Implementing a Conceptual Teleoperation Communication System for Remote Controlled Vehicles

Matthew H. Taylor

University of Illinois at Urbana-Champaign, Department of Aerospace Engineering

Table of Contents

1. Overview

John Deere recently donated an R-Gator Vehicle to the University of Illinois at Urbana-Champaign. For this research project, we are attempting to upgrade it and turn it into a full fledge autonomous vehicle. My goal is to develop a method to control acceleration and steering for our autonomous vehicle using open-source libraries in python. The code written from this semester will be used as a starting point for the ultimate software stack for R-Gator's remote-control system. Two micro-computers will be used for this project (one on the ground station and one on the vehicle) to accomplish these goals.

2. Technical Objectives

This research project is about conceptual development of a teleoperation communication system for autonomous vehicles. To accomplish this overarching goal, three objectives will need to be completed: Create a communication network using the MQTT protocol, create a graphical user interface for the acceleration and steering of the vehicle, and testing.

2.1 Creating a Communication Network with MQTT

The MQTT protocol is an application of Internet of Things that allows for fast wireless communication between devices. It uses a publish/subscribe pattern where a client device can publish information to a broker (central hub for data) [1]. Another client subscribes to this information and immediately receives it from the broker when connected to the internet. For out application, there will be one Raspberry Pi client publishing steering and acceleration commands at the ground station and another Raspberry Pi onboard the vehicle that subscribes to these messages to control the vehicle. Much of this semester will be spent researching and implementing this protocol, as there is so much information on the topic.

2.2 Creating a Graphical User Interface

A graphical user interface (GUI) will be created as a conceptual way to see how the vehicle will respond to MQTT publish commands. Included in the GUI will be a moveable slider indication the vehicle’s acceleration and a knob representing the steering of the vehicle. The GUI will be present on both micro-computers and changing the ground station’s steering and acceleration should change the vehicle micro-computer’s steering and acceleration in the same manner.

2.3 Validation and Verification

Testing will be done throughout the semester, but a large emphasis of testing will be during the last month of school. This will be done in python using industry standards in order to ensure the success of the project. This will include but not be limited to analyzing the output of the vehicle’s GUI to see how quickly and accurate it responds to its inputs.

3. Work Plan

Below are visual representations of the tasks at hand along with soft deadlines and further explanation of what each task means.

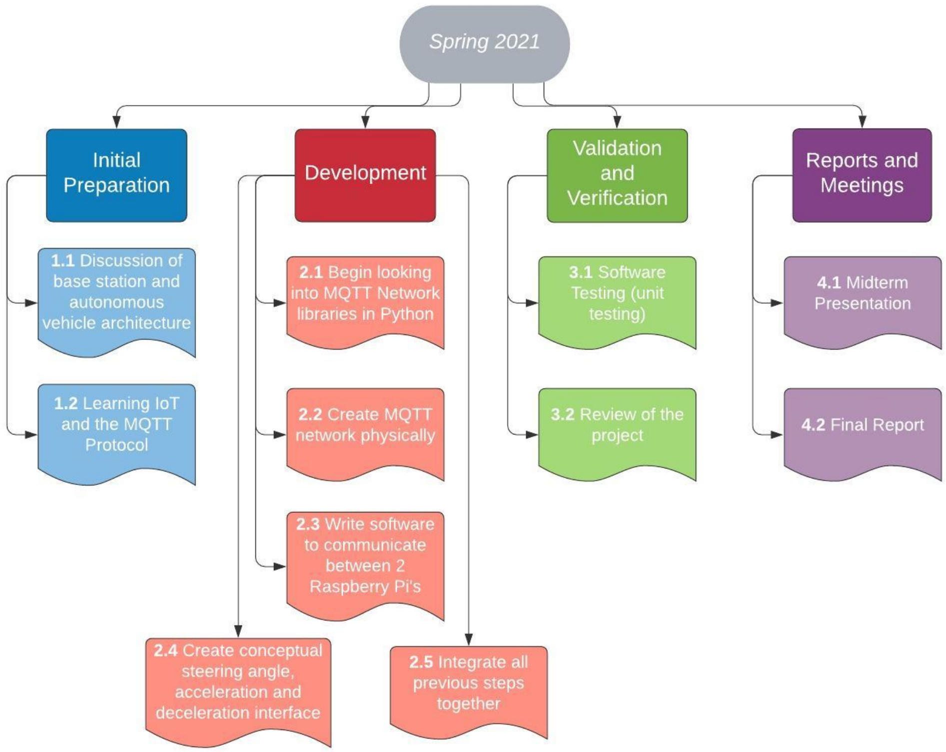

Figure 1. Block Diagram for Semester Goals

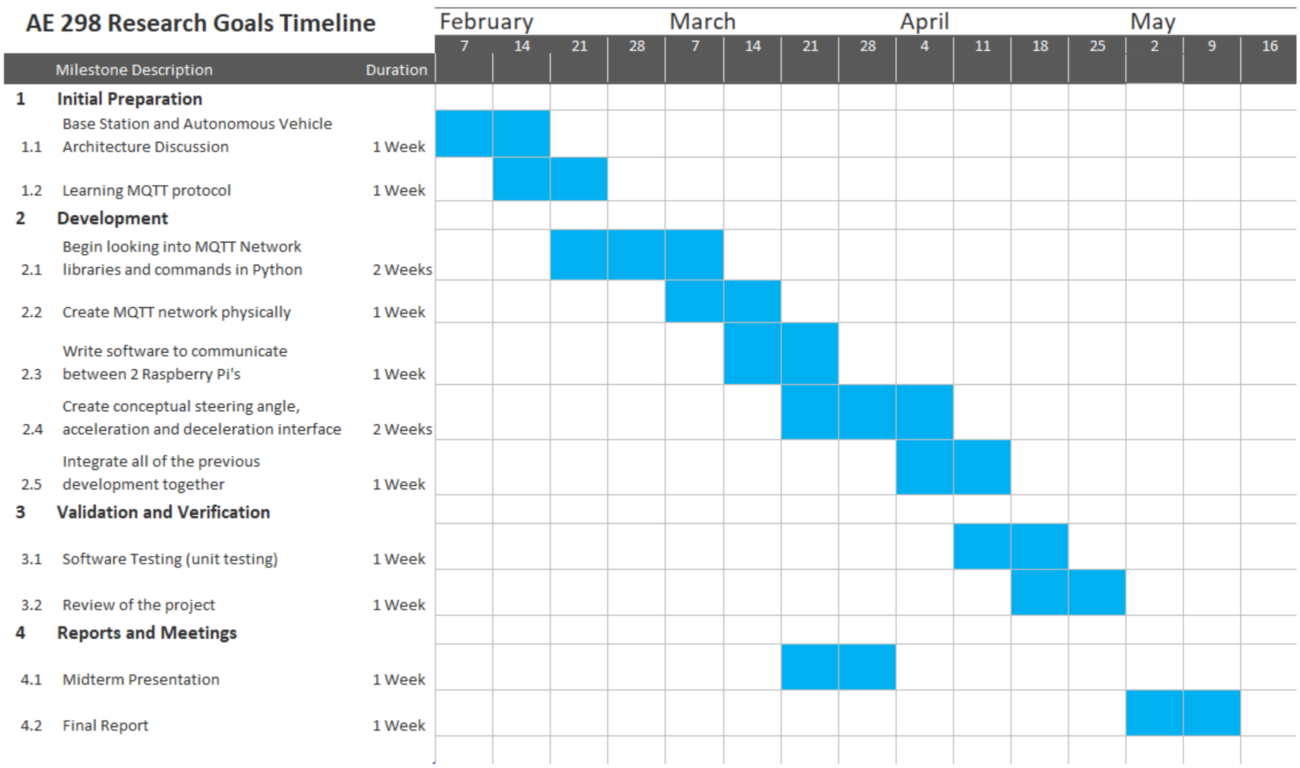

Figure 2. Gantt Chart for Semester Goals

3.1 Initial Preparation

3.1.1 Discussion of Base Station and Autonomous Vehicle Architecture

To begin, there will be a discussion of how the communication system will be set up. This will take a few hours. Understanding this is the key to completing the rest of the project successfully.

3.1.2 Learning the MQTT Protocol

Additional time will be taken to learn what the MQTT protocol is conceptually and its application to the project. Important notice will be taken to the architecture of this protocol, including the pub/sub packets and their contents. This should take about 6-8 hours to complete.

3.2 Development

3.2.1 Begin Looking into MQTT Network Libraries in Python

After an understanding what MQTT is, I will then begin looking at the open-source libraries available in python. Some very commonly used libraries are Eclipse's Paho and Mosquito. Both will be utilized to work with the MQTT interface. This will take 4-8 hours to complete.

3.2.2 Create Physical MQTT Network

After parts are ordered, a physical MQTT network will be created to simulate the ground station and what will be on the vehicle. Additionally, a local broker will be created with Mosquito on the ground station to communicate between the two Raspberry Pi's. This should take 3-5 hours to complete.

3.2.3 Write Software for Telecommunication between Raspberry Pi's

The publisher and subscriber code will be created to handle steering and acceleration commands. The ground station will be publishing packets of this data, while the R-Gator's Pi will receive this information and decode it. This should take anywhere from 3-6 hours to complete.

3.2.4 Create conceptual steering angle, acceleration, and deceleration interface

A GUI will be created to handle steering angle, acceleration, and deceleration. See section 2.2 for more information. This task should take 2 weeks and will be a time commitment of anywhere from 6-8 hours.

3.2.5 Integration of Development

The GUI along with the MQTT code will be integrated together so that an input on the ground station side will have an identical output on the R-Gator vehicle's GUI. This task should take anywhere from 2-3 hours.

3.3 Validation and Verification

3.3.1 Software Testing

Unit testing will be done using industry standards in python. See section 2.3 for more information. This task should take anywhere from 3-6 hours.

3.3.2 Review of Project

After testing is done, there will be a review of the project, what was accomplished, what went well, and what could be improved. This will just be a meeting with my mentor, Konur and shouldn't take more than an hour.

3.4 Reports and Meetings

3.4.1 Midterm Presentation

A presentation of the project's current progress will be completed halfway through the semester. Deliverables should include the GUI as well as helpful pictures of what the MQTT protocol is doing in our application. This should take 1 hour to complete, not including presenting time.

3.4.2 Final Report

A final report will be done to wrap up the semester in AIAA format. Included will be an overview of the project, what was accomplished, setbacks along the way, and an explanation of how I responded to these setbacks.

4. Facilities and Equipment

No facilities will be needed during this semester, however there may be a trip to the AUVSL lab to see the R-Gator Vehicle. There will be some equipment needed for this research, and a parts list is below. The total will come out to be close to $120.

Part	Price ($)	Quantity	Total Price ($)
Raspberry Pi 3B+	41.99	2	83.98
HDMI Cable	7.69	1	7.69
RPI Power Supply	9.99	1	9.99
32 GB Micro SD Card	6.99	2	13.99

*Note the Micro SD Card link is already for a 2 pack bundle

Total: $115.65

5. Potential Post Applications

At the end of this semester, I will have made a GUI that has a slider for acceleration and deceleration, as well as a knob that simulates the steering wheel of the R-Gator. The Raspberry Pi that is onboard the vehicle should be able to output the inputs of the slider and knob sent from the ground station. This graphical interface, however is all a simulation and won't physically control the vehicle. This is where the vehicle's controller area network (CAN) bus system comes in. CAN bus allows for any electronic control unit to communicate to the entire system in a vehicle without causing an overload to the controller computer. In the future, these slider and knob inputs will be output into the CAN bus system in the vehicle to physically control acceleration and steering.

As the current design stands, there will be a ground station with a controller that communicates to the R-Gator Vehicle over the same Wi-Fi network. From the perspective of the communication system, the MQTT protocol sees no difference between a Wi-Fi network and for example, a 4G LTE network. Connecting to a 4G LTE network would allow for long range communication between the vehicle and ground station (i.e. the R-Gator Vehicle could be controlled from anywhere in the world). There are also many applications for MQTT outside of use for remote control systems. Facebook messenger, for example, uses this protocol for low latency communication between users around the world. MQTT is a very lightweight, efficient protocol, and has encryption methods available that can keep user's information safe and secure. This protocol is a subset of internet of things, or IoT. IoT easily links data between devices in a scalable way, simplifying connections between multiple devices [1]. Hologram is a company that sells sim cards for IoT devices, allowing them to connect to a global network. There are endless opportunities with the combination of using a Hologram sim card along with any IoT device.

6. Key Personnel and Bibliography of Directly Related Work

6.1 Key Personnel

Our research team has excellent experience and will be able to accomplish the goals outlined in this proposal. Konur is a great mentor and I have confidence that he will always be there to help when I need help on the project. He has a working knowledge of IoT and the MQTT protocol and is great at explaining his knowledge on the subject. For me personally, I do not know everything there is to know about the conceptual teleoperation communication system but will research and work hard to become an expert on the matter.

Konur Yigit (Mentor)

Konur is my mentor for this project and a graduate student in the department of Aerospace Engineering at the University of Illinois. He received his M.Sc. degree in Marine Engineering from MIT in Cambridge, Massachusetts, and a B.Sc. degree in Mechanical Engineering from TR Naval Academy in Istanbul, Turkey. Konur worked for 3 years as a naval officer and has 9 years of experience in research and development in the defense industry under his belt. He is currently employed at John Deere, working on various automation projects. His main expertise is in the field of controls and path planning.

Matt Taylor (AE 298 Student)

I am a sophomore in Aerospace Engineering with a minor in Computer Science at the university of Illinois. I have ample experiences with different programming languages including Java, C, C++, and Python. I am currently involved in a registered student organization called IREC (Intercollegiate Rocket Engineering Competition), where I developed a GPS driver and am continuing to refine the embedded systems for the avionics system on our rocket. For this project, I will be implementing a conceptual communication system using the MQTT protocol with open-source libraries in Python.

Noah Jon (AE 298 Student)

Noah is a sophomore in Aerospace Engineering at the University of Illinois. He is very knowledgeable and has great leadership and development skills. In high school he worked on an autonomous vehicle project and will carry that experience to this research. For this project, he is researching the hardware that will be used in the autonomous systems, including lidar, GPS, and more. We will indirectly be working together, but he will not be working on the software.

William Robert Norris (Primary Investigator)

Dr. Norris is the lab director for the autonomous and unmanned vehicle systems laboratory (AUVSL). He has 20 years of unmanned system technology, business development, marketing, and program management experience, and is well-qualified in developing opportunities for the introduction and commercialization of new products.

6.2 References

[1] Liu, Xiangtao, et al. “The Method of Internet of Things Access and Network Communication Based on MQTT.” Computer Communications, vol. 153, 1 Mar. 2020, pp. 169–176.