UARC

Senior Design Project

Spring 2009

Group #9
Clint Mansfield
Edwin Giraldo
Jeremy Brooks

Table of Contents

Chapter 1: Executive Summary

Sophisticated machines are able to do complex tasks that humans would take prolonged hours to accomplish. This efficiency has led to a dependency on electronics to convenience man every day. When air flight took place, many different solutions were created. Transporting of goods and materials, vacations and business negotiations, and protecting civil liberties by military personnel. However, they are only useful as humans control them. Humans differ from machines in one aspect; human error. Table 1 shows data gathered on a 10-year sample of civil and commercial aviation accidents that have occurred. More importantly, this also includes casualties derived from the aviation accident itself. As one can see, many casualties have occurred due to human error or intentionally.

	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
Aviation casualties	1,169	1,429	1,396	1,437	1,275	728	1,306	1,136	931	817	137
Extra casualties on ground due to accident	36	16	5,641	167	24	2	59	4	57	60	1

Table 1: Shows aviation casualties of both civil and commercial flight. Moreover, it also includes humans lost on ground due to the accident itself. Note the casualties on ground for year 2001 which was due to the 9-11 terrorist attacks in NYC. Source provided by Richard Kebabjian.

In time of combat, the military uses aviation to transport goods, plan warfare, and move personnel. In time of war, combatants are placed in danger due to cross-fire or retaliation. The need of protecting civilians has always been of best interest by the government. Because money is of no concern when it comes to military funding, the best of the best is often considered. From binoculars, to spy bots, to aerial flight, less and less human interaction is necessary. The development of self guided systems has been proven a success in unmanned vehicles. Scout robots enter an unknown terrain and feeds information back to a user to interpret the data. With the use of a robotic arm, it is able to deploy tear gas or such to disable armed rebels. The robots sacrifice puts humans away from any danger that could result deadly. The same idea has taken afoot for flight in which aerial vehicles are guided on land while the only loss is of scrap of metal.

This idea of course is not military specific. Many companies have adopted flying vehicles for the use of business such as photography, weather predicting, and hobby-enthusiasts. Airborne Video Services specializes in low-altitude photography to acquire images where full scale aircrafts are not able to produce. This is a one of many ideas that has emerged recently. DARPA, sponsored by the Department of Defense, sponsors research projects to challenge schools and organizations to develop innovative solutions not presently found. In 2007, DARPA's URBAN CHALLENGE required teams to build autonomous vehicles capable of driving in traffic doing maneuvers such as merging, passing, and parking.

Chapter 2: Definitions and Goals

2.1 Motivation

2.1.1 Motivation

The motivation came from the desire to make an unmanned vehicle that was going to help something/someone/society in general. After much thought, Edwin proposed the idea of an underwater device. Soon the thought of propulsion, water-resistant parts, and a charging device that can all be relatively feasible was not going to be easy in the timeframe allotted. One day walking out of class, the group discovered a 'lost helicopter' flyer by a vending machine. The helicopter had similar features as those wanted in the underwater device. It was then that the group found the project they wanted to take on for research.

At first the project seemed relatively simple until the price tag on such vehicle was discovered. Similar devices found in hobby shops ranged from $600 - $18,000. It should be noted the lower price tag range was remote-controlled. The realization set in that a project was taken on that few did. As engineers though, the thought of producing something for a lower price is what made the group stick to this project. Nonetheless, there was a reason for such dramatic price tag on this unmanned vehicle.

The group was placed at ease when a small research effort was made to see what components are used to develop one. Several blogs suggested as much as 11 sensors, a microcontroller, and lightweight material is sufficient to make your very own helicopter. Much of this effort will mimic the Draganflyer, a remote-controlled aerial vehicle that uses state-of-art components to stabilize itself and has a mounted camera to stream video to a nearby computer.

2.1.2 Similar Projects

The internet was crowded with similar designs of unmanned vehicles. Most resources though tried to replicate the Draganflyer, or alter its current design. The first type of project included simple design features that made it just hover and used simple sensors to get the job done. The second type of project included various sensors and lightweight components to mimic the Draganflyers entry level helicopter they sell. Lastly, the third project found was a design where it outfitted the best that can be found in sensors, lightweight components, and complex algorithms to mimic the Draganflyers most expensive model. What makes the Draganflyer unique in replicating is the fact that the company will sell each piece individually to suit how much of the helicopter is wanted to mimic. While this is not the only model that is found out there to replicate, it seemed to be the most popular by fanatics and had vast information that can be found about people who had success in altering different components and specs for people wanting to copy the design. Figure 1 shows the Draganflyer V-ti, the entry level helicopter that can be purchased. The groups integrated a camera that will be mounted for video streaming and use relatively sophisticated programming for flight response.

Figure 1: The Draganflyer V-ti can be purchased for $1,049.95.
Courtesy of Draganfly Innovations.

The latest version of the Draganflyer encompasses a tripod design in which it uses 3 props instead of 4. However, the dynamics of using this design is out of the scope of what to accomplish since it involves aerospace engineering due to the torques the engines would see. For this reason the group decided to mesh some of the concepts found at Draganflyer Innovations from the entry-level, intermediate, and expert level helicopters.

2.2 Goals and Objectives

2.2.1 Mission of Unmanned Aerial Vehicles

There are different functions of Unmanned Aerial Vehicles that have been developed, such as: remote sensing, transport, scientific research, precision strikes, and search and rescue. UAV's are seen in biological sensors capable of detecting airborne presence of organisms as remote sensing. UAV's carry small payloads that can be taken from one point to the other are used in transport. NOAA (National Oceanic and Atmospheric Administration) uses UAV's to monitor hurricanes' real-time barometric and temperature data when one is found in the ocean as part of the scientific research classification. UAV's are now armed with missiles to hit ground targets in sensitive areas for precision strikes. Finally, search and rescue missions allow reconnaissance of photographic images relayed back to ground forces to be used as a tactical weapon as used by the military groups. A simple set of instructions can be issued to a helicopter and it can decide on its own how to gather that information and return back to base. The goal was to develop an idea that can help save lives to avoid casualties as those found in war for unfriendly fire. Much of reconnaissance deals with unknown terrain and with the help of a 'birds-eye-view' can assist soldiers in finishing a missions and avoid unnecessary deaths of soldiers.

The group had a vision that this vehicle will take on its own decision-making using sensors to be able to hover and navigate. The goal was to be able to develop a lightweight vehicle that will be able to carry twice the payload than its own weight. This is ideal as the function of creating an unmanned vehicle is to carry instruments for efficiency. Due to the nature of creating lightweight, the electronics will not be power hungry devices and will contain energy saving components. While mimicking much of the Draganflyers' features, this application will be targeted for the military. With this in mind, a camera can be mounted for video streaming as an application to assist a soldier learn his surroundings. It will take objectives from a user using wireless technology to onboard electronics to carry out the task with the use of GPS navigation. Simple sensors will assist orientation and stability to stay afloat in air. Once a mission is accomplished, return to home base without the use of human input. Recon missions can be implemented to show the true complexity of this project and the multiple uses it can bring. Random points of interest can be coded to arrive at a location and capture images and relay them back to home base.

While the concept of this project is easy, most of the beauty of this vehicle will be its feedback control mechanism. It will be able to interact with sensors to keep stabilization and be able to navigate. Feedback controls can be designed to have a user input a mission, change courses, or manual override for in case-of-emergency scenarios. It can display coordinates, and provide real-time video streaming. The group wanted to see this device use minimal user input and have the vehicle do much of the interaction.

2.2.2 Safety

One of the most important features of UARC is the possibility to prevent casualty. Human loss is a horrible yet unavoidable as part of war. Reconnaissance is necessary to retrieve vital information about an enemies force, but can be very risky. UARC will minimize this risk greatly by flying autonomously into the danger zone and sending back information to the safety of home.

2.2.3 Take off and Hovering

Take off and hovering was one of the most challenging yet crucial parts of obtaining autonomous flight with UARC. The goal was to lift off from ground and achieve a stable hover at a certain desired height. This was done by having all four motors reach an equal rpm producing a thrust greater than the weight of UARC then decreasing the thrust to the weight when the height is reached. This is shown below in figure 2 below.

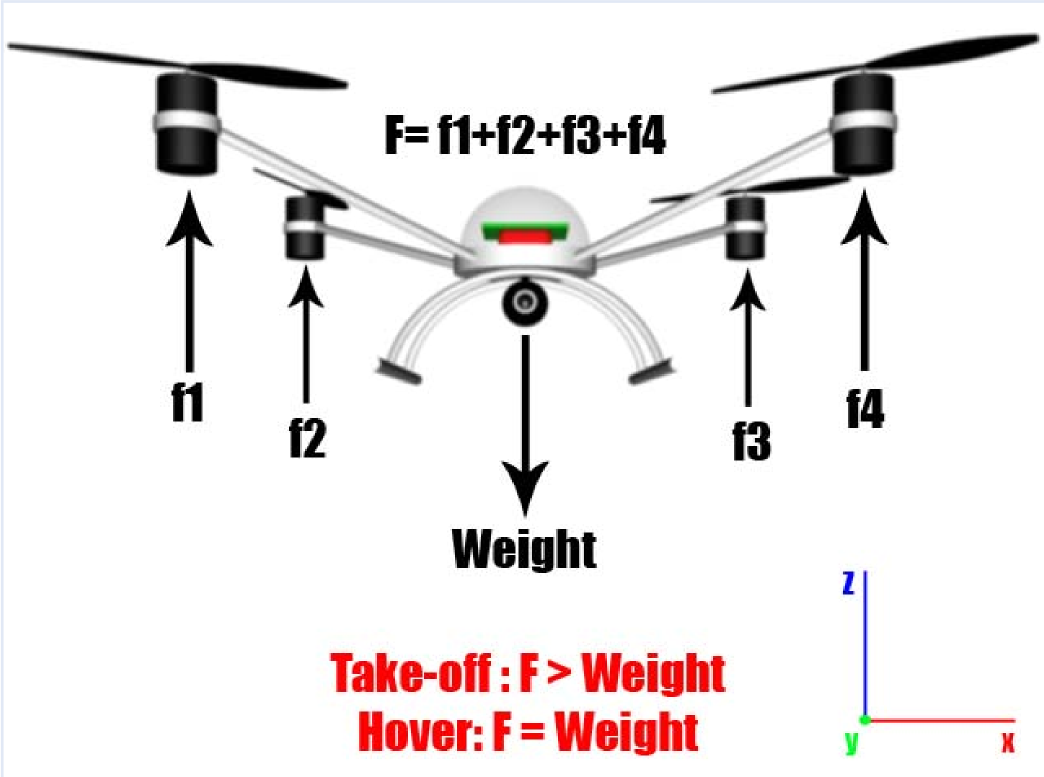


Figure 2: Thrust requirements for take-off and hover.

Take off entailed the sequential starting of the motors to prevent overload on the battery. This effect was noticed during preliminary testing of the motors. Although all the motors are of the same brand and same specifications, they will not have the same internal mechanics. Testing was done to determine the thrust versus voltage ratio for each individual motor. The control system will then have to adjust for the differences during flight.

While hovering sounds simple enough, it will be a huge step in the progress of flight stability. The control systems will be adjusting the speeds of the motors frequently causing the vehicle to drift in all directions. Staying in one area is difficult due to drift in the sensor readings, outside disturbances, and other nuisances of the like.

2.2.4 Feedback Controls

The UARC control systems will be the link between hardware and software interaction. For a vehicle of this type it will take many calculations done fast and as efficiently as possible. It required a feedback system that compares the real time signal coordinates, from the sensors, and the desired coordinates from user input. Depending on these calculations it will adjust the motor speeds accordingly to achieve maneuverability. This was basically a cascade of controllers consisting of translational and rotational blocks comprising the plant as figure 3 demonstrates.

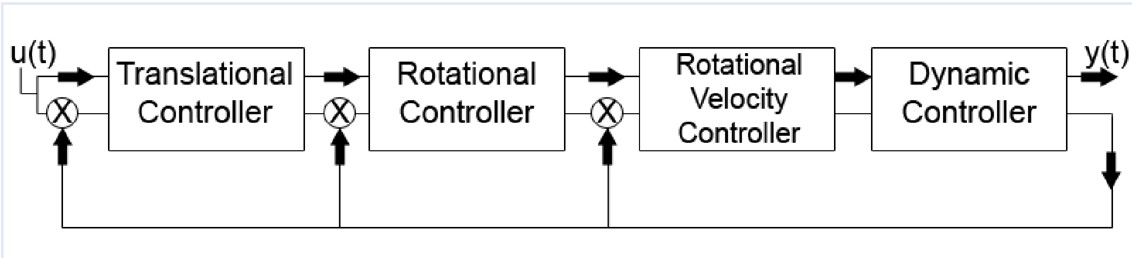


Figure 3: The Plant Control System

As one may notice, the rotational velocity controller is the inner loop since it's the faster dynamics while the translational controller is the outer loop due to the slower dynamics of position. This will maximize efficiency, speed, and accuracy. Another method of improving accuracy is the implementation of a PD (Proportional Derivative) compensator. This is a very powerful and relatively easy method of compensating for dynamic error. Figure 4 below shows the block diagram of the desired control system, where $y_d(t)$ is the desired input.

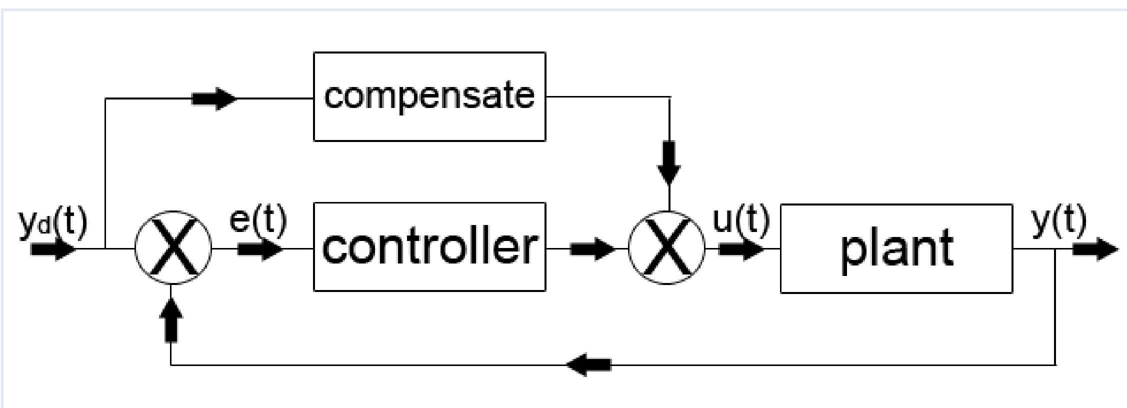


Figure 4: Implementation of PD compensator on system

There are many factors to determine in the design of this system. It was left up to simulation and real time testing to establish the coefficients needed to implement the best design. The transfer functions for each block was derived and tested before any legitimate control was established. With the aid of Matlab simulation, the testing of various situations and coefficients was easy and a rewarding task.

2.2.5 Wireless Communication

The UARC needed to communicate wirelessly with a laptop computer. Due to the nature of a moving helicopter, the only logical choice is to use wireless technology versus tethered wires. There are several methods of communicating wirelessly and each one was explored to find the appropriate one for this application.

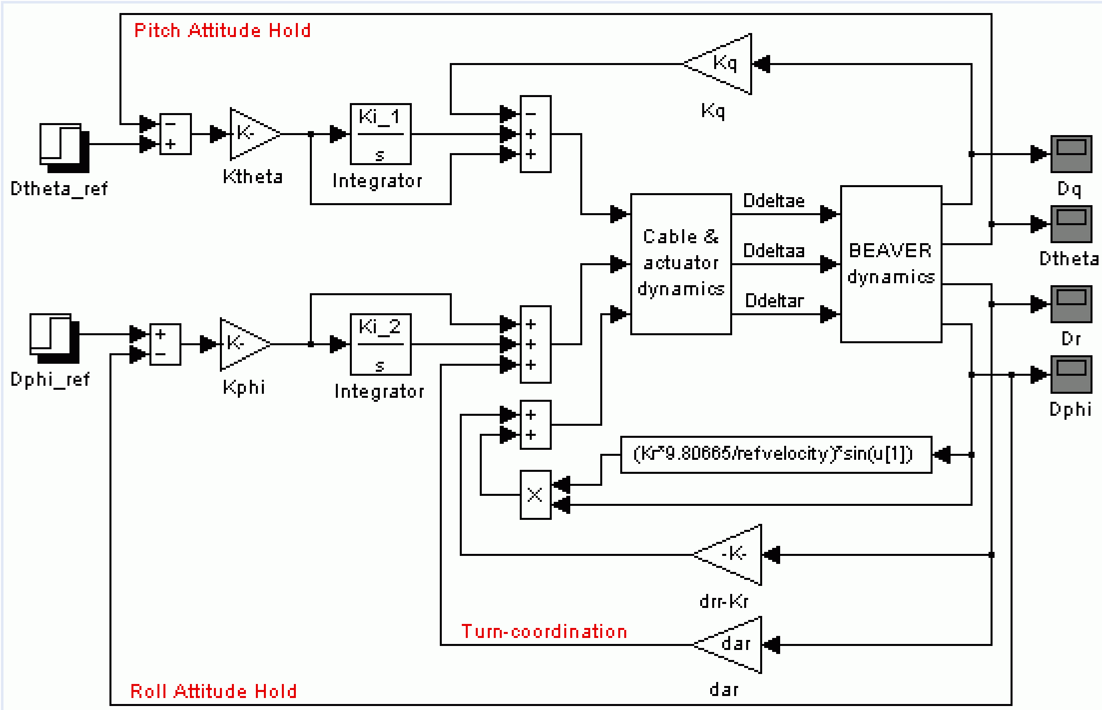


Figure 5: Matlab can assist in simulating as seen above in this custom toolbox that enables flight simulation. Courtesy of Mathworks.

Most UAV's use GPS signals to navigate and the UARC could of used the same technology. Signals received from a GPS module can be sent to the computer giving real-time location. This will serve not only as a form of tracking, but also the signal can be used by a microcontroller to stay within a set coordinate area, or to move to an exact coordinate location. Ideally this can be sent to the UARC via the laptop, or the information can be pre-programmed into the microcontroller and changed using a simple application specific GUI. The UARC has video capability that will need to be sent wirelessly to an interface as well. Most UAV's have some form of video communication that's used for reconnaissance missions or to aide in the delivering of payload. This was done by using system integration to keep costs down and allowed the group to spend their time focusing on the critical functions like flight stability and control. Safety is paramount in a project like this because although the UARC is small and uses small DC electric motors, it could still do damage if the blades hit a person or other object. Being that it had no manual control, the group implemented a program that will quickly force the UARC to descend and land automatically if a certain signal is received wirelessly. This same program was used by the microcontroller if and when the battery voltage drops to very low levels. This idea will be explained and further developed later in this paper.

2.2.6 Navigation

Navigation is essential for any autonomous vehicle. Most rely on a variety of sensors for positioning, geographic location, obstacle avoidance, tilt angle, etc. The information gathered from the sensors was filtered and used by the hardware and software to make decisions. The UARC needed a variety of sensors to navigate in 3-D space. There were many varieties that were explored to find the appropriate ones for this application.

Gyros may be used for flight stability. These will measure angular rotation around a single axis and provide information to the hardware to adjust the motors to slow or dampen the rotation. This proved useful during hovering, for example. Gyros are great by themselves but can't be used alone since they only output non-zero values when the aircraft is rotating. When it stops rotating, the gyros will output a zero value or the same as if the aircraft were flying straight and level, even though it could be flying sideways. An accelerometer should be used to measure tilt angle relative to the surface of the earth. Accelerometers measure both static and dynamic acceleration. Like gyros, they measure tilt along a single axis. However there are some inherent problems. If the aircraft is in free fall, the accelerometer would measure zero acceleration. Also any acceleration that's measured is added to the acceleration of gravity, making it hard to distinguish what accelerations are acting on the aircraft, and in what direction, when it's moving. This problem makes it difficult to use accelerometers alone.

Another way to stabilize the UARC during hovering and flight maneuvers is to use infrared horizon sensing. This technology uses temperature differences between the cold sky and the warm ground to keep the aircraft level. This would be especially useful during hovering. A GPS module is another useful navigation sensor on a UAV. This is really the most important component when conducting a mission, allowing it to find the target and return home. The UARC could use information provided by the GPS to stay within a certain area, or conduct a mission then return to its home location. Also any information gathered during the mission would include the time and location of where it was obtained. A magnetometer measures the intensity and direction of a magnetic field. It can be used as an electronic compass in a UAV or autonomous vehicle. This could help the UAV know which direction it's moving and could prove useful while locating a target or returning home. Since the UARC was only flying and hovering just feet above ground, it can measure altitude with a simple distance sensor. Ideally, the flying height could be changed from the laptop. It's important that this sensor operate correctly and accurately. This is something the group kept in mind when they chose a sensor, as there are several types of distance sensors available.

2.2.7 Video Feed

The UARC needed a way to gather information. Some of the options available are a mounted digital camera, video camera, various environmental sensors, radar equipment, etc. Since the UARC is primarily a data gathering autonomous aircraft, the group decided it would carry an onboard video camera that will feed real-time video back to a laptop.

Chapter 3: Specifications and Requirements

3.1 Motors

The UARC will use four DC powered motors to spin the propellers, steer and create lift. The motors will have to be able to rotate either clockwise or counter-clockwise. One pair of opposite motors will need to rotate clockwise, while the other opposite pair will need to rotate counter-clockwise to keep the angular acceleration about the yaw axis at zero – or essentially offset the torques normally handled by the tail rotor in a typical helicopter. These motors need to be powerful enough to allow the UARC to fly up to 10 feet in height for 15 minutes without draining the battery. Therefore they will need to have low power consumption and maximum efficiency. They cannot consume more than 11VDC since RC batteries with higher voltages aren't easily accessible. It's also important that they be low maintenance since a future application of the UARC could be military related and there would be limited parts on hand in field. They should also be as light as possible since the whole unit will need to get off the ground. The lighter the overall weight is, the longer the battery will last because the motors will be under less strain.

The motors must be small, yet powerful. However since the UARC will be built around a Draganflyer frame, there are only a few options for motor size since the motors must fit within the frame mounts. The motor size will have to be a 300 or a 280 with an adapter plate. The cost should also be a concern since some motors are very expensive and the UARC will require 4 for this application. The cost should be within budget. Some different types of motors available are brushed and brushless motors. For each of those types, there are also inrunner and outrunners. Each type operates at different RPM's and some require additional hardware to operate. The motors are easily one of the most important components and will play a large part in making the UARC live up the flight and power specifications set forth in this paper.

3.2 Microcontroller

The microcontroller (MC) is the brains of all the operation of UARC. There are many different types of MC's available and more research will need to be done to decide on the best one. UARC will comprise of at least two MC's. One main MC for motor control and stabilization while another for sensor reading and filtering. The inputs to the main MC will be all the sensor data, GPS, and input data transmitted from the user. It should be able to output voltage signals to the motors and also position data and video feed to the transmitter for the user to receive. It should be noted that the design will be switched to brushless motor should time be allowed. Therefore, the MC should be able to output both PWM signals and voltage controlled signals. The other MC will process the sensor data and send the filtered signals to the main MC. Due to the nature of this project, information will need to be processed fast, and therefore it should be able to compute multiple raw data quickly and accurately.

The MC should be at minimum 8-bit with an operating frequency between 20 and 40 MHz. There should be at least 1kB of data memory and 4KB of program memory. It should contain at least 10 I/O ports to accommodate the many inputs that it will have and any extra future inputs that may be implemented. A decision will be made to choose a name brand MC to avoid delay or problems in this project. Furthermore, choosing a reputable company allows easy implementation and should documentation be needed, suffice sources can be found online.

3.3 Sensors

3.3.1 Tilt Sensing

The requirement to measure the UARC's tilt is necessary for flight. For proper hovering and maintaining stability will require a sensor that will measure the error associated with the offset relative to ground parallel geometry. In other words, it will be necessary to measure in space when tilting occurs and is not parallel to earth. With this, the device will allow control of heading directions. During takeoff, the accelerometer will allow the monitoring of whether or not the motors are working in sync in a manner that the helicopter rises vertically. Secondly, this will allow crash avoidance into the ground while landing. Because the blades will be spinning as descending occurs, it is important to monitor tilting such that the blades do not interact with and objects in the ground. Avoiding this sort of scenario will avoid damaging the frame and other components, but also saving the motors. By subjecting the motors to a force that will stop it from spinning at an instant, such as a crash, will damage the inner workings of the motor and will not function properly. Thirdly, because the UARC will be working outdoors, many forces will act upon it spontaneously. For example, a gust of wind will act upon the rigid body and influence a change in the helicopters heading. It would be wise to know at the instance the UARC changes tilt to compensate quickly for stability. Lastly, the group would like to demonstrate the ability of feedback controls during presentation of how fascinating the UARC reacts to impulse. By shaking the helicopter in mid-air softly, it can be demonstrated that the UARC is using feedback to regain stability.

The accelerometer will have to be lightweight as well. Part of this project is to have efficiency in weight-to-parts ratio. By minimizing the weight this sensor will have, it allows flexibility in design for the use of other sensors in the event modification is needed.

Since the device will be outdoors, it is important that the device is able to operate in harsh weather and remain accurate in feedback. Furthermore, this device must not intercept or interfere with other devices found on the UARC. It must have high sensitivity to be able to track changes quicker and accurately. Bandwidth with several hundred Hz will be needed, as the application calls for fast tilting sensing.

3.3.2 Compass Sensing

The requirement for a compass is needed to guide the UARC. One of the axis will be set as the forward direction for heading and navigation. Furthermore, setting these cardinal directions will assist in smooth programming to the microcontroller by knowing how to interact with the engines. A magnetometer will provide the helicopter with a digital compass and any frame of reference so that navigation can be made possible. This device will interact with the GPS unit in order to have trajectory points and be able to distinguish in which direction it is traveling. The device will send signals to the microprocessor, be analyzed, and feedback should be sent out to adjust for heading. The sole purpose of this sensor is only to set an orientation point for the helicopter and the project can be done without it. However, much coding will have to be done in the microcontroller to substitute this device and proves that the device is powerful. This sensor will have to be lightweight and small in size to remain in the design.

The challenge will come if the UARC tilts enough where the sensor cannot give a proper value due to the orientation the helicopter is found. The objective will be that the helicopter is never found in this scenario and thus the magnetometer will give values at all times. The device will have high sensitivity to give accurate readings very frequently to assist navigation. It should be small and compact in size so that it can be incorporated into the project. Furthermore, it should be lightweight so that it does not detriment the payload it exhibits. An analog signal should be used since it is more accurate for the application. The apparatus should consume low power to leave the rest for flight time. It should withstand normal operation outdoors, since the UARC will be operated in those conditions. Resources of the device should be sufficient in order to program the device successfully. With that, it should be noted that the group will purchase the apparatus from a reputable company that has ample notes. Interfacing the device with the microcontroller should not be complicated to take away time from the other sensors that should be integrated.

3.3.3 Distance Sensing

This sensor will be utilized to measure how much distance there is between the UARC and the ground, especially during hovering. There will also be another sensor aimed upwards to detect a ceiling, branches or other obstacles that could interfere with the UARC’s flight path. The downward facing sensor will take constant measurements to sense how far off the ground the UARC is hovering. This method will be useful during outdoor and indoor flight. The sensor facing in the upward direction will have to act differently than the one facing down however. The sensor facing up will primarily be for safety. Its purpose will be to set off an alert if something comes within a specified range of about 5 feet. Ideally, this will be easily changeable in case the UARC flies indoors and the ceiling is lower. There are several types of sensors that could be used for this application. These include photoelectric or laser sensors, infrared sensors and ultrasonic sensors.

Each sensor should be as lightweight as possible and as small as possible to prolong flight time, and it's even more important because the UARC needs 2 of these sensors. The sensor should be able to measure distances between 6 inches and 15 feet for this application. This is because the UARC will fly or need to detect distances within that range. The sensor will have to be accurate, regardless of whether it's flying over grass, light concrete or dark asphalt. Also the sensor should preferably have multiple types of outputs like Analog, Digital or PWM signals. With different options available, the group can decide which output is more accurate and useful. This decision may be based on the confines of the microcontroller as well. The sensor will have to operate within the limits of the battery for voltage and current supplies. Therefore it should consume less than 11VDC and about 100mA.

3.3.4 Rotation Sensing

The purpose of the gyros is to update UARC on its rotational state. It should be able to detect a wide range of rotational speed, somewhere between 0.1deg/sec and 800deg/sec will suffice. It should be small in design to incorporate the design called for. Because weight is a detriment to the design, it should be lightweight such that the overall system is not heavy. It should run on a supply voltage no greater than 5v. The sensitivity of the gyro should be less than 1 mV/deg/sec. Temperature difference will be minimal during UARC's flight, so the sensitivity characteristics change will not be significant and any percent value will be permitted. The idea is to eventually transfer all the sensors to a PCB board; it should have a popular pin layout such that a socket can be found for it. Yaw, pitch, and roll will have to be measured to have stable flight. For this reason, an all-in-one device can be bought that incorporates all 3 axes to save money and coding. A software filter should be developed to average out signals such that sporadic signals are not accepted by the microprocessor and make flight nearly impossible. In other words, if the gyro was to output noisy measurements and given to the microprocessor, it might think the UARC is moving out of control when in reality it is stable. Therefore, the software should be able to average out the signals given from the gyros to make an assumption in what state the UARC is at the time. This is where sensitivity comes into place, by sending vast amount of outputs to help the microcontroller decide whether it should send a signal to the motors and other sensors to correct itself to achieve stability.

3.4 Flight Stabilization

In order to achieve flight stabilization a number of key factors are involved in making this a success. Countless hours have been spent on determining which sensors are needed for the appropriate feedback from the UARC to stay afloat. However, this is just the feedback circuitry that provides details to the microprocessor on how to correct itself. There are other factors that will go hand in hand to produce stable flight.

Payload is a big concern for the project. The more sensors that can be placed on the UARC, the highly the success can be. However, making the vehicle too heavy will diminish flight time or not have flight at all. Conversely, if the vehicle is made too light, outside forces can greatly influence crash and damage sensors or the rigid structure. The chosen motors should be able to lift the entire structure; on-board sensors, power supply, and motor themselves. To remain efficient in power, brushless motors have a key component on lifting twice its payload and less power consumption than its counterpart brushed motors. Therefore, finding the correct brushless motor will be factored such that there is a balance that will be suitable for the application. Ideally, this should be the heaviest components used and should be cognizant of weight for the rest of the parts. For this reason, the structure calls for a material that is able to absorb damage without breaking, can carry payload, and resist bending all while remaining light. The feedback control system should be small in size, easy to program and interface, and provide accurate results when demanded. Finally, the power supply should be able to provide enough stamina to show the work while not burdening the payload.

Correctly configuring the feedback controls is necessary for stable flight. Some sensors will have to be ignored during certain criteria. For example, if a coordinate is issued that demands the UARC in making a turn, a signal is sent to the motors to carry out the role. During this time, power is increased to the motor for the appropriate turn and less power sent to the other motor to create an arc. The accelerometer and gyros should be programmed well to allow the apparatus to carry out the sequence of turning. Should the sensors be allowed to remain on, the microcontroller will be sending a signal to turn all while the sensors are informing the microcontroller to correct itself due to tilting. Each sensor will have to undergo different scenarios to determine whether they should be on or off to carry out a request from the microcontroller. During programming, careful consideration should be made to this area to make sure no errors are made. Because constant feedback is being sent to the microcontroller, any instruction that interferes with the sensors or the UARC can have disasters.

The power supply should be scrutinized to achieve perfect balance in power-to-weight ratio. It should hold enough power to hold flight for at least 15 minutes all while powering every component on the UARC. For the one chosen, it should remain lightweight but have at least 11V of power supply. Quality will also be looked at to make sure the battery purchased can be charged multiple times before degrading the performance of the cells. A program shall be developed that determines how much power is left in the power supply before it is discharged. With this information, a safe-guard program will be running in the background for emergency landing, such to prevent damage to the UARC. A slight variation in engine speed will draw huge amounts of current. Therefore, the cells should be able to hold up as much as 50 amps of current at any point in time.

Finally, the expectation of stabilization will be to have flight time of at least 15 minutes. The sensors should accurately give feedback to assist hovering. The material being used as the body will have to absorb some damage and have high tensile strength. Sensors will be programmed to be turned off and on as needed such that they do not interfere with navigation. While remaining lightweight, the quality of the battery should have up to 11V producing 50 amps.

3.5 Frame Material

The UARC frame will have to be composed of a lightweight rigid structure. By remaining lightweight, it gives freedom to spend more payloads on electronics and power supply. Having said this, the material should not be flimsy and have deformation due to weight restrictions. The material should be able to handle involuntary crashes or sustain multiple abuses from normal wear and tear. It must remain resilient for moments of hard stress testing, but be able to conform back to its original shape. The material will have to be hollow in the center, having a straw-like shape, to reduce the unneeded weight as it serves no purpose. This will also assist in hiding the circuitry wires inside the tube-like structure to maintain a neat appearance in design. Because the vehicle will be used outdoors, it should be weather resistant to a certain extent. The UARC should not be subjected to rain, puddles, ponds, or lakes. However, it is expected the material to be subjected to moisture from the air, but the material should not absorb any that will degrade the materials performance. On the contrary, the vehicle might be subjected to extreme temperatures of the sun. It should be heat tolerant to bending due to the suns radiating heat it produces. Many materials are made for indoor use only and its properties change under extreme heat. As one can see, the material calls for very stringent requirements and finding such material will not be at a reasonable price. Due to time restrictions, the material used will have to be commercially available as research in material engineering is rather time consuming. The potential of developing such material specifically for this application could take months to develop and perfect. Furthermore, time would be needed to have a testing phase and assure the product could be used for this application. Finding a company that has afforded research, testing, and the expense to suit the project will save the group money.

To avoid damaging the material, joints will be used where necessary. Part of this project will be a test prototype to learn the dynamics and characteristics of the projects sensors. Once completed, it will be transferred into the quad-copter design to continue the project. It should be flexible enough in design to be taken apart and put back into different construction shapes to meet test and final prototype. The joints will be hard to find in the same material and should be made of plastic that is relatively cheap and lightweight. This flexibility will incur less expense and be able to change in design should it be needed.

3.6 Power Supply

The UARC will require a portable power supply to run all the electronics, sensors, motors and other hardware. It will need to supply sufficient voltage and current to allow the UARC to perform for a minimum of 15 minutes. There are many types of batteries, producing many different voltages and currents. Some types of batteries include Nickel-Metal Hydride (NiMH), Nickel-Cadmium (NiCd), Lithium-ion Polymer (Li-Poly) and Lithium-ion (Li-ion). Each one has benefits and drawbacks, which makes them more useful in specific applications. These differences will be covered later in this paper.

The primary specification for the power supply will be that it must be lightweight and small. This is the one item that will weigh the most out of every other component used on the UARC. The battery must be rechargeable; therefore the group will need to purchase a battery and a charger, which can be very expensive. So costs must be factored into the decision making process to ensure the group stays within the budget. Some chargers are powered from an AC voltage source, while others are powered by a DC voltage source. Using a DC voltage source to recharge the battery will require that the group purchase a large battery, however this would make the most sense in the event that the UARC was used in the field during a military operation. A DC battery is portable and easier to use than creating an AC voltage source in the middle of a jungle for instance. The voltage selected must be higher than any voltage required by any components on board the UARC to make the power flow easier to control. It's easier to step down DC voltage than it is to step it up. The same philosophy must be used for the current. The current supplied has to be more than the total current consumed by the motors and all devices onboard the UARC to ensure there are no power problems.

3.7 Dynamics

The dynamics of UARC are relatively simple due to the fact that it's modeled as a symmetrical rigid body system. There is a six degree of freedom movement derived from the translational and rotational physics. The inputs to the system will consist of total thrust on the vertical axis (F) and the torques about each axis (T1, T2, T3). These dynamics will also need to be represented from an inertial frame, being its initial coordinates before flight. Figure 6 below sums all this up.

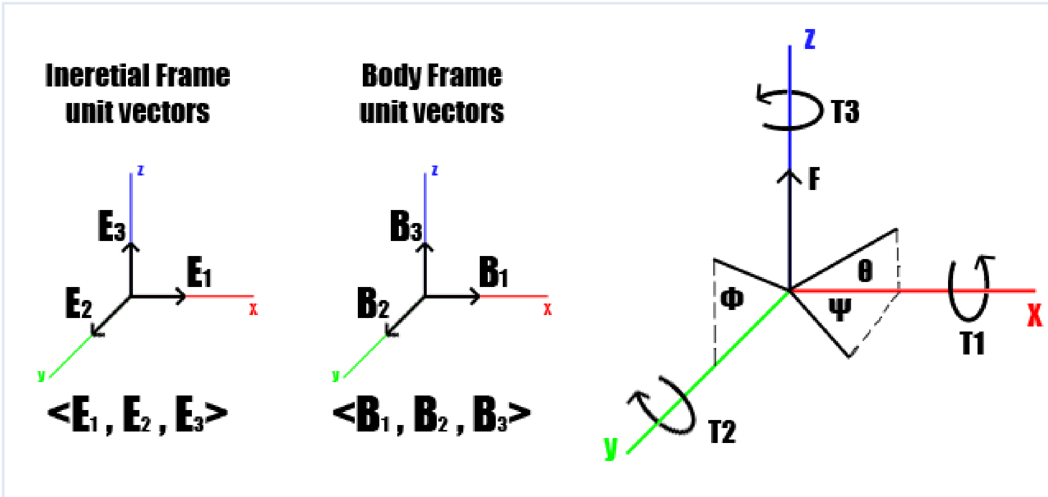


Figure 6: Earth and body unit vector coordinates and input forces.

As one can see the translational coordinates are represented by the traditional x, y, and z notation and the Euler angles of rotation are represented by Φ, θ, and Ψ. These six output coordinates are obtained from the four inputs given. Since the number of inputs is less than outputs, this system is considered under actuated. From this information the dynamic control block can be modeled as shown to the right in Figure 7. The dynamic control system should be able to maneuver UARC in any direction or angle based on the error signals obtained through these four inputs.

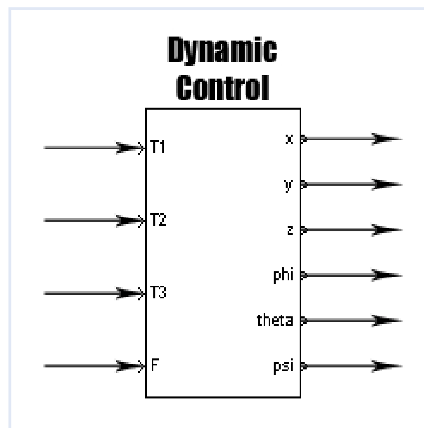


Figure 7: Dynamic control block

The translational components of the dynamic system will be derived from the use of Newton's second theory of motion.

(1) $F = ma$

The rotational components will be derived from the laws of angular momentum using the equation below.

(2) $\tau = I\alpha$

From equation 1, the sum of all forces is equal to the mass times acceleration. The thrust forces in UARC only act on the z axis according to the system derived, so F in x and y are equal to zero. This is why there are two less inputs then outputs. Now the rotational counterpart of equation 1 is equation 2. The sum of all torques is equal to the moment of inertia times the angular acceleration. By varying the thrust of individual motors, translation in the x and y coordinates can be controlled. Figure 8 below shows one way of implementing this concept.

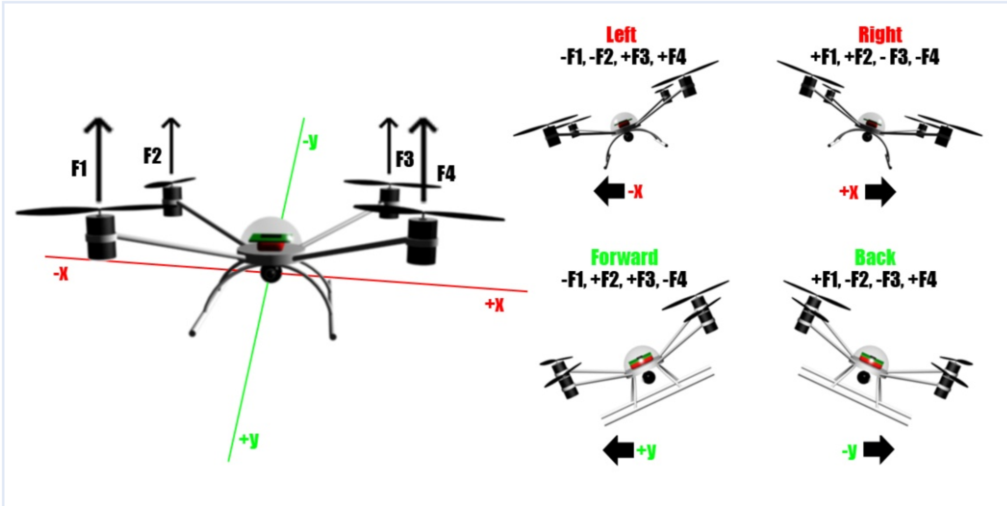


Figure 8: Individual thrust control for x and y translation

3.8 Communications

The UARC will need a form of communication to pass instructions and data. It will tailor to suit the needs of the vehicle since it will use many different channels to pass information. The sensors will communicate with the microcontroller using either analog or digital pulse. Both of these pulses will be studied to gain an understanding of which will be best suitable application. It is important that this communication not have much interference since it is dependent on feedback to assist navigation. It should be easy to manage and implement. Complicated programming will affect the outcome of this project and delay the timeline of finishing this project.

The speed controller will have a unique way of controlling the motors. It will have to take a signal that is produced from the microcontroller and be able to be interpreted by the speed controllers and further be passed to the motors. Hence, mixed signals will be used for certain instructions to be carried out. A component will be used throughout that will interpret one form of communication and output a different communication to complete the task. However, both signals should not be mixed together unless there is a component in between to interact with the signals.

Wireless communication will also be used in this project. It should be able to transmit video successfully to a portable laptop nearby. Higher bandwidth will allow larger information to be broadcasted. However, this comes at a price tag and is not suitable for the application. While the concern is not in sending high pixilated video for superior quality, a balance will be met to transmit average quality material. It should conform to FCC regulations and avoid interfering with nearby frequencies in the air. The sole purpose of this project is to show a prototype device that can be further enhanced down the line. Therefore, the video will not be encrypted to avoid data-safe signal. In a scenario for secret mission reconnaissance, the video streaming should be protected, but this is out of scope of the projects expectations.

3.9 GPS

The UARC will have an onboard GPS to allow it to maneuver through 3-D space and to locate certain points of interests. A GPS is the only logical solution for the UARC to find its target and return home. The GPS will track its location using satellites and output data to the microcontroller. That information will both be used for navigational purposes and sent to a laptop for real-time tracking of the UARC. Ideally, GPS coordinates could be uploaded to the laptop and sent to the UARC. The UARC will then move to those coordinates, or stay in a certain area marked by uploaded coordinates. Of course GPS coordinates could also be uploaded as pre-programmed flight patterns stored in a memory device where the UARC would automatically go to a location take video shots or images then return home. The UARC should also remain in one location and hover in the event that the GPS signal is lost.

The GPS module must be small and lightweight, like the other components of the UARC. The GPS should get a satellite signal within a reasonable time period of 45 seconds maximum, whether it's a cold, warm or hot start. It should consume as little voltage and current as possible to minimize the drain to the battery. The GPS should be very sensitive and have high accuracy to allow the UARC to obtain accurate readings.

3.10 Speed Controllers

An electronic speed controller does what its name implies - it controls the speed of a motor. Since each speed controller can control only one motor, the UARC will need a total of 4. The UARC's onboard sensors will gather information and send it to the microcontroller. The microcontroller processes it, and then creates 4 separate voltages or PWM signals that represent the speed of each individual motor. Some speed controllers work by using a varying voltage input instead of a PWM signal. Each speed controller takes that signal and controls its respective motor speed accordingly. There are many types, sizes and brands available on the market today for many applications. Some of these types include feedback or closed loop, open loop, brushed or brushless speed controllers.

The speed controller for this application should be chosen based on the type of motor selected since there are different speed controllers for brushed and brushless motors. The type should also be chosen depending on the current requirements of the motors. It's essential that the speed controller have a higher current rating than the motors to prevent damage. The size and weight are also a major concern since one will be required for each motor or 4 total. Speed controllers also have voltage limits, so they will have to be able to work off an 11VDC power supply. Some speed controllers have other available options that are important to consider, like low voltage cutoff or programmability features, which may require specific or additional hardware.

3. 11 Video

Part of this project is showing the aesthetics that can be used for applications such as military. Because of this, video streaming is very important in defining the purpose of this project. Unmanned vehicles need a form of transmitting back information to a centralized station to be interpreted. The video streaming device should therefore be able to transmit video wirelessly to a hub station. It should be allowed to reach 30 feet of over-the-air broadcast transmission. It should be able to work effortlessly without the use of creating a PCB board to be integrated. Furthermore, it should be plug-and-play on the UARC and have a receiver to reconstruct the video. Composite video connectors will have to be found on the receiver to plug into a monitor. The onboard video transmitter should be lightweight, small, and capture digital video. The power supply will be rerouted from the UARC to avoid extra payload.

If time allows, a motion detecting camera should be used as an idealized camera. It should capture a moving target and zero in and keep focused on the target regardless of the UARC's motion. The camera should relay feedback to the microcontroller on navigation coordinates to assist focusing on the target. In other words, if it is a moving target, it should send instructions to the microcontroller and let the microcontroller decide to ramp up or slow down motors to catch up. Should the target be motion-less, the UARC will hover over target and stream video back to hub. This cat and mouse approach could be of an advantage since no navigation is done by a user. Additional steps will be taken to assure the safety of the vehicle is not harmed or destroyed when it focuses on a target. It should not interfere with any of the onboard sensors from working accurately. The object of chasing a target should not place the UARC in danger of recklessly crashing to stay focused. Should the focus be lost, the UARC should remain hovering in the last place before it lost signal. Once the focus is lost the camera should be reset to acquire another point of interest. Implementing this idea should not degrade the rest of the project. Much of the focus being put into place in the vehicle is showing navigation and feedback control success. Although video streaming is very important, it can be shown that implementing this design would require more time. Nevertheless, it can prove to be of much importance and a powerful design that would draw attention to military aspects.

3.12 Software Programming

Software programming is the back bone of the whole project. It will guide the flow of data from the sensor input to the physical movement of UARC. There are many programming languages to choose from out there. Many popular microchips have compilers that can convert between languages so choice is really just a matter of preference. Example languages are C, C++, java, basic, and assembly.

Chapter 4: Research

4.1 Research Methods

The internet has proved to be a great source to assist in the development of this project. With the use of search engines, countless web hits were found that aided in understanding different phases of the project. Studying similar projects assisted in rejecting or accepting ideas and components based on how different projects came to a success. Moreover, it allowed the group to build a timeline of what areas would be time consuming, where other groups ran into issues, or where groups failed in succeeding. The ideas were collected and allowed judgment in determining whether they should be implemented, thrown out, or manipulated. It is assured the information gathered were from reliable sources, it was documented, and cited as needed.

The library was needed for references on dynamics and learning flight. The groups planning phase took place in study rooms located in the library and assisted in a quiet environment to carry out meetings. The UCF websites 'Databases for Articles & Other information' assisted in searching books inside the library to meet the needs of the current topic at hand. Proper bibliography was noted to make sure proper credits were issued.

As most engineers, curiosity aids in the development of a successful challenge. It should be notated that research was also performed by trial-and-error with careful testing. Sometimes with this brute-force attempt, one can learn many different do's and dont's to carry out the project. With careful testing, one can minimize damage in the event the trial proved to be ineffective. It was assured that wasteful spending was not done to go over budget or create many unsuccessful attempts with no progress.

Networking with different groups and professors also furthered the project. By visiting different department professors, it allowed the group to answer any questions that arise. Many questions were answered in subject such as dynamics, feedback control, flight, forces on rigid structure, and hardware. This proved to be the most effective resource that was used due to the fact that experience can go beyond what a book might include. For instance, while a book might give information on separate issues, a professor who has assisted in the creation of project such as this one can be more effective. Thorough searching through books could take hours of research to find the exact topic one searches. However, by asking an expert in the area could answer the questions in a matter of an office visit.

4.2 Flight Dynamics

Much research has been done on the possible ways to model the dynamical system of UARC. There are many approaches to defining a stable and efficient mathematical system. The quad rotor design has many important physical effects taking place like aerodynamic, inertial counter torques, gravity, friction, gyroscopic, etc.

First and foremost for any system is to label the frames of reference. A most common variable used to represent the inertial frame is the E vector representing the origin coordinates. Also a B vector is defined to represent the body frame of the physical device, which in this case is UARC. Due to the special symmetrical nature of UARC the center of gravity is at the origin of the B frame along with the moments of inertia. Also UARC is modeled as a rigid body. Out of every article and document referenced this is always the case. The beauty of this is that of simplification and efficiency. This aids to create the under actuated system that was described in chapter 3 section 7. Figure 9 below shows the free body diagram of UARC.

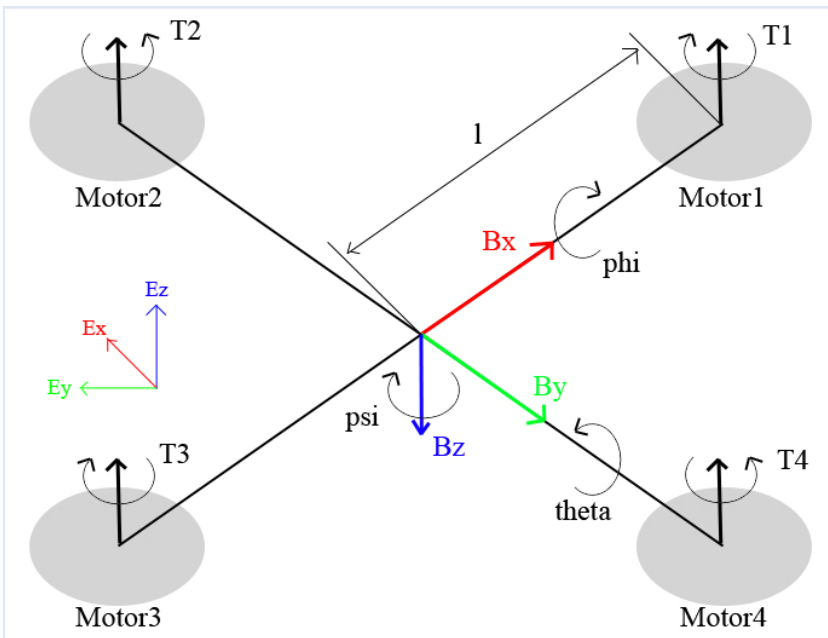


Figure 9: Free body diagram

In helicopters there is just one main rotor producing vertical lift and a smaller tail rotor to counteract the Coriolis Effect. Thanks to the symmetrical nature and even number of rotors the Coriolis Effect can be counter acted in UARC by having the moments created from opposing motors cancel out. This is accomplished by setting one motor in a clockwise motion with the other in a counter clockwise motion. Figure 10 below shows this phenomenon.

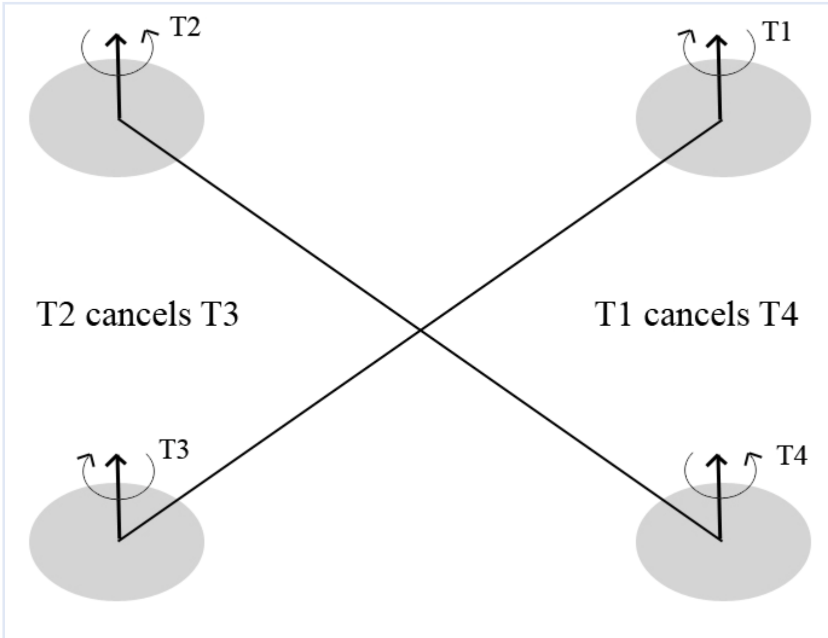
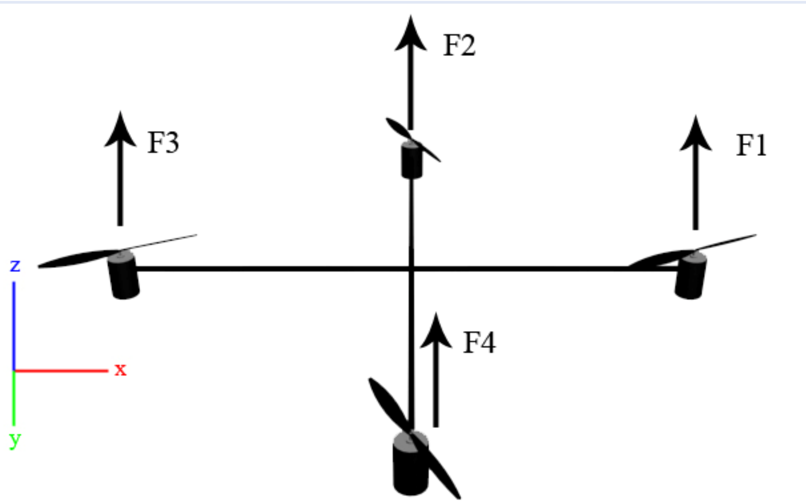


Figure 10: counter torques

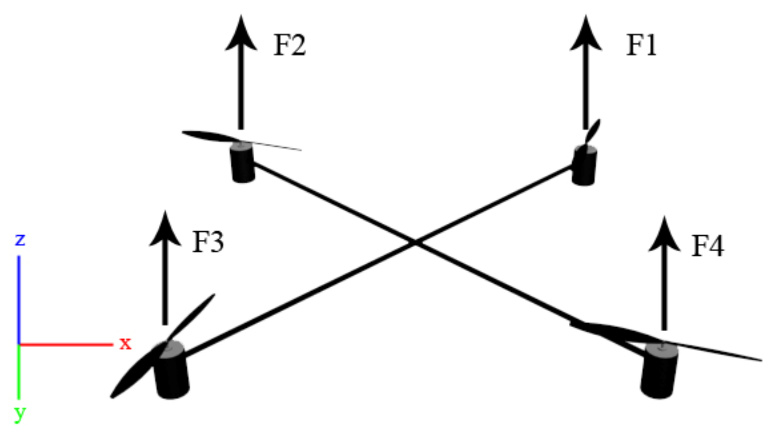
By varying individual thrust of two motors on the same axis the rotational angles will be controlled leading to translation. In the model being derived here, motors 2 and 4 can be varied to change the roll angle Φ, which will cause translation along the y axis. Complementary to that, motors 1 and 3 can vary to change the pitch angle θ, which causes translation along the x axis. This is called the + setup because of the orientation of the motors, which are on the respective axes. Please refer to figure 11.



Increase	Decrease	Direction
F1	F3	- x
F3	F1	+ x
F4	F2	- y
F2	F4	+y

Figure 11: + setup

Another setup is called X. This almost the same thing except different motors will work together to control the rotation and translation. Figure 12 shows this method and how the motors can be adjusted to accomplish movement. The method of choice would really be a matter of preference because there are no real advantages or disadvantages to either. It's all a matter of programming and consistency. UARC will apply the + setup just to avoid any conflicts later on in the design implementation.



Increase	Decrease	Direction
F1, F4	F3, F2	- x
F3, F2	F1, F4	+ x
F4, F3	F2, F1	- y
F2, F1	F4, F3	+y

Figure 12: x setup.

Now when in motion there are a few things to take into account. First of all would be the drag created by the rotors that is opposing the motion at hand. There is also a net moment along each axis of rotation created by the change in thrust. These forces and moments can all be related to the square of angular velocity of the blade, Ω, through aerodynamic coefficients C_t, C_q. Let A be blade area, ρ density of air and r be radius of the blade then,

$$T = C_T \rho A r^2 \Omega^2$$
$$Q = C_Q \rho A r^2 \Omega^2 r$$

Where T is thrust and Q is torque on the rotor shaft. When at hover T and Q can be defined as,

$$T_i = b\Omega_i^2$$
$$Q_i = b\Omega_i^2$$

This is possible because Ct, Cq, and ρ are constants where i represents each individual motor (i = 1, 2, 3, 4). Ultimately blade angular velocity Ω will be represented by applied voltage.

Now under the conditions established and through Newton-Euler formulation, one can derive the equations of motion.

$$\begin{bmatrix} m\boldsymbol{I} & 0 \\ 0 & \boldsymbol{J} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times m\boldsymbol{v}^b \\ \omega^b \times \boldsymbol{J}\omega^b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^b \\ \tau^b \end{bmatrix}$$

Where m is mass, I is an identity matrix, J is an inertial matrix, V^b represents the linear velocity in the body frame of reference, ω^b is angular velocity in body frame, f^b represents the total of all forces acting on the rigid body, and τ^b is the sum of torques applied to the body.

The translational dynamics are,

$$\boldsymbol{f}^b = \omega^b \times m\, v^b + \boldsymbol{f}_{tot}$$

f_{tot} is defined as,

$$f_{tot} = -C_{x,y,z}\left(\left(v^b\right)^2\right) + mgZ + \sum_{i=1}^{4}\left[-T_i\, z - D_i\,(x \;\; y)\right]$$

$C_{x,\, y,\, z}$ represents the drag coefficients, Z defines the vertical axis in the inertial frame, while the (x, y) defines velocity direction, g is the force of gravity, and D_i is the drag force on rotor opposing the direction of travel.

When in hover $D_i = 0$ and U_1 vertical force input can be defined as,

$$u_1 = \sum_{i=1}^{4} T_i = b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right)$$

If one neglects friction force and the effect of body moments the forces can be expressed in the inertial frame by,

$$m\ddot{X} = \left(\cos\Phi\sin\theta\cos\psi + \sin\Phi\sin\psi\right) u_1$$

$$m\ddot{Y} = \left(\cos\Phi\sin\theta\sin\psi - \sin\Phi\cos\psi\right) u_1$$

$$m\ddot{Z} = mg - (\cos\Phi\cos\theta)\, u_1$$

Where Φ represents the pitch angle around the x axis, θ is the roll angle around the y axis, and Ψ is the yaw angle around the z axis.

The Rotational dynamics are,

$$\tau^b = \omega^b \times J\omega^b + \tau_{total}$$

τ_{total} is defined as,

$$\tau_{total} = \left[\left(\sum_{i=1}^{4} \left[Q_i\, z + R_i\,(x \;\; y) + D_i\, h\,(-y \;\; x) \right] + l\left(-T_2 + T_4\right) x + l\left(T_1 - T_3\right) y \right] \right.$$

R_i represents the net moment about the roll axis when in translation, h is the vertical distance between propeller center and UARC center of gravity, l is the length of motor arm from center, and T1, T2, T3, and T4 represent the individual thrusts of the corresponding motor number.

When in hover D_i and R_i are equal to 0; U2, U3, and U4 can be defined as,

$$u_2 = b\left(\Omega_4^2 - \Omega_2^2\right)$$
$$u_2 = b\left(\Omega_3^2 - \Omega_1^2\right)$$
$$u_2 = d\left(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2\right)$$

U_2 is called the roll actuator input, U_3 the pitch actuator input, and U_4 the yaw moment input. The Rotational dynamics of UARC in the body axis are now defined as,

$$J_x \ddot{\Phi} = \dot{\theta}\dot{\psi}\left(J_y - J_z\right) + lu_2$$
$$J_y \ddot{\theta} = \dot{\Phi}\dot{\psi}\left(J_z - J_x\right) + lu_3$$
$$J_z \ddot{\psi} = \dot{\Phi}\dot{\theta}\left(J_x - J_y\right) + u_4$$

4.3 Flight Hardware

4.3.1 Frame

The most important quality to consider when doing research for suitable frame materials is weight. But also strength is important since the chance is likely that at some point the UARC will crash and fall to the earth. In-field repair to the frame could be costly and time consuming, not to mention a broken frame could cause further damage to the electrical components and wires that attach to it. There are several materials available to choose from including straws, wood, plastic, carbon fiber, steel, aluminum and cardboard. Each one was studied in detail.

Originally the idea of using straws came up since they are lightweight and weather resistant. There also was a similar small helicopter online that was made using straws as the frame material. The problem is they don’t have any real strength for this application, so straws were quickly discarded. Then the group considered plastic as frame material. Plastic is fairly lightweight, weather resistant and strong. There are several types of plastics and they can be molded solid or hollow and in any shape desired. Plastic was a contender. Wood was also considered because it's easy to work with and it mainly satisfied all the other specifications. Depending on the type of wood that's used however, it may weigh more than the plastic frame. Also there was a question about how well it would hold up in heat, humidity and moisture laden environments. The strength would most likely be less than plastic, but nonetheless wood remained a contender as well. Next the group considered carbon fiber. Carbon fiber meets all the specifications for frame materials set by the group, however it's expensive and availability was a concern. But it became another possible solution. Steel was another option. Steel is strong and fairly weather resistant but the weight would be more than plastic, wood or carbon fiber. Since weight is of paramount importance, this along with the fact that steel is harder to work with caused steel to be discarded. Aluminum was also a possibility. It's strong and meets all the specifications with the exception of weight. While it's a lightweight metal and lighter than steel, it's still heavier than plastic, wood or carbon fiber. Cardboard was also considered because it's light, however it's not very strong compared to the other materials. Cardboard is also not very weather resistant.

Once all the pros and cons were weighed, the choices were narrowed down to plastic, wood and carbon fiber. At this stage various layouts were discussed and many design choices were considered. Due to the shape and location of the motors, the group had to find various ways to attach everything together. This included motor mounts and other hardware. More research was performed to look for off-the-shelf lightweight options to use in order to keep costs down. It was during this time period when the group discovered that the Draganflyer helicopter is sold as a bare airframe assembly that includes the frame, motors, gears and rotors. The frame is made of carbon fiber with a plastic cross piece and plastic motor mounts. Once this was discovered, a careful cost analysis was performed to see if this option would be cost effective. In the end, the decision was made to order the airframe. It sells at Rctoys.com and they also carry a full line of replacement parts in case there's any damage to any of the parts. The assembly also includes four brushed DC motors that the group can use. In order to add more stability to the airframe, the group also purchased a frame bracing set that includes four carbon fiber rods that mount between the four legs of the frame using plastic connectors. This set will reduce vibration, help to stabilize flight, and give it added stability.

4.3.2 Landing Gear

The UARC will need a contraption to be able to land without any real danger of damaging system components. Part of the reason that the design calls for this application is to absorb all force and damage that might occur while landing. Furthermore, in the event a hard landing occurs, much of the damage can be taken by the landing gear without any real sacrifice. It will be easier to replace the landing gear than to replace the entire frame that the UARC is built upon. The drawback to this idea is the added weight to the vehicle itself. One can control this by implementing the same idea as referenced in the previous section in deciding which material to use with pros and cons. As a result, the best approach would be to use carbon fiber with its relatively low weight ratio. However, one must consider the price that this will cost the group to design. The typical approach, as seen in figure 13, was discarded as it would add to much weight. With this layout, the cost of having it as carbon fiber would amount to something way over the budget allotted.

Figure 13: Typical landing gear found on RC helicopters. Courtesy of Noah Kuntz.

Compensation was found by allowing the UARC to have a different approach as landing gear. Plastic ‘feet’ were used instead to implement landing gear and still maintain low weight added to the unit. Because the frame of the Draganflyer will be used, the landing gear is already incorporated into the frame.

4.3.3 Circuit Board Housing

Since the UARC will be subjected to outdoor weather, it was suggested to protect the circuitry from harsh climate. Because the circuit board will be done outside by a vendor, the beauty of it should be displayed. For this reason, it has been chosen to use a plastic clear case to house all the electronics. Another useful advantage of implementing this is to protect the electronics should the UARC land on its topside. Should this occur with no safety found, it could damage internal components. Although it can be repaired, it cannot be done without expenditure and time to fix.

4.3.4 Motor Mount Housing

The motors will take the most impact and damage should something go awry. For this reason, research was done to see if one could add some sort of protection barrier for the motors. This would assist with dust and other particles from reaching the inner workings of the motor, protect it from crash landings, and hide the motor from exposure to weather conditions. However, the added weight and changing schematics outweighed the added benefit. For this reason, the implementation was dropped from the design called upon.

4.3.5 Propellers

Propellers are just as important to flight as the motors that turn them. There are a few things to take into consideration when choosing them. First off what kind of motor is going to be turning this and how fast can it turn? What kind of material should they compose of? How long should they be? What pitch are they at? All of these factors can determine what propeller will operate best for the given task.

UARC will be designed with the frame of the Draganflyer RC quad copter. Its propellers are made of nylon, which is very durable and lightweight. Two sets of the blades are called A which go on the motors that spin counter clockwise. The other two sets are called B and they go on the motors that spin clockwise. This is important in the case of UARC because of the Coriolis Effect cancellation of the motors involved in the flight dynamics. If the blades weren't designed that way then changing rotation of the motors would cause lift in the opposite direction and that would be bad.

Figure 14: DraganFlyer rotor blades

The DraganFlyer propellers are attached to a big gear which is driven by the smaller pinion gear of the motor. This creates a lower torque on the props while high RPM coming from the motor. This allows for the blades to be more flexible and less likely to flap. Another kind of prop considered was a more rigid kind that would slide right on to the motor itself, called an outrunner. The advantage of this is performance, but at the cost of needing to replace the motors and modifying the frame.

The length and radius of the props are very crucial because as each parameter increases so does the load put on the motor. This means that if the propellers are to big then the motor will draw more amps to meet its specifications and could possibly burn itself out. These parameters also affect the lift and drag coefficients of the vehicle.

Pitch angle of the blade can affect the RPM and thrust of each propeller. As pitch angle increase so does lift. An airplane, for example, has very little pitch when in take off, to get high RPMs, and then raises the pitch when cruising. In a helicopter pitch is very much useful in control orientation. Figure 15 below shows this. UARC will have a constant pitch of about 12 degrees on its rotors and movement will be controlled instead by the manipulation of voltage between the motors.

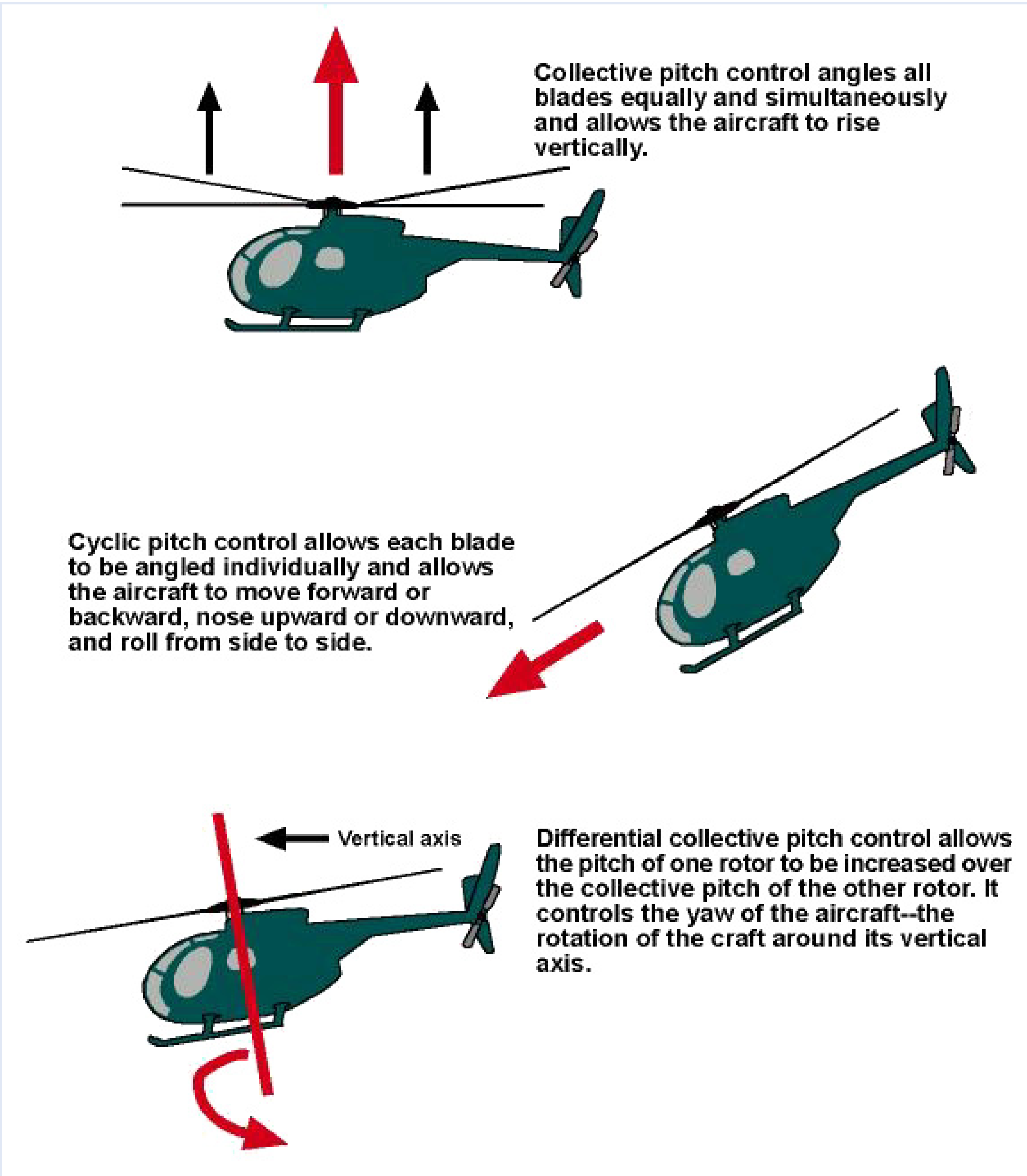


Figure 15: Pitch control for movement in a helicopter

4.3.6 Inrunners vs. Outrunners

There are two types of connections between a propeller and motor. The first connection is called direct drive or inrunner in RC terms. The second is called indirect drive or outrunner. The motor actually determines which setup is necessary for the given task at hand. A outrunner is basically just putting the propeller directly on the motor shaft. The latter is to have a gearbox with a given gear ratio to translate the motor rotation to the propeller.

Figure 16: Inrunner and Outrunner motor setups. Courtesy of hooked-on-rc-airplanes.com

Most people prefer the outrunner setup for their RC. They have better performance, but are generally not as efficient as inrunners. Outrunners spin slower than inrunners, but produce far more torque. This reduces weight and complexity. Noise is also less since there is no contact between gears. A disadvantage of outrunners is the window of available propellers. The propellers can't be to big otherwise they will destroy the motor. With the use of gears in the inrunner setup, one can adjust the torque and power through the gear ratio, allowing for a wider variety of propellers.

Inrunner motors can reach very high RPM's, but lack in the torque area. This is why a gear box needs to be implemented as to reduce the RPM's and increase torque. This just creates more things that can break! It is not uncommon for gear boxes to get bent or gears to get stripped. Also mounting an inrunner is more of a task then just throwing on an outrunner rig.

The Draganflyer frame has a gear ratio of 1:5.6. It is an inrunner setup because of the use of gears. Its brushed motors can reach RPM's of 14,000 and rotate the rotors up to 2,500 RPM's. This setup is ideal because it's a nice balance of torque and speed and its rotor blades should really not exceed that limit due to their flexible nature. Figure 17 shows the DraganFlyers gearbox.

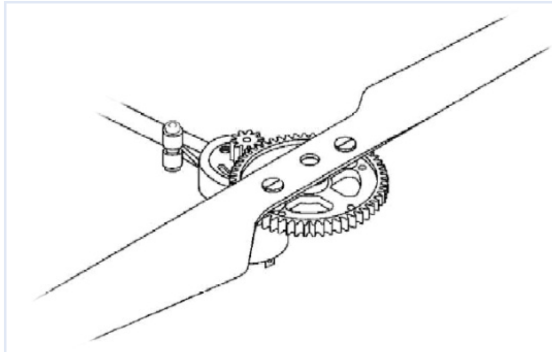


Figure 17: DraganFlyer Gear box

4.3.7 Speed Controllers

The UARC will need an Electronic Speed Controller (ESC) to control the speed of each motor. There are several options from many manufacturers and price ranges. There are different speed controllers for brushed and brushless motors. Both speed controllers create a PWM voltage signal that controls the speed of the motor. Brushed motors can also use a straight analog voltage input, although it's not as efficient. Since the 2-dimensional test setup will use brushed motors, it will require a brushed speed controller. The actual UARC will use brushless motors and will need brushless speed controllers. Most speed controllers serve the same basic function; the group looked primarily at size, customer reviews, price and availability to make a decision.

When choosing a speed controller for a given motor, the group had to ensure that the maximum current rating for the speed controller is higher than the motor's maximum current rating. Using a speed controller that can supply less current than the motor can consume will damage the speed controller. In the case of the UARC, the group has decided to use a motor with a 6A maximum current rating. Therefore the speed controller will need to be able to supply continuous current higher than 6A. Initially research was performed to find out which company makes the best and most reliable speed controllers. That research produced companies such as Castle Creations and Hacker. Both make speed controllers that are high quality and they are available online and at local hobby shops. Since the higher the current rating, the higher the price, the group decided to go with a speed controller around 10A.

Looking at Rctoys.com, the Castle Creations Phoenix 10A brushless speed controller costs $50.95 each. Seeing as how the UARC will use 4 speed controllers, this quickly adds up. It was also during this research phase that it was discovered that most speed controllers require some level of programming. These companies sell a separate programming card that allows the user to change settings like the battery cut-off voltage, battery type, air brakes on/off, number of battery cells, timing modes, start modes, motor direction, adjustable frequency, etc. This programmer along with the 4 motors is obviously more expensive than what the group budgeted. Another option was the Hacker X-12 Pro RC brushless speed controller that sells for $39.99 each. This again was a little expensive and requires an additional USB programmer that sells separately for $14.99. Alternatively, they also sell the Castle Creations Thunderbird 9A brushless speed controller for $25.49 each. What makes this controller unique is the fact that it's programmable, however it's not required. It comes ready to fly using a Li-Po battery.

After doing a lot of research online trying to find some information about what a speed controller does and how it works, Jeremy called an engineer from Castle Creations to ask these questions. The engineer told him that there's a lot going on in that little board, but he explained it in detail. He explained that Castle Creations is the first company that was able to design a speed controller that appears to have a linear output. Meaning a PWM signal with a 50% duty cycle may have not exactly mean the motor will be 50% power, but it will be close, maybe 40% - 60% power. This output it more linear than other controllers, allowing for a smooth transition. Normally with other speed controllers, the motor will slowly increase RPMs until around 70% or 80% then it will shoot up super fast. This can make the RPMs difficult to control. The 9A speed controller can run off a maximum of 15VDC. It's setup from the factory to work with a 3-cell Li-Po battery with a cut-off voltage of 3V per cell. It weighs only 0.3 oz or 8 grams. Since proper function of UARC requires that 2 motors rotate clockwise and the other 2 rotate counter-clockwise, rotation direction is changed by simply switching any 2 of the 3 wires that connect to the motor, instead of having to purchase a programmer and programming it differently.

Figure 18: Castle Creations brushless speed controller.

4.3.8 Motors

The UARC needs four motors to get off the ground and there are a wide variety of motors to choose from. In order to make the best decision, the group had to look at several factors such as the price, power supply being used, the size of the motor mounts on the frame and the many types of motors (ie. brushed versus brushless, outrunner versus inrunner, etc.) The motor that the group chooses will also have a major impact on the flight time of the UARC since the motors will consume the most power of any other device. It's important that the motors are efficient and as lightweight as possible.

The power supply will supply a DC voltage to the all the components on the UARC. Therefore the group chose to use a DC motor. At the most basic level, DC motors come in two types: brushed and brushless. Brushed motors have some advantages but several disadvantages. The main advantage is that brushed motors operate with a simple DC voltage input. Also voltage and RPM's are linearly proportional, as well as current and torque. In a brushed motor, brushes make contact with the rotating commutator/armature, forming an electric circuit. This rotates within the stationary stator where the permanent magnets are housed. These motors are cheaper than brushless motors. There are several disadvantages to this system, for example: the brushes and commutator tend to wear out and must be cleaned periodically. The friction between the brushes and commutator slow the motor down, creates heat (wasted energy) and decreases battery life. This all contributes to lower reliability and efficiency, shorter motor lifetime and a lower power to weight ratio.

With brushless motors, the armature stays stationary while the permanent magnets rotate preventing the need for brushes. This means there are no brushes or commutator to clean or replace. Also there's no friction to slow the motor down therefore the battery life is extended and the power to weight ratio is increased. The noise and the EMI are reduced as well. However brushless motors are more expensive and need a speed controller to control the speed. This requires additional hardware. Even with the added price, the choice seems clear that using brushless motors on the UARC is the way to go. The motor mounts on the frame dictate the size of the motor that the UARC can use. This is not a problem since the frame assembly came with brushed motors and since brushless motors are lighter and more powerful and efficient, the UARC can use a more powerful motor that's the same size or smaller. Something else to consider is using a inrunner or outrunner. For inrunner brushless motors, the permanent magnets are located inside the electromagnets. Some characteristics of inrunner's is that they have higher RPM's and are more efficient than outrunner's. Inrunner's need a gearbox however to reduce the high rotation of the motor down to a slower RPM for the propellers. This also means a inrunner brushless motor can be setup differently by changing the gear ratios. But there are also more parts that can fail during normal operation. Outrunner brushless motors rotate slower and have a higher torque than inrunners. They don't require a gearbox, but instead the propeller mounts directly on the motor's shaft. But each outrunner can only work for limited number of propellers. They are less efficient because they rotate slower. Another thing to consider is the Kv rating. Kv is the RPM per volt and each motor has a Kv rating. This should be chosen depending on the propellers used.

Since the frame assembly was already outfitted with inrunner brushed motors and bearing in mind the advantages and disadvantages listed above, the group decided to use inrunner brushless motors for the UARC. After doing research on several brushless motors, the group decided to use Feigao 1308441S brushless motors based on the size, price and other desired specs listed above. It has a Kv rating of 2283 RPM/V, which means using a 11V power supply, these motor have a total output of 25,113 RPMs. This motor is smaller than the frame mounts however, therefore it will require four adapter plates to mount the smaller motors to the frame mounts. For 2D test setup, the group decided to start off with the brushed motors that came with the frame assembly. This is because it will be easier to solve the control problem using motors that have a linear relationship between voltage and the RPM's, and current and torque. The speed controllers that control the speed of the brushless motors are not necessarily linear and there are other things that will have to be considered to get them to operate as intended. Once the control problem is solved, the group can then make the transition to the brushless motors and speed controllers.

4.4 Sensors

4.4.1 Gyros

To control angular rotation, the UARC will need to use a gyroscope. This is an essential piece of hardware that will sense unwanted rotation and send signals to the motors to compensate. This will be especially useful when the UARC is flying outside or in windy environments. The gyro is not without its problems however. The gyro only opposes rotation as long as the UARC is rotating. Once the rotation stops, the gyro resets and thinks that the aircraft is level or has returned to its initial heading. This can create what's referred to as gyroscopic drift. Therefore, it has to be used along with an accelerometer that will tell whether or not the UARC is level. The gyro can be purchased separately or it could be integrated into a 5 or 6 axis IMU (Inertial Measurement Unit), which integrates accelerometers and gyros into one unit.

There are two different types of gyros available. Rate gyros work as described above where when the rotation stops, the gyro resets even if the aircraft isn't level. Heading hold gyros keep track of the change in rotation then actually correct the aircraft until it returns to the original heading. While this is desirable on one hand, the immediate problem arises with cost. A single heading hold gyro can run up to $200 or more, while a rate gyro is typically around $30 each. It's for this reason this type was rejected. Since each gyro only measures rotation along one axis and there are 3 axes of rotation in the x, y and z, it makes sense that the UARC will require 3 gyros. However, the UARC would function with 2 gyros as well – just controlling the x and y axis. When selecting a specific gyro, the price, size and weight must all be considered. After researching several different gyros, the group narrowed it down to a 150 degree/sec gyro breakout board utilizing the Analog Devices ADXRS613 rate gyro, or the 300 degree/sec gyro breakout board utilizing the STMicroelectronics LISY300AL gyro. Both are available from Sparkfun.com.

The STMicroelectronics LISY300AL gyro is $29.95 each and can measure angular rates up to 300 degrees per second. This gyro runs off 2.7 – 3.6VDC at 4.8mA. It's capable of measuring rates with a –3dB bandwidth up to 88Hz. The breakout board uses all the values of components suggested in the specification drawing for the phase locked loop low pass filter. The user can reduce noise further by adding another low pass filter as well. Changing a capacitor will modify the output bandwidth. This package is a LGA or Land Grid Array package, which is a leadless package and is difficult to solder to a PCB. The price is nice, but the UARC will need something more sensitive than a 33 degree per second gyro. Also it's doubtful that the UARC would ever rotate 300 degrees.

The Analog Devices gyro is $59.95 each and can measure angular rates up to 150 degrees per second. It outputs a voltage relative to the rotation angle, but it's ratiometric with respect to the provided reference supply, which in this case is +5VDC at 5mA. The bandwidth can be set using an external capacitor. If the breakout board from Sparkfun is purchased, it includes this external capacitor. The breakout board sets the –3dB frequency response to 402 Hz, although it can range between 1Hz to 3 kHz. This could be changed by if required by replacing this capacitor, or adding a resistor between 2 of the pins. The alternative is to purchase this chip itself from Digikey for $34.59 each. The problem is the package is a BGA or Ball Grid Array, which is a leadless package and therefore difficult to solder on a PCB by hand. This component also has 2000*g* powered shock survivability, which may be useful in the event the UARC crashes. The group doesn't like to think about that however. Another positive aspect of this gyro is it weighs less than 0.5 gram. Although the price is higher than the STMicroelectronics gyro, the support from Analog Devices is better and the chip is more sensitive.

Figure 19: The ADXRS613 is a single rate gyro sold at Sparkfun.com

4.4.2 Accelerometers

Understanding how the accelerometer was to be installed was done easily by downloading the spec sheets from Sparkfun.com. From the instructions each pin out was described easily and thoroughly. It was learned that the sensor chip can withstand some amount of force of impact without damaging its internal mechanism. The ADXL202E, which is currently the sensor of choice for the project, has a 1000g shock survival capacity. This gives the assurance in case of crashing the UARC that it will continue to function. However, ESD continues to be an issue and the group will be careful when handling this sensor. Figure 20 shows the pin out layout for the chip and the specs needed for it to function.

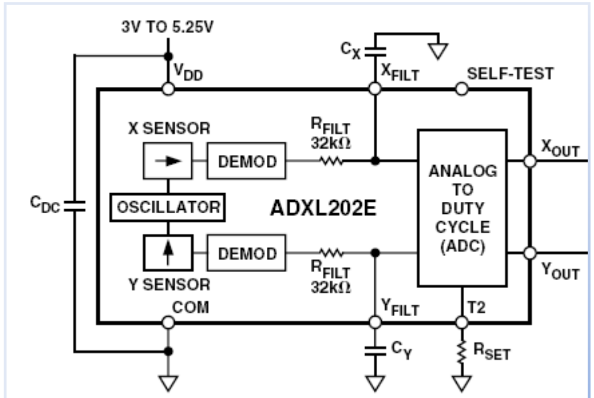


Figure 20: The pin layout for the ADXL202E accelerometer chip.

It was learned that chip can measure both vibration and gravitational acceleration. For either analog or digital outputs, their duty cycles are proportional to acceleration. With the use of a microcontroller A/D converter, the duty cycle can be measured and interpreted. To synchronize the signal, a resistor has to be placed on R_{set} that will adjust the reading from 0.5 ms to 10 ms. The bandwidth can be set by placing capacitors at C_x and C_y at pins X_{filt} and Y_{filt} respectively. Furthermore, the resolution which allows the smallest detectable acceleration is dependent on the bandwidth selected. The manufacturer supplied the following information to assist in implementing this sensor as found in table 2

Bandwidth	C_x, C_y	rms Noise	Peak-to-Peak Noise Estimate 95% Probability (rms x 4)
10 Hz	0.47 µF	0.8 mg	3.2 mg
50 Hz	0.10 µF	1.8 mg	7.2 mg
100 Hz	0.05 µF	2.5 mg	10.1 mg
200 Hz	0.027 µF	3.6 mg	14.3 mg
500 Hz	0.01 µF	5.7 mg	22.6 mg

Table 2: Filter Capacitor selection, Cx and Cy for the AXDL202E

To cancel the noise created by other circuits, input and common voltage are shunted by a capacitor. According to the specs, it is suggested to use a 0.1 µF capacitor. Additional modification should be done if the microcontroller and sensor share the same power supply by using a 100 Ω resistor in the supply line to lessen interference. This sensor was chosen for its high sensitivity and the use of lower speed counter for PWM decoding while maintaining high resolution.

4.4.3 Magnetometers

Thorough research was performed on how a magnetometer should be powered. Data sheets were found at Sparkfun.com that assisted in understanding how to configure the sensor. The Honeywell HMC6352 magneto sensor is a 4 pin layout chip as seen in figure 21.

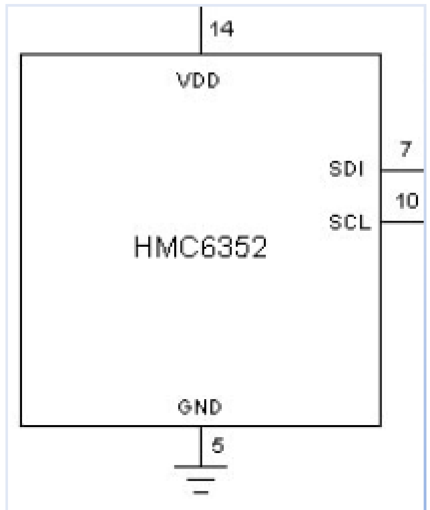


Figure 21: The pin layout for the Honeywell HMC6352.

Although the package of the chip has 24 pins, most are no connects and is placed on a PCB board with V_{DD}, SDI, SCL, and GND connections. The threshold voltage to power up is +3.0V and has a max input of +5.2V. The serial clock lines and serial data lines both require resistor pull-up between the microprocessor and this sensor of about 10KΩ. A decoupling capacitor of 0.01 µF is required from V_{DD} to ground to eliminate noise. The manufacturer suggests that the decoupling not exceed a couple of inches away from the sensor for best results. For example if a PCB board is designed with the filter already in place for the sensor, but the sensor trace line exceeds a couple of inches, they advise to place an additional one in the trace line right before soldering the sensor in place for stability. It was also interesting to learn that if the sensor is left on the shelf at constant humidity, a 24-hour bakeout period should be performed on the chip to remove humidity. This should be done due to the epoxy top encapsulation that the chip has and maintains correct functionality. Because a PCB board with all the sensors will be considered, instructions for soldering are found on the data sheet to follow to correctly set in place.

The microcontroller will have to send an 8-bit long string of data at each clock signal to configure the sensor. The first bit (bit 7) will always be ‘0’, while bits 6 and 5 set the continuous measurement rate that ranges from 1 – 20 Hz. Bit 4 set the periodic set/reset as ‘0’ or ‘1’ respectively followed by bits 3 and 2 which are set ‘0’. Finally, bits 1 and 0 are the operational instructions that set the sensor in standby, query, and continuous mode. After this, the sensor will spit out raw data in binary format that is 2 bits in length. The manufacturer has supplied the following table to describe the raw data.

Bit 2	Bit 1	Bit 0	Description
0	0	0	Heading Mode
0	0	1	Raw Magnetometer X Mode
0	1	0	Raw Magnetometer Y Mode
0	1	1	Magnetometer X Mode
1	0	0	Magnetometer Y Mode

Table 3: Binary table for the output given by the HMC6352.

The neat feature that this chip does is measurement summing. By using its internal EEPROM storage, it is able to gather up to 16 measurements and average out, or data smooth, the output for jitter free data output. The level of accuracy that can be expected is 2 degrees of compass heading error, while magnetic devices can skew the accuracy even more.

4.4.4 Ultrasonic

The UARC will need a way to measure altitude while flying or hovering. There are different ways to accomplish this, but the most reasonable way seems to be to use a distance sensor. There are several different sensors available to measure distance. Some of these include photoelectric or laser sensors, infrared sensors and ultrasonic sensors. As described in the specifications section, the sensor must be relatively accurate and measure distance regardless of the surface characteristics.

The first distance sensor that was considered is infrared. Infrared sensors work by sending out continuous infrared light that gets reflected off objects and picked up by an infrared receiver. There are several problems associated with this type of distance sensor as the group discovered. First of all since the sensor works by detecting infrared light, it becomes a lot less effective when it's being used around other sources of infrared light. This means that it won't be accurate when being used outside or in any area inside where there's sunlight present. Also the surface that the light is reflected off will also affect the accuracy. Dark or rougher surfaces will prevent the light from reflecting completely and could indicate that an object is 10 feet away when it's actually only 6 feet away for example. Then brighter surfaces will reflect more light. So if the UARC was flying over a dark area of ground then flew over a light color area, the sensor could think that the distance changed, making the aircraft climb in altitude to compensate. An advantage of infrared sensors is that it's cheaper than other forms of distance sensors. But because of these inherent problems, these types of sensors are normally used for obstacle detection rather than measuring distance.

The next sensor that was considered was photoelectric or laser sensors. These sensors measure distance by one of three ways. For shorter distances some will use geometric triangulation, or the process where the angle of the reflected light is proportional to the distance. For longer distances, others will send out a laser and compare the difference in phase between the light going out and the reflected light it receives back – this difference in phase is also directly proportional to the distance. For measuring very long distances other will send out a short light pulse and measure the time it takes for the sensitive light detector to detect a portion of the incoming reflected light that bounces off the object. These are accurate measuring devices however they are the most expensive and bulkiest of the distance measuring devices. Due to the weight restrictions of the UARC and budget of the group, this type of sensor was discarded.

Another type of distance sensor is the ultrasonic sensor (also called sonar). This sensor uses sound rather than light to take measurements. It sends out a short pulse of ultrasonic sound that reflects off an object and is received back by the sensor. It measures the time it takes for this process to occur and computes the distance by knowing the speed of sound in air. Some advantages of ultrasonic sensors are they are small and relatively low cost. They are also accurate regardless of the color or composition of the reflecting surface. The surface should only be solid enough to allow sound waves to reflect back – so it couldn't fly over sound absorbing materials such as sponges. But there are a few disadvantages as well however. One of which is that temperature and humidity can alter the accuracy because sound will travel at different speeds when these factors change. Also due to the properties of sound waves, echoes can be picked up when the wave bounces off multiple objects and arrives back to the sensor after the initial reflection was already received. These are called "ghost echoes" and can confuse the sensor. But overall the ultrasonic sensor seems to be the best overall fit for the UARC after comparing the pros and cons of each sensor. Even if high temperature and/or humidity affect the sensor, these characteristics will be fairly uniform where the UARC will fly and shouldn't vary greatly as the UARC moves. Ghost echoes shouldn't be a problem since the UARC will be flying over mostly flat surfaces as well.

When comparing the models of ultrasonic sensors, the group needed a small, low cost, easy to use type of sensor, especially since the UARC requires 2 of these sensors. After researching several models, the group narrowed down the selection to MaxSonar's EZ0 and Parallax's PING)))™ ultrasonic sensors based on online reviews and articles. The EZ0 operates off 2.5 – 5.5VDC at 2mA current draw and can detect objects from 6 inches to 254 inches (or a little over 21 feet). It also has 3 types of outputs: serial digital, analog voltage and PWM. Readings can be taken every 20ms as well. The EZ0 retails for $27.95 from the Superdroid Robots website. Alternatively, the Ping operates off 4.5 – 6VDC at 20 – 35mA current draw and can detect objects from 0.75 inches to 10 feet. The Ping has only 3 pins, which is nice, but it will only output a PWM signal. The Ping is larger than the EZ0 and retails for $29.99 at the Parallax website. After comparing the sensors, the group decided to use the MaxSonar's EZ0 based on its smaller size and weight, cheaper price, less current and voltage consumption, various types of outputs and greater range.

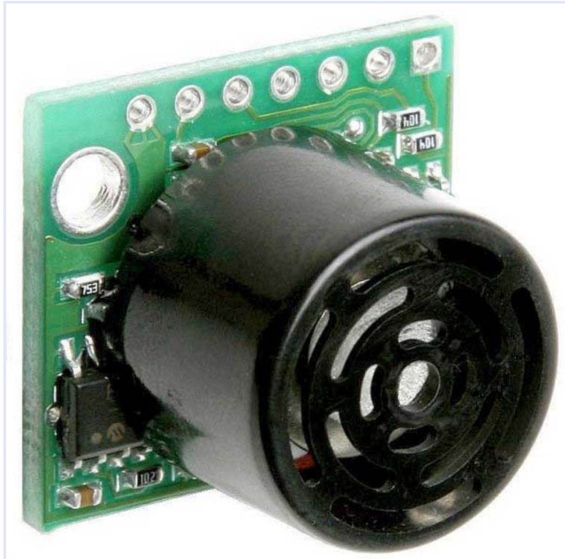


Figure 22: An ultrasonic sensor uses sound emitting waves to determine distances.

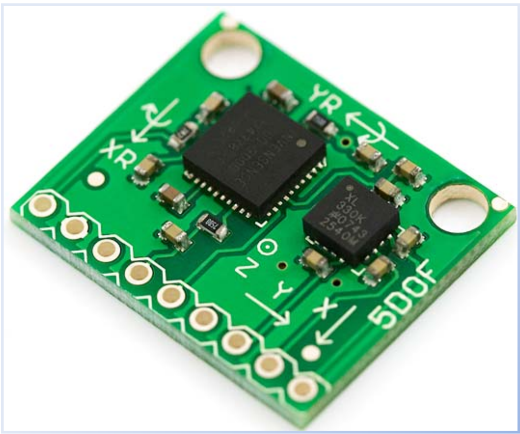
4.4.5 Thermal Intelligence

The UARC will need a way to stabilize itself when it's flying or hovering. One way this can be accomplished is with thermal intelligence. Thermal intelligence uses infrared sensors or thermopiles to detect the temperature difference between the earth and the carbon dioxide in the atmosphere. It uses this signature as a reference to keep itself flying level with respect to the horizon. FMA Direct produces a unit called "Co-pilot" that uses this technology for around $70. It senses the horizon then sends signals to the servos of the helicopter or airplane that will make it level itself out regardless of the initial position. The obvious limiting factor for this system is it can only be used outside in a clear area unobstructed by buildings or trees. The UARC should be able to fly indoors as well as outdoors and it may fly in areas that don't have a clear view of the sky, so this technology cannot be the sole method for flight stabilization. So since this isn't practical for the UARC in real-life applications, the group realized that it wouldn't be practical for the project.

Figure 23: FMA Direct Co-pilot device. Courtesy of FMAdirect.

4.4.6 IMU-with 5 Degrees of Freedom

Methods of combining sensors have been considered to save money on parts. An IMU unit was found at Sparkfun.com that incorporated 3-axis accelerometer and 2-axis gyroscopes. By combining the amount of sensors on one standard PCB board, funds were saved than buying individual parts. This particular sensor saves no time in programming, but does house 2 sensors in one. The breakout board uses the AXDL330 accelerometer chip and IDG300 gyro chip. Figure 24 shows the board which is 0.75 x 0.9 inches in length.



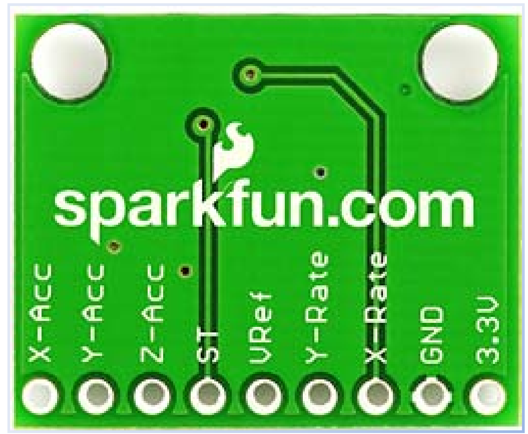


Figure 24: An IMU with 5 degrees of freedom. This one incorporates a triple-axis accelerometer and roll and pitch gyroscope.

The AXDL330 chip outputs analog voltages that are proportional to acceleration. It can measure the static acceleration of gravity as well as dynamic acceleration as a result from motion, shock, or vibration. The board already incorporates the decoupling capacitor of 0.1μF to filter noise out. The bandwidth has already been set by adding a capacitor of 0.1μF which sets it to 50 Hz on the x, y, and z component pins. Furthermore, because the resolution is dependent on bandwidth capacitance chosen, there is little that can be done to change this since the manufacturer has designed this package already. The chip itself has relatively low power consumption of 320 μA. Furthermore, exposed temperature on the sensor has very little impact on operation and accuracy and can have an error of +/- 3*g* from -25° to 70° C.

The IDG300 chip rests on the board which has power conditioning already incorporated. The power supply regulator has a resistor of 2.2Ω and a decoupling capacitance of 0.1μF before passing power to the chip. Thereafter, pins 14, 29, and 34 are decoupled with 0.1μF again since they receive power supply voltage. Compensation capacitors are added to pins 8 and 23 for the amplitude control loops and keeps precise measurements over the operating temperature. The manufacturer recommends an external low-pass filter to filter out high frequency noise by using a 750Ω for pins 3 and 28 and then shunted by a 0.1μF capacitor. Fortunately, this breakout board already includes this procedure and thus eliminates the need to spend extra time and funds to implement this.

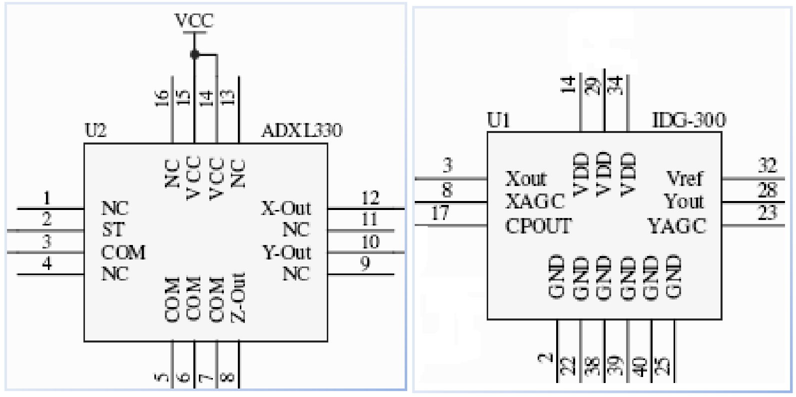


Figure 25: The pin layout for AXDL330 and IDG330 chip found in the IMU 5-Degrees of freedom.

There is another choice to make programming easier and have combined sensors on a breakout board. Figure 26 shows another setup that can be considered. The Honeywell HMC6343 incorporates 3-axis magnetometer and 3-axis accelerometer on a single IC. The major difference in this device is the tolerance to have this chip not set in a flat field. In other words, the magnetometers should be set on a flat surface in order to interact. Because the Honeywell chip is a solid-state device, it will generate readings accurately even if held at a tilt. Because the UARC may find itself in space that may not always be flat, this could be of great value to keep accuracy to avoid crashing. At the core of the chip sits a PIC processor that does all the calculations and sends out the raw information via the 4 pins.

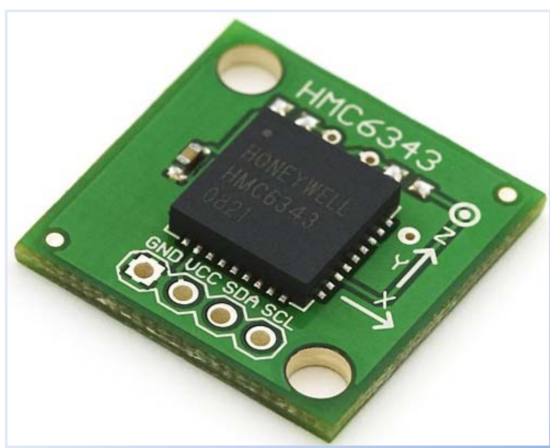


Figure 26: The Honeywell HMC6343 incorporates 3-axis magnetometer and 3-axis accelerometer all in one IC.

As always, the power supply should have a 1μF to filter out noise before connecting it to the V_{CC} on the chip. The manufacturer recommends that the trace lines surrounding this chip not exceed 10 mA of current due to inductive coupling properties that might give interference to the device. As said earlier, the chip orientation does not matter because it is a solid-state device. However, it must be set programmically as to what direction it will be mounted. For tilted placement, it must be offset by the degree in the program to get the correct reading.

Further research was taken to determine why the placement must be set. It appears the placement is important to set the chip readout, but does not affect the readings when the UARC is tilted. This reassured the idea of constant readouts regardless of orientation the UARC will encounter. Pullup resistors are required on pins labeled SCL and SDA of 10kΩ before being connected to the microcontroller. Its output is 8-bit long data stream and can send out information every 200 milliseconds. Furthermore, a hex string of 0x32 and 0x50 tell the chip to gather heading and tilt readings. Next, if one sends a hex string 0x33 and tell the microprocessor to listen in on port SDL it will send heading, pitch, and roll byte pairs. These will all be in 2’s compliment binary format. Although this sensor is being considered, it was determined that 3-axis magnetometer is not needed in this application.

4.4.7 GPS Module

GPS would allow UARC to receive its positional coordinates from satellites. The GPS module does a lot more than just position, but in the case of UARC that is all. A GPS can also tell atomic time, bearing, velocity, altitude, and date. With the use of at least 3 satellites it can determine longitude and latitude within a few meters tolerance. If four satellites are used it can determine altitude also.

There was a couple of GPS modules looked into for UARC. The first was called the 20 Channel EM-406A SiRF III Receiver with Antenna. This device is lightweight, powerful, and low cost. Together with the WAAS system it can reach accuracies of 5 meters. The power consumption is slightly high though, operating at 70mA between 4.5 and 6 volts. It only weighs 16g and claimed to be the smallest GPS module available. Figure 27 shows a picture of it, courtesy of SparkFun Electronics.

Figure 27: 20 Channel EM-406A SiRF III Receiver with Antenna

The second module looked into is the 20 Channel GS405 Helical GPS Receiver. It is similar to the EM-406A, but has a couple extra features that are very convenient. For one it has a filtering antenna that helps eliminate RF interference. It also is very low power, operating at 75mA at 3.3 volts. It can be very accurate, up to less than 5 meters, with DGPS correction.

Of course with all these better features it will cost about 30 dollars more. It is also more appealing to the eye as shown below.

Figure 28: 20 Channel GS405 Helical GPS Receiver

4.4.8 Infrared

The idea of showing the capability of the UARC came about doing reconnaissance missions. The vehicle would capture a pre-defined destination and capture video and send back to control tower. The perplexing idea of how to set this defined location was rather difficult. It was then implemented to design an emitting waveform such that the UARC can capture this destination. A cheap method would have to be constructed such that the idea doesn't exceed the budget. The idea of an emitting infrared signal can do just the job for the expectation. If target areas were fixated with an emitting signal, a GPS coordinate could be given to the UARC to scout the area and snap video. However, design specifications would have to incorporate an infrared receiver onboard to capture this signal. The concept would prove successful in the event the UARC would locate this beam and 'close in' on the target. For example, if coordinates are given to the UARC to search certain coordinates for an emitting beam it would hover over to the area and listen in on the infrared receiver for interception. Once found, it would then zero in on the source and descend onto the target and take video signal and send back to control tower. A simple PCB board with power supply generated from AA batteries can power an infrared receiver. Figure 29 shows the infrared receiver and transmitter that would be implemented.

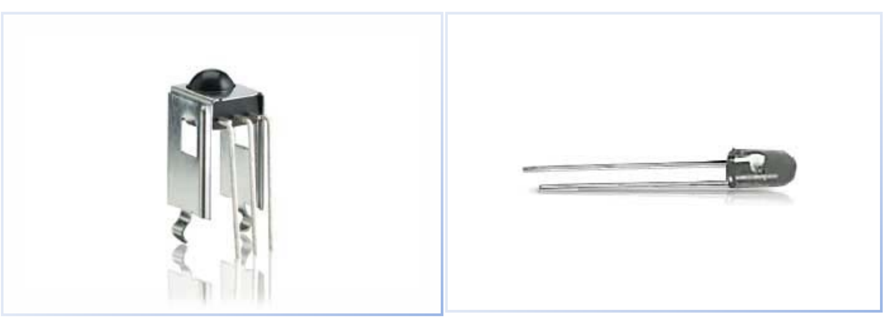


Figure 29: A radio frequency receiver (left) and infrared transmitter (right).

The infrared transmitter operates at 1.2V, and rated at 100 mA. The output that would be expected is a 940nm wavelength. Two bulbs would be incorporated to generate pulses that would be sensed at least 10 feet in the z-axis. This would ensure that the UARC can pick up the signal at the distance it will hover. The infrared receiver operates between 2.4 to 5.5V, and is rated at 0.6A. The allowable passband wavelength is 940 nm with a tolerance of 50 nm to be able to detect a signal. The elliptical lens helps capture as much of the signal in a unidirectional area as possible. A filter to block out noise is incorporated and therefore only operates at 38 KHz signal.

An ultrasonic sensor and receiver was also considered for this application. Both the receiver and transmitter operate at 10V with a tolerance up to 20V before damaging the components. Furthermore, a crystal oscillator will be needed to be implemented to have a center frequency of 40.0 KHz. The idea would be to listen in on ultrasonic ranges until detection was noticed. However, with this implementation the signal would bounce off other objects and give to many ghost echoes and have negative effects in zeroing in on targets. Voltage was also considered as a major downfall of choosing this method. Many batteries would have to be used to power up the device found on the floor level. The data sheets found for this device did not provide the consumption of power. Nevertheless, the importance of this project was not to demonstrate this feature and thus much effort should not be placed in this design.

4.5 Microcontroller

4.5.1 Chipset

There are many different available microcontroller chipsets to choose for the design of UARC. One potential candidate is of the Ti MSP430 family from Texas Instruments. Another possibility is the PIC microcontroller family from microchip. The prime candidate for UARC is developed by Coridium and called the ARMmite.

The original chipset thought to be used was the Ti MSP430. It was recommended, and since not much research had been done yet, the group decided to purchase one for each member. This was to get a microcontroller and familiarize with it. The MSP430 family has a wide array of controllers. The 20-pin MSP430F110 is on the low end of them while the 100-pin MSP430F149 is higher end. The one purchased for use was the eZ430-F2013. It is nice and portable with its own onboard USB development board.

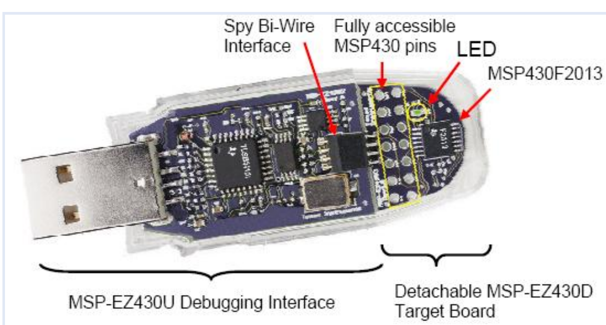


Figure 30: eZ430-F2013 microcontroller

This includes 16 MIPS performance, a 16-bit Sigma Delta A/D converter, 16-bit timer, Watchdog timer, and more. It is powerful enough for UARC, but has a couple crucial downfalls. First of all there is hardly any documentation out there aiding in programming with this chip. Secondly it does not contain a PWM module, which can be worked around, but it is easier just to get a chip with at least one PWM on it.

Now the PIC family has been around since the beginning of microcontrollers. They are very popular and there is numerous tutorials and free code written for them. The PIC seems like they ideal choice for UARC. They come in 8, 16, and 32 bit sizes with all the features like timers, EEPROM, multiple A/D converters, and more. Below is a picture of a Microchip PIC24HJ32GP202.

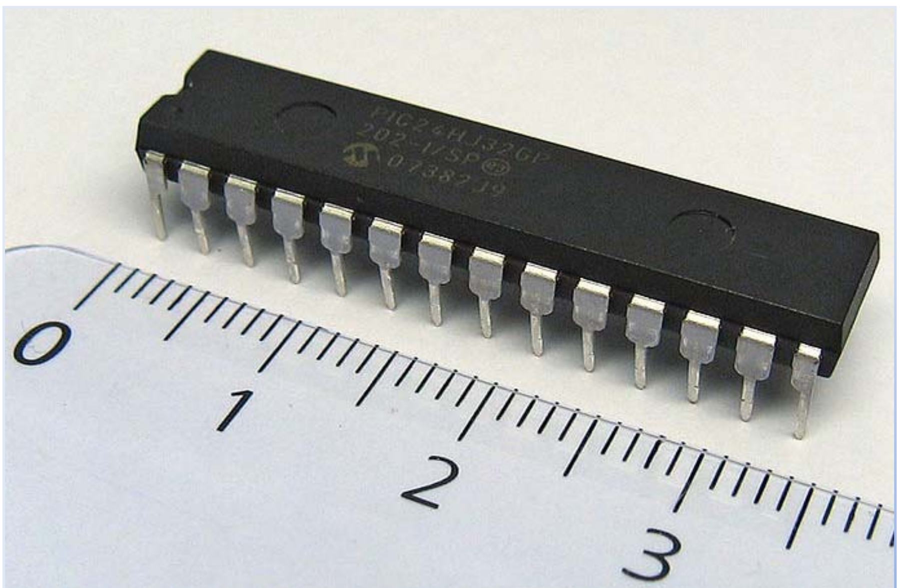


Figure 31: Photo of a Microchip PIC24HJ32GP202 courtesy of User:Acdx

Up until being referred to the Coridium ARMmite microcontroller the PIC μc was prime choice. The PIC was also referred by an engineer at Castle Creations as “slow and stupid”. The Coridium ARMmite microcontroller is perfect for the task at hand. In figure 32 below there is a picture of it.

Figure 32: Coridium ARMmite microcontroller

This controller runs very fast at 60 MHz and has many features. It is easily programmed through the USB interface and has plenty of I/O ports. There are also 8 10-bit A/D channels, 8 hardware PWM channels, and 32K of flash memory. The board is a little big, but still maintains a light weight. With all of its benefits the weight addition is miniscule.

4.5.2 Development Board

The reason for the big area of the Coridium ARMmite is that it has its own development board. It enables easy programming and uploading through the USB port attached. There is also plenty of room on the port area to attach any external IC's needed directly. For the PIC microcontrollers a development board must be created by the group or purchased separately. The price range varies and there are many online schematics to develop one solo. Figure 33 below shows an 18-pin development board sold from sparkfun.com.

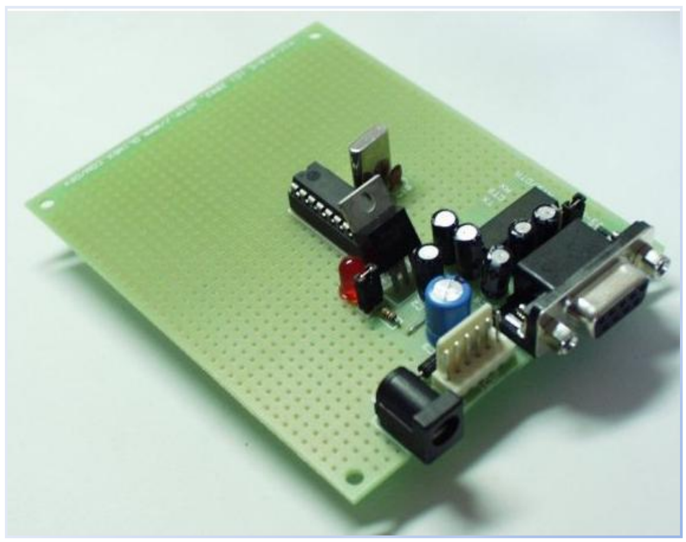


Figure 33: 18-pin PIC development board courtesy of SparkFun.com

As with the Coridium ARMmite, the Ti MSP430 comes with the development board attached to it. The difference is that it is detachable.

4.5.3 Software Programming

All microchips have their own assembly code that they can be programmed in. The PIC microcontrollers can be programmed in C or basic. The Coridium ARMmite also can be programmed in C or basic. The Ti chip has very useful development software which enables the programming in any language between its assembly and C and it gives one both. When trying to create a basic program that flashes a led for the Ti it was very tedious. The intimidation was a discouragement towards the use of this chip.

4.6 How the Draganflyer works

A great reference was found that can help with the construction of the UARC. Articles have surfaced in IEEE journals that explain, modify, or dissect the Draganflyer. In the center of the vehicle lie a PCB board that has remote control system, stabilization system, and the power systems all built in. With a 4-channel FM system controller, as found in many remote control vehicles, it controls throttle (motor speed), roll, pitch, and yaw. To decode the signal, the vehicle has an onboard receiver in which it employs an oscillator and RF crystal to determine operating frequency. The attraction to this apparatus that separates it from the rest is the quad-motor design. With ordinary helicopters, it uses a large wing blade to lift it exposing it to collide with something. Whereas, the Draganflyer has quad-rotors which means smaller blades and enclosures that protects them. Another neat feature is the construction of the blade itself. Because the blades are made from flexible plastic, air drag forces distortion of the pitch angle and thus produces more lift force with high rpms. One motor has the equivalence of ¼ lift force and thus the total lift force is a function of the four motors. As noted previously in the research, one motors mechanical behavior will differ from the same manufactured motor. Since no two motors are alike, the rotational torque produced is the difference between the lift forces from one motor to the others. Testing will have to be done to make sure the lift force of one motor is calibrated and notated when programming is constructed. Since the pivot on the motors is fixed, unlike normal helicopters, the Draganflyer can only vary rotor speed for turning.

Figure 34: The Draganflyers frame, constructed of carbon fiber, encloses and protects the motor.

To power the motors, 4 speed controllers are tied directly to the motors and communicate between the PCB board and motors. The speed controllers essentially use a FET to act as a valve of voltage to the motors. A small voltage will turn the blades at slow RPMs while turning up the voltage will ramp the blades to spin. However, turning ability is not tied directly by spinning one motor faster and slowing the opposite motor slower. While the total thrust must remain the same, other forces act upon it during this transition. If say the Draganflyers 'left' motor is sent left voltage, naturally this will steer the vehicle to roll to the right due to the imbalance of lift forces from the left and the right motors. Now dynamic forces of torque come into play and upset the yaw the vehicle will travel. Such movements will make the vehicle translate to the right, as the rotor forces are now directed toward the left as well as down. This spinning out of control effect, called the "P factor" has been neutralized in the Draganflyer by counter spinning rotors in the opposite directions. Please refer to the section 3.7 titled 'Dynamics' for the reason behind this technique. It is noted that this counter-acting spinning rotor produces significant stress on the frame due to the torques produced. Many report that screws that hold up the frame together to become lose from the vibrations and torques working together. Since the flight time will be minimal, this will be checked upon the starting of the next flight schedule.

Controlling the rotor speeds in space gives it 6-degrees of freedom. Because of this, programming code has been set in spherical coordinates for accuracy. The control electronics handle three functions: receipt of the servo commands from the radio link, closed loop stabilization of roll, pitch and yaw rates, and mapping commands from the spherical coordinates to motor speeds. The commands coming from the controls are pulse width modulated, encoded and transmitted to the vehicles. Once received, it is demodulated, interpreted, and carried out to the motors. The closed loop stabilization pattern is carried out by signals generated by the solid-state gyroscopes on board, accounts for external disturbances such as wind, and assures rotation of the craft is proportional to commands given by the operator. Mapping of commands is done by calculating sum and differences of lift forces. For the commands on the UARC, it will be done systemically. Therefore, 2 of the 3 tasks are eliminated since the need for an operator is taken out of the equation and are replaced by sensor related tasks.

The Draganflyers stabilization system comes from piezo gyros that are placed in the x, y, and z coordinates to have input sensing. In essence, these piezo sensors give feedback to the microcontroller letting it know when movement has occurred. If it senses the vehicle is banking to the left, it will send a signal to speed up the left motor to have stability. During manual operation of making a right turn let's say, the microcontroller overrides this sensor and delays any signal input from that sensor. It is important to note that this small gap in ignoring the sensor does not lose the stabilization of flight because the other sensors are still giving feedback to keep it afloat. During turning, there is significant cross coupling that occurs between the forces produced by the thrusters. Therefore, it is impossible to maneuver the vehicle to turn to the right while hold the craft in the horizontal axis.

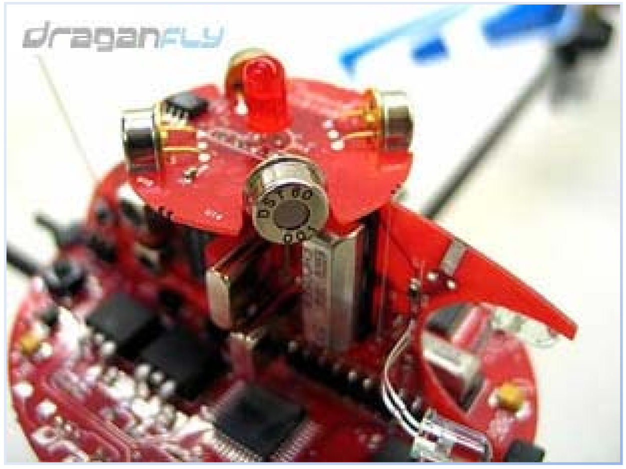


Figure 35: The Draganflyer uses Piezo gyros to sense angular rotation and gives feedback to the microcontroller to adjust rotor speed.

What oppose motion are 3 forces: gravity, inertia, and air drag. Considerable energy will be drawn to counteract gravitational force and thus diminishes flight time. Ironically, it has to overcome all the weight onboard including the battery, the heaviest component. Inertia will oppose linear and rotary acceleration which helps to stabilize motion. The battery housing is located below the center cross axis to lower center of gravity of the vehicle to counteract inertial forces which introduce unwanted pitch and roll movements.

Understanding how the Draganflyer works will assist in the development of the UARC. Careful consideration will be taken for the outside forces one will encounter such that crash can be avoided. Because the sophistication of the Draganflyer is considerable compared to the project, much of the same controls will be applied to achieve flight. With proper simulation and testing, the same results can be achieved in shorter amount of time.

4.7 Building a Test Prototype

4.7.1 Purpose of 2-D Test Assembly

The purpose of building the 2D test assembly is for that of gaining experience in the dynamics and controls that need to be implemented for the final design of UARC. This assembly will simplify the project significantly yet still define the process and methods to be accomplished. First of all the number of actuators is reduced to two, being the two motors that originally came with the Draganflyer frame. Secondly the number of output variables is reduced to two. These variables are Z and θ which represent the height and the pitch angle about the y axis respectively. Figure 36 below shows the schematic of what the contraption will look like and its free body diagram.

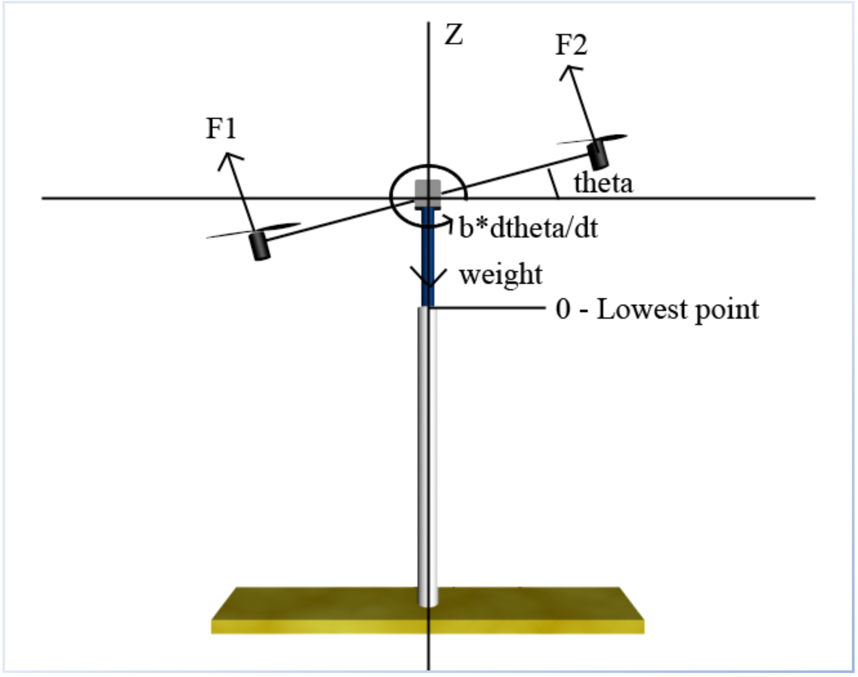


Figure 36: Free Body diagram of 2D test prototype.

Another attribute that this will simplify is the sensor readings. The 2D test assembly will not use sensors. This means it will be cheaper to make, lighter, and less code and hardware intensive. The rotational signal will not have to be filtered or calibrated for noise and drift. It will instead retrieve the angle through the voltage drop of a potentiometer. The pot will create a nice linear voltage that can be input into the computer and interpreted accordingly. Since the signal is given so nice there is no need for any system in between to filter it and create unnecessary delay.

4.7.2 Motors

To even furthermore simplify the 2D prototype it will use the brushed Mabuchi motors that came with the purchase of the frame instead of the upgraded brushless motors acquired for the final design of UARC. The benefit of this is that there is no need for speed controller hardware (ESC) to control the motors. This eliminates the possibility of hardware failure of the ESC and the unknown system characteristics of it. It is now just a matter of applying straight voltage to the motors to control their speed.

4.7.3 Building the Contraption

Building the contraption was fun, to say the least. It required a lot of eyeing things out and luck. There were originally two ideas derived for the design of it. The first was that of attaching a motor to the end of a light wood and have it hinge to allow for vertical movement. The motor on the end would just act as bearings for the rotation of the frame. This concept is shown below from a side view in Figure 37.

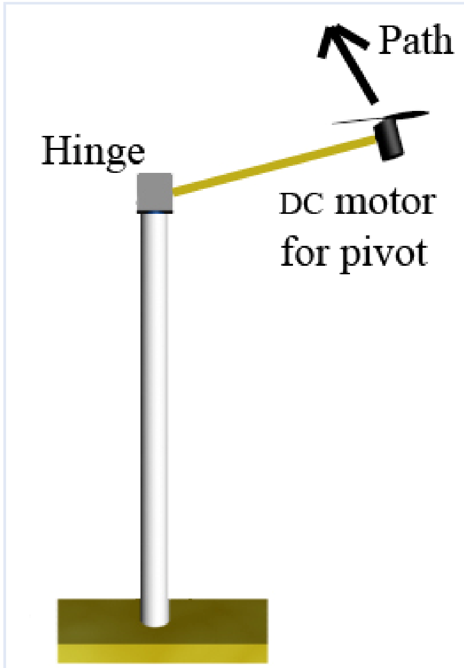


Figure 37: wood 2d original design.

The benefit of this design is that it would allow for frictionless rotation, which is the case in UARC, and there was no need to worry about restricting yaw rotation about the z axis, since the base would be rigid. A disadvantage was that the vertical flight path would be curved and not straight like desired. The biggest disadvantage and main reason this design could not be implemented was the distribution of weight. Not only is the piece of wood creating uneven weight, but also a heavy motor is throwing off inertia and dismantling the beautiful symmetry that simplified the dynamics.

The second idea was to have a frictionless pivot attached to the end of a pipe and have that pipe slide up and down inside another pipe with a greater diameter. This allowed for straight vertical motion while keeping the rotation free. This was the design which would be implemented. Originally the pivot was still going to be attached to a motor, but that would throw weight distribution and inertia off. Instead of that, a set of skate wheel bearings would be used to create a light weight and centered pivot for the frame to rotate on.

The first step was to decide what to use for the pipes. Ultimately PVC would be used because of its light weight and lesser friction. The next step was the design of a base to hold the pipes and frame. It is compiled of a piece of wood with a hole in it where a short pipe is screwed into it. The next piece of pipe is then inserted into it and pinned to hold it in place. The complete base schematic with reference names is shown in figure 38.

Figure 38: Base with other pipe2.

As one may notice, there is a long cut on both sides of Pipe2. This will be the guide for Pipe3. It was done by clamping the pipe and routing it with a machine. Although not perfectly executed it does the job. It was also sanded to decrease friction even more. Pipe3 will be added and pinned to prevent yaw rotation and enable flight height restriction. The height restriction is mainly a safety feature so that the thing doesn't fly out of control and kill somebody. Pipe3 is inserted as shown in figure 39 below.

Figure 39: Pipe3 in Pipe2.

Now the hardest thing to do was design the pivot on Pipe3. It would involve the manipulation of lightweight aluminum and metal to create casings for the rod and bearings. Luckily the Draganflyer frame center hole fit nice and snug on to the rod which held the bearings on the skates. Originally the potentiometer was going to just attach to the end of the rod and get its readings that way. It was soon realized that it would be better to just use the long potentiometer shaft as a rod and attach the frame to it. This would be way easier to accomplish and less likely to slip or give inaccurate readings. There would also be no need to create another piece to attach the pot to. With the use of a metal band saw the pieces shown in figure 40 were cut out.

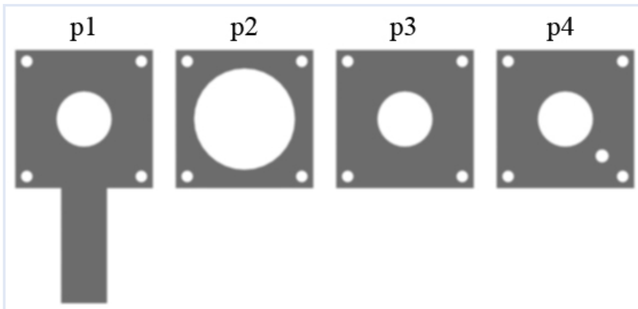


Figure 40: Aluminum and metal cut outs for bearing and rod support.

Two sets were made of the cut outs. Part p1 was attached to Pipe3 with rivets. Part p2 was set to the thickness of the bearing and held them in place. Part p3 was on the outside of the bearings to also aid in their placement. Part p4 did the same as p3, but with an extra hole so the potentiometer can lock in place and prevent it from turning. All the parts had to be tapped and died so that the pieces could be locked into place with a screw and lock washer. The actual completed piece with the frame is shown in Figure 41.

Figure 41: Complete 2D design.

With the prototype complete tests can be run. Labview will interpret signals and enable an easy form of control. Through trial and error the correct coefficients and variables can defined. This will make the task of programming a microcontroller that much easier.

4.7.4 2-D Flight Dynamics

Based on the free body diagram given a set of flight dynamics can be derived for the 2D prototype. This is done through Newton – Euler formulization. Since there are only 2 degrees of freedom the equations are in the form of standard second order.

The equations for Z directional movement are derived first in four easy steps.

$$\sum F_z = m\,a_z = m\,\ddot{Z}$$

$$F_1 \cos\theta + F_2 \cos\theta - mg - c\,\dot{Z} = m\,\ddot{Z}$$

$$T = \left(F_2 + F_1\right) \cos\theta$$

$$m\,\ddot{Z} = T - c\,\dot{Z} - mg$$

The coefficient c is the dampening factor induced by the props as they increase speed. The variable T represents the total thrust from the two motors. Depending on the error signal obtained through the control system the individual thrusts can vary, but the total will stay constant.

Now for the Rotational angle theta the equations can be derived just as easy.

$$\sum \tau = I\alpha = I\ddot{\theta}$$

$$F_2\, l - F_1\, l - b\dot{\theta} = I\ddot{\theta}$$

$$R = F_2 - F_1$$

$$I\ddot{\theta} = Rl - b\dot{\theta}$$

The coefficient b is the dampening factor resisting torque. It should actually be pretty high considering the resistance the potentiometer induces. Test and simulation will need to be done to determine the proper values for b and c. Ideally in UARC there is free rotation and b would be zero. In the rotational equation, R represents roll and is the difference of F_2 and F_1. Again the difference of these actuators will stay constant, but the individual forces will vary with the error signal given in the feedback control system. These equations are modeled and tested in the simulation section.

4.7.5 Transition to Sensors

When the time comes to switch to actual sensors the complications will start arising. First of all the contraption will have to be modified to hold the sensors in their rightful place. This will entail the use of a couple metal pieces screwed on to the top that can hold the gyro sensor and enable it to rotate with the frame. This will do away with the potentiometer. The original skate bearing rod will be placed back as the pivot. Not only does the gyro sensor need to be filtered and interpreted, but now that the potentiometer is gone there will be a lot less damping from friction. Therefore the dynamics will be quicker and more sensitive to thrust force input. This is more realistic as compared to the actual dynamics of UARC.

As for height measurement there will be an ultrasonic sensor attached by a light weight wood hanging out a foot or so as to not get interference from the base. Refer to figure 42 to visualize this concept.

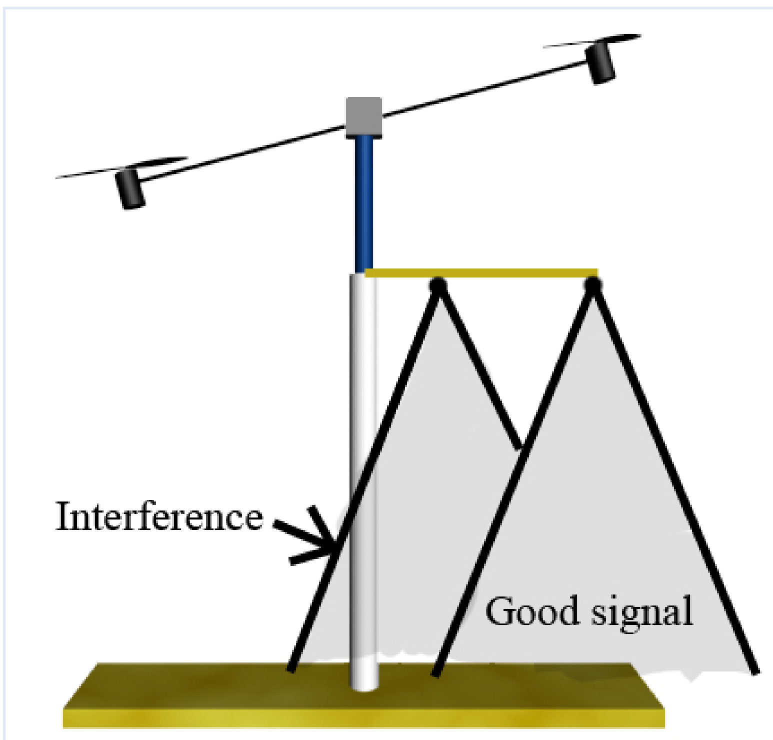


Figure 42: interference ultrasonic sensor

If the sensor is attached any other way it will not get accurate readings. Adding all this stuff will make the prototype a little heavier. The battery and other components will not be on the vehicle though so the weight difference will be negligible.

Now the goal is to have the 2D prototype now work as good as the one before. Adjustments will have to be made to accomplish this. For example, a kalman filter will need to be implemented to decipher the sensors signals. This adds delay and is not an easy task for newbies.

4.7.6 Transition to the Draganflyer

The purpose of the 2D design is to familiarize the group with the steps and hurdles that will come up as the final project is being designed and built. The same concepts used in controlling the 2D prototype will be implemented in UARC. Of course there will be more tasks to tackle, but at least there is a familiar path to follow. Hopefully with the experience gained from this project UARC will not lose control and destroy itself on its first flight.

4.8 Conversion to Brushless Motors

4.8.1 Introduction

Once the 2-dimensional test setup is complete and working correctly using the brushed motors, the motors will be upgraded to brushless. The reason for this is because the brushless motors are more efficient than the brushed motors, as explained earlier in this paper. The 2-dimensional setup will use brushed motors only for simplicity while the control system is worked out and perfected. Making the transition to brushless should be fairly straightforward from that point. Then the next transition would be from the 2-dimensional test setup to the actual 3-dimensional quad-rotor.

4.8.2 Pros and Cons

There are some pros and cons involved with making the transition from brushed to brushless motors. The pros are increased efficiency, extended battery life and increased power to weight ratio. Also the motors will last longer and require no maintenance. The EMI emitted by the motors will also decrease using the brushless motors. They are smaller and lighter than the brushed motors as well. The cons are that the brushless motors are more expensive and require expensive speed controllers. Brushed motors have an advantage because they have a linear relationship between voltage input and RPMs, as well as current and torque. This isn't exactly the case with brushless motors. When the conversion is made, the brushed speed controllers will have to be replaced with brushless speed controllers, which will mean added cost.

4.8.3 Speed Controllers

Both the brushed and brushless speed controllers will be controlled by a PWM signal generated by the microcontroller. The only differences will be the how the motors react with the same PWM input – these differences will have to be accounted for. The brushed controller and motor have a linear relationship with respect to voltage and RPMs, and current and torque as mentioned previously. The brushless speed controllers from Castle Creations are programmed to appear to have a linear relationship however it isn't 100% linear. According to an engineer from Castle Creations, most speed controllers won't change much between 10% to say 60% duty cycle. Most of the speed will be in one area, maybe between a duty cycle of 70% - 100%. Their speed controllers account for this and make the output appear linear. The group can model or test and compare the PWM input in relation to the RPMs or thrust generated to adjust for any inconsistencies. This should involve something as simple as changing a constant in the programming. The reward is worth the extra effort.

4.9 Feedback System

4.9.1 Control System

Feedback controls are crucial to the stabilization of UARC. They need to be fast and accurate. The derived subsystems for rotational and translational dynamical control are inherently unstable. It will take a non-linear control method to reach the desired degree of stability. A widely used form of system control is the PID controller. It will definitely be implemented in UARCS feedback systems. Another method researched is the use of integral backstepping. There is also a method called PDF or PDFF control, which is not as popular as the latter, but is a possibility for UARC system control.

The most popular control strategy for motion is nested PI controlling. For decades this method has been efficient and accurate enough for many applications. The PI controllers can be designed and adjusted for maximum control of the individual subsystems in UARC. The downside of traditional PI controlling is its lack of accurate disturbance adjustment.

The integral backstepping method has recently become a popular form of motion control. It's ultimately a juiced up form of the PID controller. It does have some advantages over traditional PI compensation in the sense that it enables more system bandwidth and improved disturbance control. It also considers the cross relation between inner and outer control loops, which greatly enhances control possibilities. The disadvantage is the complexity of the design.

4.9.2 Altitude Control

The first step in control design is to implement an altitude controller. The altitude stabilization will be controlled by the vertical force input u1 derived in section 4-2 Flight Dynamics. For reference u1 is defined again in equation 1 below.

$$u1 = \sum_{i=1}^{4} T_i = b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \quad (1)$$

The integral backstepping method will be implemented. The altitude tracking error and its derivative are defined as,

$$\begin{aligned} e_{a1} &= Z_d - Z \\ \dot{e}_{a1} &= \dot{Z}_d - \dot{Z} \end{aligned} \quad (2)$$

Where $\dot{Z}$ is the altitude rate of UARC and it will be defined as the virtual control. The desired virtual control can be defined as,

$$\left(\dot{Z}_d\right) = c_{a1}\, e_{a1} + K_{a1}\, \Gamma_1 + \dot{Z}_d \qquad (3)$$

Where c_{a1} and K_{a1} are positive constants that determine convergence speed of altitude tracking and $\Gamma_1 = \int e_{a1}\, dt$ is the integral of altitude error. The virtual control also has its own error which is given as,

$$e_{a2} = \dot{Z}_d - \dot{Z} = \left(c_{a1}\, e_{a1} + K_{a1}\, \Gamma_1 + \dot{Z}_d\right) - \dot{Z} \qquad (4)$$

Now the error rate of altitude can be related to the virtual control error by,

$$\dot{e}_{a1} = -\, c_{a1}\, e_{a1} - K_{a1}\, \Gamma_1 + e_{a2} \qquad (5)$$

Now taking the derivative of e_{a2} yields,

$$\dot{e}_{a2} = \left[\left[c_{a1}\left(-\, c_{a1}\, e_{a1} - K_{a1}\, \Gamma_1 + e_{a2}\right) + K_{a1}\, e_{a1} + \ddot{Z}_d\right] - \left[g - \left(\frac{\cos\Phi\cos\theta}{m}\right) u1\right]\right. \qquad (6)$$

The desirable dynamics of $\dot{e}_{a2}$ are,

$$\dot{e}_{a2} = -\, c_{a2}\, e_{a2} - e_{a1} \qquad (7)$$

Now equation (6) will be negative if u_1 is defined as,

$$u1 = \left(\frac{m}{\cos\Phi\cos\theta}\right)\left[g - e_{a1} + c_{a1}^2\, e_{a1} - K_{a1}\, e_{a1} - c_{a1}\, e_{a2} - c_{a2}\, e_{a2} - \ddot{Z}_d + c_{a1}\, K_{a1}\, \Gamma_1\right] \qquad (8)$$

Now stability analysis is performed through Lyapunov theory and the chosen Lyapunov function is,

$$V = K_1 \frac{\Gamma_1^2}{2} + \frac{e_{a1}^2}{2} + \frac{e_{a2}^2}{2} \qquad (9)$$

Plugging in equations (5) and (7) into (9) and taking its derivative yields,

$$\dot{V} = -\, c_{a1}\, e_{a2}^2 - c_{a2}\, e_{a2}^2 \le 0 \qquad (10)$$

4.9.3 Position Control

Now the desired speeds in the x and y directions are defined as $\dot{x}_d$ and $\dot{y}_d$. Now the error between the desired and actual speeds is given as,

$$e_x = \dot{x}_d - \dot{x}$$
$$e_y = \dot{y}_d - \dot{y} \qquad (11)$$

Desired roll and pitch angles described by the error in actual and desired speed are given as,

$$\Phi_d = \sin^{-1}\left(u_{ex}\sin\psi - u_{ey}\cos\psi\right)$$
$$\theta_d = \sin^{-1}\left[\frac{u_{ex}}{\cos\Phi\cos\psi} - \frac{\sin\Phi\sin\psi}{\cos\Phi\cos\psi}\right] \qquad (12)$$

Where,

$$u_{ex} = \frac{K_x e_x m}{u_1} \qquad u_{ey} = \frac{K_y e_y m}{u_1} \qquad (13)$$

K_x and K_y are positive constants and u_1 is vertical force input from the altitude control.

4.9.4 Rotational Control

The rotational subsystem will implement backstepping based PID control using the control inputs u_2 , u_3 and u_4 .

Roll tracking error, with Φ_d being desired roll, is then defined as,

$$e = \Phi - \Phi_d \qquad (14)$$

Now the first error considered in the design is,

$$z_1 = K_1 e + K_2 \int edt \qquad (15)$$

K_1 and K_2 are positive tuning parameters and $\int e\,dt$ represents the integral of roll error.

z_1 is positive definite and its time derivative negative semi definite as,

$$V_1 = \frac{1}{2} z_1^2 \qquad (16)$$

The derivative of (16) is then,

$$\dot{V}_1 = z_1\,\dot{z}_1 = z_1\left(K_1\,\dot{\Phi} - K_1\,\dot{\Phi}_d + K_2\,e\right) \qquad (17)$$

If $\dot{\Phi}$ is considered the virtual control and $\left(\dot{\Phi}_d\right)$ the desired virtual control then,

$$\left(\dot{\Phi}\right)_d = \dot{\Phi}_d - \frac{K_2}{K_1}\,e - \frac{c_1\,z_1}{K_1} \qquad (18)$$

c_1 is a positive constant and can control convergence speed of the tracking loop.

The virtual control also has its own error and is given as,

$$z_2 = \dot{\Phi} - \left(\dot{\Phi}\right)_d = \frac{1}{K_1}\left[\dot{z}_1 + c_1\,z_1\right] \qquad (19)$$

The augmented Lyapunov function and its derivative is then,

$$\begin{aligned} V_2 &= \frac{1}{2}\,z_1^2 + \frac{1}{2}\,z_2^2 \\ \dot{V}_2 &= z_1\,\dot{z}_1 + z_2\,\dot{z}_2 \end{aligned} \qquad (20)$$

Plugging in the respective variables in (20) leads to,

$$\dot{V}_2 = \begin{bmatrix} z_2\left[e\left(K_1^2 + \dfrac{c_1\,K_2}{K_1}\right) + \displaystyle\int e\,\mathrm{d}t\left(K_1\,K_2\right) + \dot{e}\left(\dfrac{K_2}{K_1} + c_1\right) + \dot{\theta}\dot{\psi}\left(\dfrac{J_y - J_z}{J_x}\right) + \dfrac{l}{J_x}\,u_2 - \ddot{\theta}_d\right] \\ - z_1\left[c_1\,K_1\,e + c_1\,K_2\displaystyle\int e\,\mathrm{d}t\right] \end{bmatrix} \qquad (21)$$

The desirable dynamics are,

$$\dot{V}_2 = -\,c_2\,z_2 = -\frac{c_2}{K_1}\left[\dot{z}_1 + c_1\,z_1\right] \qquad (22)$$

c_2 is a positive tuning parameter. Now,

$$\dot{V}_2 = -\,\dot{e}\,(c_2) - e\left(\frac{c_2\,K_2}{K_1} + c_1\,c_2\right) - \int e\,\mathrm{d}t\left(\frac{c_2\,c_1\,K_2}{K_1}\right) \qquad (23)$$

Desirable dynamics ensure negative definiteness of position tracking error, its integration and velocity tracking error.

Equation (23) is negative if,

$$u_2 = \frac{J_x}{l}\begin{bmatrix} -e\left(\frac{c_2}{K_1}K_2 + c_2 c_1 + K_1^2 + \frac{K_2 c_1}{K_1}\right) - \\ \int e\,dt\left(\frac{c_2 c_1 K_2}{K_1} + K_1 K_2\right) - \\ \dot{e}\left(c_2 + \frac{K_2}{K_1} + c_1\right) + \ddot{\Phi}_d - \dot{\theta}\dot{\psi}\left(\frac{J_y - J_z}{J_x}\right) \end{bmatrix} \tag{24}$$

As one can see (24) is a PID controller with gains given as,

$$P = \frac{c_2}{K_1}K_2 + c_2 c_1 + K_1^2 + \frac{K_2 c_1}{K_1}$$

$$I = \frac{c_2 c_1 K_2}{K_1} + K_1 K_2 \tag{25}$$

$$D = c_2 + \frac{K_2}{K_1} + c_1$$

Larger values of c_1 and c_2 makes the derivative of the Lyapunov function more negative, in turn making the regulation dynamics faster.

u_3 and u_4 can similarly be computed and are given as,

$$u_3 = \frac{J_y}{l}\begin{bmatrix} -e\left(\frac{c_4}{K_3}K_4 + c_4 c_3 + K_3^2 + \frac{K_4 c_3}{K_3}\right) - \\ \int e\,dt\left(\frac{c_4 c_3 K_4}{K_3} + K_3 K_4\right) - \\ \dot{e}\left(c_4 + \frac{K_4}{K_3} + c_3\right) + \ddot{\Phi}_d - \dot{\theta}\dot{\psi}\left(\frac{J_z - J_x}{J_y}\right) \end{bmatrix} \tag{26}$$

$$u_4 = \frac{J_z}{l}\begin{bmatrix} -e\left(\frac{c_6}{K_5}K_6 + c_6 c_5 + K_5^2 + \frac{K_6 c_5}{K_5}\right) - \\ \int e\,dt\left(\frac{c_6 c_5 K_6}{K_5} + K_5 K_6\right) - \dot{e}\left(c_6 + \frac{K_6}{K_5} + c_5\right) + \ddot{\psi}_d \end{bmatrix} \tag{27}$$

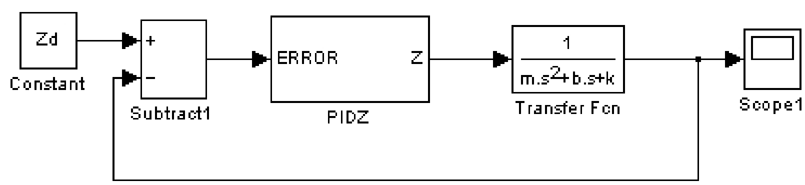
4.10 Simulation

Simulation is crucial to any aspect of engineering. UARC is no different. By using the power of matlab and simulink, UARC can be realized before it is actually physically developed. It will aid in testing the derived equations and controls by setting coefficients and determining accuracy and response. Through trial and error a good starting point can be established for the physical realization of UARC.

The first simulations ran were that of the 2D prototype. Using the dynamic equations derived they can be formed into a simulink model. Figure 43 below shows the equations for translation along the z axis and the corresponding model with step response. Figure 44 shows the rotational equations, model, and step response.

Translational Equations and model 2D:

$$m\ddot{Z} = \left(F_1 + F_2\right)\cos(\theta) - c\dot{Z} - mg$$



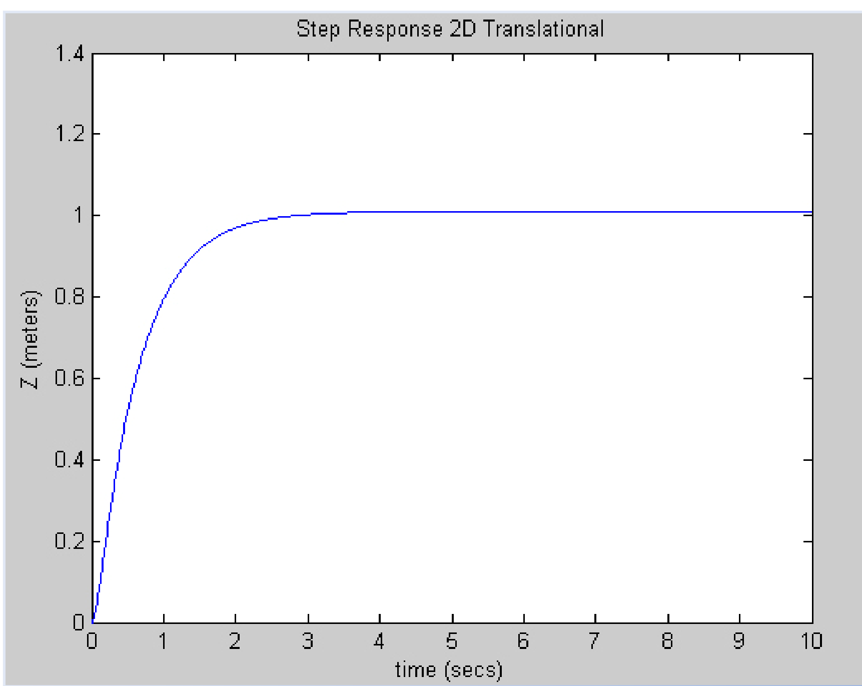
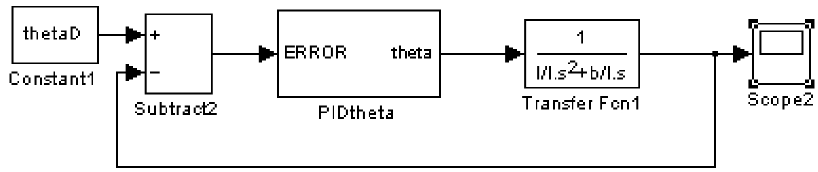


Figure 43: Translational model 2D and step response

Now there is some steady state error, but it is not a big deal when it comes to the Z coordinate so it is ignored.

Rotational Equations and model 2D:

$$I\dot{\omega} = \left(F_2 - F_1\right) l - b\omega$$



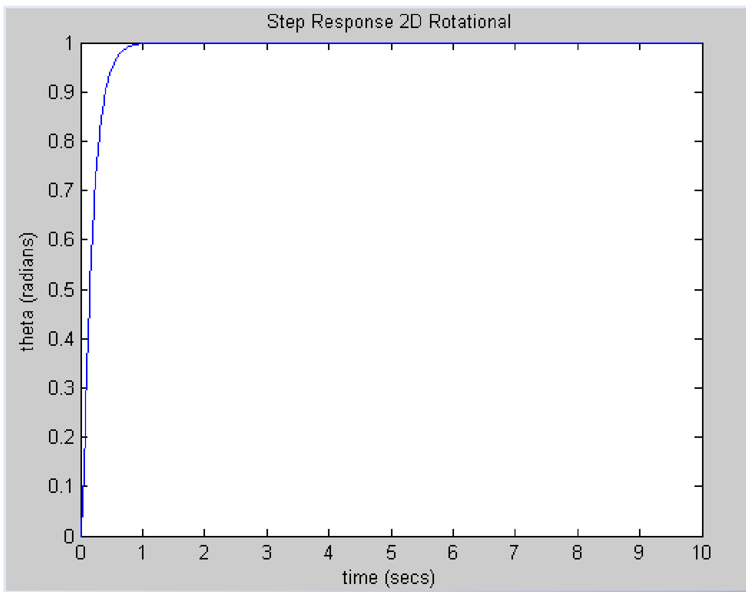


Figure 44: Rotational model 2D and step response

The step response for the rotational subsystem is much quicker due to the nature of rotation. There is no steady state error and minimal rise time to accommodate for this.

Now the real equations for the 3D model get a little more involved, but are still easily modeled in simulink.

Translational Equations and model 3D:

$$m\ddot{X} = \left(\cos\Phi\sin\theta\cos\psi + \sin\Phi\sin\psi\right) u_1$$

$$m\ddot{Y} = \left(\cos\Phi\sin\theta\sin\psi - \sin\Phi\cos\psi\right) u_1$$

$$m\ddot{Z} = mg - (\cos\Phi\cos\theta) u_1$$

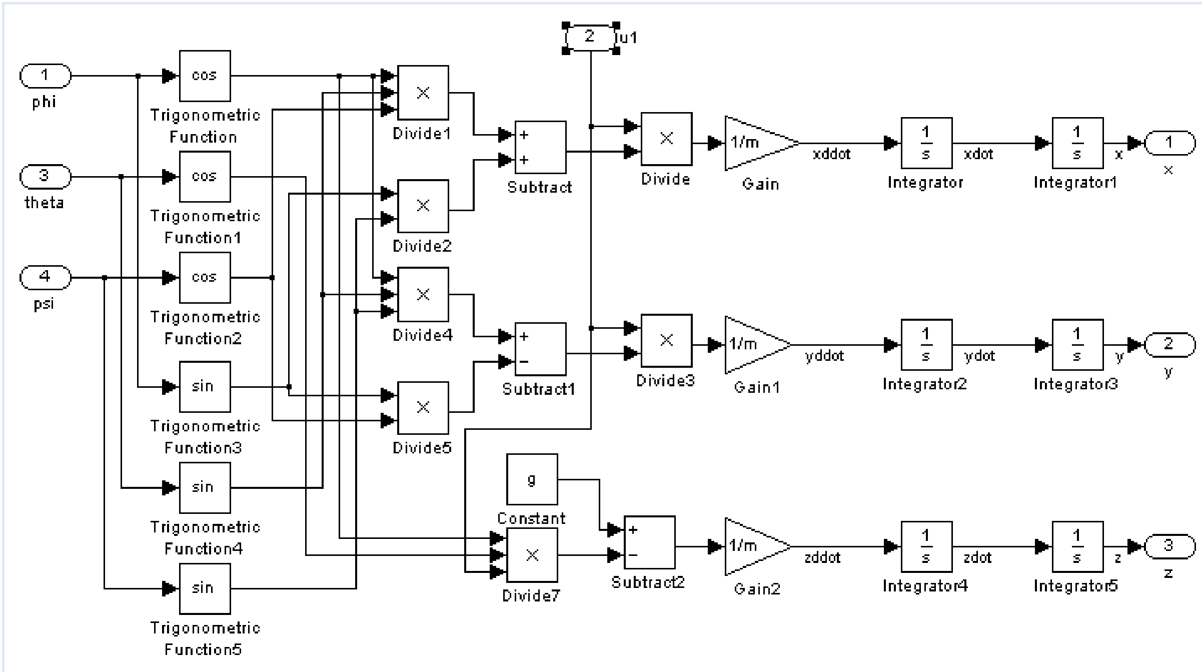


Figure 45: Translational Model

As one can see the inputs to the translational block are Φ,θ,ψ and u_1 . u_1 is the thrust control input signal arriving from the feedback loop. Φ,θ,ψ will be the Euler angles arriving from the rotational subsystem. The system outputs the inertial coordinates of UARC and the velocities for each dimension if need be.

The next subsystem realized is that of the rotational dynamics. These equations were also derived in section 4-2 and are shown again below along with figure 46 which shows the simulink model.

Rotational Equations and model:

$$J_x \ddot{\Phi} = \dot{\theta}\dot{\psi}\left(J_y - J_z\right) + lu_2$$

$$J_y \ddot{\theta} = \dot{\Phi}\dot{\psi}\left(J_z - J_x\right) + lu_3$$

$$J_z \ddot{\psi} = \dot{\Phi}\dot{\theta}\left(J_x - J_y\right) + u_4$$

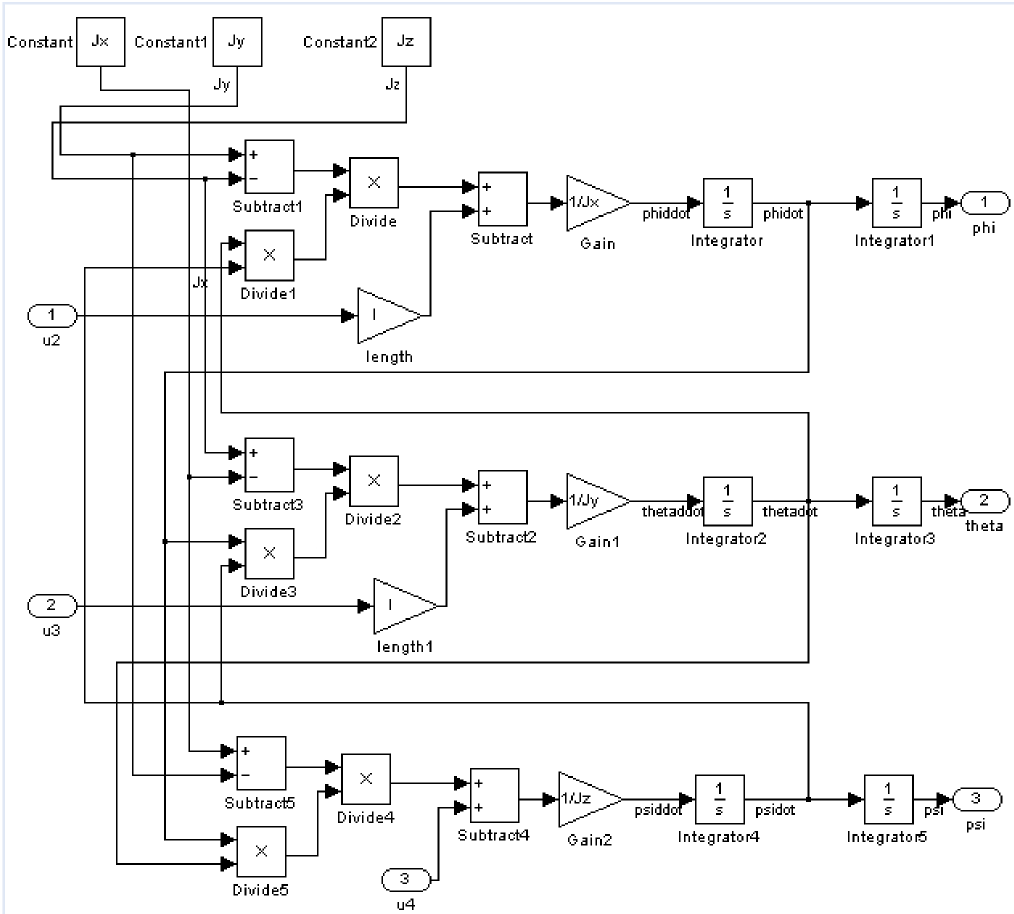


Figure 46: Rotational model

Now that those subsystems are derived they can be connected to the motors and the open loop response can be tested for the system. Below shows the compiled system with a constant voltage applied to each motor and a desired roll and pitch at -1 radians and +1 radians respectively. Figure 48 shows the translational outputs.

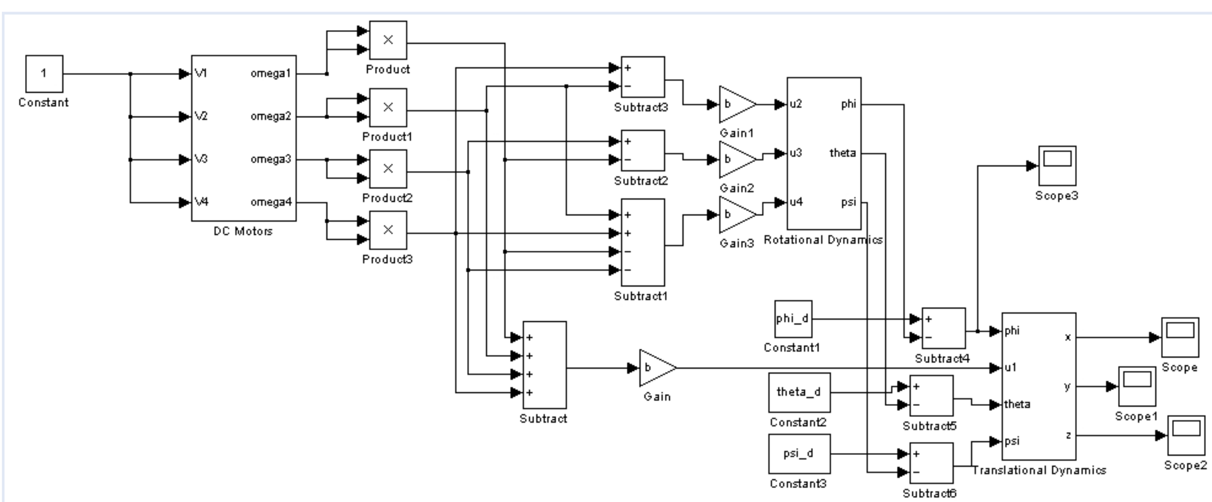


Figure 47: Open loop model of dynamics interaction with motors thrust

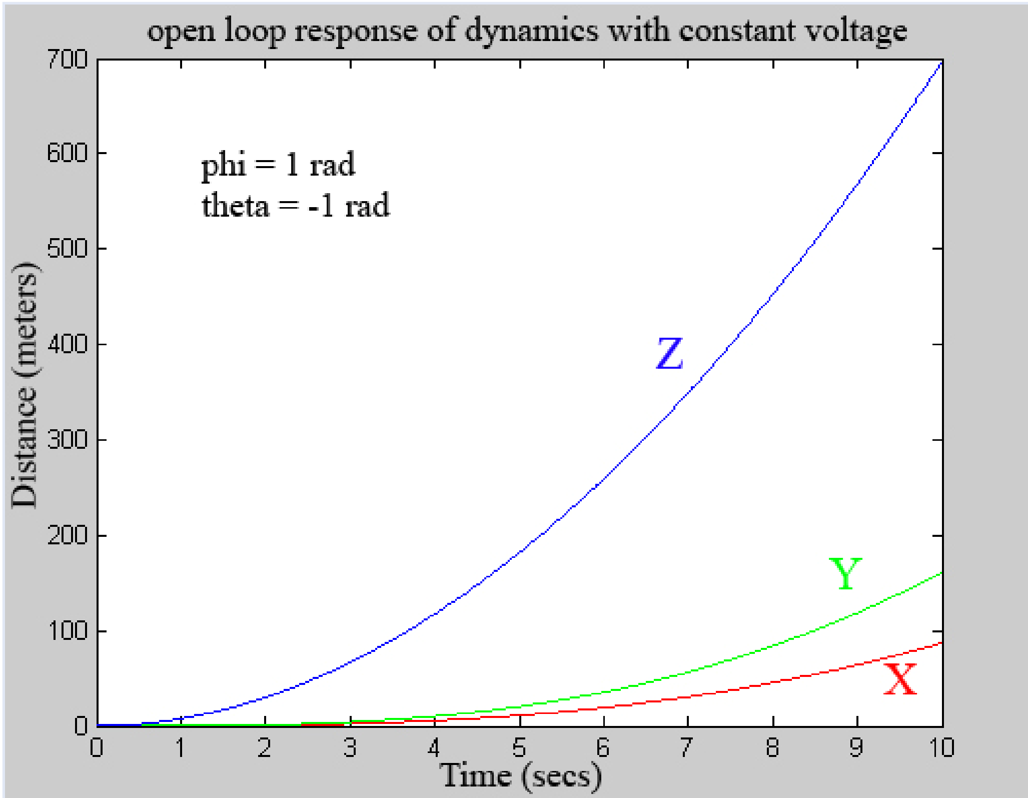


Figure 48: Open loop response

As expected the y and x coordinates change due to their respective axis angles, otherwise they would ideally stay at zero. The values given may be too high, but they do exaggerate the expected results. With further tweaking the dynamic rates and desired positional coordinates can be achieved. PD controllers will also aid in this task.

4.11 Onboard Video Communication

The video feedback system will be a separate entity considered for the project. It will be an out of the box implemented system with minimal time to startup. Because weight seems to be an interest to be kept low, a discrete camera should be used. The wireless hidden pinhole spy camera with microphone by Instapark seems to be the correct product for this project. The camera uses 8V and most of the packaging the camera sits in can be stripped to lessen weight that it produces to the UARC. Furthermore, the microphone will be of no use and doesn't enhance the objective one requires from the communications portion of the project. The consumption this device needs to operate is at 200 mA and is not needed to operate at all times. Therefore, one can trigger this device to turn on by the microcontroller when it is needed. The video can be transmitted wirelessly up to 500 ft in range in any direction. To avoid crashing or loss of the UARC, it will not be flown past 30 ft in any direction. Thus, the wireless range that can be provided by this unit exceeds expectations. The receiver will be located near a laptop with video inputs to see realtime images. All cables and adapters come with the unit and so no further accessories are needed to be able to retrieve the video from the receiver. The receiver operates at 12V with a consumption of 500 mA. This device will be plugged to an AC outlet powered by the available power plug that comes in the box. Because the microphone has been extracted from the transmitter, the receiver's audio port is not needed.

Figure 49: Instapark provides video/audio solutions all in one package. This discrete camera provides weight reduction compared to other models.

It is noted that regardless of the model used, the cameras price value does not have to be expensive to provide video feed. In fact, a simple cheap camera will be able to detect infrared signals coming from the emitter. The camera will show beams of light and thus demonstrate the ability of this concept. An example is shown in figure 50.

Figure 50: A standard remote control that uses infrared signals can be seen by video cameras. Here the illuminating diodes are sending a beam directed at a Tv.

4.12 Power Supply

One of the most important components of the UARC is the battery. The battery will power all the sensors, controllers and motors and has to be rechargeable and dependable. Due to the high current demands of the motors, standard alkaline disposable batteries won't work. There are several types and sizes of batteries available to the RC community. When choosing a battery, there are some important things to keep in mind like type, weight, maximum voltage supplied, total current supplied and cost. The battery will be useless if it's lightweight but won't hold a charge long enough, so the group must find a delicate balance between form, fit and function.

One type of battery is the Nickel Cadmium (NiCad) type. This type of battery has a few advantages like it's cheaper than the other types of batteries, it's not sensitive to over-charging and it can recharge faster than other batteries. Some disadvantages are it has a lower voltage capacity compared to other batteries the same size (about 1.25V per cell), it's heavier, contains dangerous chemicals and it has a memory. This means that if the battery isn't fully discharged after use, and then fully charged, it could get to the point where the battery won't charge at all. In order to keep the battery functioning correctly, this complete draining and recharging must be performed every 30 – 60 days. Also they don't hold a charge very well while in storage. This is a popular battery mostly because of its robustness (they have a high number of charge/discharge cycles) and the fact that they're cheap. Even though they're cheap, they won't be a good fit for the UARC because they're heavy, high maintenance and they're not as efficient as other batteries. The next type of battery that was researched is the Nickel Metal Hydride (NiMH). This battery doesn't have the memory problems like NiCad batteries, contains no hazardous chemicals and has 30 – 40% higher voltage capacity. But they're generally a little more expensive than NiCad batteries, have a reduced life cycle and they're heavy. Nickel Metal Hydride batteries don't hold a charge very well and require maintenance every 60 – 90 days where they must be completely drained and recharged like the Nickel Cadmium type. This is done to prevent large crystalline formations that decrease surface area, preventing a full charge.

Lithium Ion Polymer (Li-Po) batteries require no scheduled maintenance and the energy density is twice that of a standard Nickel Cadmium battery. Li-Po batteries have a high voltage capacity of 3.7V per cell, offering very high energy density with very low weight. There are no memory problems and no hazardous chemicals. They also retain their charges better than the other battery types. There are a few disadvantages with using Li-Po batteries however. They are more fragile than other batteries and require a special charger that won't overcharge the battery. They are more expensive that other batteries as well. They require a protection circuit built into the battery pack that ensures the voltage per cell doesn't exceed 3.7V during the charge cycle, and doesn't drop below a minimum voltage during discharge. Li-Po's also have a aging problem where they lose some charging capacity after a year or so regardless as to whether they are used or not. This is common with other types of batteries as well. Even with these drawbacks, Lithium Ion Polymer seems to be the best option for this application. The group chose to use a 2-cell battery because it's lighter in weight and the price is lower than a 3-cell battery.

Figure 51: A typical 3-cell Lipo battery. Courtesy of Draganflyer.

4.13 Acknowledgements

The group would like to acknowledge those who assisted in the development of this project and research. As having the group made up of strictly electrical engineers and having little involvement in programming made it rather difficult to construct this project. While the challenge existed, the determination to proceed overpowered the challenges that lied ahead. Don Harper, a system administrator of the college of Electrical Engineering at UCF, served not only as a reference point but as a mentor as well. Don has been involved as a lead engineer for the DARPA grand challenge project. He has more than 20 years of experience in different areas of subject such as simulation, embedded computing, wireless networking, robotics and controls. The green Subaru seen driving autonomously around UCF campus was his wife's car he donated to participate in the DARPA challenge. Today, the vehicle drives around campus demonstrating its ability and collecting data to improve its purpose. His name was given as a reference when current students in Senior Design II groups heard of the project that was taken on. After the first visit, he gave step plans in what was needed to be done to complete the project. His advice of starting with the basics resulted in creating the section 4.7 titled "Building a Test Prototype." Many components in the design of the UARC came from Don, as well as the software "Labview." Later it was realized during the interview with him that he posted the 'Lost Helicopter' flyer around school. During a test flight, the Draganflyer that he possessed flew away due to the lack of following pre-flight routines. The group is greatly appreciated for the amount of time he spent with the UARC coming to life.

Figure 52: Don Harper shown in front of his wife's autonomous Subaru seen around UCF campus.

Mr. Mansfield is also acknowledged for his mechanical contraptions that he created for the group. The ideas that were placed on papers and designs that came up turned into reality with his help. Furthermore, when modifications were needed to further the project, he was able to work around the specifications given to him. Along with his son, Clint Mansfield, the prototype served great purpose and helped the group achieve the ultimate goal. In the end, they both created the landing gear that was used during final demo.

The group would like to thank Gary Stein from the UCF Robotics Club, without whom this project would not have been possible. Gary donated his time and expertise, as well as lent the group some test equipment such as the Newton force tester that allowed us to measure the force that our speed controller, props and motor setup can produce. This information was invaluable for helping the group develop calculations such as thrust data, power consumption and weight calculations. Gary also assisted by explaining the problems that he encountered while designing a quadcopter project that is still in progress. He also assisted the group with debugging when minor feedback or stabilization control problems arose. All his help is greatly appreciated.

Chapter 5: Component System Design

5.1 Physical Layout Design

5.1.1 Frame Construction

The design of the layout of the system must take into account several factors. The device must work properly, it must be easy to put together, troubleshoot and repair, and finally it must be relatively small and portable in size. Keeping this in mind, the cost must be kept low to not go over budget. The budget chosen will affect which materials and components are used. Therefore, certain aspects of the project must take precedence in importance over other components. For instance, high grade components should be carefully considered in the application it will perform. A magnetometer with military precision may not be needed as oppose to an accurate accelerometer to know the moment where tilt occurs. One thing that was decided was to purchase a durable frame that would withstand any damage that might occur.

Because the purpose of this project was not material engineering, the frame was considered to be bought pre-made. The Draganflyer frame set was found relatively cheap considered against the MSRP from the company directly. Along with the frame came motors pre-mounted with pinion gears attached. Furthermore, the attached main gearbox for the inrunner design is accompanied with the motor mounts. The frame was taken apart and current wire taken apart in case modification is needed and saves time upfront. The landing gear, battery compartment, rotor blades, and clear canopy came with the set were dismantled as they will not be used.

At a glance it was noticed that the frame purchased was not as sturdy as expected. With the motors mounted, it was noticed the frame was rather wobbly. This instability presented many unknown forces or rotational torques that may have lead to crashing. Therefore, a bracing kit designed for the use of the UARC was found online. Much like Lego blocks, its snaps on the motor mounts and makes the complete frame more rigid and sturdy. With this improvement the bracing kit reduces unwanted oscillations and vibrations to provide efficient and stable flight. An advantage that is gained with this is video streaming being less ‘jumpy’ in normal operation.

5.1.2 Placement of Sensors

Because of the forces encountered on the UARC are greatly affected by weight distribution, the need to properly place the sensors is challenging. The sensors placed on the PCB board were not distributed evenly. This affects, the inertia, upsets torques, and overall flight dynamics. It was nearly impossible to affix the components such that the system is balanced in all aspects. However, what had to be done was to dampen the system in such a way that it is negligible. The battery, for example, hung off the center axis approximately a few inches on the z-plane, this pendulum effect, helped shift the center of mass having better system response. The program corrected itself due to the imbalance of inertia and torques. By implementing this design technique, it allowed for better turning ability as the center of mass has been shifted. Nevertheless, shifting the center of weight by placing the battery on the z-axis too low would overdamp the system and the UARC would struggle to turn. The correct placement of the battery was determined during testing to find a balance in which the system had the best response. Accordingly, some of the sensors had a certain format to be mounted to the PCB board. Because the gyro sensors were bought individually their pin layout should be affixed accordingly. The PCB board had a 3-D design structure as some breakout boards ran across the z-axis to monitor the yaw, pitch, and roll coordinates.

5.2 Programming the Microcontroller

5.2.1 Development Software

The development software for the PIC microcontroller is done either through the core machine language assembly or through a higher level language like basic. An advantage of the PIC microcontroller is that it has minimal execution assembly code. It is also very widely known and has many application notes available for it. In the case of UARC there was a combination of assembly, basic programming, and C. The development software for the Coridium ARMmite is done through basic or C. It is also a very popular chip and is very easy to program and upload. It has a USB and serial port directly on the board.

5.2.2 Basic Code

Basic code was the backbone of the programming for UARC. The code was to implement any of the derived controls systems and act accordingly to reach stabilization. It took in the force inputs and sensor data and sent the signals out through the PWM module to drive the motors as needed. It also needed to utilize the A/D converter. Below is a portion of code created by a group at Columbia University to implement motor control.

outMotorA = (kThrust(inThrust) – (kElevator*(inElevator-512) + kTiltX*(inTiltX-512)) –(kRudder*(inRudder-512) + kGyroZ*(inGyroZ-512)))*

outMotorB = (kThrust(inThrust) + (kAilerons*(inAilerons-512) + kTiltY*(inTiltY-512)) +(kRudder*(inRudder-512) + kGyroZ*(inGyroZ-512)))*

outMotorC = (kThrust(inThrust) – (kAilerons*(inAilerons-512) + kTiltY*(inTiltY-512)) +(kRudder*(inRudder-512) + kGyroZ*(inGyroZ-512)))*

outMotorD = (kThrust(inThrust) + (kElevator*(inElevator-512) – kTiltX*(inTiltX-512)) –(kRudder*(inRudder-512) + kGyroZ*(inGyroZ-512)))*

These equations basically tell the PWM how to react given the control signals deciphered by the microcontroller. There are constants that had to be decided through trial and error and simulation. There was also a need to set a thrust limit implemented on the motors. This was done by testing the motor voltage continuously throughout the program. This can be done by a few nested if- else if conditionals. An example from the same group is shown below.

```
if (inThrust < limit)
{
outMotorB=0;
PDC1L = 0b00000000;
PDC1H = 0b00000000;
}
else if (inThrust > limit && outMotorB < limit)
{
if (limit > 255)
{
PDC1L = limit;
PDC1H = 0b00000001;
}
else if (limit < 255)
{
PDC1L = limit;
PDC1H = 0b00000000;
}
}
else if (inThrust > limit && outMotorB > 1023)
{
PDC1L = 0b11111111;
PDC1H = 0b00000011;
}
else if (inThrust > limit && outMotorB > limit && outMotorB < 256)
{
PDC1L = outMotorB;
PDC1H = 0b00000000;
}
else if (inThrust > limit && outMotorB > 255 && outMotorB < 512)
{
PDC1L = outMotorB - 256;
PDC1H = 0b00000001;
}
else if (inThrust > limit && outMotorB > 511 && outMotorB < 768)
{
PDC1L = outMotorB - 512;
PDC1H = 0b00000010;
}
else if (inThrust > limit && outMotorB > 767 && outMotorB < 1024)
```

```
{
PDC1L = outMotorB - 768;
PDC1H = 0b00000011;
}
```

This code basically checks the separate intervals of induced thrust through the motor. This had to be implemented for each motor separately. Moreover, the flowchart below describes a concept of stabilization.

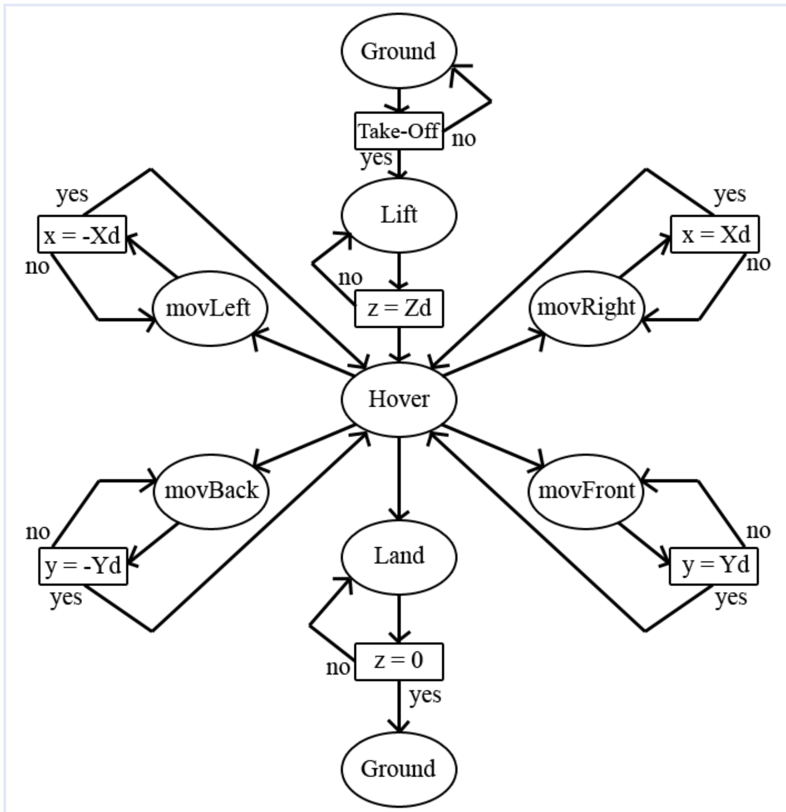


Figure 53: Basic Flight State Flowchart

5.2.3 Sensor Control

Now reading sensor data is the easiest part of the task of control. The toughest part is making sense of it so that the microcontroller can understand. There was a need for a separate microcontroller for sensor analyzation. This microcontroller would implement a Kalman filter to get a nice, efficient, and real time signal from the sensors. The preceeding code is for a Kalman filter in matlab where it shows a possible way to do this.

```
function [k,s] = kfilter(A,C,V1,V2,V12)
%function [k,s] = kfilter(A,C,V1,V2,V12)
%KFILTER can have arguments: (A,C,V1,V2) if there are no cross
% products, V12=0.
%     KFILTER calculates the kalman gain, k, and the stationary
%     covariance matrix, s, using the Kalman filter for:
%
%       x[t+1] = Ax[t] + Bu[t] + w1[t+1]
%              y[t] = Cx[t] + Du[t] + w2[t]
%
%              E [w1(t+1)] [w1(t+1)]' =  [V1   V12;
%                [ w2(t) ] [ w2(t) ]      V12' V2 ]
%
%  where x is the mx1 vector of states, u is the nx1 vector of controls, y is
%  the px1 vector of observables, A is mxm, B is mxn, C is pxm, V1 is mxm,
%  V2 is pxp, V12 is mxp.
%
m=max(size(A));
[rc,cc]=size(C);
if nargin==4; V12=zeros(m,rc); end;
if (rank(V2)==rc);
  A=A-(V12/V2)*C;
  V1=V1-V12*(V2\V12');
  [k,s]=doubleo(A,C,V1,V2);
  k=k+(V12/V2);
else;
  s0=.01*eye(m);
  dd=1;
  it=1;
  maxit=1000;
  while (dd>1e-8 & it<=maxit);
    k0= (A*s0*C'+V12)/(V2+C*s0*C');
    s1= A*s0*A' + V1 -(A*s0*C'+V12)*k0';
    k1= (A*s1*C'+V12)/(V2+C*s1*C');
    dd=max(max(abs(k1-k0)));
    it=it+1;
    s0=s1;
  end;
  k=k1;s=s0;
  if it>=maxit;
    disp('WARNING: Iteration limit of 1000 reached in KFILTER.M');
  end;
end;
```

This code gathered from the internet shows the basic process for Kalman filtering. It takes inputs signals from the sensors and pretty much averages them out to prevent outliers from getting through. Figure 54 below shows the data collected and filtered through the simulink Kalman Filter toolbox using the velocity of some object as the sensor input. Once there is a nice signal generated, it is sent to the main microcontroller to be processed.

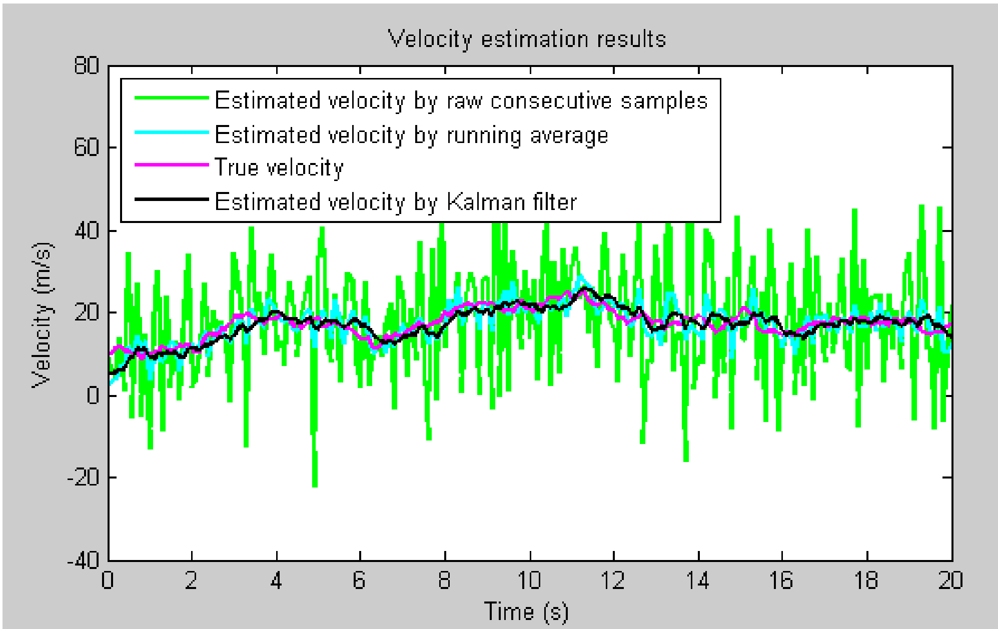


Figure 54: Kalman Filtered signal in simulink toolbox

5.2.4 Motor Ramp Function

When initializing and starting the motors of UARC, much power is drawn. This causes an overload on the battery and it could be damaging. The solution to this is to program the motors to start up in a sequential manner. The best way in programming is to create a MotorInit( ) function. This function will start the first motor then implement a delay of about 1000 milliseconds. Then it will do the same for the corresponding motors. Some example code for this is shown below.

```
Public void Function MotorInit( ){
        outMotorA += 2;      % volts applied to motor A for startup, but not take off
delay (1000);            % 1 second delay
        outMotorB += 2;
delay (1000);
        outMotorC += 2;
delay (1000);
        outMotorD += 2;
        delay (1000);
        }
```

5.2.5 GPS Coordinates

The microcontrollers' job is to log the data signal from the GPS module and store it in variables for use in calculations. The data is given in the form of a signal sentence known commonly as the "GPRMC" statement which is the "Recommended Minimum" sentence. This data line contains all the things a GPS does. Below is an example of a GPRMC statement signal.

$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,200804,,*1A (1)

Each word block is separated by a comma delimiter. The microcontroller will become what will be refered to as an NmeaInterpreter which is the main class in the programming. The first step is to separate each word and examine the first one to determine what kind of information is available. The following code does exactly this.

```
Public Class NmeaInterpreter
 ' Processes information from the GPS receiver
 Public Function Parse(ByVal sentence As String) As Boolean
  ' Divide the sentence into words
  Dim Words() As String = GetWords(sentence)
  ' Look at the first word to decide where to go next
  Select Case Words(0)
   Case "$GPRMC"      ' A "Recommended Minimum" sentence was found!
    ' Indicate that the sentence was recognized
    Return True
   Case Else
    ' Indicate that the sentence was not recognized
    Return False
  End Select
 End Function
 ' Divides a sentence into individual words
 Public Function GetWords(ByVal sentence As String) As String()
  Return sentence.Split(","c)
 End Function
End Class
```

This code first processes the GPS data in the function Parse(). It creates a string variable that calls a function named GetWords(), passing it the sentence received by the GPS. That function returns the data read up to the next sequential comma and stores it. Then a case statement is called to determine whether a sentence was recognized or not. Table 4 sums up the data within the Words variable.

Word variable	Value	Purpose
Word(0)	$GPRMC	Interpret as a recommended minimum message
Word(1)	40302.663	Time in UTC
Word(2)	A	"A" for fix; "V" for no fix
Word(3)	3939.7	Latitude in decimal degrees
Word(4)	N	Latitude hemisphere
Word(5)	10506.6	Longitude in decimal degrees
Word(6)	W	Longitude hemisphere
Word(7)	0.27	Speed
Word(8)	358.86	Bearing
Word(9)	200804	UTC date
Word(10)	*1A	Checksum

Table 4: A word data.

UARC will not make use of all the information received by the GPS module. It would only calculate the position coordinates using latitude and longitude. Other information will be obtained by the onboard sensors.

The next snippet of code appends the previous code, implementing a PositionReceived() event call. The beauty of event handling is that it will update every time the position changes.

```
Public Event PositionReceived(ByVal latitude As String, ByVal longitude As String)

' Interprets a $GPRMC message
Public Function ParseGPRMC(ByVal sentence As String) As Boolean
 ' Divide the sentence into words
 Dim Words() As String = GetWords(sentence)
 ' Do we have enough values to describe our location?
 If Words(3) <> "" And Words(4) <> "" And Words(5) <> "" And _
                Words(6) <> "" Then
  ' Yes. Extract latitude and longitude
  Dim Latitude As String = Words(3).Substring(0, 2) & "°" ' Append hours
  Latitude = Latitude & Words(3).Substring(2) & """"     ' Append minutes
  Latitude = Latitude & Words(4)    ' Append the hemisphere
  Dim Longitude As String = Words(5).Substring(0, 3) & "°" ' Append hours
  Longitude = Longitude & Words(5).Substring(3) & """"     ' Append minutes
  Longitude = Longitude & Words(6)    ' Append the hemisphere
  ' Notify the calling application of the change
  RaiseEvent PositionReceived(Latitude, Longitude)
 End If
 ' Indicate that the sentence was recognized
 Return True
End Function
```

One thing to be careful of is when the GPS sends blank data. This is handled by using the ? in the code to test each word before parsing. Now the next snippet of code is appended to the class and it performs a checksum to determine if the data is in the right format.

```
' Returns True if a sentence's checksum matches the calculated checksum
 Public Function IsValid(ByVal sentence As String) As Boolean
   ' Compare the characters after the asterisk to the calculation
   Return sentence.Substring(sentence.IndexOf("*") + 1) = GetChecksum(sentence)
 End Function
 ' Calculates the checksum for a sentence
 Public Function GetChecksum(ByVal sentence As String) As String
   ' Loop through all chars to get a checksum
   Dim Character As Char
   Dim Checksum As Integer
   For Each Character In sentence
     Select Case Character
       Case "$"c
         ' Ignore the dollar sign
       Case "*"c
         ' Stop processing before the asterisk
         Exit For
       Case Else
         ' Is this the first value for the checksum?
         If Checksum = 0 Then
           ' Yes. Set the checksum to the value
           Checksum = Convert.ToByte(Character)
         Else
           ' No. XOR the checksum with this character's value
           Checksum = Checksum Xor Convert.ToByte(Character)
         End If
     End Select
   Next
   ' Return the checksum formatted as a two-character hexadecimal
   Return Checksum.ToString("X2")
 End Function
```

The checksum basically performs an XOR between the $ and *. It then compares it to the sentence checksum and if they don't match then the sentence is discarded. This helps to make sure that valid Nmea data is received.

The complete class code for position tracking written in C# is shown in Appendix C and labeled NmeaInterpreter( ).

Now to improve accuracy a high precision class is implemented on the interpreter object. This code is also shown in appendix C and labeled HighPrecisionTest( ). This code uses the other word signals given by the GPS module which are $GPGSV and $GPGSA. These signals are retrieved the same as the original $GPRMC command. They aid in the accuracy of information by adjusting for relative satellite position.

5.2.6 Emergency Flight Landing

An emergency module was implemented in the case that UARC breaks one of its restrictions. These restrictions were checked within every few clock cycles of the microcontroller. The main restriction is that of its position. Using information of the x, y coordinates given from the GPS and the current height 'z' from the ultra sonic sensor, the following interrupt pseudo code must be implemented.

```
If x ≥ 30 or y ≥ 30 or z ≥ 15
        Break current trajectory loop
        Create new trajectory
        Keep current x and y position
        Start descent of z position
        Land
```

This pseudo code basically tells UARC to stop everything its current instruction and just descend to the ground and make a safe landing for retrieval.

5.2.7 Speed Controller Programming

The programming for speed control is very straight forward. The ARMmite microchip has built in PWM outputs. Along with the prewritten C library extension files it's easy as calling a function with a couple of arguments. Below is an example of how to send out a PWM wave with the ARMmite controller.

PWM (pin, duty, milliseconds)

This function takes the arguments of which pin number one is accessing, what percent duty cycle that one wants (a value between 0 and 255, where 127 equals 50%), and the period of the wave in milliseconds. Through research and aid from fellow engineers, a period of 2000 ms is ideal. A duty cycle of 100 ms accounted for the hover thrust. When the duty cycle is raised to 200 ms, the thrust will exceed the weight and altitude will increase.

```
If (atHover){
        atLift = false;
        PWM (1, 100, 2000) % induce thrust from motor1
PWM (2, 100, 2000) % induce thrust from motor2
PWM (3, 100, 2000) % induce thrust from motor3
PWM (4, 100, 2000) % induce thrust from motor4
}
If (atLift) {
        atHover = false;
        PWM (1, 200, 2000) % induce thrust from motor1
```

PWM (2, 200, 2000) % induce thrust from motor2
PWM (3, 200, 2000) % induce thrust from motor3
PWM (4, 200, 2000) % induce thrust from motor4
}

There was other functions similar to the previous two, but they send different duty cycles to the corresponding motors for x and y translation.

5.2.8 Transfer to Chipset

The code can be implemented in any language for use on the desired chipset. It acts as a guideline to the flow of data interpretation and execution. Most microchips are based on C, assembly or basic code. Most can be interchangeable through the compiler.

5.2.9 Testing Chipset

Testing of the chipset can be done through the development board. There are many virtual ways to test the signal and an oscilloscope can show the waveform physically. Once programmed and simulated the hardware can be attached for a real time test with load.

5.3 PCB Layout

The group decided to have a PCB board designed for the group project. As having worked 13 years in the field of circuit board designing, Jeremy has worked countless projects involving testing, producing schematics, and price quoting for projects such as military, commercial, and medical applications. His connection to the boardhouse gave the group an edge in having a high quality finished PCB board that differed from the other groups. Jeremy was able to produce a schematic file used to create a net list which is then imported to a layout program. A net list will assign pin layout, circuit connections, layout symbols, and aids in component placement to reduce real estate on board. From the layout program, the components are placed using a CAD program and traces are routed. Jeremy was able to use this software to set design constraints to set minimum spacing, trace widths, and generate art work. The software will check for inconsistency associated by the constraints he used. The generated artwork is used by the boardhouse to create the board requested. The turnaournd time for the finished project took less than a week to receive.

5.4 Power Supply System

5.4.1 Voltages

The voltage supplied by the battery on the UARC supplied 8.5V when fully charged. As power is consumed, the voltage will slowly decrease until it reaches 6.0V. At that time, the battery's chemical makeup ruins the battery permanently. For this reason the UARC will had to have a visual indicator to notify the group that the power is getting low so the UARC won't crash. That circuitry was integrated into the PCB as a visual detection feedback control. A under/over voltage detector has been placed after the battery source and monitors the voltage on the lipo battery. If the voltage is greater than 7.0V, the detector will output a high on pin 1 which outputs to a green led. As the voltage drops below this threshold, the output high then goes to pin 6 wired to a red led at the same time pin 1 goes low.

5.4.2 Wattage

The maximum power supplied by the battery is 7.4V x 1.9A/Hr = 14.06 W. This is assuming the battery is fully charged and delivers maximum current. The battery was rated to deliver 1900mAh of current, which meant that it can deliver 1.9A maximum current for one hour before needing to be recharged. Due to the current demands of the motors, the flight time was roughly around 14.25 minutes in worst case scenario. This depended on the weight of the fully assembled aircraft and motor current demands.

5.4.3 Battery Life

The typical lifespan of a Li-Po battery is typically about 200-300 charges and discharges, or about 1 year with regular use. But that depends on the operating conditions and whether or not the battery was over-discharged or used past the cutoff voltage. As stated earlier in this paper, Li-Po batteries don't tolerate being over-discharged and doing it can result in permanent damage. Over-charging can cause them to swell then explode.

5.5 Navigation

As previously discussed, the UARC can have two different formats for navigating. After meeting with several advisors and discussing the consequences for each, the UARC will fly in a '+' format. This will make the programming math easier to handle. For example, if the vehicle is to roll, the plus format would incorporate 2 motors rather than all four motors to turn. Because the feedback controls takes some time to perfect, we had to improvise and drop the magnetometer from our project. Hence, during prototyping and building, one of the motors will be set as arbitrary north. The UARC will therefore have a hardcoded coordinates to show its complexity that was placed into it. It could theoretically be picked up by a different project and have navigation integrated into it. Modification to the PCB would have to be done, and more RF components would be added to use a remote control.

5.6 Failsafe System

In the case of emergency, UARC needed a way to get itself out in the safest and quickest way possible. Examples of an emergency would be if UARC breaches its coordinate limits or the battery runs low and doesn't produce enough juice to keep the system going. When UARC breaks its range it will stop the current operation and land in an orderly fashion. The main concern was the height range, which was limited to 15 ft since the ultrasonic sensor reads up to approximately 21 ft. When this height is breached UARC will call an interrupt that turns off all sensors and incrementally reduces thrust until the ground state is reached. The following is some pseudo code showing this.

```
FailSafe( ) % fail safe interrupt
{
        gryo1 = false; % set all sensors to off
        gryo2 = false;
        accel1 = false;
        accel2 = false;
        accel3 = false;
        GPS = false;

        Land = true; % set land operational mode on so UARC knows
}
```

Now when the battery ran low and cannot support the system the same fail safe interrupt would apply. By shutting off the sensors and limiting the number of power drawn from the battery, it can focus its power on the areas necessary for landing. This also means that the control system will be less stable.

To enable this feature the LINX receiver/decoder radio frequency chip was used to pull an I/O port high. The software would monitor this I/O port and see when it goes high. A wireless transmitter was purchased to send a signal to the decoder when it should be activated. When the pin goes high and the microcontroller sees this signal, it activated the 'land' program sequence in the code. This code would ramp down the motors and halt the program awaiting the next signal.

5.7 Block Diagrams

The microcontroller had 24 analog input/output pins and 8 analog-2-digital dedicated pins. The first input to be discussed is the Accelerometers. The output of this sensor will be in digital signal as it has been converted by the MAX127. The microcontroller determined if tilt has occurred and compensated the speed controllers if it is not at level. This should theoretically control the motors, and momentarily turn off accelerometer readings to have the UARC move to a location. The gyros will output an analog signal to the microcontroller to be converted to digital signal by the MAX127. Thereafter, it had a direct impact on the trust to each rotor of the UARC. The higher the reference voltage produced by the sensor produced a higher trust voltage to the speed controllers. The coordinates of the flight path will be hard coded. This will essentially control the rotors on the UARC, while feedback from the accelerometers control tilt. The next sensor to be discussed is the speed controller, which is the compensator system for the microcontrollers. These speed controllers will take a PWM input and change the speed of the rotors respectively. The plant will provide the signal based upon the feedback from all the sensors. The speed controllers will then provide a 'switch' to the rotors. The PWM will essentially be an on and off switch rapidly being turned off and on. From this analogy, the signal given by the feedback will speed up or slow down the rotors on the UARC. Lastly, the ultrasonic sensors send out a digital signal to the microcontroller. It will sense for obstructions that may be in the way of the UARC. Hence, it will have a direct impulse on the speed of the rotors.

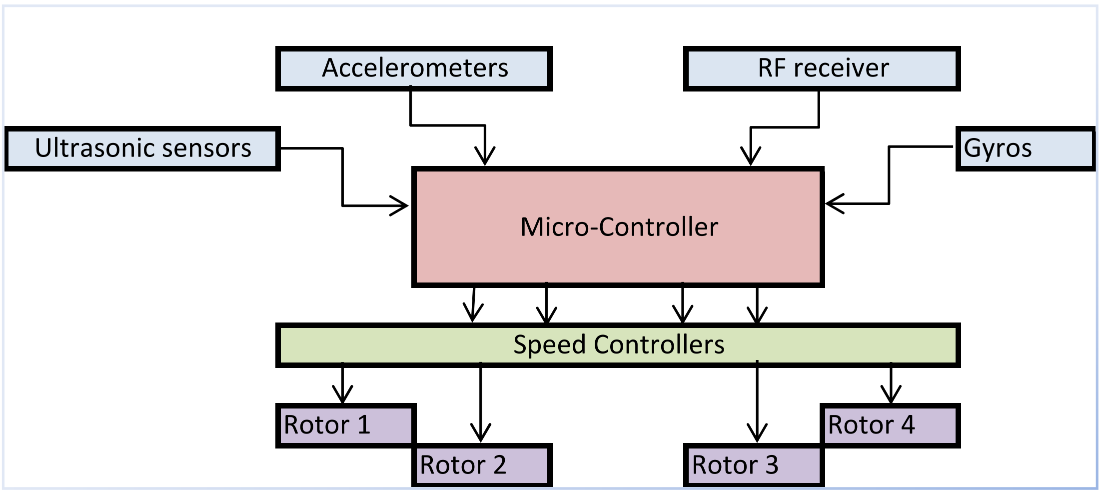


Figure 55: The block diagram for the UARC assembly.

Chapter 6: Prototype

6.1 Leadership Roles

When Dr. Richie announced the format of the paper that was to be done by the end of the semester, the group quickly jumped into research mode. Having to take on such difficult project, the group knew the challenges that lied ahead. Because of this, it was decided to start as early as possible with research, purchasing components, and getting assistance. Within the first week of class, Edwin emailed several companies to see if they could assist the project in any way. On January 29, 2009 the first component was bought; the bare-bones Draganflyer kit. This led to a domino effect of beginning to purchase several components that might assist in understanding certain components. Furthermore, brushless motors, speed controllers, and ultrasonic sensors were purchased.

Working with the groups school and work schedule was a real mission. Clint worked weekends and drove nearly 45 minutes to get to school. Jeremy works from home and drove nearly 45 minutes to attend class. Edwin worked nearby and could be at school at the drop of a hat. At first, it was set to meet twice during the week and would increase as the semester passed by. By the beginning of February, the group had access to senior design lab and testing began on Tuesdays and Thursdays. As the 'Table of Contents' assignment was issued, each developed their own idea of what it should look like and compared each other's paper. Meshing each other's ideas led to the creation of completing this task in less than 2 days. Because all of the members had similar ideas of what it should look like, each one knew the amount of time spent was equal. All of the team members of this group worked previously together and the strengths and weaknesses were known, which made it pleasant to work with rather than an outsider. Working on the 90 page report that was called for was rather easy. Each chapter was worked as a group as sections were split up to complete. As the research furthered each member because familiarized with the concept and grew strengths in the subject. The ideas were shared of what was learned at every meeting that took place. Each member had one goal in mind, 'finish the paper no matter what.' Therefore, no challenge was met on group member resisting to do a certain subject.

Each member put as much effort as the next member. Edwin served as the editor and naturally handed out due dates for certain subjects. With minimal supervising required, each member took responsibility of producing worthy output. Merging documents was relatively easy since the version of Microsoft used converted fonts and formats automatically as they were imported to the editor. Hard effort proved it was well worth it when the first meeting with Dr. Richie came with praise as the group was only 1 of 2 groups that did the assignment correctly without any issues.

6.2 Build Schedule

Originally from the Initial Project and Group Identification Document, the group came up with a Fall Semester of milestones. Figure 56 below describes the breakdown of how the project came about. As previously stated, an early start to the project was taken and understanding how the sensors worked began the first couple of weeks. A group leader was not needed as the group worked closely together to complete the project as a whole.

Clint was the main person in the group to understand simulation and used Matlab to understand the dynamics the group was faced with. Furthermore, simulation was done sporadically throughout the semester as concepts came to mind. Naturally he was the key person to take control of programming as the rest of the group had a weakness with programming. A prototype design was completed by himself along with his father's help.

Jeremy took control of component purchasing and logging. Having prior experience in PCB designing, he developed a small board for the group to use in the design. At will, he was able to find a bare-bones kit online that saved the group some funds and lowered the forecasted budget kept in mind. Because of this, the structural prototyping section was minimized.

Edwin was in charge of presentation and report materials. He took it upon himself to become the editor and assign sections to the group along with deadlines. Furthermore, he was sought for advice on writing topics for sections in this report. Along with Jeremy, much of the device characterization was done in the lab hand in hand. In his off time, he assisted Clint with programming the feedback controls.

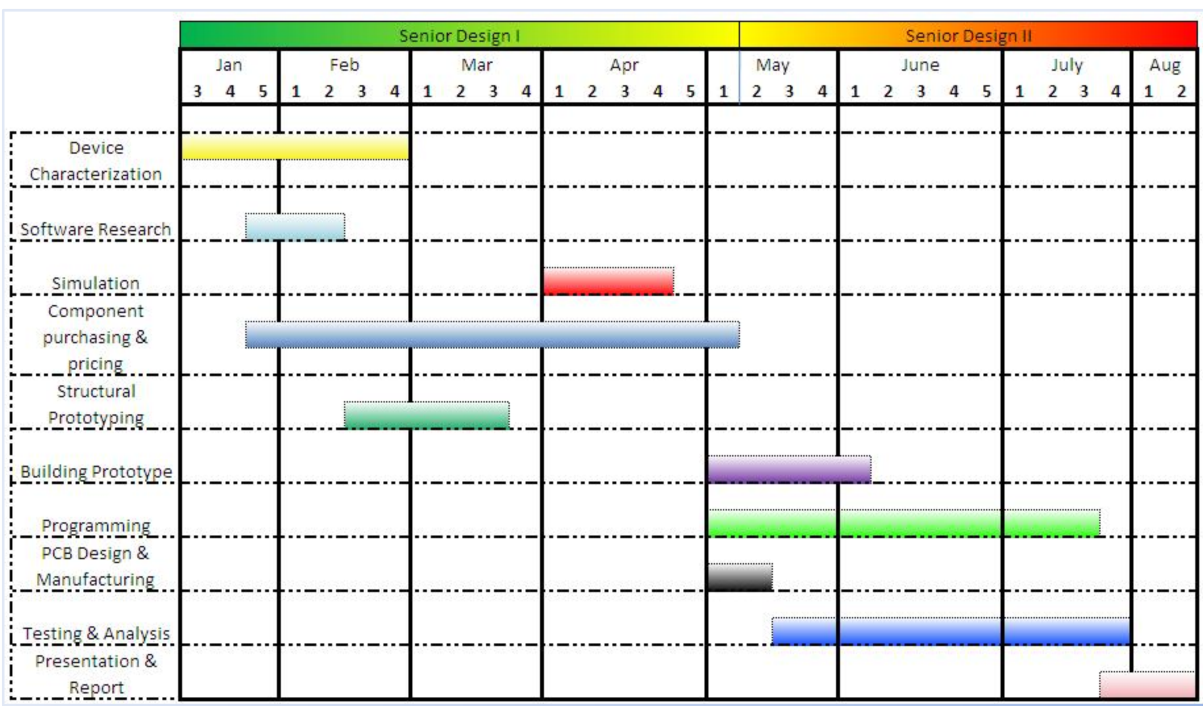


Figure 56: Group Project milestone for the creation of the UARC.

6.3 Budget/Parts list

Originally from the Initial Project and Group Identification Document, the expenses were going to be covered evenly and absorb the expenditures regardless if the product was used or not. This would alleviate disagreements in costs associated with components that were not needed. Of course, the agreement to purchase an item went through an approval process by all members. Table 5 describes the breakdown of the financing by all members.

Group Member	Projected expenses
Edwin Giraldo	$500.00
Clint Mansfield	$500.00
Jeremy Brooks	$500.00

Table 5: Group breakdown of budget forecast

Before the involvement and research of the project, much of the parts list devised were shorthanded of what was needed. Table 6 shows the theoretical parts list for the UARC as of the beginning of the semester. It was realized that the projected budget was almost at its limited with the theoretical amount. Extra parts would come along the way that would burden the budget and so the need to improvise on saving money would have to be taken into mind when building the UARC and choosing components.

	Development	Actual Cost	Cost of materials
MatLab student version	$0.00		$99.00
Brushless Motors (4)		$139.80	$139.80
Landing gear		$175.00	$175.00
Carbon fiber body kit		$300.00	$300.00
Accelerometer		$39.95	$39.95
Ultrasonic sensor (2)		$70.00	$70.00
Microcontroller		$8.80	$8.80
Development board	$25.00		$25.00
Mounted camera		$70.00	$70.00
Carbon fiber blades		$34.95	$34.95
PCB Design	$50.00		$50.00
Gyro (3)		$209.80	$209.80
GPS unit		$114.00	$114.00
Magnetometer		$52.95	$52.95
Rechargeable battery		$70.00	$70.00
USB Transceiver		$100.00	$0.00
Total	$75.00	$1,385.25	$1,459.25

Table 6: Theoretical budget originally created for initial Project and Group Identification document.

Table 7 shows the actual costs the project took to create. As the projected developed, parts were purchased along the way to prevent expenditures that could not be implemented. Because shipping might delay the project, it was well anticipated whether or not the idea could be added to the UARC. The outcome was less out of pocket than expected. Some of the costs could have been avoided and some shipping costs could have been avoided by placing fewer orders from the same merchant.

EEL4914 Senior Design Project - Bill of Materials					
Item #	Description	Part Number	Manufacturer	Qty.	Price
1	Draganflyer airframe	DF-COMPLETE-AIRFRAME	Draganfly Innovations, Inc.	1	$85.90
2	Microcontroller, USB 430 w/target board	EZ430-F2013	Texas Instruments	3	$73.59
3	Brushless motor, 2mm shaft, 6A, 2283 RPM/V	1308441S	Feigao	4	$107.30
4	Ultrasonic range finder, 42kHz, EZ0	LV-MaxSonar-EZ0	Maxbotix, Inc.	2	$61.60
5	Hose clamps	3/4" - 1 1/2"	N/A	2	$1.22
6	Motor, DC (for 2-D test assembly)	N/A	Faulhaber	1	$6.18
7	Draganflyer frame bracing	DF-FRAMEBRACING	Draganfly Innovations, Inc.	1	$22.93
8	Draganflyer pinion for motor	DF-PINION	Draganfly Innovations, Inc.	4	$22.77
9	Brushless motor adapter ring	MPI ACC3900	Himax	4	$18.91
10	Speed controller, 9A	CC-TB9	Castle Creations	2	$54.34
11	Servo tester	N/A	E Sky	2	$23.12
12	Potentiometer	10K ohm	Radio Shack	1	$3.00
13	PVC / wood for 2 dimensional test assy	N/A	Lowes	1	$3.32
14	Electrical connectors for Speed Controllers	N/A	Bob's Hobby Center	4	$12.74
15	Gyro breakout board	ADXRS614	Sparkfun / Analog Devices	2	$129.90
16	Accelerometer, triple axis	ADXL330	Sparkfun / Analog Devices	1	$34.95
17	Microcontroller	ARMmite	Sparkfun / Coridium	1	$59.08
18	Print Final Paper for Senior Design	Kinko's	N/A	1	$78.20
19	Newark passive components order	Assorted	Assorted	1	$20.49
20	Newark passive components order	Assorted	Assorted	1	$5.50
21	Mouser passive components order	Assorted	Assorted	1	$16.26
22	Mouser passive components order	Assorted	Assorted	1	$50.31
23	Li-po Battery	7.4V, 1900maH	Thunder Power	1	$26.45
24	Digikey passive components order	Assorted	Assorted	1	$4.72
25	Mouser passive components order	Assorted	Assorted	1	$16.48
26	Ball and socket	Ball caster w/ 1" plastic ball	Pololu Corporation	1	$11.94
27	Mouser passive components order	Assorted	Assorted	1	$8.02
28	Advanced Circuits - PCB	UARC PCB	Advanced Circuits	1	$62.99
29	Brushless motor, 2mm shaft, 6A, 2283 RPM/V	1308441S	Feigao	1	$32.45
30	Speed controller, 9A	CC-TB9	Castle Creations	1	$30.87
31	Speed controller, 9A	CC-TB9	Castle Creations	1	$31.90
32	Li-po Battery	7.4V, 1900maH	Thunder Power	1	$47.91
			Total:		$1,165.34

Table 7: Actual Expenditures.

6.4 2-D Test Assembly

As previously mentioned in section 4.7.3 'Building the Contraption', the 2-D test assembly was used to understand the dynamics. The majority of the project revolved on this assembly while the code and sensors are synchronized to get the correct response. For this reason the prototype of the test assembly can be seen on figure 41 of page 54 in this document. As the sensors was tested and enabled to respond, the next sensor was being implemented on the assembly structure. After all the sensors were understood and tested, it was time to transition into the 3D model.

6.5 Transition to the UARC

After the 2-D assembly had been completed, it was time to mount another axis to the contraption. To achieve this method a ball bearing joint was bought such that the axis is free to rotate in pitch, roll, and yaw. From the programming aspect, the same code used for the 2-axis is copied for the extra axis mounted. This made the implementation easier to decipher as the extra axis is treated independently. The code was revisited to adjust the PID coefficients to adjust for the added thrust attached on. Furthermore, the completed frame included added weight from the extra components that would complete the feedback system. At first the UARC encountered yaw gyration effects due to the motor dynamics characteristics being different. Because the purpose of the project was not to build the 'perfect' vehicle, this was neglected in the overall system. Furthermore, this effect did not lead to crashing of the vehicle. Once the programming and the test assembly was completed, it was time to transition to the Draganflyer bare-bones kit purchased. From the previous sections, one will notice that the test assembly parts are the Draganflyer pieces and will just be mounted to the remaining parts set aside from the bare-bones kit. This approach effectively reduced the transition from one apparatus to the other. The PCB board was mounted in the special cubby hole above the center of mass axis. All parts, wires, connections were checked for safety. Because one knows the feedback PID controls are working up to this point, the coefficients had to be revisited one last time as the complete test assembly will be detached into a lighter application. After smoothly working out the math in the programming, the UARC took form to fly autonomously.

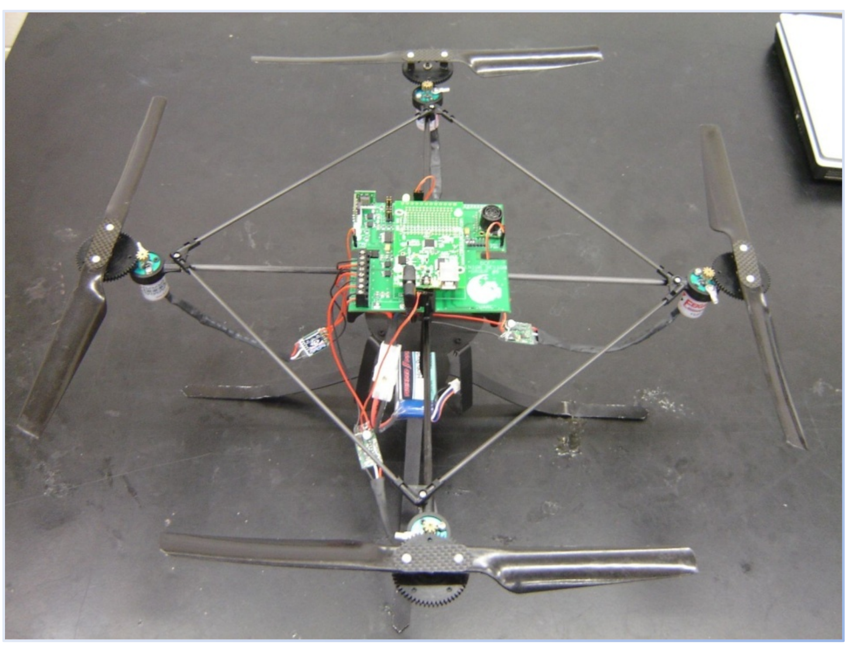


Figure 57: The transition to the 3D vehicle.

6.6 Final Touches

The group wanted to implement visual status to assist in monitoring the UARC. A power 'on' indicator light was incorporated on the PCB board to let users know when power is okay from the power supply to the PCB board. As mentioned before, a green color led would indicate a normal operation until the led turned red in which the battery should be recharged. The colors chosen could be seen from at least 15 ft or so. Because the speed controllers require a voltage of 3V to operate, once the Lipo batteries have reached this voltage the UARC will not operate. This practical implementation would allow a visual confirmation of how much flight time is left. For added stability, the brace kit purchased for the project was affixed to UARC. This dampened the system for a gain in flight.

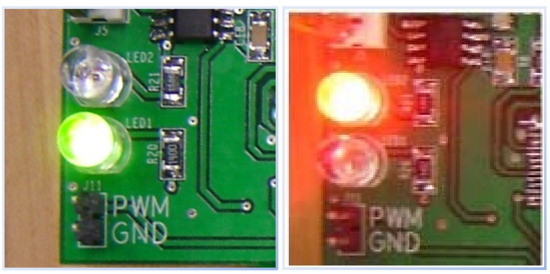
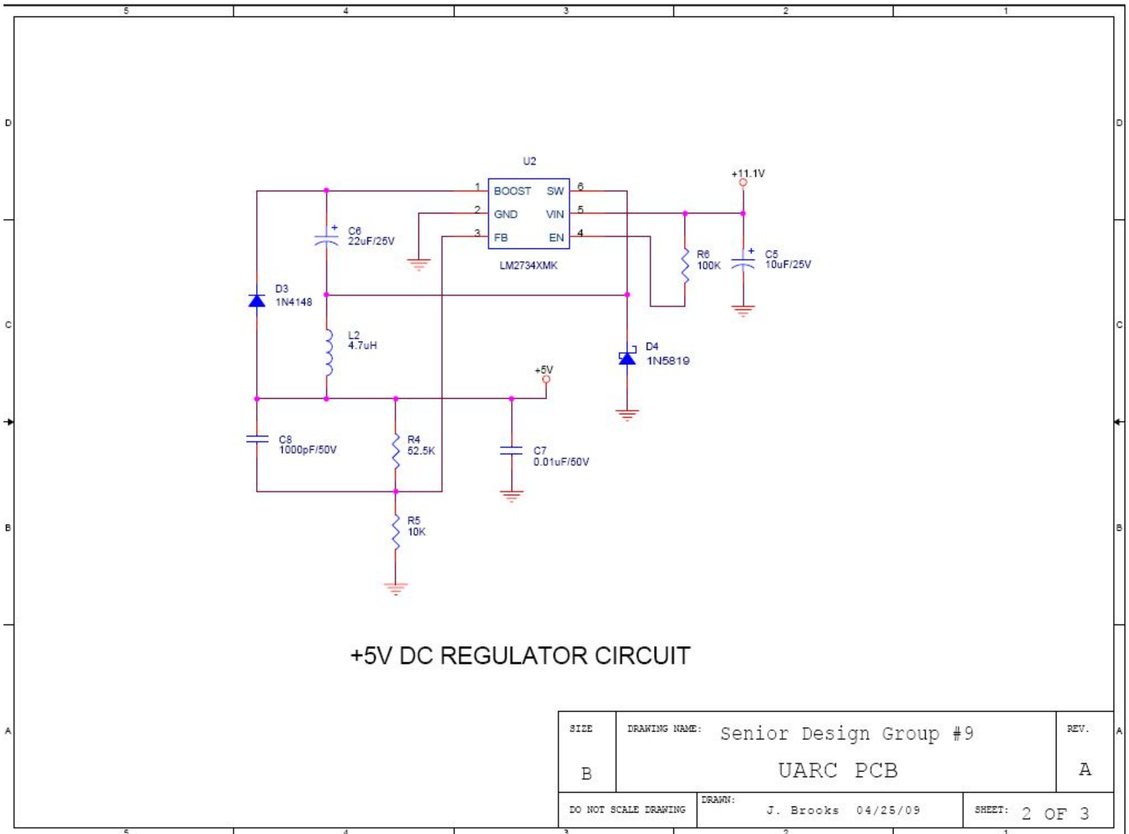
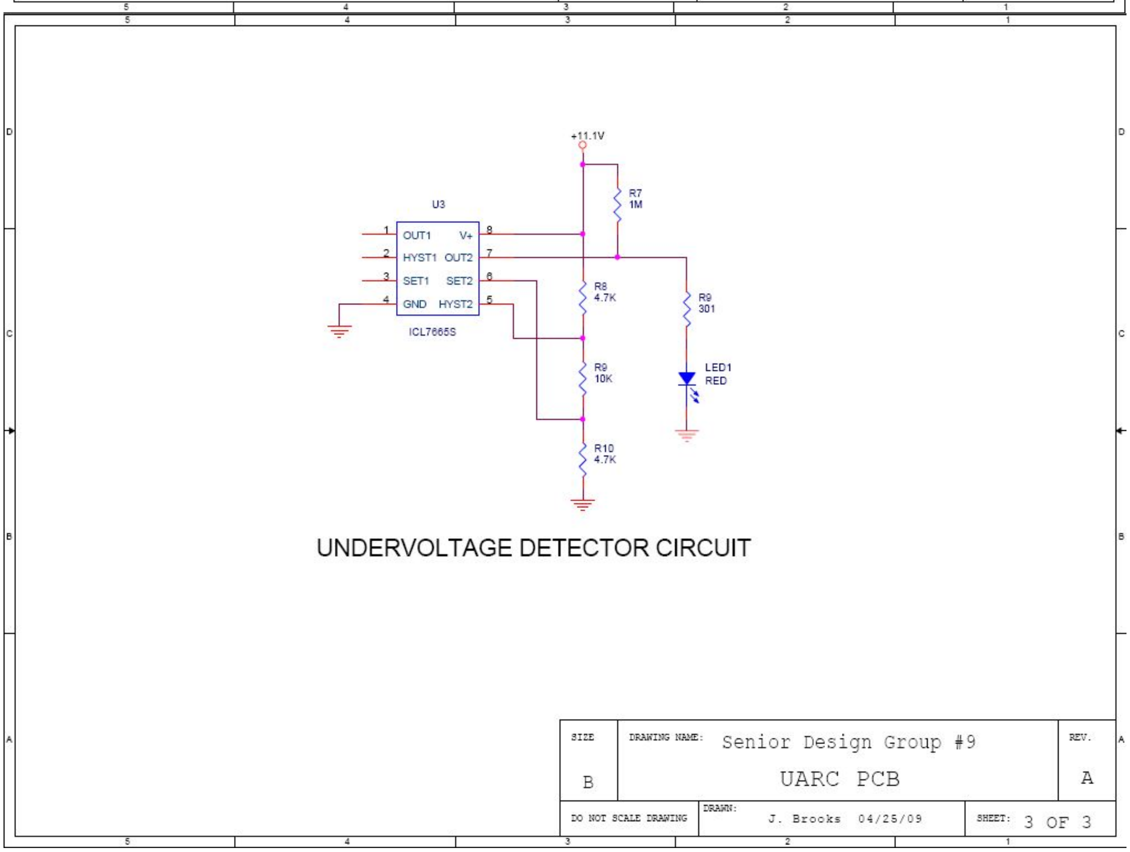


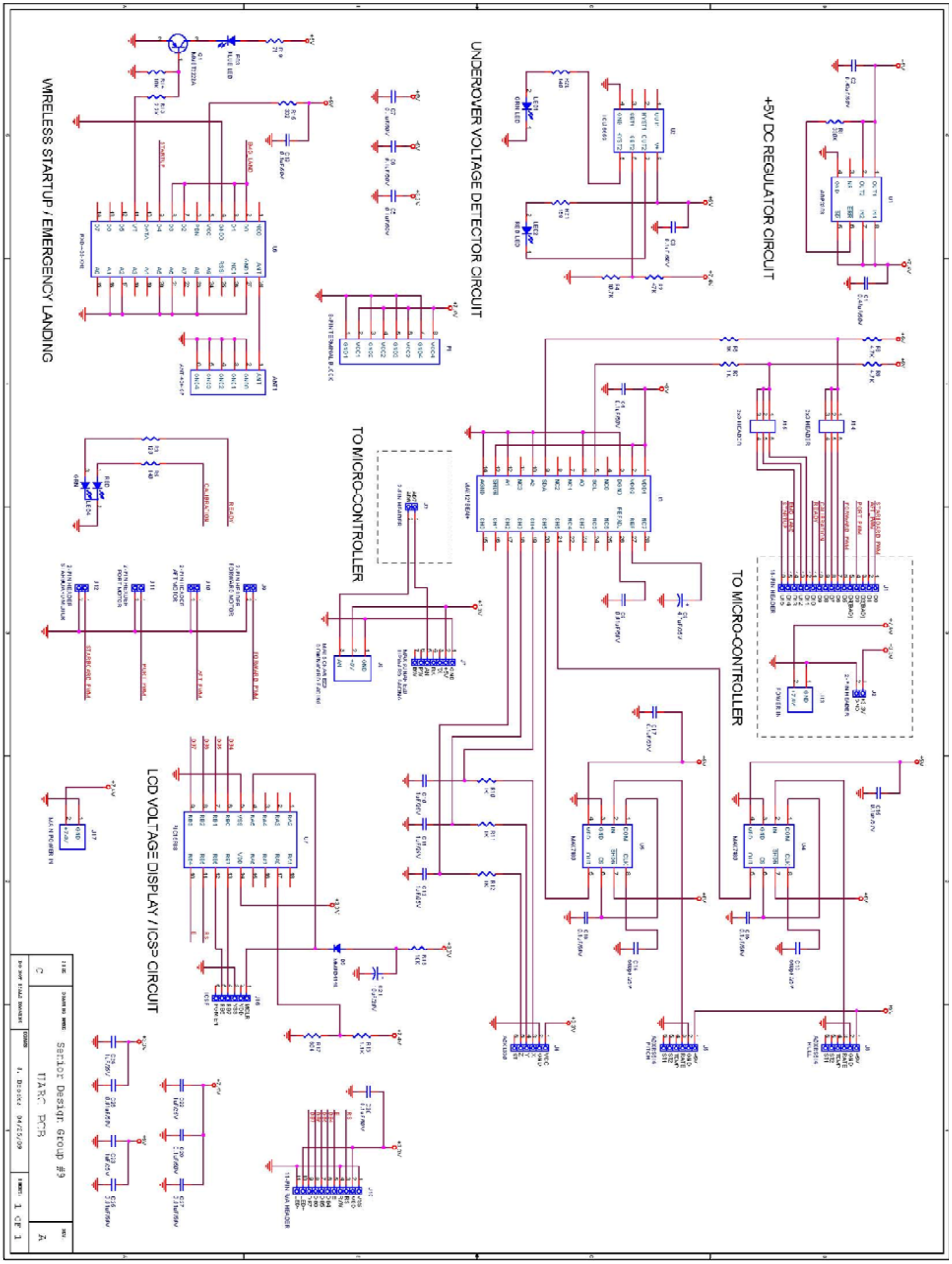
Figure 58: The Green indicator tells the user the operation is normal. Once the red led is turned on, it is time to recharge the lipo battery to avoid damage.

6.7 Schematics

The following schematics handle the power distribution of the UARC as described earlier in this paper. Also there's a circuit utilizing the Intersil P/N ICL7665S, which is a CMOS over/under voltage detector. It will sense and monitor the battery voltage output continuously until the output drops below 7.0V. At that time, the OUT2 pin will go high, turning on a red LED. This served a visual indication that the battery voltage is getting low and the UARC can be powered off for a battery re-charge. The speed controllers have a cut-off voltage of 6V (or 3V per cell) and they will shut off when the voltage falls to that level. Obviously this could be bad if the UARC is in mid-flight during this time. The group explored other means of monitoring the battery voltage including a LCD display that outputs the battery percentage remaining. Because the added weight could restrict stabilization during flight, this idea was quickly dropped from the project. In the end, the most useful method was using the over/under voltage led.


U2
BOOST
SW
GND
VIN
FB
EN
LM2734XMK
+11.1V
C6
22uF/25V
R6
100K
C5
10uF/25V
D3
1N4148
L2
4.7uH
D4
1N5819
+5V
C8
1000pF/50V
R4
52.5K
C7
0.01uF/50V
R5
10K
+5V DC REGULATOR CIRCUIT
SIZE
B
DRAWING NAME: Senior Design Group #9
UARC PCB
REV.
A
DO NOT SCALE DRAWING
DRAWN: J. Brooks 04/25/09
SHEET: 2 OF 3


+11.1V
R7
1M
U3
OUT1
V+
HYST1
OUT2
SET1
SET2
GND
HYST2
ICL7665S
R8
4.7K
R9
301
R9
10K
LED1
RED
R10
4.7K
UNDERVOLTAGE DETECTOR CIRCUIT
SIZE
B
DRAWING NAME: Senior Design Group #9
UARC PCB
REV.
A
DO NOT SCALE DRAWING
DRAWN: J. Brooks 04/25/09
SHEET: 3 OF 3


+5V DC REGULATOR CIRCUIT
UNDEROVER VOLTAGE DETECTOR CIRCUIT
TO MICRO-CONTROLLER
TO MICRO-CONTROLLER
WIRELESS STARTUP / EMERGENCY LANDING
LCD VOLTAGE DISPLAY / ICSP CIRCUIT
Senior Design Group #9
UARC PCB

6.8 Power Distribution

The Li-Po battery outputs a voltage of 7.4V, but the sensors and other circuitry needed lower voltages to operate. In order to account for this, a step down voltage regulator was placed on the PCB that outputs 5V constant. This device was chosen because it's more efficient than using a voltage divider, it's small, light weight, takes an input voltage between 3 and 12V, has an output voltage between 5V and an output current of 200mA. For this application, the UARC needed 7.4V, 5V and 3V to operate. A 3.3V rail was found on the development board of the microcontroller which was used. The 4 speed controllers will use the 7.4V supply directly from the source. The 2 gyros, 2 ultrasonic sensors and microcontroller will use the +5V supply. And finally, the accelerometer will use the +3V supply. When choosing this regulator, Jeremy had to account not only for the voltage required but also the current. The currents from all the devices were added up to ensure the output current of the device wouldn't be exceeded. The section titled 'schematics' shows all the circuitry, including protection circuitry for the UARC's power distribution.

Chapter 7: Testing

7.1 PCB Testing

After the PCB board was received from the boardhouse that was requested, it was time to test out what was delivered. The first initial steps taken was to make sure the board is working and check masks are not covering surface mount pads and vias are set in the right place. It was assured the measurement of the board that was ordered came back as specified. This is due to the placement of the board on the final product; any offset would result in design changes that were not foreseen. Pads were checked to make sure passive component would rest on the board with no offset. The connectors to the microcontroller had to be soldered with precision due to the fact that any offset would not allow communication to the sensors. Because the tools necessary for this test were not within reach, it was best to test them manually by inserting them individually into the socket or the board directly while soldering took place. As offsets were found, the pad had to be heated and adjusted to make sure the pin had perfect placement. Best practices were made to assure ESD was avoided from damaging the chips. Finally, the test points will be measured to see if connections have been made by the boardhouse. This will also facilitate when a component is not giving proper readings to determine where the error lies. Should a component be damaged or not making proper contact, one will be able to isolate the component using these test points.

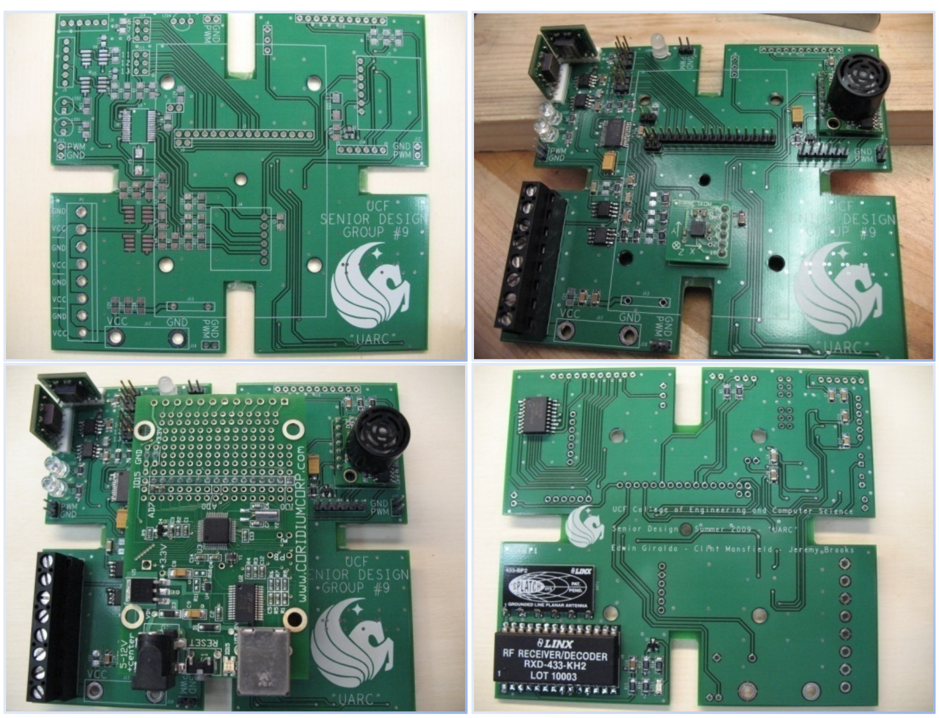


Figure 59: Pictures of the PCB created for the UARC. The fabricated board, mounted sensors, mounted development board with microcontroller, and bottom side of board is shown.

7.2 Preflight Test

To prevent catastrophic events from careless mistakes, preflight checkpoints have been developed. While this seems rather cumbersome, the thought of starting all over due to lack of repetitive steps does not seem worth it. The list below describes routines taken before any flight testing.

- Check battery charge with multi-meter
- Tighten nylon thumbscrew that hold rotors in place.
- Check main gear and pinion gear for wearing
- Tighten screws on frame as necessary.
- Check soldered wires on brushed motors.
- Check connections from sensors onto PCB board.
- Secure battery onto UARC.
- Check speed controller quick-tie connections

No extra accessories were kept on hand. If a failed component occurred, testing would cease until it was replaced. Thus, proper initial walkthrough to make sure everything is in check will prevent component damage. Although not a huge concern, weather and temperature are checked to make sure flight was done in the right conditions.

7.3 Latched Test

The fear of crashing and damaging components was one thing that was to be prevented to further the concept of the project. For this reason, a latching system was used to have a controlled experiment without disaster. With the use of nylon strings, the UARC was tethered to a platform to restrict it from flying past 1 ft. from the platform. Because the initial testing had unknown results, this would assure the safety of the project survival. One group member would also monitor the apparatus to make sure it did not slam down on the landing section from initial takeoff. Closely monitoring any jitter that might arise from the vehicle took great eye-hand coordination. As the tests further, the UARC had more freedom to take flight until the comfort to fly solo reached an agreement by the group. If any change was implemented to the software coded on the microcontroller, the latched test had to start from the beginning until testing proved successful and the tethering can be lengthened. As testing took place, the UARC was held in place as the motors ramped up to full capacity to prevent crashing. This was done because brushless motors have a tendency to spin backwards before knowing where the actuators are inside the cylinder during startup. At times the microcontrollers output produced unwanted signals to the brushless speed controller that had unknown effects. There were multiple times when the program had to be restarted due to this unknown effect and only took seconds to restart. Even at the end of the project this error was not deciphered.

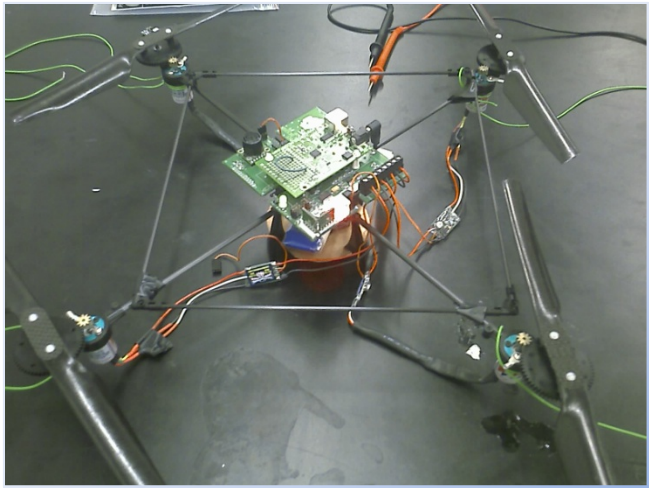


Figure 60: The UARC prepares for flight by tethering down the vehicle to avoid a crash.

7.4 Battery Stress Test

There was a need to determine the amount of flight time that one would have with each battery pack. Theoretical results could have been determined, but this does not simulate real-life experimentation. For this reason, it was voted to place the battery pack through a stress test to determine output. In some scenarios, the UARC was allowed to ramp all the way and drain the battery to determine a time of flight. Lipo batteries are able to handle the amount of current draw and voltage necessary for the project, up to 30A drawn in any point in time. The device, which was latched, would be documented to see how long it lasted during flight at this level. From the results one can derive the worst-case-scenario of how much flight time can be attempted. During this test flapping of the blades was noticed, this is due to the fact that the nylon blades have rpm restrictions. The second test was an optimum case scenario testing. The microcontroller was programmed to ramp up the motors to lift up the UARC and hover about an inch off the ground. Again, it was tethered and allowed to drain the battery completely and documented the performance of that test. The next test was a half-throttle test to test the battery at 50% duty cycle. The test was compared to the second test to determine the correlation of timing between those tests. From this one could determine how much performance is being lost to the weight of the UARC. Obviously operating at 50% duty cycle to hover is not optimal, but with this simple test one can see how efficient the UARC is.

Finally, testing was done to input a step function to determine performance. This was done by programming the microcontroller to operate at basic hover for 10 seconds and then ramp up to max performance for another 10 seconds. This mimicked the UARCs normal operation during flight. The group tried to set boundaries on how this test should be programmed and it was concluded this was the best function to set this condition. Table 8 gives a breakdown of the different test that was done. It should be noted that results will vary due to battery life and charge cycles. Because flight patterns will affect battery lifetime before charging, all results are not to be assumed for accuracy.

Battery Stress test	
Test	**Time**
Max draw	4min.
Optimum case	10min.
50% Duty cycle	7min.
Step function	15min.

Table 8: Four tests were completed to mimic different situations that the battery may encounter.

7.5 Basic Hovering

After latched testing and battery stress testing was completed, it was time to let the UARC have hovering to test if the microcontroller and feedback system are working accurately. The UARC was turned on after the initial preflight test was checked, and the group stood back and witnessed the final product that was created. The UARC had spontaneous reaction as it lifted off the ground, for the fear of crashing and having a non-working demonstration for the final presentation, we did not allow a full unmanned flight. A video recording was taken of the session to document and analyze on how to adjust the PID controls that perfected the results. A substantial amount of time was taken to assure the safety of the vehicle before allowing it to hover during each test, only momentarily was it allowed to function on its own.

7.6 Sensor Testing

7.6.1 Accelerometer

The accelerometer was tested in the lab to learn its response to tilt stimuli. This served a great purpose in understanding what type of output it was sending to the microcontroller. By simply testing its device characterization in the lab, one could predict the results of what the microcontroller will be receiving as the input and program code accordingly. This also proved worthy in flight testing when unexpected errors occurred, by pinpointing which sensor did not give accurate results. Furthermore, the accelerometer response was tested during flight by slightly tilting the UARC towards one of the cardinal coordinates and seeing if the system will tried to correct itself. Many times were different capacitors tried to filter out the noise the best. During tethered flights, the UARC seemed jumpy and was probably affected by the readings the accelerometer gave. Because the PCB was done, and much cannot be done to change the soldering, different PID values were tried to get the best flight response. Figure 61 shows lab results acquired.

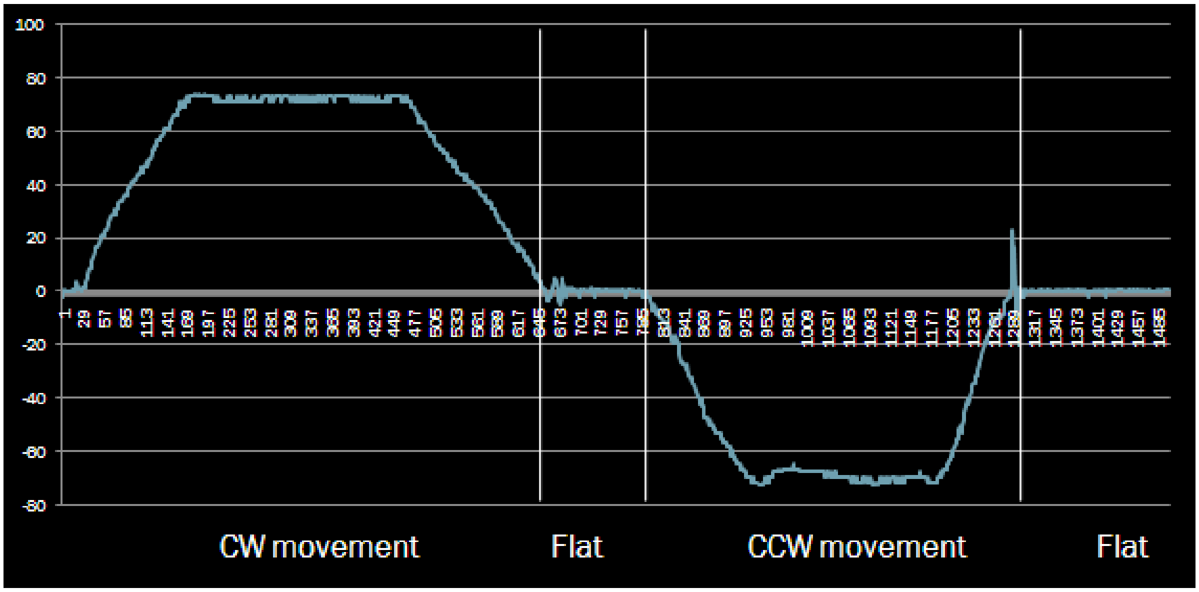


Figure 61: During testing, the Accelerometer was studied and had these characteristics.

7.6.2 Gyro

The next sensor to be tested was the rotational speed sensor. Again, in the senior design lab, the sensor was hooked up to the oscilloscope and basic tinkering around was performed to see what would be displayed. It appears the sensor displays 2.5 volts level at rest when no stimuli are given to it. Asvthe gyro was tilted in one direction or the other, a +/- voltage change from reference occurred. It was noticed that faster changes of tilts resulted in higher readings. After reading the spec sheets to comprehend what type of output would be seen, it describes the output as a voltage change proportional to the rate of tilt angle. With a description, a hands on approach of watching this occur put the pieces together of how it worked. The output would yield an analog signal that will be needed to convert into digital logic for the microprocessor to process. Theoretically, any voltage change produced by the sensor would have a direct trigger for the rotors on the UARC. A faster tilting would have to be compensated quickly by the rotors to establish balance or crash might occur. Tinkering with this sensor in the lab proved successful as it was visually proved the exact functionality of what it produces to the microcontrollers input. To test the functionality of this sensor during flight, quick jerks were applied to the UARC. The output affected the blades immediately to compensate for the velocity that was seen. Because the UARC was not predicted to be falling during flight, the response that this sensor gave was secondary with respect to the accelerometer. However, to have flight both sensors proved to be necessary, but the PID control did not have to be as cumbersome. It was important during programming to take into account the orientation the gyros were soldered onto the board and which coordinate was defined as north. Figure 62 shows lab results that were achieved during a test flight that was performed which shows clockwise and counterclockwise movements.

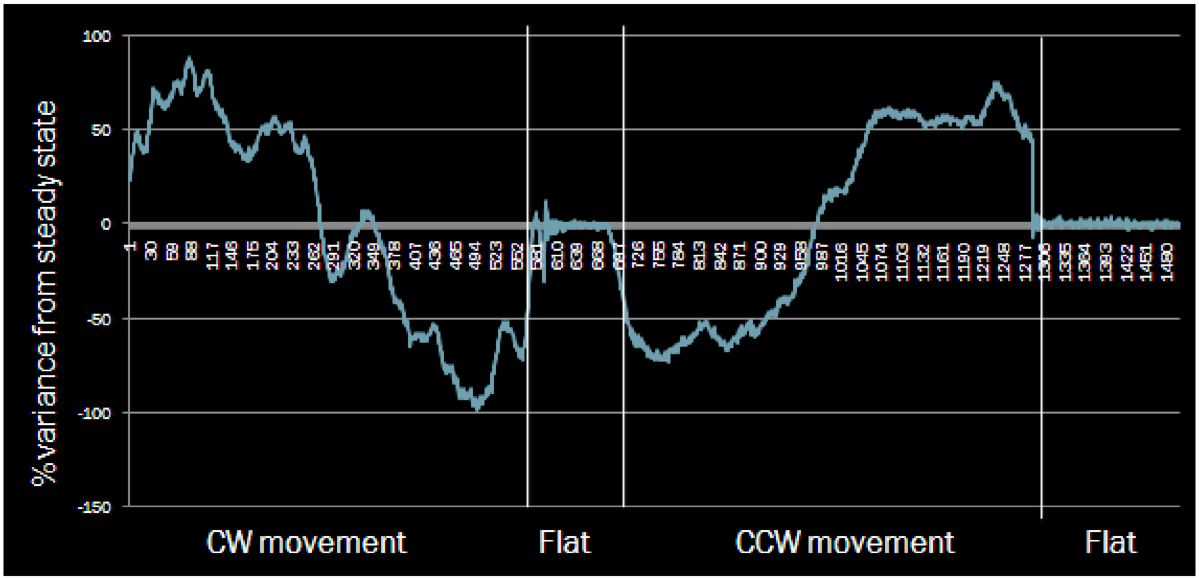


Figure 62: During testing, the gyros were studied and had these characteristics.

7.6.3 Ultrasonic

Being the first sensor bought, lots of tinkering had been performed on this sensor since its inception. Seeing the device output on the oscilloscope at first was a challenge. Thorough spec sheet research was done to assist in understanding what was to be seen on the screen. Different outputs found on the breakout board had to be ciphered to see which one is suppose to be used. After proper connections were made, the sensor was put to the test by placing objects in front of it and seeing the waveform produced on the oscilloscope. Thereafter, a copper wire was waved in front of the sensor to see the sensitivity that could be achieved. Surprisingly, it was able to detect this motion when waved in front of it. With this sensor, obstacle avoidance was achieved. Although it should be flown outdoors, certain objects might appear in the flight of path such as twigs, tree limbs, etc. Even the thinnest twig could prove fatal if the ultrasonic sensor is not sensitive. During testing, an object was placed in the way of the UARC roughly 6 inches in distance and figure 63 shows the output in the oscilloscope as changing voltage.

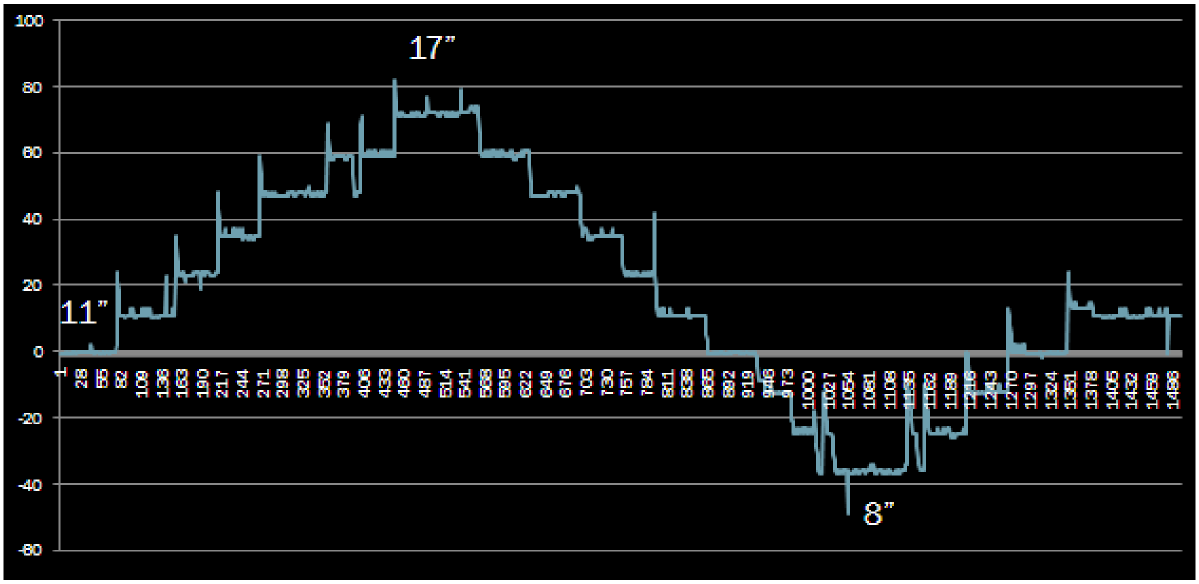


Figure 63: During testing, the ultrasonic sensor was studied and had these characteristics.

7.7 Unmanned Flight

After basic hovering has been accomplished and all sensor testing has passed, it is time to release the UARC in unmanned flight. Having trust in the electronics and feedback systems leading up to this point should satisfy the basic requirements established for this project. Again, the expectation is to have this vehicle in air for 10 minutes and demonstrate its ability to sustain flight on its own while avoiding crash. All the basic testing done on the sensors as explain in section 7.6 will be demonstrated as well. The purpose here is to demonstrate the UARCs ability to hover in place. After the group is satisfied with this challenging part, it is time to demonstrate the wireless communication as explained in the next section.

7.8 Wireless Communication

It was time to test the failsafe previously discussed. The handheld wireless transmitter has to be opened and programmed to have a unique code when transmitting wirelessly. At the same time the LINX receiver/decoder had to be soldered to high and low pins the same way. Using an exacto knife, the wireless transmitter lines were cut indicating high and low values. The receiver has a pin that goes high once a signal is received by the wireless transmitter. This pin is connected to the microcontroller and the program listens in on this port and determines when it goes high. During testing, accurate signal was being received but the pin was not going high as expected. The initial reaction was that the chip was not functioning as it was suppose to. A call was placed to the manufacturer, but they do not support the product unless the development kit was purchased, which was not. Further testing revealed that the same pin used on the microcontroller powers a led on the development board. Because of this, the microcontroller was not seeing a high do to the voltage drop on the led. It was quickly improvised that the microcontroller be jumpered to an unused port. This seemed to fix the issue at hand and was operating as the technical documents inform. The same transmitter was later programmed to be used as the start sequence of the program as well. During testing, the UARC sat on a flat surface and the wireless 'kill switch' was pushed to start the sequence of the program. It would then run the calibration test and turn on. When the group needed to end the sequence of the program, the kill switch was pushed again and the UARCs motors ramped down.

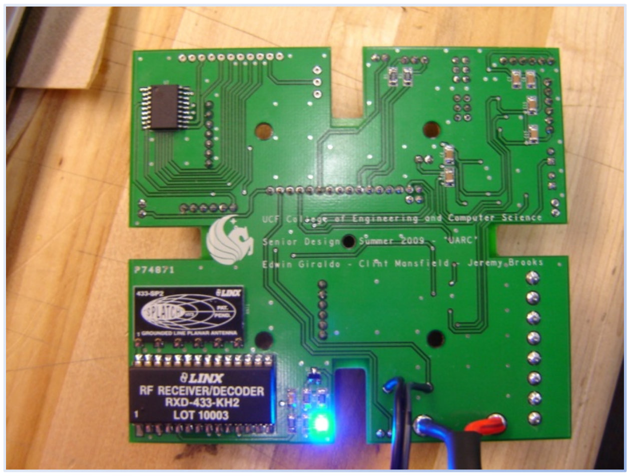


Figure 64: The UARC provides visual feedback when the kill start device is used for safe landing.

Chapter 8: Final Summary

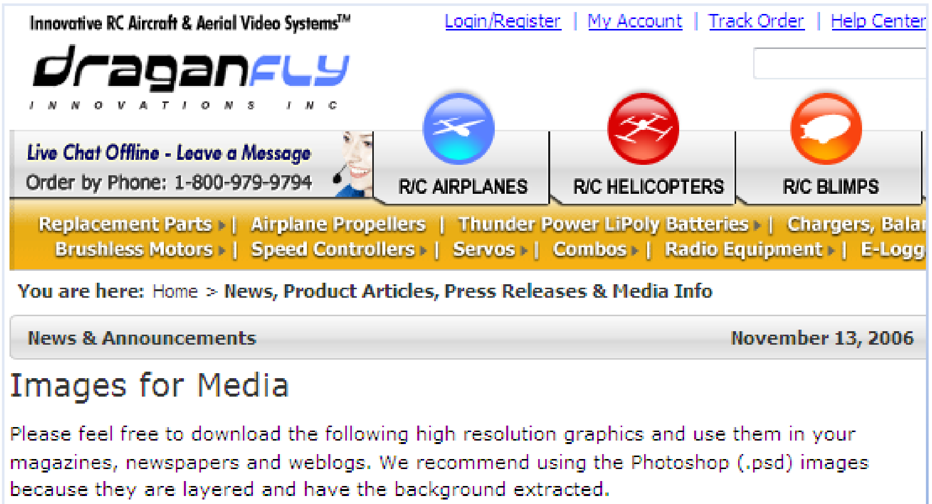
Over the past 7 months an idea turned into reality. Along the way the group faced many challenges and obstacles that could hinder the final goal. Bridging the gap between these two is what true engineering is all about. The project was started as soon as approval was given from the professor. More than half of the final parts were purchased within the first few weeks of Senior Design 1. Senior Design 2 consisted of pure testing and working to resolve software issues.

The biggest step towards the completion of this project was the constructing the 2D design using analog devices to observe and study the mechanics of stabilization, and then transferring this data to digital components. Taking on such a difficult project, it was known from the beginning that many challenges lay ahead. Many, many hours were spent in the lab on a daily basis. As difficulties arose and issues were resolved, this motivated the group more and more to move forward.

The group would like to thank Gary Stein of the Robotics Club for his assistance with this project. He was available anytime the group needed him. The group would also like to thank Don Harper. His contributions were also extremely helpful in getting this project off the ground (no pun intended). Also very helpful were all the companies that contributed free samples to our project such as Maxim-IC, Samtech, Analog Devices, Intersil, Mill-Max, Texas Instruments and Microchip Technology.

Appendix A: Emails

Chapter 2: Definitions


Innovative RC Aircraft & Aerial Video Systems™ | Login/Register | My Account | Track Order | Help Center

draganFLY INNOVATIONS INC

Live Chat Offline - Leave a Message

Order by Phone: 1-800-979-9794

R/C AIRPLANES | R/C HELICOPTERS | R/C BLIMPS

Replacement Parts | Airplane Propellers | Thunder Power LiPoly Batteries | Chargers, Bala
Brushless Motors | Speed Controllers | Servos | Combos | Radio Equipment | E-Logg

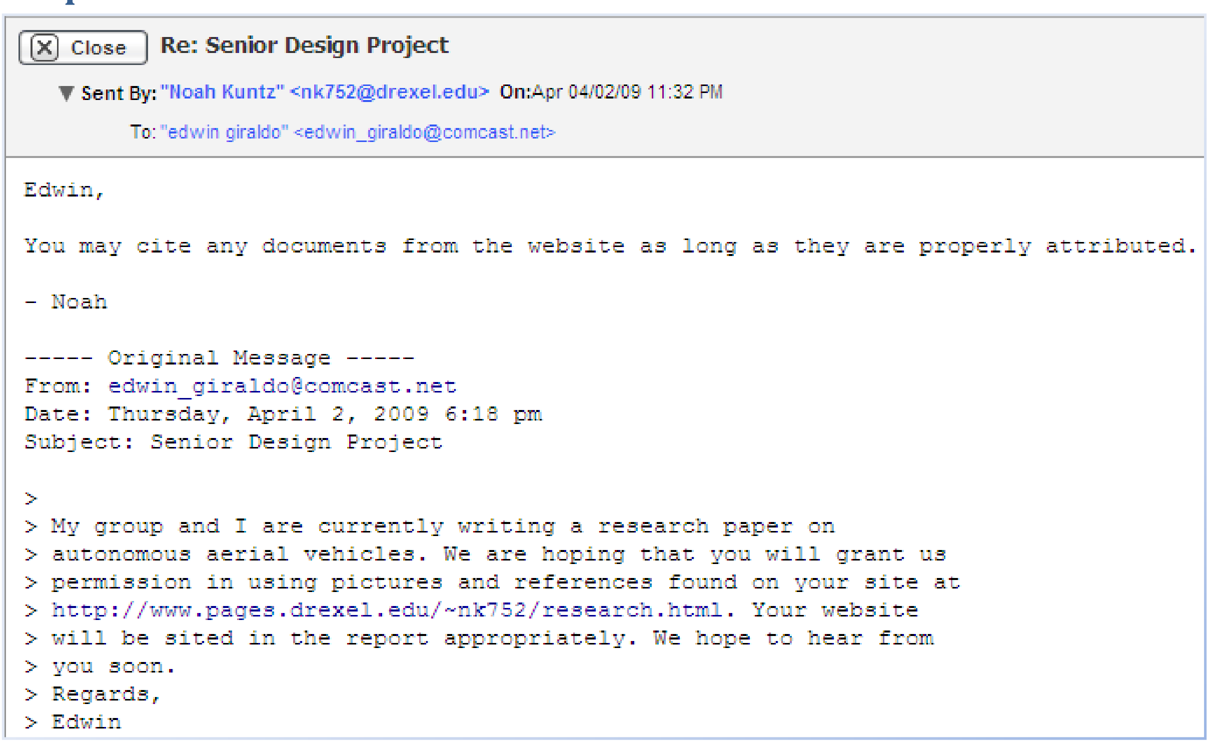
You are here: Home > News, Product Articles, Press Releases & Media Info

News & Announcements | November 13, 2006

Images for Media

Please feel free to download the following high resolution graphics and use them in your magazines, newspapers and weblogs. We recommend using the Photoshop (.psd) images because they are layered and have the background extracted.

Chapter 4: Research


Close | **Re: Senior Design Project**

Sent By: "Noah Kuntz" <nk752@drexel.edu> On: Apr 04/02/09 11:32 PM

To: "edwin giraldo" <edwin_giraldo@comcast.net>

```
Edwin,

You may cite any documents from the website as long as they are properly attributed.

- Noah

----- Original Message -----
From: edwin_giraldo@comcast.net
Date: Thursday, April 2, 2009 6:18 pm
Subject: Senior Design Project

>
> My group and I are currently writing a research paper on
> autonomous aerial vehicles. We are hoping that you will grant us
> permission in using pictures and references found on your site at
> http://www.pages.drexel.edu/~nk752/research.html. Your website
> will be sited in the report appropriately. We hope to hear from
> you soon.
> Regards,
> Edwin
```

From: <matt-brown@hooked-on-rc-airplanes.com>
To: <MagneticEnergies@cfl.rr.com>
Sent: Friday, April 03, 2009 3:52 PM
Subject: Re: Submission from "Contact" form in hooked-on-rc-airplanes.com

> Hello Clint,
> Feel free to use any images you want.
> Sounds like a fun project! Not nearly as fun as my senior design project
> ,lol. What school are you attending?
> Please let me know how it turns out! I would love to see some pictures,
> or video!
>Good luck!
>
> Matt
> matt-brown@hooked-on-rc-airplanes.com writes:
>
>> On Fri Apr 03 15:19:54 2009, the following results were submitted from
>> the "Contact" on hooked-on-rc-airplanes.com:
>>
>> -----------------------------
>> E-mail Address: MagneticEnergies@cfl.rr.com
>> Message:: Hello my name is Clint Mansfield and I am working on a senior
>> design project. We are making a quad copter. I would like to have
>> permission to use a couple of images on your site. They are of the
>> inrunner and outrunner motors. heres the link.
>>
>> http://www.hooked-on-rc-airplanes.com/brushless-rc-motors.html

castle™

Products | Support | Information | Media | Downloads | Factory Team | Home

Media > Castle Scribe | Media Kits | Movies | News | Press Room

Media Kits

Our media kits include files in TIFF, JPEG, and, when applicable, EPS (vector) formats. All available file types for each product have been compressed into one ZIP file to ensure faster downloads.

Note: EPS files require vector-compatible software to open.

Logos and photos may be downloaded and used subject to the following conditions:

- Logos may not be altered in any way, including font, color, content or format.
- Logos and photos may be re-sized as needed, but height and width must be constrained in the appropriate proportion. Please do not distort, stretch, or skew the images.
- The logo must be used in whole; parts may not be extracted for individual use.

If you need a different format than what is included here, please contact Nick del Castillo, Art Director, at nick@castlecreations.com.

Fwd: [Fwd: Permission request for use of website picture]

From: **Boe AnnDrea** (Anndrea@sparkfun.com)
Sent: Fri 4/17/09 1:19 PM
To: jeremy_brooks@knights.ucf.edu

Hello Jeremy! Yep, you have our permission to use our photos for your paper/website. We give permission to anyone that would be using our photos for educational purposes. Please pass along our permission to any of your peers who might be wondering the same thing.

Wish you could have made it to our Autonomous Vehicle Contest with your quadrocopter!!

AnnDrea Boe

Director of Marketing Communications
SparkFun Electronics
6175 Longbow Drive, Suite 200
Boulder, CO 80301

Begin forwarded message:

From: <jeremy_brooks@knights.ucf.edu>
Date: April 16, 2009 5:49:30 PM MDT
To: <customerservice@sparkfun.com>, <website@sparkfun.com>
Subject: Permission request for use of website picture

Hello,

I'm an EE student at the University of Central Florida and my group and I are working on an autonomous 4 propeller helicopter similar to the Draganflyer for our senior design project. We are writing our senior design documention paper and we are using some parts that we purchased or will purchase from you. Is it ok if we use some pictures from your website for our paper? I hope to hear from you soon.

Thank you,
Jeremy Brooks

```
-----Original Message-----
From: Bruce Eisenhard [mailto:viskr@yahoo.com]
Sent: Tuesday, April 21, 2009 11:51 PM
To: MagneticEnergies@cfl.rr.com
Subject: Re: Information Request


Hi Clint-

Feel free to use the picture.  Give me a quote and a link and I'll add
your project to our website if you don't mind

brucee

--- On Tue, 4/21/09, MagneticEnergies@cfl.rr.com
<MagneticEnergies@cfl.rr.com> wrote:

> From: MagneticEnergies@cfl.rr.com <MagneticEnergies@cfl.rr.com>
> Subject: Information Request
> To: info@coridiumcorp.com
> Date: Tuesday, April 21, 2009, 6:57 PM
> Hi, I am a EE student working on my
> senior design project. We are building a quadrotor. We plan
> to use your ARMmite controller and i am just asking
> permission to use the picture of it you have.
>
> Thank You,
>
>       Clint
>
```

Appendix B: Work Cited

Executive Summary

[1] "Database index." Plane Crash Info.com. 28 Mar. 2009 [Online]. Available: http://www.planecrashinfo.com/database.htm.

[2] "Welcome." DARPA | Home. 28 Mar. 2009 [Online]. Avialable: http://www.darpa.mil/grandchallenge/index.asp.

Chapter 2: Definitions

[3] "Mini-Pro1." Airfoil Helicam. 28 Mar. 2009 [Online]. Available: http://www.airfoilhelicam.com/minipro1.php.

[4] "Request A Quote - Contact Draganfly Innovations Inc. Draganfly.com." Draganfly.com Industrial Aerial Video Systems & UAVs. 28 Mar. 2009 [Online]. Avialable: http://www.draganfly.com/quote/DF/Helicopters/DF-X6/Base-Model/.

[5] "RE: Building a DraganFlyer - need to mix speedcontrollers - HELP." RC Universe features rc cars, rc airplanes, rc helicopters, rc electric planes, rc boats, radio control jets, rc discussion forums, rc classifieds and auctions. 28 Mar. 2009 [Online]. Available: http://www.rcuniverse.com/forum/m_1487065/mpage_13/key_/tm.htm.

[6] "Draganflyer V Ti RC Gyro Stabilized Electric Helicopter With Wireless Video Camera System." RCToys.com Sells RC Airplanes RC Blimps RC Helicopters & Parts. 28 Mar. 2009 [Online]. Available: http://www.rctoys.com/rc-toys-and-parts/DF-VTI-EYE/RC-HELICOPTERS-DRAGANFLYER-VTI.html.

[7] "Draganflyer X6 Helicopter Technical Specifications - Draganfly Innovations Inc." Draganfly.com Industrial Aerial Video Systems & UAVs. 28 Mar. 2009 [Online]. Available: http://www.draganfly.com/uav-helicopter/draganflyer-x6/specifications/.

[8] "Polar Remote Sensing Using an Unpiloted Aerial Vehicle (UAV)." Cooperative Institute for Research in Environmental Sciences (CIRES). 28 Mar. 2009 [Online]. Available: http://cires.colorado.edu/~maurerj/class/UAV/aerosonde.htm.

[9] "Tiltrotor transport UAV : Empyrean Studio." Mike Spike Dot Com. 28 Mar. 2009 [Online]. Available: http://www.mike-spike.com/wp/rd/aerospace/tiltrotor-transport-uav/.

[10] Aerosonde. 28 Mar. 2009 [Online]. Available: http://www.aerosonde.com/.

[11] "Factsheets : MQ-1 Predator Unmanned Aircraft System : MQ-1 Predator Unmanned Aircraft System." Air Force Link - Home. 28 Mar. 2009 [Online]. Available: http://www.af.mil/factsheets/factsheet.asp?fsID=122.

[12] "Predator RQ-1 / MQ-1 / MQ-9 Reaper - Unmanned Aerial Vehicle (UAV) -." Air Force Technology. 28 Mar. 2009 [Online]. Available: http://www.airforce-technology.com/projects/predator/.

[13] "Embedded.com - PID without a PhD." Embedded Systems Design - Embedded.com. 28 Mar. 2009 [Online]. Available: http://www.embedded.com/2000/0010/0010feat3.htm.

[14] "ADI - Analog Dialogue | iMEMS Gyro." Analog Devices, Inc.: Converters Amplifiers Processors MEMS A/D Converters Analog to Digital Video Converter Temperature Sensors Analog Device RF Amplifiers Differential Amplifiers Digital Signal Processing Thermal Management D to A Converters Microcontroller and Integrated Circuit Systems Embedded Processing and DSP. 29 Mar. 2009 [Online]. Available: http://www.analog.com/library/analogDialogue/archives/37-03/gyro.html.

[15] "Accelerometer Design and Applications | FAQs | FAQs | Technical Library | Design Center | Analog Devices." Analog Devices, Inc.: Converters Amplifiers Processors MEMS A/D Converters Analog to Digital Video Converter Temperature Sensors Analog Device RF Amplifiers Differential Amplifiers Digital Signal Processing Thermal Management D to A Converters Microcontroller and Integrated Circuit Systems Embedded Processing and DSP. 29 Mar. 2009 [Online]. Available: http://www.analog.com/en/technical-library/faqs/design-center/faqs/CU_faq_MEMs/resources/fca.html#SECTION_A.

[16] B. Taylor, C. Bil, S. Watkins, and G. Egan, "Horizon Sensing Attitude Stabilisation: A VMC Autopilot" IEEE Journal 2003.

[17] Yunchun Yang, "Magnetometer and Differential Carrier Phase GPS-Aided INS for Advanced Vehicle Control", IEEE journal on Transaction on Robotics and Automation, Vol. 19 No. 2. April 2003.

Chapter 3: Specifications and Requirements

[18] E-flite - Advancing Electric Flight. 01 Apr. 2009 [Online]. Available: http://www.e-fliterc.com/Products/Escs.aspx.

Chapter 4: Research

[19] "Propellers and Rotors." 04 Apr. 2009 [Online]. Available: http://www.centennialofflight.gov/essay/Theories_of_Flight/props/TH18.htm.

[20] "Understanding Brushless RC Motors." RC Airplanes, Basics to Extreme! 05 Apr. 2009 [Online]. Available: http://www.hooked-on-rc-airplanes.com/brushless-rc-motors.html.

[21] Research Online. 05 Apr. 2009 [Online]. Available: http://ro.uow.edu.au/cgi/viewcontent.cgi?article=1099&context=infopapers.

[22] "Understanding Brushless RC Motors." RC Airplanes, Basics to Extreme! 06 Apr. 2009 [Online]. Available: http://www.hooked-on-rc-airplanes.com/brushless-rc-motors.html.

[23] "How Motors Work - stefanv.com." Home Page - stefanv.com. 06 Apr. 2009 [Online]. Available: http://www.stefanv.com/rcstuff/qf200212.htm.

[24] **"**Backstepping based Nonlinear Flight Control Strategy for 6 DOF Aerial Robot" Ashfaq Ahmad Mian, Mian Ilyas Ahmad, Daobo Wang pp. 15637-15638, 2008

[25] "FMA Direct : Product Detail." FMA Direct - High Quality Equipment for the R/C Enthusiast. 06 Apr. 2009[Online]. Available: http://www.fmadirect.com/detail.htm?item=1489&section=1.

[26] Phillip M^{c}Kerrow. "Modelling the Draganflyer four-rotor helicopter." IEEE International Conference on Robotics & Automation. April 2004.

[27] Nick Sacco. "How the Draganflyer Flies". Rotary Magazine.

[28] "Infrared Vs. Ultrasonic – What You Should Know" 11 Apr. 2009. [Online]. Available: http://www.societyofrobots.com/member_tutorials/node/71.

[29] "Encyclopedia of Laser Physics and Technology - distance measurements with lasers, range finding, time of flight, triangulation, phase shift, laser distance meters." RP Photonics Consulting - laser and amplifier design, nonlinear optics, fiber optics, fiber lasers and amplifiers, ultrashort pulses, optoelectronics, consultant, training. 11 Apr. 2009 [Online]. Available: http://www.rp-photonics.com/distance_measurements_with_lasers.html.

[30] "General Questions." Disto measuring devices and accessories - Distagage.com. 11 Apr. 2009 [Online]. Available: http://www.distagage.com/gequ.html.

[31] "Choosing an Ultrasonic Sensor for Proximity or Distance Measurement Part 1: Acoustic Considerations - Sensors." Sensors - Home Page. 11 Apr. 2009 [Online]. Available: http://www.sensorsmag.com/articles/0299/acou0299/index.htm.

[32] "Brushless ESC Speed Controllers." RCToys.com Sells RC Airplanes RC Blimps RC Helicopters & Parts. 16 Apr. 2009 [Online] Available: http://www.rctoys.com/rc-products-catalog/RC-PARTS-SPEED-CONTROLLERS.html.

[33] "Introduction to RC Helicopter Gyros, Rate & Heading Hold." RCHelicopter.com is Your Source for RC Helicopter News & Information. 16 Apr. 2009 [Online]. Available: http://www.rchelicopter.com/2008/02/13/rc-helicopter-gyro-introduction-rate-gyro-heading-hold-gyro-explained/.

[34] "RC Battery Tips." Nitro Powered RC Cars. 21 Apr. 2009. [Online]. Available: http://www.rccarssite.com/rcbattery.htm.

[35] "What's the best battery?" Welcome to Battery University. 21 Apr. 2009. [Online]. Available: http://www.batteryuniversity.com/partone-3.htm.

[36] "The Nickel-based battery, its dominance and the future." Welcome to Battery University. 21 Apr. 2009 [Online]. Available: http://batteryuniversity.com/partone-4.htm.

[37] "Is lithium-ion the ideal battery?" Welcome to Battery University. 21 Apr. 2009 [Online]. Available: http://www.batteryuniversity.com/partone-5.htm.

[38] "Texas Instruments - Terms of Use." Analog Technologies, Semiconductors, Digital Signal Processing - Texas Instruments. 22 Apr. 2009 [Online]. Available: http://www.ti.com/corp/docs/legal/termsofuse.htm?DCMP=TIFooterTracking&HQS=Other+OT+footer_terms.

Chapter 5: Component System Design

[39] "The GPRMC Sentence." Startpage. 21 Apr. 2009 [Online]. Available: http://www.codepedia.com/1/The+GPRMC+Sentence.

[40] "CodeProject: Writing Your Own GPS Applications: Part 2. Free source code and programming help." CodeProject. Free source code and programming help. 21 Apr. 2009 [Online]. Available: http://www.codeproject.com/KB/mobile/WritingGPSApplications2.aspx.

[41] "Batteries for radio controlled models." Radio Controled Model Reviews. 23 Apr. 2009 [Online]. Available: http://www.rcmodelreviews.com/baffledbybatteries3.shtml.

Appendix C: Sample Code

Class NmeaInterpreter ( )

```
//*********************************************************************
//** A high-precision NMEA interpreter
//** Written by Jon Person, author of "GPS.NET" (www.gpsdotnet.com)
//*********************************************************************
using System;
using System.Globalization;

public class NmeaInterpreter
{
 // Represents the EN-US culture, used for numers in NMEA sentences
 public static CultureInfo NmeaCultureInfo = new CultureInfo("en-US");
 // Used to convert knots into miles per hour

 #region Delegates
  public delegate void PositionReceivedEventHandler(string latitude,
   string longitude);
  public delegate void FixObtainedEventHandler();
  public delegate void FixLostEventHandler();
  public delegate void SatelliteReceivedEventHandler(
   int pseudoRandomCode, int azimuth, int elevation,
   int signalToNoiseRatio);
  public delegate void HDOPReceivedEventHandler(double value);
  public delegate void VDOPReceivedEventHandler(double value);
  public delegate void PDOPReceivedEventHandler(double value);
 #endregion

 #region Events
  public event PositionReceivedEventHandler PositionReceived;
  public event FixObtainedEventHandler FixObtained;
  public event FixLostEventHandler FixLost;
  public event SatelliteReceivedEventHandler SatelliteReceived;
  public event HDOPReceivedEventHandler HDOPReceived;
  public event VDOPReceivedEventHandler VDOPReceived;
  public event PDOPReceivedEventHandler PDOPReceived;
 #endregion

 // Processes information from the GPS receiver
 public bool Parse(string sentence)
 {
  // Discard the sentence if its checksum does not match our
  // calculated checksum
  if (!IsValid(sentence)) return false;
  // Look at the first word to decide where to go next
  switch (GetWords(sentence)[0])
  {
   case "$GPRMC":
    // A "Recommended Minimum" sentence was found!
```

```
      return ParseGPRMC(sentence);
    case "$GPGSV":
      // A "Satellites in View" sentence was received
      return ParseGPGSV(sentence);
    case "$GPGSA":
      return ParseGPGSA(sentence);
    default:
      // Indicate that the sentence was not recognized
      return false;
  }
}

// Divides a sentence into individual words
public string[] GetWords(string sentence)
{
  return sentence.Split(',');
}

// Interprets a $GPRMC message
public bool ParseGPRMC(string sentence)
{
  // Divide the sentence into words
  string[] Words = GetWords(sentence);
  // Do we have enough values to describe our location?
  if (Words[3] != "" && Words[4] != "" &&

    Words[5] != "" && Words[6] != "")
  {
    // Yes. Extract latitude and longitude
    // Append hours
    string Latitude = Words[3].Substring(0, 2) + "°";
    // Append minutes
    Latitude = Latitude + Words[3].Substring(2) + "\"";
    // Append hours
    Latitude = Latitude + Words[4]; // Append the hemisphere
    string Longitude = Words[5].Substring(0, 3) + "°";
    // Append minutes
    Longitude = Longitude + Words[5].Substring(3) + "\"";
    // Append the hemisphere
    Longitude = Longitude + Words[6];
    // Notify the calling application of the change
    if(PositionReceived != null)
      PositionReceived(Latitude, Longitude);
  }
  // Does the device currently have a satellite fix?
  if (Words[2] != "")
  {
    switch (Words[2])
    {
      case "A":
        if(FixObtained != null)
          FixObtained();
        break;
```

```
        case "V":
          if(FixLost != null)
            FixLost();
          break;
      }
    }
    // Indicate that the sentence was recognized
    return true;
  }

  // Interprets a "Satellites in View" NMEA sentence
  public bool ParseGPGSV(string sentence)
  {
    int PseudoRandomCode = 0;
    int Azimuth = 0;
    int Elevation = 0;
    int SignalToNoiseRatio = 0;
    // Divide the sentence into words
    string[] Words = GetWords(sentence);
    // Each sentence contains four blocks of satellite information.
    // Read each block and report each satellite's information
    int Count = 0;
    for (Count = 1; Count <= 4; Count++)
    {
      // Does the sentence have enough words to analyze?
      if ((Words.Length - 1) >= (Count * 4 + 3))
      {
        // Yes.  Proceed with analyzing the block.
        // Does it contain any information?
        if (Words[Count * 4] != "" && Words[Count * 4 + 1] != ""

          && Words[Count * 4 + 2] != "" && Words[Count * 4 + 3] != "")
        {
          // Yes. Extract satellite information and report it
          PseudoRandomCode = System.Convert.ToInt32(Words[Count * 4]);
          Elevation = Convert.ToInt32(Words[Count * 4 + 1]);
          Azimuth = Convert.ToInt32(Words[Count * 4 + 2]);
          SignalToNoiseRatio = Convert.ToInt32(Words[Count * 4 + 3]);
          // Notify of this satellite's information
          if(SatelliteReceived != null)
            SatelliteReceived(PseudoRandomCode, Azimuth,
            Elevation, SignalToNoiseRatio);
        }
      }
    }
    // Indicate that the sentence was recognized
    return true;
  }

  // Interprets a "Fixed Satellites and DOP" NMEA sentence
  public bool ParseGPGSA(string sentence)
  {
    // Divide the sentence into words
```

```
  string[] Words = GetWords(sentence);
  // Update the DOP values
  if (Words[15] != "")
  {
   if(PDOPReceived != null)
    PDOPReceived(double.Parse(Words[15], NmeaCultureInfo));
  }
  if (Words[16] != "")
  {
   if(HDOPReceived != null)
    HDOPReceived(double.Parse(Words[16], NmeaCultureInfo));
  }
  if (Words[17] != "")
  {
   if(VDOPReceived != null)
    VDOPReceived(double.Parse(Words[17], NmeaCultureInfo));
  }
  return true;
 }

 // Returns True if a sentence's checksum matches the
 // calculated checksum
 public bool IsValid(string sentence)
 {
  // Compare the characters after the asterisk to the calculation
  return sentence.Substring(sentence.IndexOf("*") + 1) ==
   GetChecksum(sentence);
 }

 // Calculates the checksum for a sentence
 public string GetChecksum(string sentence)
 {
  // Loop through all chars to get a checksum
  int Checksum = 0;
  foreach (char Character in sentence)
  {
   if (Character == '$')
   {
    // Ignore the dollar sign
   }
   else if (Character == '*')
   {
    // Stop processing before the asterisk
    break;
   }
   else
   {
    // Is this the first value for the checksum?
    if (Checksum == 0)
    {
     // Yes. Set the checksum to the value
     Checksum = Convert.ToByte(Character);
    }
```

```
        else
        {
          // No. XOR the checksum with this character's value
          Checksum = Checksum ^ Convert.ToByte(Character);
        }
      }
    }
    // Return the checksum formatted as a two-character hexadecimal
    return Checksum.ToString("X2");
  }
}
```

Class HighPrecisionTest ( )

```
// High precision class
public class HighPrecisionTest
{
  private NmeaInterpreter MyInterpreter = new NmeaInterpreter();
  private int MaximumDOPAllowed = 6;
  private double CurrentHDOP;

  public HighPrecisionTest()
  {
    // Bind events for dilution of position
    MyInterpreter.HDOPReceived += new System.EventHandler(OnHDOPReceived);
    MyInterpreter.PositionReceived += new System.EventHandler(OnPositionReceived);
  }

  public void Test()
  {
    // Parse satellite information (HDOP is 50.0)
    MyInterpreter.Parse(
    "$GPGSA,A,1,,,,,,,,,,,,,50.0,50.0,50.0*05");
    // Parse the current position
    MyInterpreter.Parse(
    "$GPRMC,225233.990,V,3939.4000,N,10506.4000,W,0.00,51.40,280804,,*35");
    // Parse satellite information (HDOP is 1.2)
    MyInterpreter.Parse(
    "$GPGSA,A,3,11,29,07,08,19,28,26,,,,,,2.3,1.2,2.0*30");
    // Parse the current position again
    MyInterpreter.Parse(
    "$GPRMC,012558.584,A,3939.7000,N,10506.7000,W,0.00,198.07,290804,,*11");
  }

  private void OnHDOPReceived(double value)
  {
    // Remember the current HDOP value
    CurrentHDOP = value;
  }

  private void OnPositionReceived(string latitude, string longitude)
  {
    // Is the HDOP at least six?
```

```
    if (CurrentHDOP <= MaximumDOPAllowed)
    {
      // Yes.  Display the current position
      Debug.WriteLine("You are here: " + latitude + ", " + longitude);
    }
    else
    {
      // No.  Discard this positional measurement
      Debug.WriteLine("The received location is not precise enough to use.");
    }
  }
}
```