Robotics 1: Intro to ROS

Eric McCann,
Nat Tuck
What is ROS?

- R.O.S., the Robot Operating System, is a collection of tubes.
Why ROS?

- Automagically manages dependencies
- Automagically handles communication
  - Interprocess
  - Intermachine
  - Includes an arsenal of neat tools
  - Powerful visualization and transformation tools + libraries
- ROS primarily supports C++, python, lisp.
  - ROSJava exists (but undergoes superfluous backwards compatibility-breaking reorganizations)
  - ROS# (Microsoft .NET (C#)) exists, and is a work in progress
- Regardless of the language, the form and function of a ROS program is the same.
What are master/nodes/topics?

- **Master** maintains a list of nodes, as well as lists of what each is looking to *subscribe* to, and offering to *publish*.
- **Nodes** are individual executables. In C++, every node has a main function. Nodes register with master, and then offer to *publish* and ask to *subscribe* to *topics*. Each node can have as many publishers/subscribers/*other things we’re skipping for now*) as it wants.
- **Topics** are directional “stuff tubes”. Each has a name, and a known data type.
Publishers and subscribers

- Publishers send messages on a topic
- Subscribers receive messages on a topic by way of a callback (a void function that takes a message of a given type as an argument)
Messages

- Messages are composed of literals (strings, integers, booleans) and/or other messages.
Instant messaging example

- 2 people, communicating over 2 directed channels. A talks to B over “/AtoB”, and B talks to A over “/BtoA”. Messages contain 2 strings, one being the sender’s name, the other being the message.
Instant message example
What’s a service?

- Don’t worry about this so much for now, BUT... a service is essentially a bi-directional topic.
- The service server has a callback with a prototype that can have void or non-void argument and return. The service client calls that callback, and receives the return back from it synchronously.
What are we going to do with ROS?

Use it as a framework for ALL of the software for our rover!

... but first, some *tutorials* on [http://www.ros.org](http://www.ros.org) and *something else*...