Kendra Andersen

ECEN 491 w/ Dr. Huff

Project Notebook

**9/18/17 – First Group meeting**

Monday Team meeting: Ryan, Chloe, Sarah, Jacob and I

* Got big rover to run around outside autonomously using the lab’s linux computer
* Sarah isn’t sure how we interact other than her using our data to corroborate her model.
* Ryan is in charge of developing the heat map of the engineering quad
* Chloe needs to speak to the appropriate people about the reconfigurable antenna, isn’t sure what she’s doing
* I need to figure out how to set waypoints/program the rover’s route programmatically instead of just using the default mission planner method.

**9/19/17**

Came in to get started on the rover routing.

Steps to get mission planner going: (Mission Planner failed, APM planner works?)

1. Plug in LiPo battery.
2. Get computer running with APM Planner (Mission Planner doesn’t have an arm function under actions, so I couldn’t get it to work). Plug in USB antenna to computer and set to proper COM port and baud rate of 57600. Pixhawk should be blinking blue if in the lab – there’s no GPS fix.
3. Arm with the computer under the Actions tab.
4. Press arm switch on device until it turns solid red.
5. Turn on controller with joysticks set to middle on manual mode. When it beeps, lower throttle. Note: May have to switch AUX 3 switch. Tab pointed left is manual, tab down/right is auto.
6. Turn on motor switch. It should work.

hrgsafe is the password for the laptop in the lab, if I can’t use my own laptop.

On to programmatically setting waypoints:

* Use save WP file and load WP file for automatic waypoint entry? Not sure if I’m allowed to edit these.
* <http://ardupilot.org/copter/docs/common-planning-a-mission-with-waypoints-and-events.html>
* Also, there’s an Auto grid function. We draw a box around the desired area then Auto WP, Grid.
* <http://qgroundcontrol.org/mavlink/waypoint_protocol#waypoint_file_format>
* That link gives the waypoint file format, I could add appropriate points into it and import it to the rover at the beginning of the mission (idk if this could be updated during the mission) so we might only be able to use past data values, no current information.
* <http://diydrones.com/forum/topics/waypoint-file-format>

Notes:

* Got sample WP file from APM Planner. It seems to agree with the second link above.
* Figuring out what the parameters are:

|  |  |  |
| --- | --- | --- |
| Parameter name | Sample numbers | Explained |
| <VERSION> | 110 | Just a version number. Should stay the same. |
| <INDEX> | 0, 1, 2, 3 | This increases as we go down the file. Start from 0. |
| <CURRENT WP> | 1, 0 |  |
| <COORD FRAME> | 3 | Global or local coordinate frame. |
| <COMMAND> | 16 | MAV\_CMD\_NAV\_WAYPOINT? |
| <PARAM1> |  | Hold time in decimal seconds (ignored by fixed wing, time to stay at MISSION for rotary wing) |
| <PARAM2> | 5 | Acceptance radius in meters (if the sphere with this radius is hit, the MISSION counts as reached) |
| <PARAM3> | 0 | 0 to pass through WP, >0 is the radius in meters to pass by WP. Positive for clockwise orbit, negative for CCW orbit. |
| <PARAM4> | 0 | Desired yaw angle at MISSION (rotary wing) |
| <PARAM5/X/LONGITUDE> | 30.6250429573508285, 30.624636732124813 | Longitude coordinate. Varies between 16 and 15 decimal places, sometimes fewer. |
| <PARAM6/Y/LATITUDE> | -96.3352274894714355, -96.3346481323242188 | Latitude coordinate. Varies between 16 and fewer decimal places. |
| <PARAM7/Z/ALTITUDE> | 0, 20 | 0 at sea level. 20 in College Station. |
| <AUTOCONTINUE> | 1 | I suppose 1 means yes. |

* There should also be a communication protocol method like we used with Railbot and what Ryan is writing right now. Ryan’s only worried about receiving data from the rover, but I would need to send it commands.

Questions to get answered:

1. Do we need to update the route dynamically or can we do it between runs? Basically, are we only using old data to make decisions or also the data we’re measuring as we go along?
2. Figure out how to use the communication protocol. Look into what we did with Railbot last year.
3. If we’re doing the pre-program approach, write a simple python script to create the WP file to load into APM Planner.
4. How to run the other two rovers? Any differences?

**9/25/17 – Second Group Meeting**

Group meeting on Monday. Dr. Huff was present, as was Ryan, Chloe, Sarah, Jacob, & other people I don’t know yet.

Tasks for this upcoming week:

* Get all three rovers running around this week.
* Try to have something to present next Monday.
* Get RSS from mavlink somehow. It should be easy. Start collecting that data (for free!).
* Get 1 or 2 slides together for next week? Just a quick update – this will be part of our documents. Should be on the github somewhere.

Updates:

* Sarah is struggling with terrain files, talking to IT people, etc. They suggested upgrading the system.
* Me: got the big rover running from my laptop.
* Ryan: wrote a program which can get GPS data from the pixhawk. (using serial port) He’s trying to concatenate with RSS data from another HackRF. He can probably get time as well.
* Chloe: set up meetings with the appropriate people. Starts sauntering this week (apparently it’s tiny). She needs to start getting pattern data. Download KICAD?
* Joseph/Jacob: got some object detector working for 5 sec, aiming for it to work in Python – Is he on our project?? I think it’s facial recognition.
* Other people gave updates. Idk them.
* Huff: there’s a tutorial on how to make an antenna if I’m interested.

Questions answered:

1. First we’re gathering data. We will use the data to update decisions in real time later on. (If we experience an anomaly, we either disregard it or do something to improve the situation.)
2. STILL NEED TO DO
3. Not using that approach.
4. How to run the other two rovers? Any differences? Ask Ryan.
5. Team Github? What happened to that?

Assigned it to a guy who works in the lab. He needs to collect everyone’s information. (Vincent)

**9/29/17 – Personal Work in the lab**

* Note: cannot get rovers running by myself – it’s not safe! Need two people to operate. Set up meeting with Ryan next week to make progress on that.
  + Figured out that small rover uses the DXe transmitter on the bench, but could not get any manual movement whatsoever. Not sure what I’m doing wrong. Tried lots of configurations.
  + Large rover is with some guy in the shop? Not sure where it is rn, but it’s the only one operational.
  + Medium rover doesn’t seem to be set up – no pixhawk or other antenna system set up.
* We can get RSSI to show on the computer screen, but idk how to store this data or if it’s even the data we want to be getting. I kinda need a conversation with Ryan on this because I suspect he already has something going.
  + Real-time RSSI display: <http://ardupilot.org/copter/docs/common-rssi-received-signal-strength-indication.html>
  + Downloading data from pixhawks: <http://ardupilot.org/copter/docs/common-downloading-and-analyzing-data-logs-in-mission-planner.html>
  + Ryan already has this pretty much working. Get together with him to start operating next week. He’ll need a raspberry pi.
* In the absence of progress on the assigned topic, let’s look at other topics:
  + Machine Learning techniques
    - <https://www.analyticsvidhya.com/blog/2015/06/machine-learning-basics/>
  + Navigation methods (path planning) given heat maps of areas.

**10/2/17 – Group Meeting**

Ran late b/c of an interview.

My update:

* Worked on smallest rover, struggled with radio.
* Looked into machine learning techniques.

Dr. Huff’s suggestions:

* Look at how Google does their path planning & lists results by travel time. See how we can apply that technique to us with average RSS to get the best.
* Python libraries so we can start running it with dummy data/pre-known answers.
* Get the smaller rover going.
* Start writing my thesis (intro, abstract)
* The antenna tutorial thing?? Do I need to do it? On the Y drive.
* Make a github account because they don’t let the TAMU one work.

Scheduling with Ryan:

Dr. Huff on URS:

* Reference tool management (he recommends EndNote) for the thesis. (On software.tamu.edu for free) Recommends having at least 10 references for our project with some relevance.
* There’s a required event on Oct. 12th from 6-8 \*\*\* CONFLICTS WITH ECEN SCHOLARSHIP BANQUET \*\*\*
* Also the first installment is due Oct. 30th.
* Abstract – very straightforward accounting of everything that will be in the paper. No subjective information in the paper at all (only comparisons, don’t say things are better). “This work examines how to bake a banana nut muffin” – read like an old telegraph. “This includes a discussion on muffin tin. Next is oven settings.” Don’t make it exciting. Make it a reference, citable.
* Dr. Huff will proofread these to an extent, but wants us to use the buddy system. Provide two review before he gets it (for grammar, etc). SHORT SENTENCES – try to avoid long even if they’re grammatically correct. Do the peer review. Have all of the peer reviews by that Friday.

**10/5/17 – Working with Ryan**

* Still working on updates to code. Need to add RADIO\_STATUS message to the pixhawk firmware;
* Useful websites for this process:
  + Source code: <https://github.com/PX4/Firmware/tree/master/src/modules/mavlink>
  + Starting point for enabling/disabling the message : <http://discuss.px4.io/t/enable-disable-mavlink-message-px4fmu-v2/2944/2>
  + Common messages: <http://mavlink.org/messages/common>
  + Making a new message type (not just for firmware): <http://qgroundcontrol.org/mavlink/create_new_mavlink_message>
* **Tuesday 10/3** I got the leads for the batteries on the smallest rover re-soldered to connectors so we can charge the LiPos! The medium one should be chargeable as is because contacts are accessible without changes.
* Maybe need to get the PX4 developer suite installed? Probably not unless I’m recompiling the pixhawk firmware sometime.
* A telemetry wire fried. Idk why – concern!!

Got the smallest rover going! Procedure as follows:

1. Spektrum radio ON first (because we keep forgetting to turn it off) and stick in the middle
2. Plug in pixhawk battery. Arm pixhawk using button once it’s blinking blue. (flashing red to solid red)
3. Turn on APM planner and plug in USB. Connect to pixhawk and arm under actions.
4. Turn on radio. Throttle in the middle. Lights should turn on.
5. Turn on switch for motors. It should run (although only forward) and turn as commanded.

**10/9/17 – Group Meeting**

I have a powerpoint for this one ☺ but didn’t end up using it. (I should have)

Dr. Huff won’t be here next Wednesday

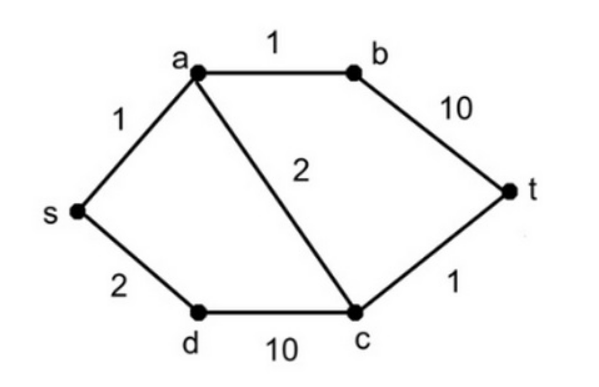
Next Monday: have an idea of how google does mapping & how to do it from an RF standpoint.

Ryan presents:

* PX4 vs Ardupilot – what has what? Ardupilot sounds like a better system for what we’re doing. PX4 is good but not stable.
* Continue with adding what Ryan & I planned on Thursday.
* This week: firmware polling at a constant rate & testing.

**10/16/17 – Personal Research**

Looking into path planning algorithms today –

* <https://motherboard.vice.com/en_us/article/4x3pp9/the-simple-elegant-algorithm-that-makes-google-maps-possible> Dijkstra’s algorithm is how it’s possible.
  + Setup: beginning node is 0 and we have other nodes connected to it and each other by various weights – this can be path length or average strength, etc.
  + 
  + First iteration: look at closest node & next new nodes, as well as previous other nodes. (a, then b, c, and d). The shortest path FROM THE BEGINNING is s -> d (2)
  + Next iteration: look from d to c (2 + 10) and from a, where we can get to c with (1 + 2) and to b with (1 + 1).
  + Next iteration: start at b (2) and can only get to t with (2 + 10) but there’s a shorter path from a -> c (3) and we can discard d -> c.
  + Final iteration: look from b -> t and get total of 12. Look from c -> t and get (3 + 1) so the shortest path is 4.
* <http://www-m3.ma.tum.de/foswiki/pub/MN0506/WebHome/dijkstra.pdf>
* How to code it? <http://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/> Something to consider – the algorithm we’ll be using for this project will need TWO weight variables – average RSSI (computed as a weighted average) and path length. The path length will need to be below a maximum while we need to maximize RSSI. Another consideration – do we need to change speeds? Or just assume one constant speed always.
* <https://www.quora.com/How-does-the-algorithm-of-Google-Maps-work> Another source saying that this is how Google does it.

Libraries for path planning:

* <https://pypi.python.org/pypi/Dijkstar/2.2> Is an implementation of Dijkstra’s single-source shortest-paths algorithm. Accepts an optional cost (“weight”) function to be called each iteration. Decent performance on a graph with 100000+ nodes. (How to import graphs to it?)
* <https://pypi.python.org/pypi/python-graph> is a library for working with graphs in Python. It provides a shortest path search, but doesn’t say what inputs or configurations are available. Documentation is on github? It looks like I need to install it to get the documentation.

**10/16/17 – Group Meeting**

Dr. Huff’s comments on my presentation:

* Start with 100m2 & some sources, generate data to plot and put a grid together.
* Start finding paths that maximize exposure to sources A & B, minimize exposure to C, etc.
* We won’t be worried about RSSI over time. That is another dimension for this problem, however.
* Smallest distinguishable path – we’ll be using sidewalks, etc. Don’t worry about a field. Once I get the demo program working this will probably be pretty simple.
* He was surprised to see Google was still using Dijkstra’s.
  + So I looked into it: Waze might use hybrid things? <https://www.quora.com/What-is-Google-Maps-algorithm-on-recommending-routes>

Dr. Huff’s comments on Ryan’s presentation:

* Ryan recommends scratching the RSSI from pixhawks. Dr. Huff said we had this running quickly from XBees previously, agrees it’s reasonable to scratch on to the more pertinent part of the project since this hardware seems to be operating differently.
* Ryan plans to get HackerRF up this week and getting data this Thursday.

**10/23/17 – Personal Work on project**

Looking into other algorithms for this since Dijkstra’s might not be enough:

* <https://developers.google.com/optimization/routing/tsp/vehicle_routing_time_windows> has multiple variables to optimize through. Not really analogous to this project, though
* <https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm> under “Related problems and algorithms,” it may be best to generate less optimal solutions that have better constraints.
* K shortest path routing <https://en.wikipedia.org/wiki/K_shortest_path_routing>
* How to implement it? <https://stackoverflow.com/questions/13519030/python-dijkstra-k-shortest-paths> <https://www.ics.uci.edu/~eppstein/161/python/dijkstra.py>

Code:

Kendralyn@Zarek MINGW64 ~

$ pip install -U pip setuptools

c:\python27\lib\site-packages\pip\\_vendor\requests\packages\urllib3\util\ssl\_.py:318: SNIMissingWarning: An HTTPS request has been made, but the SNI (Subject Name Indication) extension to TLS is not available on this platform. This may cause the server to present an incorrect TLS certificate, which can cause validation failures. You can upgrade to a newer version of Python to solve this. For more information, see https://urllib3.readthedocs.io/en/latest/security.html#snimissingwarning.

SNIMissingWarning

c:\python27\lib\site-packages\pip\\_vendor\requests\packages\urllib3\util\ssl\_.py:122: InsecurePlatformWarning: A true SSLContext object is not available. This prevents urllib3 from configuring SSL appropriately and may cause certain SSL connections to fail. You can upgrade to a newer version of Python to solve this. For more information, see https://urllib3.readthedocs.io/en/latest/security.html#insecureplatformwarning.

InsecurePlatformWarning

Requirement already up-to-date: pip in c:\python27\lib\site-packages

Collecting setuptools

Downloading setuptools-36.6.0-py2.py3-none-any.whl (481kB)

100% |################################| 481kB 146kB/s

Installing collected packages: setuptools

Found existing installation: setuptools 34.2.0

Uninstalling setuptools-34.2.0:

Successfully uninstalled setuptools-34.2.0

Successfully installed setuptools-36.6.0

c:\python27\lib\site-packages\pip\\_vendor\requests\packages\urllib3\util\ssl\_.py:122: InsecurePlatformWarning: A true SSLContext object is not available. This prevents urllib3 from configuring SSL appropriately and may cause certain SSL connections to fail. You can upgrade to a newer version of Python to solve this. For more information, see https://urllib3.readthedocs.io/en/latest/security.html#insecureplatformwarning.

InsecurePlatformWarning

Kendralyn@Zarek MINGW64 ~

$ pip install c:/Users/Kendralyn/Downloads/Dijkstar-2.2.tar.gz

Processing c:\users\kendralyn\downloads\dijkstar-2.2.tar.gz

Building wheels for collected packages: Dijkstar

Running setup.py bdist\_wheel for Dijkstar ... done

Stored in directory: C:\Users\Kendralyn\AppData\Local\pip\Cache\wheels\9a\37\29\750bf01a8e2a4612742b1ea49bc5c20b814860799a000470a6

Successfully built Dijkstar

Installing collected packages: Dijkstar

Successfully installed Dijkstar-2.2

Kendralyn@Zarek MINGW64 ~

$ python

Python 2.7 (r27:82525, Jul 4 2010, 07:43:08) [MSC v.1500 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> from dijkstar import Graph, find\_path

>>> graph = Graph()

>>> graph.add\_edge(1, 2, {'cost': 1}\_

File "<stdin>", line 1

graph.add\_edge(1, 2, {'cost': 1}\_

^

SyntaxError: invalid syntax

>>> graph.add\_edge(1, 2, {'cost': 1})

>>> graph.add\_edge(2, 3, {'cost': 2})

>>> cost\_func = lambda u, v, e, prev\_e: e['cost']

>>> find\_path(graph, 1, 2, cost\_func = cost\_func)

([1, 2], [{'cost': 1}], [1], 1)

>>> find\_path(graph, 1, 3, cost\_func = cost\_func)

([1, 2, 3], [{'cost': 1}, {'cost': 2}], [1, 2], 3)

>>> graph.add\_edge(1, 3, {'cost':4})

>>> find\_path(graph, 1, 3, cost\_func = cost\_func)

([1, 2, 3], [{'cost': 1}, {'cost': 2}], [1, 2], 3)

>>> graph.add\_edge(1, 4, {'cost':4})

>>> graph.add\_edge(1, 3, {'cost':2})

>>> find\_path(graph, 1, 4, cost\_func = cost\_func)

([1, 4], [{'cost': 4}], [4], 4)

>>> graph.add\_edge(3, 4, {'cost':2})

>>> find\_path(graph, 1, 4, cost\_func = cost\_func)

([1, 4], [{'cost': 4}], [4], 4)

>>>

**10/23/17 – Group Meeting**

Ryan’s presentation:

* How to power the neuc? He might order power packs from somewhere, but it’s a 12V battery on the rovers – try to use that?
* Neuc is 19V, he doesn’t see why we can’t use the rover power. Else grab a marine battery.

Dr. Huff’s comments:

* Multi-dimensional gradient descent. Through the pathway for each one
* K shortest path – he’ll look at it again, I think it might work.

First draft due next Monday at noon.

* Introduction & Abstract, but up to us as to how much is included in this draft.
* Reference management tool – start putting the shell of the document together.
* Try to have something he can look at on/by Friday.

**10/30/17 – Working on algorithm**

* Realized that the cost-func/Dijkstar implementation won’t work because it just adds the costs of the different lines.

**10/30/17 – Group Meeting**

* Next week, have a more comprehensive mini-presentation for the group meeting.
  + 4-5 charts, more detail about the current state
  + 5 slides, one with a really good description of the project. One slide with overview of the current state of the project, 2+ outlining where we’re at, what we’ve been doing (pictures/findings/progress)
  + Final presentation type slides – basic information about the progress of the project
  + How we arrive at the result and what the result is.
  + Also put the presentation in a common collection point – drive
* Build a dataset generator using isotropic sources.
  + Pick a 100m x 100m space, randomly position transmitters & use as a simulation space. (Friis transmission formula). This gives datasests to put in.
  + Just generate it and make an arbitrary path or two for me to minimize & maximize over.

**11/3/17 – Personal Programming Work**

* Useful websites I used:
  + Wikipedia on the Friis transmission equation:

<https://en.wikipedia.org/wiki/Friis_transmission_equation>

* + Python basics: <http://www.astro.ufl.edu/~warner/prog/python.html>
  + matplotlib 3D tutorial: <https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html>
  + Forum post about color plotting: <https://stackoverflow.com/questions/7229971/2d-grid-data-visualization-in-python>
  + Imshow info: <https://matplotlib.org/devdocs/api/_as_gen/matplotlib.pyplot.imshow.html>
  + Pyplot tutorial: <https://matplotlib.org/users/pyplot_tutorial.html>
* Useful websites from previous web searching while working on Dijkstar:
  + <https://wiki.python.org/moin/KeyError>
  + <https://www.python-course.eu/lambda.php>
  + <https://www.youtube.com/watch?v=V3RJGWaYqxQ>
  + <http://code.activestate.com/recipes/117228/>
  + <https://bitbucket.org/wyatt/dijkstar/src/ef3a762d4f15bb7df25aa7244c2677eef8acfa24/dijkstar/algorithm.py?at=master&fileviewer=file-view-default>
  + <https://www.ics.uci.edu/~eppstein/161/python/dijkstra.py>
  + <https://stackoverflow.com/questions/13519030/python-dijkstra-k-shortest-paths>
  + <https://en.wikipedia.org/wiki/K_shortest_path_routing>

**11/6/17 – Group Meeting**

Presentation about my project (what it is) today!

* Ryan: check the powers on the Xbees b/c there should be better signal from further away.
* Not far from using actual data
* Take one of the sources and apply a sin2 envelope (figure 8 looking) -> more for 2 weeks from now. Basically play with the directionality of the antennas.
* Focus on the optimization problem, use dijkstra’s and min/max different transmitters.
* Plot the ratio of best received over the path. (plot what we want/what we don’t want for each point on the path)
* Continue to build on these powerpoints, another update next week.

**11/13/17 – Group Meeting:**

My questions:

* Not sure where to go from where I am – In the maximization problem, how should I handle more than a simple SNR configuration?
* He mentioned having directional antennas?
* How to implement dijkstra’s on this type of problem.

Huff comments:

* Start adding dijkstra’s using the hash (or H or Y) from the starting point,
* Using this data, start integrating the path planning/optimization algorithms.
* If we have 4 transmitters, max power overall
* Next problem: turn one transmitter into hostile, what are the different paths looking like? Which one is recommended, it should change depending on which is the bad transmitter.
* Start/stop entry points remain the same.
* Once we have this, we can start with the realistic framework. (more from Sarah or buildings or directional or stuff)