Quadrupedal Robot

Senior Design Project

Ryunghoon Ahn, Christopher Valdes, Samuel Lee

Students

University of Texas Rio Grande Valley, Edinburg, Texas, 78539

Dr. Dongchul Kim

Supervisor

University of Texas Rio Grande Valley, Edinburg, Texas, 78539

In the field of robotics, there is an ongoing effort to utilize artificial intelligence to perform obstacle avoidance. During the Spring 2023 semester, our team hoped to start research into this field, by first constructing a quadrupedal robot and then programming this quadrupedal robot to move in a straight line. Furthermore, documentation in the github repository is meant to help future students, who build upon our knowledge in the future.

Most computer science programs focus on software, theoretical math, and programming. However, some fields of research require hands-on experience with hardware. In this project, by building, testing, and writing code for a quadrupedal robot, our team aimed to gain hands-on experience with not only hardware, but how this hardware integrated with the software component of our quadrupedal robot.

Boston Dynamics has already created quadrupedal robots, such as their Spot robot, which is able to navigate terrain and even stand upright after being flipped over. However, the extremely high financial cost for robots such as Spot may not be an option for every research lab or university. Our quadrupedal robot, using 3D printed parts and other electronics such as a Jetson Nano and jumper wires, aims to have a cost effective yet fully functional robot capable of being used as research into obstacle avoidance and artificial intelligence.

Our end goal for this semester was to have a walking quadrupedal robot that is able to move in a straight line using servos, a jetson nano, and 3D printed parts. The 3D printed parts were printed at the Makerspace lab in the engineering building of UTRGV (University of Texas Rio Grande Valley). Electronic parts were purchased using funds from the Artificial Intelligence department, as were tools, batteries necessary for robot construction. Pybullet simulator was used to program the robot to move.

During the end of April, our team managed to make the robot stand upright using servos. Here is a video demonstrating this fact:

https://youtube.com/shorts/rAv_qdHrh14?feature=share

In this video, you can see the servo leg extending.

https://youtube.com/shorts/udVggbsvQW4?feature=share

In this video, you can see the servo leg retracting.

https://youtube.com/shorts/w0CVjQnS3aU?feature=share

In this video, you can see the robot standing up from its resting position.

https://youtube.com/shorts/45N6XMv1CY0

In this video, you can see the robot moving all legs in a walking motion.

Instead of the Issac Gym simulator, our team chose to work with the Gazebo simulator as the Gazebo simulator has Robot Operating System (ROS) support. ROS libraries and tools were helpful in the running of the quadrupedal robot simulations. One trade off was that less information, guides, and helpful informational threads were available for Gazebo, although plenty of documentation and similar support were readily available for Gazebo. We also used the Jetson Nano as our main computer for the robot, which is less beginner friendly than Arduino, but is able to run Linux and is essentially a fully functioning computer.

Towards the end of the semester, we also used Pybullet, which allowed us to simulate both soft and rigid body dynamics. We were unable to implement OpenAI within the time frame allotted for this project, so instead we took inspiration from the previous quadrupedal robot communities who worked on this project and used Pybullet to determine which policies our simulation should use. This helped us figure out which policies to use in our Gazebo and ROS implementation.

Q-learn was the first algorithm we started with. This algorithm seemed like a promising reinforcement learning algorithm, which chooses actions at random and maximizes reward. However, Q-learning does not fit the scope of our project, and requires us to know specific states which is not feasible with the massive scope of our project.

The next algorithm we tried was Double Deep Q-Network (DDQN). DDQN uses two neural networks, one to select the best action and the other to estimate its value. By doing this, we prevent overestimation of Q-values, which leads to bad policies. A policy is a mapping from states to actions that an agent follows during environment interaction. The policy determines what action the agent should take to get the maximum reward. So DDQN allows us to avoid any bad mapping, essentially. The target network is frequently updated to match the online network, which helps the learning process.

The final algorithm we would like to cover in this report is Proximal Policy Optimization (PPO), a reinforcement learning algorithm that finds the best policy for an agent interacting with an environment. PPO updates the policy by maximizing the expected reward, while limiting the update to be within a trust region. This prevents the agent from doing anything risky. This constraint is achieved by using a surrogate objective function that calculates the true objective function within the aforementioned trust region.

Reinforcement Learning is all about testing and trials. We tested that our controller was actually connected to the joints of the robot in the simulation, which allowed us to ensure that the joints worked correctly. It was also important to ensure that the physics within the environment were correct. Finally, all of the parameters have to be perfect to make a reinforcement learning simulation work. Some examples include set actions, observations, rewards, and constraints.

Our team did not have a strict weekly schedule. Tasks were assigned based on completion. We laid out what each teammate was responsible for at the beginning of this project, although even this was flexible and subject to change. In short, the schedules for all three teammates changed according to progress made with the project.

Initially, Ryunghoon would focus on the procurement of parts for, and the assembly of, the quadrupedal robot. Samuel would focus on 3D printing. Christopher would focus on researching different simulators, and installing, running, and learning how to program in Gazebo.

Then, Samuel completed 3D printing and worked on implementing the gyroscope. Ryunghoon successfully completed robot construction. Christopher was able to successfully get the robot to stand up using servos. Ryunghoon began working on the simulation after completion of the quadrupedal robot. Christopher also focused his efforts on the simulation. Then, Samuel finished the gyroscope implementation and worked on gathering documentation, and the final report. All members contributed to the final presentation.

We must emphasize that there was a lot of back and forth help between teammates, and even help requested from other people within the department. When we discovered that the stl files had to be modified, Christian Narcia, a fellow student in the Artificial Intelligence lab, helped modify these files for us. Christopher was able to print a few of the smallest parts that were not printing correctly at the engineering building, using his friend’s 3D printer. Samuel also researched both the simulators and simulation, and information was shared between all three teammates constantly.

Additionally, since work was divided into sections that did not overlap at the start, for the first half of the semester, each teammate was assigned a task different from other teammates, so there was not really inefficient overlap. Later in the semester, to avoid unnecessary overlap and redundancy, Samuel focused on documentation/presentations, and Ryunghoon and Christopher worked on the simulation. Even then, initially Ryunghoon worked on a simulation without ROS, and Christopher worked on a simulation with ROS.

Discord was used in order to both communicate between teammates and communicate with the professor. “MI@UTRGV” was the research lab’s official discord channel, where the team could communicate with other students in the research lab. Whenever a teammate had a question for anyone, discord was the best method of communication besides in person conversations.

Schedules of all teammates tended to change based on what was needed from each teammate. For example, programming the quadrupedal robot to move was a lot more difficult than anticipated, so for our weekly presentation on April 21, 2023, we moved our planned presentation on the robot back a week, and focused instead on presenting the gyroscope and progress made on the simulation. Our schedules and timetables changed constantly given the current state of progress with this project.

One major change to this project was right at the start of the project. Our team realized quickly within the first week of the project that it was impossible to build a quadrupedal robot, program it to move with simulators, incorporate Artificial Intelligence, and then use that Artificial Intelligence to avoid obstacles all in a single semester. A more feasible yet still challenging goal of having the quadrupedal robot completed and be able to move its servos using our simulation was agreed upon.

Secondly, Samuel originally printed a much smaller robot during the winter of the year 2022. However, we decided on a larger robot using another Thingiverse project that was able to house more electronics and circuitry. So, all the original parts for the smaller robot had to be put aside for any future use, and Samuel re-printed the robot during the first few months of Spring 2023.

Both Gazebo and the Jetson Nano are complex, powerful tools capable of bringing this project to life. However it was because of this complexity that progress with programming was slow, as there was both a lot of research that had to be conducted and a wide array of knowledge needed before programming could begin. For this reason, it is highly recommended that future students look into and research Gazebo, the Jetson Nano, ROS, and python extensively before picking up this project.

Neither Artificial Intelligence or obstacle avoidance could feasibly be implemented into the quadrupedal robot given our small timeframe. So our team was right to focus instead on a working walking quadrupedal robot. Now, the quadrupedal robot is able to move thanks to the Pybullet simulation, and we have left documentation, a simulation, and the physical robot for future students to work off of and improve.

A project management obstacle we ran into was near the end of the semester. After spring break, we wanted to spend half of our remaining time on the simulation and the latter half on using the code from the simulation to make the robot move by itself. After trying both Issac Gym and Gazebo simulators, we decided instead to work with the Pybullet simulator. This simulator allowed us to finally get a working simulation, and helped us with the robot moving demonstration shown in the earlier videos.

Teamwork was absolutely essential, as Samuel and Christopher had to work together to get a few small pieces 3D printed, Christopher and Ryunghoon shared information and ideas about the simulation, Samuel and Ryunghoon communicated back and forth to ensure that

Ryunghoon had all the 3D printed parts necessary for assembly. These are just a few examples of many where teamwork was essential for this project.

There is always going to be a difference between what you see in a computer and how that translates into a real life scenario. 3D part files that seemed like they would print perfectly took two or even three tries to get right. Helpful guides on the internet cannot account exactly for what you are trying to do, as evidenced by the fact that the team had to learn firsthand that the Jetson Nano, by itself, could only connect to one SCL/SDA dependent device at a time. Even youtube videos of talented programmers showing a complete simulation of a walking robot did not translate exactly into code we could use, as our robot did not have the same exact parts/orientation, or we ran into other difficulties. For a more specific example, Samuel constantly had to change splicer settings and model layout during 3D printing to avoid bad prints, as some prints were disfigured or unusable and had to be reprinted. However, despite all of this, we managed to fulfill most of our original goals and have left documentation, simulation, and the physical quadrupedal robot for future students.

Bibliography:

Addison. “Visualize IMU Data Using the MPU6050, ROS, and Jetson Nano” automaticaddison.com, April 19, 2021. https://automaticaddison.com/visualize-imu-data-using-the-mpu6050-ros-and-jetson-nano/.

Kim, Deok-yeon. “SpotMicroAI.” spotmicroai.readthedocs.io, N.d. https://spotmicroai.readthedocs.io/en/latest/.

Kubina, Michael. “SpotMicroESP32.” github.com, June 1, 2021. https://github.com/michaelkubina/SpotMicroESP32.

Martínez, Joël. “Ros Based Control With Gazebo Simulation For SpotmicroAI” gitlab.com 2021. https://gitlab.com/public-open-source/spotmicroai/simulation/-/tree/master/UserJo%C3%ABl%20Martinez.

Nicrusso. “Rex: An Open-Source Quadruped Robot.” github.com, May 15, 2021. https://github.com/nicrusso7/rex-gym.

Parrilla, Miguel Ayuso. “DIY Hobby Servos Quadruped Robot.” hackaday.io, June 5, 2020. https://hackaday.io/project/171456-diy-hobby-servos-quadruped-robot.

Schoeffer, Michael. “Tutorial: How to use the GY-521 module (MPU-6050 breakout board) with the Arduino Uno.” mschoeffler.com, October 5, 2017. https://mschoeffler.com/2017/10/05/tutorial-how-to-use-the-gy-521-module-mpu-6050-breakout-board-with-the-arduino-uno/.