LEMUR Project Proposal

Ben Limpanukorn *Last Updated: 2018-10-31*

Big Picture

Cooperative localization has many applications for systems that use multiple robot, transportation, and any other system that relies on inferring the state of the system (e.g. positions of the robots) from incomplete information. The CoLo Project aims to build a simulation environment to streamline testing and development of cooperative localization algorithms. The environment is capable of using real world data as well as simulated data to further evaluate the limits of the algorithm being tested.

By enabling faster testing and iteration of these algorithms, the CoLo Project will empower researchers with the tools they need to make multiple robot systems a viable solution for many of the world's problems.

1 Specific Project Scope

Currently, data is recorded by the individual robots and collected to be used afterwards for testing in the CoLo environment. In order for Cooperative Localization to occur in real-time, each of the robots must not only record information from their perspective, but also share it with each other and/or a centralized computer.

To solve this problem, some form of wireless networking (e.g. Bluetooth) is proposed to augment the already collected data with information about connection strength, and to communicate said data. Establishing a network also facilitates over-the-air updates to the robots and wireless retrieval of the recorded data at the end of a session. This problem will be considered solved once information can be sent from robot to robot, and reasonably accurate distance data can be derived from the connection strength.

2 Background

My prior experience working with Python will be necessary to utilize the rospy software package to establish a network connection over Bluetooth and extract information about its connection strength. My understanding of using a command line shell in Linux is also necessary for learning how the ROS framework functions in order to control the robot.

3 Related Work / References

A paper published in IEEE Xplore[1] details techniques for estimating the position of a wireless node using the positions of other nodes in the network. However, since these techniques are designed to be used on a larger scale it remains to be seen whether or not it will scale down for our purposes.

For Bluetooth communication integration with ROS, the textbook "Robot Operating System (ROS) for Absolute Beginners" by Lentin Joseph discusses using the PyBluez module to create a ROS package that acts as a driver for Bluetooth.

MIT's Albert Huang has also written "An Introduction to Bluetooth Programming"[2] document that details how to communicate over Bluetooth.

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4802193
[2] https://people.csail.mit.edu/albert/bluez-intro/x264.html

Goals / Deliverables / Tasks

The primary goal for the next two to three weeks is to develop a communication network between the robots in the system. This will enable the robots to share information from their perspective and cooperatively estimate the state of the system as a whole.

Goals

1. **Communication between two robots**

   The primary goal is to achieve two-way wireless communication between two robots. Achieving this will establish which technology—Bluetooth, WiFi, etc.—would be most suitable for our application.

   Deliverables

   (a) **Python script to ping another robot over Bluetooth** The first step before integrating communication with ROS is to test possible solutions with regular python scripts.

   Tasks

   i. **Test the PyBluez module**
   The PyBluez[3] module appears to be the most widely used module for Bluetooth communication and thus have the most complete documentation and support for its usage. Using these pre-written bindings would also improve cross-platform compatibility (Windows and Linux) if needed.

   ii. **Test native Python sockets for Bluetooth**
   Since version 3.3 Python[4][5] has native built-in support for communicating over bluetooth. However, the documentation about its usage is not as comprehensive as the PyBluez module.

   (b) **Custom ROS package that bridges ROS Topics over Bluetooth** After writing a Python script to communicate over Bluetooth, it must be integrated with ROS in order to communicate with the rest of the system.

   Tasks

   i. **Create a python script to echo messages over Bluetooth**
   When the script receives a message it will send the same message back over bluetooth while simultaneously logging the message.

   ii. **Create a python script to send messages over Bluetooth**
   The script should provide a ROS service that sends messages over Bluetooth to another computer and waits for the messages to be received.

   iii. **Measure and log connection information**
   Using the L2CAP protocol for Bluetooth, measure the number of dropped packets and latency

[3] https://github.com/pybluez/pybluez
[4] https://docs.python.org/3.3/library/socket.html
[5] http://blog.kevindoran.co/bluetooth-programming-with-python-3/

2. **Communication between multiple robots**

   Once two robots can communicate, we need to find the best way to allow one robot to connect to multiple other robots and vice versa.

   Deliverables

   (a) **Updated ROS package**

   This package will contain an extended version of the Bluetooth wrapper outlined by the goal above. In addition to the ability to communicate with another robot, this package should allow for networking between multiple other robots.

   Tasks

   i. **Multi-robot connection**
   The script should either periodically connect with specific other robots, or simultaneously maintain a connection between multiple other robots and itself.

   ii. **Automatic discovery of other robots in range**
   The ROS node should be able to automatically discover other robots in Bluetooth range.

   iii. **Information Relay**
   Besides providing data about the connection strength, the package should also provide a service that relays messages to other robots in the network.

Timeline

See the Goals / Deliverables / Tasks section above for more details.

Week 7: PyBluez Module and Python sockets for sending bluetooth messages (see goal 1-a above)

1) Test the PyBluez module
2) Test native Python sockets for Bluetooth

Week 8: Build a ROS package to encapsulate the connection logic (see 1-b)

1) Create a python script to echo messages over Bluetooth
2) Create a python script to send messages over Bluetooth
3) Measure and log connection information

Week 9: Connect multiple robots together (see 2-a)

1) Multi-robot connection
2) Automatic discovery of other robots in range
3) Information Relay