Prototyping and Testing Machine Learning in an Application

DESIGN DOCUMENT

sdmay22-45
Adviser: Dr. Rover
Team Members/Roles:
Amy Wieland - Project Manager
Tyler Ingebrand - Project Manager and Machine Learning Manager
Nathan Bruck - External Hardware/Arduino Manager
Yi Ting Liew - Task Board Manager
Sean McFadden - Machine Learning Manager
Nayra Lujano - Research Manager
Chris Hazelton - Security Manager
Team Email: sdmay22-45@iastate.edu
Team Website: https://sdmay22-45.sd.ece.iastate.edu

Revised: 12/9/21 Version 2

Executive Summary

Development Standards & Practices Used

- IEEE P2940 - Standard for measuring robot agility
- IEEE P1872.2 - Standard for autonomous robot ontology
- IEEE 1725-2021 - Standard for rechargeable batteries
- IEEE 802.11 - Standard for wireless LANs
- I2C Bus Protocol

Summary of Requirements

First semester tasks and deliverables:

- Complete the Coursera course, Introduction to Embedded Machine Learning
- Attend one or more keynotes/workshops at Imagine 2021
- Explore and document useful resources about embedded ML
- Define the problem and requirements for a selected application
- Select a microcontroller platform; demonstrate progress with embedded ML

Second semester tasks and deliverables:

- Demonstrate embedded ML functionality on the microcontroller platform
- Prototype and test the application on the microcontroller platform, including use of embedded ML
- Recommend options for course/curriculum integration of embedded ML.

Applicable Courses from Iowa State University Curriculum

- CPRE 288 - Embedded Systems 1
- EE 333 - Electronic Systems Design
- EE 285 - Problem Solving Methods and Tools for Electrical Engineering
- ENGL 314 - Technical Communication
- MATH 207 - Matrices and Linear Algebra
- COM S 309 - Software Development Practices
- COM S 327 - Advanced Programming Techniques
- SE 339 - Software Architecture and Design

New Skills/Knowledge acquired that was not taught in courses

- I2C Bus Protocol
- Machine Learning (reinforcement learning, neural networks, actor critic method)

Table of Contents

Team **5**

1.1 Team Members 5

1.2 Required Skill Sets for Your Project 5

1.3 Skill Sets covered by the Team 5

1.4 Project Management Style Adopted by the team 5

1.5 Initial Project Management Roles 5

Introduction **6**

2.1 Problem Statement 6

2.2 Requirements & Constraints 6

2.3 Engineering Standards 7

2.4 Intended Users and Uses 7

Project Plan **8**

3.1 Project Management/Tracking Procedures 8

3.2 Task Decomposition 8

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria 8

3.4 Project Timeline/Schedule 9

3.5 Risks And Risk Management/Mitigation 9

3.6 Personnel Effort Requirements 11

3.7 Other Resource Requirements 12

4 Design **13**

4.1 Design Context 13

4.1.1 Broader Context 13

4.1.2 User Needs 15

4.1.3 Prior Work/Solutions 15

4.1.4 Technical Complexity 16

4.2 Design Exploration 16

4.2.1 Design Decisions 16

4.2.2 Ideation 16
4.2.3 Decision-Making and Trade-Off 16
4.3 Proposed Design 17
4.3.1 Design Visual and Description 17
4.3.2 Functionality 18
4.3.3 Areas of Concern and Development 18
4.4 Technology Considerations 19
4.5 Design Analysis 19
4.6 Design Plan 19
5 Testing 20
5.1 Unit Testing 20
5.2 Interface Testing 20
5.3 Integration Testing 20
5.4 System Testing 21
5.5 Regression Testing 21
5.6 Acceptance Testing 21
5.7 Results 21
6 Implementation 21
7 Professionalism 22
7.1 Areas of Responsibility 22
7.2 Project Specific Professional Responsibility Areas 23
7.3 Most Applicable Professional Responsibility Area 24
8 Closing Material 25
8.1 Discussion 25
8.2 Conclusion 25
8.3 References 25
8.4 Appendices 26
8.4.1 Team Contract 27

List of figures, tables, symbols and definitions

Figures

Figure 1. Project Timeline Gantt Chart 10

Figure 2. Hardware Platforms 17

Figure 3. Software Platforms 18

Figure 4. Conceptual Design Diagram 19

Figure 5. Component Diagram 22

Figure 6. Actor Critic Method 25

Figure 7. Petoi Bittle Use Manual 30

Tables

Table 1. Project Tasks and Risks 12

Table 2. Personnel Effort Requirements 14

Table 3. Potential Society Areas of Impact 15

Table 4. Professionalism - Areas of Responsibility - IEEE vs NSPE canon 23

1 Team

1.1 Team Members

Amy Wieland

Tyler Ingebrand

Nathan Bruck

Yi Ting Liew

Sean McFadden

Nayra Lujano

Christopher Hazelton

1.2 Required Skill Sets for Your Project

C++ Programming

Python Programming

Virtual Training Environment Skills

Embedded Systems

1.3 Skill Sets covered by the Team

C++ Programming - Everyone

Python - Nayra, Sean, Tyler

Virtual Training Environment Skills - Tyler, Sean

Embedded Systems - Everyone

1.4 Project Management Style Adopted by the team

Agile Management Style

1.5 Initial Project Management Roles

Amy Wieland - Project Manager

Tyler Ingebrand - Project and Machine Learning Manager

Nathan Bruck - External Hardware/Arduino Manager

Yi Ting Liew - Task Board Manager

Sean McFadden - Machine Learning Manager

Nayra Lujano - Research Manager

Christopher Hazelton - Security Manager

2 Introduction

2.1 Problem Statement

The main goals of our project are to successfully demonstrate embedded machine learning on an interesting application and to make recommendations for incorporating embedded machine learning in a course for the CPR E department. Initially, one of our main tasks has been to decide on what application we would like to apply embedded machine learning to. With the open-endedness of our project, defining requirements has been a little bit challenging as there is not a specific application and hard requirements or constraints detailed in our project abstract.

Our team has decided to train a robot to walk via reinforcement learning as our chosen application for our embedded system machine learning project. The machine learning algorithms that are chosen must be applicable to the robot and, thus, will be restricted to the resources of the robot's embedded system. We plan to utilize a combination of virtual environment and physical environment to train the robot throughout the project time duration.

2.2 Requirements & Constraints

Our functional requirements include training the robot virtually using the OpenAI Gym and MuJoCo toolkits. Virtual training allows us to utilize more computing power for training, and the physical components of the robot will take no wear or damage. The inference process must be done locally on the robot. The robot should be able to walk stably without having to send information to an external device to process the data. Our current goal is to have the robot travel a minimum of three feet within 20 seconds, but this is subject to change once we get access to the robot and test its capabilities.

Another type of requirement the team needs to be aware of is the different types of resource requirements the project has. First, the team intends to use the Petoi Bittle Robot Dog as the test bed for our embedded machine learning application. The reasoning for using the Petoi is because it is easily repairable, and we wanted to be able to have a usable device that could easily be repaired and maintained in working condition. With those considerations in mind, the Petoi Bittle is now the chosen platform to use for our application, hence it is a resource we will require. Secondly, we need to ensure that we are using a platform that supports the embedded system usage we are aiming for. Because of this, we will require a resource such as a Raspberry Pi or a type of Microcontroller that allows us to program the Petoi Bittle Robot Dog.

These two resources in mind, we will also need to keep our system somewhat modular, since we want to be able to add external functionality to our application for scalability and practical usage for student projects or other classroom and university applications. Lastly, one of the other most important resource requirements we've identified is being able to use resources a university would have access to for course implementation. This includes using affordable and accessible components and using systems and coding languages the university has access to.

Other requirements of our project include compiling a list of useful machine learning resources that demonstrate new learning the team has acquired as well as attending keynotes at Imagine 2021 and completing the Coursera course on embedded machine learning. These resources and new learning will help facilitate the development of a machine learning course. Additionally, given our project is more open ended, part of the main initial requirements of the project is to define exactly what our project focus will be and figure out what requirements are necessary for the selected application. One of the main goals of the project is to be able to incorporate what we develop and learn into a course at ISU; therefore, it is important that we develop in a way that is modular so that course implementation is feasible.

As for constraints, we are working on a clock and need to have the project done by the end of the spring semester of 2022. We need to be able to get the robot in time and then train the robot.We are working with a budget of $600, which should be plenty as the robot costs $300. Since we are trying to create a class or concepts for a class out of this project, we need to use reusable components such as common coding languages, specifically C++ and Python. The tools that we will be using for training the robot are OpenAI gym and MuJoCo as the 3D simulator training area. The inputs that we get are IMU readings and servo positions and the

2.3 Engineering Standards

The first engineering standard associated with this project is the UM10204 I^2Cbus specification. The chip onboard the robot is the ATMEGA328 and will not likely have the computing power necessary for our application. The board has the ability to switch communication from the ATMEGA328 chip to an I^2C bus controlled by a Raspberry Pi. The next two standards relate to the design and testing of autonomous robots. The IEEE P1872.2 standard for autonomous robots ontology set guidelines for building autonomous systems consisting of robots operating in various environments, and the P2940 standard for measuring robot agility will provide quantitative test methods that are useful to show how well our robot walks. The Petoi Bittle robot dog is powered by a lithium ion battery pack sized for its current peripheral load. However, as we add more functionality, the battery pack may need to be expanded to accommodate the extra load. The standard that will apply to this is the IEEE 1725-2021 standard for rechargeable batteries for host devices such as mobile phones. Lastly, we may want to incorporate wireless communications for control of the robot, which will be accomplished through the Raspberry Pi. The IEEE 802.11 standard for wireless local area networks will guide our use of the wireless network.

2.4 Intended Users and Uses

Being an open-ended project for a professor, the intended users and uses are more broad. First of all, our project will be a proof-of-concept for creating a machine learning application using a robot in one semester. In doing so, our project will provide resources for learning about and jump starting a machine learning application. The intended use of our project is to demonstrate that a one semester course could be designed around machine learning in embedded systems, and to provide resources for creating that course. The user of this project will be Dr. Rover, who will take the information we provide and the results we achieve to determine if she is interested in creating a course.

3 Project Plan

3.1 Project Management/Tracking Procedures

Our group is utilizing an agile project management style. With the main goals of the project being to successfully demonstrate embedded machine learning (ML) on an interesting application and to make recommendations for incorporating embedded ML in a course for the CPR E department, we will be able to break down the implementation of a walking robot, our selected application, into tasks that can readily be completed in an agile environment. Our group is using github to track our progress on the project. We are using Slack to communicate with the project owner Dr. Rover and update her on the progress.

3.2 Task Decomposition

1. Get linux running on robot, run a basic hello world (C++)
2. Define the Action and State structs/arrays
    a. What values go in it and what they represent
    b. This is based on what is available from the robot
3. Create interfaces for both the embedded and agent (C++)
    a. This allows us to replace the agent easily without changing the application main loop
4. Create general use logger (C++)
5. Implement embedded side (C++)
6. Implement Stretching agent - IE proves that embedded side works and test robot joints if one is acting up (C++)
7. Implement NN agent (C++)
8. Implement application main loop - makes calls to interfaces, not to specific implementation (C++)
9. Set up environment for training (Python)
10. Create virtual dog robot in MuJoCo (Python)
11. Create and train a model to walk using any pre-built virtual robot- proof of concept for training in virtual env using a pre-built virtual robot such as the spider (Python)
12. Create and train a model to walk using our robot in virtual environment (Python)
13. Use model in NN agent class - actually trying the model in real life (C++ and Python)
14. Refine virtual environment to improve model as needed (Python)
15. Refine NN agent (C++) as needed - IE possibly modify the actions to account for friction or something

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

M1. Assemble both robots and run factory code.
- Evaluation: complete when all servos move using the code provided by Petoi.

M2. Complete the "Mountain Car" problem.
- Evaluation: complete when the car can reach the top of the mountain using both DQN and DDPG algorithms.

M3. Move the servos on the robot through the Pi.

- Evaluation: complete when the Raspberry Pi can move all servos through I2C communication with the robot's microcontroller.

M4. Train the virtual robot to walk 3ft within 20s.

- Evaluation: complete when the robot in the virtual requirement can stably walk at least 3 feet within 20 seconds of execution.

M5. Deploy the trained model onto a physical robot.

- Evaluation: complete when the agent on the robot uses local inference from the trained neural network to control the robot servos.

3.4 Project Timeline/Schedule

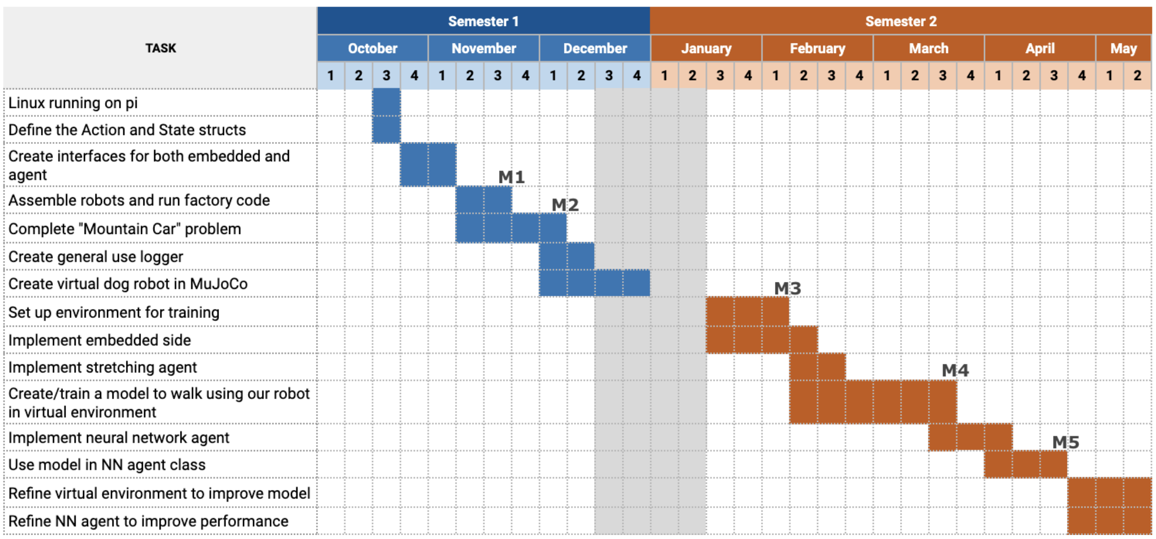

Figure 1: Gantt chart representing the schedule of our project

This schedule allows us to divide our tasks into sprints typically lasting two weeks. Because we will be training virtually, much of the training can be done in parallel while work on the robot is being done. We will divide our team into one group for starting the virtual training process and another group for preparing to deploy our model on the robot. Towards the end of the second semester, we will all be able to help refine the ML model. There may need to be some training on the robot itself if the virtual training fails to accurately simulate the real world.

3.5 Risks And Risk Management/Mitigation

Table 1: Project risks and mitigation plan

Task	Risk	Risk Probability	Risk Mitigation
Get linux running on robot, run a basic hello world (C++)	Robot delivery is delayed	0.2	

Define the Action and State structs/arrays	Incorrect action and state structs/arrays defined	0.2	
Create interfaces for both the embedded and agent (C++)	Implementation of interfaces takes longer than expected or doesn't work as expected when using to easily replacing the agent	0.1	
Create general use logger (C++)	Logger does not work as expected	0.1	
Implement embedded side (C++)	Inexperience of team with implementing embedded side especially with ML	0.5	To mitigate this risk, we will seek external resources that have implemented something similar for assistance. We will also account for the lack of experience in our project timeline by allowing for some flexibility with this task if it is to take longer.
Implement Stretching agent	Trouble defining what actions we should specify	0.2	
Implement NN agent (C++)	Inexperience with tensor flow and loading a model	0.4	
Implement application main loop	More complex than we are predicting	0.3	
Set up environment for training (Python)	Environment is harder to set up then expected	0.8	Spend more time on it as needed
Create virtual dog robot in MuJoCo (Python)	We can't get the data values we want	0.3	Figure out a way to convert the available data values to the ones we want by adding additional broken down tasks if necessary
Create + train a model to walk using any pre-built virtual robot- proof of concept for training in virtual	Pre-built virtual robot is significantly different than our robot	0.5	Research pre-built robots thoroughly before selecting to make sure it will align well with how our robot works. If the options for a pre-built virtual robot are limited and none align well with our robot, then we can add additional tasks to our project plan to

environment using a pre-built virtual robot such as the spider (Python)			convert the pre-built implementation to our robot.
Create and train a model to walk using our robot in virtual environment (Python)	This tasks could be more challenging than we initially expected, could take more time than we initially plan for	0.5	In order to reduce this risk, we should allocate some extra time for this task when creating our project timeline
Use model in NN agent class	The model set up in the virtual world does not align well with the physical world, causing the completion of this task to take longer	0.7	It is likely that some modification will be needed when transitioning from virtual to physical environment, therefore, we should account for this upfront in our project plan and allocate enough time for refinement of the model.
Refine virtual environment to improve model as needed (Python)	More refinement is needed than planned for, we are unable to refine to where we need to be	0.8	To mitigate this risk, we are planning to allocate a significant amount of time towards this part of the project. We also will utilize our resources to become as educated as we can about our application, NN's, and ML in order to reduce exposure to this risk.
Refine NN agent (C++) as needed	More refinement is needed than planned for, we are unable to refine to where we need to be	0.7	To mitigate this risk, we are planning to allocate a significant amount of time towards this part of the project. We also will utilize our resources to become as educated as we can about our application, NN's, and ML in order to reduce exposure to this risk.

3.6 Personnel Effort Requirements

Table 2: Personnel effort requirements

Task	Person-hours	Explanation
Interface robot with Linux	2h	It's designed for it, should be easy
Create interfaces for both the embedded and agent	2h	These should be easy to write as we have already discussed what they will look like, though its not set in stone
Create general use logger	1h	This is just a convenience class for logging, should be some basic file management

Implement embedded side	30h?	We are unsure what this will look like so we are allotting a significant amount of time to figure it out
Implement Stretching agent	4h	Just has to implement a few functions, can be a series of hard-coded actions
Implement NN agent	8h	This is more complicated, we need to read the docs for tensorflow to figure out how to load and use a model in C++. Note this does not include training the model, only loading it and getting an output from an input. We also will not be able to test it until we have a model that can be loaded.
Implement application main loop	4h	Well abstracted, this should only be a few lines of code. May require some additional arg parsing.
Set up environment for training	10h	Setting up envs in Python is notoriously a pain
Create virtual dog robot in MuJoCo	4h	This might require a lot of file editing and trial and error
Create + train a model to walk using any pre built virtual robot	15h	Figuring out how to properly train a model in MuJoCo will be a lot of learning and experimenting
Create + train a model to walk using our robot in virtual environment	15h	Once we have a virtual model of the dog, and we have completed the above step, training for the dog should be doable. Training itself may take some time
Use model in NN agent class	1h	If we have the agent class done, and the model done, we just need to load the model into the agent class. Should just be file transfer
Refine virtual environment to improve model as needed	TBD	
Refine NN agent	TBD	

3.7 Other Resource Requirements

- OpenAI Gym: toolkit for training an agent using reinforcement learning
- MuJoCo: physics engine for building a virtual representation of our robot to train with OpenAI Gym
- Tensorflow: machine learning framework used to design our models and deploy them onto our robot
- Petoi Bittle: robot dog to deploy machine learning on.
- Raspberry Pi / microcontroller: Run ML model and interface with Petoi Bittle via I2C.

- A Linux machine for training

4 Design

4.1 Design Context

4.1.1 Broader Context

Our robot application serves as a proof of concept that can be utilized in several different contexts. For example, a walking robot could be used in search and rescue scenarios. It could also serve as a home assistant for elderly or people with disabilities. It can be used as a learning tool in a classroom setting. These are just a few examples of areas where a walking robot could be utilized. With the varying areas that our application could be used, the particular community we are designing for can differ. Based on our project abstract, we are most focused on designing an application that could be integrated into a classroom environment. In this context, we would be focused on the college community. Many communities could potentially be impacted by our design. Focusing on the classroom context again, if students are provided with more knowledge about machine learning, they can then utilize this knowledge in communities outside of the college community. As machine learning is becoming more and more popular, our project is providing a potential method to better educate students about machine learning at the undergraduate level.

Relevant considerations related to our project:

Table 3: Discussion of the potential societal areas of impact that can be experienced through different implementations of this project

Area	**Description**	**Examples**
Public health, safety, and welfare	Our project is a proof of concept for a robot that can perform actions based on its environment. This can be implemented into society to perform tasks that are typically too dangerous or not cost effective for a human.	• Monitoring of sensitive ecosystems • Search and rescue operations (typically lack numbers and coordination) • Disaster relief/cleanup (Fukushima disaster and WTC collapse)
Global, cultural, and social	As autonomous systems develop and the cost lowers, society may see jobs that are performed by humans start to disappear. This can exacerbate existing problems like unemployment and poverty. A possible positive cultural impact could be seen in the elderly and handicap	Development or operation of the solution would violate a profession's code of ethics, implementation of the solution would require an undesired change in community practices

	communities since this application could assist them.	
Environmental	The attainment of materials utilized to manufacture the robot as well as other resources we need for our project, like a Raspberry Pi, could potentially have negative impacts on the environment. Additionally, the energy usage to run the robots could also be a potential negative environmental impact.	• For a classroom setting, the robots would ideally be purchased once. Therefore, there wouldn't be much concern around the negative environmental impacts resulting from high manufacturing rates. The robots would hopefully be highly used, so there may be some slight negative impact to the environment through energy consumption. • For use of the robot in other applications where the robot would be sold as a product, there would be more concern with the environmental impacts that result from manufacturing as well as the energy used to manufacture and use the robot itself.
Economic	Our project requires a robot, which can be somewhat expensive. The particular robot we have selected is around $300. This cost was well within our $600 project budget. Additionally, the robot we have chosen has the ability to replace single parts on the robot. This allows for maintenance and repair without having to replace the entire robot.	• For a classroom setting, this product would be affordable. In the classroom, the robot would be used as a learning tool to gain machine learning experience. Robots would be a one-time purchase and specific parts could be repaired on a case-by-case basis. • As a home assistant for elderly or people with disabilities, the development cost and price of the robot could make it not a very

		feasible product for everyday citizens. • For a disaster relief type situation where a company is interested in investing a larger sum of money to improve their processes, the robot would be useful and affordable. The robot would present an opportunity for economic advancement for the company if it improves their processes, leading to faster, better results.

4.1.2 User Needs

Teachers - Teachers need a way to confirm what the student is doing is correct because that would be how they would get their grades.

Students - Students need a way to code the robot to perform the task assigned to them by the teacher because that would be their project for the week/month.

Search and Rescue - Search and rescue would need a way for the walking robot to be able to find people who are in danger and be able to help them to a safe location because the main focus of search and rescue is to get people out of harm's way and make sure that they are safe and sound

Delivery Service - Delivery service would need a way for the walking robot to go to a specific location based on where the package needs to go because that would get the package there safely and to the correct location.

4.1.3 Prior Work/Solutions

The development of a walking robot via machine learning is something that has been done before by several different companies. However, our project brings the angle of utilizing our product to integrate machine learning into the classroom. As machine learning becomes a more important tool, integrating this concept into undergraduate courses will be beneficial for computer science students. Teaching machine learning at the undergraduate level can be somewhat challenging because it requires students to make “linkages between complex concepts in linear algebra, statistics, and optimization” (Sahu et al., 2021). In a preliminary study by Sahu et al. titled, “Integrating machine learning concepts into undergraduate classes”, methods to best teach machine learning were explored. The researchers explored teaching machine learning in a side-by-side method and as stand alone workshops. In the side-by-side approach, the machine learning concepts were integrated into a pre-existing signals and system course. The researchers found that “while students like the side-by-side delivery better, the workshops showed improved student learning” (Sahu et al., 2021). As we develop our machine learning walking robot, we will consider the findings in this study as we consider methods to best help undergraduate students learn and implement machine learning.

4.1.4 Technical Complexity

The hardware involved in this project is of sufficient technical complexity because it involves many components and subsystems communicating with each other to allow the robot to walk. The main component inside the robot dog is the ATMega328P microcontroller. However, this chip is most likely insufficient to run our program allowing the robot to walk, so we will interface a Raspberry Pi to act as the brain. The physical walking of the robot is accomplished by servos that are driven by the PCA9685 which provides 16 PWM 12-bit channels. Our robot also requires positional data of all the external limbs and shell of the robot itself which is generated by the MPU6050 which is a 6-axis IMU. All of these components communicate through an I^2Cbus featuring the Raspberry Pi as the master. The following diagram provides an overview of the hardware platforms discussed above.

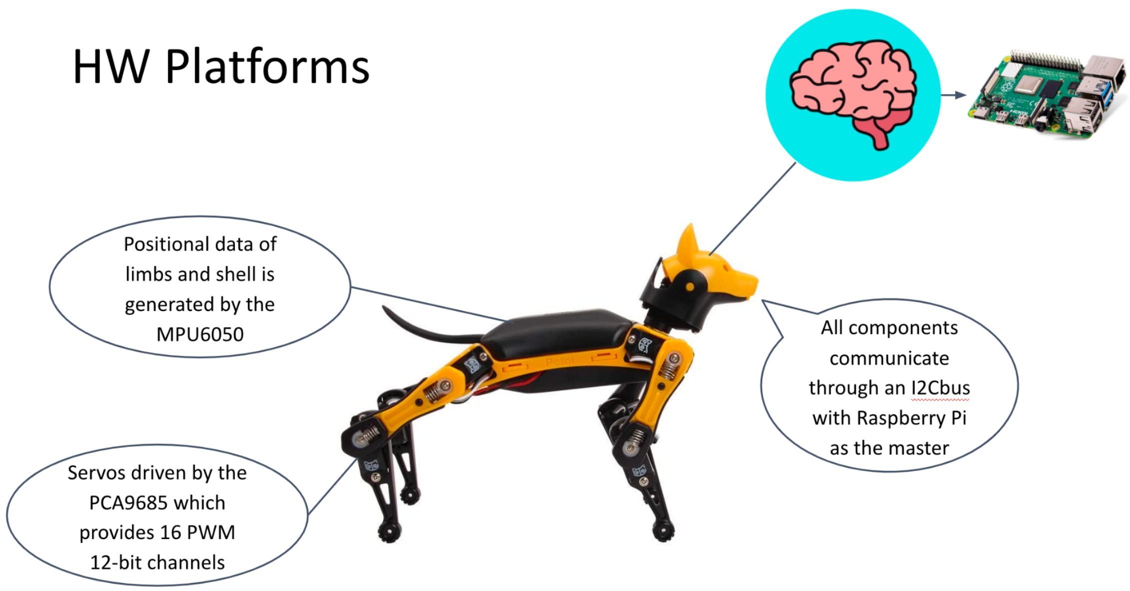

Figure 2: Hardware Platforms

The machine learning aspect of this project will consist of training a neural network in a simulated environment through reinforcement learning (using OpenAI Gym, TensorFlow, and MuJoCo). Utilizing linear algebra concepts, we will have to develop the different layers of the neural network for our robot dog to be able to walk correctly. Using various open source projects, we will develop a suitable environment to train a virtual representation of the robot in. The coding for our neural network and simulated environment will be in C++ and Python. Since this project is restricted to the resources available on an embedded system, our solution will have to be complex enough for the robot to walk but simple enough to not require an abundance of resources.

Figure 3: Software Platforms

In order to implement our proposed project, multiple different software platforms will need to be utilized (Figure 3). OpenAI Gym will provide a reinforcement learning environment; TensorFlow will provide libraries for developing the neural network; MuJoCo will facilitate virtual training by providing a 3D simulation environment; Frugally-deep will facilitate loading our model onto the robot by assisting in conversion from Python to C++; And lastly, C++ and Python will be the languages being utilized across the project.

Our project clearly consists of multiple components that will utilize the scientific, mathematical, and engineering concepts of machine learning. Additionally, we have challenging requirements that need to be met through our implementation such as successfully training the robot virtually, demonstrating the virtual model works in the physical world, and meeting our goal of the robot walking 3ft in 20 seconds.

4.2 Design Exploration

4.2.1 Design Decisions

There are some key design decisions that were made in the development of this project. First, we've decided to use a Raspberry Pi controller device to use with the Petoi Bittle robot dog. These main decisions will help define the possibilities of what we can add to the robot dog and how we can work with it. As for tools, we've settled to work with OpenAI Gym and TensorFlow. These resources will help us develop the learning testbed that we will use to train the robot dog. We opted to go with TensorFlow over something like PyTorch because several group members already have previous experience with TensorFlow. This will provide some familiarity as we take on the challenge of teaching a robot to walk.

4.2.2 Ideation

Our main design decision in this project was to choose between a microcontroller or microprocessor to operate our robot dog. For our options, we could have used an Arduino, a Freescale Semiconductor, a Raspberry Pi, a Texas Instruments Tivia C Series (which is used in CPRE 288, so the students would be familiar with the device), or a NXP Semiconductor board (ARM). We identified these options based on their potential compatibility with our project for our project needs and price.

4.2.3 Decision-Making and Trade-Off

When looking at the pros and cons of each of the options, we looked for the price, availability, compatibility with our needs, modularity for interfacing with new hardware (for the classroom setting), availability of documentation and ease of use. With this criteria in mind, we quickly decided the best options were between a Raspberry Pi and Arduino. Based on availability and the need to interface with components, we chose the Raspberry Pi.

4.3 Proposed Design

As of now, we have been researching reinforcement learning and embedded machine learning. We have also obtained two Petoi Bittle robots and have assembled them with the factory code from the producers. We are creating mounting standoffs to attach a Raspberry Pi 3B to the robot. Lastly, we have done deep Q network and deep deterministic policy gradient with the mountain car training problem.

4.3.1 Design Visual and Description

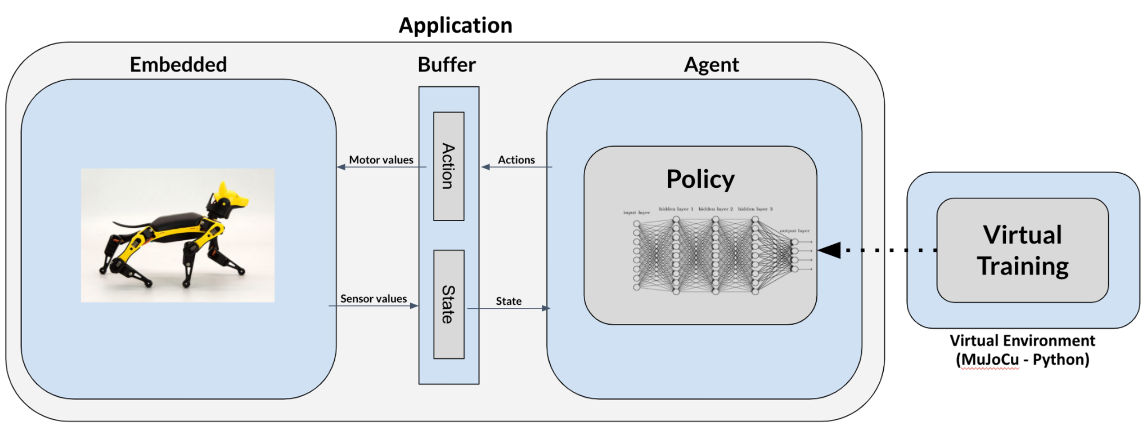

Figure 4: Conceptual Design Diagram

Our application will be broken into two parts: An agent side and an embedded side as seen in Figure 2. The two sides are linked via a buffer that passes the action and state values between them. The policy will be trained virtually and uploaded into the agent. Once the policy is in the robot, it will no longer interact with the virtual environment. Furthermore, the policy does not necessarily have to be a neural network. For testing, we will also create a policy that activates the motors in a predetermined way to make sure all motos are functioning properly and that the embedded side is working as expected. For example, we can put the robot on its back and iterate over the full range of the motors (kind of like stretching). Another important note is that this mimics a typical control theory system. The embedded side acts as an environment that has a state and accepts an action. The agent can contain any policy that makes decisions based on the current state. In addition to the above design, we have some basic pseudocode that will be implemented:

```
Class IAgent (Interface for the agent)
       Constructor(String model) //needs file location of NN if using one
       Destructor
       setState(State s)
```

```
	getAction()
	learn(Reward r) //this one most likely isn't needed unless we want
to learn on the robot

Class IEmbedded (Interface for the embedded side)
	Constructor()
	Destructor()
	getState()
	setAction(Action a)
	getReward() //this one most likely isn't needed unless we want to
learn on the robot

Class Application (IE the main)
main()
	Init embedded
	Init agent
	while(1)
		State s = embedded.getState()
		agent.setState(s)
		Action a = agent.getActions()
		embedded.setActions(a)
```

Note: We want to use interfaces for the agent and embedded side because this allows us to replace it with a different implementation and still use the same main application code. For example, we should implement an agent "Stretching" that verifies all motors can activate. We also need an agent "Walking" that uses the NN to walk. If both agents can be used based on the interface above, then the application code for both of those agents would be exactly the same. Note, there are also potential implementations for the embedded side. For example, one embedded side might not support on-robot learning. Another may support learning, so it also has to calculate reward and return it.

4.3.2 Functionality

One of the main goals of this project is to propose a way to implement machine learning with embedded systems in a course at Iowa State. In a real-world class, students could work throughout the course to implement something similar to the design we have developed here. The professor could decide the extent of what students should implement. For example, students could be provided with the embedded side code and complete assignments focused more on developing the machine learning algorithm that the embedded side will implement.

We have put thought into the components that were selected for the project as well to better cater to our intended user. For instance, the Petoi Bittle's are a relatively cost effective option, small in size, easy to work with, and have a wide range of capabilities. Purchasing multiple of these to re-use from semester to semester would not be out of the question. Additionally, the Petoi Bittle's are easy to fix as the user can purchase a new component individually and replace the broken component, making this an even more cost effective solution. We have also put thought into the software that was selected for the project. OpenAI Gym, TensorFlow, MoJuCo, C++, and Python are easily accessible and free.

So far, the current design satisfies our functional and nonfunctional requirements well. The design has been built with these requirements in mind and also our intended user.

4.3.3 Areas of Concern and Development

Our biggest concern is that the training done in a virtual environment will not translate to the physical world as well. Our best option to address this is to adjust our virtual environment to better match the real world. If this does not work, we will need to train our model using the physical robot. Because training the network is computationally expensive, it may not be possible to do efficiently on the Raspberry Pi. In this case, we may need to wirelessly communicate between the Raspberry Pi and a more powerful computer for training.

The next concern we have is being able to communicate from the Raspberry Pi to the NyBoard on the robot itself. The NyBoard has I2C which we will use. There is documentation of the NyBoard which should show us how to communicate between the two devices. There should also be examples of other people interfacing with the board which we can build off of. If we are still not able to communicate between the boards after this, we will ask for input from Dr. Rover.

4.4 Technology Considerations

The Petoi Bittle robot dog is one of the technologies that we are using in this project. One of the reasons we chose this platform for our application is that it is an open-source low-cost option. The low-cost and open-source nature of the Petoi Bittle allows us to 3D print or easily swap-out or repair components that may break in any test we may run. This also allows a low-cost option for deploying multiple units to a classroom setting. The weakness with the Petoi Bittle is that it is not the most cutting edge or complex technologies that are being used today. This becomes a weakness when considering the goal of our group is to provide a course that educates future engineers in the field of machine learning. If we wish to use these complex technologies, the tradeoff is the course may no longer easily fit into an introductory course in machine learning, for it will be far more complicated.

Another technology we use is TensorFlow, an open-source platform that assists in creating machine learning applications . We chose to use TensorFlow due to some of the features available in TensorFlow and that could take advantage of TensorFlow Lite, which provides TensorFlow options for embedded devices. These features were appealing to our team for the features provided and for other team members being familiar with the framework of TensorFlow. A weakness of using TensorFlow is that PyTorch is another option that could be used that is perfectly viable. PyTorch is often used for more research applications whereas TensorFlow appears to be an industry option. We chose to use TensorFlow because of previous team familiarity and wanted to make the project easier for others to adjust to and ensure team members new to TensorFlow could get support.

OpenAI Gym is the third technology we will discover in this section. OpenAI Gym is a toolkit for reinforcement learning algorithms. It helps create an interface where we can teach agents in the process. We chose an open AI gym due to our chosen reinforcement learning options that we wanted to use for the project. We chose this technology because OpenAI Gym works to be fairly easy to set up and use, and ensure that the reward functions and set of actions available help standardize environments to train in reinforcement learning.

Our team has prioritized low-cost and open-source options in this project. The two largest driving factors for these decisions are to keep the choices within a budget and easily accessible, modifiable and repairable.

4.5 Design Analysis

Our current project plan is working and things are going according to plan. Our proposed design from Section 3.3 is something we are continually evaluating our progress towards. We have successfully completed our first milestone and are on track to finish the second milestone before the end of the semester. We are remaining close to our scheduled progress based on our original timeline. Any work scheduled for the first semester that does not get completed by the end of the semester will be finished over winter break. This will put us on schedule for the start of next semester.

4.6 Design Plan

The main use case for our project is to develop something that could be utilized in a classroom to teach machine learning concepts with embedded systems. In order to accomplish this, we plan to implement the interfaces seen in the figure below (Figure 5). The agent interface and embedded interface will interact as seen in Figure 4 from above.

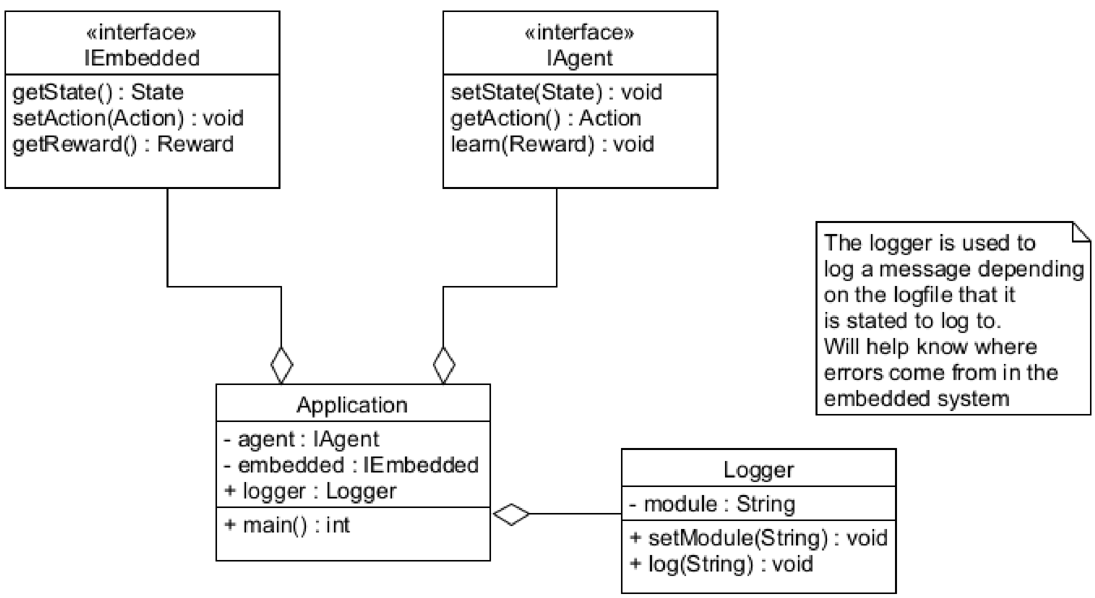

Figure 5: Component Diagram

The agent interface will implement a setState, getAction, and learn method. On the other hand, the embedded interface will implement a getState, setAction, and getReward method. These interfaces are also laid out in section 4.3.1. We also plan to implement a logger that will be used to help track down errors in the development process. As far as concurrency we have two options. One option is to run the agent side and the embedded side in two separate threads. The other option is to call them in a loop in one thread. With one thread, we will be able to use function calls for the implementation. If two threads are used, we will have to implement classes with thread safety. Currently, we plan to stick with one thread and utilize function calls because we believe this will be sufficient for our needs.

One constraint in our design is that our interfaces should allow a variety of policies to be implemented without having to change the main application code. As stated earlier, we plan to implement a stretching agent and a walking agent. The stretching agent will be used to simply verify that all motors can activate; whereas, the walking agent will contain a neural network that the robot uses to walk. By using interfaces, our application becomes a lot more flexible and opens the door to teaching the robot to do other actions beyond walking.

With respect to one of the use cases of this project being suggestions for implementation of machine learning in a classroom environment, students could do something similar to the design plan that we will follow in our project. Professors could even decide to provide students with the main application, agent interface, and embedded interface code and allow students to focus on the machine learning aspect of the project. Students could virtually train as we have in this project and implement all different kinds of machine learning techniques to get the robot to accomplish different tasks.

5 Testing

5.1 Unit Testing

Embedded: We could create some basic test scripts that run through the different range of motor values, and print/log the sensor values as a result. This is not automated as it requires us to physically handle the robot.

Python environment: Render the neural network in the virtual environment and visually inspect that it is behaving as expected. This is the best way to ensure that it is working properly. Python comes with various rendering tools for this purpose.

Neural network agent: We can unit test this by creating a test neural network and applying it in the CPP agent. It should output random actions which would demonstrate that the CPP code is working. Note this does not test the NN, that is done in Python.

5.2 Interface Testing

The microcontroller of our robot will have to efficiently send signals to each of the servos and pull movement data from the IMU to make the robot walk in a coordinated manner. We will have to test the ability to send and receive signals which can be accomplished in real time using a DMM or oscilloscope. The I2C bus our robot relies on can be tested using an Arduino or a simple I2C testing board (ex. Bus Pirate).

Since the training of the NN is accomplished virtually, the results will be loaded onto the embedded system after training. This could pose a problem since the training environment may not be perfectly reflective of the actual environment the robot is used in. Therefore, we will have to test how well the robot performs in the real environment and make adjustments to the virtual environment accordingly.

5.3 Integration Testing

The integration path between the Raspberry Pi and the microcontroller on the robot will be critical. If the Raspberry Pi is unable to communicate with the robot, our robot will not move at all. We can test this by having control scripts that perform set actions on the robot like moving each leg to certain positions. If we run the script on the Raspberry Pi and the robot moves accordingly, we will know the two units are properly communicating with one another.

There is also a link between our virtually trained model and the Raspberry Pi. The Raspberry Pi must be able to deploy the virtual model in order to tell the robot how to act. The model will be trained on a separate computer. This means the Raspberry Pi must be able to download a saved version of the model and interface with it. This can be tested by comparing the output actions of the model on both the Raspberry Pi and the computer the model was originally trained on. We can have a list of sample input input data, and both models should output the same actions.

5.4 System Testing

Every time we make a change, we will make assumptions about which systems may be affected and run the corresponding tests from above. Additionally, we will occasionally run all tests to catch unexpected issues.To get into the tests that suffice for system level testing includes pre-train and post-train tests. The main objective of the pre-train test is to identify errors/issues in advance so that we can avoid a wasted training job. On the other hand, the post-train test could be used to interrogate the logic learned while we are training the model and provide us with a behavioral report of the model.

5.5 Regression Testing

To ensure that our new functionality additions do not break the old functionality, we will need to run tests after each new addition has been applied. Making multiple applications of new features without testing can cause a lot of headaches trying to figure which feature broke the functionality. Critical features that we need to make sure do not break are the movement of the legs of the robot and the ability to stabilize the robot so it does not fall over. The regression testing would be driven by the requirements of the walking robot.

5.6 Acceptance Testing

For our project, we are conducting a proof of concept of a ML walking robot. In the end, our goal is to demonstrate to Dr. Rover, our client, a successful NN model and a physical robot that can learn how to walk based on said model. These are both items that can be physically demonstrated to Dr. Rover. Another goal of our project is to propose a way to implement machine learning into an undergraduate course. Therefore, it would be beneficial to lay out a development process that could be followed along in a classroom setting. For example, we could present modules that mirror the development process that we followed to reach our final product. These modules will rely on each stage of the development process being completed accurately, emphasizing the importance of other types of testing for our ML robot. We will need our development process and implemented system to be reliable and maintainable in order to be utilized from semester to semester. We should also ensure that our robot performs to the expectations of our client. In our functional requirements, we set a goal for our robot to travel at least three feet within 20 seconds.

5.7 Results

The results of testing will be determined next semester when we begin implementing our application.

6 Implementation

To begin our walking robot implementation, we have experimented with a deep Q network and a deep deterministic policy gradient to solve the mountain car problem. This allowed us to practice implementing reinforcement learning before taking on the more challenging task of implementing an algorithm for our robot application.

Through this practice implementation, we have developed a better understanding of the actor critic method. Initially, the neural network will know nothing. The actor will take in a state and then output an action. On the other hand, the critic takes in a state as well as the agent's action and then predicts the value of the next state. The actor will learn based on the critic's suggestion. Utilizing reward, the critic is able to learn and adjust to the desired outcome. As the critic improves, the actor gets better feedback and thus is able to improve. The figure below demonstrates this process.

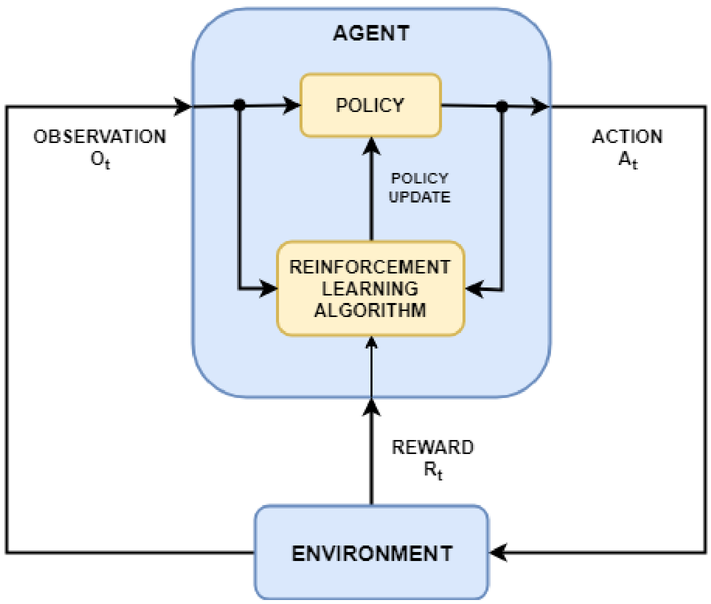

Figure 6: Actor Critic Method

Looking forward, next semester we will establish communication between the Raspberry Pi and the robot. This will give the robot enough resources for us to deploy our model and agent onto the robot. As we are establishing communication, we will complete the virtual model of the robot and train in the virtual environment using a Deep Deterministic Policy Gradient algorithm. Finally, the model and agent will be deployed onto the robot and tested. We have specifically allotted time at the end of the semester for refining our model and fine tuning our algorithm to produce the best results in the physical environment.

7 Professionalism

7.1 Areas of Responsibility

Table 4: Professionalism - Areas of Responsibility - IEEE vs. NSPE

Area of responsibility	**Definition**	**NSPE Canon**	**IEEE**

Work Competence	Perform work of high quality, integrity, timeliness, and professional competence	Perform services only in areas of their competence; Avoid deceptive acts.	The IEEE states as engineers we shall maintain and improve our technical competence and to only take on technical assignments from others if you are qualified by training or experience or have specified your technical limitations in regards to the assignment. This differs from the NSPE because it is more specific about tasks being technical tasks. Additionally, the IEEE mentions how one should act when a task is assigned by another person. The NSPE does not do this.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs	Act for each employer or client as faithful agents or trustees.	Be honest and realistic in stating claims or estimates based on available data. This involves not overcharging for your products and accurately sharing the costs with those who need it. The IEEE and NSPE Canon are very similar in these regards.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	The IEEE states we shall help society understand the capabilities and societal implications of conventional and emerging technologies. Additionally, the IEEE states we shall be honest and realistic when stating claims based on evidence. This differs from NSPE because it talks more about presenting technical information in a correct and understandable way rather than issuing public statements.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	To avoid injuring others and to hold public safety paramount. This is basically exactly the same.

Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	To reject conflict of interest and bribery, and to avoid destroying others' property. This is more direct about what can be done to protect property ownership.
Sustainability	Protect the environment and natural resources locally and globally.		IEEE says to strive to comply with sustainable development practices, and disclose factors that might endanger the environment. This means considering the impact your work will have on the environment. NSPE does not give any information on this topic.
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	To treat others fairly, to not discriminate, and to help others follow this code of ethics. This differs from NSPE because it is more specific about actions to do.

7.2 Project Specific Professional Responsibility Areas

Workplace Competence - This category refers to the area where our project's professional context is able to perform in a high quality of work, integrity, timeliness and professional competence. To imply this professional responsibility, always ensure that the project has been worked in a proper and equally fair timeliness. In addition, weekly meetings with our faculty would be helpful in order to keep track of our project's status and progress as well as to make sure the work quality plus integrity is always on point. Professional competence plays an important role in terms of our team demonstrating a high ability to complete a task in an efficient way.

Financial Responsibility - This responsibility area applies to our professional context because at the beginning of this project, we were given a budget to work within. Another reason that it applies to our context is that part of the reason for our project is that we could possibly introduce this as part of a class for future students, so it would need to be rather affordable. Our team is performing highly in this area as we were given a budget of $600, and we were able to use that budget efficiently and got two robots to work with instead of just one.

Communication Honesty - This responsibility area applies to our professional context because in order to create a well-done project, communication is a key factor. With truthful communication, work can be distributed and put in the right hands for how it needs to be done. Our team is performing highly in this area as we have weekly meetings with our team and the project leader to go over what needs to be done and what has already been done.

Health, Safety, and Well-Being - Our project only has one potentially hazardous aspect which is the battery. Li-ion batteries have the potential to explode if not handled properly, so we will ensure the battery is properly handled at all times. By training our robot virtually, we minimize the potential for damage occurring to the battery. As this project is scaled up, more hazardous conditions will arise. Larger servos can create pinch points that can easily remove a finger if caught, and the fact that a human is not in complete control can create dangerous situations.

Property Ownership - Our project is going to involve integrating some software tools and resources that were developed by others. Therefore, this area definitely applies to our project. Our team is performing well with regards to this responsibility area so far. At this point there haven't been many opportunities where we needed to make sure credit was given to a product creator, but in our presentations and assignments, we have attempted to note where different products come from (such as our Petoi Bittle Robot) and cited a couple research articles that were utilized. As we pull in these resources, we need to make sure we have an area to document and give credit to those that developed the resources we make use of in our project.

Sustainability - Sustainability of the project will be in consideration based on the materials, methods, and practices used to give life to the project. Usages of plastics and microelectronics can have an impact on the environment through how they are sourced from companies. Additionally, other waste may be created by the project. This project performs well with respect to sustainability because it produces very little waste. This is due to the fact that a big part of the project is code (no waste), and the Petoi Bittle is small and tough (will not break, and produces little waste if it does).

Social Responsibility - Our project involves social responsibility. Even with the smaller scope of the project, as in the project is meant to specifically create an impact at Iowa State University, the social responsibility extends to the students' education and beyond that student workplace competency through the framework we hope to create through this project for a future course at Iowa State.

7.3 Most Applicable Professional Responsibility Area

Social responsibility is the most important professional responsibility of this project. The main thought behind this project is to create a framework for students at Iowa State University in order for students to explore and learn concepts of embedded machine learning. Not only do we have a responsibility to provide a useful outline students can use to complete this course, but also the knowledge they learn in this potential course would extend to their knowledge in the workforce they might need to continue to provide new engineering solutions. Because Iowa State students have been hired for many different positions across the world, the impact for this course could help new engineers apply this knowledge across the many different areas they could work in. Our team has demonstrated this responsibility not only through our engagement of students ourselves, but also through the diligent research and the process we are working with to achieve this project. Every team member has gone through self-learning for machine learning and worked to ensure understanding of the concepts before applying them. Through teaching each other and ensuring our design and documentation of the project is thorough, our team demonstrates this professional responsibility in the project.

8 Closing Material

8.1 Discussion

As of this semester, we have begun testing our reinforcement learning algorithms on simple problems, which allows us to identify potential issues prior to implementing them on a complex system. We have not encountered a problem that has not been able to be solved through more testing and research. Additionally, the robots have performed their basic functions well, which places our team in a good position for implementing our own code to integrate the neural network.

8.2 Conclusion

For the embedded aspect of this project, we have fully assembled both robot dogs. An open source library is available to use with these robots and contain simple commands such as walk, sit, and play dead. This software doesn't completely satisfy the needs of our project because the onboard microcontroller is the master of the I2C bus. For our implementation, a Raspberry Pi is used as the master since it has more resources available to implement our neural network, but the open source software serves as a proof of concept.

For the machine learning aspect of this project, we have successfully trained a neural network to solve a basic problem using DQN. This algorithm works for continuous state space, discrete action space problems. Next, we are working on DDPG for the same problem. DDPG works for continuous state, continuous action spaces. Due to the infinite possible action values, DDPG is a more difficult algorithm to implement successfully. Therefore, we are still working on DDPG. Once successful, it will allow us to model our problem, which has continuous state space (sensor values as a float) and continuous action space (motor values as a float). Other algorithms such as PPO can be used if DDPG proves to be more challenging then expected.

8.3 References

1. Seno, Takuma. "Welcome to Deep Reinforcement Learning Part 1 : DQN." *Medium*, Towards Data Science, 21 Oct. 2017, https://towardsdatascience.com/welcome-to-deep-reinforcement-learning-part-1-dqn-c3cab4d41b6b.

2. OpenAI. "A Toolkit for Developing and Comparing Reinforcement Learning Algorithms." *Gym*, https://gym.openai.com/.

3. "Overview." *Overview - MuJoCo Documentation*, https://mujoco.readthedocs.io/en/latest/overview.html.

4. Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971* (2015).

5. "Getting Started." *Petoi Bittle User Manuals 🐶 - Petoi Bittle Manuals*, https://bittle.petoi.com/.

6. “Specification.” – *I2C Bus*, https://www.i2c-bus.org/specification/.

8.4 Appendices

OpenCat
Communication Protocol and Parsing

Interface		Token	Encoding	Parameters		Format	Bytes	Function
RasPi Serial Port	Arduino Serial Monitor	'h'	Ascii			char	1	print help information
		'c'		idx*,angle**	'\n'	string	strlen + 2	calibrate $servo_{idx}$ by angle
		'm'		idx*,angle**	'\n'	string	strlen + 2	move $servo_{idx}$ to angle
		'j'				char	1	show all 16 joint angles
		'd'				char	1	shut down servos
		'p'				char	1	pause motion
		'a'				char	1	abandon calibration
		's'				char	1	save calibration
		'k'		abbreviation	'\n'	string	strlen + 2	load skill
		'w'		command	'\n'	string	strlen + 2	some future command words
		'r'				char	1	reset board
		'i'	Binary	idx_1 a_1 … idx_N a_N	'~'	string	strlen + 2	list of indexed rotation angles
		'l'		a_1 a_2 … a_{DoF}	'~'	string	DoF + 2	list of all DoF rotation angles

* index range: 0 ~ (DoF - 1)

** angle range: -90 ~ 90. fits in the range of signed char (-128 ~ 127). Also depends on the servos’ parameters

Figure 7: Petoi Bittle Use Manual

8.4.1 Team Contract

Team Members:

1) Tyler Ingebrand
2) Yi Ting Liew
3) Amy Wieland
4) Nayra Lujano
5) Chris Hazelton
6) Sean McFadden
7) Nathan Bruck

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:

   The team plans to meet face-to-face or virtually depending on the content needing to be covered during the planned meeting. For meetings with lots of content to discuss or complex problems to solve, the team will meet in person. For meetings focused on progress updates, the team will meet virtually. Regular team meetings will take place on Thursday's at 3:00PM and Friday's at 10:30AM. In person meeting locations will be on campus and virtual meetings will take place through Discord.

2. Preferred method of communication updates, reminders, issues, and scheduling:

   The preferred method of communication will be through Discord. All updates, reminders, issues, and scheduling can take place through Discord. During in person meetings, updates, reminders, issues, and scheduling should be discussed face-to-face as well.

3. Decision-making policy (e.g., consensus, majority vote):

   The team will use a consensus decision making policy. If the team is struggling to reach a consensus, then a majority vote will be conducted.

4. Procedures for record keeping :

   A running Google Doc shared with all team members will be utilized for meeting minutes. The scribe for meeting minutes will be rotated between team members. The meeting minutes will keep brief bullet points covering important topics discussed as well as summarize people's deliverables for the upcoming week

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

   Team members are expected to be punctual. All team members are expected to be present at scheduled team events. If a team member expects to be late or miss a meeting, this should be communicated well in advance of the meeting. At team meetings, all team members are expected to be engaged and participate so that meetings can be conducted efficiently and effectively.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

   Team members are expected to uphold their agreed upon responsibilities, assignments, and deadlines. Team members are expected to make continuous progress on assignments and seek feedback frequently. Procrastinating responsibilities until the last does not meet the expectations of the team. If a problem with a deadline or assignment arises, communicate with the team early. We will all work together to help each other out.

3. Expected level of communication with other team members:

   Team members are expected to communicate often with each other. Over communication is preferred to under communication. As a team, everyone is here to help each other out, but we can only do so if we communicate with one another.

4. Expected level of commitment to team decisions and tasks:

   Team members are expected to be 100% committed to the team decisions and tasks. If a member does not feel they are 100% committed to a team decision or task, they need to communicate this with the rest of the team. The team can then work together to derive a solution so that everyone can be onboard with the plan.

Leadership

1. Leadership roles for each team member:
   - Amy: Project Manager
   - Tyler: Project Manager & Machine Learning Manager
   - Nathan: External Hardware/Arduino Manager
   - Yi Ting: Task Board Manager
   - Sean: Machine Learning Manager
   - Nayra: Research Manager
   - Chris: Security Manager
2. Strategies for supporting and guiding the work of all team members:

   Team members should reach out to each other whenever they have questions. The team should work hard to fairly assign tasks and to not assign any single team member a large solo task. In weekly team meetings, the team should make sure each team member has a clear set of tasks to complete for the next week. A task management board will be created to keep track of what is done and what needs to be completed.

3. Strategies for recognizing the contributions of all team members:

   At each meeting, each team member should state what they have been working on, what they plan to do next, and any problems they have run into.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

   Amy Wieland:
   I am majoring in software engineering and have experience with Java, C#, C/C++, Git, as well as a little bit of experience with HTML, CSS, and JavaScript. This summer I worked at Parametric Studios, an ed tech company at the ISU Research Park, developing an AR application to help young students learn STEM concepts. At parametric, I gained experience utilizing the UnityGame Engine, rebuilding key parts of the UX and UI, and testing application features. Currently, I am in a database course at ISU acquiring experience with MySQL. I enjoy frontend development and gaining new skills, so I am excited to learn more about machine learning and embedded systems.

   Tyler Ingebrand:
   I am majoring in computer engineering with minors in general business and Spanish. My background is in C++ although I have some experience in Java, C#, Python, Julia, and VHDL. I have been doing research in the Virtual Reality Applications Center (VRAC) with projects on various topics from real time application development to supervised NN data collection and training (via Tensorflow). I am also currently working on NNs using Flux (a replacement for Tensorflow) in Julia.

   Sean McFadden:
   I am majoring in computer engineering and have experience with C/C++, Java, Python, and VHDL. I enjoy low-level programming and computer architecture. I am currently involved in research studying fault tolerance in deep learning, so I have recently been learning about CNNs and hardware accelerators. I am also familiar with the PyTorch machine learning framework.

   Nayra T. Lujano
   I am majoring in Computer Engineering with a minor in Cyber Security. I have experience with C/C++, Java, Python, VHDL, QuestaSim and Bash. I’ve worked on different security tasks such as networking through my minor. I’ve worked with 3D modeling, Arduinos and Raspberry Pis.

   Yi Ting Liew
   I am majoring in Computer Engineering and have experience with C, Java, HTML, CSS, MySQL, Git, and VHDL. I am still actively searching for the field that interests me and besides, I am taking a user interface project class which makes me excited to learn more through Javascript and reactJS. I would like to learn about Machine Learning Applications throughout this class along with my teammates.

   Nathan Bruck
   I am majoring in Electrical Engineering and have experience with C, Verilog, and VHDL. I also have experience in circuit design/simulation/construction and PCB design. During co-ops and internships, I have worked on building a flight simulator for aircraft flight management software and designed testing rigs for aircraft components.

<u>Chris Hazelton</u>
I am majoring in Cyber Security Engineering and have experience with C, Java, some Verilog, some VHDL, Bash Scripting, Powershell Scripting, Computer Networking. I have worked an internship the last 2 years at Collins Aerospace with a network security team working on API Gateways and Web Application Firewalls.

2. Strategies for encouraging and support contributions and ideas from all team members:

   To encourage contribution, the team will ensure there is an inclusive culture present within the team. Team members should encourage others to share their ideas by asking what others think. When an individual shares an idea, team members should not shut it down immediately. Individuals should push themselves to not be afraid to share their ideas. When trying to solve a problem, having a brainstorming session may be useful in order to encourage ideas from everyone and prevent groupthink.

3. Procedures for identifying and resolving collaboration or inclusion issues:

   If there is an issue, team members should be upfront and honest about it. Team members need to listen when someone communicates there is an issue, and everyone needs to work together to develop a solution. Open communication will allow everyone to be on the same page and allow team members to make changes to address an issue.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:
   - Have a complete design for the project ready by the end of the semester, so the next semester can be focused on implementation.
   - Define project focus
   - Identify tools needed for project
   - Purchase necessary hardware

2. Strategies for planning and assigning individual and team work:

   A task management board will be created to keep track of what everyone is working on, what tasks need to be assigned, and what tasks have been completed.

3. Strategies for keeping on task:

   At weekly meetings, each team member should share what they have been working on, what they plan to do next, and any problems they have come across. A task management board will be created to help the team know what each other is working on and completing.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

Communicate with the person that they are not meeting the expectations that have been set by the team. Then, as a team, develop a resolution plan to help resolve the issue and prevent any future occurences of the infraction.

2. What will your team do if the infractions continue?

   If the person still fails to meet the expectations, the team will then discuss with the TA and/or advisor to figure out an action plan on how to resolve the situation.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the

consequences as stated in this contract.

1) Tyler Ingebrand DATE Dec 5, 2021

2) Nathan Bruck DATE Dec 5, 2021

3) Chris Hazelton DATE Dec 5, 2021

4) Sean McFadden DATE Dec 5, 2021

5) Amy Wieland DATE Dec 5, 2021

6) Yi Ting Liew DATE Dec 5, 2021

7) Nayra T. Lujano DATE Dec 5, 2021