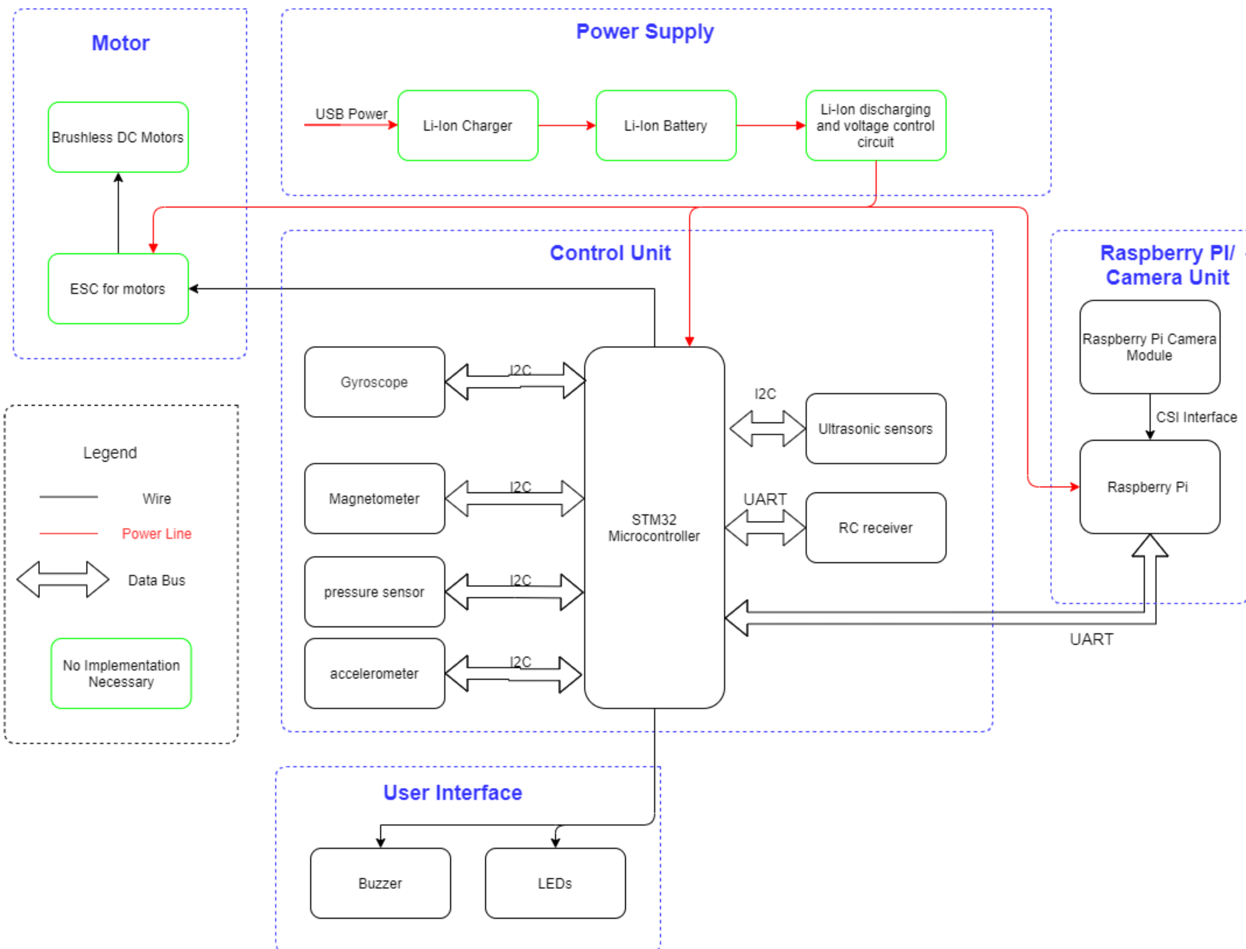
1.4.1: Our project must be able to use a camera, mounted to a drone, to track a fixed sized QR code from a certain distance.

1.4.2: The drone must be able to fly stably according to commands based on our flight controller built from the PCB we design.

1.4.3: Our drone must be able to fly safely by maintaining obstacle avoidance with the use of ultrasonic sensors.

2. Design

Block Diagram



2.1 Power Supply

The power supply is responsible for keeping the drone operational at all times. The power supply voltage control is responsible for a regulated 3.3V output to the control unit and 5.1V to the raspberry pi. It has an over discharge protection for the li-ion batteries and it powers the ESC with the protected output. We won't be building the power supply or motor from scratch, rather we are buying an already built power supply and ESC motor.

2.1.1 Li-ion Charger

A li-ion battery charger that can boost a 5v input voltage to that appropriate for charging the batteries in a 3S configuration. Should have constant current and constant voltage capabilities and follow the charge profile for the IMR type li-ion batteries we will be using. A battery management system (BMS) is also required to ensure all 3 cells have the same voltage as they are charged and the BMS would also provide overcharge protection.

2.1.2 Li-ion Battery

Three 3.7V Li-ion batteries of IMR type chemistry in series to have enough voltage to power the ESC motors. The voltage control unit will regulate the voltage from the Li-ion batteries to also power the control unit and raspberry pi at their optimal voltages.

Requirement: the battery operates at 3.7-4.2V where 3 in series can supply enough voltage to power the ESC motors at a max of 12.6V. Provide 20A continuous and 35A peak current.

2.1.3 Li-ion discharging and voltage control circuit

Overdischarge and short circuit protection circuit for the li-ion batteries to prevent unsafe conditions. Two buck convertors with 3.3v+-5% and 5v+-5% output to step down the battery voltage for powering the control unit and raspberry pi. Both buck convertors should be able to output 500mA at 90%+ efficiency.

2.2 Raspberry Pi/Camera Unit

The raspberry pi/camera unit recognizes the fiducial patterns and prepares data to be sent to the stm32 microcontroller through UART RX/TX ports.

2.2.1 Raspberry Pi Camera Module

The camera module is connected to the raspberry pi through the CSI interface. It is responsible for capturing the fiducial pattern the user shows to the camera. The camera module is light, 3 grams, and can support resolution up to 1080p30, 720p60 and 640x480p90.

Requirement 1: Camera Module must be light, 3 grams
Requirement 2: Must be small with size 25x24x9 mm
Requirement 3: should have enough resolution for recognizing the fiducial patterns
Requirement 4: should have enough frame rate for tracking an object

2.2.2 Raspberry Pi Zero

The raspberry pi zero is responsible for taking the fiducial pattern, recognized from the camera module connected through the CSI interface, and sending data through the UART to the stm32 microcontroller. Each fiducial pattern encodes a different function that the raspberry pi has to recognize. The stm32 microcontroller is signaled to make the drone do a specific action.

Requirement 1: must be provided 5V from the power supply
Requirement 2: small in size, 65mmx30mmx5mm, and lightweight, 9 grams
Requirement 3: must have the CSI interface to connect to the camera module

Requirement 4: must be able to run openCV in real time.

2.3 Flight Controller/Control Unit

2.3.1 Microcontroller

We will be using a microcontroller of the STM32F4xx family which is based on the Arm Cortex M4 core. This chip would be responsible for processing the data from the various sensors and use a PID algorithm to send the appropriate commands to the ESC for the motors.

Requirement 1: A floating point unit to make fast calculations on the sensor data.
Requirement 2: An I2C interface to be able to communicate with all the sensors.
Requirement 3: A UART interface to accept commands from the raspberry pi.
Requirement 4: Should have enough memory to support processing sensor data and necessary computations

2.3.2 Gyroscope

Measure angular velocity .

Requirement 1: Should communicate using an I2C interface.
Requirement 2: Must measure angular velocity in 3 axes.

2.3.3 Accelerometer

Measure the acceleration experienced by the drone during flight to figure out its orientation with respect to the gravity vector and the change in velocity when its moving for use in the PID algorithm used to control the flight.

Requirement 1: Should communicate using an I2C interface.
Requirement 2: Detect accelerations in range +-2g.
Requirement 3: Must measure accelerations in 3 axes.

2.3.4 Magnetometer

Detect the magnetic north vector in 3 dimensions to determine its orientation with respect to earth's magnetic field. Would be used to add extra redundancy to the noisy accelerometer and gyroscope data for the PID calculations for better estimation of position and orientation.
Requirement 1: Should communicate using an I2C interface.

2.3.2 Pressure sensor

Measure the atmospheric pressure the drone experiences at its current position. This will be used to determine what altitude the drone is at so it can maintain its distance from the ground.

Requirement 1: Should communicate using an I2C interface.

2.4 Motor Unit

2.4.1 ESC

Used to control the speed of the motors to maneuver the drone.

Requirement 1: Abled to be powered by a 3S configuration of Li-ion batteries and be able to draw 20A continuous and peak 35A current.

2.4.2 Motors

We will be using brushless dc motors because of their higher efficiency compared to brushed DC motors as we have a limited battery capacity we need to maximize flight time.

2.5 User Interface

2.5.1 LED strips

We will be using a LED strip to indicate the status of the drone including fiducial pattern not detected, battery low, too close to obstacle or ground to the user.

Requirement 1: Programmable or can present 3 different types of pattern

2.5.2 Buzzer

We will be using a buzzer to warn the user when the fiducial pattern is not detected.

Requirement 1: Must be audible from a distance of at least 30ft.

2.6 Risk Analysis

A big risk we have in our design choice is choosing the right stm32 microcontroller family vs an arduino uno or something similar. The specific microcontroller must have the ability to process the signals sent from the raspberry pi and communicate with the ESC motors in a small time frame. It must also have enough RAM to have the ability to perform these actions, so that there are no limitations in what controls we can send to the motors/drone. This is why we are using a microcontroller of the STM32F4xx family. The F4 refers to the Cortex-M4 core which has a FPU (floating point Unit). The data from the sensors would be in decimal numbers and we would require an FPU to process these values as floating points in hardware without having to simulate the floating point numbers in software. If there is no FPU option, we would possibly run out of RAM, and would not have enough CPU clock cycles to process it as fast as it needs to be used to control the motors. The sensors should also communicate with the I2C interface, so having the correct number of those on the microcontroller is crucial in picking the correct microcontroller.

Since we are piloting a drone, we must limit the extra weight we put upon it from the pieces we buy. This limits what we can attach to the drone since the drone we are buying is a cheap, easy to build model that is inexpensive. The model we buy must be inexpensive, and must be able to withstand extra weight we put upon it. The heaviest parts would be the ultrasonic sensors, the power supply, and the camera module attached to the raspberry pi. We chose the raspberry pi zero because it has dimensions of 1.18 inches X 2.55 inches and weighing 9 grams while the raspberry pi 3's dimensions are 2.22 inches X 3.37 inches and weighing 45 grams. The extra specs of the raspberry pi 3 isn't worth the extra weight, so implementing the drone with the raspberry pi zero with the camera module holds a significant decrease in weight. The weight of the ultrasonic sensors are 8.5 grams each and having four, to accommodate for each side of the drone, adds extra necessary weight in order for the drone to not crash/run into any obstacles. Especially considering taking off and landing, it is crucial that we have these sensors attached to all four sides.

The design of our PCB also holds risk in communicating each sensor's data to the microcontroller. We must also have enough I2C ports on our PCB that can attach to the microcontroller. This also relates to our microcontroller choice we buy, so that it is able to use floating point unit to process the data fast. If this isn't done, the drone's ability to fly will be greatly hindered and "lag" will be produced increasing the chance of a crash.

3. Ethics and Safety

Since the goal of our project is to implement a new approach for the user to control the drone, we assume that the action of the drone under ethical evaluation is based on the action of the person controlling the drone. The most relevant codes of ethics for this project are IEEE Code of Ethics #1 and #9, which stresses the importance of ensuring the safety of the public and their properties. Both the drone and its operator must follow the IEEE Code of Ethics and develop safety protocols in case of failures such as communication failure, drone is out of battery or controller malfunction. For the purpose of this project, we assume that the usage of the drone is recreational. Thus, under Federal Aviation Administration's (FAA) policies on Recreational Flyers, our project can only be flew at or below 400 feet above the ground when in uncontrolled airspace. In addition, under university policy Fo-05, in order to test our project on campus, we must obtain approval from the Division of Public Safety by submitting a request to dpscomments@illinois.edu. If the aerial testing happens inside campus property, we must obtain approval from Code Compliance & Fire Safety as well as from faculty.

Usage of Li-Ion batteries also raises several safety concerns. First, they need to be stored at room temperature and not under direct sunlight or other heat sources. Second, Li-Ion battery cells should never be stored fully charged. Third, a battery protection circuit is required to control overvoltage, undervoltage and current surges.

Due to the high current applied to the motors, we also need to follow the safety rules regarding lab fire safety and electrical safety.

References

a. https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/1
b. R_Pi_Camera_Module_Specs
c. Raspberry_Pi_Connection_with_stm32
d. ESC_Motor_Spec_Reqs
e. Best_drones
f. Samsung_Hand_movement_controlled_drone
g. DJI_spark_hand_movement_controlled_drone
h. Fiducial_pattern_video
i. RaspberryPiZero_VS_RaspberryPi3
j. LiIon Safety Precautions
k. https://cam.illinois.edu/policies/fo-05/