



---

# Laboratory Notebook

**Indoor Robot Localization and Mapping using ZigBee Radio  
Technology**

---

Kyle Hevrdejs and Jacob Knoll

[khevrdejs, jknoll]@mail.bradley.edu

Beginning September 22, 2016



## Index

- AIM 2017 Conference Paper, 177, 181, 197, 199, 201
- Beaglebone Black Setup, 127–131, 141, 147
- Conference paper, 79–80, 90, 93, 113, 119, 121
- ECE 497, 19, 28–29, 41, 45, 47, 70
  - Subsystem level functional requirements, 29, 41, 45, 47, 70
  - System level functional requirements, 19, 28–29
- ECE 498, 93, 105, 109, 111, 113, 115
  - Project Proposal, 93, 105, 109, 111, 113, 115
- Equations, 33–37
  - Augmenting new features in the map, 37
  - Choosing a waypoint, 35
  - Compute steering angle, 34–35
  - Control noise, 33
  - Data association, 35–36
  - Get observation, 35
  - Observation noise, 33
  - Update, 36–37
  - Vehicle Model, 33–34
- Lab work, 21, 23, 25, 41, 47, 49, 51, 55, 57, 59, 63–73, 75, 77, 79, 81, 85, 87, 89–91, 93, 95, 99, 101–103, 123, 131, 135–137, 139, 142–145, 147, 149–151, 153, 155–161, 167–169, 171, 173, 175, 179, 181, 183, 185, 187, 189, 191, 193, 195, 205–209, 211, 213, 215, 217–218, 221–223, 225, 227–229, 233, 235–239, 241, 243, 245, 249, 271, 273
- Code Conversion, 205–209, 211, 213
- Designing Experiments, 135–137, 139, 142–145, 147, 149–151, 153, 155–161, 167–169, 171, 173, 175, 179, 181, 183, 185, 187, 189, 191, 193, 195, 205
- Matlab with ROS, 215, 217–218, 221–223, 225, 227–229
- V-REP Simulation, 47, 51, 55, 57, 63–66, 69–73, 75, 77, 79, 81, 85, 87, 131
- Xbee Modules, 21, 23, 25, 41, 47, 49, 51, 55, 59, 63, 67–69, 71, 75, 77, 79, 81, 85, 89–91, 93, 103
- Meeting with Dr. Miah, 19–20, 27, 39, 42, 53, 61, 65, 77, 83, 87, 103, 107, 111, 117, 141, 161–164, 187, 203, 217, 229–231, 243, 265, 269, 277
- Xbee Modules, 27
- New Reflector, 195, 211
- New uploads, 20, 28, 57, 61, 64, 66, 70, 83, 87, 131–133, 150, 213
  - Designing Experiments, 131–133
- Notation, 43
- uploads, 191
- Website, 175, 203



# Contents

<b>Index</b>	<b>3</b>
<b>22 September 2016</b>	<b>19</b>
1 ECE 497 . . . . .	19
1.1 System level functional requirements . . . . .	19
2 Meeting with Dr. Miah . . . . .	19
3 New uploads . . . . .	20
<b>26 September 2016</b>	<b>21</b>
1 Lab work . . . . .	21
1.1 Xbee Modules . . . . .	21
<b>27 September 2016</b>	<b>23</b>
1 Lab work . . . . .	23
1.1 Xbee Modules . . . . .	23
<b>28 September 2016</b>	<b>25</b>
1 Lab work . . . . .	25
1.1 Xbee Modules . . . . .	25
<b>29 September 2016</b>	<b>27</b>
1 Meeting with Dr. Miah . . . . .	27
2 Meeting with Dr. Miah . . . . .	27
2.1 Xbee Modules . . . . .	27
3 ECE 497 . . . . .	28
3.1 System level functional requirements . . . . .	28
4 New uploads . . . . .	28
<b>2 October 2016</b>	<b>29</b>
1 ECE 497 . . . . .	29
1.1 System level functional requirements . . . . .	29
1.2 Subsystem level functional requirements . . . . .	29
<b>3 October 2016</b>	<b>31</b>

<b>4 October 2016</b>	<b>33</b>
1 Equations . . . . .	33
1.1 Control noise . . . . .	33
1.2 Observation noise . . . . .	33
1.3 Vehicle Model . . . . .	33
1.4 Compute steering angle . . . . .	34
1.5 Choosing a waypoint . . . . .	35
1.6 Get observation . . . . .	35
1.7 Data association . . . . .	35
1.8 Update . . . . .	36
1.9 Augmenting new features in the map . . . . .	37
<b>6 October 2016</b>	<b>39</b>
1 Meeting with Dr. Miah . . . . .	39
<b>12 October 2016</b>	<b>41</b>
1 ECE 497 . . . . .	41
1.1 Subsystem level functional requirements . . . . .	41
2 Lab work . . . . .	41
2.1 Xbee Modules . . . . .	41
3 Meeting with Dr. Miah . . . . .	42
<b>13 October 2016</b>	<b>43</b>
1 Notation . . . . .	43
<b>16 October 2016</b>	<b>45</b>
1 ECE 497 . . . . .	45
1.1 Subsystem level functional requirements . . . . .	45
<b>17 October 2016</b>	<b>47</b>
1 Lab work . . . . .	47
1.1 Xbee Modules . . . . .	47
1.2 V-REP Simulation . . . . .	47
2 ECE 497 . . . . .	47
2.1 Subsystem level functional requirements . . . . .	47
<b>18 October 2016</b>	<b>49</b>
1 Lab work . . . . .	49
1.1 Xbee Modules . . . . .	49
<b>19 October 2016</b>	<b>51</b>
1 Lab work . . . . .	51
1.1 Xbee Modules . . . . .	51

1.2	V-REP Simulation . . . . .	51
<b>20 October 2016</b>		<b>53</b>
1	Meeting with Dr. Miah . . . . .	53
<b>21 October 2016</b>		<b>55</b>
1	Lab work . . . . .	55
1.1	Xbee Modules . . . . .	55
1.2	V-REP Simulation . . . . .	55
<b>22 October 2016</b>		<b>57</b>
1	Lab work . . . . .	57
1.1	V-REP Simulation . . . . .	57
2	New uploads . . . . .	57
<b>24 October 2016</b>		<b>59</b>
1	Lab work . . . . .	59
1.1	Xbee Modules . . . . .	59
<b>25 October 2016</b>		<b>61</b>
1	Meeting with Dr. Miah . . . . .	61
2	New uploads . . . . .	61
<b>26 October 2016</b>		<b>63</b>
1	Lab work . . . . .	63
1.1	Xbee Modules . . . . .	63
1.2	V-REP Simulation . . . . .	63
2	New uploads . . . . .	64
<b>27 October 2016</b>		<b>65</b>
1	Lab work . . . . .	65
1.1	V-REP Simulation . . . . .	65
2	Meeting with Dr. Miah . . . . .	65
3	Lab work . . . . .	66
3.1	V-REP Simulation . . . . .	66
4	New uploads . . . . .	66
<b>28 October 2016</b>		<b>67</b>
1	Lab work . . . . .	67
1.1	Xbee Modules . . . . .	67

<b>31 October 2016</b>	<b>69</b>
1 Lab work . . . . .	69
1.1 Xbee Modules . . . . .	69
1.2 V-REP Simulation . . . . .	69
2 ECE 497 . . . . .	70
2.1 Subsystem level functional requirements . . . . .	70
3 New uploads . . . . .	70
<b>1 November 2016</b>	<b>71</b>
1 Lab work . . . . .	71
1.1 Xbee Modules . . . . .	71
1.2 V-REP Simulation . . . . .	71
<b>2 November 2016</b>	<b>75</b>
1 Lab work . . . . .	75
1.1 Xbee Modules . . . . .	75
1.2 V-REP Simulation . . . . .	75
<b>3 November 2016</b>	<b>77</b>
1 Meeting with Dr. Miah . . . . .	77
2 Lab work . . . . .	77
2.1 V-REP Simulation . . . . .	77
2.2 Xbee Modules . . . . .	77
<b>4 November 2016</b>	<b>79</b>
1 Lab work . . . . .	79
1.1 Xbee Modules . . . . .	79
1.2 V-REP Simulation . . . . .	79
2 Conference paper . . . . .	79
<b>7 November 2016</b>	<b>81</b>
1 Lab work . . . . .	81
1.1 Xbee Modules . . . . .	81
1.2 V-REP Simulation . . . . .	81
<b>8 November 2016</b>	<b>83</b>
1 Meeting with Dr. Miah . . . . .	83
2 New uploads . . . . .	83
<b>9 November 2016</b>	<b>85</b>
1 Lab work . . . . .	85
1.1 Xbee Modules . . . . .	85
1.2 V-REP Simulation . . . . .	85



<b>10 November 2016</b>	<b>87</b>
1 Meeting with Dr. Miah . . . . .	87
2 Lab work . . . . .	87
2.1 V-REP Simulation . . . . .	87
3 New uploads . . . . .	87
<b>11 November 2016</b>	<b>89</b>
1 Lab work . . . . .	89
1.1 Xbee Modules . . . . .	89
2 Conference paper . . . . .	90
<b>14 November 2016</b>	<b>91</b>
1 Lab work . . . . .	91
1.1 Xbee Modules . . . . .	91
<b>15 November 2016</b>	<b>93</b>
1 Lab work . . . . .	93
1.1 Xbee Modules . . . . .	93
2 Conference paper . . . . .	93
3 ECE 498 . . . . .	93
3.1 Project Proposal . . . . .	93
<b>16 November 2016</b>	<b>95</b>
1 Lab work . . . . .	95
<b>17 November 2016</b>	<b>97</b>
<b>18 November 2016</b>	<b>99</b>
1 Lab work . . . . .	99
<b>19 November 2016</b>	<b>101</b>
1 Lab work . . . . .	101
<b>21 November 2016</b>	<b>103</b>
1 Lab work . . . . .	103
1.1 Xbee Modules . . . . .	103
2 Meeting with Dr. Miah . . . . .	103
<b>22-27 November 2016</b>	<b>105</b>
1 ECE 498 . . . . .	105
1.1 Project Proposal . . . . .	105

<b>28 November 2016</b>	<b>107</b>
1 Meeting with Dr. Miah . . . . .	107
<b>29 November 2016</b>	<b>109</b>
1 ECE 498 . . . . .	109
1.1 Project Proposal . . . . .	109
<b>1 December 2016</b>	<b>111</b>
1 Meeting with Dr. Miah . . . . .	111
2 ECE 498 . . . . .	111
2.1 Project Proposal . . . . .	111
<b>2-5 December 2016</b>	<b>113</b>
1 ECE 498 . . . . .	113
1.1 Project Proposal . . . . .	113
2 Conference paper . . . . .	113
<b>6 December 2016</b>	<b>115</b>
1 ECE 498 . . . . .	115
1.1 Project Proposal . . . . .	115
<b>7 December 2016</b>	<b>117</b>
1 Meeting with Dr. Miah . . . . .	117
<b>8-9 December 2016</b>	<b>119</b>
1 Conference paper . . . . .	119
<b>10 December 2016</b>	<b>121</b>
1 Conference paper . . . . .	121
<b>11 December 2016</b>	<b>123</b>
1 Lab work . . . . .	123
<b>8 January 2017</b>	<b>125</b>
<b>9 January 2017</b>	<b>127</b>
1 Beaglebone Black Setup . . . . .	127
<b>10 January 2017</b>	<b>129</b>
1 Beaglebone Black Setup . . . . .	129
2 Lab work . . . . .	131
2.1 V-REP Simulation . . . . .	131
3 New uploads . . . . .	131
3.1 Designing Experiments . . . . .	131

<b>11 January 2017</b>	<b>135</b>
1 Lab work . . . . .	135
1.1 Designing Experiments . . . . .	135
<b>13 January 2017</b>	<b>139</b>
1 Lab work . . . . .	139
1.1 Designing Experiments . . . . .	139
<b>19 January 2017</b>	<b>141</b>
1 Meeting with Dr. Miah . . . . .	141
2 Beaglebone Black Setup . . . . .	141
3 Lab work . . . . .	142
3.1 Designing Experiments . . . . .	142
<b>20 January 2017</b>	<b>147</b>
1 Lab work . . . . .	147
1.1 Designing Experiments . . . . .	147
2 Beaglebone Black Setup . . . . .	147
<b>23 January 2017</b>	<b>149</b>
1 Lab work . . . . .	149
1.1 Designing Experiments . . . . .	149
2 New uploads . . . . .	150
<b>24 January 2017</b>	<b>151</b>
1 Lab work . . . . .	151
1.1 Designing Experiments . . . . .	151
<b>25 January 2017</b>	<b>153</b>
1 Lab work . . . . .	153
1.1 Designing Experiments . . . . .	153
<b>26 January 2017</b>	<b>155</b>
1 Lab work . . . . .	155
1.1 Designing Experiments . . . . .	155
<b>27 January 2017</b>	<b>161</b>
1 Lab work . . . . .	161
1.1 Designing Experiments . . . . .	161
2 Meeting with Dr. Miah . . . . .	161

<b>30 January 2017</b>	<b>167</b>
1 Lab work . . . . .	167
1.1 Designing Experiments . . . . .	167
<b>31 January 2017</b>	<b>169</b>
1 Lab work . . . . .	169
1.1 Designing Experiments . . . . .	169
<b>1 February 2017</b>	<b>171</b>
1 Lab work . . . . .	171
1.1 Designing Experiments . . . . .	171
<b>2 February 2017</b>	<b>173</b>
1 Lab work . . . . .	173
1.1 Designing Experiments . . . . .	173
<b>3 February 2017</b>	<b>175</b>
1 Lab work . . . . .	175
1.1 Designing Experiments . . . . .	175
2 Website . . . . .	175
<b>4 February 2017</b>	<b>177</b>
1 AIM 2017 Conference Paper . . . . .	177
<b>6 February 2017</b>	<b>179</b>
1 Lab work . . . . .	179
1.1 Designing Experiments . . . . .	179
<b>7 February 2017</b>	<b>181</b>
1 Lab work . . . . .	181
1.1 Designing Experiments . . . . .	181
2 AIM 2017 Conference Paper . . . . .	181
<b>8 February 2017</b>	<b>183</b>
1 Lab work . . . . .	183
1.1 Designing Experiments . . . . .	183
<b>9 February 2017</b>	<b>185</b>
1 Lab work . . . . .	185
1.1 Designing Experiments . . . . .	185

<b>10 February 2017</b>	<b>187</b>
1 Lab work . . . . .	187
1.1 Designing Experiments . . . . .	187
2 Meeting with Dr. Miah . . . . .	187
<b>13 February 2017</b>	<b>189</b>
1 Lab work . . . . .	189
1.1 Designing Experiments . . . . .	189
<b>14 February 2017</b>	<b>191</b>
1 Lab work . . . . .	191
1.1 Designing Experiments . . . . .	191
2 uploads . . . . .	191
<b>15 February 2017</b>	<b>193</b>
1 Lab work . . . . .	193
1.1 Designing Experiments . . . . .	193
<b>17 February 2017</b>	<b>195</b>
1 Lab work . . . . .	195
1.1 Designing Experiments . . . . .	195
2 New Reflector . . . . .	195
<b>18 February 2017</b>	<b>197</b>
1 AIM 2017 Conference Paper . . . . .	197
<b>19 February 2017</b>	<b>199</b>
1 AIM 2017 Conference Paper . . . . .	199
<b>20-23 February 2017</b>	<b>201</b>
1 AIM 2017 Conference Paper . . . . .	201
<b>24 February 2017</b>	<b>203</b>
1 Website . . . . .	203
2 Meeting with Dr. Miah . . . . .	203
<b>27 February 2017</b>	<b>205</b>
1 Lab work . . . . .	205
1.1 Designing Experiments . . . . .	205
2 Lab work . . . . .	205
2.1 Code Conversion . . . . .	205

<b>28 February 2017</b>	<b>211</b>
1 Lab work . . . . .	211
1.1 Code Conversion . . . . .	211
2 New Reflector . . . . .	211
<b>1 March 2017</b>	<b>213</b>
1 Lab work . . . . .	213
1.1 Code Conversion . . . . .	213
2 New uploads . . . . .	213
<b>2 March 2017</b>	<b>215</b>
1 Lab work . . . . .	215
1.1 Matlab with ROS . . . . .	215
<b>3 March 2017</b>	<b>217</b>
1 Meeting with Dr. Miah . . . . .	217
2 Lab work . . . . .	217
2.1 Matlab with ROS . . . . .	217
<b>4 March 2017</b>	<b>219</b>
<b>6 March 2017</b>	<b>221</b>
1 Lab work . . . . .	221
1.1 Matlab with ROS . . . . .	221
<b>7 March 2017</b>	<b>223</b>
1 Lab work . . . . .	223
1.1 Matlab with ROS . . . . .	223
<b>8 March 2017</b>	<b>225</b>
1 Lab work . . . . .	225
1.1 Matlab with ROS . . . . .	225
<b>9 March 2017</b>	<b>227</b>
1 Lab work . . . . .	227
1.1 Matlab with ROS . . . . .	227
<b>10 March 2017</b>	<b>229</b>
1 Lab work . . . . .	229
1.1 Matlab with ROS . . . . .	229
2 Meeting with Dr. Miah . . . . .	229

<b>20 March 2017</b>	<b>233</b>
1 Lab work . . . . .	233
<b>21 March 2017</b>	<b>235</b>
1 Lab work . . . . .	235
<b>22 March 2017</b>	<b>241</b>
1 Lab work . . . . .	241
<b>23 March 2017</b>	<b>243</b>
1 Lab work . . . . .	243
2 Meeting with Dr. Miah . . . . .	243
<b>27 March 2017</b>	<b>245</b>
1 Lab work . . . . .	245
<b>26 March 2017</b>	<b>247</b>
<b>27 March 2017</b>	<b>249</b>
1 Lab work . . . . .	249
<b>28 March 2017</b>	<b>251</b>
<b>29 &amp; 30 March 2017</b>	<b>253</b>
<b>3 April 2017</b>	<b>255</b>
<b>4 April 2017</b>	<b>257</b>
<b>5 April 2017</b>	<b>259</b>
<b>6 April 2017</b>	<b>261</b>
<b>7 April 2017</b>	<b>263</b>
<b>10 April 2017</b>	<b>265</b>
1 Meeting with Dr. Miah . . . . .	265
<b>11-13 April 2017</b>	<b>267</b>
<b>14 April 2017</b>	<b>269</b>
1 Meeting with Dr. Miah . . . . .	269
<b>17 April 2017</b>	<b>271</b>
1 Lab work . . . . .	271

Contents

---

<b>18 April 2017</b>	<b>273</b>
1 Lab work . . . . .	273
<b>18-19 April 2017</b>	<b>275</b>
<b>20 April 2017</b>	<b>277</b>
1 Meeting with Dr. Miah . . . . .	277
<b>23 April 2017</b>	<b>279</b>
<b>24 April 2017</b>	<b>281</b>
<b>25-27 April 2017</b>	<b>283</b>
<b>28 April 2017</b>	<b>285</b>
<b>1 May 2017</b>	<b>287</b>
<b>2 May 2017</b>	<b>289</b>







## 22 September 2016

### 1 Work completed during ECE 497 class time

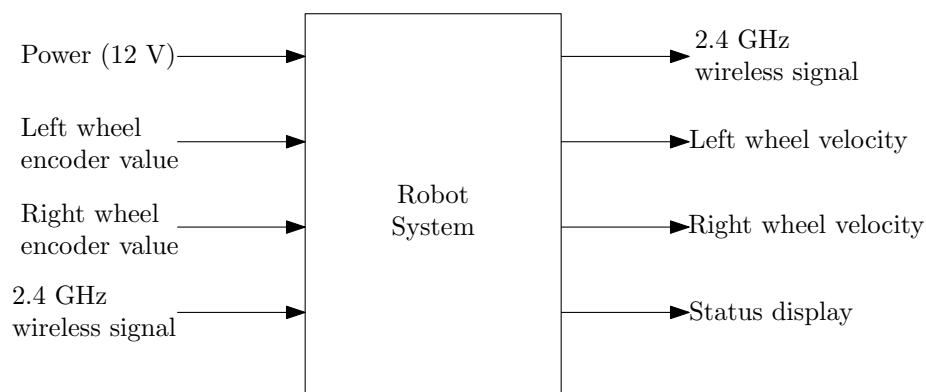
We received a better description of what should be included in the deliverable of the high level system requirements.

#### 1.1 Preparation of system level functional requirements deliverable

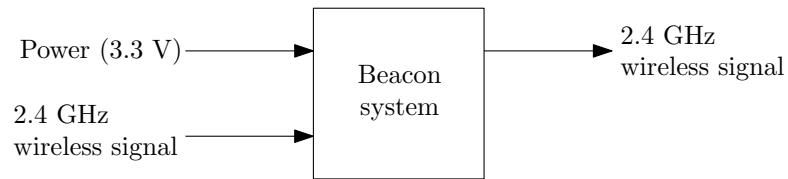
We discussed what kinds of inputs and outputs we should include in our high level system requirements deliverable that is due on Monday, September 26. We decided on two systems, one for the robot and one for the Xbee beacons. The block diagrams for the robot system and Xbee system can be see in [Figure 1](#) and [Figure 2](#), respectively.

### 2 Weekly meeting to discuss progress with Dr. Miah

We had our first official meeting with Dr. Miah today to discuss a schedule and any progress we have made. We talked about high level system requirements as well. For next Thursday, September 29, we should work on interfacing an Xbee module with a BeagleBone Black so it can send a request for signal strength to another Xbee being powered by a battery. The signal strength should then be displayed so the data can be



**Figure 1:** High level block diagram of the robot system



**Figure 2:** *High level block diagram of the beacon system*

viewed. We might also want to look into converting RSSI to distance. If we can get all that done before Thursday, it would be a good idea to run some tests and record signal strengths at certain measured distances and come up with a plot. A plot such as that would look good in a paper later on or when we are presenting our work at the end of the year.

### **3 New uploads for the day to Google Drive**

- Block diagrams included in today's lab notebook entry, *highLevelSysRequirements\_robot* and *highLevelSysRequirements\_beacon*
- High level system requirements deliverable for ECE 497
- The  $\text{\LaTeX}$  source for the lab notebook

## 26 September 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

We are working in the lab today, interfacing an Xbee with the BeagleBone Black. I found a decent tutorial on a way to do it <http://www.jerome-bernard.com/blog/2012/06/04/beaglebone-serial-ports-and-xbees/>. Its for an older version of the BeagleBone but it should still help us get things set up initially. It suggests using an Xbee Explorer Regulated to make for easier connections. We don't have any of these interfacing boards yet so we are just trying to do all the connections through a breadboard. Once we start making the beacons and setting things up we should get some of the boards so there is less circuitry for us to design and implement.

My goal for today is to at least be able to power the Xbees (beacon and transmitter) and be able to get RSSI from the beacon. We do not need to write a program to do it (since we can use minicom for testing) but if the BeagleBone does not have minicom we will have to write a program.

For the circuit we will need resistors (in case we need to make a voltage divider), a voltage regulator, wires, and a breadboard. Since it probably won't be battery powered (yet) we may end up using a power supply, which we will have to get from somewhere.

In the lab we realized that we must have an interfacing board for the Xbees since they are too small to fit in a breadboard. We looked at how to enable the UART functionality on one of the BeagleBones in the lab as well. Since they are running Ubuntu 12, there is no cape manager included in the kernel. This is an issue so I used my BeagleBone and was able to successfully enable UART2 using the following command (in Ubuntu 16.04):

```
sudo su
echo BB-UART2 > /sys/devices/platform/bone_capemgr/slots
```

Then by reading the contents of the file slots we were able to verify that the "cape" had been loaded, meaning UART2 was enabled.



## 27 September 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

as the beacon. In ECE 497 we did Design Review Teams (DRTs). We haven't gotten the review of our system level functional requirements yet. Hopefully we get this tomorrow or Thursday so we can update the document in a timely manner.





## 28 September 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

We tried using the interface boards provided by Dr. Miah. We were unable to get them to work. We suspect this is due to the fact that the PCB doesn't have the connectors soldered on and we could only have the pieces pressure fitted together.

We looked at some adapters that we can discuss with Dr. Miah tomorrow.



## 29 September 2016

### 1 Weekly meeting to discuss progress with Dr. Miah

The weekly meeting was today. This is what we need to do before the next meeting:

- Look at Xbee interfacing boards and email Mr. Mattus tomorrow (9/30). Also go in and talk with him in person so he knows time is a factor.
- Read through all the papers in Google Drive and see how they found their angle measurements. See if these methods could be adapted to our purpose.
- Go through the MATLAB code from a high level point of view. Understand roughly what each part does, what the functions do, etc.
- Write up the functions from the notes that Dr. Miah gave us. Should write them in LaTeX so they are easy to copy and paste in the future.

### 2 Weekly meeting to discuss progress with Dr. Miah

#### 2.1 Working with the Xbee Modules

Here is the list of Xbee adapters/interfacing boards we think look good for our needs:

- <https://www.sparkfun.com/products/11812>
- <https://www.sparkfun.com/products/11373>
- [http://www.miniinthebox.com/ft232rl-xbee-usb-to-serial-adapter-v1-2-board-module-for-arduino-p1141448.html?currency=USD&litb\\_from=paid\\_adwords\\_shopping&utm\\_source=google\\_shopping&utm\\_medium=cpc&adword\\_mt=&adword\\_ct=94710053401&adword\\_kw=&adword\\_pos=1o5&adword\\_pl=&adword\\_net=g&adword\\_tar=&adw\\_src\\_id=1364803665\\_328601281\\_21739983481\\_pla-146293783018&gclid=CMPCq9\\_Ctc8CFQUuaQodzgtgBcw](http://www.miniinthebox.com/ft232rl-xbee-usb-to-serial-adapter-v1-2-board-module-for-arduino-p1141448.html?currency=USD&litb_from=paid_adwords_shopping&utm_source=google_shopping&utm_medium=cpc&adword_mt=&adword_ct=94710053401&adword_kw=&adword_pos=1o5&adword_pl=&adword_net=g&adword_tar=&adw_src_id=1364803665_328601281_21739983481_pla-146293783018&gclid=CMPCq9_Ctc8CFQUuaQodzgtgBcw)

### **3 Work completed during ECE 497 class time**

#### **3.1 Preparation of system level functional requirements deliverable**

We got the reviews of our system level functional requirements back. We did pretty well. The only notes we got was to explain some terminology a little better so it can be understood by someone without robotics knowledge.

### **4 New uploads for the day to Google Drive**

- The DRT's notes on our system level functional requirements

## 2 October 2016

### **1 Work completed during ECE 497 class time**

#### **1.1 Preparation of system level functional requirements deliverable**

I corrected the system level functional requirements from the review we got. There was not much that needed to be changed.

#### **1.2 Preparation of subsystem level functional requirements deliverable**

Our next deliverable for ECE497 is the subsystem level functional requirements. We will start working on this once we finish up the system level functional requirements.



## 3 October 2016

We talked with Mr. Mattus today about the XBee adapters. He suggested we just use a USB to serial adapter with the interface board he gave us. He also said he would be able to solder them for us so the connections are better. We plan on doing this on Wednesday (10/5).





## 4 October 2016

### 1 Equations

Here are the equations from *roboticsNotes.pdf* on Google Drive:

#### 1.1 Control noise

$$\mathbf{Q} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\delta^2 \end{bmatrix} \quad (1)$$

#### 1.2 Observation noise

$$\mathbf{R} = \begin{bmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_B^2 \end{bmatrix} \quad (2)$$

where  $\sigma_B$  is the bearing noise standard deviation

#### 1.3 Vehicle Model

In the notes this is labeled as step #1

Continuous time model:

$$\dot{x} = v \cos(\theta + \gamma) \quad (3a)$$

$$\dot{y} = v \sin(\theta + \gamma) \quad (3b)$$

$$\dot{\theta} = v \frac{\sin \gamma}{\ell} \quad (3c)$$

where  $v$  is the linear velocity,  $\theta$  is the robot's orientation,  $\gamma$  is the robot's current steering angle, and  $\ell$  is the distance between the driving wheels of the robot.

Discrete time model:

$$x_{k+1} = x_k + Tv_k \cos(\theta_k + \gamma_k) \quad (4a)$$

$$y_{k+1} = y_k + Tv_k \sin(\theta_k + \gamma_k) \quad (4b)$$

$$\theta_{k+1} = \theta_k + Tv_k \frac{\sin(\gamma_k)}{\ell} \quad (4c)$$

Most of the variables in this model have the same meanings as the continuous model.  $T$  and  $k$  are the only new ones and they represent the time step and the iteration number, respectively.

This is labelled as step #2

$$\mathbf{G}_v = \frac{\partial f}{\partial q_k} = \begin{bmatrix} 1 & 0 & -Tv_k \sin(\theta_k + \gamma_k) \\ 0 & 1 & Tv_k \cos(\theta_k + \gamma_k) \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$\mathbf{G}_u = \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial v_k} & \frac{\partial f_1}{\partial \gamma_k} \\ \frac{\partial f_2}{\partial v_k} & \frac{\partial f_2}{\partial \gamma_k} \\ \frac{\partial f_3}{\partial v_k} & \frac{\partial f_3}{\partial \gamma_k} \end{bmatrix} = \begin{bmatrix} T \cos(\theta_k + \gamma_k) & -Tv_k \sin(\theta_k + \gamma_k) \\ T \sin(\theta_k + \gamma_k) & Tv_k \cos(\theta_k + \gamma_k) \\ T \frac{\sin \gamma_k}{\ell} & Tv_k \frac{\cos \gamma_k}{\ell} \end{bmatrix} \quad (6)$$

$$\mathbf{P}_{k+1} = G_v \mathbf{P}_k G_v^T + G_u Q G_u^T \quad (7)$$

## 1.4 Compute steering angle

This is labelled as step #3

$$\delta = \text{atan2}(\omega_{py} - y, \omega_{px} - x)$$

where the coordinates of the waypoint are  $(\omega_{px}, \omega_{py})$  and  $\delta$  is the angle from the positive  $x$ -axis and the waypoint at  $(\omega_{px}, \omega_{py})$

$$\Delta\gamma = \delta - (\theta + \gamma)$$

$\Delta\gamma$  should be wrapped so the value is within the range  $[-\pi, \pi]$ . However, the maximum allowable change of steering angle  $\Delta\gamma_{\max} = \dot{\gamma}T$  where  $\dot{\gamma}$  is the maximum allowable steering rate (given). Therefore, actual change of steering angle that needs to be applied is given by

$$\Delta\gamma = \text{sign}(\Delta\gamma) \min(\Delta\gamma_{\max}, |\Delta\gamma|) \quad (8)$$

Compute the final steering angle  $\gamma = \gamma + \Delta\gamma$ . If the maximum allowable steering angle is  $\gamma_{\max}$ , then the actual steering angle is also given by

$$\gamma = \text{sign}(\gamma)\min(\gamma_{\max}, |\gamma|) \quad (9)$$

## 1.5 Choosing a waypoint

This is still step #3

$d^2 = (\omega_{Px} - x)^2 + (\omega_{Py} - y)^2$  is the equation of a circle with the current waypoint at the center

If  $d^2 < (\text{minDistance})^2$ , choose next waypoint. If there are no more waypoints, stop moving and exit the algorithm.

## 1.6 Get observation

Step #4

$$z = \begin{bmatrix} \sqrt{(dx)^2 + (dy)^2} \\ \delta - \theta \end{bmatrix} \quad (10)$$

There is a figure that goes with this step. It can be found in the notes on Google Drive.

## 1.7 Data association

Step #5

The predicted measurement  $z_p$  is in the form,

$$\begin{bmatrix} z_p^j(1) \\ z_p^j(2) \end{bmatrix} = \begin{bmatrix} \sqrt{(dx_j)^2 + (dy_j)^2} \\ \text{atan2}(dy_j, dx_j) - \theta \end{bmatrix} \quad (11)$$

where  $dx_j = (x_j - x)$ ,  $dy_j = (y_j - y)$ , and  $(x_j, y_j)$  is the position of the  $j$ th feature already in the map, i.e. in the state vector  $q$ .

Observation Jacobian:

$$H = \begin{bmatrix} \frac{\partial z_p^j(1)}{\partial x} & \frac{\partial z_p^j(1)}{\partial y} & \frac{\partial z_p^j(1)}{\partial \theta} & \cdots & \frac{\partial z_p^j(1)}{\partial x_j} & \frac{\partial z_p^j(1)}{\partial y_j} \\ \frac{\partial z_p^j(2)}{\partial x} & \frac{\partial z_p^j(2)}{\partial y} & \frac{\partial z_p^j(2)}{\partial \theta} & \cdots & \frac{\partial z_p^j(2)}{\partial x_j} & \frac{\partial z_p^j(2)}{\partial y_j} \end{bmatrix} \quad (12)$$

Innovation

$$v = z^i - z_p^i$$

Normalized innovation matrix:

$$S = HPH^T + R$$

Normalized innovation square (nis):

$$nis = v^T S^{-1} v$$

Normalized distance (nd):

$$nd = nis + \ln(\det(S))$$

## 1.8 Update

Step #6

$H \equiv$  observation vector

$$S_{k+1} = H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1}$$

where  $S_{k+1}$  is the innovation covariance

Kalman gain:

$$K_{k+1} = P_{k+1}^- H_{k+1}^T S_{k+1}^{-1}$$

Updated state estimate:

$$q_{k+1}^+ = q_{k+1}^- + K_{k+1}(z_{k+1} - h_k(q_{k+1}^-))$$

We shall be using cholesky decomposition technique to determine the Kalman gain  $K$

$$PH_+ = P_{k+1}^- H_{k+1}^T$$

$$S = H_{k+1} PH_+ + R_{k+1}$$

$$S = 0.5(S + S')$$

$$\text{schol} = \text{chol}(S)$$

Kalman gain:

$$K_{k+1} = PH_+ (\text{schol})^{-1} (\text{schol})^{-T}$$

Using cholesky decomposition, it is easier to take inverse of big matrix  $S_{k+1}$  by inverting upper triangular matrix  $\text{schol}$

Update covariance estimate:

$$P_{k+1}^- = P_{k+1}^- - (PH_+(\text{schol})^{-1})(PH_+(\text{schol})^{-1})^T = P_{k+1}^- - PH_+(\text{schol})^{-1}(\text{schol})^{-T}(PH_+)^T$$

## 1.9 Augmenting new features in the map

Step #7

This step doesn't introduce any new equations, but it does present the algorithm with some pseudocode. It can be found in the last two pages of the notes on Google Drive.



## 6 October 2016

### 1 Weekly meeting to discuss progress with Dr. Miah

We had our meeting with Dr. Miah today. We went through the notes in detail so we understand exactly what we are trying to do in the project. Here is the list of things to work on:

- Keep working with the Xbees in the lab
- Start simulating the algorithm in V-REP. Use the the MATLAB to add noise to distance measurements from V-REP and the like
- Subsystem functional requirements

For work with the Xbee modules, Dr. Miah said we should consult Morgan Fields since he has been working with them for a couple months and might have some insight on what to do. We should just tell him when we plan on working in the lab so he can meet us there.

Since fall break starts tomorrow, we don't think we will be working on this much until after we get back to Bradley.





## 12 October 2016

### 1 Work completed during ECE 497 class time

#### 1.1 Preparation of subsystem level functional requirements deliverable

We have started working on the subsystem level requirements instead of working in the lab today. So far we only have the rough draft (in a Google Drive document). Once we figure out all the details we will write it in  $\text{\LaTeX}$  and draw all the figures and flowcharts. Hopefully we can get most of this done before the meeting tomorrow so Dr. Miah has a chance to look over it and we can address his comments before turning in the rough draft for review. This way we will be much better prepared and in theory, have less work to do to get it completed for the final draft.

For the block diagrams we will continue to use IPE<sup>1</sup>. At this point it might be a good idea to make a flowchart to include (to make it easier to understand data flow in each block). For these flowcharts we will use Dia<sup>2</sup>.

### 2 Work done in the lab for the day

#### 2.1 Working with the Xbee Modules

Since we must find the angle between the robot's current orientation and each beacon, we need to find a way to actually do that. Two possible methods include directional antennas [5] [2] and using phase shift data [4] [1]. Since we are trying to use as many off the shelf components as possible (while still being original with our implementation), I think we could implement a system similar to the one seen in [3] since it will give us the exact value we are trying to find. The technique is also already used by the military for localization and mapping, lending to its credibility as a localization method.

<sup>1</sup>For details. see <http://ipe.otfried.org/>

<sup>2</sup>See [www.wiki.gnome.org](http://www.wiki.gnome.org)

### **3 Weekly meeting to discuss progress with Dr. Miah**

We met in with Dr. Miah today. We went over what should be in the subsystem level functional requirements. He said that our main goal should be to make progress with the XBees and with V-REP.

## 13 October 2016

### 1 Notation

Throughout this document, the vectors (matrices) will be denoted by lowercase (uppercase) bold letters while the lowercase non-bold letters will denote scalar quantities. Sets will be denoted by calligraphic letters. For positive integers  $m, n > 0$ ,  $\mathbb{R}^n$  ( $\mathbb{R}^{m \times n}$ ) denotes  $n$ -dimensional column vector ( $m \times n$ -dimensional matrix) with entries taken from a set of real numbers  $\mathbb{R}$ .  $(\cdot)^T$  denotes the transposition of quantity  $(\cdot)$ . The standard Euclidean norm of the vector  $\mathbf{x} \in \mathbb{R}^n$  and the matrix  $\mathbf{A}$  are given by  $\|\mathbf{x}\| = (\sum_{i=1}^n |x_i|^2)^{1/2}$  and  $\|\mathbf{A}\| = (\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2)^{1/2}$  with  $x_i, a_{ij}$  being the entries of  $\mathbf{x}$  and  $\mathbf{A}$ , respectively.



## 16 October 2016

### **1 Work completed during ECE 497 class time**

#### **1.1 Preparation of subsystem level functional requirements deliverable**

I started working on the changes suggested by Dr. Miah in the subsystem level functional requirements. So far I have only moved around some sections and started writing the description of the subsystem blocks.



## 17 October 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Kyle worked on getting two of the XBees communicating with each other. He was able to set up a circuit to power both and then connect to them using two serial-to-usb cables. They were set up but unable to be tested due to computer problems. He will try again with his other laptop tomorrow since it is more powerful. The setup for our testing can be seen in [Figure 1](#).

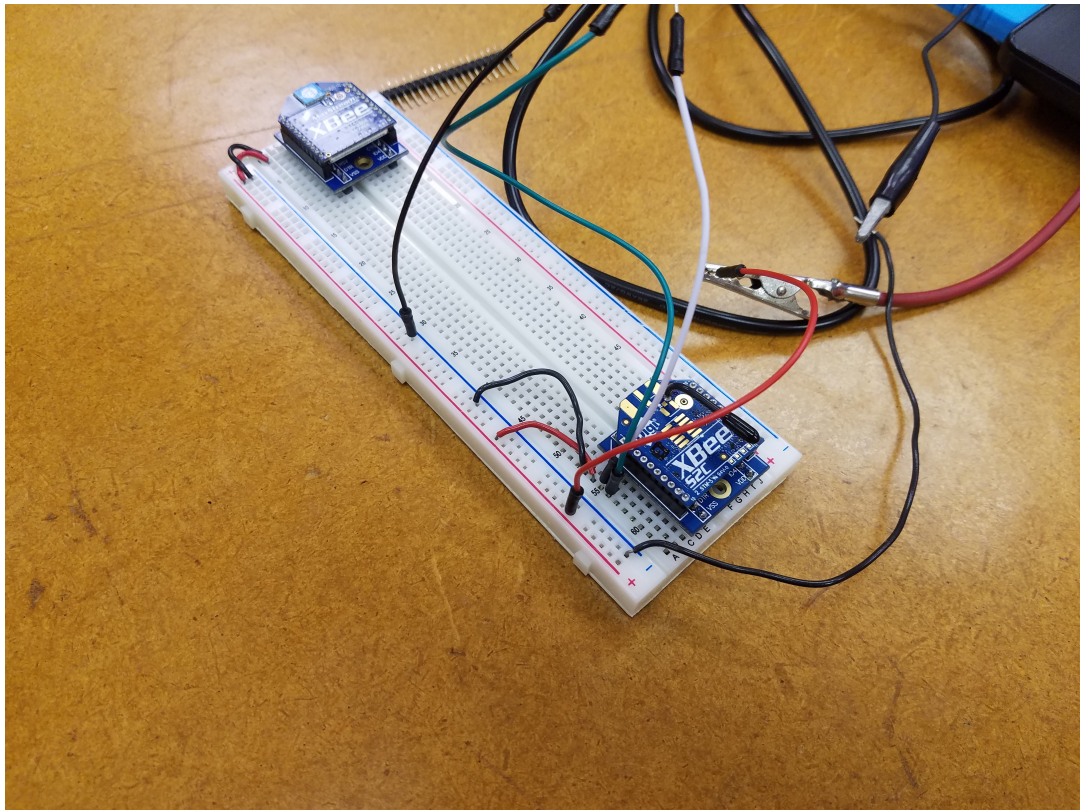
#### 1.2 Simulating the algorithm using V-REP

I started converting the MATLAB code to be used in V-REP. It is taking much longer than I expected since many of the functions are nested and the code is not well documented. I was only able to set up the V-REP scene and the beginning of the program in MATLAB. Another source of difficulty is the fact that the algorithm uses a vehicle model that is not a differential drive model, so I am not very familiar with the model being used.

### 2 Work completed during ECE 497 class time

#### 2.1 Preparation of subsystem level functional requirements deliverable

I worked on the related works, block diagrams, and some of the descriptions. The only things left to finish are the flowcharts, their descriptions, and the description of the subsystem block diagram.



**Figure 1:** XBee setup for testing communication



## 18 October 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Kyle was able to communicate between two XBees using the setup from yesterday. He was also able to get the RSSI using the *ATDB* command. He did another setup with the XBees placed farther apart (so the RSSI value would change) but there was a problem with one of the XBees. It got very hot and would not connect to XCTU anymore. We had been having trouble with it before but up until that point it had been working fine. We are unsure of the cause, whether it be a wrong connection or just overall failure. Either way we only have one Xbee with the external antenna now. It is not a terrible problem since they are not that expensive but we will still need to get more for the final implementation.



## 19 October 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Since Kyle was able to communicate between two Xbees yesterday, he is starting to look into how to get the IDs and RSSI data from them. Since we want the beacons to be very simple, we do not want to have them being controlled by a microcontroller or microcomputer. Ideally they will just be battery powered so when the battery dies they can just be replaced. Right now, it looks like there is a way to use the API mode to send remote AT commands. When using this style of commands, the response is sent to the coordinator over the wireless connection instead of through UART.

#### 1.2 Simulating the algorithm using V-REP

I continued working on the V-REP simulation. I ended up having to rewrite all the MATLAB code so it would be easier to convert to V-REP. I also added comments to every line so it is more easily understood. Using this code, I started adding V-REP functions.



## 20 October 2016

### **1 Weekly meeting to discuss progress with Dr. Miah**

We had a short meeting with Dr. Miah today. We did not have anything to show him so he asked us to just keep working on the implementation and simulation.



## 21 October 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Kyle continued to research how to use the API mode today. He found some code online that was made to interface an XBee module with the BeagleBone Black. We will write our own program but this will be a good starting point or reference.

#### 1.2 Simulating the algorithm using V-REP

I have the robot driving in V-REP but the trajectory is wrong. I am having trouble converting from the steering vehicle model to the differential drive model.

I talked to Dr. Miah to see if he could help find what my error was. It turns out that my equations were right but in the estimation step I wasn't using the correct kinematic model. I will fix this and upload the version to Google Drive so Dr. Miah can look at it.





## 22 October 2016

### 1 Work done in the lab for the day

I did not go to the lab to work since it was only simulation but it was still technically lab work.

#### 1.1 Simulating the algorithm using V-REP

I fixed the equations Dr. Miah pointed out but the simulation still doesn't work quite right. The noisy trajectory is way to far off to be just from noise so I think I have some wrong parameters. I emailed Dr. Miah to inform him of this.

### 2 New uploads for the day to Google Drive

- simulationV2.zip



## 24 October 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Kyle was able to send remote commands to one XBee and receive their response on another using the API mode. We were able to get the RSSI but only after sending a message and then requesting for the signal strength using the remote command. We think it might be possible to get all the data we want in one packet but we cannot find any clear way to do it. Most tutorials online use two or three packets to accomplish it. It is difficult for us to test this since we only have two XBee interface boards. On Wednesday we will talk to Mr. Mattus to see if we can get another interface board to test multiple XBees.

We also started looking into how to interface the BeagleBone Black more. As expected, the code did not work without some modification. We had to make some changes just to get it to compile. When we ran it on my BeagleBone Black, the program could not find the correct files because they were in different locations altogether. We are going to look into this more on Wednesday. I think we should be able to get it working by modifying the file locations in the code so they are correct.



## 25 October 2016

### 1 Weekly meeting to discuss progress with Dr. Miah

This isn't our actual weekly meeting with Dr. Miah but we did meet with him today to go over the MATLAB code. He pointed out some errors with the model I was using in the code. He also explained more about how the code is working to implement EKF-SLAM. I corrected the things he pointed out and uploaded it as *simulationV3.zip* on Google Drive.

### 2 New uploads for the day to Google Drive

- *simulationV3.zip*



## 26 October 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

We worked more on the code that we tried on Monday. We were able to get it to compile on the BeagleBone Black. As a test, we ran the program without anything connected to see what would happen. To our surprise it said that it send and received a message. We think this is a good sign that it will work.

We connected one of the XBees to the BeagleBone and started the program. This time the program got stuck before right after it said it initialized the UART ports. We has to force close the program to get out of it. We tried again just to be sure and got the same result. I think it might be a good idea for us to just write our own UART function library so we know it works with our BeagleBone Black. I am not familiar with UART on the BeagleBone so we will definitely need to do some research first.

#### 1.2 Simulating the algorithm using V-REP

I am still working on the V-REP simulation. I think there are other equations wrong in the code since the algorithm still doesn't seem to be very accurate. I have gotten too confused with my current version of the simulation so I started again at version 1. I called this version *simulationV1R1*, it stands for version 1 revision 1. This version is the same as version 1 with all the corrections given by Dr. Miah.

I changed the equation used for the EKF predict step to the differential drive model. It doesn't make sense that we are using a different model to predict the SLAM state when the robot we are using uses a completely different one. When I changed this there was a noticeable improvement in the algorithm's performance. I also tweaked the maximum detection range since 30 m is large relative to the size of our environment. After doing this, I noticed that the issue with extra features being added to the map became less frequent.

$$x_{k+1} = x_k + v * T * \cos(\theta_k + \frac{T * \omega_k}{2})$$

$$y_{k+1} = y_k + v * T * \sin(\theta_k + \frac{T * \omega_k}{2})$$

$$\theta_{k+1} = \theta_k + T\omega_k$$

The problem with extra features being added is back again. I think the error for the previous runs of the simulation were just low enough that it didn't happen or something. I tried finding the cause of the issue but I have only been able to narrow it down to the data association step. The code isn't commented very well there so I do not really know what is happening.

I converted version 1 revision 1 to work with V-REP. This version is version 3 revision 1 since V-REP is in version 3. I don't like how I'm organizing versions so I will try to figure out some way to organize them differently. This new revision of version 3 works even less than the other one. I think its because in V-REP, the motors on the robot are controlled using a PID controller. In the MATLAB simulation, it is an instantaneous velocity for the simulation. That means the estimate of the next iteration isn't anywhere close to correct after about 10 iterations. I will have to think of a way to get around this.

## 2 New uploads for the day to Google Drive

- *simulationV1R1.zip*
- *simulationV3R1.zip*



## 27 October 2016

### 1 Work done in the lab for the day

#### 1.1 Simulating the algorithm using V-REP

I worked on version 1 revision 1 some more today. I am reasonably happy with how the algorithm is working so I am really just adding some plotting to the code. I added plotting for only dead reckoning so you can see how well the localization is working. I also added a plot showing the  $(x, y, \theta)$  for each of the three cases (perfect odometry, EKF-SLAM, and only dead reckoning). I emailed Dr. Miah about this version so he could try to look at it before the meeting.

### 2 Weekly meeting to discuss progress with Dr. Miah

We had our weekly meeting today. I spent way too much time on the simulation because it ended up being wrong. The equations I replaced with the differential drive kinematic model (specifically for state prediction) were wrong. In addition, I used the Runge-Kutta integration method instead of the actual model. I don't know why it worked to be honest. Dr. Miah showed me the equations that they should be so I can replace them. I explained to him how I am getting stuck on the V-REP since I have to simulate with a differential drive robot. He said he will sit down with me once I start working on the V-REP so he can help explain things better.

We also discussed the DRT's review of our subsystem level functional requirements (the review is uploaded on Google Drive). He agreed with most of their comments and asked us to address them and email him once they are fixed so we can make any other final changes.

For next week, we need to have a demonstration ready for him showing the communication between two Xbees. I think it would be a very good idea to have something working with the BeagleBone as well. I want to have the V-REP completed by the next meeting as well.

### **3 Work done in the lab for the day**

#### **3.1 Simulating the algorithm using V-REP**

I corrected the mistakes brought up by Dr. Miah. The new revised MATLAB code is in *simulationV1R2.zip* in the simulation folder on Google Drive.

### **4 New uploads for the day to Google Drive**

- *simulationV1R1.zip*
- *simulationV1R2.zip*
- *Robo Localization using Zigbee-Subsystem Review.pdf*

## 28 October 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

We were able to use the XBee's UART 2 port to communicate with another computer over serial. Since we had issues using the program we found online, we decided it would be a good idea to something simple to make sure the BeagleBone was actually working. To setup UART 2, we used the following commands in Ubuntu 16.04 LTS:

```
sudo su
echo BB-UART2 > /sys/devices/platform/bone_capemgr/slots
```

We then verified that this worked by using the `cat` command to show that the port was enabled. I added the following line to the `/boot/uEnv.txt` file so UART2 would be automatically setup when the BeagleBone boots.

```
cape_enable=bone_capemgr.enable_partno=BB-UART2
```

Using `minicom` on the BeagleBone Black, we were able to communicate over the serial connection.

We are setting up the XBees to work in transparent mode so we can see if we can communicate using them now. The communication works when the XBees are in transparent

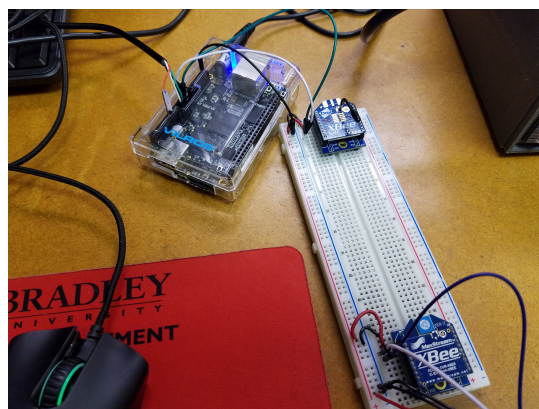


Figure 1: Experimental setup using two XBee modules

mode. For future reference:

- Transmit on BBB goes to Din on XBee
- Receive on BBB goes to Dout on XBee

We were able to use the `+++` command to enter AT mode, then give the `ATDB` command to return the RSSI data. With the two XBees about 6 inches apart, we got an RSSI value of 19. We are pretty sure this value is in hexadecimal but we aren't really concerned with value, only that we were able to get the value in the first place. Right now we are setting up the XBees to be in AP mode so we can try some remote commands. We will also put the XBee connected to the BBB farther away on a separate breadboard to see if the RSSI changes at all.

In AP, mode we sent AT commands from XCTU. First we sent the `ND` command to get the nodes on the network. This command actually didn't give us the information we wanted, since it would only respond with the acknowledgement "OK". We changed the command to `SL`, which will return the lower half of the remote XBees serial number.

Using this method, we were able to send both AP commands from Kyle's laptop (using XCTU) to the BBB, which acted as a beacon. We were unable to do this in the other direction. We think it is something to do with how XCTU converts the packets for sending (something we wouldn't be able to do on the BBB without knowing what to do). We will work writing UART functions and a program to test some functionality from BBB on Monday.

Here are AP packets we were sending:

For getting the serial number,

```
7E 00 04 08 01 53 4C 57
```

For getting the RSSI,

```
7E 00 04 08 52 44 42 1F
```

I think enabling UART2 in `uEnv.txt` messed something up on my BBB. When I went to shutdown, it gave me an error saying it could not find the correct way to shutdown and aborted. It isn't really a huge issue since I can just reflash the eMMC but it is something to be careful of when we are implementing later on. It will probably be best to figure out how to enable UART2 when the mapping and localization program is started and then disable it when it closes.

I will probably put Ubuntu 14.04 LTS on my BeagleBone Black since that is the last version of Ubuntu that the ROS package for the Pioneer will run on. Luckily, this version also has the cape manager.

## 31 October 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

I forgot to install Ubuntu over the weekend so I am doing that now. After installing Ubuntu 14.04, Kyle began working on the UART functionality again. Again, it says it is able to initialize everything but when we send a message nothing is received.

#### 1.2 Simulating the algorithm using V-REP

I took a break from working with V-REP since I was having so much trouble with the simulation. I think I know how I am going to simulate with V-REP now though.

- I won't have an idea robot in V-REP, just the robot driving with the noisy control values
- Somehow I will add noise to the pose from the robot driving with the noisy control values, to simulate noisy encoders (this will be like the dead reckoning robot)
- Using the "dead reckoning" pose, I will implement EKF-SLAM. The pose information from this version will be used for the navigation in V-REP.

Most of the issue comes from the fact that V-REP is too accurate about some aspects of the simulation (PID control for motors, perfect position data, etc). Another thing I realized is that having steering angle noise doesn't really make sense in the simulation since there is no steering angle for our robot. It makes more sense to have noise for the linear and angular velocities instead. I will need to figure out a way to add encoder noise since V-REP does not have any way to implement encoders currently. I think I could do it by finding the distance traveled for the current timestep, adding some small noise to that, then recalculate the pose. The recalculated pose would be separate from the one we get from V-REP and the pose from V-REP will only be used to update the noisy encoders.

I changed the code in *simulationV1R2.zip* and added the V-REP parts. This version has the V-REP robot as *qTrue*, then some noise added to the position (*qNoisy*), and no plot for just dead reckoning. Overall, it works decently well. The robot drives in V-REP and you can

see its augmented state in MATLAB. There is still a problem with the data association. I know now that it is with how association is calculated (Maholonobis distances). In reality, we are able to associate with exact accuracy since we will be getting the serial number of the beacon with each measurement. I want to talk to Dr. Miah and see if I can just change the data association functions to take advantage of this.

## **2 Work completed during ECE 497 class time**

### **2.1 Preparation of subsystem level functional requirements deliverable**

I corrected the comments from the DRT's review of our subsystem level functional requirements. Most of the comments were about discrepancies in wording. I corrected those so they were more clear and updated the figures accordingly. I also moved some of the figures around so all the flowchart steps are together. I uploaded this version on Google Drive.

## **3 New uploads for the day to Google Drive**

- *simulationV4.zip*
- *subsysLevelFunctionalRequirementsV4.zip*

# 1 November 2016

## 1 Work done in the lab for the day

### 1.1 Working with the Xbee Modules

Kyle is continuing to work on the UART on the BeagleBone Black. We were testing to see if anything was being sent by attaching an oscilloscope to read the signal. We used the oscilloscope in the cabinet. Mr. Schmidt came in and pointed out that the probes we were using aren't actually scope probes. He brought us an oscilloscope (EQ-2352) with actual probes and took the oscilloscope we were using before. Using Kyle's modified code, we are now able to see an output coming from UART2 on the BeagleBone Black microcomputer. The image in [Figure 1](#) is the reference we are using for connections to the BeagleBone Black.

### 1.2 Simulating the algorithm using V-REP

Dr. Miah gave us a better way of representing the error in our new model. Since we are using a differential drive mobile robot, it does not make sense to have a steering angle covariance (as previously presented). A typed out version of the notes he gave us is below.

Let  $v_k = v_k^d + \xi_{1,k}$  and  $\omega_k = \omega_k^d + \xi_{2,k}$ , where

$$\xi_k = \begin{bmatrix} \xi_{1,k} \\ \xi_{2,k} \end{bmatrix} \sim \mathcal{N}(0, \mathbf{Q}) \quad (1)$$

$$\mathbf{Q} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} \quad (2)$$

Thus, the problem is to find  $\sigma_v$  and  $\sigma_\omega$ .

**Solution:** Suppose that the robot is equipped with a wheel encoder and an angular rate gyro. In addition, let us assume that the wheel has an  $N$ -bit encoder and a radius of  $r$  [m]. Suppose that at the  $k$ th sampling time instant, we measure the wheel displacement  $d_k + n_k$  [m], where  $n_k$  is the noise.

Therefore, the approximate speed (linear) is determined by backward difference approximation:

## 4.5 serial UARTs

P9				P8			
DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	GPIO_38	3	4	GPIO_39
VDD_5V	5	6	VDD_5V	GPIO_34	5	6	GPIO_35
SYS_5V	7	8	SYS_5V	GPIO_66	7	8	GPIO_67
PWR_BUTTON	9	10	SYS_RESETN	GPIO_69	9	10	GPIO_68
UART4_RXD	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
UART4_TXD	13	14	GPIO_40	GPIO_23	13	14	GPIO_26
GPIO_48	15	16	GPIO_51	GPIO_47	15	16	GPIO_46
GPIO_4	17	18	GPIO_5	GPIO_27	17	18	GPIO_65
UART1_RTSN	19	20	UART1_CTSN	GPIO_22	19	20	GPIO_63
UART2_TXD	21	22	UART2_RXD	GPIO_62	21	22	GPIO_37
GPIO_49	23	24	UART1_TXD	GPIO_36	23	24	GPIO_33
GPIO_117	25	26	UART1_RXD	GPIO_32	25	26	GPIO_61
GPIO_125	27	28	GPIO_123	GPIO_86	27	28	GPIO_88
GPIO_121	29	30	GPIO_122	GPIO_87	29	30	GPIO_89
GPIO_120	31	32	VDD_ADC	UART5_CTSN+	31	32	UART5_RTSN
AIN4	33	34	GND_ADC	UART4_RTSN	33	34	UART3_RTSN
AIN6	35	36	AIN5	UART4_CTSN	35	36	UART3_CTSN
AIN2	37	38	AIN3	UART5_TXD+	37	38	UART5_RXD+
AIN0	39	40	AIN1	GPIO_76	39	40	GPIO_77
GPIO_20	41	42	UART3_TXD	GPIO_74	41	42	GPIO_75
DGND	43	44	DGND	GPIO_72	43	44	GPIO_73
DGND	45	46	DGND	GPIO_70	45	46	GPIO_71

There is a dedicated header for getting to the UART0 pins and connecting a debug cable. Five additional serial ports are brought to the expansion headers, but one of them only has a single direction brought to the headers.

Figure 1: UART pinout for the BeagleBone Black

$$v_k \approx \frac{1}{T}(d_k - d_{k-1})$$

The displacement (in meters) due to one count of wheel encoder is given by

$$\frac{2\pi r}{2^N} \text{ [m]}$$

Therefore, the resolution of the wheel encoder is

$$\frac{2\pi r}{2^N} \text{ [m]}$$

The quantization error in displacement is given by

$$e_d = \pm \frac{1}{2} \times \text{resolution}$$

$$e_d = \pm \frac{1}{2} \times \frac{2\pi r}{2^N}$$

$$e_d = \pm \frac{\pi r}{2^N}$$

Suppose that 99.99% of the time the displacement estimate lies within  $\pm \frac{\pi r}{2^N}$ . Therefore,  $3\sigma_d = \frac{\pi r}{2^N} = e_d$ , where  $\sigma_d$  is the standard deviation of the displacement noise  $n_k \sim \mathcal{N}(0, \sigma_d^2)$ .

$$\sigma_d = \frac{1}{3} \frac{\pi r}{2^N}$$



But the noise associated with  $d_k - d_{k-1}$  has the distribution  $\mathcal{N}(0, 2\sigma_d^2)$  since  $E((n_k - n_{k-1})^2) = E(n_k^2) + E(n_{k-1}^2) = 2\sigma_d^2$  assuming  $n_k$  and  $n_{k-1}$  are uncorrelated.

Now, mean

$$\begin{aligned} v_k^d + \xi_{1,k} &\approx \frac{1}{T}(d_k + n_k - d_{k-1} - n_{k-1}) \\ E[v_k^d + \xi_{1,k}] &= \frac{1}{T}E[d_k + n_k - d_{k-1} - n_{k-1}] \\ E[v_k^d] + E[\xi_{1,k}] &= \frac{1}{T}(E[d_k] - E[d_{k-1}] + E[n_k] - E[n_{k-1}]) \\ E[\xi_{1,k}] &= \frac{1}{T} \times 0 = 0 \end{aligned}$$

variance

$$\begin{aligned} E[(v_k^d + \xi_{1,k})^2] &= \frac{1}{T^2}E[(d_k + n_k - d_{k-1} - n_{k-1})^2] \\ E[\xi_{1,k}^2] &= \frac{1}{T^2}E[(n_k - n_{k-1})^2] \\ \sigma_v^2 &= \frac{1}{T^2}2\sigma_d^2 \end{aligned}$$

Suppose that the angular rate gyro is able to measure the angular rate of the robot with noise

$$\xi_{2,k} \sim \mathcal{N}(0, \sigma_\omega^2)$$

Example: If the manufacturer specifies that 99.99% of the time angular rate estimate lies within  $\pm 0.2^\circ/\text{s}$ , then

$$3\sigma_\omega = 0.2 \rightarrow \sigma_\omega = \frac{0.2}{3} [^\circ/\text{s}]$$

Therefore,

$$\sigma_\omega^2 = \left(\frac{0.2}{3} \frac{\pi}{180}\right)^2 [\text{rad}/\text{s}]$$

Now,

$$\mathbf{Q} = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} \quad (3)$$

I was not able to add any of this new error modeling to the MATLAB code. I plan on working on this tomorrow when we are working in the lab again.



## 2 November 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Kyle was able to send a message from the XBee connected to the BeagleBone Black and receive it in XCTU on his laptop. He also added the reading function back into the program. Data sent from XCTU is able to be received by the BeagleBone Black. The data is all garbled up but we are pretty sure that is just because of some timing issues between us sending then receiving. We will probably have to implement some sort of flow control for UART so all the data can be received correctly every time.

Originally, the UART2 port was enabled before the program was started. Kyle added code to the program so UART2 is automatically started with the program.

#### 1.2 Simulating the algorithm using V-REP

I have added the new error modeling in the MATLAB code for the vrep simulation. The simulation still behaves more or less the same but the model is a bit more accurate now. Right now I am just going through the code and updating/fixing comments so they are correct to this version of the code.

I also took out some of the plotting for the dead reckoning only pose since I wasn't using it anymore. But, now the path following is kinda weird. It weaves around a lot more than it did before. I think it's because I changed it so the noisy pose is used instead of the true pose like it was before (since it didn't make sense to use the perfect pose when in reality, the robot would go off the augmented pose).



## 3 November 2016

### 1 Weekly meeting to discuss progress with Dr. Miah

For this week's meeting we are showing the progress we have made in the implementation to Dr. Miah in the lab. So really just the communication from the BeagleBone Black to the laptop running XCTU over the XBees. I might show what I have in V-REP so far if there is time but I think the XBee communication is good enough progress.

### 2 Work done in the lab for the day

#### 2.1 Simulating the algorithm using V-REP

I need to figure out how to add a gyroscope sensor to the Pioneer model in V-REP. There is one already made but I do not know how to interface it so I can use the remote API to get the data.

#### 2.2 Working with the Xbee Modules

We found that the two commands we thought we could use to get the serial number and RSSI actually just return the RSSI of the main XBee and not the remote ones. We looked through the data sheet and found that we could use the node discover command instead, since it returns the serial number and RSSI of all modules within range. This should make it easier than sending two commands each time.



## 4 November 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Kyle is working on getting the program we have working with API commands, as a test. He was able to use XCTU to send the node discover command and get a response from our one beacon XBee with the information we needed.

Once he started trying to do it in C++ there were issues. We do not know the exact format the packets sent by XCTU are in. It gives a format to follow but when we use that format nothing happens. We connected a debug cable in parallel with the XBee's input so we could see exactly what XCTU is sending to the XBee. We were surprised because it is nothing like the format given in the datasheet. This could be caused other things like noise or connecting incorrectly instead of a wrong datasheet. We will spend more time on it on Monday.

#### 1.2 Simulating the algorithm using V-REP

I started working on the V-REP simulation again using two Pioneer models (one for reference and the other for the noisy driving). It really isn't difficult but some parts are confusing about what model is used for which purpose, how to correct for offset, etc. I have the scene set up with two robots and I connect to them to get handles and poses. That is about as far as I got before helping Kyle brainstorm ideas for the XBees. I will work more on Monday.

### 2 Working on the conference paper

We asked Dr. Miah when he wants to submit the first conference paper. He told us that the deadline for the first submission is November 21. The content of the paper is mainly just real-time simulation results and the beginnings of some hardware implementation. I don't think it should be an issue to get it done in time but that is still only about two weeks to write the paper.

Dr. Miah said I should finish the simulation as soon as possible then switch to helping Kyle with the hardware. He will send me a  $\text{\LaTeX}$ template for the paper and said it would be similar to another paper he had already written. We should still put in a lot more work to make sure we can get as much done as possible though.



## 7 November 2016

### 1 Work done in the lab for the day

We were going to work in the lab today but Dr. Miah was not in his office. We decided to just work on stuff on our own.

#### 1.1 Working with the Xbee Modules

Kyle found a GitHub repository for the XBees. I don't know if it will work for our purposes but there are a lot of good examples and functions we can at least use for reference.

#### 1.2 Simulating the algorithm using V-REP

I worked on simulation version 5. I think I am almost done with it.



## 8 November 2016

### 1 Weekly meeting to discuss progress with Dr. Miah

This isn't our weekly meeting but we met with Dr. Miah today to discuss the paper we need to finish by November 21st for submission to the conference. Below is a list of the things we need to change.

- Rewrite the introduction
- Rewrite the abstract
- Find more recent references (within the past 2 years) from ISIE
- Update the references to our sources
- Add simulation results
- Add some experimental results
- fix the author section so it has the proper names

### 2 New uploads for the day to Google Drive

- *simulationV5.zip*



## 9 November 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

The XBee library that Kyle found will work for our implementation. He was able to run the functions on the BeagleBone when they are connected to an XBee and they work very well. We did some quick distance measuring (up to  $\cdot 5\text{m}$ ) and the RSSI did change slightly. This is a good sign for the implementation. We will probably try to get another XBee interface board from Mr. Mattus on Friday so we can try with multiple XBees. We will also probably test over larger distances as well.

#### 1.2 Simulating the algorithm using V-REP

I kept working on the V-REP simulation. Its not really version 5 but I'm just making modifications to the code. The biggest one I have made is calculating the steering angles for the reference robot separate from the noisy robot (since I think that is why there is always an offset). The other changes are just for plotting to make it look better. I don't know if I will upload this code yet.



## 10 November 2016

### 1 Weekly meeting to discuss progress with Dr. Miah

We met with Dr. Miah today. There was not much to discuss. We mainly just went through my simulation code and found every thing that I did wrong again. I will fix it again and upload to Google Drive as usual.

### 2 Work done in the lab for the day

#### 2.1 Simulating the algorithm using V-REP

I made all the corrections to the simulation code. These include but are not limited to:

- Clearing up some variable names
- Fixing how the offset is handled
- Adding a visible path to the V-REP scene
- Adding a dummy to the V-REP scene to represent the offset reference robot (it moves along the visible path), this makes it easier to see things working in V-REP

These changes were made to *simulationV5* and re-uploaded to the Drive. I also added the same changes/fixes/modifications to the first revision of version 5

### 3 New uploads for the day to Google Drive

- *simulationV5.zip* - This is the fixed version
- *simulationV5R1.zip* - Has the same modifications as the base version but steering angles are calculated seperately for each robot





## 11 November 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Today we just did some experiments to see how RSSI changes with distance. We placed the beacon XBee on the floor (powered by a power supply) and moved the transitting XBee away from it, recording the RSSI at a regular interval (30cm). For the experiment, we powered the transmitting XBee off the BeagleBone Black, which was being powered over Kyle's laptop USB.

For the first test, we moved from one windowed wall to the other (about 5m). For this test, we noticed that the measurements were not linear at close to mid-range. We did some research and found another group's results that showed very little linearity at close range. The plot of the RSSI can be seen in [Figure 1](#).

As you can see in [Figure 1](#), the values we obtained in our experiment were not very linear with distance. However, using the curve fitting toolbox in MATLAB, a linear fit was found. This is the solid red line in the figure. Even though our points are not clearly linear, they are decreasing with distance, which is the desired response.

For the next experiment, we decreased the power level of the XBee to the lowest value,  $PL = 0$ . We did this with the hopes that it will give us more sensitivity in a smaller range. We tested this setup but this time started at the far wall and moved toward the door to the lab. We noticed a more linear response at close to mid range but when the XBee was near the 2 metal cabinets the values would change drastically. We are pretty sure this is just interference from the cabinet. This noise would not be as noticeable since our beacons will be on the ceiling. We only changed the power level for the main XBee (on the BeagleBone Black) so it is possible we may get better values if we decrease the power level on the beacon as well.

With the onboard chip antenna, we also noticed that the signal was stronger when the XBee's antenna was pointing **away** from the main XBee. We found other data online that supported this result as well<sup>1</sup>. We are unsure why it works better this way and don't think

<sup>1</sup>[https://www.ece.lsu.edu/scalzo/FDR\\_Final.pdf](https://www.ece.lsu.edu/scalzo/FDR_Final.pdf)

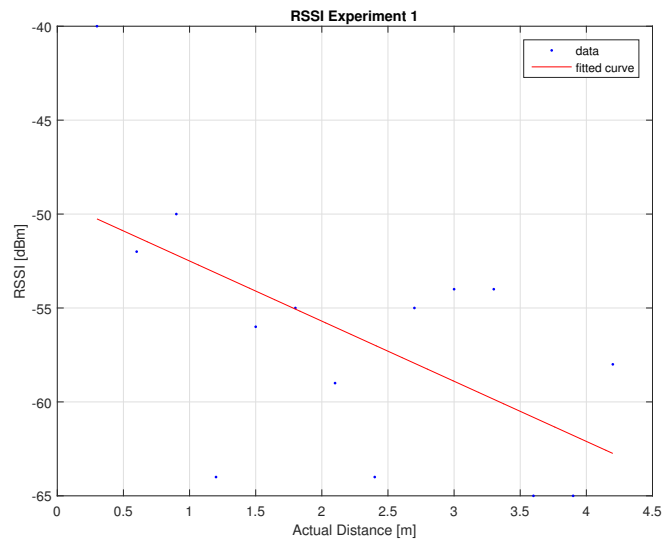


Figure 1: RSSI data for our first test

it will have any specific impact on our implementation. In that document we also found that they used a removable antenna for their beacons, which increased the sensitivity greatly. This may be good for us since we want more sensitivity. We would just need to buy an XBee module that accepts a removable antenna and a removable antenna to use with it, neither of which is terribly expensive.

Once we get another interface board we will probably run these tests again with multiple XBees.

## 2 Working on the conference paper

I started to work on the ISIE 2017 paper using the template provided by Dr. Miah. The due date was originally November 21 but it was changed to December 15. We have kinda shifted this to the back burner for now so we can work on implementation some more. We are also starting to work on the ECE 498 product proposal.

So far in the paper I have only started the abstract. I expect the introduction section to be very similar to the related works section we did for the subsystem level functional requirements.

## 14 November 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Now that we have an XBee library that works pretty well we are looking at how to produce a PWM signal with the BeagleBone Black. We have found many snippets of code for doing it but they all require specific device tree overlays in Linux, which are apparently not in Ubuntu 14.04. We have a couple of options:

- Find the device tree overlays
- Write our own device tree overlay (probably combining UART and PWM in the process)
- Revert back to a kernel version where the device tree overlays are included in the kernel

Since we already have the UART communication working, I'd like to avoid changing Linux versions so we don't lose any of our progress so far.



## 15 November 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

Kyle went to go get another XBee interface board today. We decided to just wait on figuring out the PWM until we have all the XBee stuff sorted out. We realized that it may also be possible to use a microcontroller to do the PWM instead of the BeagleBone Black. The only problem if we do it this way is that the BeagleBone will not directly know what the angle of the servo is. Our assumption may be wrong but it is still something we need to think about.

### 2 Working on the conference paper

I sent Dr. Miah a list of references that would be good for the ISIE paper. He uploaded them to Google Drive so we can start reading them. I think they will most likely be used in the introduction section (mainly) although we may also be able to reference their methods when we are talking about our setup or our calculations.

This paper is not really the preference right now since we need to finish the product proposal, since our presentation is after Thanksgiving Break...

### 3 Work completed for ECE 498

#### 3.1 Project Proposal Presentation Preparation

I started writing out some notes for the product proposal. Dr. Miah will send us a template for the presentation portion.



## 16 November 2016

### 1 Work done in the lab for the day

Today we did not work on any sort of implementation and instead focused on coming up with a parts request list to send to Mr. Mattus. Most of the stuff on the list is for making the beacons:

- 3.3V regulators
- 9V battery clips
- Perforated circuit board
- XBee S2C modules

Having most of the parts we need will make it easier for us to generate the parts list for the product proposal.





## 17 November 2016

Weekly meeting moved to tomorrow.



## 18 November 2016

Weekly meeting moved to Monday.

### **1 Work done in the lab for the day**

I emailed the parts request to Dr. Miah. He said everything looks good but said we should try to order everything at once. That means we need to either find a stepper motor or servo that we can use for our project. I will work on this over the weekend.



## 19 November 2016

### 1 Work done in the lab for the day

After doing a little research I have found out a few things:

- Stepper motors need a controller
- Stepper motors are better for lower RPM applications
- You do not need PWM to run a stepper motor
- Stepper motors are cheaper than servos

Based on these things, I think it is in our best interest to use a stepper motor instead of a servo (also because servo motors are expensive even for small ones). There are a couple things I think we should look for in the stepper motor:

- The voltage should be 12V, that way it can be powered off the Pioneer's battery directly, instead of the BeagleBone Black
- A lower step angle will allow us get more data and hopefully find the angle of arrival with more accuracy
- I am pretty sure we need a bipolar motor
- We do not need high torque since it won't be lifting anything, only rotating

Using these criteria, I found the following motors that would fit our needs:

- <http://www.digikey.com/product-detail/en/sparkfun-electronics/ROB-09238/1568-1105-ND/5318747>
- <http://www.digikey.com/product-detail/en/adafruit-industries-llc/324/1528-1062-ND/5022791>
- <http://www.digikey.com/product-detail/en/portescap/42M048C2B-R21/403-1028-ND/806356>

We will also need a stepper motor controller that meets these requirements:

- Able to control in both directions

- Can use the 12V power supplied by the Pioneer
- Inexpensive

A lot of the motor controllers I found on [digikey.com](http://www.digikey.com) were quite expensive. There were a couple less expensive ones though.

- <http://www.digikey.com/product-detail/en/sparkfun-electronics/ROB-12859/1568-1066-ND/5172292>
- <http://www.digikey.com/product-detail/en/trinamic-motion-control-gmbh/TMCM-1070/1460-1177-ND/6207048>

Both of these look like they will work for our project but I think the second one looks best for us. We will need to go over all of this with Dr. Miah on Monday so we can finish the parts request and send it to Mr. Mattus before we go on Thanksgiving Break.

## 21 November 2016

### 1 Work done in the lab for the day

#### 1.1 Working with the Xbee Modules

We redid the XBee ranging experiment today. We tried the three lowest power levels to see if affected the values at all. The plots for these can be seen in [Figure 1](#) and [Figure 2](#). [Figure 1](#) was converted to [Figure 2](#) using [Equation 1](#).

$$d = 10^{\frac{-\text{RSSI} - P_{\text{ref}}}{10\eta}} \quad (1)$$

where  $d$  is the calculated distance in meters, RSSI is the measured signal strength in  $dBm$ ,  $P_{\text{ref}}$  is the absolute value of the RSSI at 1 m, and  $\eta$  is the propagation constant. For our experiment we found that  $P_{\text{ref}} = 60$  and  $\eta = 4$ .

### 2 Weekly meeting to discuss progress with Dr. Miah

We met with Dr. Miah today instead of Thursday since we will be gone for Thanksgiving Break. All we did was discuss the stepper motor and controller a bit, then the content of the project proposal, which we will be giving on November 29. We also need to send the part request to Mr. Mattus.

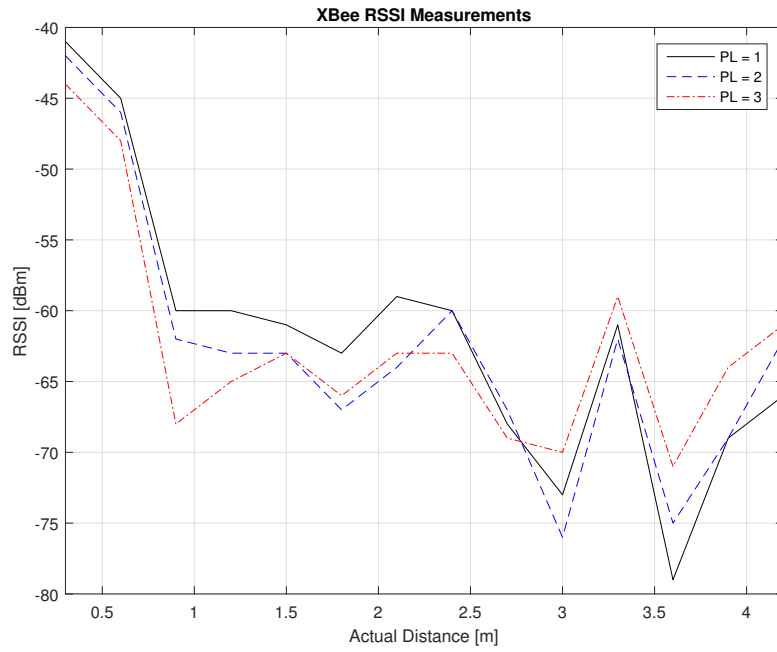


Figure 1: XBee ranging data plotted as RSSI vs. actual distance

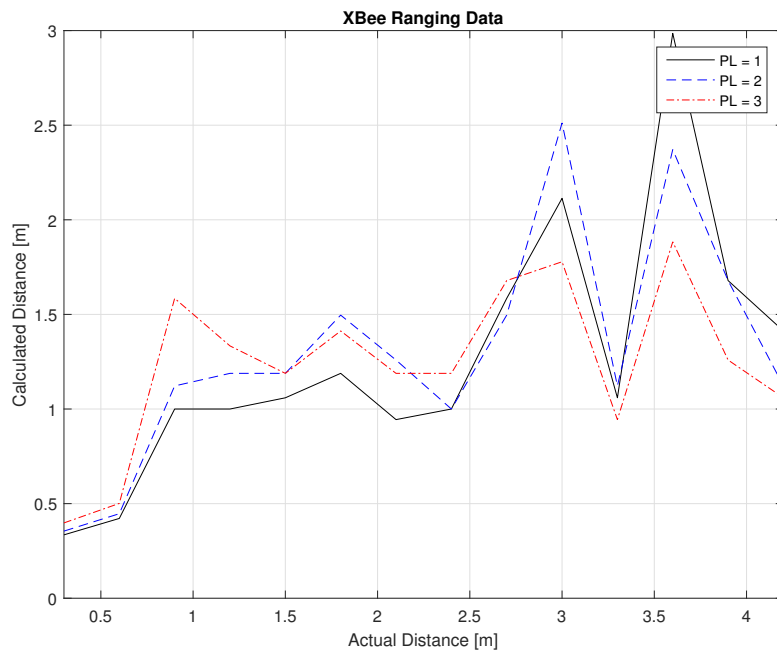


Figure 2: XBee ranging data plotted at calculated distance vs. actual distance



## 22-27 November 2016

### 1 Work completed for ECE 498

#### 1.1 Project Proposal Presentation Preparation

Since our project proposal presentation is the Tuesday after we return from Thanksgiving break, we worked it on over break. We did not focus on anything else since this was the priority.



## 28 November 2016

### **1 Weekly meeting to discuss progress with Dr. Miah**

We met with Dr. Miah today for last minute feedback on the presentation. He pointed out a couple things that we needed to change. He also suggested that we have a video of the simulation to show after the presentation as well.



## 29 November 2016

### 1 Work completed for ECE 498

#### 1.1 Project Proposal Presentation Preparation

We gave our project proposal presentation today. Overall it went well but a lot was brought up that we will need to address in the Spring. The biggest of these was fixing the problem we are having with multipath. This will be a complicated issue to fix but there are numerous papers that work at fixing it and applying it to distance measurement.



## 1 December 2016

### 1 Weekly meeting to discuss progress with Dr. Miah

We met with Dr. Miah again today to go over what the structure of the ISIE 2017 paper will be. He also told us that we need to submit the written project proposal before next Tuesday since it can't be submitted after that.

### 2 Work completed for ECE 498

#### 2.1 Project Proposal Presentation Preparation

Our main focus now is to finish the project proposal before next Tuesday. Also, since finals start next week, the entries following this one may not be very long as we will be preparing for finals.





## **2-5 December 2016**

### **1 Work completed for ECE 498**

#### **1.1 Project Proposal Presentation Preparation**

Working on the project proposal.

### **2 Working on the conference paper**

We started working on the ISIE paper more, when we weren't working on the project proposal.



## 6 December 2016

### 1 Work completed for ECE 498

#### 1.1 Project Proposal Presentation Preparation

We have finished the project proposal and emailed it to Dr. Miah.



## 7 December 2016

### **1 Weekly meeting to discuss progress with Dr. Miah**

We met with Dr. Miah today to discuss the structure of the first conference paper. It is due by the 15th so we need to submit it by then. Even though it is finals week we need to finish this paper as soon as possible.



## 8-9 December 2016

### **1 Working on the conference paper**

We are working on the ISIE 2017 paper.





## 10 December 2016

### **1 Working on the conference paper**

We have finished the ISIE 2017 paper and it has been submitted. Since this was the last thing we needed to do for the semester, there will likely be no more entries until next semester.



## 11 December 2016

### **1 Work done in the lab for the day**

I emailed Mr. Mattus and asked about the status of our parts request. The parts from Jameco have arrived but the others have not arrived yet.



## 8 January 2017

Since we will have a lot to get done once we are back at Bradley, I decided I would try to write the programs for interfacing with the stepper motors now. I also decided to install all the ROS components we will need to control the robot.

I tried to install ROS on Ubuntu 14.04.3 but since we installed it on the Beaglebone's onboard memory, there was not enough space. After doing a little more research, I found that we could instead be using Ubuntu 16 since it still has the cape manager from 14.04.3, there is a ROS version made for 16 (kinetic), and the ROS package needed for the Pioneer is supported in the latest ROS version. I put all of the work we had done already and copied it to my computer for safekeeping so I could run Ubuntu 16 on my Beaglebone.

Before we were running off the onboard memory and there were space issues so this time I decided to just boot straight from the SD card since it is 16 GB. This means there will be no issues of running out of storage space.

After getting everything running I had the storage space issue again. Turns out I forgot that putting an image on the micro SD will only make a partition big enough for the OS and a little extra. I can increase the partition size using *fdisk* but I need another package as well and since there is no extra space, I can't do it. The easiest solution now is to just start from scratch with Ubuntu 16 again. This isn't a huge issue since I still have time and it really isn't that difficult, I just have to sit and watch things install.



## 9 January 2017

I am going to start looking into the GPIO programming on Beaglebone Black today, after I get everything installed correctly. I do not think it will be too difficult to write since I am able to find some good tutorials online. The only issue I am going to have is that I won't be able to test it until we are back at Bradley and have the rotating reflector built, or at least the motor setup. In the meantime, I can probably think up some experiments we can do and write a couple programs to do it. I know that all this will end up being done with ROS so I might just write the code for controlling the motor integrated with ROS to begin with. For easier data gathering, I might also try to make it so the program just controls when the motor moves and in what direction, as well as when the XBee does its part. Then we can interface through MATLAB to control it for various experiments. This means easier writing of experiment code (although I might have to do it all since I don't know if Kyle has MATLAB 2015b to use the Robotics Toolbox) but that is not the end of the world.

I think the plan at this point is to have the final system be written entirely in C++. Depending on how quick we can work at the beginning of the semester, it might be worthwhile to figure out how to do matrix multiplication in C++ and have it be scalable to any size matrices. It is easy in MATLAB since that's what it is designed to do but I am unsure about C++. In addition to that, the amount of code to convert for just the algorithm is a good amount and some of the functions are not simple. If I find time, I might start writing a library for some of the functionality like the matrix multiplication, the predict step, the augment step, the update step, data association, calculating steering angle, etc. Whatever I can get done before going back (even if its not complete) will be useful and allow us to work a little faster down the road.

It took me awhile to get things installed so I am going to include how I did everything below. I am using Ubuntu 16.04 downloaded from [here](#). The instructions start right after booting from the SD card.

### 1 Setting up Ubuntu 16.04 LTS, ROS, ARIA, and ROSARIA on the Beaglebone Black

```
sudo apt-get update
sudo apt-get install parted
df -h /
```

Take note of the name of the boot partition. For me it was called `/dev/mmcblk0`. Use whatever yours is in the next step.

```
sudo fdisk /dev/mmcblk0
p
```

Using `p` will print all the partitions on the boot device. Take note of the sector that the actual boot partition starts at because it will be needed later.

```
d
n
```

Use the defaults until it asks for the starting sector, input the SAME value as before so the new partition will start exactly where the old one did. Accept the defaults until it says a new partition has been created. Use `p` to view the list again. If there is not an asterisk under the boot column use the following command:

```
a
```

If there was already an asterisk you do not need to do anything. The instructions continue below.

```
w
sudo partprobe
sudo resize2fs /dev/mmcblk0p1
sudo apt-get upgrade
cd
mkdir seniorProject
cd seniorProject
git clone https://www.github.com/attie/libxbee3
cd
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /e
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9
sudo apt-get update
sudo apt-get install ros-kinetic-ros-base
sudo rosdep init
rosdep update
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
sudo apt-get install python-rosinstall
```

After completing up to this point, ROS is installed on the Beaglebone Black. However, Aria and the ROS package for the Pioneer must still be installed. I am currently working on getting Aria installed since it is proving to be more difficult (there is no premade package to easily download and install) and I am compiling the library from source on the Beaglebone Black. I will continue with the instructions tomorrow most likely.



## 10 January 2017

### 1 Setting up Ubuntu 16.04 LTS, ROS, ARIA, and ROSARIA on the Beaglebone Black

I downloaded the premade package in a virtual machine to see what is actually installed and a lot of it is just example code. The most important thing is a `.so` file. I think there are symbolic links in that entire folder so I might have to compile ARIA in `/usr/local/Aria` so that the links work correctly. Reading a little further ahead, it is going to ask for an IP address and since I am not in the lab with one of the USB WiFi adapters and the router, I can't really go past downloading the ROS package for the Pioneer. The instructions from yesterday are continued below.

```
. /opt/ros/kinetic/setup.bash
cd seniorProject
mkdir -p /catkin_ws/src
cd /catkin_ws/src
catkin_init_workspace
cd ..
catkin_make
. /devel/setup.bash
cd /src
git clone https://github.com/amor-ros-pkg/rosaria.git
sudo apt-get install ros-kinetic-tf
```

Before making the ROS package for the Pioneer (we just downloaded it), ARIA must be installed. Since there is no precompiled package available it must be compiled from the source and installed. The instructions to do this are detailed below with the assumption that you are already in `/seniorProject/catkin_ws` following the previous instructions.

```
cd ../
wget http://robots.mobilerobots.com/ARIA/download/current/ARIA-src-2.9.1.tar.gz
tar -xvzf ARIA-src-2.9.1.tar.gz
cd Aria-src-2.9.1/src
make
cd ../ArNetworking
make
cd ../
sudo make install
```

I am pretty sure that you can skip the two `make` commands and just do `sudo make install` but I did not do it that way. I have copied the instructions to what you are reading based on how I did it. After doing the last command ARIA will be installed but there is still a couple steps to add it to the library cache.

```
sudo nano /etc/ld.so.conf
```

In the file that was just opened, add the following line of text in a new line at the end of the file. This will tell the location of the library when programs go looking for it.

```
/usr/local/Aria/lib
```

After adding the line, save and exit from the file. Continuing with the instructions:

```
sudo ldconfig  
rosdep install rosaria
```

The second command installs the dependencies of rosaria. I tried continuing without that command and it said that it was unable to continue so I used it. I don't really know what it is doing exactly, only that it is necessary. It will probably ask if you'd like to replace a location with the files provided by the maintainer. When/If prompted, say no since it is trying to replace Aria (which we just built and installed anyway) and our version is more up to date (by 2 years). There are only two more commands left to finish building the rosaria package:

```
cd ../catkin_ws  
catkin_make
```

If there is an error with *catkin\_make* about not being able to find a directory or something similar, you can try using *catkin\_make --force-cmake*. This will force catkin to re-run cmake. You will probably only have to use that command if you tried to build rosaria before installing the ARIA library (which shouldn't happen if you are following these instructions).

At this point, rosaria is installed but there will still be some setup once we are in the lab with the actual robot. We will need to do the following:

- Install drivers for the USB WiFi adapter
- Connect the Beaglebone Black to the router in the lab, over wireless
- Probably some other debugging with the serial connection

Even without doing the above, I was still able to run roscore and start the rosaria node. The rosaria node of course did not run but it did show that it was trying to and failed when a connection to the robot could not be found, so I am hoping that is a sign that I have done everything correctly. I think I will try and copy the image off the SD card so it can be put directly onto another card (as a backup). Right now the whole image is going to be rather large since it takes up all of my 16 GB SD card. I should be able to compress it though, I think I can at least. For now, this image backup will just be used for us in the lab in case we mess something up. Once we are at a point where everything works, I will

make another image with the size only big enough for what's included in the image plus a little bit extra. That way if someone wants to implement it again they only need to put the image on an SD card and boot from it.

## 2 Work done in the lab for the day

### 2.1 Simulating the algorithm using V-REP

Dr. Miah ask that I go through the simulation code to see why the figure in our ISIE paper had one beacon estimate that was not very close to its actual position. After going through it (and comparing to the original code from before my modifications), I can find no differences. I don't think its an error in the code. More likely, it is just because that beacon is the last one to be added to the map and it is done at a long distance so there is a lot more noise. I think those two things combined are causing the issue. Obviously it would be good to figure out a way to fix it but if I can't I don't think it matters since the paper is already submitted and I don't think there will be more V-REP simulations.

## 3 New uploads for the day to Google Drive

- *Implementation/code/diskImages/BBB-16.04-ros-aria-rosaria.img.7z*

### 3.1 Designing Experiments to Conduct in the Lab

I think it will be a good idea to design some experiments to do once we get back in the lab. I will probably also write a simple program on the Beaglebone that uses GPIO so we can control the rotating reflector assembly (once it is constructed). I don't think I will do it today, but definitely sometime before we go back I will start on it. I haven't decided if I want to integrate it with ROS yet. It would make the data gathering much easier since we won't have to figure out how to copy stuff from the Beaglebone but we would have to do the ROS setup before we can start the experiments. If I do anything today it will just be some jotting down on paper of pseudocode that I might write in C++ (with or without ROS). If I do this I will probably copy the pseudocode into an algorithm in this lab notebook.

The one in Algorithm 1 is one that I came up with real quick as a place to start:

Another thing we will need to consider is how to alleviate the effects of multipath fading. I have done a little research and there are a couple things we might be able to try. First, there are a couple of papers that I have uploaded in the Google Drive in the *refPapers/multipath* folder. They detail some methods used to address multipath but I do not know if they can all be applied to our project (some assume known distances between transponders which isn't always known for our algorithm). If anything, those

---

**Algorithm 1:** Lab experiment for use without the rotating reflector.

---

**Input:** Number of iterations  $k_{max}$ , time delay between measurements  $t_d$ **Output:** XBee range data over time for multiple beacons

```
1 begin
2   • Start program and initialize variables
3   • Check to see if UART2 cape is initialized
4   if UART2 is initialized then
5     | continue
6   else
7     | Initialize UART2
8   • Send initial node discover
9   • Store results for processing (data association?)
10  • Pause for  $t_d$  seconds
11  for  $k = 1, k < k_{max}, k ++$  do
12    | • Node discover
13    | • Store results (data association?)
14    | • Pause for  $t_d$  seconds
15  • Process the collected data
```

---

could be used as maybe a calibration step before the algorithm starts. The second method would be to apply a Kalman filter to the data after it is collected. This would do some smoothing on the measurements so they are better estimates of distance. We would have to do multiple observations over a period of time (and at each angle for the reflector) so this method might take a lot more time. It would also be more computationally complex since the Kalman filter would need to be calculated for every beacon at every reflector angle.

Another method would be RANSAC<sup>1</sup>. Again this would take time since it depends on sampling over a period of time but it is a less complex calculation to do so it could be done a little faster. It comes to mind that if we are feeling particularly brave, we could use multithreading to do data processing as a new observation is being done. This wouldn't be as necessary for Kalman filtering since it is recursive and we could just process the next set of data between observations (maybe multithread individual calculations?).

I will look into these methods a little before going back and maybe write a little more here about them. There is only really a week left so my first priority will be writing a couple programs for experiments then the data processing. Most of the data processing

<sup>1</sup>[https://en.wikipedia.org/wiki/Random\\_sample\\_consensus](https://en.wikipedia.org/wiki/Random_sample_consensus)

algorithms aren't that complex to implement or I am able to find code online to start from.



## 11 January 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

I looked at a paper that addressed multipath in a range-only SLAM algorithm<sup>1</sup>. In this paper, multipath was modeled as a 50% to 100% overestimate but only happening 25% of the time. I don't think this is a bad way to model it since multipath doesn't always occur the same way. I wrote a MATLAB script that implements this idea. So far the results look somewhat like what we experienced in the lab. In my simulation I used the same probability for more noise being added. I decided to just add noise to the distance and not the RSSI measurement. The MATLAB figure produced can be seen in Fig. 1.1.

I think I could use this simulation (probably with some modifications) to try and create a random multipath effect so some data processing can be done on it. I think increasing the chance of noise as distance increases would be a good idea since that is how it seems to happen in the lab environment.

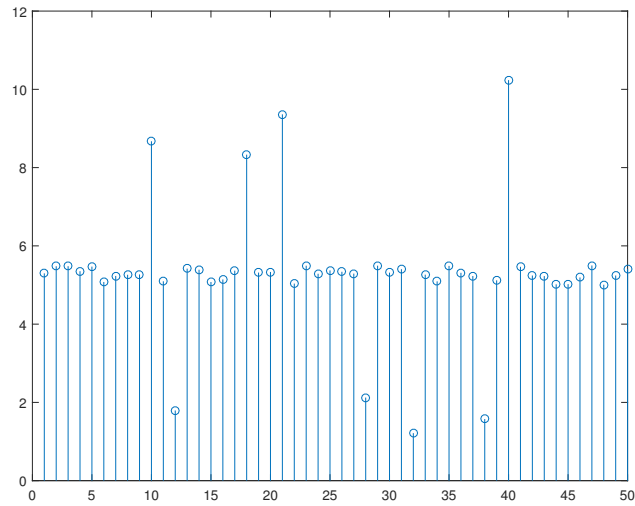
With this consideration added to the MATLAB script, the plot shown in Fig. 1.1 looks pretty similar to some of our lab data, shown in Fig. 2. I think this will be a decent starting point to figure out what methods might work to improve the distance estimates made by our system. Since most of the methods are linear, they should be applied once we convert the RSSI to distance. We will just need to make sure we calibrate accurately at a distance of 1 m so everything will be consistent.

Here is the list of methods I want to try using this MATLAB script:

- RANSAC
- A Kalman filter
- Moving average filter
- A basic line fitting

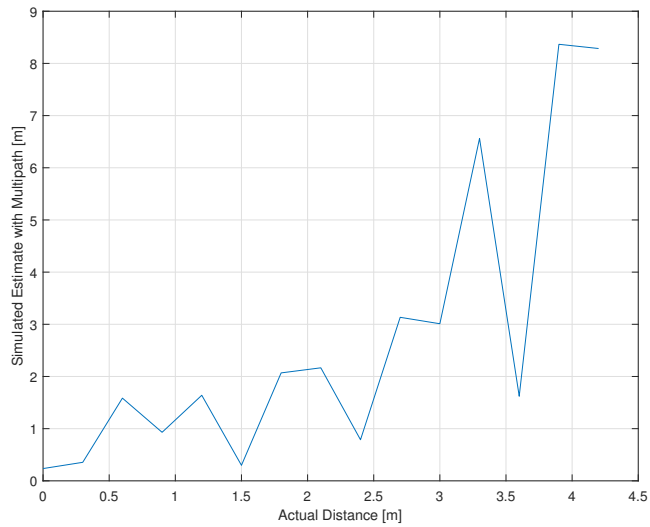
I will probably add to this list as I find more methods to test. I have already found some MATLAB code that implements RANSAC but it is confusing me so it is taking a little

<sup>1</sup><http://wbox0.cse.lehigh.edu/~spletzer/publications/tr008.pdf>



**Figure 1:** *Multipath simulation plot*

*Table X and Y axes*



**Figure 2:** *Another multipath plot*



longer to go though. RANSAC also requires parameters that must be carefully chosen for the situation, which might be more difficult in our environment.



## 13 January 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

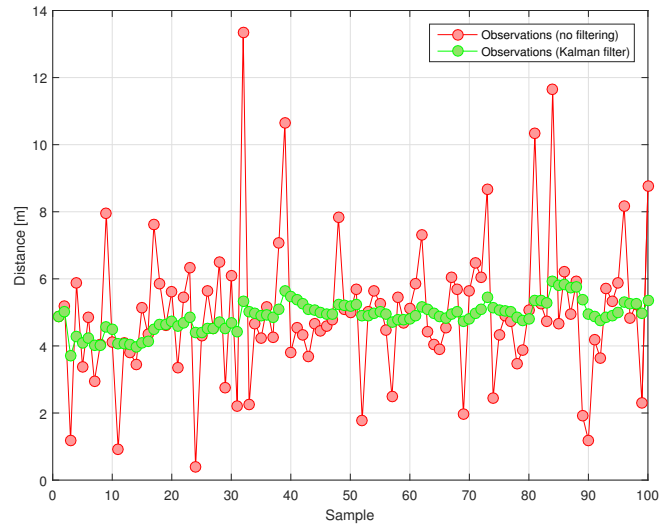
First of all, I took RANSAC off the list of things to try since it is really just a line fitting algorithm.

So far I have just tested a Kalman filter on my simulated data. Its only a one-dimensional filter so it is pretty fast, and I can only assume that it will be faster when written in C++. It will definitely work better the more samples there are. Right now I am just using 10 because I'm betting that is the bare minimum we will need to do (but I really don't know until I'm in the lab). The plot produced by MATLAB can be seen in Fig. 1.1.

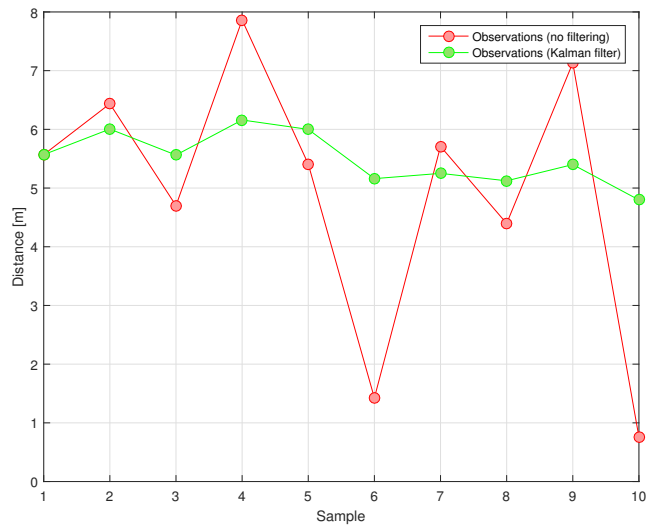
While this result does look somewhat promising, it is done over 100 samples which might not be possible to do in the lab (realistically). Another plot can be seen Fig. ??.

In both figures, the actual distance is 5 m. As you can see both are converging toward that value (more or less, there were a couple runs of the program that weren't as close to the actual value but were still converging). Based on these preliminary results, I think a Kalman filter paired with some other method may give us decent results. But of course, there is no way to know for sure until we are in the lab.

I tried a moving average filter and it does a decent job of smoothing the data but that's about it.



**Figure 1:** The effects of a Kalman filter on simulated range estimates over 100 samples



**Figure 2:** The effects of a Kalman filter on simulated range estimates over 10 samples

## 19 January 2017

### 1 Weekly meeting to discuss progress with Dr. Miah

We had our first meeting of the semester today. We went over the work we have done so far on the paper that must be submitted by February 11. This paper focuses on experimental results for estimating range and bearing using our proposed experimental setup. For this paper, there are a couple things we need to do:

- Make sure all figures are in vector graphics format (.eps)
- Make the figures described by Dr. Miah
  - Interfacing between the Beaglebone Black and the Pioneer 3-DX (the exact interface circuit)
  - Interfacing between the Beaglebone Black and the XBee module being used as the base beacon (including stepper motor, stepper motor driver, etc. for the rotating reflector), a picture of this will also be included
  - A figure showing the world coordinate system with a grid resembling the lab's floor. Pose of the robot will be placed in this figure for each test case. The locations of beacons will also be placed in this figure
- The results we want:
  - Range
  - Bearing

For the results, we will display them in a table for 10 robot poses, with a range and bearing estimate for 4 different beacons at each pose.

### 2 Setting up Ubuntu 16.04 LTS, ROS, ARIA, and ROSARIA on the Beaglebone Black

Since we will be using GPIO pins to interface with the stepper motor controller, I found a library on Github<sup>1</sup> that provides functions for all types of interfacing (since the XBee library is made specifically for interfacing with XBees and nothing else). I cloned this repository onto the Beaglebone Black.

<sup>1</sup><https://github.com/yigityuce/BlackLib>

### 3 Work done in the lab for the day

I uploaded all the datasheets for the parts on our order to the Google Drive so they are all in one place. For the motor driver, it says that screw terminals, pinouts, or directly soldering wires can be used. We did not order any screw terminals or pinouts but we may be able to use the leftover bits from the XBee interface boards. Mr. Mattus might also have some screw terminals we could use so we should ask him. I'm betting we use the pinouts since those will be available (hopefully) and easier to use.

I have drawn the schematics for the wiring on the Pioneer and on the beacons. I am not going to make them in ipe yet since we haven't actually made the circuits yet and we may need to add some capacitors or resistors in some places. The image in Fig. 3 is the connection diagram provided for the motor driver we are using.

In the user manual it also says that we can use the VCC from the motor driver for other purposes as well, but it should be limited to 85 mA. I don't think we should do this and instead keep the circuits separate.

#### 3.1 Designing Experiments to Conduct in the Lab

Since I did not write any programs for the experiments I am going to start one now. I am going to try and make it as modular as possible so we can test functionality of each piece (motor, XBee) then combine it. Off the top of my head, some of the functions we would want are:

- Initialize UART cape (with check to make sure it isn't already initialized)
  - Inputs: none
  - Output: error code (one for initialized, one for already initialized, one for other)
- Initialize GPIO pins needed for the motor control
  - Inputs: none
  - Outputs: error code (one for initialized, one for already initialized, one for other)
- Motor test (move the motor through the full range of motion, will need to make sure that the wires are long enough to reach the main XBee, probably a rotation in each direction)
  - Inputs: pointer to current angle counter
  - Outputs: error code (successful or not?)
- Rotate reflector to angle

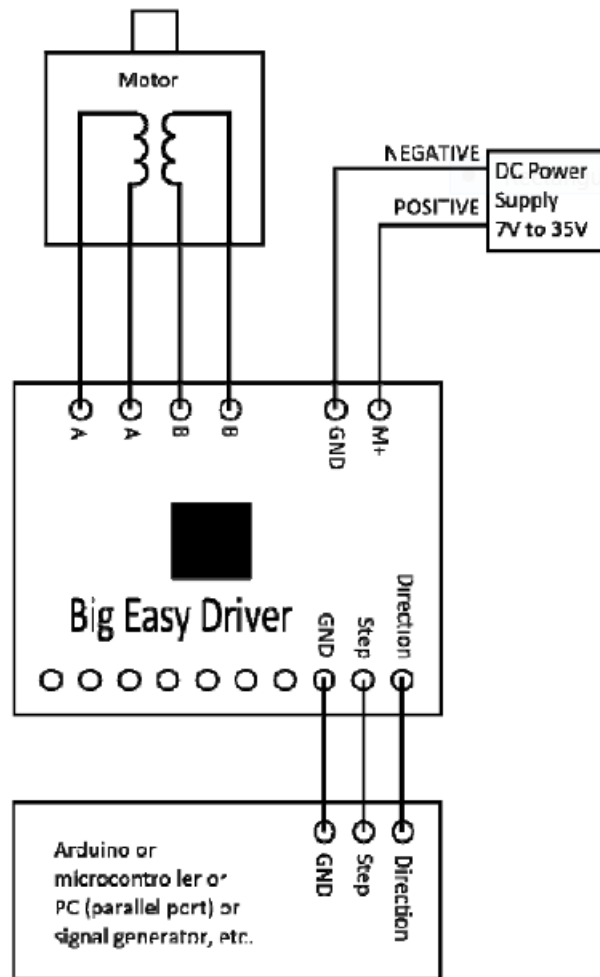


Figure 1: The connection diagram provided in the Big Easy Motor Driver's user manual

- Inputs: pointer to current angle counter, desired angle
  - Outputs: error code (successful or not?)
- Discover nodes and do data association
  - Inputs: pointer to current angle counter, pointer to data association table, pointer to matrix of results (so the new data can be added)
  - Outputs: number of results saved
- Increment the stepper motor number of given steps
  - Inputs: number of steps to take (or degrees to change?), direction to move, pointer to current angle counter
  - Outputs: error code (one for exact degrees moved, one for moving to the nearest step to desired degrees, one for other)
- Process data (range and bearing)
  - Inputs: pointer to the matrix of results (not converted to distance yet), variables from below needed to convert RSSI to meters, DA table
  - Outputs: pointer to matrix containing range estimate in meters and bearing estimate in degrees (relative to the robot's orientation, not the global reference frame) for all beacons (saved in the same order as the DA table)
- Calibrate (measure signal strength from  $nm$  away from a beacon, this function will probably only be used for the range and bearing experiments and not the final system)
  - Inputs: pointer to  $\eta$  value chosen, calibration distance  $n$
  - Outputs: reference power level from  $nm$
- Convert RSSI data to distance (does the entire matrix)
  - Inputs: pointer to matrix of results,  $\eta$  value,  $n$  value, reference power level at  $nm$
  - Outputs: pointer to matrix of results (probably in the same matrix that was given as the input)
- Data associate - convert the matlab code for data association to C++ (should have the same inputs and outputs)
- Print the results for debugging (estimated bearing to each beacon with the associated range value, in a nicely formatted table or something)
  - Inputs: pointer to matrix containing range and bearing estimates for each beacon (from process data function)



- Outputs: none since data will just be printed to the console window
- Shutdown GPIO
  - Inputs: none
  - Outputs: error code (successful or not?)
- Shutdown UART
  - Inputs: none
  - Outputs: error code (successful or not?)
- Combined shutdown function - combine the previous two functions for one easy function call (will process error codes from each nested shutdown function and print errors accordingly)
  - Inputs: none
  - Outputs: none
- Save range and bearing results to file
  - Inputs: pointer to matrix of results from process data function, filename to save to (full path)
  - Outputs: error code (successful or not?)

I started writing the initialization functions but decided that it would probably be best to write some of these functions as we are able to test them in the lab. I think to make building and running the experiments easier I will write them in the ROS workspace so I can make use of *cmake* to compile. This will also allow me to add a little ROS to the main experiment code so instead of trying to retrieve a text file off the Beaglebone Black, it can just be sent over ROS and be plotted immediately.



## 20 January 2017

### 1 Work done in the lab for the day

Today we are going to go get the parts we ordered from Mr. Mattus. As much as I would like to start putting things together today, I think we will probably just be testing things on a breadboard (make sure XBees work, putting together some beacon circuits, seeing if we can assemble the motor control circuit) and writing some code. For the motor control circuit, we will have to take some of the pinouts provided with the XBee adapters so we can actually connect to it. These will obviously need to be soldered on so they work. For that it may be wise to ask Mr. Mattus to solder it since there are active components that could be damaged by heat.

#### 1.1 Designing Experiments to Conduct in the Lab

Today Kyle worked on putting together a simple reflector for the XBees. Initially, only one reflector was made for the base XBee. Since we are only testing to see that the idea actually works, it was only constructed out of a thin piece of aluminum sheet and some cardboard. Placing this behind the main XBee (so it points toward a beacon), the RSSI was improved. We also tried rotating the reflector so it pointed in different directions. The RSSI decreases when it wasn't pointing at the beacon. Our testing distance was only about 1.5 m so there is still a chance that the results won't work as well at higher distances (where there could be more multipath) but so far so good. We also tried adding a sheet of aluminum foil behind the beacon. This also improved the RSSI but not nearly as much as the parabolic reflector behind the main XBee.

### 2 Setting up Ubuntu 16.04 LTS, ROS, ARIA, and ROSARIA on the Beaglebone Black

I set up the wireless adapter on the Beaglebone Black so we will be able to connect to it wirelessly. I also got the parts we ordered from Mr. Mattus and asked about the router that used to be in the lab, which I will have to go get sometime next week.



## 23 January 2017

### 1 Work done in the lab for the day

We worked in the lab today. Mostly just trying to get code to compile since over the weekend we each worked on writing some pseudo code for the functions that were previously listed.

#### 1.1 Designing Experiments to Conduct in the Lab

I (Jacob) was able to get the majority of the code compiling using `cmake` and `catkin` (since that is what is included with ROS and it is much easier to use than `makefiles`). There was some trouble with namespaces in C++ since *libxbee* is actually in C. For now I just split up the functions that are for XBees from the ones that are in C++. This corrected the issue but for the final implementation I want to fix it the correct way.

Kyle worked on testing the GPIO library I found (*BlackLib*). There were issues with compiling the library since the writer of it wrote the `makefiles` so it would cross-compile. Since Kyle was compiling directly on the Beaglebone there were lots of errors. Eventually he found the solution by simply compiling the example code in the command line (no `makefiles`). Testing this program with an LED revealed that nothing was happening. We tried multiple GPIO pins to make sure that wasn't the issue. I think it is the delay we have before the LED is toggled. Unfortunately, I had to go to class so we did not get the chance to check to see if this was the issue. I took home Kyle's bag of LEDs, wires, and resistors to see if I could find the solution at home.

The delay was not the issue. It was actually the permissions on the file. Since the GPIO files are in the root directory, you must be root in order to modify (or the program must be root). I discovered this by running the program with `sudo` and it worked. Obviously we won't be able to run functions as root so I used the following commands to change the ownership of the file to root.

```
sudo chown -v root:root /path/to/program
sudo chmod -v 4755 /path/to/program
```

This allows the program to be run as root without using `sudo`. These two commands will need to be run every time we recompile since recompiling essentially overwrites the old

file. We will probably have to do this for the entire `catkin_ws` to ensure that all files that need this permission will have it. We should look into a way that they will be executed automatically after compilation (probably a `.sh` file).

I also kept working on the code and it is closer to being complete. We still need to finish the functions for the XBees (modifying so data is saved instead of just printed), the data associated, and the main experiment file so it runs the actual experiment we want. I am sure there will be more troubles getting everything right with `cmake` because that was quite confusing. I have uploaded a zip file containing the ROS package for our implementation thus far to the Google Drive. Tomorrow I will take a picture of our experimental setup so it can be included here. For the experiment I will also probably include all the interfacing diagrams here as well since they will need to be made for the paper.

## **2 New uploads for the day to Google Drive**

- */implementation/ROS/slam\_exp.zip*

## 24 January 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

Jacob worked on finishing the code for the experiment. There are a lot of issues with compiling since we are using the tools provided with ROS for creating packages (catkin and cmake) and we do not have much experience with these programs.

Kyle worked on setting up the XBees so they can be used as beacons for our experiment. He also tested them with an improved version of his reflector and the results improved from the last test since both XBees are the same type with the whip antenna.

I ended up working on the code at home and I was able to get it to compile but I did not test it.





## 25 January 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

Kyle again brought an improved reflector following a guide published online for an antenna designed for use with WiFi routers. This reflector showed even more improvement since the tape that held the last one together has been replaced with glue and the reflector stays together.

I was not able to get the GPIO working on the Beaglebone Black even with the commands I had used earlier. Most of the time in the lab was spent trying to figure out how to fix it. In the end we could not figure it out but I still think it is something to do with the permissions on the files used to run the program.



## 26 January 2017

Before working in the lab, I uploaded a package<sup>1</sup> that uses both ROS and BlackLib to see if it would work on the BeagleBone Black (since the package is known to work for other people). There was an error given with this package about not having the right sudo abilities when using a *roslaunch* file. I searched Google and found the solution was to add a line to the *sudoers* file in Ubuntu. This line is shown below. To access the *sudoers* files you need to use *sudo visudo* since it is a protected file.

```
ALL ALL=(ALL) NOPASSWD:ALL
```

After adding this line, I was able to run the *.launch* file and the package worked correctly. I just ran the pulse test program to see that the GPIO was working and some LEDs would blink on and off.

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

I made a *.launch* file for our slam library and was able to use the GPIO through our library in the *incrementMotors* function. Since it works now, we need to figure out how to get the XBees working as well with our library.

Kyle did a range test for the XBees using our reflector and took down the data. The experimental setup and a plot of the data can be seen below in Figs. 3 and 1.

I think for the final implementation (not the experiment for the paper), we should write a C++ class for the motor similar to the one seen in *BenderControl* (the package I used to test if GPIO was actually working).

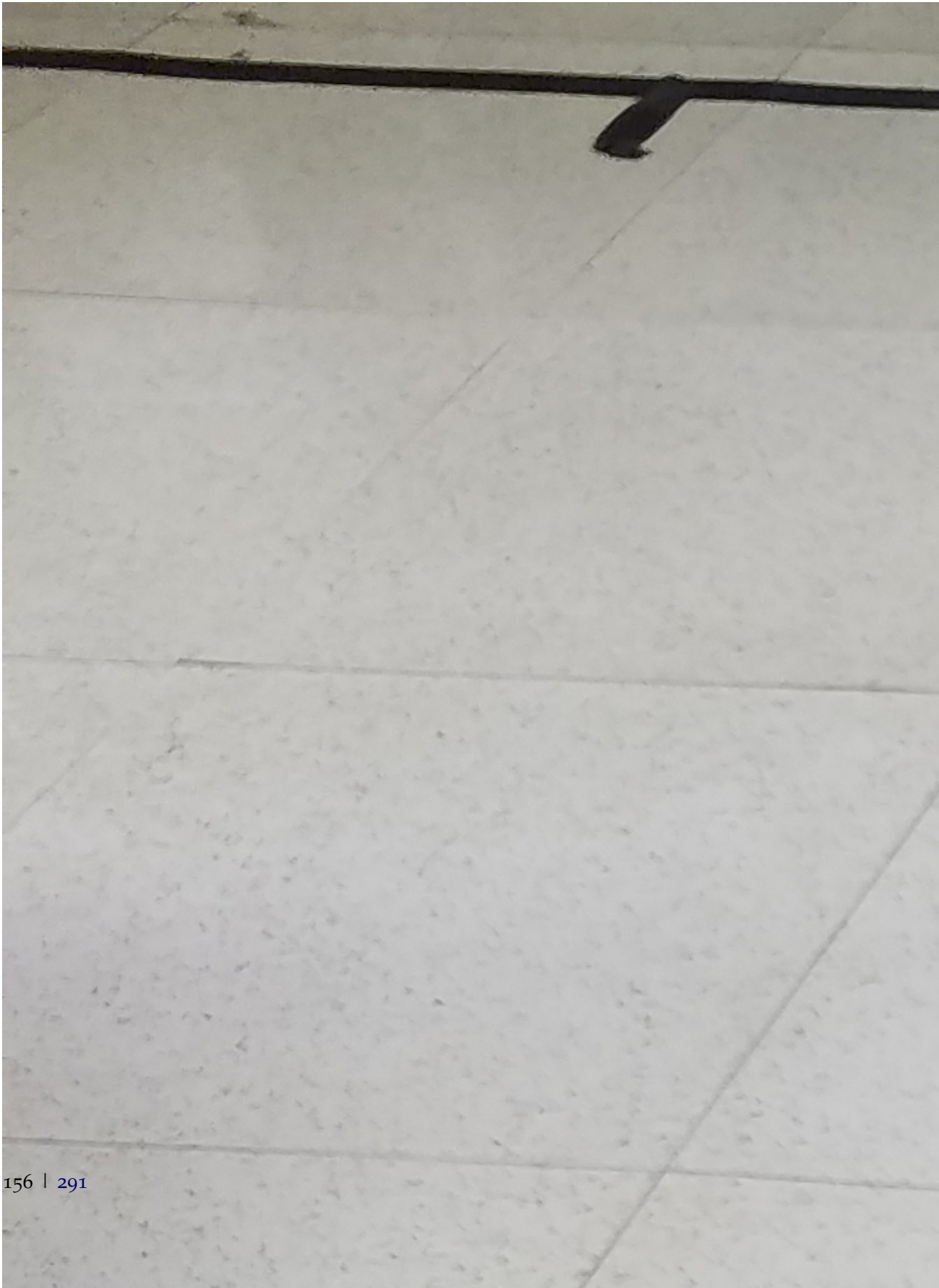
Since the GPIO is now working correctly, we decided to test the XBee functionality of our library. We connected an XBee to the BeagleBone Black (using UART2 again). This setup can be seen in Fig. 4.

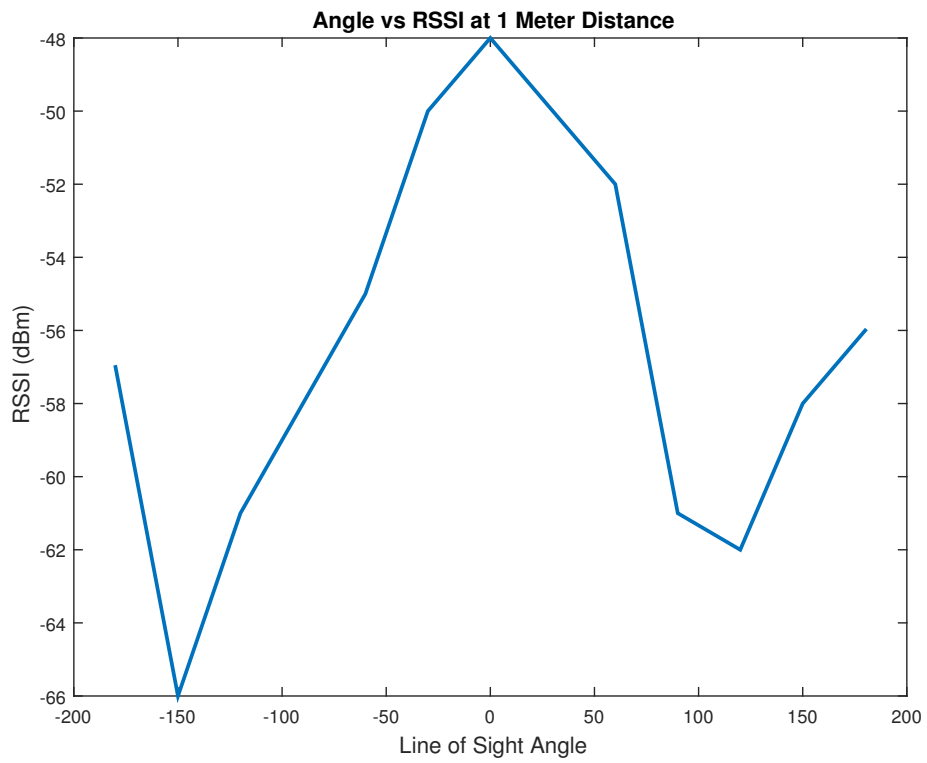
The plan for tomorrow is to talk to Mr. Schmidt about getting the motor controller soldered so we can test the motor. We should also ask about getting the reflector made

<sup>1</sup><https://github.com/OpenArmFramework/BenderControl>

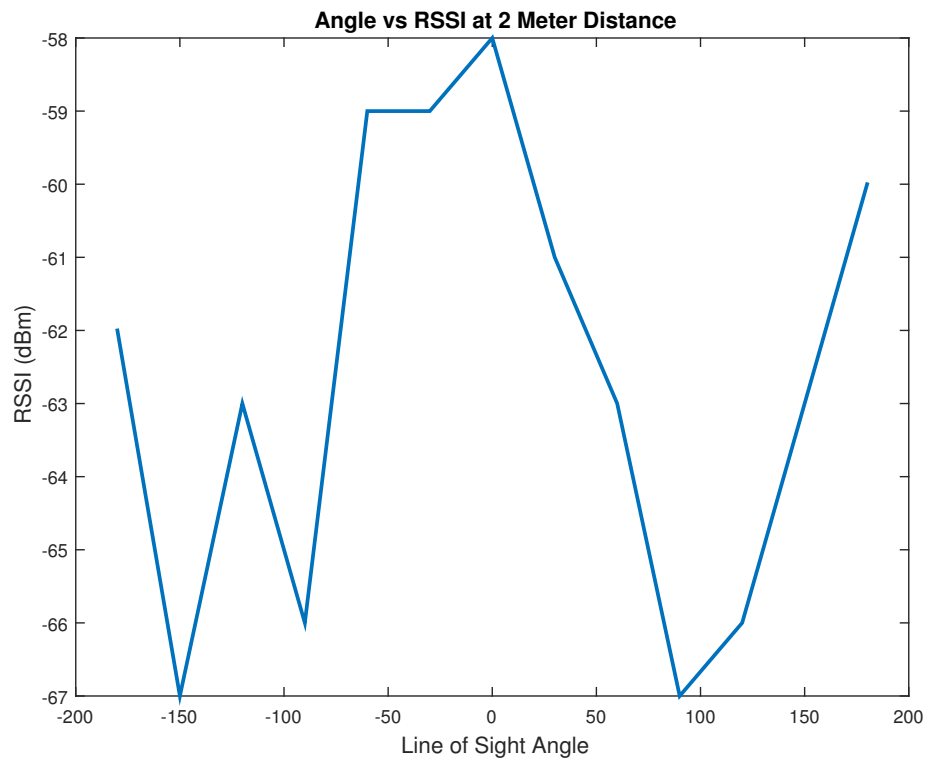
26 January 2017

---

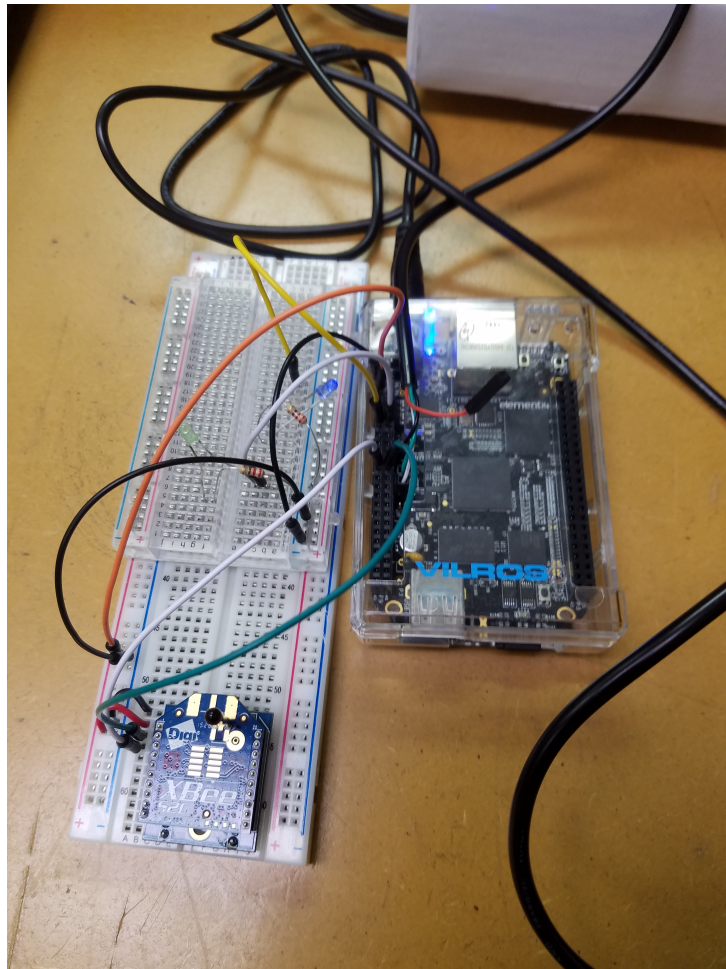




**Figure 2:** A plot of the data obtained from the reflector test at a distance of one meter



**Figure 3:** A plot of the data obtained from the reflector test at a distance of two meters



**Figure 4:** *The connections to the Beaglebone's GPIO for both an XBee and the motor controller (represented by LEDs since motor controller is not soldered yet)*

26 January 2017

---

from something a little more robust, like wood or something. We will also find a way to save the data from the libxbee callback function so instead of just printing out the results of the node detect it will save the data (either to a matrix or to a file if need be).



## 27 January 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

I went and got the motor controller soldered so we can test the actual motor with our code. I had Mr. Mattus do it since there are active components on the board and we don't know how to solder well enough to not break things.

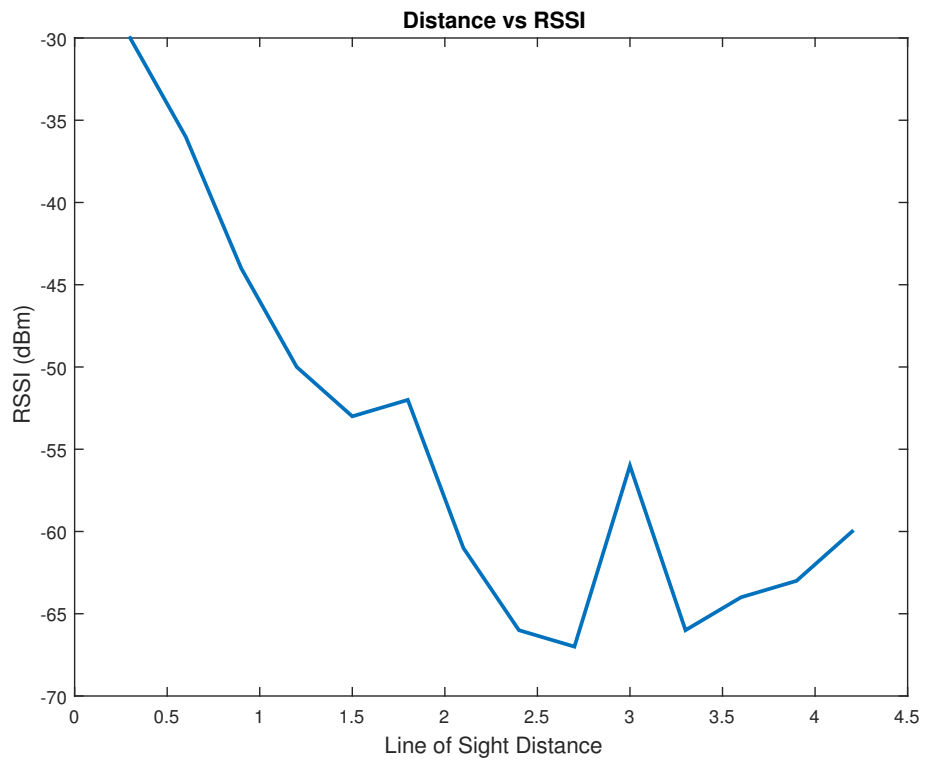
Kyle did another experiment with the XBees and the reflector. He took readings at a set distance with the reflector at a variety of angles. These results can be seen in Fig. 3 with the setup in Fig. 2

Once the motor controller was soldered, the test setup shown in Fig. ?? was assembled. Our test program only used the stepper motor since we already know that the XBee functionality is working and we were running short on time. In the end we were not able to get the motor running. But we only had time to run the program once and I am pretty sure its because I put the wrong value for the enable signal (should be active low but I put active high in the code). This was confirmed in the datasheet and I changed the value in the code before leaving the lab but did not test it.

### 2 Weekly meeting to discuss progress with Dr. Miah

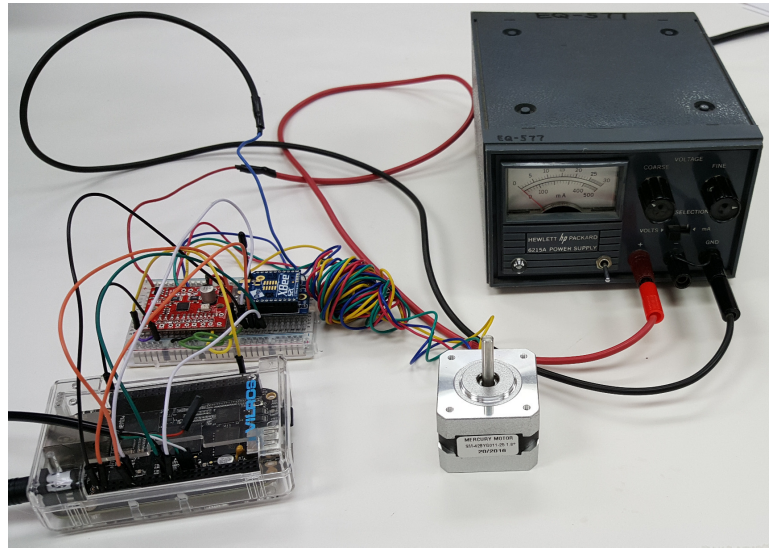
We had our weekly meeting today. Dr. Miah asked that we try to finish up the experiment by the end of next week so we can get the results in the paper and have time to do writing before we submit it. Here is what we still need to do:

- Solder remaining XBee interface boards
- Solder together beacons
- Build a better (more sturdy) reflector that can mount to the stepper motor's axle
- Power the stepper motor by the Pioneer 3-DX (12V accessory connection on the main board, its just a screw terminal. May want to put a 2 A fuse on it just in case)
- Figure out how to mount the stepper motor on the Pioneer so it is stable but also easy to remove when we are done testing (since other people use the Pioneer)



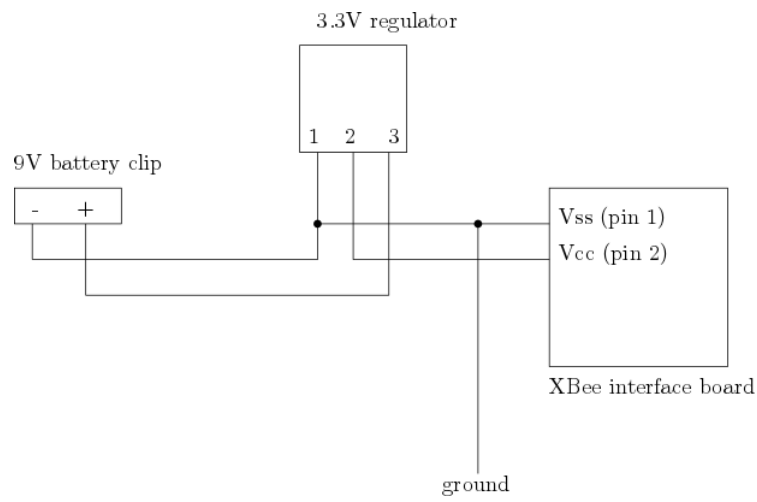
**Figure 1:** Experimental results showing the difference in RSSI at different reflector angles





**Figure 3:** Test setup of Beaglebone Black with connections to stepper motor (through the controller) and the XBee

I also need to email Dr. Miah this list and a schematic of the beacon circuit so it can be given to Mr. Schmidt and Mr. Mattus so they can work together on building them (to speed up the process a little). I should email Mr. Mattus and inform him of this plan as well since he has already been helping us with soldering. The circuit diagram for the beacon is shown in Fig. 4. I tested this circuit on a breadboard to ensure that it works.



**Figure 4:** *The circuit diagram provided for the construction of the XBee beacons*



## 30 January 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

Working in the lab again today. Our main goal is to test the motor controller today and do any debugging necessary for that so we know it is all ready for the experiment.

Using our program we are able to get the stepper motor to move but it is not anything close to correct. It is very jumpy and not even making full steps. Since the inputs are tied high with pull up resistors, we decided to do another little test by just connecting the inputs to pull them low then take the connection away to simulate a pulse input. Using this method we were able to see noticeable rotation in the stepper motor.

Since there shouldn't be any difference between our program and this, we went back through the datasheet for the controller. We also consulted a hookup guide that was given for the controller (a basic tutorial on how to use it to control a stepper motor with an Arduino Uno). There were a couple ambiguous parts of the manual that might actually affect our implementation. First, the controller can either be 3.3 V or 5 V. Obviously we want it to use 3.3 V since that is what the GPIO of the Beaglebone is capable of. The ambiguous part of this is how to change between these two modes. In the manual it talks about a solder jumper for controller the logic levels but the way it is worded makes it sound as if that jumper only takes affect if another jumper is being used. However, in the hookup guide, it is made clear that this jumper affects the logic level all the time. More reading told us that by default (jumper open), the logic level is 5 V. We corrected this by having Mr. Mattus solder the jumper closed.

We also kept reading the manual to make sure we had the potentiometer on the controller set to the correct position. There is an equation in the manual that tells what the current is limited to based on a changing reference voltage pin. We adjusted this value so it would be limited to 0.33 A (the spec of the motor).

Once these fixes were made, we tested our program again and the movement is definitely more consistent but still in very small increments (less than the quoted  $1.2^\circ/step$  from the motor's datasheet. This was a pretty simple fix, the controller implements microstepping

and those control pins are pulled high. Pulling them low removed the microstepping.

We were able to control the motor with decent accuracy so it should work well for our implementation.

Unfortunately, while testing my Beaglebone decided that it no longer wanted to work and died completely. There were no connections that would have caused this so it must have just been a defect in the board. Luckily all of our data was on the microSD card and we will be able to continue testing on another Beaglebone in the cabinet until a new one can be ordered.



## 31 January 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

For the soldering of the beacons, Mr. Mattus said he would show us how to solder so we could do it ourselves. First he told us to figure out if we need to put a heatsink on the voltage regulator. We would need to look at the datasheets in order to calculate how much heat will be dissipated (based on the voltage drop across the regulator and the current draw of the XBee).

We determined that no heat sink would be necessary for this circuit, since the heat being dissipated is only about 12 degrees Celsius more than ambient.

The other thing we needed to determine is if we need capacitors on the input and output of the regulator. We tested all of them and none were terribly noisy. Since the regulator would be close to the power source, we decided that we would only need a capacitor on the output (as suggested by the datasheet as well). We used a 10  $\mu\text{F}$  capacitor.

Mr. Mattus then showed us the best way to solder them together. By the end of the day we only had the one beacon done and it was the one he showed us how to make. We will be making as many as we can tomorrow while we are in the lab. An image of a completed beacon can be seen in Fig. 1.



# 1 February 2017

## 1 Work done in the lab for the day

### 1.1 Designing Experiments to Conduct in the Lab

All we did today was assemble/solder the beacons together. We were able to finish three so there are only two more to be made tomorrow. Our plan for the rest of the week is to complete all the beacons, mount the reflector to the motor somehow, and run a quick test of the whole system. This will let us use next week to do the experiment and get all the data for the paper.

We need to start working on the website for our project. We all got a quick lesson from Dr. Malinowski on how to do it using Microsoft Expressions. Kyle has the permissions to upload stuff to the actual server the page is on so if I want to update it I just have to send him what I do so it can be uploaded.



## 2 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

We are finishing up the last of the beacons today. It shouldn't take long so we should also be able to test the data association in our program, although we still need to figure out how to save the data in the first place.

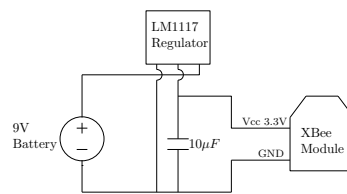
An updated beacon circuit diagram can be seen in Fig. 1. The only addition is the capacitor on the output of the regulator and the diagram is a bit better organized and has relevant details included.

Tomorrow we will work on attaching the reflector to the stepper motor so we can run some tests of the experiment. We probably won't work in the lab much tomorrow since Kyle cannot stay for the usual amount of time.

I have been thinking about what needs to be on the next order we place and have come up with the following list:

- Beaglebone Black rev C
- 8 GB microSD card
- Beaglebone prototyping cape
- 9 V batteries (rechargeable?)
- axle mount for stepper motor (for a better reflector)

I'm sure we'll think of more but we have time to figure that out later.



**Figure 1:** An updated circuit diagram for the beacons



## 3 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

We attempted to mount the reflector to the motor's axle today. Kyle had added a thicker piece of cardboard to the bottom in order to make it more stable. However, it was quite difficult to get the reflector to stay on and remain stable.

In addition to this, I changed some of the GPIO pins used by the program and for some reason this kept anything from happening with the motor. I had used the GPIO ports before so I am not really sure what was wrong. I changed them back to the originals and everything worked fine.

Since we were not able to mount the reflector, we made the decision to order something called a universal mounting hub. It fits over the axle of the motor and is held in place with a set screw. This will let us more easily attach things on top of the motor (and it won't flop around). Unfortunately we do not have a lot of time before the AIM paper is due so we made the decision to order the part on Amazon using Kyle's account since it will be able to get here by Monday. They were not expensive so its not that bad and trying to order them through Mr. Mattus could take longer to get here so we felt this was our only option.

### 2 Creating the website for our project

Since Dr. Malinowski showed us how to make a website on Wednesday, we need to start working on the one for our project. I looked through some templates today. We should probably ask Dr. Miah if there is anything special he wants us to do for the website.





## 4 February 2017

### 1 Writing the Conference Paper for AIM 2017

I made some figures for the AIM 2017 paper and included some of the pictures included in this lab notebook (picture of a completed beacon). I also added some more text to certain sections and corrected the wording of others. The biggest addition was the table that will hold all the results. There are still a lot of parts that need to be finished though.



## 6 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

The universal hub mounts arrived so we were able to attach the reflector to the stepper motor. The reflector is now very stable and moves with the stepper motor. We ran some tests and the accuracy is reasonable (good enough for our application). There is some error as the step size gets smaller and more steps are taken.

There was some issues with our node detection function. Since we are doing the function pretty fast, there were some timing issues. The function is a copy of the sample in *libxbee* so it turns on the XBee at the start and then shuts down at the end. The shutdown actually takes longer than the function runs for. For now we just added a delay after the shutdown function to make sure the XBee is shutdown before it is started again. Without this we only get measurements at every other angle.

We also noticed that the stepper motor gets pretty hot when we are using. It might be worth looking into a mount for the stepper motor and a heat sink to mount on the bottom. Or we could just use a fan to blow cold air over the Beaglebone Black and the motor. I think we can also use the sleep mode of the motor driver to stop supplying current to the motor (as an alternative). Best choice for the final implementation is obviously the heat sink.

Hopefully we can run the experiment tomorrow. We just need some wire to connect the motor to the power management board inside the Pioneer 3-DX.



## 7 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

We set up the entire system today and ran a quick version of what our experiment will be. There were some troubles initially with the XBees because someone had decided to use them without telling us and we were only able to communicate with one of the beacons. With this corrected back to the right settings we ran our pseudo-experiment.

It does work as we intended but it isn't very fast right now. This is partly due to the delays we have in the code and the speed we are moving the motor at. Dr. Miah wants it to do all the observations in a time around a minute. We are not sure if this is actually feasible but we will obviously try to make it as fast as possible regardless. About halfway through the experiment the reflector came loose from the mount so the data we got had very little meaning. We will rerun the experiment once we have more time and the reflector is secured better, probably Friday.

### 2 Writing the Conference Paper for AIM 2017

We have decided that we will not be submitting a paper to the AIM conference. There are a couple reasons:

- Neither of us can go to Germany
- Dr. Miah can't find anyone to present for us
- We probably won't be able to get it done in time

Instead, we will submit a shorter paper (4 pages instead of 6) to a conference in Windsor, Ontario. This way both of us will be able to go and it is within driving distance. We have until February 24 to finish the experiment and modify the paper we already have so it is the correct length for submission.



## 8 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

Today we just worked on connecting the Beaglebone Black to a WiFi network. We were going to do this by creating an ad-hoc network and then we would be able to connect to it from our laptops. We spent a lot of time trying to do this and it was very confusing since there were multiple different ways of doing it. We abandoned this idea and decided to use a program called *Virtual Router* to create an ad-hoc network on one of our laptops and then just follow a simpler guide on connecting a Beaglebone Black to a WiFi network. This method actually worked and we were able to SSH into the Beaglebone. We were also able to do this after a reboot meaning it doesn't have to be set up every time.

Since we are going to be writing more code and it will be getting more complicated, we are going to start using Git to manage versions of the code. The version control is much easier and it will allow both of us to make changes more easily. It will also allow us to setup the new versions much faster since we can clone the repository directly onto the Beaglebone Black. The repository can be found at [https://github.com/TheSmallHill/ekf\\_slam](https://github.com/TheSmallHill/ekf_slam).





## 9 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

We mostly just worked on the code today for the experiment to get it working better. We were unable to test anything on the Pioneer since it was being used for the robotics lab.

We are also trying to find a way to store the results of the experiment. We wanted to save things to a file from within the program but we cannot find a way to make it work with the callback function in the code.

I am going through the code seeing if there is another way to save the data in the program. The code is updated on GitHub.



## 10 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

As an alternative we tried using the *script* command in Linux to save the output of the console to a file. This worked fine in a small test but once we did a full scan the Beaglebone froze and the data was lost. This method did seem to work well enough (at least for the experiment we want to do for the paper). We think it would be a lot easier if we were able to power the stepper motor from the Pioneer to make moving things around easier. It would also help if we had a router so we could wirelessly connect to the Beaglebone. Unfortunately the virtual router software we tried is too inconsistent for us to use in the experiment (connectivity loss, slow transmissions).

### 2 Weekly meeting to discuss progress with Dr. Miah

Dr. Miah wants us to work harder on getting the experiment done so we don't have to worry about it. We need to revise the paper so it meets the page requirement and so it is written better. We also need to update the figures in the paper so they are correct and add descriptions for each one. The main focus right now is to finish the experiment and not waste time on other things. We have decided that we should take measurements from the entire 360° instead of just the forward facing 180° like we originally planned. This should eliminate the issue with a beacon being stuck behind the Pioneer and having abnormal measurements. Dr. Miah also explained some other experiments we can run by placing a beacon at a variety of distances and angles in relation to the Pioneer. This set of experiments will help illustrate the method of determining the angle of arrival.

In regards to the router, I emailed Mr. Mattus about borrowing one from him on Monday. When we go get it we should ask about getting some longer, more flexible wire to connect the main Xbee to the Beaglebone (our wires now are short and stiff so it is affecting the movement of the reflector). We should also ask about getting some wire so we can power the stepper motor off of the Pioneer instead of a power supply.



## 13 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

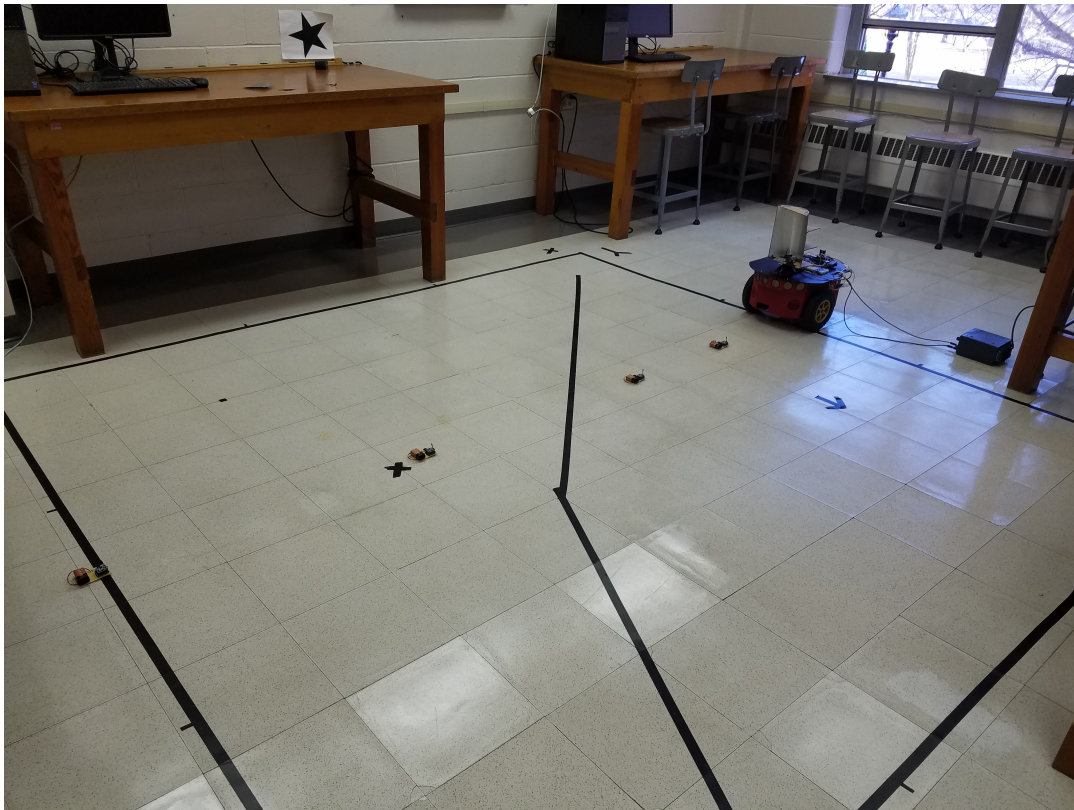
Kyle brought in a router so we could use SSH to connect to the Beaglebone Black instead of the serial-to-USB connector like we have been doing so far. This time connecting to the router was much quicker and we had it done in about 10 minutes. This is the command we used to make the connection profile in *nmcli* on the Beaglebone Black:

```
sudo nmcli device wifi con "NETGEAR81" password "silkycartoon776"
```

This command connects the Beaglebone Black to Kyle's router. Right now we are not giving it a static IP. We can just log into the router and see the connected devices we can just look for the one named *arm* and use that IP address to SSH into the Beaglebone Black.

We are running an experiment today. We have all four beacons in a line at a known angle. This experimental setup is shown in Fig. 1.1 We are then taking RSSI measurements for all 360°.

We have learned that putting XBees on the ground is a bad idea since they can get stepped on really easily. Kyle stepped on one. The beacon circuit is fine (just bent the pins back to the correct positions), but the voltage regulator short circuited and 9 V went to the XBee and killed it. Its unfortunate but we have other XBees that we can use for the experiment and we will just add 2 XBee S2Cs to our next order. One to replace the one that was broken today and one as a backup just in case.



**Figure 1:** *Experimental data from the experiment*

## 14 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

We got the router from Mr. Mattus today. We will probably set up the Beaglebone to use it tomorrow since we already have Kyle's router working for now. The plan today is to run as much of the experiment as we can. There won't really be much to write since running the experiments is just gonna be us waiting for it to finish.

We were able to do the experiment at 5 of the 9 positions today which is pretty good. I used an Ubuntu virtual machine so I could download files from the Beaglebone over the network instead of using a USB. The form of the command to do this is:

```
sudo scp ubuntu@192.168.1.3:/path/to/source path/to/destination
```

I did this for each of the tests and uploaded the text files to Google Drive. I also started processing some of the data and for position 1 it looks like there is something interfering with the measurements. That's okay since we are just showing that the system is working, but we will probably need to figure out a way to reduce it for the journal paper.

### 2 uploads

These 5 files were uploaded into *implementation/experiments/Experiment1/*. There is also a spreadsheet where we will be putting the data so we can plot it.

- *angle\_p1.txt*
- *angle\_p2.txt*
- *angle\_p3.txt*
- *angle\_p4.txt*
- *angle\_p5.txt*





## 15 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

We set up the router that Mr. Mattus provided. I used a different command this time so it would have a static IP address. This is the command:

```
sudo nmcli con add con-name wireless ifname wlx74da385be415 type wifi ssid ECE-Robots-1 \ip4
```

We are now able to SSH into the Beaglebone Black again. We are going to try to run some more of the experiments again today and then go through all the data we already have while they are running.

I took a backup of our microSD card. The compressed version will be uploaded to the Google Drive.



## 17 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

Kyle finished the last of the experiment today. The data is a little bit more noisy than the previous data we have collected (most positions had one beacon that we could not accurately estimate the angle). We think that the reason some beacons can not be found as accurately is due to some multipath introduced by the metal objects in the lab where we are doing our experiment. For this conference paper it is not as big of a deal but we will need to figure out how to handle this for our final implementation.

We are still making plots of the data but once they are complete I will add the rest here. Fig. 1 shows the positions of the XBee beacons in our experiment and the nine robot poses where measurements were taken from.

### 2 Making a better reflector for the system

Since we have finished the experiment for the upcoming conference paper, I have decided to try and make a better reflector. I am going home so I can use the tools at my house to make a sturdier system to hold the Beaglebone Black, the motor controller, and the reflector assembly. In addition to being more accurate (hopefully), it will be easier to setup the system since we will just be able to pick up the whole thing instead of taping the individual components to the top of the Pioneer. I am still drawing out the plans but once I finish them I will upload them here.

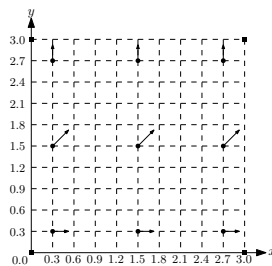


Figure 1: XBee positions and robot test poses for Experiment 1



## 18 February 2017

### 1 Writing the Conference Paper for AIM 2017

We are working on finishing the second conference paper so we have a rough draft complete for Tuesday. Today Kyle added the table of data we gathered from the experiment we did for the paper. This data can be seen in Table 1.

This table was included in the paper. We are also going to add an analysis of the error at the first test pose and a plot of the RMSE for all positions and beacons.

The goal is to get the paper to be 4 pages maximum. To do this we will need to work on these sections:

- Introduction
- Experimental Results
- Conclusion
- Prior Work
- Hardware Overview

Some of these sections can probably just be combined but others need to be shortened to make room for the other table and plot.

**Table 1:** Range and bearing measurements taken in Experiment 1

Robot Pose ( $x, y, \theta$ )	XBee 1		XBee 2		XBee 3		XBee 4	
	$r_1$	$\beta_1$	$r_2$	$\beta_2$	$r_3$	$\beta_3$	$r_4$	$\beta_4$
(0.3, 0.3, 0)	0.56	$-153^\circ$	2	$-36^\circ$	2.8	$81^\circ$	2.23	$31.5^\circ$
(1.5, 0.3, 0)	1	$-180^\circ$	3.55	$-9^\circ$	2	$112.5^\circ$	2.24	$54^\circ$
(2.7, 0.3, 0)	4	$175.5^\circ$	0.56	$-67.5^\circ$	2.82	$121^\circ$	7.94	$90^\circ$
(0.3, 1.5, $\frac{\pi}{4}$ )	1.41	$-157^\circ$	3.55	$-99^\circ$	1.26	$45^\circ$	2.82	$-31.5^\circ$
(1.5, 1.5, $\frac{\pi}{4}$ )	2	$171^\circ$	2.51	$-103^\circ$	1.78	$81^\circ$	2.82	$-18^\circ$
(2.7, 1.5, $\frac{\pi}{4}$ )	2.52	$157.5^\circ$	1	$-135^\circ$	2.82	$99^\circ$	1.26	$31.5^\circ$
(0.3, 2.7, $\frac{\pi}{2}$ )	2.24	$162^\circ$	7.07	$-148^\circ$	0.25	$40.5^\circ$	1.58	$-85.5^\circ$
(1.5, 2.7, $\frac{\pi}{2}$ )	4	$144^\circ$	1.77	$-180^\circ$	2.51	$72^\circ$	1.12	$-85.5^\circ$
(2.7, 2.7, $\frac{\pi}{2}$ )	4.66	$126^\circ$	2.52	$175.5^\circ$	2	$67.5^\circ$	0.35	$-54^\circ$

## 19 February 2017

### 1 Writing the Conference Paper for AIM 2017

I added the table with the analysis and put in a missing figure that we can replace with the finished MATLAB plot of all the RMSEs. Kyle added the data to the analysis table. I also started rewriting the introduction so it is a bit more relevant to the paper and so it is shorter overall.

I did a quick read through the paper and it is definitely closer to being a real rough draft. I changed "active beacon network" to just "beacon network" so it is consistent with the figures (this includes "active beacon" to "beacon"). I also changed "RF sensor" to "beacon" for consistency with the figures.





## 20-23 February 2017

### 1 Writing the Conference Paper for AIM 2017

All we have been doing for the past three days is making revisions to the conference paper. We have met with Dr. Miah each day to discuss what needs to be changed.



## 24 February 2017

### 1 Creating the website for our project

Since we have our meeting today with Dr. Miah about the next paper (the journal paper), we are not going to work in the lab today. Instead, we are working on the website that we need to maintain for our project. It isn't really a priority right now but the earlier we get it done, the easier it will be to maintain and the less we will have to do later.

### 2 Weekly meeting to discuss progress with Dr. Miah

Today is the kickoff meeting for the TII journal paper. The submission date is March 30 so we are really going to need to work hard to get everything done by then. Here is the current list of what needs to be done:

- Finish writing algorithm in C++
- Finish the new reflector
- Test and debug as necessary
- Place another order with Mr. Mattus
- Run experiments for the completed project
- Write the paper

Dr. Miah would like us to be done with everything and have about a week to finish the paper before the submission. We do not think this is unrealistic but it will still be a lot of work for two people. I (Jacob) am taking charge for the journal paper since I have submitted one before but Dr. Miah said he will help write some of it as well.

At Dr. Miah's suggestion, we are taking the weekend off from working on the project so we will be well rested next week. Probably a good idea since we are probably going to be in the lab a lot in the coming weeks.



## 27 February 2017

### 1 Work done in the lab for the day

#### 1.1 Designing Experiments to Conduct in the Lab

I have just realized that I forgot to include the plots of all the data we obtained from the experiment done for the second conference paper. There are nine plots (one for each test pose). The first five of them are shown below and I will include the other four soon. I have also included figure showing the locations of each test pose (the one that was put in the conference paper).

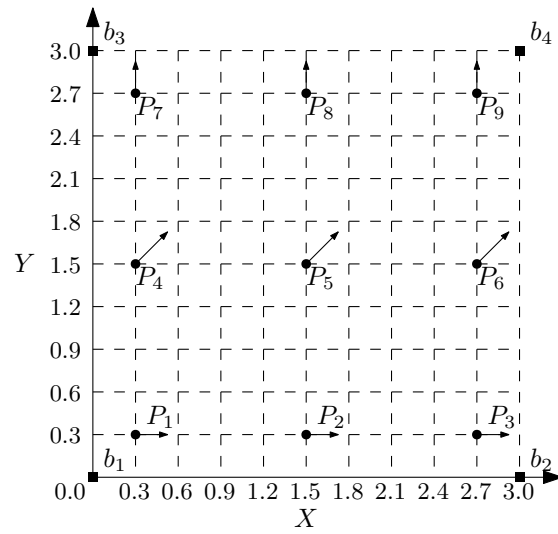
### 2 Work done in the lab for the day

#### 2.1 Converting the code to C++ and adding SLAM algorithm

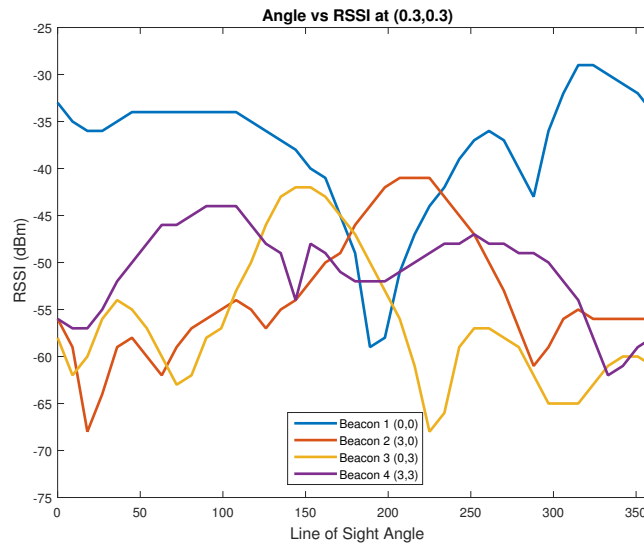
We made a lot of good progress today. Kyle tested to see just how fast the XBees would respond and he was able to get it to be almost instantaneous in XCTU. Since XCTU is kinda bloated to begin with, we figure it should be a lot faster when done in C++ with libxbee. I downloaded the C++ version of the code we have. It uses classes and the C++ interface for libxbee instead of the C code.

Before testing the code we mounted the rotating reflector that I had made out of plexiglass to make sure it actually fits on the universal mounting hub. It fits well enough for us to use (one hole didn't line up). We did however find one issue with the piece that the motor is mounted to. With the motor sitting how it was planned, you cannot get your hand between the two pieces in order to tighten the set screws onto the axle. This will keep us from using that piece at least. Kyle had the idea to use something like the pieces in an erector set to make a motor mount so it is easier to get your hand in there. For now it doesn't matter but later on we will look into it.

We tested the motor object first. Only issue we had was that we reversed the two connections for the driver and instead of sending pulses to the step we had them going to direction. Nothing bad happened and it was a quick fix. We did a quick test spin and everything works like it used to. The major additions include a better way of tracking



**Figure 1:** Locations of the nine test poses where data was gathered



**Figure 2:** Experimental data at test pose 1

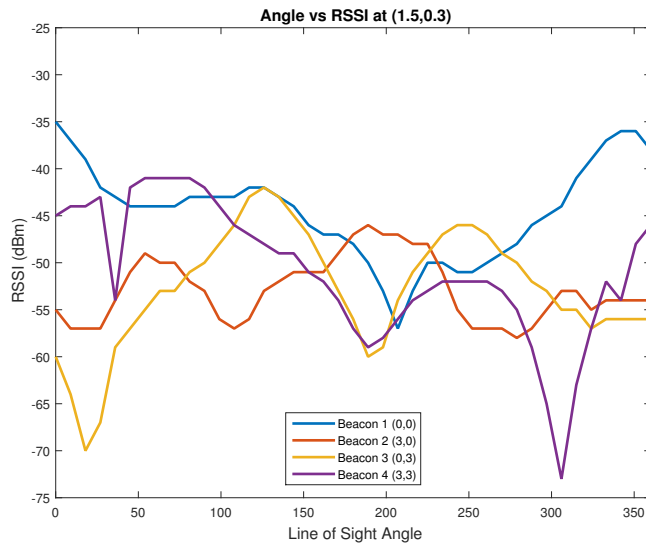


Figure 3: Experimental data at test pose 2

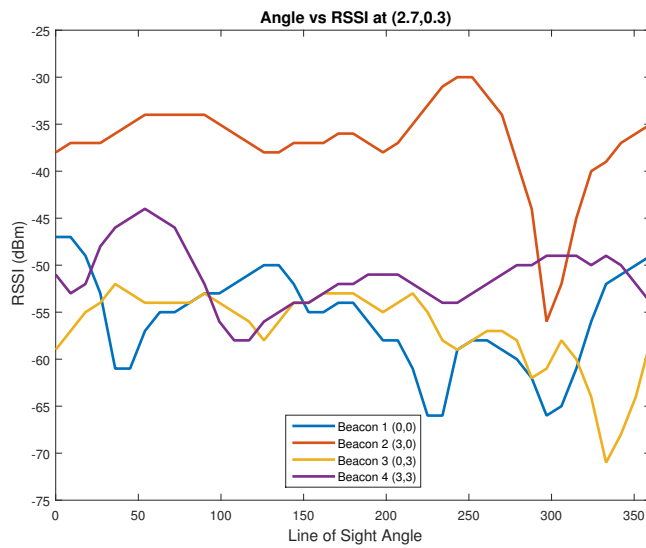


Figure 4: Experimental data at test pose 3

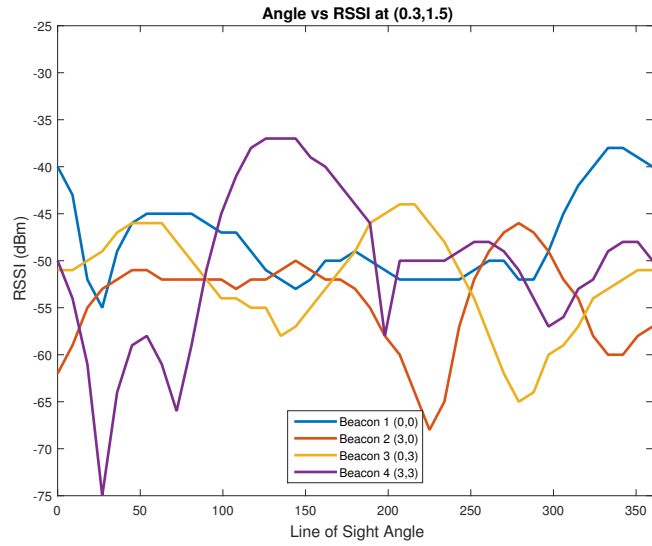


Figure 5: Experimental data at test pose 4

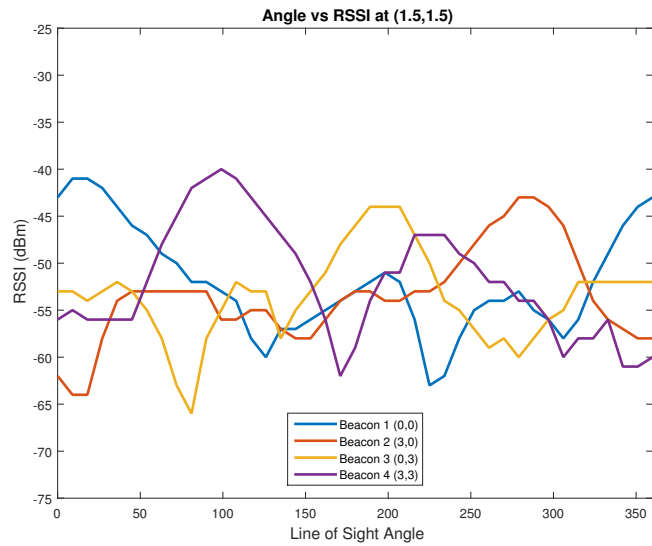


Figure 6: Experimental data at test pose 5



the angle and the inclusion of the microstepping pins. There were more issues with the observer object.

I had added code to automatically start UART but that did not work. This isn't a big deal now since we can still manually start UART. There was also an issue with accessing the XBee but that was just because I forgot a forward slash. At this point we were still not able to get data to print. Turns out there is essentially an infinite while loop because it is waiting for a special packet to be received (an empty packet). In our three tests this packet never happened and it was impossible to stop the program since the callbacks happen in a different thread. We had to reboot the BeagleBone Black to get it to stop. In order to fix this, we found the while loop and put a counter so it would break free after a certain number of iterations. This will probably get changed to a for loop and return statement that is somehow triggered after the counting is done.

Now we are able to print out the data. We left the lab today with the only thing left to do is parse the packets to find the correct RSSI bytes. We have something that prints a value but we are unsure if this is the correct bytes or not.

The changes we made today will be updated in the master branch of our GitHub repository.

As we rewrite the MATLAB code in C++, we will have to implement matrix multiplication among other things but matrix multiplication will probably be the most confusing. I have been looking for C++ libraries that make better matrix objects and implement the multiplication. The two I have found are Boost and Eigen. Right now I think Eigen is the best choice since it is made for linear algebra and that is the only part we need. Tomorrow we will actually start converting the MATLAB to C++.

We still need to finish the parts order to send to Mr. Mattus so we can get the parts sooner rather than later.



## 28 February 2017

### 1 Work done in the lab for the day

#### 1.1 Converting the code to C++ and adding SLAM algorithm

We have started converting the MATLAB code into C++. Right now we are just going one function at a time. I spent most of the time in the lab getting the Eigen library to be in the package because there was not much in the way of tutorials and we are still not very familiar with cmake.

Using the C++ code we already have for working with the XBees and UART, we were able to test how fast measurements could be taken for a full revolution. Doing a measurement every  $9^\circ$ , we were able to complete a full rotation in about 26 s. This is a vast improvement from our previous code. This could probably be sped up even more by some clever coding in the callback (instead of a simple timeout function), but it is not necessary right now.

Kyle took a video of the new code running with a stop watch to show that it is actually working.

### 2 Making a better reflector for the system

Kyle also started gluing together the reflector. It already looks better than the other one. We are going to wait to attach it to the piece of plexiglass until we have figured out how everything will be attached.



## 1 March 2017

### 1 Work done in the lab for the day

#### 1.1 Converting the code to C++ and adding SLAM algorithm

Kyle and I each started working on C++ functions (converting from MATLAB). So far it is not overly difficult since we are using Eigen but there are some things in MATLAB that don't translate to C++ very well and it can be confusing to rewrite it. As always, the code we are writing is being put on GitHub so it is easier to keep track of everything.

It just occurred to me that we really don't need to write everything in C++ for the journal paper. We can just add ROS code to allow control from MATLAB instead. I might start writing this tomorrow in the lab so possibly Friday we can actually drive a robot around in the lab via MATLAB.

I have also been thinking about how to do real time plotting of the robot's environment. I am pretty confident that we could do it with MATLAB with relative ease but I think it would be cool to make an Android app. Obviously this is not necessary for the journal paper but it would very interesting to have for the final presentation and the student expo. Definitely something to consider once we submit the journal paper since the project will essentially be done at that point. For now I may just play with making some apps in my own time to see if it is something feasible for us to do.

### 2 New uploads for the day to Google Drive

The only thing uploaded today to the Google Drive was the spreadsheet for the second order that we need to complete, which I think we will finish doing tomorrow in the lab.



## 2 March 2017

### 1 Work done in the lab for the day

Its Thursday so we can't test anything on the robot today (since the robotics lab is using it).

#### 1.1 Writing a ROS node for use with Matlab

We are working on writing the C++ node that will run on the Beaglebone Black, allowing us to control it from Matlab. Since I got Eigen included in the package, it should be easier to do matrices for everything. Unfortunately the code we wrote to do all the measurements still uses vectors.

Instead of just changing all the code to use matrices, we just copied the data returned by our code into a matrix. This is a really lazy solution but right now we are just trying to get things working together. So far everything is compiling in my virtual machine so it should compile once we get it on the Beaglebone Black.

I think we have enough code now to actually try it on the Beaglebone. We have just been using scp to copy things over the wireless connection so it was easy to get it on there. Unfortunately, Eigen is not installed on the Beaglebone (I chose the minimum install for ROS kinetic when I installed it so it wasn't included).

We are going to ask Mr. Mattus tomorrow to connect the router he lent us to BUsecure so we are able to download thing. We can't do anything else with the code today so we finished up the second order. We are going to go over it tomorrow with Dr. Miah and then send it to Mr. Mattus.





## 3 March 2017

### 1 Weekly meeting to discuss progress with Dr. Miah

We had our weekly meeting today with Dr. Miah. It was a quick meeting since all we need to focus on is finishing the implementation for the journal paper. This should go a little faster now that we are using MATLAB to control everything via ROS.

### 2 Work done in the lab for the day

We were not able to find Mr. Mattus today to have him change the settings on the router so we are looking into a different way of installing the Eigen library. We found the library online at [packages.ubuntu.com](http://packages.ubuntu.com). We just searched for the Eigen package and the one dependency it requires that was not already installed on the Beaglebone Black. We used scp again to just transfer the two .deb files.

On the Beaglebone, we just used this command to install the packages (first the dependency then Eigen):

```
sudo dpkg -i <package-name>
```

#### 2.1 Writing a ROS node for use with Matlab

Once we got the two packages installed we were able to get our code to compile. We had a couple problems with getting the custom message to compile (things are compiling out of order). We added a dependency line in the CMakeLists.txt in order to force it to build in a certain order. The compilation still took awhile to compile.

Something weird is going on with the launch file. When we use roslaunch to start the node, it says that the environment variable `ROS_MASTER_URI` has not been defined in the environment. I forgot about this step so we just defined the variable now and tried again.

```
export ROS_MASTER_URI=http://192.168.1.3:11311
```

Yet again, the program will not run, it just says the variable is not defined, even though we are able to echo and read the environment variable from the terminal. Just to check, I tried running the node using `rostrun` instead of `roslaunch`. Strangely, the node runs just fine, apart from the GPIO not working. We had this issue before since accessing GPIO requires root privileges. In MATLAB I was able to connect to the master (on the Beaglebone), create a node, create a publisher, and publish a message to a topic created by the C++ ROS node. We took the code for working with the XBee out of the node (just commented out for now) and replaced it with a quick print function that prints on the console window for the Beaglebone. When the message is sent to the node, the callback is triggered and we see the test message.

This is a good sign that at least we are able to connect to the node but it does not address our issue with `roslaunch` though.

After some thinking, it seems like `roslaunch` is overwriting the environment variable instead of reading it like it is supposed to. Since `roslaunch` files are in XML, we used one of the XML tags (`< env >`) created for launch files to write the environment variable again from within the file. This produced the same result as before. We still think it is a problem with `roslaunch` and the environment variable but we have run out of time to work in the lab since both of us have prior obligations and need to leave.

## 4 March 2017

I was doing some more thinking about causes of the error we were having yesterday. Comparing the way `roslaunch` starts the node to how `roslaunch` does, I remembered that we had to use the `sudo` prefix in the node entry of the launch file. This is why the GPIO had been working before we added the ROS functionality to the code. Does using `sudo` change environment variables. Googling that, I was able to find that, in fact, environment variables are not preserved when using `sudo`. The solution is to add the environment variables we need to `/etc/environment`. This should fix the problem we were having with the node not starting correctly with `roslaunch`. Unfortunately I have no way to actually test that this works other than that I have found other people saying that their programs cannot access environment variables when they start them using `sudo` and this was the solution that other people gave.

Assuming it works, it should be pretty easy to get the robot doing something on Monday. We will at least be able to test that we can control the estimation subsystem can be controlled via ROS.

We already have the framework for the MATLAB code since it is going to be very similar to the simulation (except using ROS instead of the V-REP remote API). Later today, I think I am going to start writing the MATLAB code (more like copy and paste with some modifications).

Another thing we should do Monday is get the router settings changed so we can have Internet on the Beaglebone. I also realized that we didn't put any standoffs for the XBee board (the one in the reflector) on our order but I think we will be able to come up with some other solution instead. We should also talk to Mr. Mattus about those stands he mentioned he would be able to make for our beacons.

I have been trying now for awhile to just get the custom message we made for our data imported into MATLAB. Its imported now but I changed it so much just to get it to work that I don't even think it will work for our needs anymore. I looked through the list of supported ROS messages and there is one called *MultiEchoLaserScan*<sup>1</sup>. The structure looks like it would work for our application.

<sup>1</sup>[http://docs.ros.org/api/sensor\\_msgs/html/msg/MultiEchoLaserScan.html](http://docs.ros.org/api/sensor_msgs/html/msg/MultiEchoLaserScan.html)

Just thought about it a little more and decided it won't work since it is only a 1 dimensional array and we need 2 dimensional. I am going to try and make another message instead. This new message will follow a similar structure to the *MultiEchoLaserScan* message type but it will have 2 dimensional array for data and I will take out the parts that are not needed for our implementation.

The new message is in three parts. The smallest part (*radioResponse.msg*) holds the data from an XBee response packet. In next part holds the angle, an array of radio responses (the smallest part), and the number of elements in the array. The largest part (the one we will send back and forth using ROS) has a header, the minimum angle, maximum angle, the angle increment, and an array of radio response arrays (making it two dimensional).

I updated the rest of the config files for the ROS package so it would compile these new messages correctly. I also wrote a short MATLAB script that connects to the Beaglebone, makes a node, sends a request for observation data, then waits to receive the response. This should work assuming we fix the whole environment variable problem from yesterday first.

## 6 March 2017

### 1 Work done in the lab for the day

#### 1.1 Writing a ROS node for use with Matlab

We were able to get the environment variable issue from last Friday solved by just adding the environment variables we needed to */etc/environment* and the node launches. There is a new problem now, we cannot use the publisher for the data inside the callback function since it is not a global object.

We did some google searching and decided it is just easiest to make a class for the Matlab interface. We followed a style similar to <http://answers.ros.org/question/59725/publishing-to-a-topic-via-subscriber-callback-function/>, although it is a bit more complicated with the motor and everything else included. For this class, we just had it take the same inputs as the motor object so they could just be passed directly. This is not really the best way to do this and we could easily change it so the interface object takes the motor object as a parameter. One other way would be to rewrite the motor class so it uses virtual inheritance but this is a bit more than we need right now. Also, if we did it for the motor we would have to do it for the observer as well.

Once we got everything on the C++ side done we moved to the Matlab. I have already written enough for it to run (since that is what I used to test that the environment variables solution worked).

We are having some trouble receiving the message in Matlab. I set a breakpoint on the line where the data is received so the Matlab script could be run and we would be able to see things happening in the node. We also set it up so the messages would appear on my virtual machine (since I have ROS installed there). Yet again there was trouble with some environment variables. It turns out that each machine in the ROS network needs to be assigned a *ROS\_IP* so it can be found by other nodes. Fixing that, we were able to receive the messages on my laptop but only in Linux, not Matlab.

After thinking about the way the receive function works, it occurred to us that if the message was already sent and the breakpoint kept Matlab from receiving, the message wouldn't be received. Like we have to be waiting to receive the message when the mes-

6 March 2017

---

sage is sent and not the other way around. We removed the breakpoint and big surprise it worked fine.

Right now we are only sending a basic message using ROS (literally just the minimum and maximum angle) but it is being sent which is a huge improvement from Friday. Now we just need to get all the data that is gathered into the message so it can all be sent and we can just use Matlab to process it for now.

The code has been modified on GitHub in the *ros-matlab-1* branch.

## 7 March 2017

### 1 Work done in the lab for the day

#### 1.1 Writing a ROS node for use with Matlab

Since we were able to send basic messages yesterday, we are working on sending all of the data from the Beaglebone to Matlab using ROS. Kyle is working on the C++ part and I am doing the Matlab (since I have the necessary toolbox and he doesn't).

Kyle has the data being put into the message and I am able to retrieve it and read it but something is going wrong with angle passing because it is just a bunch of nonsense numbers. It is probably just something with how the data is being copied so for now he has changed it to a simple counter instead of the motor object's implementation of the angle.

Another thing that seems to be going wrong is that the data that is being sent is all the same between positions. This happens even if we move the beacons during the scan (so we know the RSSI will change). I think this is something to do with how data is transferred to the message (shallow copy vs. deep copy). We are currently using the *push\_back* functionality of the ROS messages to push the data into the variable. This should not be terribly difficult to fix.

One other issue, when we use ctrl-C to exit the node, the node does not exit correctly and we get an *flock()* error when we try to start the node again. I think this is because when you force close that way it just stops the program abruptly and does not do any cleanup so its like there are still objects open even though the program is not running anymore. We were trying to figure this out for awhile and did not figure it out until we were leaving. To fix it we just need to define how ctrl-C is handled. It will probably be something similar to <http://stackoverflow.com/questions/4250013/is-destructor-called-if-sigint-or-sigstp-issued> and <http://stackoverflow.com/questions/482702/using-sigint>. We just need to fill the function with all the cleanup functionality we need. Should be a pretty simple fix.





## 8 March 2017

### 1 Work done in the lab for the day

#### 1.1 Writing a ROS node for use with Matlab

We tried implementing the solutions we found online yesterday. To our surprise, none of them actually worked. We did notice that we were missing a parameter in our ROS initialization function (still worked since it was overloaded) and the parameter was for disabling signal handling. Our solution ended up being relatively simple. We added some curly brackets so we could force some variables out of scope then made a ROS topic that initiates the shutdown when a message is sent to it. This also allows the node to be shut down remotely.

Unfortunately this took most of the time we were in the lab so once we got that all fixed we only had a little time to work on transferring the data using ROS. There is still something wrong with the way things are being copied. We are going to spend all of our time tomorrow on it.



## 9 March 2017

### 1 Work done in the lab for the day

#### 1.1 Writing a ROS node for use with Matlab

We are looking at the data sending today. Kyle is trying to get the code we have working with the addition resizing functions included by ROS for messages. Whenever he tries to compile it has an error about incorrect syntax even though we cannot find the problem. The errors are too vague to be helpful unfortunately.

While Kyle is working on the Beaglebone, I am writing code in a virtual machine (so I can see if things compile). I have something that I think will work where data is consecutively pushed into larger containers. I think this will solve the issue of the copying we had a couple days ago and it is really simple so it is easy to implement.

Kyle added my code to the Beaglebone and everything compiled. Running the node, I was able to connect using ROS, trigger a scan, and retrieve the data. And this time the data was correct! This would have been a real issue if we weren't able to get it working so it is really good that we did.

Since we have the message passing done we can work on the processing now. It really wasn't that complicated, we just transferred the message into a cell array in Matlab then iterated through it. While it was iterating, we have it sorting the data according to which beacon it came from and the angle, that way it will be easier to use. Once it is all processed, we convert the end result (range and bearing for each beacon) to distance and return it in the same form as needed by the Matlab functions we already have.

We decided that we would stay a little later today so we can get more done (and because we want to use the robot and we will have to wait until the robotics lab is over to use it). So once we were done with all the data processing stuff, we worked on setting up the rosaria node for the Pioneer. This was actually a little more complicated than we anticipated. Fortunately we did not have to write it all because I was able to find the modified code from last year's robotics lab that implemented the code to reset the Pioneer's pose.

Once it was all compiled, which took awhile, we ran the node and I ran one of the

9 March 2017

---

programs from last year's robotics lab. The robot drove in a circle just as it was supposed to. Before we left, I took a copy of the entire ROS workspace and made a copy of the microSD card so we have a current backup. This will be uploaded to Google Drive and the code will be updated on Github.

## 10 March 2017

We are supposed to hear from ISIE today about whether or not our paper was accepted. Hopefully it is.

### 1 Work done in the lab for the day

#### 1.1 Writing a ROS node for use with Matlab

We were going to work in the lab today but then we realized that we can't do anything until the stepper motor is powered by the Pioneer. We were going to just connect it in parallel with the voltage regulator for the Beaglebone but the fuse on that connection is only for 1 A. If we connect the stepper motor with the controller (probably about 0.5 A) it may be too much for the fuse since we are also powering the Beaglebone on the same connection and that is also really close to being 1 A. I am going to email Mr. Mattus to see if it would be okay for us to open the robot, remove the microcontroller board, and attach wires to the motor power board to power the stepper motor. We were looking into this before and it should work fine. If a connector could be fitted to those wires it would be even better since we don't want to live wires just sitting uncovered on top of the robot (pretty dangerous...). An alternative to connecting to the motor power board would be to just get a bigger fuse and connect in parallel anyway.

### 2 Weekly meeting to discuss progress with Dr. Miah

We had our weekly meeting today. We updated Dr. Miah on our progress (pretty much ready to do experiments, just need the stepper motor power to be wired up). Most of the meeting was spent talking about what to do on the TII paper that is due at the end of March. Here is the list of things we need to do:

- Change figure 1A
  - Redraw in ipe (optional)
  - Rotate the environment within the figure so it is aligned with everything else
  - Add dashed circles around the RF modules to symbolize the radiation pattern

- Change figure 1B
  - Remove the XBee end devices block and replace with a couple blocks representing XBees (similar to project proposal)
  - Add dashed circles around the XBee antennas to symbolize the radiation pattern
- Write the related study section
  - Look at [5, 4] for examples of good related study sections
- Change figure 5
  - Look at the comments beneath this figure to know what to change (this figure is the subsystem block diagram from the project proposal)
- Performance Evaluation section
  - Add 2 figures for each subsection, each will have four subfigures (images from either the simulator or the video we will take of the experiment), similar to the ICRA paper that was submitted at the end of the summer
    - \* For now, just use missing figures as a placeholder until we get the actual results
  - Write explanations of the results (evaluate the performance)
  - Literally put in as many of the figures as we can (error, trajectories, etc.), we can figure out what to keep and what to get rid of later
- Add the algorithm right before the Performance Evaluation section using the *algorithm2e* package
- Rerun the V-REP simulations
  - Do two more simulations in V-REP, two different scenarios
  - Add more bearing estimation noise so it is more similar to the data we have already collected
  - Implement the fuzzy logic controller for calculating the linear velocity
    - \* Instead of turning as we drive, just turn the robot to face the next waypoint then apply the velocity
    - \* I already have code that will do this, just need to adapt it into our algorithm

In order to calculate the error between the robot's estimated position and its ideal position, we should use this equation for its ideal position.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + v_k T \frac{\begin{bmatrix} x_p \\ y_p \end{bmatrix} - \begin{bmatrix} x_k \\ y_k \end{bmatrix}}{\left\| \begin{bmatrix} x_p \\ y_p \end{bmatrix} - \begin{bmatrix} x_k \\ y_k \end{bmatrix} \right\|} \quad (1)$$

where  $k$  is the current iteration,  $v_k$  is the current linear velocity from the fuzzy logic controller, and  $p$  is the index of the current waypoint.





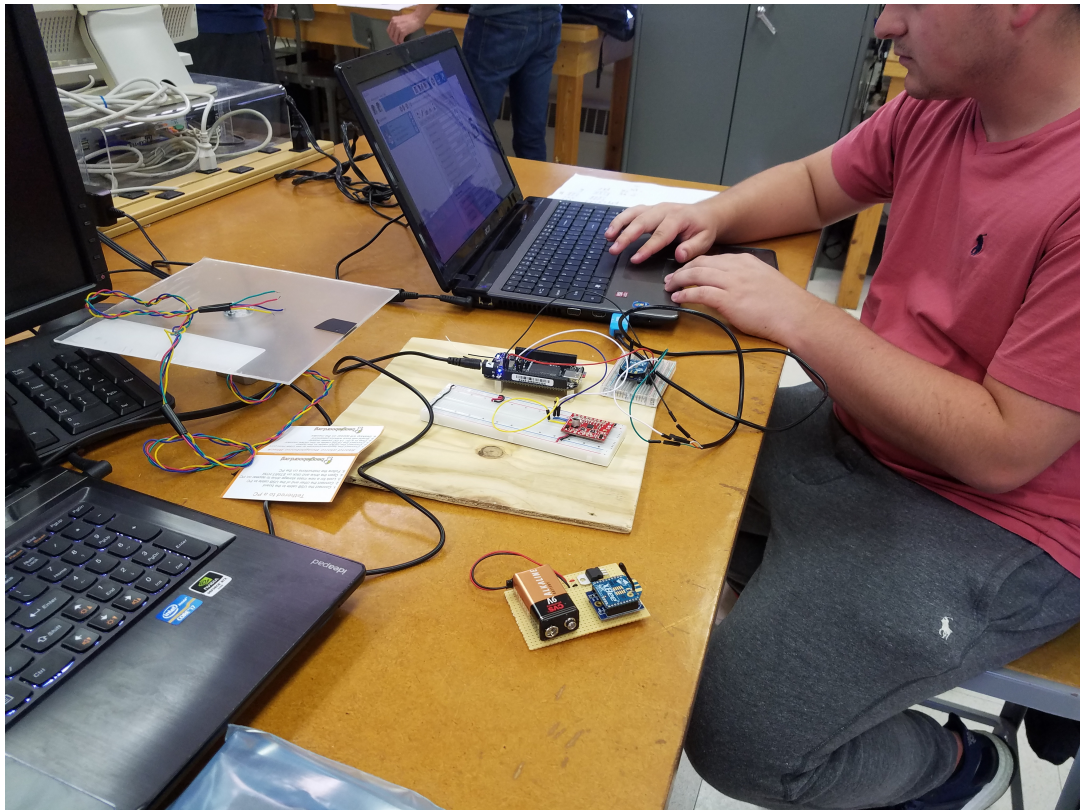
## 20 March 2017

First day back after break. Both of our conference papers are accepted. We made the revisions that reviewers suggested (really just some grammar or things that we were not able to do because of time). I worked on the journal paper over break and updated figures. We are still working on that.

### **1 Work done in the lab for the day**

Today we went to talk to Mr. Mattus about our parts order and powering the stepper motor. We got the parts and he said he would come to the lab tomorrow to help us with the Pioneer.

We spent the day today setting up the new BeagleBone Black (with built in wireless), setting up the new XBees, and putting on standoffs. This took most of the time in lab but will make the implementation much easier from now on because we just have to pick up the piece of wood that everything is mounted on.



**Figure 1:** Kyle programming the new XBees

## 21 March 2017

### 1 Work done in the lab for the day

We are soldering together all of the boards that hold our subsystems. A picture of the board for the mounting the main XBee within the reflector can be seen in Fig. 1.

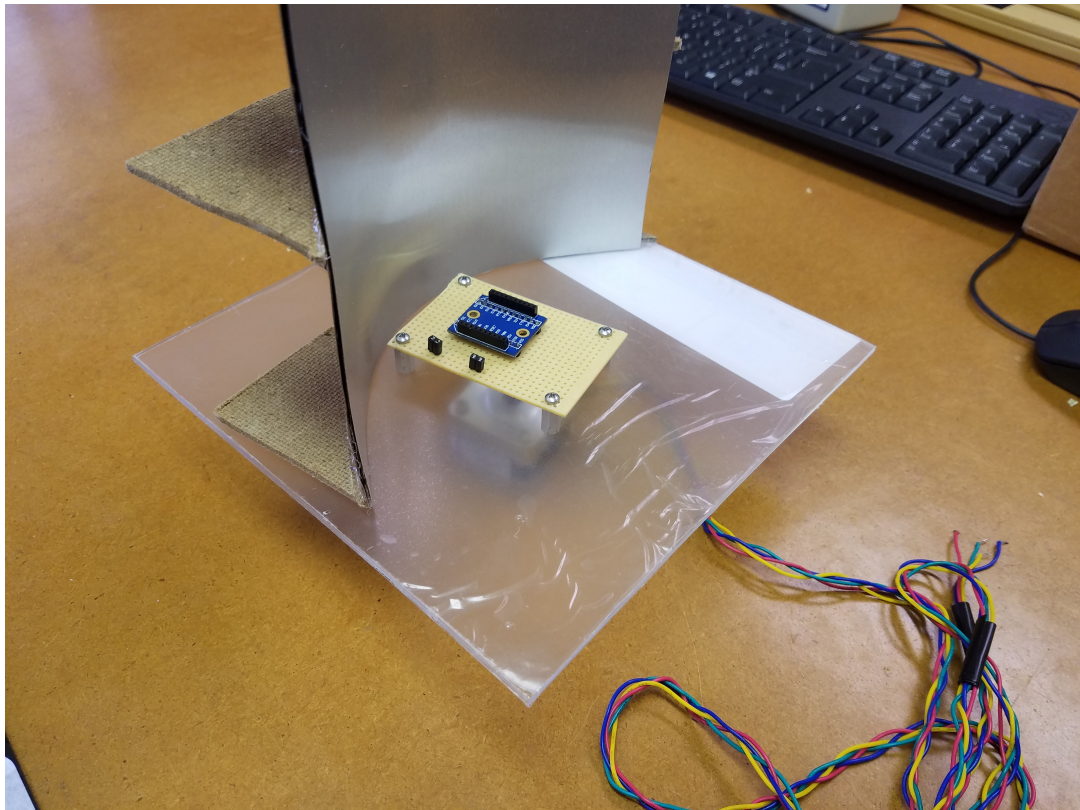
Mr. Schmidt also came and helped us wire up power for the stepper motor on the Pioneer. He gave us a DC power jack so we can plug our stepper motor controller board in easily. This board was harder to put together since there are more connects and some of the pins are very small. A picture of this board (close to being complete) can be seen in Fig. 2.

It did not take us long to finish soldering everything so we are going to try and run the MATLAB test program with everything set up (similar to Fig. 3 but with all the connections made).

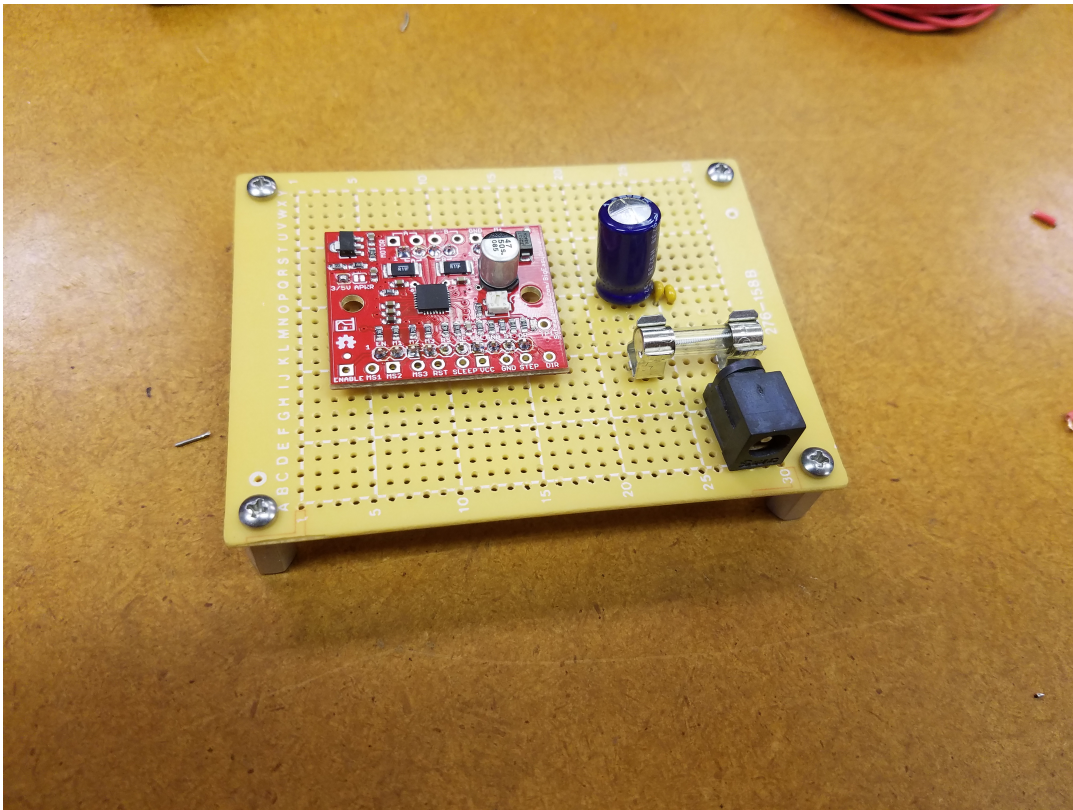
Once we had everything connected, we turned on the Pioneer (which also turns on everything else) and the motor controller board is now powered. Unfortunately, the BeagleBone is now not receiving any power at all. Just to make sure it isn't the BeagleBone, we plugged it into the wall adapter and it turns on without any issues at all.

We used a multimeter to check the voltage on the voltage regulator and we discovered that there isn't any power being delivered at all, so something must have gotten messed up when the other power cables were being put in. We also used the multimeter to check the voltage between the regulator and the fuse (still nothing), between the robot and the fuse (again nothing), and then we opened the robot and probed the screw terminals (12 V as expected). We suspect that the connection is not being made in the crimped connectors that Mr. Schmidt made for us. We emailed him to let him know that we would be needing his help again tomorrow but we will also go and look for him as well since we are running out of time to get this done.

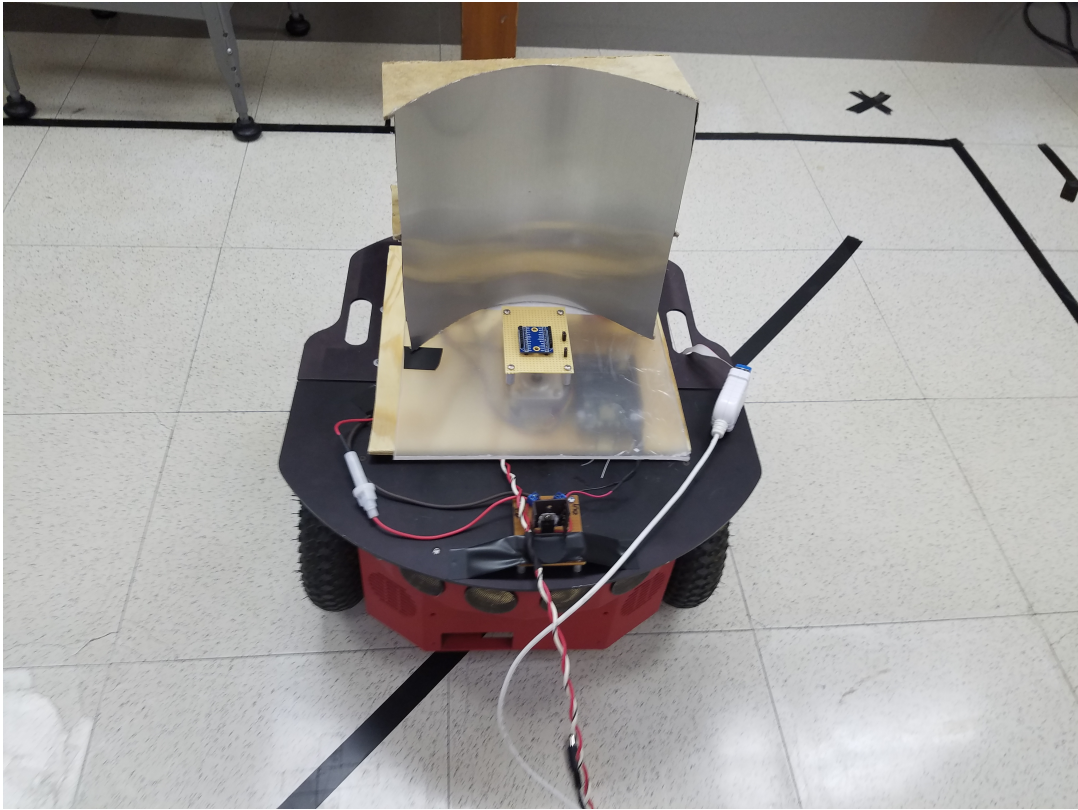
With this setback, we did not get to run any of our code today and we are actually a little behind where we want (need) to be. It looks like we might be staying a bit later on Thursday and Friday. If Mr. Schmidt can fix the power thing tomorrow, we should probably be able to do all the debugging necessary tomorrow too. That means that we can start running the experiments Thursday evening (since we have to wait until the lab is done using the Pioneer). We are assuming that we will be able to get all the experiments done Thursday and Friday, convert the videos we take into pictures that can be included in the picture, and do any other data processing before the weekend.



**Figure 1:** *Reflector subsystem*



**Figure 2:** *Work in progress picture of motor controller board with power connection*



**Figure 3:** *The final product of all our work today*

We are only including images of two different experimental cases in the paper but I think we will end up running it more so we can have more data to choose from.

Tomorrow I will go and talk to the AV people down the hall to see if I can get a little video camera like I did at the beginning of last semester. Shouldn't be an issue. I also need to make another version of the implementation code that uses the fuzzy logic controller that Dr. Miah shared with us today. Also shouldn't be that difficult but we need to use it since we say we are.

Another thing, for the paper we should have at least five references related to our work that were published within the last 18 months. We should also have two or three references that were published in the TII journal (this is an actual requirement for submission to the journal). We will work on this once we have all the experimental data collected and in the paper.

There is a lot we need to get done all at once because of our issue today so we are going to have to work a lot harder for the next couple of days.

Just thought of a couple more things we still need to do. I'll just list them since that is easier:

- Senior project website
- Student expo abstract
- Student expo poster
- CCECE presentation





## 22 March 2017

### 1 Work done in the lab for the day

Mr. Schmidt came and fixed the wiring for BeagleBone's power, it was something with the fuse. We can now power everything off the Pioneer. While Kyle finished assembling everything I started working on the simulation code for the next paper. Since we do not have 2 robots in the lab, it doesn't make sense that we should be using two robots in the simulation. I want the simulation to be as close to the real thing as possible.

I commented out all the code for the second robot and used the equation Dr. Miah gave us a couple days ago in order to estimate where the robot would be without the EKF-SLAM. The algorithm still works but this new estimate just seems really far off. I think it is something to do with the wheel diameter and the distance between the wheels being different in V-REP (this would affect applied velocities so they'd end up different than we want). I had to use the positions of the joints to try and get closer to the dimensions since they did up being a little bit different than those of the real Pioneer.

I'm running the simulation again and it still isn't working. Its like the new estimate is about  $\frac{4}{5}$  the size it should be. I have no idea why. I'm gonna look at this again later tonight.

It is really easy to setup the system on the robot now since it is all mounted on one board we can just pick up and carry. We did a couple tests of driving the robot around and doing observations. We were having some trouble with the rotation direction of the reflector. Turns out that I forgot that I made a function to set the direction when I wrote all the objects for the node and we just weren't using it. We changed all the direction code to use that function when doing observations and it all works fine now. We also took out one step since it was going past a full revolution. Once it does the first revolution, it now will do another in the opposite direction so the wires do not get tangled up around the motor.



## 23 March 2017

### 1 Work done in the lab for the day

We have almost everything assembled and ready to run. It looks a lot better than it did before (its more contained now). Here a couple pictures of it all together and ready to go. All we really did today was debugging, trying to get everything working correctly. We are really close to it running, just some strange things with data association and processing the observations correctly.

### 2 Weekly meeting to discuss progress with Dr. Miah

We had our meeting. Really just talked about what to change about the paper (figures, what sections to work on, etc). We have been very busy trying to get everything done in time for the journal paper.

Kyle is still finding references to include and I am working on the simulations in V-REP.

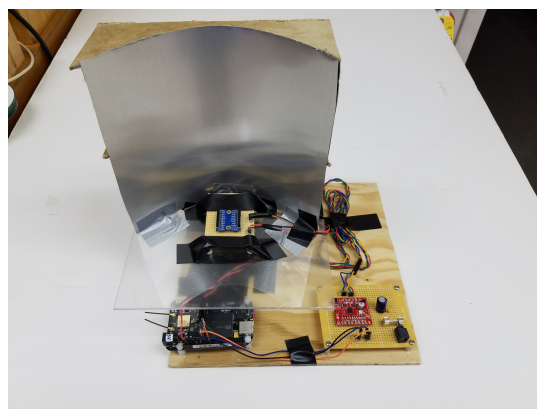
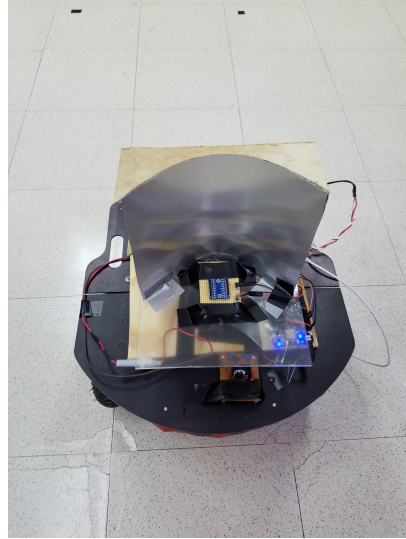


Figure 1: A picture of the system



**Figure 2:** *System on the robot from the back*



**Figure 3:** *System on the robot from the front*

## 27 March 2017

### 1 Work done in the lab for the day

Back in the lab again. We are still working on the debugging. Right now the issue is something with the ftags being generated in our observation processing. We discovered some peculiarities of the algorithm's code:

- ftags HAVE to be in order
- observations in the z matrix must be in the same order as ftag
- if there are  $N$  beacons, the beacons you use must be called *beacon1*, *beacon2*, ..., *beaconN*.

The first two make sense to us but the third does not. We are not going to waste time trying to make it general since time is short so we will just make sure we use consecutive beacons instead.

By the end of the day we were able to get the robot making observations without doing any movement and the observations are actually pretty decent, especially after the algorithm runs for awhile (since it is filtering as it goes). We did not get the chance to run any tests before we had to leave for prior engagements. I will be working on the simulations later tonight.

I was able to finish the first simulation before falling asleep. It is just a square trajectory like last time since it isn't complicated and a good general case.



## 26 March 2017

We are just in the computer lab today working on the paper (also it is Sunday). I am finishing up the second simulation first though. Kyle rewrote the related study section and improved what I had a lot. Once I finished the last simulation I added it to the to them to the paper in the spots I had put missing figures before. We also looked at and improved some of the figures.





## 27 March 2017

### 1 Work done in the lab for the day

We want to try and run some experiments today once we get it all working. We have added the fuzzy logic controller to the Matlab (copy and paste from the simulation basically). We are working on debugging that. We are having some issue with  $qTrue$  not being saved correctly. Doesn't affect the algorithm at all but it screws up the plots. It was something to do with the ordering of some code that gets the pose from the Pioneer. I also changed how some things are plotted since there were errors with the number of iterations not matching how much data there actually was. Now it just plots everything available based on size of the data matrix. Also changed some of the timing ( $qTrue$  would become undefined due to a 0 in the denominator, then every subsequent  $qTrue$  would become NaN).

We have the entire algorithm running now. It drives between 2 waypoints pretty well. We are now running it with 4 waypoints in a square and it is doing remarkably well (considering all the noise). Tomorrow we will be running all the experiments since we did not get to today. We can work on figures and writing tonight. It should not be too bad tomorrow since everything was working today just fine...



## 28 March 2017

All we are doing today is running the experiment and taking videos for the TII paper. We are making the plots and stuff as we go so we are going to put that all into the paper later.



## 29 & 30 March 2017

We are in the computer lab working on the TII paper (putting figures in, proofreading, finishing up sections). We are spending a lot of time working on this paper.

The paper was submitted in the evening so hopefully we are done with most of the work for senior project now (apart from final project reports and presentations).



## 3 April 2017

We took a break from working on senior project since we have been spending so much time on it all for the past couple weeks. Over the weekend the paper was actually withdrawn since they found our names on the paper and it was over the page limit of 8 pages.





## 4 April 2017

We are working on shortening the paper. Dr. Miah was able to get it reclassified as a special session paper and now we can have 10 pages total. We are going through it now and abbreviating the references and making the figures smaller or taking things out entirely.



## 5 April 2017

The paper is finally the correct length. We are proofreading one last time to make sure everything looks okay.

Dr. Miah resubmitted the paper to TII. We are working on the poster for the student expo now.



## 6 April 2017

Still working on the student expo. We are having to play with a lot of the spacing to get things looking good and fit everything we want.

Dr. Miah looked at it and asked that we put everything in bullets instead of paragraphs. We had been following the style of last year's winner so we thought ours was okay. We made the poster with bullet points and change some of the figures and add more math.



## 7 April 2017

Kyle met with Dr. Miah today to discuss the poster and any last changes that needed to be made (Jacob was out of town all day). It was more just going over some figures and some of the writing. The introduction, conclusion, and some of the formatting was changed. It should be ready for the expo now (we hope).





## 10 April 2017

### 1 Weekly meeting to discuss progress with Dr. Miah

We had our meeting today instead of last Friday since I was gone. We went over the poster, talked about the final project report (which is due today), and then the CCECE presentation.

For the poster, he asked that we change the title of the poster so it looks better when it is listed online with the other work.

I emailed a very rough draft of the final report since we not have time to make an actual one. It won't be hard for us to write the final report anyway since we have already written so many papers at this point.

For CCECE, we are scheduled to present on study day which obviously is not really an option for either of us. Dr. Miah is going to email the conference to see if we can get the presentation put on a different day (its gonna have to be Monday if anything).

Kyle is working on the website now and I'm just gonna start working on all the final presentation and reports and stuff.



## **11-13 April 2017**

We are presenting in the Student Expo so we really haven't had the time to go and work in the lab. If we worked on anything, it was just the final report.



## 14 April 2017

### 1 Weekly meeting to discuss progress with Dr. Miah

We had our weekly meeting today, it was pretty short as there isn't much that has happened since the last meeting. We discussed the content of the final report a little more, possible future work that later senior projects would be able to work on, and the presentations we need to make. Here is the list of the current things to do for senior project:

- Senior Project website
- Senior Project final report
- Senior Project final presentation
- Industrial Advisory Board presentation
- ISIE presentation
- CCECE presentation

Once we finish the final report it should be pretty straightforward to make the corresponding presentation. We will be working on the final report a little over the weekend.



## 17 April 2017

### 1 Work done in the lab for the day

We are finally back in the lab. We are just finishing up some last minute experiments so we have some more data to include in the final report. We also did a little tuning with the P controller in the motion control strategy so the turns are a little tighter. I have started cleaning up the code so it is more readable and easier to customize as well.

The experiment we ran today was just a small figure eight (something we haven't done before). It works decently well. I think since the robot's process noise is so low, there isn't much of a change in the estimation of its state. Not bad, but it is hard to get our predictions accurate so there is just a little bit of persistent noise but it is usually less than 10 cm.





## 18 April 2017

### 1 Work done in the lab for the day

We decided that we are going to run a longer experiment today. We have marked out waypoints so that the Pioneer does a loop around the entire lab (under desks unfortunately). It took a little while to get it all measured out to put it into the MATLAB code. We also had to do a bit of tuning with the velocity and the P controller since some of the waypoints are pretty far apart and the robot drives a little too fast.

We have finally started the robot on this big loop. Its working well enough to be included in the final report. The error for estimation is a little higher since the trajectory is longer. You can really see it for the estimation of the robot's position and a couple of the beacons.

Well it has been running for an hour and half and MATLAB just failed... We think that one of the batteries for the beacons died and caused an error in the algorithm which caused MATLAB to fail. Unfortunately when MATLAB crashed we lost all the data from the experiment. All we have is the value for  $\mathbf{q}$  before it failed and a picture of the trajectory from MATLAB. We are a little dissappointed but at least we can put the final error values for the beacon estimates and the picture of the final trajectory.

This is probably gonna be the last day we spend in the lab since we have to finish the final report and the final presentation. If we have time we might try the long experiment again so the data can be used in future.



## **18-19 April 2017**

We are just in the computer lab working on the final report. It doesn't make sense to write more about it in the notebook.



## 20 April 2017

### **1 Weekly meeting to discuss progress with Dr. Miah**

We had our meeting today instead of tomorrow since I won't be in Peoria. We just went over the final report and Dr. Miah gave us some things to fix and change.

We will just work in the computer lab the rest of today on the things Dr. Miah pointed out to us. We were hoping to start on the final presentation today but we will see.

We did not get the chance to start working on the presentation so that is what we will be doing all of next week.



## 23 April 2017

I started working on the slides for the final presentation. At this point it is really just the structure of the slides (pretty much the same as the structure of the final report). It should not be difficult to fill in the slides tomorrow with stuff from the final report.





## 24 April 2017

Kyle and I worked in the computer lab today on the final presentation. We also got the schedule for the Industrial Advisory Board presentation (which is this Friday in the afternoon) and the final presentations. For the final presentation, we are presenting on 5/2 from 3:00-3:30 PM. We are a little upset that we only get a half hour to present when other groups get an entire hour.

When we left the lab today there were 62 slides in our presentation. If we don't get more time we are gonna be really flying through the slides. If anything we will just keep talking anyway.

I am working on the video clips that will be embedded in some of the slides right now.



## 25-27 April 2017

We have been working on the final report and the slides. Also started working on the CCECE slides.

We received an update from the conference organizers. The way everything was registered, only one of us can present the paper. We do not need to drive to Canada for it so it will be a Skype presentation.

Since Kyle is the first author on the paper, he was chosen to present via Skype on Wednesday, May 3.

On the 27th we met with Dr. Miah to quickly go over our presentation since we have finished the slides. We have not really practiced it yet so this will be good to see what we really need to work on before the presentation on Tuesday. After we practiced a little we went and watched the presentations that are happening today.

Watching the presentations, I am a little less worried about our presentation. Some of them are pretty good but some are lacking results and explanation. If we practice on Monday, we should be in pretty good shape on Tuesday.



## **28 April 2017**

We had the industrial advisory board presentation today. Overall it went pretty well.



## 1 May 2017

We are practicing our final presentation today. It is right around 30 minutes long so that is really good. We presented it to Dr. Miah for practice and he liked it so I think we are pretty much set for tomorrow.





## 2 May 2017

Doing some last minute practice.

We are done with the presentation so this is the last entry in the lab notebook. It went really well.



## Bibliography

- [1] E. DiGiampaolo and F. Martinelli. Mobile robot localization using the phase of passive uhf rfid signals. *IEEE Transactions on Industrial Electronics*, 61(1):365–376, Jan 2014. 41
- [2] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and localization with rfid technology. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 1015–1020 Vol.1, April 2004. 41
- [3] B. N. Hood and P. Barooah. Estimating doa from radio-frequency rssi measurements using an actuated reflector. *IEEE Sensors Journal*, 11(2):413–417, Feb 2011. 41
- [4] F. Martinelli. A robot localization system combining rssi and phase shift in uhf-rfid signals. *IEEE Transactions on Control Systems Technology*, 23(5):1782–1796, Sept 2015. 41, 230
- [5] D. Song, C. Y. Kim, and J. Yi. Simultaneous localization of multiple unknown and transient radio sources using a mobile robot. *IEEE Transactions on Robotics*, 28(3):668–680, June 2012. 41, 230