Modeling and Control of Tracked Mobile Robot

Iann Akoulouze Bika
Intro to Robotics EE-4953
Electrical and Computer Engineering Department
The University of Texas at San Antonio
San Antonio, Texas
ianngeorges@gmail.com

***Abstract*—This article serves as preliminary report on the initial construction and assembly of the robot. The problems that have accrued through the installations of software necessary to operate the robot are considered.**

***Keywords*—ROS, Ubuntu, SSH, Python, Adeept.com, Virtual Machine, Mobile Robot, Tracked Robot**

I. Introduction

The field of robotics is constantly evolving to meet the vast needs and requirements of our modern world. In the endless range of applications robotics serves, our research will focus primarily on tracked vehicles. Tracked vehicles are specifically aimed to tackle tasks in a variety of terrains, such as mining, logging, agriculture and military purposes. This paper will provide a strong baseline and foundation which can later be used to develop robots capable of operating in these niche industrial fields.

This article is organized as followed:

Section II will first cover the initializing the proper environment for our robot's microcomputer in order to later configure tasks and assignments for the robot to complete. Section II will go into depth and cover the issues encountered through this process and how we've overcame these said issues.

II. Configuring ROS

A. Ubuntu

Considering our robot will need a system hierarchy to run and distribute various scripts and programs during its operation, we choose to boot Ubuntu 18.04 for its ROS compatibility and the Operating System's modularity. By using Raspberry Pi Imager, we were able to boot Ubuntu onto the SD the card without any issues.

B. SSH

By far this next step in our configuration for the ROS was the most tedious part of the build. Once we we're able to boot and start Ubuntu on the Raspberry Pi, we connected to Wi-Fi and attempted to gain remote access to the RPI through SSH. Although I undermined the importance of gaining remote access to the computer, I soon realized for robotic applications this is a very significant feature. Considering we may not have access to the robot's internal computer once it is built, SSH is vital to the robot's operation as it might need different programs or may need an update from time to time. Furthermore, if the robot is located in a remote or isolated location, remote access is vital to any robotic operation. At first, we we're unable to connect by SSH this was very worry some. Fortunately, after realizing there was an issue with the encryption properties of our SSH file, upon reinstallation we were able to gain remote access to the microcomputer.

C. Adeept Proprietary Software

This is the phase I am currently working on. Once I got the SSH to work I then promptly started installing various software provided by Adeept to set up the robot. First, we had to install git on Ubuntu in order to clone their git repository. Next, we had to run a python script to initialize the robot. At first, I had to install python to run the script and then I realized I also need pip to install other python packages. I made the mistake of installing python and pip to my root directory. I have been unsuccessful with running this script because my pip directory is messed up. In hindsight I should have install all the software in a folder so that it'd be easy to delete the folder and back track our step if an issue were to arise. I am currently reinstalling setting up Ubuntu so that I can properly configure my directory and file system properly this time around. I will keep documenting the process and make note of the issues I have resolved along the way. I plan on building the robot this weekend once I am able to resolve my issues with the installation of Adeept's software.