BAT BOT LAUNCHER

DESIGN DOCUMENT

Group 39 - Abhishek Bhandari (anb4), Kousthubh Dixit (kmdixit2), Vyom Thakkar (vnt2)

ECE 445 Proposal- Spring 2020

TA : Jonathon Hoff

1. Introduction

1.1 Objective:

Problem statement: We are working on a project that was pitched by Jonathan Hoff. Jonathan and his research group developed a bio-inspired robotic flapping-wing bat robot that mimics the agility and efficiency of bats using silicone membrane wings [9] . The bat bot launcher used presently does not have a mechanism to control the initial launch condition of the bat [10]. It is important to control the initial condition of launch because the trajectory of the robot is highly dependent upon the initial launch condition.

Proposed solution:
We are developing the automated launcher release mechanism for the robot bat that will allow the user to control the timing delay associated with the launch as well as choose the trajectory that he/she requires the bat bot to take, and the launcher will launch the bat bot such that it takes the chosen trajectory.

We will use seven sensors that are vertically stacked such that each position corresponds to a specific wing position. This vertically stacked column of sensors will be placed on the side of the right wing of the bat bot. The lowest sensor will correspond to the lowest wing position and the highest sensor will correspond to the highest wing position possible. All the sensors in the middle will correspond to the respectively desired launch positions. The wings operate with a frequency of 8.5Hz. So, in $t = 1/f = 0.117647s$ the wing completes one oscillation. The wings move across an angle of 60 degrees from one extreme position to the other. Vertically they traverse 22 cm from one end to the other and back.

In the first flap of the wing, we will start recording data from our sensor output, and match all the timestamps of when the sensor outputs become high with the arrival of the wing at each of those positions. All these times would be relative to the top-most position of the wing. We will then negatively offset these launch times to compensate for the delay associated with the IR sensors (2ms) and the delay associated with the servo motor(0.04s), and trigger the launch of the bat bot based on the user input.

1.2 Background:

The Bat Bot or B2 mimics the complex flight kinematics and the unique flexibility, agility and efficiency present in the flight of bats, as is highlighted in the papers "*A biomimetic robotic platform to study flight specializations of bats*" by Ramezani, Chung and Hutchinson (2017) [11] and "*Reducing Versatile Bat Wing Conformations to a 1-DoF Machine*" by Hoff, Ramezani, Chung and Hutchinson (2017) [12].

The paper "*Trajectory planning for a bat-like flapping wing robot*" by Hoff, Syed, Ramezani and Hutchinson (2019) [13], discusses trajectory optimization and planning flight maneuvers for the Bat Bot (B2). The paper notes that the complex aerodynamics of the B2 wings make it extremely hard to plan flight trajectories and make the problem of trajectory optimization for the B2 a hard one to solve. The paper also highlights that there has not been much research done in the area of wing-flapping based robotic systems. We think that this is because the B2 is a very unique robot, and there are not many systems like the B2 in existence today. Because of this, to our knowledge, there have been no previous efforts made to design an automated launcher mechanism either for the B2 or a B2-like robotic system.

We looked at other flapping-based robotic systems like the Robofly, a wireless insect-like flying robot which was developed by researchers at University of Washington as highlighted by the paper "*Liftoff of a 190 mg Laser-Powered Aerial Vehicle: The Lightest Wireless Robot to Fly*" by James et al. (2018) [14], but we could not find any information about the launcher that they used for their robot. We also do not know if they had any automated launching mechanism for their robot. We also came across another wing-flapping based robotic system VAMP RC ornithopter, developed by Interactive Toy and referenced in the paper "*Optical Flow on a Flapping Wing Robot*" by Bermudez and Fearing (2009) [15], but we could not find any information about any automated launching mechanism that was being used. The lack of any existing solution and/or research on this topic is a challenge for our group, and something that we are looking forward to working on over the course of the semester.

As mentioned in our proposed solution for the project, the initial conditions of the Bat Bot determine the flight trajectory that it takes. In order to achieve consistency in flight trajectories during experimentation as well as use, we aim to develop an automated launcher for the Bat Bot that will allow the user to control the initial conditions like wing position and time delay that will determine the launch trajectory.

1.3 High-level requirements:

1)For a flapping frequency rate (8.5 Hz), the system must be able to accurately signal a launch to an angular precision of $\pi/40\ rads\ =\ 4.5°$ which corresponds to 0.02 seconds between two launches in a series of 10 launches.

2) System must accurately trigger the launch of the robot after a user-specified period of time utilizing the switches on the controller.

3) The system must be seamlessly integrated with the launcher in order to avoid collisions and interference with the launch path of the robot.

1.4 Visual Aid:

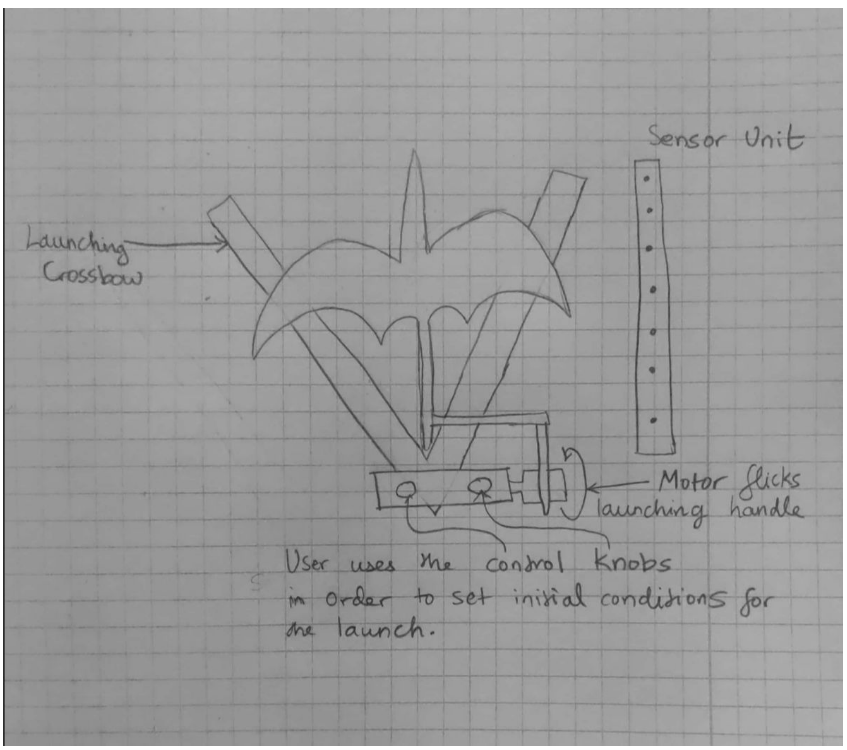

Figure 1: Visual Aid

2. Design

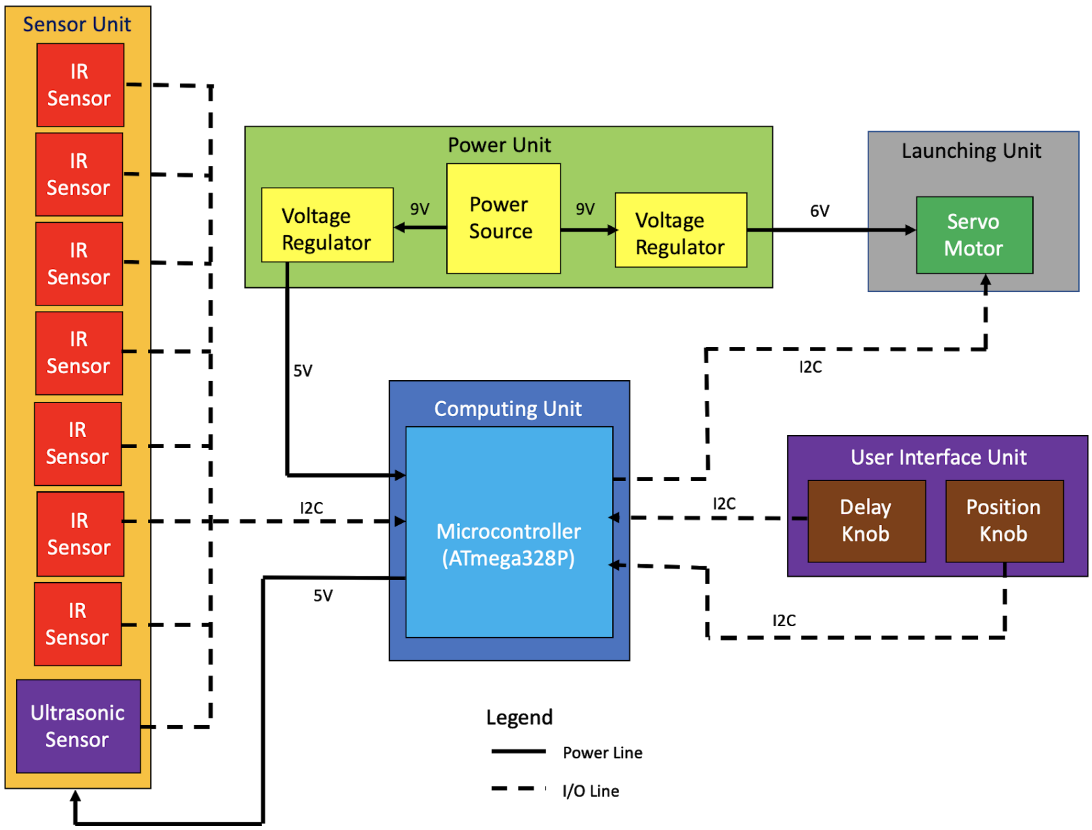

Figure 2. Block Diagram

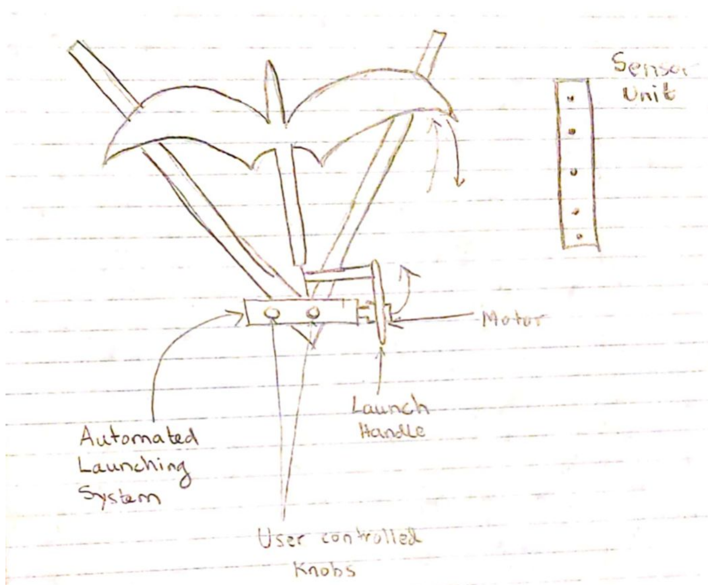

Figure 3. System Physical Design

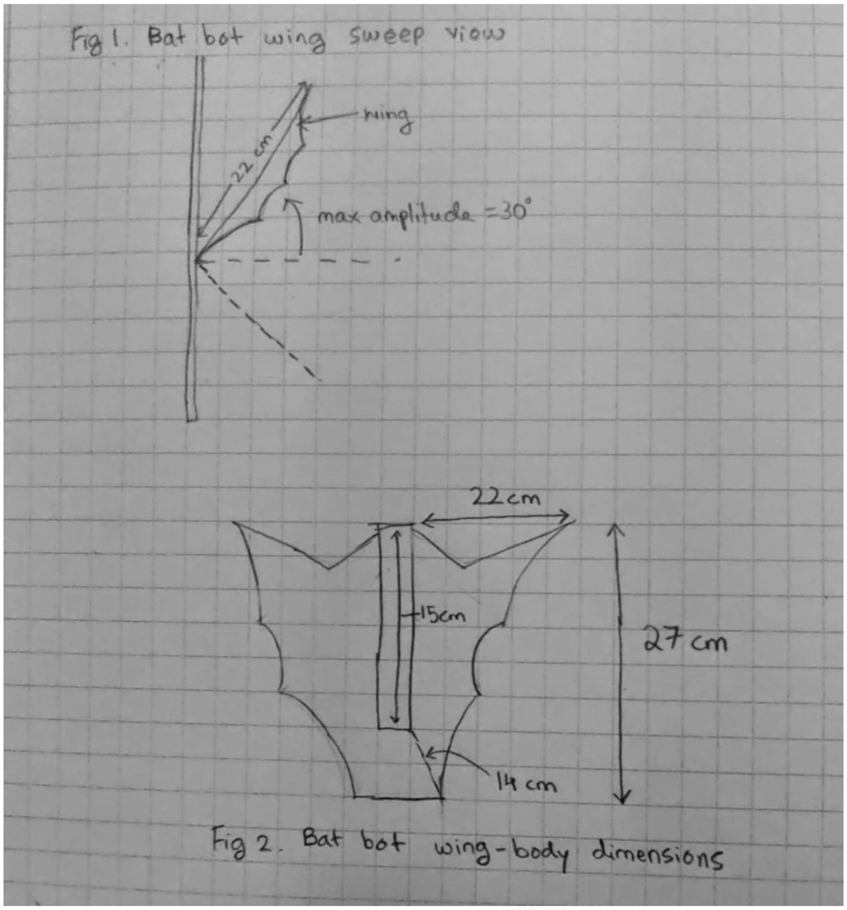

Figure 4. Physical design of the Bat Bot

2.1 Power Unit:

The function of the power unit is to provide stable DC power to all of the electrical components in our system at their specified operating voltage. The voltage regulators will be responsible for bucking the voltage and maintaining it at stable output voltages of 5V and 6V. Each sensor will draw about 20mA, the micro-servo motor will draw about 150mA, the microcontroller also requires a drive current of 300 mA. and the lithium battery that should be able to provide up to a continuous current of 1000mA.

2.1.1 Power Supply:

The power supply that we are planning to use to power our system is supplied by 9 volt lithium batteries. This will then be regulated by two voltage regulators to provide the appropriate voltages of 6V and 5V to the micro-servo motor and the sensor components respectively.

2.1.2 Voltage Regulator:

The voltage regulators in our system would step-down input 9V voltage (supplied by batteries):
1) to a steady 5V which is what is needed to power our microcontroller , and
2) to supply servo motor with 6V.
We have two voltage regulators: one for the microcontroller and the other for the servo motor since these two components have different voltage requirements as mentioned above. We plan on using off-the-shelf linear voltage regulators in our system. This is because linear regulators provide constant output voltages with less noise as compared to switching regulators. They are also rather inexpensive and since we only need to step-down our supply voltage to distribute to our different units, it is advisable to use a linear regulator.

2.2 Computing Unit:

The high level function of the computing unit is to gather sensor output as well as user provided launch-specific inputs (delay and position) to trigger the release of the bat robot by activating the servo motor of the launching unit. The computing unit will be programmed with an algorithm that models wing position (state of the bat robot) by processing the sensor output. The computing unit communicates with the sensor unit, user interface unit as well as the launching unit using I2C lines.

2.2.1 Microcontroller:

We plan on using an ATmega328P microcontroller chip for our project. The purpose of the microcontroller will be to interface with the sensor unit as well as the user interface unit to eventually trigger the launching unit by activating the motor. The communication between the microcontroller and the sensor unit, trigger controls unit as well as the motor will be through General Purpose IO (GPIOs) pins on the microcontroller.

2.3 Sensor Unit:

The high-level function of the sensor unit will be to accurately model the periodic wing motion of the wings of the bat robot. We plan on using an array of vertically placed IR sensors in order to model the position of the bat wings. The IR sensors will relay the analog output of the intensity of the reflected signal to the microcontroller for processing.
The data collected from the 5 IR sensors (each placed such that it corresponds to a particular wing position) will be used to extrapolate the launch time for the other 5 positions. We will verify and validate mathematically the relationship between our projected theoretical values for time taken to move from position to position and the observed values.
We will use an ultrasonic sensor to validate the data generated by the IR sensors.
Each of these sensors draw about 20mA.

2.4 User Interface Unit:

The user interface unit allows the user to specify the timing delay associated with the launch and the wing orientation from a list of seven preset values each. The user interface unit would consist of two control knobs each capable of enumerating seven different analog values corresponding to the seven different possibilities for each parameter.

2.5 Launching Unit:

The launching unit of our system interacts with the launching handle of the bat launcher (developed by Jonathan Hoff) to release the bat robot when triggered by the computing unit. The torque specifications are mentioned in the sub-system requirements.

2.5.1 Motor:

The function of the servo motor in our system would be to produce suitable torque to release the launching handle of the bat launcher when triggered to do so. It is suitably placed so as to not be an impediment to the launching path of the motor.

2.6 Circuit Schematic:

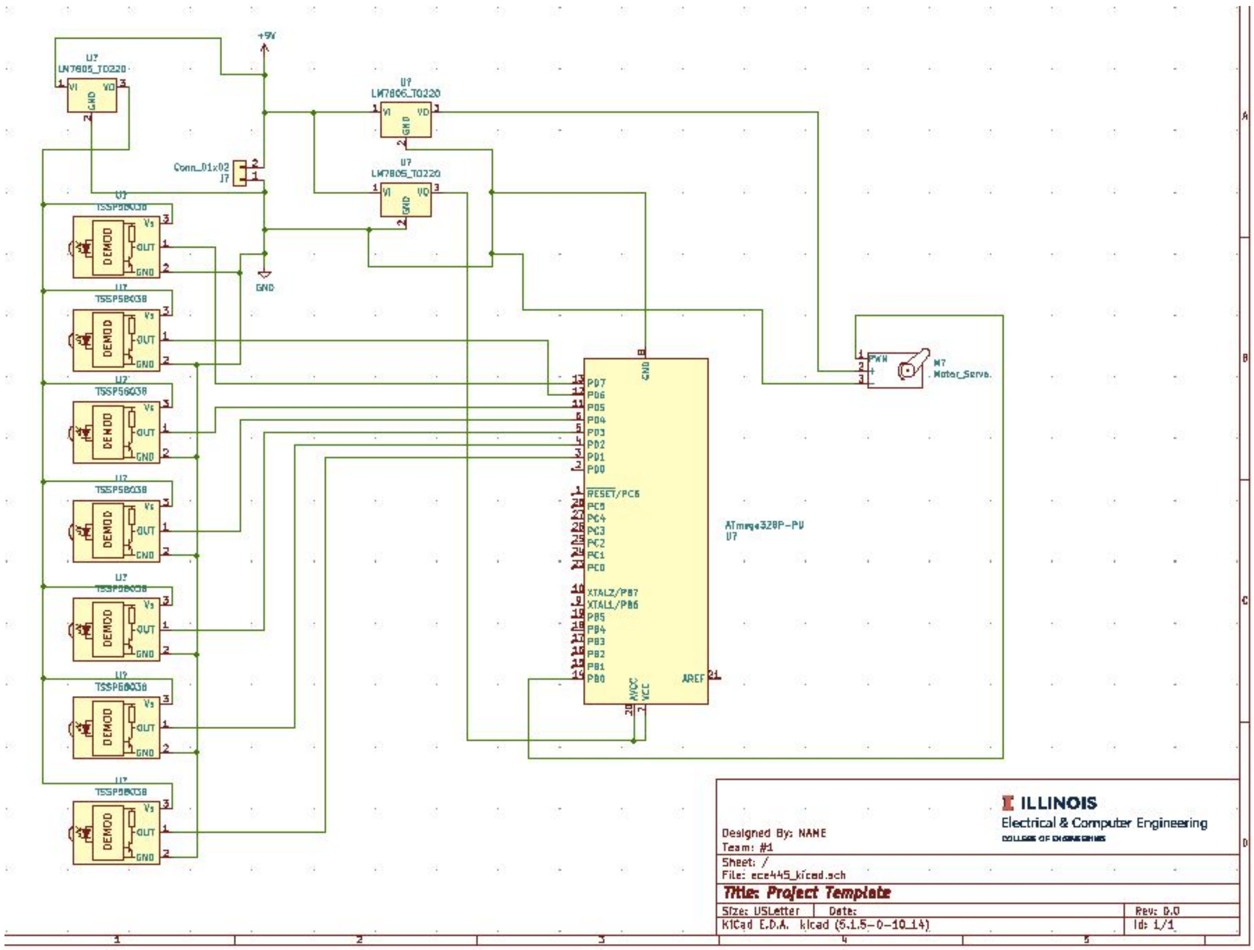

Figure 5: Circuit Schematic

Table 1. Subsystem Requirements and Verification Table

Subsystem	Requirements	Verification
Power Unit	1) Must be able to provide regulated, constant DC voltage based on the requirements of the computing unit (5V), and the servo motor (6V) with a tolerance of +- 5% in supply voltage. 2) Voltage regulator must be able to regulate voltage at 5V and 6V with 2% regulation.	1) We would measure the open-circuit voltage of the Li-ion battery to ensure that it is 9V 2) We would connect a multimeter and oscilloscope across the linear regulator to ensure that there is a stable output voltage and we would also check whether the output voltage is within the 2% tolerance range. We could also use 10 uF electrolytic capacitors at both the input and output pins of the regulator to ensure stability.
Computing Unit	1) The computing unit must correctly estimate wing position from sensor outputs to within 7.5% accuracy (half of 15% from the high level, as 7.5% error in both directions gives 15% error between two flights). In terms of phases, that is a phase change of $/40$ rads. 2) The total time of the microcontroller computation should be no more than 1/1000 of the time period of the completion of one flap.	1) 7.5% accuracy in terms of vertical distance corresponds to 1.65 cm, and an angular change of $/40$. It takes the wing 4.4ms to move through this distance. The way we will test for accuracy is we will place one IR sensor at the height that corresponds to our launch position, and two others: one 1.65 cm above and one 1.65 cm below. Based on the timestamp associated with the launch signal, we will check the outputs of the three IR sensors. Based on their outputs, we would know if the wing had exceeded 1.65 cm at the time of launch. Note: So for a given position, the middle IR sensor will output 1. If one of the other two sensor's outputs changes to a 1 from a 0 that means the wing managed to deviate by 1.65cm and hence did not pass the test. If the other two sensors change from a 1 to 0 or hold their value, that means that the wing orientation was within tolerance limits. 2) We would make use of in-built embedded C time libraries while programming our microcontroller to track the time taken to execute the program performing the computations.
Sensor Unit	1) Possess the ability to produce a digital high/low output corresponding	1) In order to verify the functionality of the IR sensors, we would connect pin 1

	to the wing motion at a detectable range of 1m with a drive current of 20 mA. 2) It should be possible to come up with a one-to-one correspondence between a point in the sensor output to a particular wing motion. 3) The sampling frequency of the sensors (kHz) must be of an order of 3 higher than the frequency of the wing motion. (Hz) to allow for accurate calibration and error minimization. 4) The sensing unit should be placed in a position that does not hinder the bat robot launch path.	to a LED and a pull-up resistor, pin 2 to ground and pin 3 to a power source that provides 5V to the sensor. We would then place an object place an object at 0.5m from the sensor and check whether or not the LED lights up. We would then remove the object and place it at a distance of 1.5m from the sensor and then observe whether or not the LED lights up. If the LED does light up we know that the sensor is detecting an object at 1.5m, which means that it would also detect an object at 1m as well. 2)We plan to place seven sensors corresponding to each position. We will then check when the output signals become high. We will compare this with the theoretical time taken for the wing to move through that distance relative to the top position(base position) (refer to equation 1 in the appendix). Every time a sensor's output signal turns high, it means that the wing is sweeping past the sensor. 3) In order to ensure that the unit does not hinder the launch path of the robot we would place our sensor unit at least 22cm(wingspan) away from the robot. We can verify the smoothness of the launch by testing a number of launches and checking whether the robot comes in contact with the unit. We could additionally have a proximity sensor placed in our sensor unit that would alert us if the robot's wings are too close.
User Interface Unit	1) The user interface unit must allow the user to calibrate launch specifications like wing position and launch delay time as desired from a choice of 7 preset delay and wing position parameters. 2) The preset wing position parameters must cover different possibilities of launch trajectories (upwards and downwards).	1) We would start a stopwatch immediately after the user sets the desired wing position and launch delay time, and observe the amount of time taken for the bat to launch. We could also mount an extra IR sensor on the launcher itself to detect the passing of the robot and use the timestamp**.** We would then verify whether the user defined launch delay time matched the time we recorded with the stopwatch

		accounting for tolerances in wing positions and sensor delays. 2) In order to verify whether the bat follows the trajectories listed, we would just use a camera and observe the launch frame by frame and check the direction in which the robot travels once it has been launched. We would then check whether the trajectory matched the way in which the robot theoretically must have travelled based on the user defined wing position. We can also corroborate this with our data from the computing unit verification.
Launching Unit	1) Must be capable of producing 0.01 N*m torque to launch the bat from the rest position by triggering the release of tension in the strings. 2)The servo motor itself must not in any way obstruct the path of the bat bot during its launch.	1) We would pull the bat robot back to its initial resting position using the strings on the launcher. After the motor gets the signal from the microcontroller we must see to it that the motor rotates by 30 degrees to release the tension in the strings. We can check whether or not the motor received a high signal by connecting it in series with a LED, if the LED lights up it means that the motor received a high and should have triggered the launch of the robot. 2) The launch of the robot takes place seamlessly and does not come into contact with the servo motor. We can verify this by testing a number of launches and checking that it does not come into contact with the motor.

2.6 Risk Analysis:

In the case of our launcher, the greatest risk to the success of our project is the failure of our sensor unit to produce data that can be used in a meaningful way (i.e noise). We have made sure that the sensor's sampling frequency is orders of magnitude higher than the frequency of motion of the wing, but despite that parameters such as the detection angle of each IR sensor is a limitation that we have to work around. We have used multiple IR sensors at equally spaced points such that they cover points spanning from each extremity in the wing's motion (Highest point to lowest point). Using mathematical modelling , we will make theoretical predictions of the time taken for the wing to move from position to position and calculate an error margin based on

when each sensor outputs a high signal. After incorporating the error margin and linearly scaling it between positions, we will be able to make more accurate predictions of the time taken to move from a given position to another. This technique will incorporate the effects of air resistance as well. An assumption made here is that between 2 launch positions, the error margin (deviation from theoretical predictions) is varying linearly. If while calibrating our sensors, we find that the non-linearity in error introduced due to air-drag is creating an error in our predictions, we may have to refine our mathematical model.
Another factor that could affect the performance of our IR sensors is that during outdoor testing light, dust, moisture and so on might interfere with the performance of IR sensors. As a work-around, we are planning to focus our IR sensor beams using a tunnel-like contraption around the sensors.

2.7 Tolerance Analysis:

The subsystem that we will be performing a tolerance analysis on, is our sensor unit. This is because the sensor unit is integral to the successful execution of our project, since it is the sensor unit that is responsible for modelling the wing movement. The infrared sensors that are a part of our sensor unit have two main parameters that are of interest to us in discussing the tolerance of our system: effectual angle and response time. The effectual angle of a sensor is the maximum angle at which an object can be placed with respect to the sensor in order to obtain accurate readings from the sensor [16].

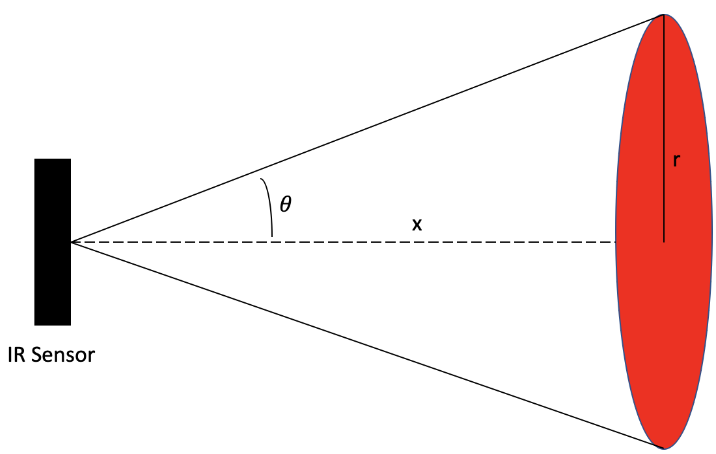

Figure 6: Effectual Angle

The above diagram illustrates the idea of the effectual angle of an IR sensor where θ represents the effectual angle, r represents the radius of the sensitive area that the IR sensor can accurately capture information about, which is placed at a horizontal distance of x from the sensor. In order to tune any given IR sensor to our application we can place the IR sensor at a particular horizontal distance which gives a desirable value of r (sensitivity of the sensor) for a given effectual angle. If we know the effectual angle of our IR sensor θ and the horizontal distance from the region of interest that our IR sensor is placed at (x), then we can compute r for the particular configuration as follows: $r = x \tan\theta$.

The second parameter of interest is the response time of an IR sensor. The response time is the maximum guaranteed time taken for the IR sensor to produce an output for any measurement that the sensor makes. For our application, the response time dictates how much the wing moves from the time that the sensor takes a measurement to the time that the sensor produces an output for the measurement. Since the wings flap with a reasonably high frequency of 8.5 Hz, it is imperative to pick an IR sensor that has a very small response time. We know that the wings span a vertical distance of 22cm with a periodic frequency of 8.5 Hz. Hence, the time that it takes for the wing to complete one vertical sweep is half the time period = (1/(8.5))/2 = 0.059s = 59 ms. Therefore, the distance (Δd) that the wing moves between the response time (Δt in ms) of an IR sensor can be expressed as follows: $\Delta d = (22 \cdot \Delta t) \div (59)$.
In order to choose the IR sensor for our project we compiled a list of different IR sensors available in the market and compared their response times and effectual angles as summarized in the table below:

Table 2: IR Sensor Table

IR Sensor	Response Time (ms)	Effectual Angle (degrees)
HD-DS25CM-3MM	**3 ms**	**10**
Sharp GP2D15	39 ms	2.15
TS105-10L5.5MM	20 ms	5
TP337	16 ms	30
MLX90616ESF-HCA	20 ms	40
ZTP-135SR	25 ms	40
ZTP-115	20 ms	30
ZTP-148SR	32 ms	30
ZTP-101T	22 ms	30

As can be seen from the table, the IR sensor HD-DS25CM-3MM has a reasonably small response time of 3ms and a reasonably low effectual angle of 10 degrees. We have chosen to use this sensor for our project because the low response time will ensure that Δd is low and the low effectual angle means that we will not have to place the IR sensor very close to the wings in order for r (the radius of the sensitive region) to be as small as possible. Since we have seven IR sensors in our project and each sensor must be capable of identifying a particular wing position, we want each IR sensor to focus on as narrow of a region as possible. If the radius of the sensitive area is very large then each IR sensor would output a digital high output for every wing position and thus, we would not be able to extract useful information from our sensor unit. Since we have seven IR sensors for our project we plan to place the seven IR sensors in a 24cm vertical stand with a 4cm separation between each IR sensor. As a design choice, we want each IR sensor to have a r value of 3cm which means that each sensor must capture a 3cm radius region out of the 22cm vertical sweep of the wing. We can now calculate x (horizontal distance from the wings from which to place the sensor stand) and Δd (error in wing position due to the response time of the IR sensor):

$$x = 3 \div \tan(10) = 17.01 \text{ cm}$$

$$\Delta d = (22 \cdot 3) \div 59 = 1.12 \text{ cm}$$

Thus, we must place the sensor stand 17 cm from the wings and the error in wing position due to response time is 1.12cm which corresponds to 5.09% worst-case error in wing position measurement for this configuration.

Below, we have simulated the IR sensor output for each of the seven IR sensors in the above mentioned configuration. Each plot captures digital sensor output (0 or 1) for each wing position in a 22cm vertical wing sweep (half a period). We have also incorporated the worst case wing position estimation error due to sensor response time.

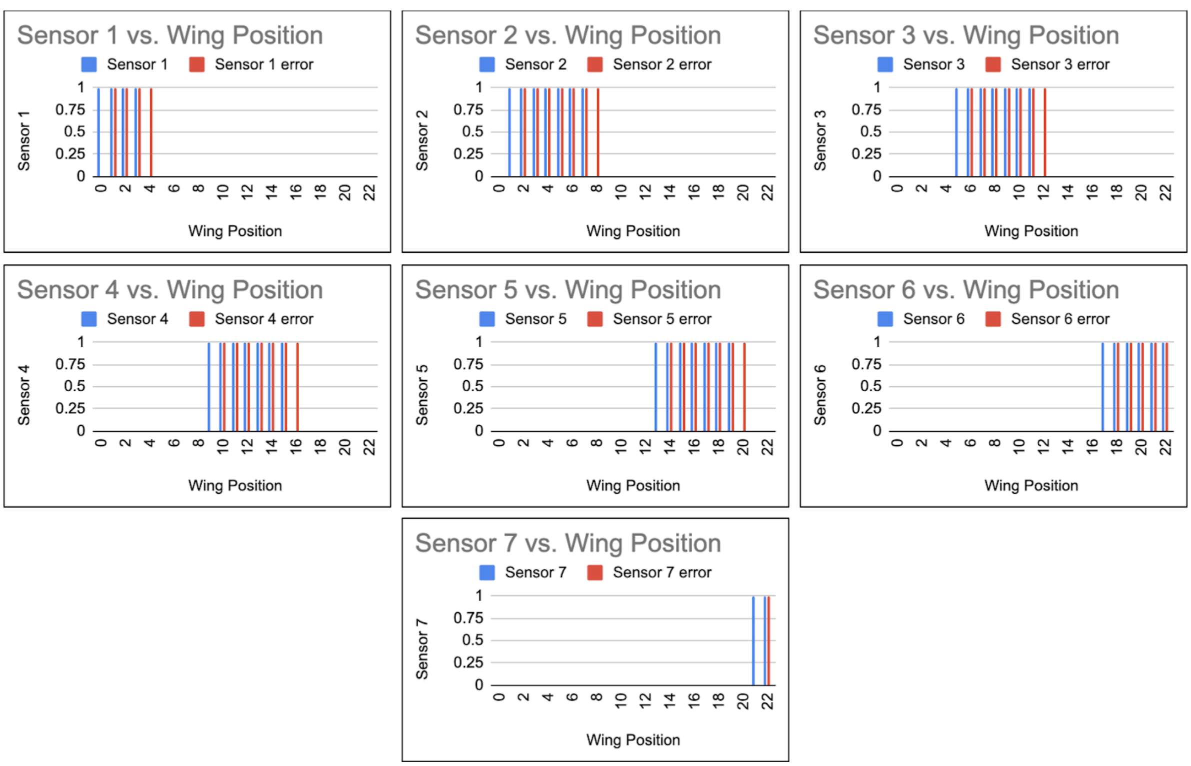

Figure 7: IR sensor output simulation error analysis

3. Costs :

We will assume that each of the 3 members spends 10 hours a week on this project for the remaining 11 weeks of the semester. Under this assumption we can calculate the total labor costs to be:

$3 * \$40/hr \ * 10\ hr/week \ * 11\ weeks \ = \$13200$.

Total component cost = $7 * 1.95 \ + \ 2.08 \ + 2 * 0.75 + 11.95 + 0.75 = \$29.93$

We must also include the cost of the electronic equipment necessary in the design:

PART	COST	BULK COST
IR Break Beam Sensor - 3mm LEDs (HD-DS25CM-3MM)	$1.95	$1.76
ATmega328P-PU	$2.08	$1.91
5V 1.5A Linear Voltage Regulator - 7805 TO-220	$0.75	$0.68
Micro Servo - High Powered, High Torque Metal Gear	$11.95	N/A
6V 1.5A Linear Voltage Regulator - 7806 TO-220	$0.75	$0.68

4.Schedule

Week of	Kousthubh Dixit	Abhishek Bhandari	Vyom Thakkar
2/24/20	Prepare design document	Prepare design document	Prepare design document
3/2/20	Finalize initial circuit schematic and start assembling circuit on breadboard.	Order IR sensors and commence initial testing, experiment with sensor placement and start testing on bat robot.	Begin discussions with the machine shop and work on the first iteration of the system physical design. Work on ordering the parts needed for each component.
3/9/20	Debug the breadboard circuit and finalize PCB design for early bird PCB	Continue initial testing and configuration of IR sensors and start coming	Come up with the first iteration of physical design using CAD. Send

	orders.	up with algorithm to process the sensor data.	the physical design to the machine shop for review. Work on algorithm to process sensor data.
3/16/20	Spring Break	Spring Break	Spring Break
3/23/20	If early bird PCB order did not pass audit/go to plan work on the first round of PCB orders. Start testing control knobs for the user interface unit and interface with microcontroller.	Work on getting the physical module of the system ready to test. Try to interface servo motor using microcontroller and calibrate the motor such that it is able to flick the handle of the launcher.	Start programming the algorithm for processing the sensor unit on the microcontroller and work on debugging.
3/30/20	Unit test the user interface unit and ensure that input can be successfully obtained from the user.	Unit test the launching unit and ensure servo motor functions as desired.	Unit test the sensor unit, ensure that it is able to model wing position accurately.
4/6/20	Work on integrating all the subsystems of the project.	Work on integrating all the subsystems of the project.	Work on integrating all the subsystems of the project.
4/13/20	Test the system on the actual bat bot launcher. Ensure that it meets the specifications of Jonathan Hoff.	Test the system on the actual bat bot launcher. Ensure that it meets the specifications of Jonathan Hoff.	Test the system on the actual bat bot launcher. Ensure that it meets the specifications of Jonathan Hoff.
4/20/20	Finishing touches and details.	Finishing touches and details.	Finishing touches and details.
4/27/20	Demo the project and work on mock presentation.	Demo the project and work on mock presentation.	Demo the project and work on mock presentation.
5/4/20	Project presentation and work on final paper.	Project presentation and work on final paper.	Project presentation and work on final paper.

5. Ethics and Safety :

There are a few things that we need to take care of during our project. The main safety concern with launching the bat bot is the safety of the environment, the people surrounding the outdoor testing areas/facilities, and ourselves. We must take the necessary precautions to preserve the safety of the surroundings and public in accordance with the first and ninth IEEE code of conduct and ethics [1]. We have to ensure that our testing area is isolated from the public.

The Infrared Sensors that we will be using in our project produce low-levels of infrared radiation that has negative impacts on the human eye in the case of extensive exposure. Manufacturers of IR sensors are required to adhere to the IEC-62471 Standards which relate to Eye Safety [3]. When we purchase IR sensors for our project we will make sure that they are produced by credible manufacturers. While testing our project, we will also make sure that these IR sensors are not directly firing on human eyes.

Since we will be using lithium batteries in order to power our project, a hazard that is often associated with them is called “thermal runaway” which results in overheating and battery failure [4]. Thermal runaway often causes the battery to ignite. In the case of such an emergency we will deal with the fire using appropriate training that was introduced in the mandatory safety training online module.

While working on our project we will also make sure to not expose the system to water and extensive dust which can cause damage to our circuit or other associated electrical components which can often result in safety hazards.

Since we will be presenting results, data, progress and other important findings and observations from our project throughout the course of the semester, we will do so in an honest and reliable manner by keeping in mind point 3 of the IEEE code of ethics [1] and points 1.3 and 2.2 of the ACM code of ethics [2].

We vow to value inclusivity in our work, celebrate diversity and reject discrimination in all forms by upholding point 8 of the IEEE code of ethics [1] and point 1.4 of the ACM code of ethics [2].

Over the course of the semester we will also be going through some of the previous work that was done by others and we will be using findings from previous work to guide certain aspects of our project. We vow to cite and credit other people’s work in accordance with point 7 of the IEEE code of ethics [1] and point 1.5 of the ACM code of ethics [2].

Furthermore, as mentioned in the online Safety Training modules, whenever we work in the ECE 445 laboratory, each of us will ensure that we are not working alone in the laboratory and that there is at least someone else present.

References :

[1] “IEEE Code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 13-Feb-2020].

[2] “The Code affirms an obligation of computing professionals to use their skills for the benefit of society.,” *Code of Ethics*. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 13-Feb-2020].

[3] “Eye Safety for Proximity Sensing Using Infrared Light-emitting Diodes,” *Renesas*, 28-Apr-2016. [Online]. Available: https://www.renesas.com/us/en/doc/application-note/an1737.pdf. [Accessed: 13-Feb-2020].

[4] “The Hazards of Lithium Batteries,” *Vice Provost Tufts*. [Online]. Available: https://viceprovost.tufts.edu//ehs/files/The-Hazards-of-Lithium-Batteries.pdf. [Accessed: 13-Feb-2020].

[5] J. Barani, “Difference between sensor response time and sensor time constant τ (tau) 63.2%.,” *BARANI DESIGN Technologies*, 07-May-2019. [Online]. Available: https://www.baranidesign.com/faq-articles/2019/5/6/difference-between-sensor-response-time-and-sensor-time-constant-tau. [Accessed: 25-Feb-2020].

[6] P. by D. P. E. Team, “Servo Motor Controlled Wireless Light Switch,” *Device Plus*, 29-Sep-2017. [Online]. Available: https://www.deviceplus.com/how-tos/arduino-guide/servo-motor-controlled-wireless-light-switch/. [Accessed: 25-Feb-2020].

[7] Adafruit Industries, “Micro Servo - High Powered, High Torque Metal Gear,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/2307?gclid=Cj0KCQiApt_xBRDxARIsAAMUMu8DJ-uuFs-6tOg6P9fujQ1yQXtjG4Daq4FBgwxpiNMOfPCaNIVJCmcaAkHdEALw_wcB. [Accessed: 25-Feb-2020].

[8] Adafruit Industries, “IR Break Beam Sensor - 3mm LEDs,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/2167?gclid=Cj0KCQiA4sjyBRC5ARIsAEHsELEQvSkj3I_fo28Zwd1InnSDBPLw06ZITB79wKVqhUBJxGf0ArZXZI0aAmTIEALw_wcB. [Accessed: 25-Feb-2020].

[9] J. Hoff, A. Ramezani, S.-J. Chung, and S. Hutchinson, “Robotics Proceedings ,” *Robotics Proceedings* , Jun-2016. [Online]. Available: http://www.roboticsproceedings.org/rss12/p09.pdf. [Accessed: 27-Feb-2020].

[10] J. Hoff, A. Ramezani, S.-J. Chung, and S. Hutchinson, “SAGE Journals,” *SAGE Journals*, 11-Nov-2018. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/0278364918804654. [Accessed: 27-Feb-2020].

[11] Ramezani, A., Chung, S. and Hutchinson, S. (2017). *A biomimetic robotic platform to study flight specializations of bats*. [online] Science Robotics. Available at: https://robotics.sciencemag.org/content/2/3/eaal2505 [Accessed 28 Feb. 2020].

[12] J. Hoff, A. Ramezani, S.-J. Chung, and S. Hutchinson, 16-Jul-2017. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-63537-8_16. [Accessed: 27-Feb-2020].

[13] J. Hoff, U. Syed, A. Ramezani and S. Hutchinson, "Trajectory planning for a bat-like flapping wing robot," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019, pp. 6800-6805.

[14] J. James, V. Iyer, Y. Chukewad, S. Gollakota and S. B. Fuller, "Liftoff of a 190 mg Laser-Powered Aerial Vehicle: The Lightest Wireless Robot to Fly," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 3587-3594.

[15] F. Garcia Bermudez and R. Fearing, "Optical flow on a flapping wing robot," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, 2009, pp. 5027-5032.

[16] “Ultrasonic Sensors,” *Cusp Sensors*. [Online]. Available: https://cuspsensors.files.wordpress.com/2015/05/ultrasonic_sensor_comparison.pdf. [Accessed: 27-Feb-2020].