**Boston University**

**Electrical & Computer Engineering**

**EC463 Senior Design Project**

First Semester Report

Mars Rover: Autonomous Navigation

Submitted to

Osama Alshaykh

8 St.Mary’s Street

Room 424

858-361-9043

[osama@bu.edu](mailto:osama@bu.edu)



by

Team #28

Mars Rover

Team Members

Daniel Crawley crawley@bu.edu

Brian He brianhe@bu.edu

Tommy Lam tlam11@bu.edu

SeungYeun Lee sylee538@bu.edu

Linden Vo [lindenvo@bu.edu](mailto:lindenvo@bu.edu)

Submitted: 12/10/19

#### Table of Contents

[**Table of Contents**](#_heading=h.gaspa1z7x9w1) **2**

[**Executive Summary**](#_heading=h.bk9hi692o011) **3**

[**Introduction**](#_heading=h.1fob9te) **4**

[**Concept Development**](#_heading=h.3znysh7) **5**

[**System Description**](#_heading=h.tyjcwt) **7**

[**First Semester Progress**](#_heading=h.3dy6vkm) **10**

[**Technical Plan**](#_heading=h.1t3h5sf) **13**

[**Budget Estimate**](#_heading=h.4d34og8) **15**

[**Attachments**](#_heading=h.2s8eyo1) **16**

[**Appendix 1 – Engineering Requirements**](#_heading=h.17dp8vu) **16**

[**Appendix 2 – Gantt Chart**](#_heading=h.3rdcrjn) **17**

[**Appendix 3 –**](#_heading=h.26in1rg) **Team Information Sheet 18**

# 

# Executive Summary

Mars Rover Autonomous Navigation

Team 28 – Mars Rover

Since the 1970's, scientists have been sending rovers to Mars in hopes to better understand the planet. Our team plans to build a rover that can accomplish a variety of tasks that might one day assist astronauts working on the surface of Mars. Specifically, we will be focusing on the issue of autonomous navigation.

Our senior design project objective is to build an autonomous rover that can traverse the terrain of Mars or similar and withstand weather and atmosphere conditions using visual navigation and inertial navigation to get to a specified location. Our final deliverable will be a custom rover equipped with a system of hardware. We plan on using the Intel RealSense depth and tracking camera for our primary tool for navigation. We will also have a compatible web application used to stream our rover's cameras and if needed, ability to switch to manual control.

# Introduction

In the grand scheme of things, the main problem we are trying to solve revolves around the issue of space exploration and information gathering. Scientists need a way of exploring areas that are out of human reach. Rovers are a good solution to this problem. They are specifically designed as land vehicles to traverse the terrain of other planets and in the past, have been able to successfully collect information about the terrain, and to take crust samples such as dust, soil, rocks, and even liquids. Our senior design team is designing a Mars Rover specifically to solve the issue of autonomous navigation, with influence of guidelines issued by the University Rover Challenge.

Besides applications on Mars, autonomous navigation technology has many widespread applications. Many distribution centers, such as Amazon's warehouse, use autonomous robots to move heavy equipment around the warehouse, all the while avoiding collision with other robots. Companies like Tesla are also developing software to have vehicles drive autonomously in cities. This technology will revolutionize the way we work and travel.

The vehicle we are developing should be able to navigate the harsh terrain of Mars and avoid any obstacles in its way to the specified location. The natural terrain includes soft sandy areas, rough stony areas, rock and boulder fields, vertical drops and steep slopes. Terrain will range from flat in some areas to exceedingly difficult obstacles involving both small and large rocks. We will approach this problem using visual navigation. Our Intel RealSense, which will serve as our primary tool for navigation will be mounted onto the front of the rover. We have built a neural network to accurately detect the probability that our incoming path is blocked with an object or not. To model conditions similar to that of Mars, our neural network is trained with various objects to act as obstacles.

We will also have a web application to stream the video from the Rover's onboard cameras. Autonomous navigation is an unsolved problem. In unexpected situations, we should be able to switch into manual control to maneuver the rover out of upcoming problems.

# Concept Development

Unlike the traditional senior design concept of designing a product to solve a customer’s problem, our team has our orders given to us by the Mars Rover Society’s competition documents. The problem which we have to solve is one of the several challenges that comprise the competition. It centers around having a system that can autonomously navigate between worksites on Mars. From the beginning this challenge came with a detailed list of tasks that our rover will be expected to perform. They can be boiled down to the following:

1. Given GPS coordinate of post, autonomously navigate to within 3m of it
2. Given GPS coordinates of up to 5m from post, autonomously navigate to within that distance and follow an AR tag to the post
3. Given GPS coordinate, pass through gate autonomously
4. GPS coordinate 10m from gate, pass through a gate autonomously
5. GPS coordinate, navigate to next post while avoiding obstacle(s)
6. GPS coordinate 10m from gate, avoid obstacles and pass through gate

These tasks centered around having the rover navigate on its own via GPS, and some sort of camera system to read AR tags as well as move into a gate and around obstacles. From here we started our design and planning.

Having been given a specific set of requirements and expected technologies, our plan throughout the semester was made quite early and has since not diverged too significantly. We anticipated having a powerful processor onboard the rover that would be able to process image data from a camera and translate that to commands to two motors and a servo for movement.

On a conceptual basis, we planned on these tasks being performed in the following ways: An onboard camera would output data to a powerful onboard computer. The computer would also receive data from a GPS module. Using this data, the computer would use a Machine Learning algorithm to detect potential obstacles as well as other algorithms which would keep the rover on track on moving the desired GPS coordinate.

There are a few major ways in which our design has changed conceptually from the plan we first made. They centered mostly around the means by which the aforementioned goals would be achieved.

Firstly, we experienced quite some difficulty with using the onboard computer we first acquired, causing us to depart from the original plan in the hope of having some deliverable ready for the prototype test deadline. So, for this test, our control flow centered around a smaller onboard processor that took in data from an array of sensors positioned around the rover and made movement decisions based on this information. Although the system could avoid collisions and take turns quite well, we decided not to pursue it further and instead commit to our original control-flow concept. This was due to one: the fact that the hardware we used was on-loan and would be returned at the end of the semester, and two: the system would not be capable of delivering on our competition tasks due to its computational limits.

Next, we quickly came to realise that having one onboard computer would not be enough. Any onboard device that could easily communicate with our low-level speed control and turning control systems would not have the computational “manpower” necessary to receive and process the video feed from the camera - and conversely any computer capable of processing the video feed would not be able to easily communicate with the low-level controls. So, to get around this, our concept turned into a two-computer system. One would be the mediator between the speed/turning control and the more powerful computer. This concept required us to implement a communication protocol between two computers which would be simple. Simplicity was key in this new design because of how often the communication would be used.

The other change in our conceptual design came from watching various tutorials related to the hardware we acquired. We saw that in order to navigate around objects, our system would have to build a map of its surroundings and know its place within such a map. This would require having a two-camera system: one for depth and one for tracking. Our design at that time only allotted for the depth camera. Having realised this towards the end of the semester, we made the judgement call that it would be better to have some sort of system working with our current hardware rather than have new equipment arrive days before our demonstration. Therefore, we plan instead on acquiring a tracking camera during the next semester.

# System Description

The Mars Rover Autonomous Traversal Senior Design team needs to essentially deliver a complete robust solution that would potentially have been able to be submitted to the University Mars Rover challenge. The project entails delivering functional vehicle with a chassis and suspension and fully integrated sensors such as a tracking and depth camera system, a powerful computer and image processing unit, a global positioning system (GPS) and/or inertial measurement unit (IMU), and a second microcontroller capable of driving motors and controlling the rover’s movement. A significant portion of the deliverables will be autonomous path planning and control algorithms in the form of a convolutional neural network (CNN), and smart decision making with obstacle detection and collision avoidance.

Overall, the team provides a fully integrated vehicle as a proof of concept that can autonomously traverse terrain and reach a specified GPS location and detect an ARTag while using inertial and visual navigation to avoid obstacles.

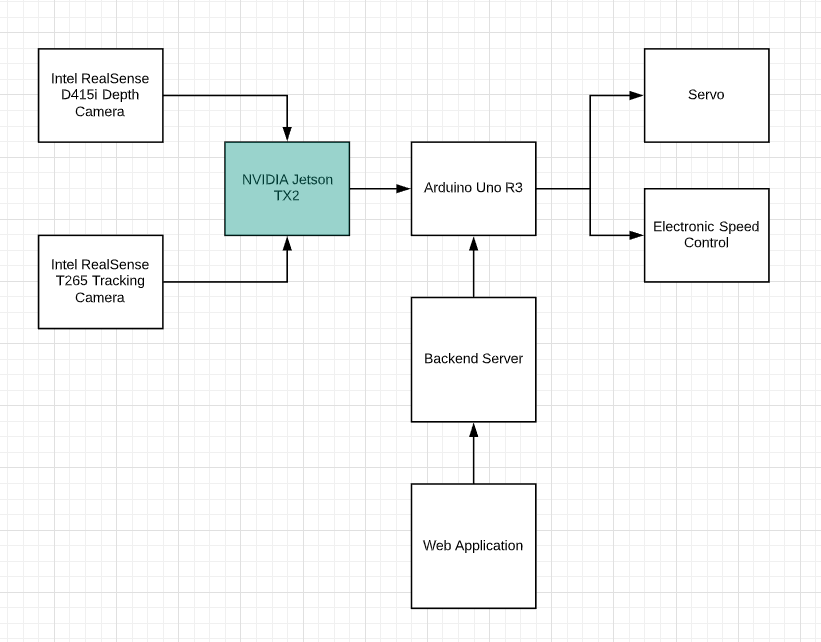
The rover is a Traxxas 1/8th 4WD Electric Power R/C Rock Crawler with front and rear stepper motors, an optical wheel speed encoder, an electronic speed controller (ESC), a differential steering servo for the front axis, and a 7.2V, 2000mAh LiPo battery for powering the rover. For the purpose of the first prototype demonstration, the rover included two TFMini - Micro LiDAR Modules and LIDAR lite v3 for distance sensing and ranging from the wall. Moving forward, we will not be using LIDAR in our final deliverable as camera data and GPS data are higher priority in the integration timeline for the purpose of our project.

The main computer that performs image processing and makes smart decisions for the Mars rover will be the Nvidia Jetson TX2. An Arduino Uno R3 acts as the microcontroller which drives the ESC and motors on the rover and interfaces with all other sensors for movement. One large challenge in the system design involved interfacing the Jetson TX2 with the Arduino Uno via UART protocol. Originally, the team planned on using the ESP32 Huzzah Feather as the microcontroller for controlling the rover however the Arduino was chosen for ease of programming and similar IO options to the ESP32. The majority of the team was more comfortable working with the Arduino IDE and had no difficulty interfacing with sensors and obtaining data readings and controlling the vehicle.

The cameras used come from the Intel Realsense family of cameras. Specifically, an Intel Realsense D415 Depth Camera and Intel RealSense T265 Tracking Camera are used in tandem to detect obstacles, perform simultaneous localization and mapping. and navigate towards an ARTag target. Image and video data taken from the cameras is used to train the neural network which determines if the rover’s path is currently blocked or unblocked. In unfamiliar environments the rover will use the trained CNN to make smart decisions on its movement by attempting to find unblocked paths and navigate around obstacle-blocked paths.

The GPS data will be sourced from an Adafruit FONA 808 Shield. This device has all of the capabilities of a cell phone, but we will only be using the GPS functionality (other functionality requires a SIM card and phone plan). It requires a small GPS antenna and its own 500mAh battery. Data will probably be sent from the shield to the Arduino Uno R3 and then transmitted to the Jetson TX2 via the serial connection.

Due to the nature of our project, our components will all have to rely on battery power. The Jetson TX2 is quite power hungry for an embedded system, and requires ~19 V input. The TX2 also will be powering the Intel Realsense cameras, the Arduino Uno R3, and a USB hub. The rover we are using has a battery connected to the stepper motors and ESC. This battery may not be able to hold enough charge or output enough current for our desired configuration, thus requiring a more intense battery.



*Figure 1: System Block Diagram for controlling the rover*

# First Semester Progress

During the first few weeks of our semester, our team was doing extensive research to familiarize ourselves with the problem statement: What methods are there for autonomous navigation? We found many different methods addressing this problem, including using Sonar, LIDAR and SLAM. We eventually chose a visual navigation approach because of the high amount of information we receive from cameras as opposed to other sensors. The one disadvantage of this is that computer vision applications are very computationally expensive. We would need to purchase hardware capable of high performance on-board computing.

We had to choose our system of hardware for this project, including cameras, microcontrollers and the type of rover to work with. To do this, we did many quantitative analyses for different cameras and microcontrollers. We considered the trade-offs of the Jetson Nano, Raspberry Pi, Arduino Uno and the Jetson TX2. For cameras, we considered Intel RealSense cameras and also many popular Logitech cameras. We ended up picking the Jetson TX2 after considering its performance power and advice we acquired from Shark Tank. We also picked the Intel RealSense camera because of its built in depth and tracking utilities that would be very useful for our specific needs.

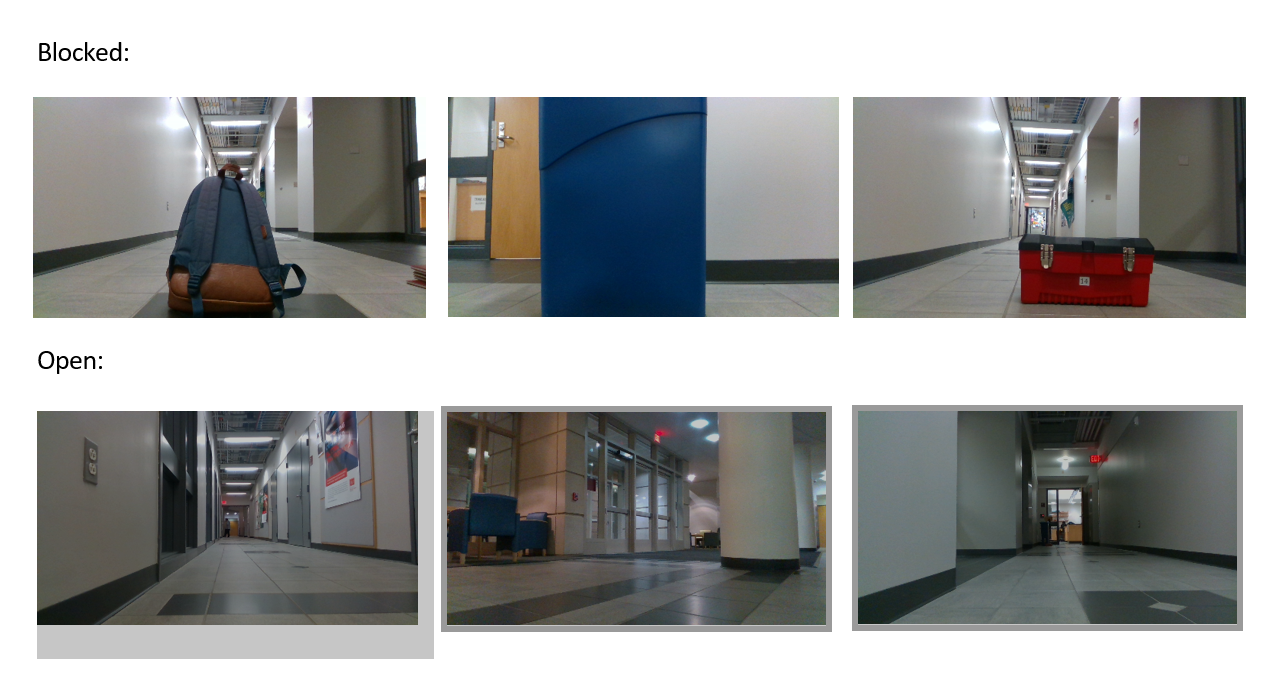
Getting all these components to work together has been difficult for us. For the first prototype testing, we navigated primarily with sensors controlled by the ESP32 microcontroller. We were able to successfully program the 1/8th 4WD Electric Power R/C Rock Crawler with a system of hardware including the ESP32 microcontroller, TFMini MicroLidar, MaxSonar Ultrasonic Range Finder and the LIDAR Lite V3. We were able to drive the rover with basic controls and stop within 20cm from a detected wall. Our TFMini Micro Lidars were mounted on the left side of our rover and served as the basis of the steering logic.



*Figure 2: Overview of Steering Logic*

Currently, the progress our team has made involves being able to interweave all of our hardware components together, which was a problem during our first prototype. We are able to run a python script on our Jetson TX2 and have it communicate with the arduino connected to the rover. We are also able to utilize our Intel RealSense D415 camera and process the data it receives into meaningful information which we can output to the arduino.

Our team was also able to successfully train a neural network to detect whether the rover was facing a *blocked* or *open* path using 200 training images collected around the first floro of photonics. We have two classes: *blocked* and *open*. Blocked refers to when something is in front of the rover while open refers to when the rover sees a free path that it is able to move towards. Figure 3 provides some examples.



*Figure 3: Sample of six images out of two hundred that is used for training data for our neural network*

To give the rover a good variety of data to learn from, we used a variety of backgrounds and objects around photonics.

Training an accurate machine learning model takes millions of training images. To avoid this, we used a method called Transfer Learning that allows users to repurpose a pre-trained model to perform a new task with much less data available. In this case, we used Alex-Net, which was trained on millions of images. To repurpose this model, we replace the final layer of the neural network with a new untrained layer that has only our two outputs: *blocked* or *open*.

The results were very good, averaging around 90% - 98% accuracy on detecting *Blocked* and *Open* paths. Our next steps involve using this model to interpret live data from our Intel RealSense cameras and send a string output to our Arduino, which will move accordingly.

Because autonomous navigation is not a solved problem, there may be certain scenarios where our rover will reach a destination and not be able to navigate out of that. Because of this, we are also in the process of developing a web application to monitor the live video feed provided from the cameras mounted onto the rover. We are also developing a method for manual control from our computers to the arduino controlling the rover.

# Technical Plan

The performance period of this plan is December 5, 2019 - April , 2020. Gantt Chart for each task is shown in Appendix 2.

Task 1. Object and Obstacle Detection

*By using the Intel Realsense D415 Depth Camera, the rover should be able to detect obstacles obstructing its path. Our team captured 200 images to train our neural network to learn whether there is an obstacle in front of the rover or not.*

Task 2. Object and Obstacle Avoidance

*By having object and obstacle detection done, we should be able to implement it to our crawler so that it can change direction when it faces an obstacle. The direction control of our crawler will be from Jetson TX2, which communicates with Arduino. This task will also help the crawler decide on which direction to steer.*

Task 3. GPS Navigation

*GPS Navigation is a main component for autonomous navigation. Our crawler will be moving according to a given GPS coordinates. Using inertial measurement unit(IMU) will be useful to navigate between given coordinates.*

Task 4. Route Planning

*With the use of Intel Tracking Camera, the crawler will be able to keep track of the map and planning a route to the destination. For this specific task, the crawler needs a route planning algorithm, such as Dijkstra’s single source shortest path algorithm, that plans the path. This task also involves finding a geometric path from an initial configuration to a given configuration so that each configuration and state on the path is feasible.*

Task 5. AR Tag Detection

*This task should enable us to detect AR tags using our intel realsense camera. One of the good resources for AR tag detection would be OpenCV and ROS package named ar\_track\_alvaar.*

Task 6. Integrate All Subsystems

*This task is the most important part of our project. We will be combining all the previous tasks mentioned above. It may be the easiest yet hardest part for our group. We have to figure out how we are going to implement two cameras onto TX2 and have them both communicate with each other to control the crawler. We are using a lot of resources, and combining all of those resources may cause some troubles. That is why we are planning on taking some time to finish this task.*

Task 7. Test Launch Outside

*In order to test our crawler that can sustain an environment similar to Mars, we need to test our project outside rather than inside the building. Since we have been testing our crawler inside the building, we assume that testing and debugging would take some time.*

Task 8. Further Testing and Debugging

*By having our major tasks done, it gives us more time to do further testing and debugging for our project. This time period will be used to improve our project that we have done so far.*

# Budget Estimate

|  |  |  |
| --- | --- | --- |
| **Item** | **Description** | **Cost** |
| 1 | Intel RealSense T265 Tracking Camera | $199 |
| 2 | Intel RealSense D415 Depth Camera\*\* | $149 |
| 3 | Arduino Uno R3 set\*\* | $35 |
| 4 | Adafruit FONA 808 Shield - Mini Cellular GSM + GPS for Arduino\*\* | $49.95 |
| 5 | Passive GPS Antenna uFL - 15mm x 15mm 1 dBi gain | $3.95 |
| 6 | Lithium Ion Polymer Battery - 3.7v 500mAh | $7.95 |
| 7 | Arduino Ethernet Shield | $23.65 |
| 8 | NVIDIA Jetson TX2 Developer Kit (Education Discount) | $330.09 |
| 9 | Servo | $21.47 |
| 10 | Battery (for drivetrain) | $22.95 |
|  | Total Cost (before tax) | $843.01 |

\*\*Provided via Donation

A green background indicates that the item has yet to be purchased

The dominating figure in our budget is the NVIDIA Jetson TX2. As mentioned in other sections, we found this device to be necessary over computers because of its ability to process video data and make movement decisions in real time. We received a $100 Education Discount from NVIDIA for this item, bringing it down from the original pre-tax pre-shipping price of $400.

The other dominating figures on this are the Tracking and Depth Cameras. Thankfully we were donated the Depth camera by our customer, Professor Alshaykh. The Tracking camera will likely be a necessary addition and we plan on purchasing it in time to work with it during the Spring semester.

# Attachments

# Appendix 1 – Engineering Requirements

Team #28 Team Name: Mars Rover

Project Name: Mars Rover

|  |  |
| --- | --- |
| **Requirement** | **Value, range, tolerance, units** |
| Dimensions | 1.2m x 1.2m |
| Weight | 50 kg |
| Environment | Withstand heat of up to 100°F, dust, and light rain |
| Navigation via GPS to post | Must move to within 3m radius of post |
| Navigation via AR to post | Must move from 5m away from post, to the post |
| Navigate between posts | Must move from 10m away to gate (consisting of 2 posts 2m apart) and pass through it |
| Navigate around obstacles | Must be able to move from post A to B, moving around rocks in the way |

# Appendix 2 – Gantt Chart

# 

# Appendix 3 – Team Information Sheet

**History of Team and Company**

Team 28 assembled over the summer of 2019. We anticipated on proposing a new and brilliant design concept to work on for the year. After much deliberation we decided on making a water-traversing, bird-hazing hovercraft. Unfortunately, this idea did not come to fruition and we were instead offered a chance to work on the Mars Rover project.

The original concept for the Mars Rover project was for us to build part of the system to participate in the annual Mars Rover Society University competition. Our design would be fundamental to tackling the Autonomous Traversal Challenge set out by the Mars Rover Society. As of now, we are unclear as to whether we will enter the competition this year, as the club is in its first year and there are many dependencies on other teams and club members.

**Team Member Information**

**Daniel Crawley**

[crawley@bu.edu](mailto:crawley@bu.edu)

Daniel was born in Boston and has since not ventured too far, attending high school a stone’s throw away from Boston University. He is currently pursuing a BS in Computer Engineering with an Electrical Engineering minor. His current academic interests include High Performance Computing and Embedded Systems. One of the few interesting facts about him is that he is a citizen of three different countries, and may give you a penny if you can guess all three.

**Brian He**

[brianhe@bu.edu](mailto:brianhe@bu.edu)

Brian was born in New York. One interesting fact about him is that in highschool, he was on his school's Table Tennis and Swimming team. He does not exercise anymore because he is too busy studying Computer Engineering at Boston University. His favorite food is sushi and he loves traveling to different places.

**Tommy Lam**

[tlam11@bu.edu](mailto:tlam11@bu.edu)

Tommy was born in Minnesota and moved to Boston when he was 8. Some time has since passed and he is now pursuing a BS in Computer Engineering with a minor in Computer Science at Boston University. He is looking for a full time employment opportunity in software engineering. A fun fact about him is that he once gained 30 pounds over one summer.

**Kelly Lee**

[sylee538@bu.edu](mailto:sylee538@bu.edu)

Kelly is from South Korea, and she came to the United States for school. She’s been in the States for 10 years, and now she is currently pursuing a BS in Computer Engineering at Boston University. She is currently looking for a software engineering/computer engineering job in Boston. One fun fact about Kelly is that she came to America alone when she was 12 years old.

**Linden Vo**

[lindenvo@bu.edu](mailto:lindenvo@bu.edu)

Linden, hailing from Randolph, Massachusetts, is pursuing a BS in Electrical Engineering. He is seeking full time positions in digital hardware and software design. He enjoys working with FPGAs and processors and looks to further hone his skills in Verilog, VHDL, and System Verilog for system design, test, and verification. Linden loves playing games such as League of Legends and was captain of his high school varsity volleyball team. He is also the president of BU’s chapter of the Society of Asian Scientists & Engineers and actively involved with BU’s Vietnamese Student Association.