3 Design

3.1 Design Context

3.1.1 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:

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 communities since this application could assist them.	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
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.

3.1.2 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 search and rescues main focus is to get people out of harm's way and make sure that they are safe and sound

Disabled/Older people who cannot perform the functions that they once used to need a way to have the robot be able to perform the functions that the person desires in their home because they are not able to do it themselves.

3.1.3 Prior Work/Solutions

The development of a walking robot via machine learning is something that has been done before by quite a few 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.

3.1.4 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 Machine Learning aspect of this project will consist of training a neural network in a simulated environment through reinforcement learning. 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.

3.2 Design Exploration

3.2.1 Design Decisions

There are some key design decisions that were made in the development of this project. Firstly, 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 Open AI Gym and TensorFlow. These resources will help us develop the learning testbed that we will use to train the robot dog.

3.2.2 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.

3.2.3 Decision-Making and Trade-Off

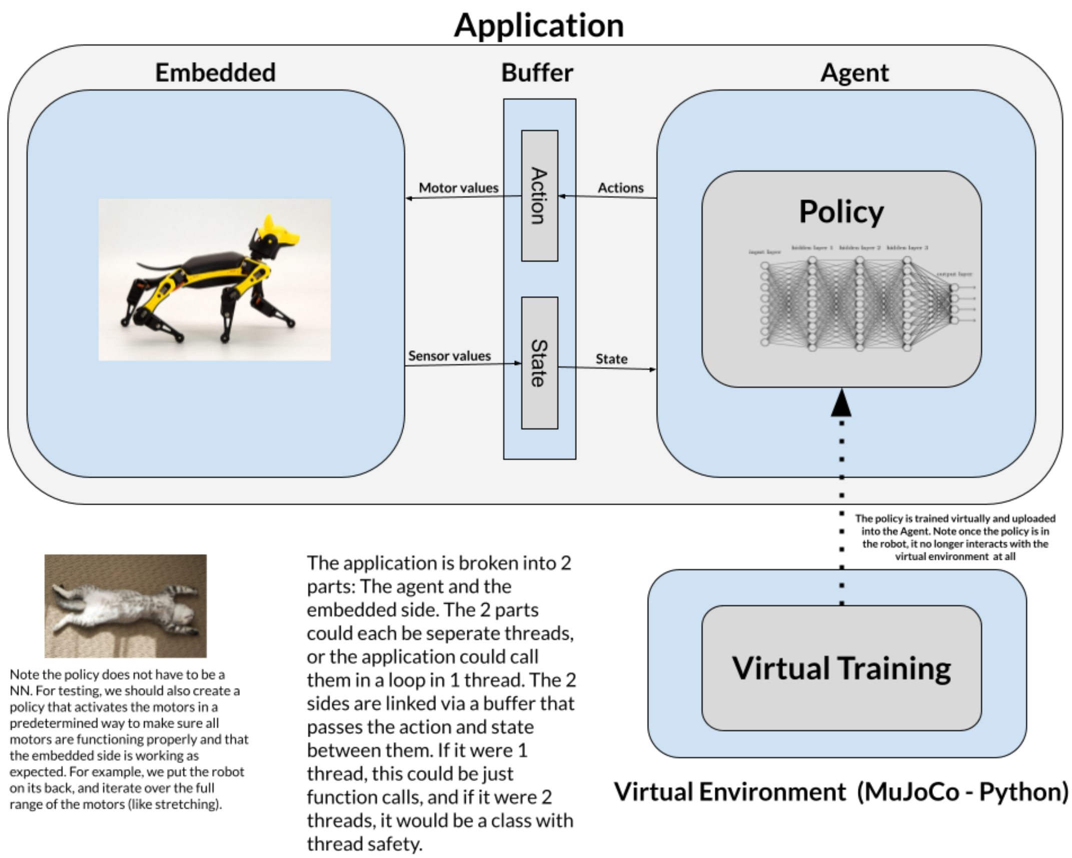
When looking at the pros and cons of each of the options, we looked for the price, availability, compatibility with our needs, modularity for interfacing with new hardware (for the classroom setting), availability of documentation and ease of use. With this criteria in mind, we quickly decided the best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

3.3 Proposed Design

From a machine learning perspective, we have tried some basic training. We have done supervised learning as a first step. We are also experimenting with reinforcement learning on a virtual machine provided by ECG.

From an embedded/hardware perspective, we have ordered the robot and will begin experimenting with it when it arrives.

3.3.1 Design Visual and Description



In addition to the text in the image, 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.

3.3.2 Functionality

The walking robot design is a potential supporting feature for many applications. The ability to traverse complex environments that are not necessarily conducive to wheels is essentially what we are solving, or aiming at taking a step towards solving. Such a system has applications in many fields, such as disaster recovery, home assistants, manufacturing, and other work too dangerous for a person to do.

The current design satisfies our design requirements, which were to create an embedded application that utilizes machine learning. Based on our desired goal of creating a robot that can walk on its own, the current design should allow for that to be possible.

3.3.3 Areas of Concern and Development

Our biggest concern is that the training we do in a virtual environment will not translate to the physical robot. 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.

References

Sahu, Chinmay et al. “Integrating machine learning concepts into undergraduate classes.” Clarkson University, 23 July 2021. Accessed 17 Oct. 2021.