



---

# Laboratory Notebook

Senior Project

---

Eric Jones and Dakota Adra

emjones@mail.bradley.edu  
dadra@mail.bradley.edu

Beginning August 30, 2018



## Contents

|                                    |           |
|------------------------------------|-----------|
| <b>Thursday, August 30 2018</b>    | <b>9</b>  |
| <b>Tuesday, September 4 2018</b>   | <b>11</b> |
| <b>Thursday, September 6 2018</b>  | <b>13</b> |
| <b>Friday, September 7 2018</b>    | <b>15</b> |
| <b>Tuesday, September 11 2018</b>  | <b>17</b> |
| <b>Thursday, September 13 2018</b> | <b>19</b> |
| <b>Friday, September 14 2018</b>   | <b>21</b> |
| <b>Tuesday, September 18 2018</b>  | <b>23</b> |
| <b>Thursday, September 20 2018</b> | <b>25</b> |
| <b>Friday, September 21 2018</b>   | <b>27</b> |
| <b>Tuesday, September 25 2018</b>  | <b>29</b> |
| <b>Thursday, September 27 2018</b> | <b>31</b> |
| <b>Tuesday, October 2 2018</b>     | <b>33</b> |
| <b>Thursday, October 4 2018</b>    | <b>35</b> |
| <b>Friday, October 5 2018</b>      | <b>37</b> |
| <b>Wednesday, October 10 2018</b>  | <b>39</b> |
| <b>Friday, October 12 2018</b>     | <b>41</b> |
| <b>Tuesday, October 16 2018</b>    | <b>43</b> |
| <b>Thursday, October 18 2018</b>   | <b>45</b> |
| <b>Tuesday, October 23 2018</b>    | <b>47</b> |

|                                  |           |
|----------------------------------|-----------|
| <b>Thursday, October 25 2018</b> | <b>49</b> |
| <b>Friday, October 26 2018</b>   | <b>51</b> |
| <b>Tuesday, October 30 2018</b>  | <b>53</b> |
| <b>Thursday, November 1 2018</b> | <b>55</b> |
| <b>Friday, November 2 2018</b>   | <b>57</b> |
| <b>Tuesday, November 6 2018</b>  | <b>59</b> |
| <b>Friday, November 9 2018</b>   | <b>61</b> |
| <b>Tuesday, November 13</b>      | <b>63</b> |
| <b>Thursday, November 15</b>     | <b>65</b> |
| <b>Friday, November 16</b>       | <b>67</b> |
| <b>Tuesday, November 20</b>      | <b>69</b> |
| <b>Thursday, November 22</b>     | <b>71</b> |
| <b>Friday, November 23</b>       | <b>73</b> |
| <b>Tuesday, November 27</b>      | <b>75</b> |
| <b>Thursday, November 29</b>     | <b>77</b> |
| <b>Friday, November 30</b>       | <b>79</b> |
| <b>Wednesday, December 12</b>    | <b>81</b> |
| <b>Thursday, December 13</b>     | <b>83</b> |
| <b>Friday, December 14</b>       | <b>85</b> |
| <b>Monday, December 17</b>       | <b>87</b> |
| <b>Tuesday, December 18</b>      | <b>89</b> |
| <b>Wednesday, December 19</b>    | <b>91</b> |
| <b>Thursday, December 20</b>     | <b>93</b> |

|                              |            |
|------------------------------|------------|
| <b>Friday, December 21</b>   | <b>95</b>  |
| <b>Friday, January 4</b>     | <b>97</b>  |
| <b>Monday, January 7</b>     | <b>99</b>  |
| <b>Tuesday, January 8</b>    | <b>101</b> |
| <b>Wednesday, January 9</b>  | <b>103</b> |
| <b>Thursday, January 10</b>  | <b>105</b> |
| <b>Friday, January 11</b>    | <b>107</b> |
| <b>Monday, January 14</b>    | <b>109</b> |
| <b>Tuesday, January 15</b>   | <b>111</b> |
| <b>Wednesday, January 16</b> | <b>113</b> |
| <b>Thursday, January 17</b>  | <b>115</b> |
| <b>Friday, January 18</b>    | <b>117</b> |
| <b>Monday, January 21</b>    | <b>119</b> |
| <b>Tuesday, January 22</b>   | <b>121</b> |
| <b>Wednesday, January 23</b> | <b>123</b> |
| <b>Thursday, January 24</b>  | <b>125</b> |
| <b>Monday, January 28</b>    | <b>127</b> |
| <b>Tuesday, January 29</b>   | <b>129</b> |
| <b>Wednesday, January 30</b> | <b>131</b> |
| <b>Thursday, January 31</b>  | <b>133</b> |
| <b>Friday, February 1</b>    | <b>135</b> |
| <b>Friday, February 8</b>    | <b>137</b> |
| <b>Monday, February 11</b>   | <b>139</b> |

Contents

---

|                               |            |
|-------------------------------|------------|
| <b>Tuesday, February 12</b>   | <b>141</b> |
| <b>Wednesday, February 13</b> | <b>143</b> |
| <b>Thursday, February 14</b>  | <b>145</b> |
| <b>Friday, February 15</b>    | <b>147</b> |
| <b>Monday, February 18</b>    | <b>149</b> |
| <b>Tuesday, February 19</b>   | <b>151</b> |
| <b>Thursday, February 21</b>  | <b>153</b> |
| <b>Friday, February 22</b>    | <b>155</b> |
| <b>Monday, February 25</b>    | <b>157</b> |
| <b>Monday, March 4</b>        | <b>159</b> |
| <b>Wednesday, March 6</b>     | <b>161</b> |
| <b>Thursday, March 7</b>      | <b>163</b> |
| <b>Friday, March 8</b>        | <b>165</b> |
| <b>Monday, March 11</b>       | <b>167</b> |
| <b>Wednesday, March 13</b>    | <b>169</b> |
| <b>Thursday, March 14</b>     | <b>171</b> |
| <b>Friday, March 15</b>       | <b>173</b> |
| <b>Monday, March 25</b>       | <b>175</b> |
| <b>Wednesday, March 27</b>    | <b>177</b> |
| <b>Thursday, March 28</b>     | <b>179</b> |
| <b>Monday, April 1</b>        | <b>181</b> |







## Thursday, August 30 2018

Did some testing with the BBBlue and BBBlack. I wanted to see whether or not they would be able to connect to the router in the Robotics lab, and if that connection would allow me to set their ROSIP's correctly. This did indeed work. Also, I believe that the IPs given to the 2 robots could be static. This could allow us to more easily automate the setup process.

Checked Mutex locking and unlocking in order to minimize stolen CPU time. This was done separate from the ROS C++ code. It was determined that calculations should be done outside of the mutex locks and stored into temporary variables. This way the shared memory variables could simply be set equal to the temporary variables inside of the mutex lock. This will minimize the time that the code is operating under mutex and will prevent any unnecessary waits for resources.



## Tuesday, September 4 2018

Arrived in the lab at 9:14. Began to set up for a demo for Dr. Miah regarding our progress in ROS. We found some strange errors regarding the BBBlue's connecting to the network. For now it seems that the router assigns a static IP to the robots. This will be very helpful if we wish to make the process much more autonomous.

Further, it seemed unable to boot of the flash. When we were first setting it up today we did bump the SD card, so maybe that was the issue. After making sure the Blue booted off the SD card ONLY, it started up and we were able to gain access. This is not the first time this has happened, though we hope that it is simply due to us nudging the SD card, and not a much more pressing issue.

Today I tested the ROS code we were unable to demo to Dr. Miah during the meeting two weeks ago. We were supposed to meet Dr. Miah down in the Robotics Lab but he did not show up. As the lab time was coming to a close, we decided to just take a video of it's operation and submit that onto the drive. During the testing we noticed that the Blue had significant problems with latency. The robot was supposed to go 0.15 [m] but ended up going app. 0.25 [m]. As we ended Tuesday's lab session, we decided to spend our home research time looking into ways to combat this.

Created Github accounts and uploaded the current working code. Currently we have 3 working branches: master, ros, and multithread. Once the feature branches are complete they will be merged with the master. You can find the Github repository here: <https://github.com/tepidbread/SeniorProject2018-2019>



## Thursday, September 6 2018

Changed every single variable in the multithreading feature branch to comply with camel-Case convention. This change will later be reflected in the master branch as well.

Attempted to multithread dead reckoning code and failed. Received 100+ lines of compiler errors. The issue likely lies within the makefile. The make file may need to be rewritten from scratch to solve these errors. Although, the the first suspect seems to be the compiler options in the makefile.

Today I worked on some extra ROS functionality that I would be useful in making the process of "setting up" much faster. In that I wanted to edit the machines `/.bashrc` file (a bash script that runs on sourcing) to have it export the `ROS_IPs` and `ROS_MASTER_URIs`. This way all I would have to do is include the sourcing of the `setup.bash` files for base ROS and the package specific sourcing ONLY. I could probably include those in the bash file as well, but I am concerned about the impacts that would have on the catkin building. I did not get a definitive answer today, but I have edited both files.

I also want to do some more testing with a bridged network in my VM. If I cannot setup my laptop to run in the ROS system in this way, I will have to dual-boot linux (as I am more familiar troubleshooting network problems in this OS)

Itinerary for Friday September 7 meeting:

- What is going on with Overleaf and ShareLaTeX?
- Using Trello? Or similar service?
- Senior Project webpage
- github



## Friday, September 7 2018

Met with Dr. Miah:

He CAN see the changes we use in ShareLaTeX and Overleaf

Use the google calendar instead of Trello - unless you wish to.

The website can wait for now, should be fine.

Use Dr. Miah's papers as a ref, ADD THE .BIB TO THE PAPER.

CHANGE PASSWORD FOR THE ECE227 EMAIL

USE THE NETWORK LAYER STRUCTURE TO DEFINE MAFOSS (?)

Data Link, Application, etc... High Level as Possible

Write paper as homework by October 15th

Setup data logger when operating with ROS and log data

Download "Solving Optimal Control Problems With State Constraints Using Nonlinear Programming and Simulation Tools" by Victor M. Becerra IEEE Transactions ON Education Vol 47 on 3 august 2004 (use as example)

[http://robots.mobilerobots.com/docs/all\\_docs/P30pMan6\\_5.pdf](http://robots.mobilerobots.com/docs/all_docs/P30pMan6_5.pdf)

Figures:

- Table
- software architecture (lower level block diagram)
- high level model with actual pictures

Look at the Pioneer and Khepra Hardware Interfacing





## Tuesday, September 11 2018

Arrived in the lab around 9:30. Discovered that the formerly assumed makefile error was simply a line of misplaced code before the include statements at the top of the code. Somehow it must have been copied to the top of the code without us noticing. However, some strange errors are still occurring such as main being reported as not having a return type when it clearly does.

Ran the ROS system on the three MCUs, but am still having trouble with running ROS over my Virtual Machine. More specifically, I am getting an Address Family error. It seems that even though I am connected to the same local network as the BeagleBones, the IP address that I am using for my ROS\_MASTER\_URI is incompatible. This will be solved simply by dual booting a Linux Operating system on the laptop and running the ROS system there. This can be done using the following tutorial: <https://itsfoss.com/guide-install-linux-mint-16-dual-boot-windows/>

On the ECE-Robots1 network the IPs for the BBBs are:

- 192.168.1.91 - BBBlue w/o eduMIP modification
- 192.168.1.75 - BBBlue eduMIP
- 192.168.1.125 - BBBlack



## **Thursday, September 13 2018**

Due to the job fair and exam prep we were not able to find the time to come to the lab.



## Friday, September 14 2018

Meeting minutes:

Paper content: Dakota: Introduction for paper Eric: other content



## Tuesday, September 18 2018

Attempted to find out why the multithreaded code branch failed to compile and discovered that the robotics control library has already implemented its own multithreading that is available to be used. Read through documentation in preparation for further implementation.

Today I began to begin to assemble a second BBBlue to be used in testing a leader-follower configuration and algorithm. We will need to 3D print the caster attachment for this new robot as well.





## Thursday, September 20 2018

Started implementing the new multithreading code in the robotics control library. **Software Architecture Framework from BEMOSS:**

Documenting so that I can use this as reference when I translate this into our application.

### **User Interface:**

consists of user management, web browser interface and mobile interface, BEMOSS UI is a responsive user platform that allows seamless interaction with the connected devices using a web browser from any type of device. Implemented using role-based access control to allow different levels of access to users based on their role in the system. The UI layer has built-in authentication and authorization, thus providing secure web access to the connected devices for monitoring.

Web UI:

Mobile UI:

User Management:

### **Application:**

includes various applications, which allow intelligent control of hardware devices interfaced with ROS. Possible applications: fault detection & diagnostics, load shape analysis, demand response, planning and scheduling, behavior pattern analysis, load management and notification analysis.

Scheduling:

Alarm/Notifications:

Demand Response:

Price-Based Management:

Behavior Analysis:

Load Management:

Metadata Database (PostgreSQL): Storing metadata like device information etc.

**Operating System and Agent:** serves as a medium that enables communication among all BEMOSS agents. Communication with the UI layer is also established using IEB BEMOSS agents include device discovery agent, control agents, monitoring agents, and other service agents.

Device Discovery Agents:

Monitoring Agents:

OpenADR Agent:

Time-Series Database (Cassandra): Continuous time-series data collection (power measurement)

VOLTRON Information Exchange Bus (IEB):

Network Agents:

Control Agents:

Platform Agent:

**API Translator:**

takes care of the communication between BEMOSS agents and all physical hardware devices. These translators have been developed to allow BEMOSS agents to communicate with a group based on their Application Programming Interface (APIs). They provide a layer of abstraction for BEMOSS agents so that agents can obtain readings and send control commands to devices regardless of their APIs using simple functions.

API Translator for RadioThem:

API Translator for Wattstopper:

API Translator for WeMo:

Cloud Sources (e.g. OpenADR)

## Friday, September 21 2018

### Meeting Agenda:

- Multithreading Progress
- Example using Two BeagleBone Blues - Leader Follower Algorithm also just Area Coverage
- Printing the New Caster Attachment - Meeting with Dr. Reyer?
- Paper Content - Software Architecture Section and Introduction

### Meeting Minutes:

- Deliverable due Sept 27th: Dr. Miah provided a template of the system level description of our project.
- Describe the problem about 1/2 page - high level no math
- System architecture drawn in IPE. Gradient Density and light colors.
- Not matlab -> make it generic -> display agents as well
- Modes of operation:
  - Density static
  - Density variable
  - Phone control of area area coverage
  - one additional mode
- Final Version of the system level description is due (to Dr. Miah) on Tuesday September 25th
- Put matlab area coverage code in Google Drive! Just do it!
- Generalized block diagram with subsystems
- Discussed multithreading
- Discussed figures

Friday, September 21 2018

---

- Discussed the MAFOSS paper. Follow the example paper from IEEE Trans. on Education that Dr. Miah shared.
- Check with Dr. Reyer on Monday, Sep. 24th about access to the 3D printer.

## Tuesday, September 25 2018

Dakota and I have been working on documenting the paper for the last few days. We are also currently in the process of drafting the Project Proposal for ECE497. Items Completed:

- Put Matlab code into simulationResults folder on Google Drive.
- Gained access to 3D printing lab
- Proposal Intro and Functional Diagrams



## Thursday, September 27 2018

Began setting up the website after Malinowski's lecture. Note that Dakota and I have opted to setup the website using HTML code directly, rather than through a GUI. Dakota has experience in programming websites, so it should not be much of an issue.





## Tuesday, October 2 2018

Went to the 3D printing lab to start printing the part we need for three Beaglebones. The process will take a few hours so in the rest of the time this week I continued to work on the leader follower code. I think it is pretty much ready to be tested now, however I am trying a new method of calling a ROS function, so we will see how well that goes. If I can't get the function to do what I need, I can always switch back to the way we have done it previously.



## Thursday, October 4 2018

Went down to the ME lab and started the print again as the last one messed up during the process. The part itself did not turn out very good. If the part still fits, we will keep it and start testing the code for a leader-follower configuration. In light of this, we only need two robots, HOWEVER we will need three later on so I believe we should order the Beaglebones now. My goal over Fall break is to finish up the writing portion of the first draft of the research paper. Once we get back, we can finish up the Beaglebones and test some of the code.



## Friday, October 5 2018

Meeting with Dr. Miah:

We need to try to talk to Nick or Dr. Reyer about the 3D printing stuff. Any settings -> Or we can print it at home. We will most likely do this at home. Setup a part's request for 3 Beaglebone Blues -> Send Dr. Miah the final list of parts. Complete set of parts including kits. Put the algorithm we are going to implement in VREP -> but we can't put ROS into the system? Implement the ROS code into VREP -> First with the ROS from Jacob and then with then with our system. MAFOSS paper: Include simulations and improve figure. Assume zero knowledge of ROS.

Summery: Send Email by Monday Evening.

Videos need a voiceover. Experimental results section with two Beaglebone Blues talking to each other with and without ROS.

Agenda:

- Ordering more Beaglebones
- First Draft of Paper - Complete this over Fall Break
- 3d printing of new parts?
- IAB feedback and submission?



## Wednesday, October 10 2018

Had a brief meeting with Dr. Miah today regarding how we are going to go about the simulation of our examples. He wants us to try to use Matlab exclusively as the interfacing platform. We will communicate to the robots through the Robotic Systems Toolbox and ROS. As of now we are beginning to research our options as well as look into the specifics of creating models in VREP. I have been working on both a test model and a far more accurate model of the BBBlue that we will use in simulation.

Thus far:

Regarding the parts request form, keeping in mind our \$500 budget, we need to get a whole new version of matlab with simulink and the robotics system toolbox on an Ubuntu operating system. If we include the 3 extra BBBlue robots and eduMIP kits Dr. Miah wants us to get as well, we are pushing this budget.

Current Estimates:

- \$50 for base install of Matlab student edition (for Linux)
- \$10 for base install of Simulink (for Linux)
- \$10 for base install of Robotics Systems Toolbox (for Linux)
- \$88 for BBBlue with appropriate JST connectors (x3)
- \$50 for eduMIP kit (x3)
- \$?? for the casters (x3) - **Need to find out where we got these from!**
- \$7 for 3M10 mm machined screws - [https://www.amazon.com/uxcell-M3x10mm-Stainless-Phillips-Fastener/dp/B01KFQ3NX4/ref=sr\\_1\\_1?s=hi&ie=UTF8&qid=1539222615&sr=1-1&keywords=M3x10mm+machine+screws&dpID=415fKQ0lCpL&preST=\\_SY300\\_QL70\\_&dpSrc=srch](https://www.amazon.com/uxcell-M3x10mm-Stainless-Phillips-Fastener/dp/B01KFQ3NX4/ref=sr_1_1?s=hi&ie=UTF8&qid=1539222615&sr=1-1&keywords=M3x10mm+machine+screws&dpID=415fKQ0lCpL&preST=_SY300_QL70_&dpSrc=srch)

Total Estimate: \$491.00 + caster's cost (not including shipping)

If we can get away with only purchasing 2 Beaglebones, (we would still have 5 Beaglebones in total) then we would be well within the budget should something unexpected occur.





## Friday, October 12 2018

Meeting:

- Need the USB wifi adapter stuff - From Mattus?
- Also need to talk to Mr. Mattus about the software we may need
- Dakota - Send Dr. Miah the type of 3D printer you have
- Send out the partsRequestForm!!!!

<https://www.pololu.com/product/2692> for the casters!



## **Tuesday, October 16 2018**



## Thursday, October 18 2018



## **Tuesday, October 23 2018**





## Thursday, October 25 2018



## Friday, October 26 2018

### Meeting Agenda:

- Parts Request Status
- Progress Regarding Matlab implementation
- Simulation?

### TO DO:

- Send the parts request form to Mr. Mattus and ask for feedback regarding the header connection to breadboard
- Ask for meeting with Mr. Mattus on Monday regarding the breadboard implementation.
- OVER THIS WEEKEND finish the implementation section and update it in the paper.



## Tuesday, October 30 2018

Discovered an issue that we don't entirely understand. When attempting to send any messages via ROS we would receive an `AF_INET` error followed by the master's IP address. We were unable to solve it. This we



## Thursday, November 1 2018

We continued working on the AF\_INET error that we were getting. We were met with little success until we decided to recompile the entire ROS package. This resulted in us no longer getting the error. After solving this issue we concluded in the Lab for the day.





## Friday, November 2 2018

We were finally able to communicate wirelessly via ROS. However in doing so, we discovered that we were not able to use any of the variables that were sent over ROS. ROS uses its own variable types of ros messages. In this case `std_msgs/Float64` was the ROS equivalent to the double variable that we were trying to send. Upon receiving a `Float64` we were unable to use it in any other functions. We assume that this issue is due to it being a non-C standard variable type.



## Tuesday, November 6 2018

For our lab-work today, we continued to research a solution for the typecasting problem. We didn't make any headway by trying to change up how we accepted the data into the callback. Any solution we tried, even changing the message type itself, did not seem to work.



## Friday, November 9 2018

Today we met with Dr. Miah down in the robotics lab to show him the working Matlab to BBB code. Right before our meeting we discovered that the "msg" accepted by the callback function was actually an object and thus we couldn't attempt to change the variable type of this object. We instead had to typecast SPECIFICALLY the msg."data" data portion attached to the object. After solving the problem of typecasting into variables that the RC functions could use, we worked on implementing the line following code. Over this weekend we plan on finishing the code for the line follower from Matlab so that we can begin testing again on Monday. Further, we will begin to document the implementation and simulation results into both the MAFOSS paper and our final draft of the proposal. The deadline for the paper's submission is on November the 19th, so we need it done by the 15th in order to proofread. Under Dr. Miah's direction, Eric will be taking the lead on the first paper, implementing the line-follower and lead-follower algorithms, and Dakota will take the lead on another paper (submission due in some time in January) wherein we implement the full area coverage algorithm.



## Tuesday, November 13

We discovered an error in our low-level control code in the function callbacks. In order to solve this problem we need to figure out a way to include more variables into the callback function.





## Thursday, November 15

To solve the problem we came into on Thursday, we decided to use a class and separate out the variables in a way that ROS can understand. By making the variables public members of a listener class, we can put a function in that class to access these variables as the code runs.



## Friday, November 16

Our meeting today revolved around our progress so far in the implementation side of the leader-follower algorithm. Due to the problems we were facing during this week, we were not able to start any implementation. We continued to update the project proposal with graphics and new information.



## Tuesday, November 20

Sent out the finalized parts request form to Mr. Mattus.  
THANKSGIVING BREAK



## Thursday, November 22

THANKSGIVING BREAK





## Friday, November 23

NO MEETING BECAUSE THANKSGIVING BREAK



## Tuesday, November 27

We continued to work on the implementation, however we decided to spend most of our time on the project presentation, which is due tomorrow.



## Thursday, November 29

We did not meet today.



## Friday, November 30

During the meeting this week Dr. Miah gave us feedback on the project presentation. After this, he instructed us to not do anymore research or work on senior projects until finals are over.





## Wednesday, December 12

We met today to discuss plans for working on our senior project over the winter break.

Meeting Minutes:

- Simulate everything at home using ROS, Matlab, and VREP. Work on implementing everything in the lab. Show the VREP to Dr. Miah by Friday, December 14.
- We discussed possible times that we may be gone over the winter and the times that the lab may be closed.
- Work in the lab from 9:00 am until 4:00 pm. Do research at home.
- Document your hours and show them to Dr. Miah, he will be looking at the budget.
- Start working Wednesday, December 13.
- We may be asked to look at some papers for reference for the senior project paper.
- Last time we checked Mr. Mattus was not available, so we were unable to get updated on the status of our parts request forms. We are going to check again after this meeting.
- Goals for the winter break:
  - Get the static density implementation done.



## Thursday, December 13

Looked into alternative methods for encoder reading and encoder control. Had dentist appointment and didn't come into lab. Also looked into some ways to interface ros and VREP together. Lastly, I looked into possibly changing my Matlab license from Windows to Linux, so that I could run Matlab, VREP, and ROS side-by-side.



## Friday, December 14

TIME SPENT IN LAB 9:30am-4:30pm

We went through a few tutorials on how to get ROS and VREP working together. Note that we will not be able to use Matlab in Linux alongside VREP as we would have to purchase an additional license for the software. So in light of this, we went through getting just VREP on Linux. After going through this tutorial:

- <http://www.coppeliarobotics.com/helpFiles/en/rosTutorialIndigo.htm>

we were able to get linux to run the VREP software. We then downloaded a series of plugins for VREP ROS. It required us to go through and update/install all of the dependencies for the rosInterface plugin.

Finally, in order to test if the rosInterface plugin was functioning, we ran a simulation to test whether or not the plugin was available. Also, we looked at the rosnodestack list output from the console to check if the VREP node was running. Note that that VREP application requires us to run a roscore master on the same local server (i.e. the same computer). This is a serious concern for us as we need to find a way to simulate our implementation with Matlab. From our success in the previous test, we are beginning to develop a simulation using the vrep\_skeleton\_msg\_and\_srv as the barebones. We will also be using the external controller tutorial as a reference:

- <http://www.coppeliarobotics.com/helpFiles/en/externalControllerTutorial.htm>
- <http://www.coppeliarobotics.com/helpFiles/en/rosInterf.htm>

Lastly, we downloaded the package for the vrep\_plugin skeleton if we decided an external controller plugin is the route we want to take:

- [https://github.com/CoppeliaRobotics/vrep\\_plugin\\_skeleton](https://github.com/CoppeliaRobotics/vrep_plugin_skeleton)

In order to run VREP in a linux shell, navigate to the folder that contains the shell script vrep.sh and run:

```
./vrep.sh
```

This will initialize a ROS node on a localhost master through the rosInterface plugin.

Meeting Notes: We met in the lab to show Dr. Miah our implementation of the VREP ROS interface test. After showing the demo we discussed ways to proceed. There were

Friday, December 14

---

two primary ways to proceed after our implementation of the VREP-ROS test. We could either run everything on windows or split the implementation across a windows and linux system. Over the weekend we decided to investigate both of these possible courses of action.

## Monday, December 17

TIME SPENT IN LAB 9:45am-4:30pm (minus 1 hour for lunch)

Over this last weekend Dakota and I looked into the things Dr. Miah suggested during our meeting on Friday. In order to try and get one computer to run I went through and downloaded Windows Subsystem for Linux (WSL) and downloaded ROS. We had some issues in downloading ROS kinetic, so I went ahead and downloaded ROS lunar instead. There should be no compatibility issues as far as I can see. Dakota went through Jacob's old code and how he went about getting ROS and VREP to communicate through Matlab.

This morning when we came in we decided to try to run the system from one master node on Matlab through the lab computer, and one node through VREP on another laptop. After testing this we could see the ros nodes and topics made by each computer on the network. Next we tested to see if we could send a message between the two computers. We started off simple and only sent simulation time data between the two computers. We adjusted the lua code of the VREP simulation to send time data and received it on Matlab. Upon successfully doing this we decided to send commands to VREP by using Matlab. At first we just sent a dummy variable, then we sent a series of numbers. Finally, we were ready to test the line following code on VREP over ROS. We tested it and were met with some success. Our line following algorithm was flawed but did produce a result that looks fairly close to what we should be receiving. We will look into further improving our line following code and documenting a step by step tutorial in the future.





## Tuesday, December 18

TIME SPENT IN LAB 10:30am-4:00pm (minus 1 hour for lunch)

Today Dakota and I started by trying to take our control code apart piece by piece and start implementing it via Matlab. I am almost done creating the guide to setup our VREP and Matlab simulator, I just need to add some pictures and clean up the document.

We spent most of our time today troubleshooting an error with Matlab and the robot. The way they tried to communicate worked for one instance, and when we tried to add any more complexity we got no output. Then, when we went back and tried to implement some old code, that also didn't work. We tried many ways to get around this problem, but eventually we had to do a full reset of our connections. This fixed the issue, however we still don't know exactly why this error happened in the first place. It may have something to do with adding a "node" in the Matlab script, but we have not checked that yet. As it stands, we were able to output the encoder pulses to a Matlab script in real-time. Our next step is to instead send an RPM to Matlab. If we can send the correct RPM, which will include some timing, then we will be in the clear for the rest of the implementation.



## Wednesday, December 19

TIME SPENT IN LAB 9:45am-4:00 (minus 1 hour for lunch)

We finished up the simulation for line following in VREP-ROS using the Pioneer. We fixed the problem with the algorithm where it would just spin in place. Turns out we got the wrong omega value from the arc tangent function. It has since been fixed.

We proceeded to write some code to calculate linear speed from the BBB encoder data. We also discovered that our previous issues with extremely large data was due to signed encoder data being cast as unsigned. We were able to send this data over the ROS network and verified that it worked. The next step will be to add the other wheel so that we can begin to calculate the angular speed of the robot and dead reckon. Once done with this we should be able to send positional data to the line following code. Now we have to rewrite some of the control code, which will be the most time-consuming part.

Meeting: Send the Left and Right Wheel speeds to the VREP environment as well! Such that we can see both the real-time implementation and the simulation at the same times. Get a video of the implementation tomorrow!



## Thursday, December 20

TIME SPENT IN THE LAB 9:45am - 4:45pm (minus 1 hour for lunch)

We began the day by giving feedback for the MAFOSS paper abstract, then proceeded to write code for the line following implementation. We finished writing the code before lunch and then went to lunch.

Dr. Miah wants us to output all of our data from simulation and experimentation to .mat files so that we can look at them. So like a spreadsheet with  $x$ ,  $y$ ,  $\theta$  and coverage vs. time etcetera...



## Friday, December 21

### TIME SPENT IN THE LAB

(Dakota) 9:30am - 3:30 (minus 1 hour for lunch)

(Eric) 10:30 - 3:30 (minus 1 hour for lunch)

We did some research Thursday night and discovered how to use ROS multiarrays. The focus in the lab was to successfully implement the line following algorithm over ROS by using multiarrays. We managed to succeed in sending a message over the ROS network and getting a physical response from the robot despite running into a few issues with clock skews and segmentation faults earlier on in the day. These errors required us to make numerous changes to the code. After fixing these issues we discovered that there were still some bugs in our algorithm that need to be fixed.

Had a meeting with Dylan and Caleb -> need to give them the VREP materials as they will be creating the VREP model for the Beaglebone.





## Friday, January 4

### BACK FROM VACATION

Today we spent most of the time in the lab creating and editing the parts request forms for Dr. Miah's course and for our senior project. Below is the list Dr. Miah instructed us to create for his class (per one group):

- 1 BBBlue
- 1 Plastic base (in EE supplies closet)
- 1 set of JST connectors
  - 2 2-pins ZH
  - 2 4-pins SH
  - 2 6-pins SH
- SD card (at least 8BG)
- MicroUSB cable
- Male header pins
- Female header adapters (for crimping)
- 1 Breadboard

We sent out the completed form to Dr. Miah on Friday.



## Monday, January 7

Continued tuning the PID loop for the motor velocity. We ended up just writing the output response to a file so that we could graph it and get a better understanding of what terms we needed to increase or decrease to get a stable and fast response.

Pictures gains:

First integral gain

$K_p = 2.0$

$K_i = 1.0$

Too much Integral gain

$K_p = 2.0$

$K_i = 200.00$

Too much Derivative gain

$K_p = 2.5$

$K_d = 1.5$

$K_i = 150$

No derivative

$K_p = 2.5$

$K_i = 150$

Too much proportional

$k_p = 5.0$

$K_i = 100.0$

Max proportional

$K_p = 3.5$

Max proportional and integral

$K_p = 3.35$

$K_i = 50$

Monday, January 7

---

Max all and final values

$$K_p = 3.5$$

$$K_i = 25$$

$$K_d = 0$$

Pictures taken while we were tuning parameters will be added at a later date. At this points we do not have any derivative gain as we did not think it was necessary. We also fixed an error regarding the integrator not taking all previous points into account (did not do any summation). The output response from the graph was very close to what we wanted, with a deviation of  $\pm 1.5\text{cm/s}$  at worst case.

## Tuesday, January 8

We added some derivative gain after we noticed some jittering in the output of the PID, then we started the line following implementation.

With the initial gain values in the line following matlab code we saw severe wagging when the robot was attempting to follow the line. In light of this we lowered the steering gain in an attempt to mitigate this. Reducing the steering gain resulted in an algorithm that did not jitter nearly as much, however it no longer followed the line. We spent the majority of the day tuning these gain values. During which we reordered our PID controller to further optimize it.

We took a video of the robot operating and put it in the Google Drive under Implementation/Videos. It was very good at following this reference line, however the matlab plot did not line up exactly with what it was doing in real-time. At the end of the test it would deviate greatly from what matlab was plotting. We think part of this is due to the fact that we were picking up the robot at the end of all the tests, before it was actually complete. It is also possible that the dead reckoning algorithm is not taking any wheel slippage into account as it has no way of knowing how much it has slipped if any. The rest is most likely due to the robot picking up little errors as it runs, like slippage, and steadily diverging as it is not able to correct itself without an external method. Putting the robot on a flat surface helped quite a bit, so we are currently looking into ways to make the robot's movement more rigid. We may end up flipping the robot around and using the caster as the back wheel.



## Wednesday, January 9

Today in the lab we started testing the VREP code to simulate the full area coverage system.

We managed to get the area coverage algorithm to track the positions of the robots in vrep. However, the data that is being sent to the robots may be wrong, or the gain values for the robot control in the area coverage algorithm.

We are going to look into both of these possibilities tonight.





## Thursday, January 10

We realized that our area coverage algorithm will only work if we have good control code running on the robot, so we decided to take some more time to improve the control code by tuning values through the line following algorithm. We began by getting an idea of what needed to be fixed:

- Real-time implementation
- More accurate external control gains
- 

We spent most of the day testing values for the external controller running in Matlab, in order to find a good compromise between the proportional gain on  $d$  and steering gains.



## Friday, January 11



## Monday, January 14



## Tuesday, January 15





## Wednesday, January 16



## Thursday, January 17

How to fix clock skew error:

Go to your catkin workspace (catkin\_tmp) and type in the following command:

```
sudo find . -exec touch {} \;
```

This will "retouch" all of the makefiles in this workspace, resetting the timestamps associated with each file. After this command, they will all have the same timestamps thereby eliminating any clock skew error. If some clock skew error persists, it will not be in a file required to run Cmake in catkin, and thus is a non-issue.



## Friday, January 18

In light of the errors we saw on Wednesday and Thursday, we decided that we should try to re-tune our internal PID values for the controller running on the BBBlue. After re-tuning we were able to get rid of the oscillations and run the code to follow a line. We continued to get errors in our implementation tests as the line the robot wanted to follow was not the one on the ground. At this point we started to think it had to be in our dead reckoning algorithm running on the Beaglebone, however we checked both Euler and RK and both methods gave us very similar results.



## Monday, January 21

We were able to finish up the testing of the line following algorithm today. While we still have some small errors in its operation, there is really nothing we can do right now to fix them. Things like a millimeter of wobble in the wheels, or any slippage on the operating surface.

We solved the issues we were having previously by adjusting the length of the wheel base ( $d$  or  $l$  in the algorithms) and seeing what values would get us to the line experimentally. With this method we were able to follow the line very consistently.





## Tuesday, January 22

We fixed several issues with the area coverage algorithm simulation. The simulation in VREP now has a texture applied to it so that the density regions can easily be seen by the user. Additionally, the pioneers now correctly navigate to their respective centroids. The eduMOD model still needs to be reworked for a more accurate simulation.

We have decided to rename the robot: the new official name will be:  
edu Modified Open-Source Differential Drive Mobile Robot or the eduMOD.



## Wednesday, January 23

Today was the first day of class, so we did not spend much time in the lab. After class at 1:30pm we stopped by and took another look at the Khepera's documentation. We fixed our problem with the ssh over USB and were finally able to access the Linux distribution running on the Khepera. It is a custom build of Linux Yocto. The onboard system seems fairly robust, and we were able to run some tests to make the Khepera move via precompiled code that was present on the robot. We googled the operating system itself and found out that it is based on an outdated "distribution". We really don't want to try to re-flash or update the base operating system, as it seems to be very finicky. With that in mind, we will probably not try to download ROS on the Khepera OS itself. It seems that we will follow Jacob Knolls implementation as it is completed already. We will just need to figure out exactly how he did it.



## Thursday, January 24

Today we decided to create some bash scripts to speed up the ROS setup process for each Beaglebone.

We discovered that in order to get a bash script to retain variables, it needs to be ran with:

```
. ./filename.sh
```

Additionally, we began getting the second eduMOD ready for the leader follower implementation. During this process we noticed that one of the eduMODs had its drive train attached backwards. This resulted in the eduMOD going backwards. We fixed this issue and plan to make corrections to the code to account for this in the future.



## Monday, January 28





## Tuesday, January 29



## **Wednesday, January 30**

Due to the extreme cold and the closing of all campus facilities we did not meet in the lab on this day.



## Thursday, January 31

We started working more on the leader follower implementation and simulation more today. We decided to change some of our code in favor of a more robust algorithm. Also given space constraints we decided to change our leader to follow a circular path rather than a sinusoidal one.

Due to the heavy snow fall we decided to leave a bit early before the roads got too bad.



## Friday, February 1

We worked more on our leader follower algorithm today. We had difficulty getting the robot to follow the circular path as we attempted to use legacy code that we didn't fully understand. As of now we are going to rewrite the code and attempt to get it working without ROS.





## Friday, February 8

We rewrote the dead leader follower code to incorporate a better following algorithm as well as changed the way we calculated robot's current position. Today we also had a brief meeting with Caleb to give him the Beaglebones for building as well as ask him for a status update. He will be starting to build the robot as well as working on the VREP modelling software very soon.



## Monday, February 11

We continued to run into issues using this new leader follower algorithm so we decided to scrap it all and try again from scratch. This time we decided to start our simulations on the circle following itself and build up the rest of the code later. We were able to see the target and one follower moving together, however there are still some bugs.



## Tuesday, February 12



## Wednesday, February 13





## Thursday, February 14

Today we finished testing the leader follower simulation using two Pioneers with and without ROS. As it stands, the gain values tied to each robot are not perfect, but they will do for now. From here we are waiting on Caleb and Dylan to finish modelling of the eduMOD so that we can import it into VREP and tune the simulations gains for those robots. From here we also need to wait on Dylan and Caleb to build the third robot, however we think we may just build it ourselves to lighten their load and speed up the process. Copies of the MATLAB code have been updated to the google drive, however we should try to find a more rigorous method to save the data. In that, we should save each simulations output to a corresponding folder, instead of just using code that overwrites other simulations each run.



## Friday, February 15

Today, we documented and uploaded the previous work that we did with the Pioneer on VREP. All the code that we uploaded has full ROS utilization. Along with the code that has been uploaded we uploaded MATLAB workspace files, videos of both VREP and MATLAB, figures, and the scenes for each. The algorithms used in each implementation are as follows:

The line following algorithm that was used was also used in the ECE 467 labs prior to this project. A simulated line was made and the distance between the robot and that line was calculated. This distance times some gain would determine how aggressively the robot converges towards the line. In addition to this, the angle of the robot with respect to the line was also calculated. This angle was then multiplied time by some gain value to determine how aggressively the robot turns to follow the line.

The leader follower code functions by finding the angle between the leader and follower robots and comparing that angle to the current pose of the follower to find an error. That error is then multiplied by a gain to give a angular speed command to the follower. In a similar fashion, the linear speed of the follower is calculated based on a distance error multiplied by the gain. When both the linear and angular speed gain are set correctly the follower will follow the leader closely.

The area coverage optimization code works much the same as what is currently described in Area Coverage Optimization using Heterogeneous Robots paper by Dr. Suruz Miah and Jacob Knoll.



## Monday, February 18

Today we wrote some readme files for the google drive, uploaded some remaining files that were missed Friday, and captured some more video to be uploaded to the Google Drive.

We implemented the homogeneous leader follower configuration with the EduMOD. Whilst doing this we noticed some strange latency issues that caused inconsistencies.



## Tuesday, February 19

We didn't have any time to work in the lab today. However, we did discuss what might be causing the latency issues. Upon doing some quick math we noticed that we are sending 3 Mbps per robot. We could likely send less data to matlab and from matlab. We hope that this won't change the timing so much that we have to retune the PID. If we do have to re-tune the PID then we will also implement trapezoidal integration.





## Thursday, February 21

Today was the job fair so we did not come into the lab.



## Friday, February 22

We decided to send new data via ROS every 5 iterations of the internal controller. This resulted in much better timing characteristics. Gain values will still have to be tuned for the implementation, however this should not take long.



## Monday, February 25

Today we set out to tune the gain values before the meeting. We were able to see the leader-follower algorithm working, however we still were seeing some problems with the offset on the leader. It appeared to be following a circle 3[cm] below the circle we told it to follow. The Matlab plots indicated that the robot thought it was in the correct position, so the error lies within either the dead-reckoning code on one of the robots, or something on the physical robot itself. Perhaps the robots have slightly different wheelbase lengths.



## Monday, March 4

Did not work in the lab as we had a large Controls exam to complete.





## Wednesday, March 6

The second portion of the Controls take home exam was due so we could not work on the project.



## Thursday, March 7

Today we went into the lab to work on interfacing with the Khepera. We continued where we left off last week and looked more at what was actually on the Khepera. We believed that the code Grant and Jacob wrote was still on the robot somewhere. We were unsuccessful in locating the code so we decided to switch our focus and go through all of the documentation left over by Grant and Jacob to see if they had a copy of the image saved on the Drive somewhere.



## Friday, March 8

We met with Caleb and Dylan to explain how to download ROS on the Beaglebone Blue. In addition to this we helped them construct the final EduMOD.



## Monday, March 11

Meeting:

- Talked about meeting Dylan and Caleb about ROS.
- Eric was assigned to continue working on the Beaglebone Blue
- Talk to Mr. Mattus about a rubber pad for the robotics lab.
- Talk about Autonomous Stuff conference that will be happening. Contact Dr. Driscoll.





## Wednesday, March 13

Today we could not do much in the lab so we took the Beaglebones home to work on them. Eric took the Beaglebone Blue home with him whilst I worked on the black. Eric had trouble with read only errors when trying to update the Beaglebone Blue's SD card, so we resolved to clone the image of one of the existing EduMODs in the morning. I had slightly more success and managed to get the complete working code for the Khepera on the Beaglebone Black. I currently have the code saved on the flash memory. However, ROS will be needed to fully test this code.



## Thursday, March 14

We started this morning by cloning the image from one of the EduMODs to an SD card that would later be used in the 3rd EduMOD. Next we made some minor changes to the MAFOSS paper, including screenshots for implementation. However, we noticed that we were missing some data and attempted to run the relevant code again to recover it. During this process we ran into an issue regarding the configuration of the third EduMOD's wheel configuration. The juniors have apparently soldered the motor leads on in the reversed biased configuration and have reversed the drive train as well. We will need to rebuild the robot in the proper configuration before proceeding with the Homogeneous Area Coverage implementation.



## Friday, March 15

SPRING BREAK STARTS!!!



## Monday, March 25

Before the meeting we managed to get the 3rd eduMOD built and ready to use.

Meeting Minutes:

- Discussed the poster. Display many figures and support them with high level diagrams. Demonstrate that this could not be implemented by just anyone. Highlight the complexity of the project. We have one week to get the skeleton done.
- We need to order 10 more platforms, one monitor (cheap), 2 raspberry pi 3's (look at phone picture), and one rubber mat to test on.
- We will be using the raspberry pi as a display output and server to communicate with the robots.
- Dell UltraSharp 27 Monitor U2719D is the monitor Dr. Miah was looking for.
- Also look at a 24 inch version





## Wednesday, March 27

Today in the lab we set up all of the Beaglebones to run the area coverage algorithm. We also set up a series of .txt files that we are able to put the robot's starting poses into, instead of having to recompile via catkin if we change them in the source code. We also spent some time writing/rewriting the code we are using for implementation.

At the end of the day we ran the code for area coverage and it worked! We added in an offset for the robots to their corresponding centroids in order for us to have them not circle the point infinitely. If they come within 15[cm] of the modified centroid they will stop moving. Tomorrow we will further improve the steering gain values and make some other small adjustments.



## Thursday, March 28

Today we continued to run the area coverage code. We made some minor changes to the threshold and increased the steering gain on the robots. Overall the performance of the algorithm remained about the same. In addition to this we attempted to validate whether the robots were actually traveling to the right positions. We had a lack of tape in the laboratory that prevented us from putting marks down on the floor that halted progress. Before proceeding we want to get a more professional looking work space that could be used in figures for a research paper.

In addition we have been trying to get a parts request form created to send to Mr. Mattus.



## Monday, April 1

### Meeting Minutes

- Finish up the area coverage video-taping
- Getting the Khepera IV robot up and running
- After this we will be going through documentation and appendices









## **Bibliography**