Multi-Robot Control (MRC)

W1: Tools
Plan of the Day

Overview of robotic control tools
• Linux, ROS, MATLAB and Git – Oh My!

Assignment 1 Brief

Setup Computers
• User accounts
• Wifi access
Working Hypothesis

- Students focusing on unmanned systems should be able to...
  - Proficient *users of ROS*
    Which means they need to be users of Linux, Git, etc.
  - Proficient *developers in MATLAB/Simulink*

This is a hypothesis; I'm interested in your experience and observations throughout the course.
Implementation Tools

Linux, Ubuntu

ROS

MATLAB/Simulink

Git
What is ROS

Robot Operating System (ROS)

History
- Originally developed at Stanford AI Lab and Willow Garage
- Currently maintained by Open Robotics

Plumbing – Inter-process network communication
Tools – Introspection, visualization, debugging and automation
Capabilities – Pre-built algorithms
Ecosystem – Community, documentation, tutorials, etc.
Why ROS?

Inter-process network communication
• Modularization by enabling programs (nodes) to communicate

Very large user community
• We don't have to re-invent the wheel
• Community support/knowledge
• Increasing commercial compliance

Introspection and tools
Encourages good practice – ROS conventions

But there are good alternatives
• JAUS, LCM, MOOS, etc.
Turtlebot Example

360° LiDAR for SLAM & Navigation
Scalable Structure
Single Board Computer (Raspberry Pi)
OpenCR (ARM Cortex-M7)
DYNAMIXEL x 2 for Wheels
Sprocket Wheels for Tire and Caterpillar
Li-Po Battery 11.1V 1,800mAh

```
/joy_node  /joy  /teleop_twist_joy  /cmd_vel  /turtlebot3_core
            /scan
             /version_info
             /imu
             /sensor_state
             /joint_states
          /turtlebot3_diagnostics
              /robot_state_publisher
                  /tf_static
                      /tf
```
Linux

Why Linux?
- Standard support for development.
- Despite the steep learning curve, software is simpler and more powerful.
- ROS (and most other frameworks) are much easier to use in a Linux environment.
- Used by lower level computational elements (e.g., Raspberry Pi).

Ubuntu
- One of many distributions of Linux.
- Simplified software distribution (package management).
- Well integrated with ROS.
The Linux Command Line

Why?
• A complementary user interface model.
• The art of Unix:
  – Using collections of small programs to do big jobs
  – Avoiding manual tasks
• “Necessary” for ROS.
A Standard NPS Robotics Environment

Ubuntu 16.04 LTS (Long-term support)

ROS Kinetic Kame

MATLAB 2017b
Simulink
- Robotic System Toolkit
Git

Version Control Software (VCS)
- Seamlessly keep all edits and history
- Collaboration and sharing
- Revert to prior edits
- Use code from others

Git vs. Mercurial vs. Subversion, etc.

Why?
- Knowing the basics provides access to lots of community code.
- Pervasive use by ROS community
“Simple” Git Workflows

Existing Repository
- Clone (get a working copy)
- Make changes...
- Stage and Commit (local check-in)
- Push (remote check-in)

New Repository
- Make new space gitlab server via web interface
- Init local copy
- Stage/Commit
- Push to server

In the Future
- Pull
- Make changes...
- Stage/commit
- Push

In the Future
- Pull
- Make changes...
- Stage/commit
- Push
Assignment 1

Edit the Robotics Computation Wiki

Linux Command Line
- File/directory manipulation
- (A word about text editors)

Git
- Cloning a repository
  - Workflow: Clone/pull, change, commit, push
- Creating a repository

If possible, do the reading in front of a computer!
Next Steps

Today
• Everyone should have an account and start on Assignment 1

Tomorrow
• Goal is to finish Assignment 1!