



Laboratory Notebook

Senior Project

Eric Jones and Dakota Adra

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

Beginning August 30, 2018

Contents

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

Contents

Thursday, October 25 2018	47
Friday, October 26 2018	49
Tuesday, October 30 2018	51
Thursday, November 1 2018	53
Friday, November 2 2018	55
Tuesday, November 6 2018	57
Friday, November 9 2018	59
Tuesday, November 13	61
Thursday, November 15	63
Friday, November 16	65
Tuesday, November 20	67
Thursday, November 22	69
Friday, November 23	71
Tuesday, November 27	73
Thursday, November 29	75
Friday, November 30	77

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

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

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.

Bibliography