# “Envoy Commander” — Implementing Model-Free, Collaborative, and Centralized Reinforcement Learning

Andrew Cuevas, Yash Gharat, Anthony Soffian

University of Central Florida, Department of Electrical and Computer Engineering, Orlando, Florida, 32816

***Abstract* — This paper presents the construction and design of a framework for testing and developing a centralized, collaborative, model-free reinforcement simulation. Although similar algorithms exist and the general trend is towards distributed or federated learning, centralized learning can often be cheaper since the field, or “dummy”, agents can be affordable and standardized when controlled by a stronger “commander” machine learning agent that controls the dummy agents when trained or reinforced in some way. This project built this framework and uncovered the significance of sensor noise and slipperiness, even in a controlled environment. Additionally, the framework could potentially be used with many other algorithms as well.**

***Index Terms* — Q-learning, Reinforcement Learning, Buck Converters, Shift Registers, Bluetooth,, Sockets, Servomotors**

## I. Introduction

Many textbooks have explored simple reinforcement learning problems to introduce fundamental principles. The multi-armed bandit problem is a classic example of this. In this problem there is a learning agent with multiple arms situated in front of some number of slot machines. Each slot machine is given its own distribution of success that is unknown to the learning agent. When each lever is pulled on a slot machine, the learning agent is given either a reward of 0 or +1 for failure or success, respectively. The learning agent wants to maximize the rewards it receives from the machines, without knowing exactly when each of the machines will result in a success. There are many solutions to the multi-armed bandit problem, but many of them are just virtual simulations. In the Envoy Commander, the goal is to investigate a more tangible version of this problem to incorporate real-world concepts into machine learning.

### *A. Context*

As the world of machine learning and advanced AI grows, there is a general trend towards cooperative, web-like, systems of AI. Additionally, there is a need for more affordable solutions, as traditional distributed AI can be expensive because those agents need to have equal processing capabilities. This could range from powerful hardware or just better sensors. It is logical to just throw in exceptional AI that can not only support themselves, but also intelligently cooperate when completing the task.

There are a few ways to go about this. As mentioned previously, distributed machine learning models help with the usual scalability and efficiency issues. Instead of centralizing the collection and processing of data, it can be distributed across many agents to take care of computation issues in a divide-and-conquer approach. Recently there has been a concern for data privacy issues so a new algorithm was devised known as federated machine learning. Federated machine learning only stores the model parameters in nodes on cloud servers and uses user devices for training. As a result, these devices such as mobile phones serve the user without risking their privacy. A common example is Google’s GBoard, the mobile keyboard whose suggestions and swipe inference are cached locally.

Although distributed learning is a solution for the need of scalability, it requires powerful machines to train a large dataset. On the other hand, federated learning is good when there are privacy concerns and when the model is being served on a device that can handle its heavy load. Typical modern mobile phones come with machine learning capable cores so this may not be an issue, Because this project is not not concerned with privacy a different approach is required.

The goal of this research is to solve the problem of scalability in machine learning systems while also keeping the solution affordable. The proposed solution utilizes a centralized machine learning approach. In this algorithm, the scalability problem works differently for a few reasons. Instead of having a strict, real-time algorithm, a type of reinforcement learning known as Q-Learning is implemented. In this case, the model will be pre-trained based on incentivization and exploration. The benefit of this method is that the agents performing the task and cooperating need very little computing power. These agents only need to be able to communicate effectively with a centralized learning agent, known in this context as the Envoy Commander. An added benefit is that this method is scalable by adding “dummy agents” to the communication network if necessary. To implement the approach, dummy agents are placed in a controlled environment and use localized sensors and external stimuli. Sensor measurements communicated to a

centralized Q-learning agent which will return an optimized action.

### *B. Motivation*

Dr. Chinwendu Enyioha conducts basic and applied research in the areas of distributed optimization and control of networked dynamic and cyber-physical systems. Dr. Enyioha's research interests also include communication-efficient optimization in distributed decision making. As the project's advisor, Dr. Enyioha suggested this path for research and presented the problem to be solved.

### *C. Problem Statement*

This project aims to create a centralized, collaborative, model-free implementation of reinforcement learning. There are few similar implementations readily accessible for study although real-world tasks are cooperative. There are even fewer implementations that account for noisy data and slipperiness of the surrounding environment. By using a simulated, controlled environment and multiple components, the weight of these factors can be observed. The Envoy Commander encapsulates three major components as illustrated by Fig. 1. The main component is the learning agent, or Commander, which transmits optimized, incentivized actions to the dummy agents, akin to the arms in the multi-armed bandit problem. These Dummy agents only perform two functions. They transmit sensor data to the Commander and execute the action echoed back to them, in this case pressing one of four buttons. Both of these components interact with the Environment that controls the game's flow and transmits appropriate rewards to the agent as buttons are pressed.

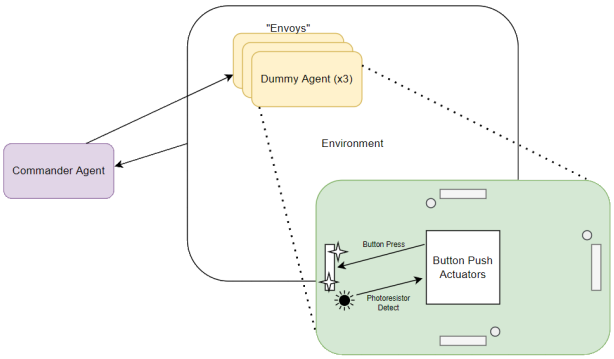


Fig 1. General overview of the Envoy Commander

In this project, it was assumed that all sensor noise was uniform and the sensors were identical such that noise was a side effect of the project but not directly accounted for. Another assumption made was that each observation was single and discrete. In more practical problems, multiple sensors would give multiple input observations that would rarely be perfectly discretized. Finally, it was assumed that the Commander and its agents would have unlimited resources. For example if the batteries were to die, they would simply be replaced, that episode discarded and learning continued.

## II. Envoy Commander

The Envoy Commander is the centralized learning agent that uses Q-learning to make optimized decisions. The Commander must use its supervised envoys to explore the environment and adapt with Q-Learning. The Envoy Commander, lacking direct input from the environment, can remain location agnostic by using the standardized dummy agents.

In order to discover and achieve the goal of its task, the Commander initially only knows its possible states and actions. In order to explore its environment, decisions are initially made at random and conveyed to its envoys. With the feedback received from the environment, the Commander explores less, and instead exploits its knowledge of the quality of each state-action pair. The reward function, detailed in the environment specifications, must be optimized in order to show that learning is progressing at a faster than random rate.

For reward optimization, the Commander agent draws connections between inputs it receives from both the environment and dummy agents, which will both be described in detail soon-after. Although the physical connection and interaction is tangible from an observer perspective, the Commander agent is only exposed to rewards from the scenario as a response to the actions it commands the dummy agents to perform. This repetitive cause and effect relationship learns using a concept referred to as Q-learning. While a form of machine learning, Q-learning is present in absence of a predetermined model, and seeks to assign optimized values to different state-action pairs based on the current information present.

### *A. Q-Learning*

Reinforcement Learning (RL) is a type of machine learning that uses incentivization to reinforce an expected behavior using a learning agent's observations of the current state of its environment. In general, many value or policy-based learning algorithms use state-action pairs, *(S, A)* to represent that in every state, an action is taken to transition to a new state *S'* to produce new observations. Policy-based RL optimizes the action that will result in the optimal state after the transition with maximal rewards. It uses a policy, $\pi$, which is the probability of taking *A* to transition from *S* to *S'*. this policy is updated as the environment is explored. However, Q-Learning is a

value-based algorithm that instead picks *A* independently of the agent's prior actions/policy. Since it is model-free, the agent maintains no knowledge of its environment. It only maintains a record of the reward received when taking an action from a state. Thus, the reward function should appropriately incentivize the true goal.

Q-Learning maintains a table with dimensions SxA to maintain the value each action has in a respective state. This value, or "quality", updates throughout the learning process using the Bellman Equation (1).

$$V(s) = max_a(R(s,a) + \gamma V(S')) \quad (1)$$

This equation states that given the reward, *R(s,a)*, a discount factor, γ, and the value of the next state, *s'*, the Q-table with the maximum value we can receive. γ represents the agent's focus on immediate vs long term rewards and its usage depends on the problem. In this project, there is a short-term goal of correctly pressing the lit up button but there is also a more incentivized, long term goal of collaboration between the dummy agents. Equation (1) needs to have optimized parameters to fit this goal, paired with an appropriate reward function. The overall flow of Q-Learning is shown in Fig. 2. The overall flow of Q-Learning is shown in Fig. 2.

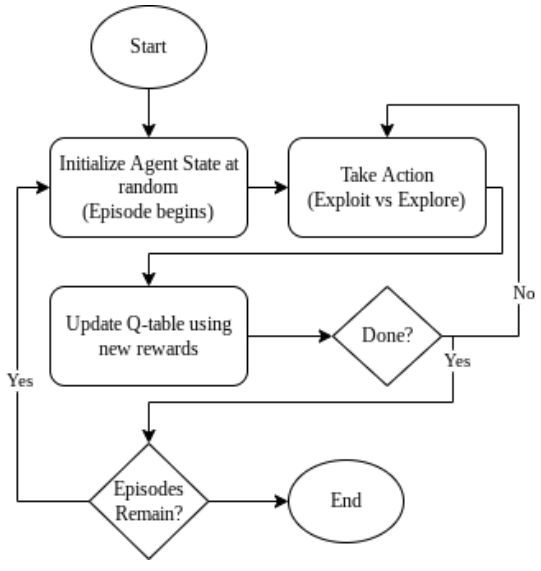


Fig 2. Q-Learning flowchart

The project stays true to the algorithm but adds an extra dimension to the states. A state in the context of this project is the current dummy agent and the button it detects. Thus, assuming agents, *D,* and buttons, *B*, the Q-table has dimensions *D*x*B*x*A* and the learning agent maintains this to facilitate the collaborative learning aspect of the project.

### *B. Communication*

The Commander uses a bluetooth server instead of client-client communication to better facilitate the flow of the game. The main difference is that the Raspberry Pi (Pi) acts as a slave and the HC-05 bluetooth modules on the agents and arena are masters. They initiate a connection with the Pi, who accepts any incoming connection and creates a socket for that connection. There are many advantages to this method of connection. The main one is scalability. Instead of socket amount being based on the max number of agents available, this dynamically allocates sockets as connections are needed and dispose of them when connection is lost. The method saves communication bandwidth and computation by iterating through only the necessary sockets. Another important advantage is overall game synchronization. With dynamic sockets in a bluetooth server, there is no concern for the game being played out of order due to sensor noise. In the scenario where an agent incorrectly detects a button due to sensor noise, that agent will send a message to the commander. The programming will have the agent press the incorrect button but a reward of zero will be provided and the game will not be affected. So it will not overly impact the learning process unless there is a systematic error affecting all runs. This will have to be resolved by one of the researchers. The bluetooth server is also device agnostic such that the clients do not need to be identical. As long as they are able to enter master mode and bind to the Pi address they will connect. So for future upgrades of dummy agents, the legacy ones will not have to be updated at least due to communication protocol.

## III. Dummy Agent

The dummy agent is a standardized component that represents the "arms" from the traditional multi-armed bandit problem. Its purpose is to transmit its observations and execute commands received from its Commander. The Dummy Agents also demonstrate slipperiness in its interaction with the environment. On occasion, systematic and random errors can occur during the learning process that result in incorrect behaviors. While the systematic errors such as the sensor getting disconnected or batteries dying may be significant, the Commander should be agnostic to the random errors. In this project, the observations from the dummy agents are considered ground-truth to the Commander and the focus is not evaluating the uncertainty in them. Instead the observations cause either exploration or exploitation in the environment.

### *B. Scalability and Standardization*

A primary focus for the implementation of the Dummy Agents is the idea that the design must be expandable to a model with a similar game but many more agents working in tandem. The purpose for this design principle is that traditional machine learning models can distribute work or training across different machines or processors, and collaborative solutions are a necessity for the future of problem solving in all forms of machine learning. In the interest of scalability, this focus reduces the need to design multiple different machines which would imply extra cost in materials, and unrealistic time requirement scaling, as each agent would require meticulous planning in terms of function and execution.

This principle enables the idea that problem solving machines can be mass produced efficiently and designed to work seamlessly together, which is also applicable in the context of this project as the cascading design solution provides more avenues for the environment to provide more reward for specific actions down the line. This in turn enables more efficient and faster learning being performed by the Commander-Dummy system being monitored. Additional design considerations not included in this project also include the idea that scalability can also exist if modules of different machines are constructed in groups to maintain the scalable ideal of cooperative learning. Although this design was considered, the implied costs and display difficulties were far too great to keep the project affordable, and thus a simpler approach to scalable design was selected. These principles of scalability, affordability, and expandability were the primary focus in all decision making processes for design.

### *C. Mechanical Design Choices*

The mechanical design for the dummy agent stemmed from the basic principles surrounding this project, scalability and standardization. All of the dummy agents are exact clones of each other and perform the same functions. The mechanical task they were to accomplish was as simple since they needed only to push buttons. This resulted in starting with linear actuators as the base design that converted rotational motion of servos to linear motion using a rack and pinion design. Since only about 90 degrees of motion were needed to press a button, one servo was able to press two buttons with crossed racks. Some basic supports were created to apply normal force and position the racks. Additionally, the racks were tested with a few different lengths to ensure proper length for optimal button-pressing on both sides. Adequate space was also needed to seat the PCB so this was given at the base. Finally, laser-cut MDF wood was used for construction for optimal cost and usability.

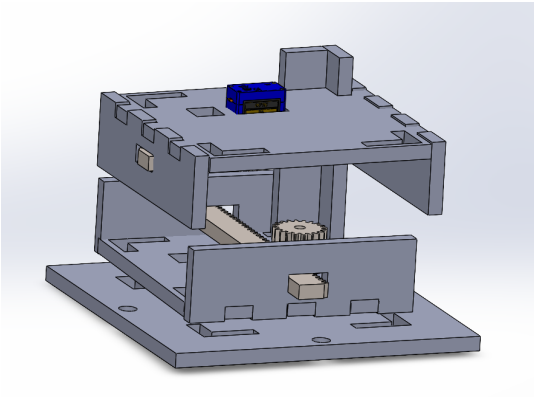

Fig 3. Dummy Agent deconstructed

### *D. Hardware Design Choices*

Choosing the components to fit the physical design constraints led to the PCB being constructed using SMT, which minimized the Dummy Agents to a size appropriate to the Requirement Specification of less than $3ft^2$. These dimensions were finalized to a 71mm x 73mm x 10mm design that fit within the 82mm x 82mm x 25mm space allotted. The components mounted to this board included the ATMEGA168PA-PU, HC-05 Bluetooth Module, Switching/Linear Voltage Regulators, and utilized header pins to connect the board to the two SG90 9g Servos and four GL5549 photoresistors. Along with a 9.6V voltage source, this would provide a supply current of 1.5A to power all of the components at a regulated 5V.

The microcontroller to run this section of the design was chosen to be the ATMEGA168PA-PU due to its compatibility with the Arduino IDE that we utilized throughout the testing process to affirm the design choices, and to run this code a bootloader was burned onto this IC before the Dummy Agent was ready for service. This specific model was used due to the limited availability of the ATMEGA328 in SMT form from a reputable supplier in a reasonable timetable, as this MCU was on the Arduino Nano and Uno's that verified the feasibility of the project in its initial stages.

Since the pins on MCU this had less significance in comparison to the Environment PCB, it allowed the focus of the design on the implementation of these key components, such as the twelve GL5549 photoresistors, four for each agent, that gave the necessary data extracted from the Environment to send to the Envoy Commander. It was known that these photoresistors lacked a quality sensitivity rating, as well as environmentally unfriendly, so initially photodiodes were used as the sensing IC, but after some testing comparable data between them, the photodiodes could not recover accurate light levels from the Environment. This is likely due to either improper design of the sensing schematic, possibly needing an

envelope detector system, or the photodiodes themselves were of cheap quality. Regardless, these photoresistor were able to read the environment to a higher degree despite the outside light interference, and allow the agent to send the readings to our Envoy Commander to interpret which buttons to press using the two motors residing on the chassis above the Dummy Agent's location.

These motors, two SG90 9g's, are non-continuous micro servos that further maximizes the space we have within the labeled constraints, pulling approximately 750mA during the stall period. This current draw was the basis of the voltage regulator decision process as the minimum of 1.5A was tasked based on this start up draw alone. Using in conjunction with the gear and actuator design, this system was able to properly apply enough force to register a button press in reaction to the photoresistor data collection. To control this motor from the Envoy Commander module based on this data, an HC-05 Bluetooth module was chosen for its simplicity and familiarity in its master form as this IC was used throughout the prototyping phase to communicate data between the different Dummy Agents and their Envoy Commander. Theoretically, within the limits of the Envoy Commander's current iteration, a limit of six Dummy Agents can be constructed that would be able to communicate using their HC-05 with the Envoy Commander's six available sockets to speak with. But for this current set of components, the necessary power for this PCB was established and the task of powering this system then began to be worked on.

To power this system, eight AA batteries were used in series that gave an estimate of 2100-2400mAH of life based on testing done, and with them being rechargeable their voltage ranged from 1.2V to 1.4V, making the final output at reasonable conditions 9.6V, with 11V only existing for a short amount of time after initial recharging. This voltage was chosen to minimize the size and cost of the capacitors and inductors on the Dummy Agent as using a potential boost, or step up, converter required components either unavailable or too large to fit within the design constraints. The Voltage Regulators came to be a buck, or step-down, converter and a LDO Voltage Regulator to compensate for the ripple effect of the switching regulator. This tandem formed a system that was as efficient as possible to remove any noise as well as potential heat issues, through bucking the 9.6V to 5.33V with the LM2576D2T-ADJ4G, and then dropping that voltage again with the MCP1827T-5002E/ET LDO, as this had a typical dropout voltage of 330mV, to a clean 5V with a current output of 1.5A to supply the PCB.

Bringing all these components together, this SMT board was designed with the EasyEDA software using the technical assistance of David Jones's PCB guide [1], and then fabricated by PCBWay and JLCPCB with the appropriate stencil provided. This was assembled by hand and soldered using the reflow oven in UCF's Integrated Design Studio Lab under the supervision of Dr. Kundu and Lab Specialist Annabelinda Zhou, along with contingency PCBs constructed with the hot air gun in the Senior Design Lab.

## IV. Environment

The environment in the context of this project is present to bridge the gap between conceptualized Q-learning and the physical game that is being modeled. The environment is concerned with responding to the actions performed within the scenario and returning information to the Commander for reward optimization. The construction and inconsistencies within the environment are meant to manifest a concept referred to as slipperiness in the scope of Q-learning and decision making. As a result, this maintenance of disconnection between dummy agents and the environment they are placed in enables the model to maintain scalability and relevance to realistic models where certainty is never promised.

In order to bridge the gap between reality and abstracted Q-learning, the environment is the sole communicator of reward for the Commander agent. By separating the decisions being made, and returning feedback to the Commander, the decision making process is divided into actions taken against the result of these actions, which better realizes the cause and effect relationship between the Dummy Agent's actions and learning that is performed. Without the predetermined rewards that are applied depending on the different kinds of actions taken by the constituents of the participants, there is no game, goal, or model. By setting this scenario beforehand and not allowing prior knowledge to affect decisions made by the Commander agent, the discretized actions and values that are performed and received as reward are the manifestations of previously abstracted q-learning. Additionally, this presents the concept of intentional slipperiness.

Intentional slipperiness is important due to the reflection of this concept in real scenarios that require a similar style of learning. By separating and exposing the algorithm chosen within this project to real uncertainty brought about by real sensors, the idealized algorithm breaks down to a point to simulate the disparity between idealized performance and realistic performance. Necessarily, this implies that connections can be derived from the performance of the model in an ideal setting with simulated slipperiness, and the happenings of false trials and actions that take place in the constructed model within this project. These construction decisions thus

provide a way to extend the current scenario to a larger scenario with more participants, and maintain the scalable nature of the Q-learning ideal.

As the model is scalable and attempts to apply real work concepts to the model, the game selected is a modified version of the game Simon, where a sequence of buttons must be pressed in order of being shown, and increases in length depending on how long the game has proceeded. The core characteristics taken and applied to this game are the use of multiple buttons and a relatively cooperative model where different dummies are presented the same task, and are expected to press the buttons that are lit up. By restricting the buttons to light up relationally to each other, staying consistent across rooms, the environment is able to emphasize reward for responding to the correct light, and pressing the same buttons as the previous room. As the framework for the environment consists of software controlled buttons and rewards, it is also then possible to emphasize a different characteristic the Commander is expected to learn depending on the response to action ratio of rewards given.

However, For the purposes of this project the game "mode" selected and displayed are detailed in the next section. A simpler game with clear reward was selected for a simpler and easier to understand demonstration.

*B. Game Explanation*

The environment in RL consists of what the agent interacts with and its states. In our case, the game played by the Commander and its Dummy Agents is the environment. This game consists of a series of "rooms" in which four buttons are placed at opposite sides of each room. Each button can light up and is controlled by a central processor that overlooks and manages the game. At the start of every episode, a random button is chosen to be used at every time step. Once the agent detects and performs an action, the environment communicates the appropriate rewards to the Commander and lights up the same button in the next room. This process repeats as the game progresses and episodes continue.

There are two main learning objectives to this game. First, the Commander must recognize, as a learning agent, that he should press the lit up button to get a single reward. Secondly, the Commander should learn that he must use his "arms", the dummy agents, to collaborate over time. To accomplish both of these, the reward function hands out a reward of 0 when the incorrect button is pressed but a reward of 1 for the correct button. As the sequence continues and the agent is correct, this reward continues to stack exponentially with some multiplier. The multiplier is reset when the agent presses an incorrect button in the episode.

This project effectively demonstrates slipperiness in an environment. Slipperiness is when an unintended action is taken due to the environment. In this project's context, the agent may wish to press button 1 but because of a random error, it incorrectly actuates and holds down button 1. Thus in the next state, it will receive the incorrect reward. If this is not remedied in time, the episode will have to be scrapped. Slipperiness often happens in real environments due to sensor noise, communication faults, and other random or systematic errors.

*C. Mechanical Design Choices*

The Environment consists of multiple rooms that are connected to a central processor with a scalable shift register design. The design choices centered on an easily assembled room design with access to the photoresistors and any wires that may need to be external, most notably the power source for the dummy agents. Due to their design, they can be placed in any orientation as long as the dummy agent is placed to align the proper buttons to the actuation points.

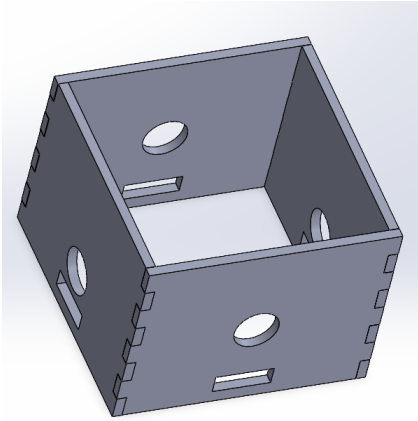

Fig 4. Arena Room

*D. Hardware Design Choices*

To develop the environment of the Dummy Agents, this box design was put in place initially to decrease outside interference of light from the LEDs being activated from the servo system, but after the photoresistors deemed this effect small enough to still calculate a difference between the different LEDs being activated the box continued to represent the surroundings of the Dummy Agent to hold the LEDs, and be controlled by the Environment Agent which would receive data from the Commander Agent on which LEDs to activate, and send the button push data from the Dummy Agent back. To satisfy this process the following components were selected as appropriate.

The first major design choice was to make this board a throughhole, or TH, as it was not constrained by part size or availability in this instance, as most of the components

for this agent were readily available. Because of this choice, the parts were able to be soldered by hand due to the components being larger in nature. Additionally, the fact that only one agent was required for operation led to this choice, as if this project required more Environment Agents, the costs for making multiple of these throughole boards would not be within our budget constraints, as SMT components and fabrication are significantly cheaper to order and build.

Using the same ATMEGA168A-PU as the Dummy Agent for the same reasoning of the Arduino IDE system, and Bluetooth module as the HC-05 to send this data interchangeably, the major obstacle faced when designing this was accounting for the limited amount of pins for all the LEDs to be activated from, as after accounting for the USB bootloading system, four buttons from the LEDs, and the HC-05's TX/RX pins, there were only three remaining digital pins for potentially eight to twenty-four LEDs, minimum of two rooms and maximum of six. A solution soon arose in the form of shift registers to activate the system based on a clock and address system that would activate the LEDs one by one.

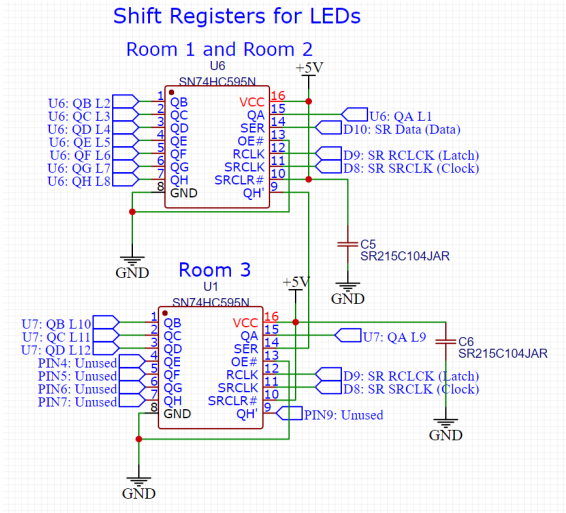


Fig 5. Shift Registers Daisy-Chained: Serial-In/Parallel-Out

These shift registers, the SN74HC595N, were daisy-chained to one another to form the basis of the two room minimum design, as seen in Fig 3, and would require three of these shift registers to reach maximum capacity of this current software design using these HC-05's. With this additional component, as well as developing the software necessary to manipulate, store, and flush the data of these registers this led to the conclusion that only three pins were required, the exact amount that this 168A could support, and even if a MCU with a higher amount of I/O pins were chosen, this system is ideal for future upscaling as the end goal would be have a significant amount of Dummy Agents to run the python software as efficiently as possible. But, in this current design these shift registers would be able to control up to 16 LEDs, or 4 rooms, individually assigning them to light up based on any given instruction from the Commander Agent. Additionally, the buttons on this LED, an Adafruit LED that included the button and resistors, were tied in parallel to one another for each room, with pressing LED 1 in Room 1 would trigger the same response in the following rooms, as minimizing this pin system was a great deal simpler, as the only adjustments made were to properly debounce the buttons, done on the software end of detection.

The remaining components regarded power, as the same logic was used here to use a 9.6V battery source consisting of eight AA batteries, to provide a current supply of 500mA to the system as these LEDs only required 10mA each, and the remaining components additionally required little draw to properly function. The voltage regulators chosen was the MC34063ABN DC-DC buck-boost switching regulator, used as a buck in this system, and the same MCP1827-5002E/AT LDO regulator, which in tandem used the same logic to create a output voltage of 5V at 500mA with a very limited ripple voltage.

This Throughhole board, also designed with EasyEDA and aid from various PCB manuals [1], was ordered through PCBWay, and then assembled and soldered with a few minor modifications. Later, this was placed at the center of the Environment system to allow for the smallest amount of wire required to send data between the Environment box of LEDs and the Environment PCB.

## V. Conclusion

The best takeaway from this project was that textbook problems rarely work out nicely in the real world. Environment noise and slipperiness are bigger factors in learning than originally thought.

### *A. Results*

As far as the development for the framework to demonstrate centralized, collaborative, model-free learning goes, the project was successful. All of the requirement specifications were met and the project is modular enough for anyone to be able to build and expand. This makes the project scalable since the LEDs can be daisy-chained with shift registers and rooms can be added, given some minor modifications to the PCB design. Also with the use of wood and laser cutting, the project was kept cheap and affordable, the largest chunk of the budget is attributed to the PCB shipping and general parts shortage. To reiterate, the project, as a

framework, worked so it can be further expanded on by other researchers.

That being said, in the final stages of the project, many issues arose with the actual connectivity of the bluetooth modules and the Raspberry Pi. This resulted in the machine learning algorithm struggling since it would only last for a few episodes at a time. This can be attributed heavily to the outdated pyBluez library and lack of documentation regarding the socket library's connection. Due to this, sufficient learning wasn't demonstrated. Although all the hardware was working as expected.

However, the concept was previously tested in simulation with noise and reward system included. It was tested with up to 15 agents and with the same reward function. In this case, the algorithm worked and significant learning was shown within 30 episodes. After 100 episodes, it was performing very well.

### *B. Problems Faced*

The problems in this project, as anticipated, were not related to the agent training and software but rather the hardware. The biggest roadblock was the unreliability of the bluetooth connections. Since the HC-05 modules were the masters, the only option was to wait for them to request a connection from the slave Pi. This caused issues because they would often connect slowly, causing the game to progress before all components were connected. There was no real way to understand why the connection was slow or sometimes just unavailable.

Another issue that set this project back was a broken laser cutter that was the primary source for building materials. This is important to note because many alternative options had to come out of it. The first was to 3D print the entire dummy agent. Unfortunately, this was too expensive and time consuming as one dummy agent would have taken 24 hours at least. The next option was to build the agents and rooms by hand with normal wood tools. The rack and pinion would still have to be 3D printed since the fine teeth were not feasible to hand craft. However due to the precise measurements of the holes and alignment, this option also was not practical. The most affordable and sensible option was laser cutting which fortunately started working after a 3 week setback.

### *B. Next Steps*

This project could be expanded in many directions on the hardware and software side. Implementing Bluetooth Low Energy (BLE) would be the logical first step since this would take less power and be most up-to-date for modern applications. It is slightly more expensive but given advanced notice, it could be found at an affordable price point. Another direction to take would be to create a game with a more complicated ruleset that could explore other aspects of collaborative model-free learning such as the decaying learning rate or parameter grid search, since the boiler-plate hardware has already been built. Additionally, it would be beneficial to explore other designs for the dummy agent. Perhaps non standardized agents, varying sensor fidelity, or even multiple sensors could all be good points of study for this project in the future. When adding more factors to this project, another topic of discussion could also be the uncertainty and error associated with each observation to build a sort of trust model between the Commander and his field agents.

## Acknowledgement

The authors wish to acknowledge the guidance of Dr. Chinwendu Enyioha and Dr. Samuel Richie. They also wish to acknowledge the assistance of Dr. Avra Kundu and Annabelinda Zhou with the reflow oven.

## The Engineers

Yash Gharat is a senior at the University of Central Florida and will receive his Bachelor's of Science in Computer Engineering in May of 2022. He plans to continue his education with a Masters in Computer Science while working at CAE Inc. His primary interests are in software engineering and full-stack development.

Andrew Cuevas is a senior at the University of Central Florida and is to receive his Bachelor's Degree in Computer Engineering in the May of 2022. He has freelance software development experience and intends to look for a more permanent position in the near future, but is open to work in any areas of his interests including creative fields such as dance, music, and game development.

Anthony Soffian is a senior at the University of Central Florida, and will receive their Electrical Engineering Bachelors Degree in May of 2022. Currently, they plan to find an entry level position in the greater Orlando or Jacksonville area to further expand their experiences as well as their interests in athletics, medical, and education inclined topics on EE or CS pathways.

## References

[1] Jones, David L. "PCB Design Tutorial - Alternatezone.com." PCB Design & Layout Tutorial, 2004, [Online] Available:http://alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf