# SPIKING NEURAL NETWORK DRONES
# FINAL PROJECT REPORT

May 1, 2024


University of Tennessee, Knoxville

COSC 402 Senior Design

**Team 1: AirBrane**

Tyler B. Nitzsche, Jihun Kim, Seoyoung An, Jonathan E. Skeen

**Advisor**

Dr. Catherine D. Schuman

# Executive Summary

The purpose of this project was to train and simulate a Crazyflie drone using a spiking neural network so that it can operate in a real-world environment by itself. Our goal was to demonstrate a drone taking off, hovering, and landing by itself in the simulation.

The first step was to train spiking neural networks (SNNs). SNNs are inspired from the human brain and take information from a given state to make computations based on that given data. By using a genetic algorithm, we have built spiking neural networks that are optimized to get the optimal output. We used EONS, Evolutionary Optimization for Neuromorphic Systems developed by TENNLab, for model training. EONS utilizes evolutionary optimization for the applications of spiking neural networks, and we used it to train our drone in a simulation environment to check its performance.

To test performance, we used a virtual simulation environment. We had three options: the pre-existing environment from last year's team, a simulator, and an environment from Dr. Simon D. Levy from Washington and Lee University, and to develop our own simulation environment if neither of these work. We chose to use a pre-existing environment from last year's team.

Our initial goal was that after testing the drone in the virtual environment, start to interact with and test our network on the hardware of the Crazyflie drone. However, due to time constraints, we left it as a future work.

Each step of training networks, observing performance, and adjusting reward function for improvement was repeated until we reached the goal: a drone that can successfully take off, hover, and land by itself without any problem in the simulation.

# Table of Contents

# List of Figures



# List of Tables

## PROBLEM DEFINITION & BACKGROUND

Autonomous drones have been a goal the artificial intelligence community has been working towards for decades at this point. Whether this idea is marvelous or terrifying, with the recent spark in interest for artificial intelligence, it seems as if the advancement of this technology is inevitable. Thus, our team has decided to approach this problem with neuromorphic computing. More specifically, we will train a spiking neural network to operate a small drone, performing a variety of simple tasks.

**Figure 1:** Crazyflie 2.1 Drone [1]

When approaching this problem today, one of the more popular approaches is deep learning – a subset of traditional neural networks. Traditional neural networks, as a whole, have seen an enormous rise in popularity over recent years, mainly due to their remarkable performance improvements. When comparing machine learning approaches for problem-solving, often deep learning is the path that many developers will pursue. While there are great results that people have had with deep learning, the downsides to these networks can outweigh the benefits.

While there is a continually growing list of concerns about using deep networks (such as the enormous amounts of data required to train these networks), our team's main concerns with using deep learning involve memory capacity and efficiency. Our network will need to operate within the drone's hardware. Thus, the network's storage and computational resources are severely limited, making a deep-learning approach unsuitable.

Enter, the spiking neural network (SNN). These networks operate differently when compared to traditional neural networks. Both types share the idea of neurons and synapses/edges, but spiking neural networks attempt to more closely replicate biology by incorporating time as a component of the network. In addition, SNNs often have fewer neurons and edges, helping with the storage concerns. Furthermore, they have relatively lower power consumption (due to the high average idleness of the neurons), helping with the computation concerns. These networks do, however, require the use of a specialized piece of hardware called a neuro-processor, but since we already have this hardware, this isn't a concern to us.

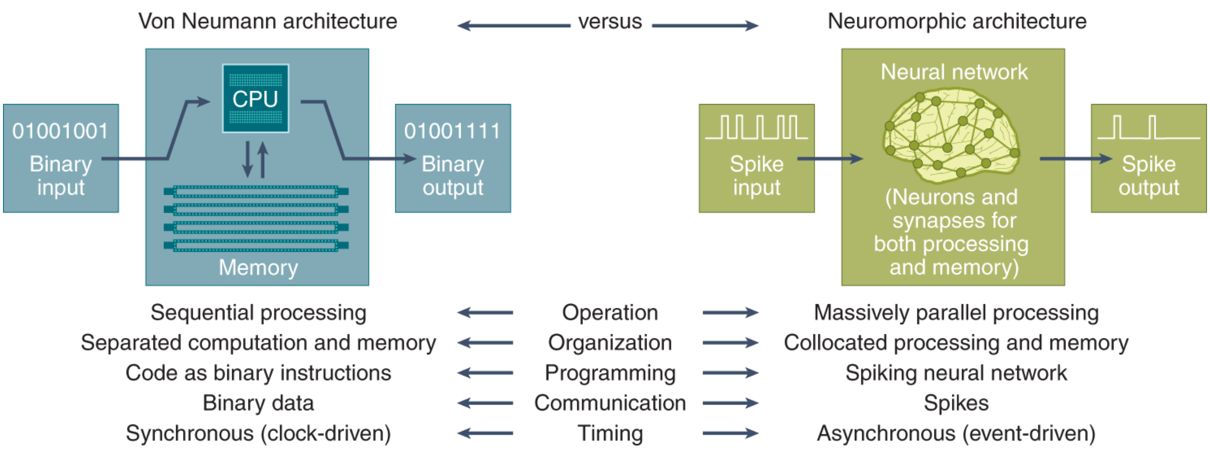


**Figure 2:** Comparison between traditional Von Neumann and Neuromorphic architectures [2]

Another consideration for this problem is determining the network structure. Even with limited inputs and outputs (less than 20 in total), there could be countless amounts of possible networks. Using SNNs makes these issues worse since these networks are usually not grid-like nor fully connected, as with traditional neural networks.

Yet another problem to consider for this task is the lack of knowledge regarding neural networks in general. Neural networks lack a key component essential for many problem-solving tasks: explainability. As of now, it is very difficult to accurately determine exactly how a network structure will affect output and the network itself cannot explain. Given all of this information, it becomes apparent that manually solving this problem is not optimal. Fortunately, Dr. Schuman and TENNLab have developed an algorithm that helps with creating model structures, tailored specifically to SNNs: Evolutionary Optimization for Neuromorphic Systems (EONS) [3]. Another algorithm attempting to replicate biology, EONS combines randomness and performance to enhance and modify spiking neural networks. Thus, this is our key to creating and training SNNs [3].

## THE REQUIREMENT SPECIFICATION

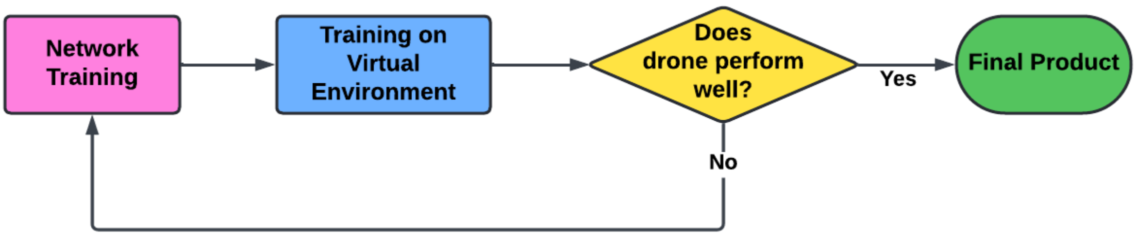


**Figure 3:** Overall Project Process

Our requirement specifications can be broken up into two main parts: Simulation and real-world specifications. Because our physical hardware is expensive and can pose a safety risk, the majority of our project will pertain to a simulation environment. This environment must contain three things to successfully fulfill our needs: the ability to simulate the drone within the environment, all relevant and useful data required by our SNN, and a high similarity to the real world to ease the transition process from the simulation to the physical environment. Thus, the first step in our process is to find the best simulation environment among the three options provided to us by Dr. Schuman. After an environment is chosen, our team must incorporate our drone into the environment. This process includes ensuring proper flight operation as well as proper sensor readings, as described in Fig 3.

After the preparation tasks are finished, we will begin developing various fitness functions for our networks. Maximizing these functions is the ultimate goal of SNNs and is our way of telling a network how it's performing. Correctly and accurately defining these functions is a notoriously difficult problem and will likely take up a sizeable chunk of our time. Since we are ambitious, we plan to define multiple fitness functions, which each correlate to a different task.

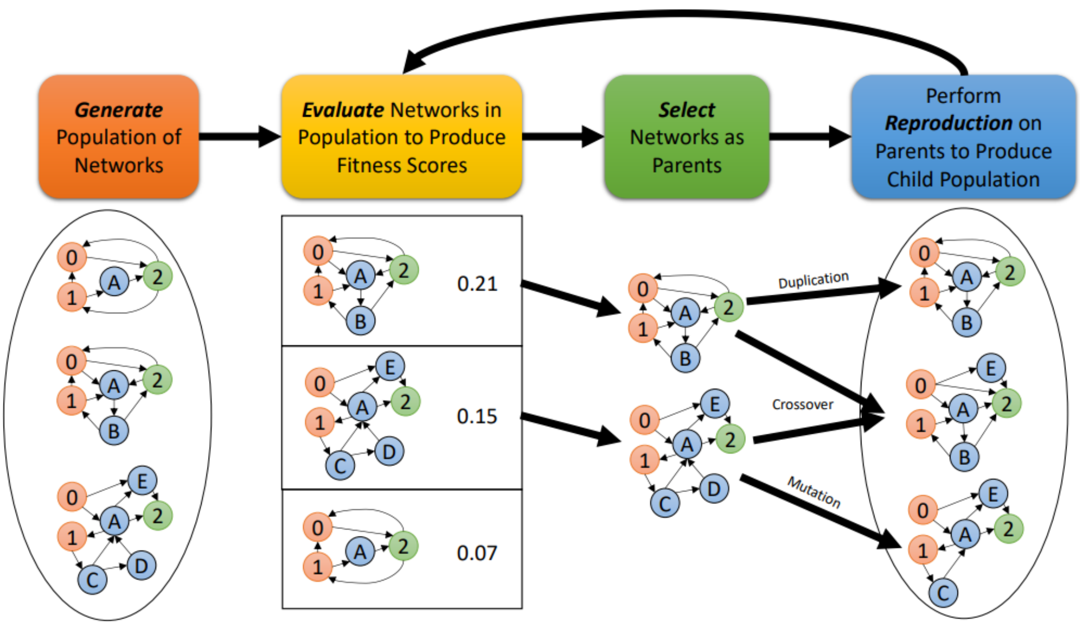


**Figure 4:** Overview of EONS Algorithm [3]

Once we have fine-tuned our fitness functions, we will implement the EONS algorithm into the simulation and begin training our networks [3]. Figure 4 shows each step of EONS [3]. Once again, we anticipate that this phase will also take up a large portion of our time [3]. If there is success in the simulation environment, we will begin the transition into real-world operation.

Transitioning into the physical environment will require a few different steps, with the first involving connecting to the drone API. This step could be completed in the simulation phase (when implementing the drone into the simulation environment), but if not, it will be the first task after finishing the simulation phase. From the simulation, we expect to get a few different network structures. We will load each of these networks onto the drone and measure how they perform in the real world. Just as with the simulation, we will have multiple fitness functions, ranging from very simple, to slightly complex, but the tasks will remain basic.

## TECHNICAL APPROACH

The project consists of three different components, all of which are unique and warrant their own approaches. The first component is the simulation environment, which is responsible for hosting the training of the drone's algorithm. Then there is the evaluation function, which is expected to comprise the bulk of the development time for the project and controls what the machine learner values in training. Finally, there are hardware considerations and adjustments to ensure that the physical drone matches its virtual counterpart.

For the simulation environment, there are three predominant options that have been considered. The first option is to develop an entirely novel simulation environment that is tailored to the testing of our machine-learning model. The second is to use a simulation environment developed by Simon Levy specifically for the Crazyflie drone [4]. Third is the option which is the one which is currently being most heavily considered, and that is using the simulation environment developed by the prior senior design class group who worked on this project. This third option is what we decided to use because it combines many of the benefits of the prior two options since it is both designed for this specific problem and does not require much more development time.

The evaluation function is the main way the software developed in this project will interact with the drone. It consists of a function that uses the data available to the drone to mathematically determine if the drone is currently flying well. The development of an evaluation function is a complex series of decisions in which the data generated by the model's actions are evaluated to determine how 'well' the model should be considered to be performing. These adjustments allow for the model to assess itself and improve in accordance with the function. Currently, the evaluation function considers four main things, the angle of the drone to check stability, the position of the drone to make sure it is within the desired area, landing checks to make sure it lands without crashing, and a check to make sure it isn't launching again after it has landed.

The hardware considerations and decisions will begin to be made later in the project after some preliminary models have been trained and evaluated, and consist mostly of determining which ways the physical drone needs to be changed in order to align with the simulated version, or if the simulation needs to be altered to match the physical version. Furthermore, there are decisions to be made on which additional components need to be added to the physical drone in order to allow it to have added functionalities, such as reading more data points. As of now hardware functionality is a stretch goal and is considered to be something we may work on in the future.

These three design components combine to create an autonomously operating drone, the simulation allows the model to train, the evaluation function tells the simulation how successful the training is, and the hardware is the application of that simulated model. The end result should be a Crazyflie drone able to hover stably a few feet above the ground.

## DESIGN CONCEPTS, EVALUATION & SELECTION

For the simulator considerations usage of the prior senior design team's simulator is what we decided on. The alternatives were using Simon Levy's simulator or creating a new simulator [5]. There were three main factors to choosing the simulator, that being how well suited it is to the problem, ease of use, and development time. We chose the previous senior design team's simulator as it was developed specifically for this project, suited the project's needs very well, and required little to no development time devoted to tweaking the simulator. Much of the same is true with Simon Levy's simulator, however as it was not designed for this project specifically, it is likely it is not as well suited to the project, and therefore the senior design simulator is preferable. The third option of development of a new simulator was not employed because, while it is possible that a simulator that is more effective and easier to use would be created, such an effort would have likely consumed all the development time and have left the primary objective of training a drone to hover to be uncompleted [5].

The decisions in the evaluation function are in constant flux and change from week to week as the process of creating a machine learning model is very much one of trial and error. Currently, the most successful models we have created account for drone stability, presence within a specified area, and quality of landing. We are looking into adding further considerations to stop undesirable behavior that we are encountering such as the drone repeatedly hopping to get the landing benefit several times.

The decisions associated with hardware will, similarly to the evaluation function, need to be determined by extensive testing, some hardware adjustments will become necessary to accommodate new sensors for the drone to utilize in its data gathering so it can accurately reflect the sensing capabilities of its virtual twin. Other hardware adjustments will need to be made in order to keep it more in line with this virtual twin and make sure it is performing similarly enough to the simulation. Both of these decision points require a somewhat functioning model and a preliminary evaluation function to begin being considered, therefore it is impossible to say with certainty what these decisions look like at this stage, and it is likely it will end up being beyond the scope of the project.

The broadest possible hardware consideration that is being considered is the choice of drone. Currently, the Crazyflie drone is being considered for its small size and already established code base, but should it prove uncooperative other autonomous drone options exist, such as the flapper drone, but the flapper drone has the issue of being much more experimental, as well as expensive, furthermore it has a much more unfamiliar code base, that our researched simulators cannot accommodate. If other drones were implemented it would likely restructure the entirety of the project, and as such should be only considered if usage of the Crazyflie proves to be completely impractical.

## PRODUCT EVALUATION & TEST/EVALUATION PLAN

The project has distinct yet interconnected physical and digital characteristics that need to be developed.

We first familiarize ourselves with the simulation environment created by the previous senior design group, from there the majority of the workflow will be discovering what the best test plan is, and that affects how the prototype makes itself. The simulation already creates a machine learner to control the drone, the main problem we are solving is that our reward structure, which is effectively test evaluation, is not optimally rewarding the learner for stable flight, and it is therefore not flying properly. Deciding what characteristics need to be rewarded and which need to be penalized.

For hardware, we will not be able to complete it in time. Dr. Simon will be here over the summer and working on implementing our network into the physical drone, and we will be helping him after graduation as our future work.

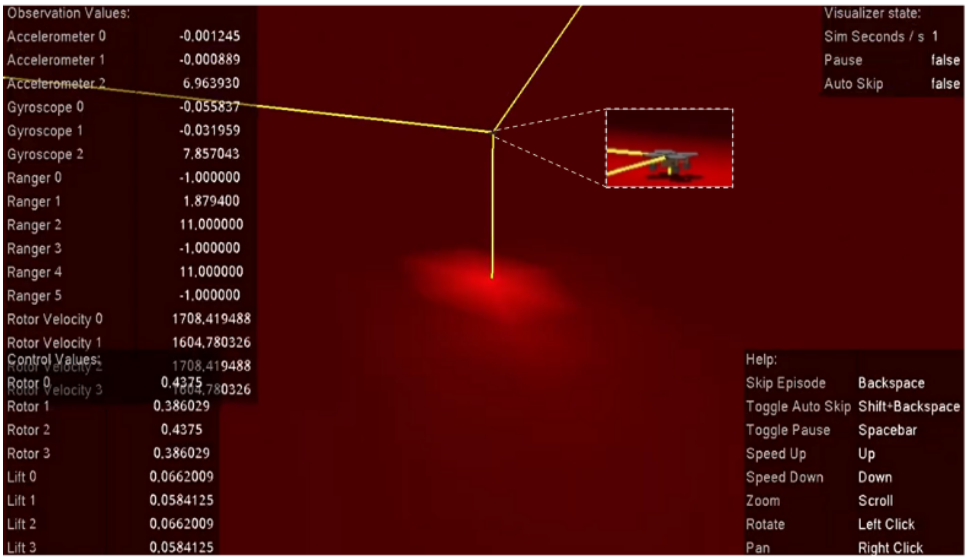


**Figure 5:** Drone Test Simulation

## Social Impact Evaluation

The current scope of our project is research. While we currently do not anticipate full-scale development and distribution of this autonomous drone, we would still like to speculate and identify its social impact. A small form factor, SNN-controlled drones could have a variety of use cases, including sorting, identification, and reconnaissance. The needs for these scenarios, that we would like our drone to solve, might include efficiency, compactness, routing, etc. We plan to solve these needs in the future.

Our project is trying to implement autonomous drone control. With this type of technology, many people will be affected in a variety of ways. This technology could be a catalyst for a multitude of new laws and regulations. In addition, successful deployment of this drone could put pressure on certain job positions, perhaps eliminating certain human-designated roles.

When working with neural networks and autonomous drone operation, there could be many ethical dilemmas that arise. These include privacy and ethical issues that we, as a team, have to address while working on this project. In addition to ethical concerns, our team also has professional responsibilities to which we must adhere to maintain good engineering practices. Examples will be added in the future.

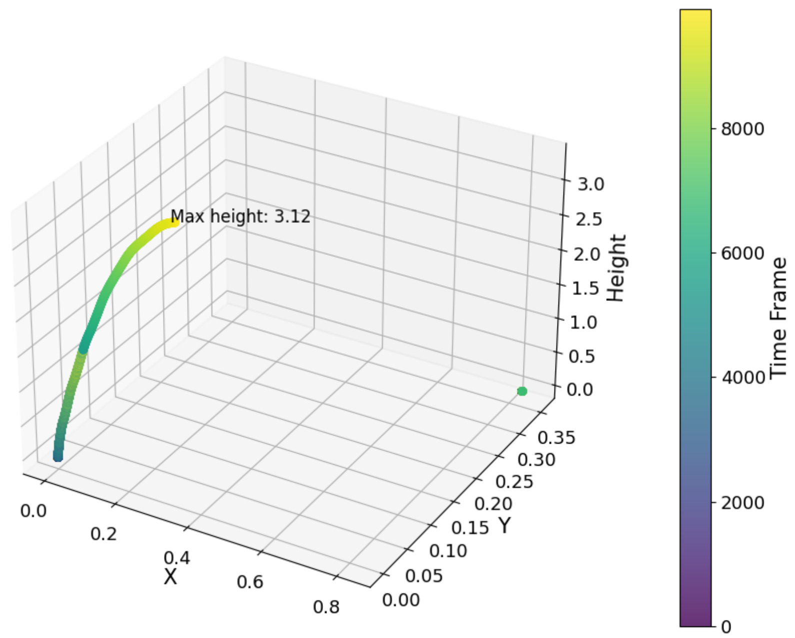


**Figure 6:** Drone Flight Path

Figure 6 shows an example of the flight path of a drone for one of the simulations. Each point represents the location of the drone when the reward was stored. The color bar on the right represents each time step when the reward was recorded. The drone flies, hovers, and then lands without crashing. There is a point that seems to be an outlier with a location of (0.8, 0.0) with a height of zero. We assumed that it might be a noise to our data.

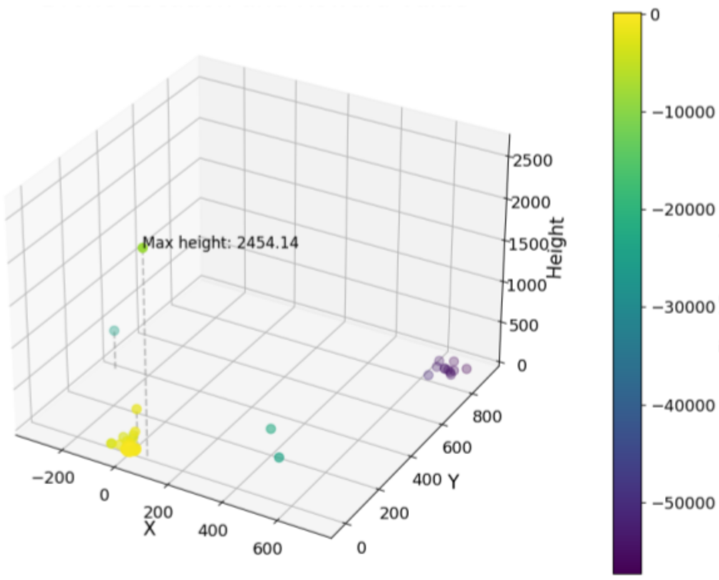


**Figure 7:** Drone Location and Corresponding Final Reward

Figure 7 shows the location of the drone and corresponding final reward values. For this graph, the color bar represents the 'final_reward' value that is being stored after many reward values are being stored. Higher final reward values are located closer to the location of (0, 0) which is the start location of the drone. As the final reward decreases, the location of the drone is clustered towards (600, 800), which is further from the starting point. This is the behavior that we intended because we wanted our drone to stay where it is in terms of (x, y) location. Therefore, if the drone got further from the starting point, it was penalized so that the drone would stay within the origin.

## Deliverables and Milestones (need to be updated)

The primary deliverable of this project is providing the customer with a drone simulator that is controlled autonomously using spiking neural networks. The initial goal of the drone simulator is to achieve stable drone operation in a simulated environment. The stretch goals of the drone are to achieve stable drone operation in a real-world environment and make the drone safely follow a leader drone and once this is working, we want to apply swarming together. The drone simulator delivers the following components:

1. Evolutionary Optimization for Neuromorphic Systems(EONS) for model building and training

2. Simulation environment for training and experimenting with the drone and the network

The stretched goal could deliver the following components:

1. Crazyflie drone
2. Neuromorphic Hardware to run the spiking neural network controlling the drone

Software development was based on the existing network which needs some control refining. In addition, we applied EONS to train networks for an application. To test and develop our software, we used the existing physics simulator which is for takeoff and landing [3]. All the deliverables will be provided to the customer before the senior design showcase, on May 3rd of 2024.

# PROJECT MANAGEMENT

## Roles

Our project, which consists of four people, aimed to have each member choose a role they are confident in, faithfully carry out their duties, and coordinate opinions with each other to create optimal work.

As software developers, all team members understood how the previous network developed and then worked to improve it. We focused on the process of training a Spiking Neural Network using EONS, and the implemented Spiking Neural Network will be tested using a physical simulator [3] [4] [5].

As a team leader, Tyler Nitzsche monitored the overall quality of the project. His experience in conducting projects using the Spiking Neural Network was useful in our project. Based on his experience and knowledge, we could draw a blueprint for building a Spiking Neural Network model. Furthermore, we were able to develop a fine drone flying simulator.

As a project organizer, Jihun Kim planned the overall schedule of the project and rearranged it considering the circumstances of each group member. He adjusted the sprint and schedule according to progress and tried to create a schedule that would achieve optimal efficiency within a limited time.

As hardware developers, Seoyoung An and Jonathan Skeen planned to apply the software we developed to hardware so that drones can utilize it. To achieve this, they could be responsible for designing and designing the drone's hardware and choosing what hardware to use. However, the project was adjusted to focus more on improving the simulator using SNN. Hardware development was changed to a stretch goal, and then it was temporarily postponed before it was ultimately canceled.

This project covers a lot of tasks, and given the different experiences of each group member, we aim to create an environment where each member not only plays his or her role but also supports each other during the work. Each member was a reviewer or tester, finding problems and improvements in each other's work. This work helped prevent bigger problems and create better software.

## Milestones

To ensure deadlines and manage schedules from the Fall 2023 semester to the Spring 2024 semester, we divided the work into five sprints as Table 1 shows. There may be changes in the order of work, additions, or exclusions of tasks in the future.

The Gantt chart in Figure 8 shows the overall schedule of the project in a table.

| WBS NUMBER | TASK TITLE | ASSOCIATED FILE | TASK OWNER | START DATE | DUE DATE | DAYS | TASK PROGRESS |
|---|---|---|---|---|---|---|---|
| **1** | **Project Establishment** | | | | | | |
| 1.1 | Meeting with Dr. Schuman | NOTE | Everyone | 10/19/23 | 10/19/23 | 1 | 100% |
| 1.2.1 | Project Vision Presentation (3-min video) | | *Individual* | 10/20/23 | 10/20/23 | 1 | 100% |
| 1.2.2 | Project Vision Presentation (5-min video) | | Everyone | 10/26/23 | 10/27/23 | 2 | 100% |
| 1.3 | Preliminary Design Presentation (8-min video) | | Everyone | 10/30/23 | 11/03/23 | 4 | 100% |
| 1.4 | Project Design Presentation (12-min video) | PPT | Everyone | 11/29/23 | 11/29/23 | 1 | 100% |
| 1.5 | Project Design Report | REPORT | Everyone | 11/29/23 | 12/06/23 | 8 | 100% |
| **2** | **Software Development** | | | | | | |
| 2.1.1 | TENNLab's Introductory Video | | Everyone | 01/31/24 | 01/31/24 | 1 | 100% |
| 2.1.2 | Connect to the TENNLab's BitBucket | | Everyone | 01/31/24 | 02/02/24 | 3 | 100% |
| 2.1.2.1 | ~~EONS Application (Minesweeper)~~ | | Amy | 02/03/24 | ~~03/22/24~~ | 50 | ~~80%~~ |
| 2.1.3 | Review the network from previous research | Bitbucket | Everyone | 02/08/24 | 03/05/24 | 28 | 100% |
| 2.99.1 | Product Design/Early Prototypes (2-min video) | | Everyone | 02/12/24 | 02/14/24 | 3 | 100% |
| 2.99.2 | Project Report 1 | REPORT | Everyone | 02/12/24 | 02/15/24 | 4 | 100% |
| 2.99.3 | Recording of Project Meeting (how to work better?) | | Everyone | 02/12/24 | 02/16/24 | 5 | 100% |
| **3** | **Network Optimization** | | | | | | |
| 3.1 | Review the previous simulation | | Everyone | 03/06/24 | 03/22/24 | 17 | 100% |
| 3.2 | Improve the Physical Simulator | | *Individual* | 03/06/24 | 04/15/24 | 40 | 100% |
| 3.3 | Create Network Map | | *Individual* | 04/12/24 | 04/15/24 | 4 | 100% |
| 3.98.1 | Product MVP (5-min video) | | Everyone | 03/01/24 | 03/06/24 | 6 | 100% |
| 3.98.2 | Project Report 2 | REPORT | Everyone | 03/01/24 | 03/07/24 | 7 | 100% |
| 3.98.3 | Recording of Project Meeting (feedback) | | Everyone | 03/06/24 | 03/08/24 | 3 | 100% |
| 3.99.1 | Product Pre-Release (7-min video) | | Everyone | 04/03/24 | 04/03/24 | 1 | 100% |
| 3.99.2 | Project Report 3 | REPORT | Everyone | 04/03/24 | 04/04/24 | 2 | 100% |
| 3.99.3 | Project Meeting (Progress Review) | | Everyone | 04/03/24 | 04/05/24 | 3 | 100% |
| **4** | **Complete Drone Simulator** | | | | | | |
| 4.1 | Make a poster | | Everyone | 04/15/24 | 04/17/24 | 3 | 100% |
| 4.99.1 | Final Poster and Product Pitch (1-min video) | | Everyone | 04/15/24 | 04/17/24 | 3 | 100% |
| 4.99.2 | Product Release (7-min video) | | Everyone | 04/17/24 | 04/17/24 | 1 | 100% |
| 4.99.3 | Final Project Report | REPORT | Everyone | 04/17/24 | 05/03/24 | 17 | 50% |
| 4.99.4 | Project Archive | | Everyone | 05/03/24 | 05/03/24 | 1 | 0% |

**Figure 8:** Part of the Gantt Table of the Project

The Gantt chart has been modified to improve accessibility and make schedule management more convenient by modifying the template provided by Google Spreadsheet to automatically fill the table when someone enters the start and due dates. It also has the advantage of being able to easily determine who is working on what task. We expect that by updating the Gantt chart, we will be able to easily check the progress of the project and through this, we will be able to control the progress of the project.

# BUDGET

We didn't have a definitive budget. To order necessary parts or equipment, we would create a list of parts, their prices, and associated links that lead to where the parts can be bought. The list would be confirmed by our advisor, Dr. Schuman. Then, we would send an email to eecs-orders@utk.edu and CC the teaching assistant of the COSC 401 course, including the name of our project, team lead, list of items, price range, and explanation of why we need them. The possible items that we might need to buy were another drone and sensors like gyroscopes for the drone.

As we approached the end of the project, we realigned our objective to prioritize the refinement of our simulator framework over the implementation of the physical drone. Furthermore, acquiring a new drone and getting familiar with it will not be possible within less than two months period. Therefore, we have opted to utilize the existing drone available in Dr. Schuman's lab, which will maximize efficiency through leveraging pre-existing resources. Therefore, we did not use any budget.

# Reference


[1] Bitcraze. Crazyflie 2.1.

[2] Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Prasanna Date, and Bill Kay. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022.

[3] C. D. Schuman, J. P. Mitchell, R. M. Patton, T. E. Potok, and J. S. Plank. Evolutionary optimization for neuromorphic systems. In *NICE: Neuro-Inspired Computational Elements Workshop*, 2020.

[4] Simon D. Levy. Multicoptersim, 2023.

[5] Simon D. Levy. gym-copter, 2023.

# APPENDIX

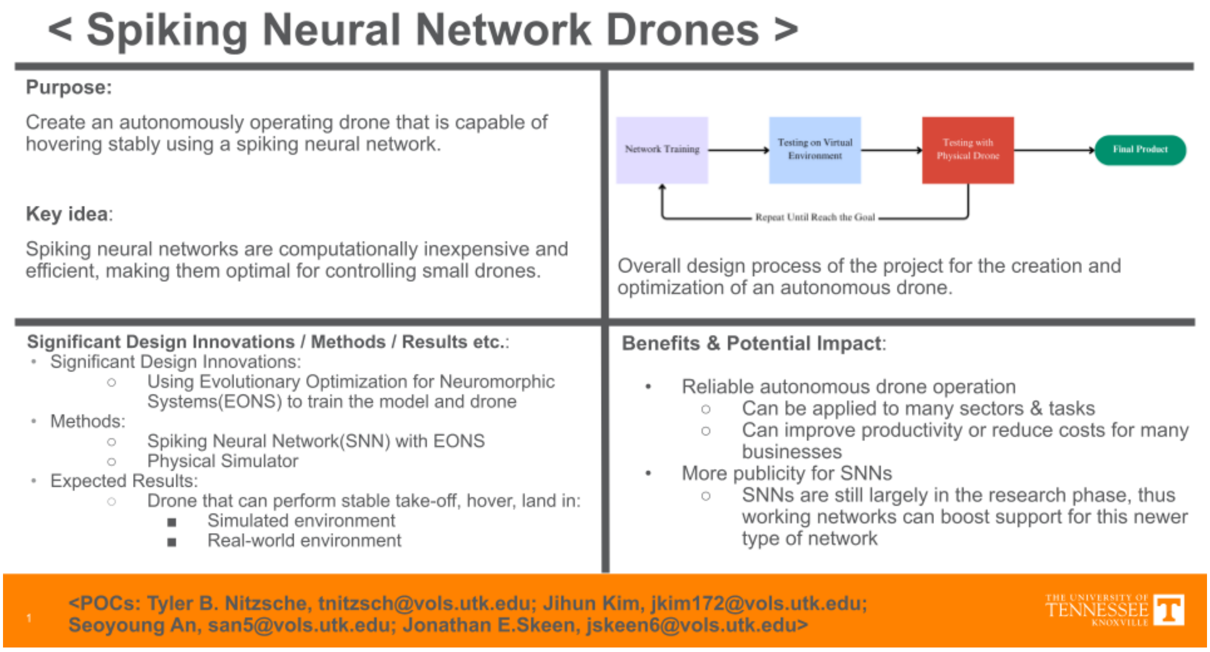


**Figure 9:** Project Quad Chart

**Table 1:** Milestone in Each Sprint

<table>
<tr><td>Sprint 1</td><td>10/02/2023 – 01/22/2024</td></tr>
<tr><td colspan="2">Sprint 1 is the longest, running from the start of the project until the start of Spring 2024.<br>During Sprint 1, group members identify the size of the project, set project goals,<br>and devise detailed goals for this goal.<br>During this period, group members met with our mentor, Dr. Schuman, to hear<br>about the project and gain insight into the project. This information includes<br>software resources, hardware resources, place support, and funding available to us.</td></tr>
<tr><td>Sprint 2</td><td>01/21/2024 – 02/17/2024</td></tr>
<tr><td colspan="2">Sprint 2 is the period from the start of the 2024 spring semester to the filming of the first video.<br>The goal of Sprint 2 is to produce an early prototype. During this period,<br>group members can increase their understanding of TENNLAB's resources, including EONS,<br>through TENNLAB's introductory videos, and acquire basic knowledge to implement<br>the software needed for the project. After this, we will find out the level of the network<br>that has already been implemented, establish directions on how to improve it,<br>and begin software development.</td></tr>
<tr><td>Sprint 3</td><td>02/18/2024 – 04/06/2024</td></tr>
<tr><td colspan="2">Sprint 3 is the period from the first video recording to the third video recording.<br>The goal of Sprint 3 is the complete (or near complete) implementation of the software<br>for hardware. During this period, we will seek to improve the early prototype implemented in<br>Sprint 2 through network optimization and training. In this process, we will utilize a<br>physics simulator. We can either use the simulator designed by the previous senior design<br>course or one that has been developed by Simon Levy.</td></tr>
<tr><td>Sprint 4</td><td>04/06/2024 – 05/03/2024</td></tr>
<tr><td colspan="2">Sprint 4 is the period from the third video recording to the end of the project.<br>The goal of Sprint 4 is to prepare for the presentation.<br>Sprint 4 includes making a poster and printing it.</td></tr>
</table>