Baxter_ROS 04/03/2015 ROS Indigo

BAXTER and ROS image

This report is written assuming you have read and understood the *Introduction to Baxter*, *The Baxter User's Guide*, and the *Turtlesim Guide* written by members of the Baxter Lab at UHCL.

Contents

Contents

Table of Figures

⚠

ROS TERMS

Before a beginner even opens a web tutorial or book or sees a ROS video, it is helpful to learn a few terms that pertain to ROS. These terms describe the main components of a ROS system.

Table 1. ROS Useful Terms		
Item	Type	Comment
Repositories	A **software repository** is a storage location from which software packages may be retrieved and installed on a computer.	http://en.wikipedia.org/wiki/Software_repository GitHub is used to download the ROS packages used by the Baxter system: http://sdk.rethinkrobotics.com/wiki/Workstation_Setup
Packages	Contains files to allow execution of ROS programs	A package typically contains source files and executable scripts that can be BASH, Python, or other code.
Manifest Package.xml	Information about a package	The manifest defines properties about the package such as the package name, version numbers, authors, maintainers, and dependencies on other packages.
ROS Master	Registers the name and location of each node.	Allows nodes to communicate. Nodes can be in different computers.
Parameter Server	Data types that define certain information for nodes.	Certain nodes require parameters to define aspects of the node.
Nodes	Processes that execute commands.	Executable code written in Python or C++ usually. Python nodes use the client library rospy
Topic	Name of a message.	For example, Baxter's cameras "publish" the image they receive as a topic with a name that indicates it is a camera image.
Services	Allows communication between nodes.	Used by nodes to communicate with other nodes and request a response.
Messages	Data sent between nodes.	Messages are "published" by a node and "subscribed to" by another node.
Bags	Data storage for messages.	Used to save and playback data such as sensor data.

Table 1 ROS Terms

The following tables define the help for the various ROS commands. The work station is communicating with Baxter after the Baxter shell is executed.

```
tlharmanphd@D125-43873:~/ros_ws$ ./baxter.sh
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$
```

rospack help, list, depend, find

```
tlharmanphd@D125-43873:/$ rospack help
USAGE: rospack <command> [options] [package]
  Allowed commands:
    help
    cflags-only-I     [--deps-only] [package]
    cflags-only-other [--deps-only] [package]
    depends           [package] (alias: deps)
    depends-indent    [package] (alias: deps-indent)
    depends-manifests [package] (alias: deps-manifests)
    depends-msgsrv    [package] (alias: deps-msgsrv)
    depends-on        [package]
    depends-on1       [package]
    depends-why --target=<target> [package] (alias: deps-why)
    depends1          [package] (alias: deps1)
    export [--deps-only] --lang=<lang> --attrib=<attrib> [package]
    find [package]
    langs
    libs-only-L     [--deps-only] [package]
    libs-only-l     [--deps-only] [package]
    libs-only-other [--deps-only] [package]
    list
    list-duplicates
    list-names
    plugins --attrib=<attrib> [--top=<toppkg>] [package]
    profile [--length=<length>] [--zombie-only]
    rosdep  [package] (alias: rosdeps)
    rosdep0 [package] (alias: rosdeps0)
    vcs  [package]
    vcs0 [package]
  Extra options:
    -q     Quiets error reports.

 If [package] is omitted, the current working directory
 is used (if it contains a manifest.xml).
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rospack list
```

The list for Baxter is very long. For example, the package for Baxter examples in directory
baxter_examples /home/tlharmanphd/ros_ws/src/baxter_examples
is just one of many.

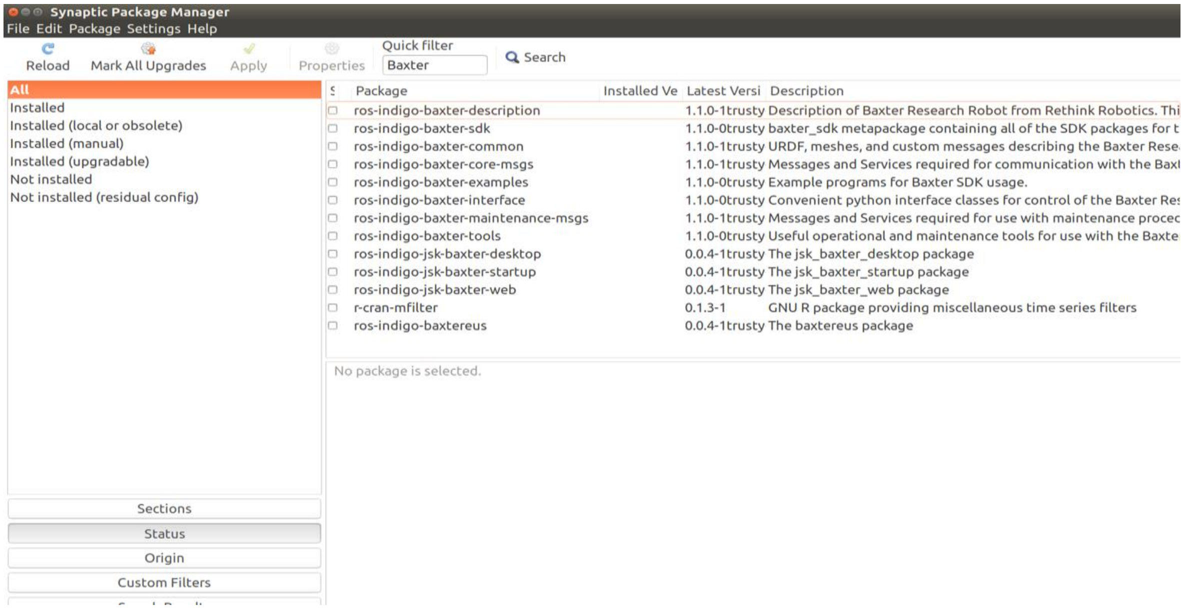
```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rospack depends baxter_examples
            genmsg                (45 total dependencies)
            .
            .

[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rospack find baxter_examples
            /home/tlharmanphd/ros_ws/src/baxter_examples
```

More details will be given later.

BAXTER PACKAGES

Synaptic Package Manager can list the packages for Baxter. It requires a password.



S	Package	Installed Ve	Latest Versi	Description
□	ros-indigo-baxter-description		1.1.0-1trusty	Description of Baxter Research Robot from Rethink Robotics. Thi
□	ros-indigo-baxter-sdk		1.1.0-0trusty	baxter_sdk metapackage containing all of the SDK packages for t
□	ros-indigo-baxter-common		1.1.0-1trusty	URDF, meshes, and custom messages describing the Baxter Rese
□	ros-indigo-baxter-core-msgs		1.1.0-1trusty	Messages and Services required for communication with the Baxt
□	ros-indigo-baxter-examples		1.1.0-0trusty	Example programs for Baxter SDK usage.
□	ros-indigo-baxter-interface		1.1.0-0trusty	Convenient python interface classes for control of the Baxter Res
□	ros-indigo-baxter-maintenance-msgs		1.1.0-1trusty	Messages and Services required for use with maintenance procec
□	ros-indigo-baxter-tools		1.1.0-0trusty	Useful operational and maintenance tools for use with the Baxte
□	ros-indigo-jsk-baxter-desktop		0.0.4-1trusty	The jsk_baxter_desktop package
□	ros-indigo-jsk-baxter-startup		0.0.4-1trusty	The jsk_baxter_startup package
□	ros-indigo-jsk-baxter-web		0.0.4-1trusty	The jsk_baxter_web package
□	r-cran-mfilter		0.1.3-1	GNU R package providing miscellaneous time series filters
□	ros-indigo-baxtereus		0.0.4-1trusty	The baxtereus package

No package is selected.

Figure 1 Synaptic Package Manager

Terminal list of packages – Edited to list Baxter Packages

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rospack list > packages3_5_2015.txt
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ ls
baxter_old.sh  build       install            rosparameters.txt   src
baxter.sh      CarolsPics  packages3_5_2015.txt  rosparameters.txt~
baxter.sh~     devel       realtimeloop.txt      run_baxter
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ gedit packages3_5_2015.txt
```

```
baxter_core_msgs /home/tlharmanphd/ros_ws/src/baxter_common/baxter_core_msgs      (Used later)
baxter_description /home/tlharmanphd/ros_ws/src/baxter_common/baxter_description
baxter_examples /home/tlharmanphd/ros_ws/src/baxter_examples                        (Used later)
baxter_gazebo /home/tlharmanphd/ros_ws/src/baxter_simulator/baxter_gazebo
baxter_ikfast_left_arm_plugin /home/tlharmanphd/ros_ws/src/moveit_robots/baxter/baxter_ikfast_left_arm_plugin
baxter_ikfast_right_arm_plugin
/home/tlharmanphd/ros_ws/src/moveit_robots/baxter/baxter_ikfast_right_arm_plugin
baxter_interface /home/tlharmanphd/ros_ws/src/baxter_interface                      (Used later)
baxter_maintenance_msgs /home/tlharmanphd/ros_ws/src/baxter_common/baxter_maintenance_msgs
baxter_moveit_config /home/tlharmanphd/ros_ws/src/moveit_robots/baxter/baxter_moveit_config
baxter_sim_controllers /home/tlharmanphd/ros_ws/src/baxter_simulator/baxter_sim_controllers
baxter_sim_hardware /home/tlharmanphd/ros_ws/src/baxter_simulator/baxter_sim_hardware
baxter_sim_io /home/tlharmanphd/ros_ws/src/baxter_simulator/baxter_sim_io
baxter_sim_kinematics /home/tlharmanphd/ros_ws/src/baxter_simulator/baxter_sim_kinematics
baxter_tools /home/tlharmanphd/ros_ws/src/baxter_tools
```

BAXTER FILES

Files in ros_ws

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ ls -la
        total 1432
        drwxrwxr-x  7 tlharmanphd tlharmanphd   4096 Apr  5 13:22 .
        drwxr-xr-x 65 tlharmanphd tlharmanphd   4096 Apr  5 13:14 ..
        -rwxrw-r--  1 tlharmanphd tlharmanphd   6137 Mar 16 21:50 baxter_old.sh
        -rwxrwxr-x  1 tlharmanphd tlharmanphd   6138 Mar 16 22:15 baxter.sh
        -rwxrwxr-x  1 tlharmanphd tlharmanphd   6122 Mar 16 22:14 baxter.sh~
        drwxrwxr-x 14 tlharmanphd tlharmanphd   4096 Mar 18 15:50 build
        drwxrwxr-x  2 tlharmanphd tlharmanphd   4096 Mar 25 10:00 CarolsPics
        -rw-rw-r--  1 tlharmanphd tlharmanphd     98 Mar 16 21:43 .catkin_workspace
        drwxrwxr-x  6 tlharmanphd tlharmanphd   4096 Mar 18 15:50 devel
        drwxrwxr-x  6 tlharmanphd tlharmanphd   4096 Mar 16 22:13 install
        -rw-rw-r--  1 tlharmanphd tlharmanphd  14013 Apr  5 13:22 packages3_5_2015.txt
        -rw-rw-r--  1 tlharmanphd tlharmanphd  27977 Apr  4 14:13 realtimeloop.txt
        -rw-rw-r--  1 tlharmanphd tlharmanphd 674547 Apr  3 19:25 rosparameters.txt
        -rw-rw-r--  1 tlharmanphd tlharmanphd 674547 Apr  3 19:24 rosparameters.txt~
        -rwxr-xr-x  1 tlharmanphd tlharmanphd   9376 Dec  5 12:01 run_baxter
        drwxrwxr-x  9 tlharmanphd tlharmanphd   4096 Mar 18 15:46 src
```

The important files are the shell **baxter.sh** and the program **run_baxter**. The directory **src** holds Baxter packages files.

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ cd src
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src$ ls -la
        total 44
        drwxrwxr-x 9 tlharmanphd tlharmanphd 4096 Mar 18 15:46 .
        drwxrwxr-x 7 tlharmanphd tlharmanphd 4096 Apr  5 13:22 ..
        drwxrwxr-x 4 tlharmanphd tlharmanphd 4096 Mar 16 21:45 baxter
        drwxrwxr-x 7 tlharmanphd tlharmanphd 4096 Mar 16 21:46 baxter_common
        drwxrwxr-x 8 tlharmanphd tlharmanphd 4096 Mar 16 21:46 baxter_examples
        drwxrwxr-x 6 tlharmanphd tlharmanphd 4096 Mar 16 21:45 baxter_interface
        drwxrwxr-x 9 tlharmanphd tlharmanphd 4096 Mar 16 22:09 baxter_simulator
        drwxrwxr-x 6 tlharmanphd tlharmanphd 4096 Mar 16 21:45 baxter_tools
        lrwxrwxrwx 1 tlharmanphd tlharmanphd   49 Mar 16 21:43 CMakeLists.txt ->
        /opt/ros/indigo/share/catkin/cmake/toplevel.cmake
        drwxrwxr-x 9 tlharmanphd tlharmanphd 4096 Mar 18 15:46 moveit_robots
        -rw-rw-r-- 1 tlharmanphd tlharmanphd  777 Mar 16 22:10 .rosinstall
        -rw-rw-r-- 1 tlharmanphd tlharmanphd  681 Mar 16 22:08 .rosinstall.bak
```

BAXTER CORE MESSAGES

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src$ ls
        baxter         baxter_examples   baxter_simulator  CMakeLists.txt
        baxter_common  baxter_interface  baxter_tools      moveit_robots
        [baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src$ cd baxter_common
        [baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter_common$ ls -la
                total 44
                drwxrwxr-x 7 tlharmanphd tlharmanphd 4096 Mar 16 21:46 .
                drwxrwxr-x 9 tlharmanphd tlharmanphd 4096 Mar 18 15:46 ..
                drwxrwxr-x 2 tlharmanphd tlharmanphd 4096 Mar 16 21:46 baxter_common
                drwxrwxr-x 4 tlharmanphd tlharmanphd 4096 Mar 16 21:46 baxter_core_msgs
                drwxrwxr-x 4 tlharmanphd tlharmanphd 4096 Mar 16 21:46 baxter_description
                drwxrwxr-x 3 tlharmanphd tlharmanphd 4096 Mar 16 21:46 baxter_maintenance_msgs
                -rw-rw-r-- 1 tlharmanphd tlharmanphd 1648 Mar 16 21:46 CHANGELOG.rst
                drwxrwxr-x 8 tlharmanphd tlharmanphd 4096 Mar 16 22:11 .git
                -rw-rw-r-- 1 tlharmanphd tlharmanphd  453 Mar 16 21:46 .gitignore
                -rw-rw-r-- 1 tlharmanphd tlharmanphd 1504 Mar 16 21:46 LICENSE
                -rw-rw-r-- 1 tlharmanphd tlharmanphd 2083 Mar 16 21:46 README.rst
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter_common/baxter_core_msgs/msg$ ls -la
    total 112
    drwxrwxr-x 2 tlharmanphd tlharmanphd 4096 Mar 16 21:46 .
    drwxrwxr-x 4 tlharmanphd tlharmanphd 4096 Mar 16 21:46 ..
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   46 Mar 16 21:46 AnalogIOState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   45 Mar 16 21:46 AnalogIOStates.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   82 Mar 16 21:46 AnalogOutputCommand.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd 1160 Mar 16 21:46 AssemblyState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   37 Mar 16 21:46 AssemblyStates.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd  384 Mar 16 21:46 CameraControl.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   90 Mar 16 21:46 CameraSettings.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   63 Mar 16 21:46 CollisionAvoidanceState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   44 Mar 16 21:46 CollisionDetectionState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   90 Mar 16 21:46 DigitalIOState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   46 Mar 16 21:46 DigitalIOStates.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   80 Mar 16 21:46 DigitalOutputCommand.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd  842 Mar 16 21:46 EndEffectorCommand.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd 1060 Mar 16 21:46 EndEffectorProperties.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd 1290 Mar 16 21:46 EndEffectorState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   95 Mar 16 21:46 EndpointState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   39 Mar 16 21:46 EndpointStates.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd  102 Mar 16 21:46 HeadPanCommand.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   42 Mar 16 21:46 HeadState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd  207 Mar 16 21:46 ITBState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   77 Mar 16 21:46 ITBStates.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd  135 Mar 16 21:46 JointCommand.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd  156 Mar 16 21:46 NavigatorState.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd   90 Mar 16 21:46 NavigatorStates.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd 1115 Mar 16 21:46 RobustControllerStatus.msg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd 1620 Mar 16 21:46 SEAJointState.msg
```

BAXTER SCRIPTS IN BAXTER EXAMPLES – THE PYTHON SCRIPTS

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src$ ls
    baxter         baxter_examples   baxter_simulator  CMakeLists.txt
    baxter_common  baxter_interface  baxter_tools      moveit_robots
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src$ cd baxter_examples
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter_examples$ ls -la
    total 60
    drwxrwxr-x 8 tlharmanphd tlharmanphd 4096 Mar 16 21:46 .
    drwxrwxr-x 9 tlharmanphd tlharmanphd 4096 Mar 18 15:46 ..
    drwxrwxr-x 2 tlharmanphd tlharmanphd 4096 Mar 16 21:46 cfg
    -rw-rw-r-- 1 tlharmanphd tlharmanphd 1911 Mar 16 21:46 CHANGELOG.rst
    -rw-rw-r-- 1 tlharmanphd tlharmanphd  850 Mar 16 21:46 CMakeLists.txt
    drwxrwxr-x 8 tlharmanphd tlharmanphd 4096 Mar 16 22:11 .git
    -rw-rw-r-- 1 tlharmanphd tlharmanphd  466 Mar 16 21:46 .gitignore
    drwxrwxr-x 2 tlharmanphd tlharmanphd 4096 Mar 16 21:46 launch
    -rw-rw-r-- 1 tlharmanphd tlharmanphd 1504 Mar 16 21:46 LICENSE
    -rw-rw-r-- 1 tlharmanphd tlharmanphd 1402 Mar 16 21:46 package.xml
```

```
-rw-rw-r-- 1 tlharmanphd tlharmanphd 3179 Mar 16 21:46 README.rst
drwxrwxr-x 2 tlharmanphd tlharmanphd 4096 Mar 16 21:46 scripts
-rw-rw-r-- 1 tlharmanphd tlharmanphd  254 Mar 16 21:46 setup.py
drwxrwxr-x 3 tlharmanphd tlharmanphd 4096 Mar 16 21:46 share
drwxrwxr-x 4 tlharmanphd tlharmanphd 4096 Mar 16 21:46 src
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter_examples$ cd scripts
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter_examples/scripts$ ls -la
total 164
drwxrwxr-x 2 tlharmanphd tlharmanphd  4096 Mar 16 21:46 .
drwxrwxr-x 8 tlharmanphd tlharmanphd  4096 Mar 16 21:46 ..
-rwxrwxr-x 1 tlharmanphd tlharmanphd  3632 Mar 16 21:46 analog_io_rampup.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  3731 Mar 16 21:46 digital_io_blink.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  4570 Mar 16 21:46 gripper_action_client.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  6345 Mar 16 21:46 gripper_cuff_control.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  8624 Mar 16 21:46 gripper_joystick.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  7835 Mar 16 21:46 gripper_keyboard.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  4095 Mar 16 21:46 head_action_client.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  4073 Mar 16 21:46 head_wobbler.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  5347 Mar 16 21:46 ik_service_client.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  7245 Mar 16 21:46 joint_position_file_playback.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  7927 Mar 16 21:46 joint_position_joystick.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  6608 Mar 16 21:46 joint_position_keyboard.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  6819 Mar 16 21:46 joint_position_waypoints.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  3494 Mar 16 21:46 joint_recorder.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  7375 Mar 16 21:46 joint_torque_springs.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  5548 Mar 16 21:46 joint_trajectory_client.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd 13543 Mar 16 21:46 joint_trajectory_file_playback.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  5719 Mar 16 21:46 joint_velocity_puppet.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  5508 Mar 16 21:46 joint_velocity_wobbler.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  4172 Mar 16 21:46 navigator_io.py
-rwxrwxr-x 1 tlharmanphd tlharmanphd  3776 Mar 16 21:46 xdisplay_image.py
```

CD .. to the src directory to see the other package file directories.

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src$ ls
baxter         baxter_examples   baxter_simulator  CMakeLists.txt
baxter_common  baxter_interface  baxter_tools      moveit_robots
```

CD to the other directories with Python scripts to see the contents. For examples:

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter_tools/scripts$ ls
calibrate_arm.py   enable_robot.py  tare.py       update_robot.py
camera_control.py  smoke_test.py    tuck_arms.py
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter_interface/scripts$ ls
gripper_action_server.py  head_action_server.py  joint_trajectory_action_server.py
```

These scripts can be used as a start for those wishing to write their own Python scripts for Baxter. The Rethink wiki gives more information: http://sdk.rethinkrobotics.com/wiki/Foundations#Writing_Programs

rosnode, rostopic, rosservice help

```
tlharmanphd@D125-43873:~$ rosnode help
        rosnode is a command-line tool for printing information about ROS Nodes.

        Commands:                                                              Example
                rosnode ping      test connectivity to node        ($ rosnode ping <node>)
                rosnode list      list active nodes
                rosnode info      print information about node   ($ rosnode info <node>)
                rosnode machine list nodes running on a particular machine or list machines
                rosnode kill      kill a running node
                rosnode cleanup   purge registration information of unreachable nodes
        Type rosnode <command> -h for more detailed usage, e.g. 'rosnode ping -h'

tlharmanphd@D125-43873:~$ rosnode list -h
Usage: rosnode list

Options:
  -h, --help  show this help message and exit
  -u          list XML-RPC URIs
  -a, --all   list all information        rosnode kill       kill a running node
        rosnode cleanup   purge registration information of unreachable nodes
```

For Baxter, the list contains 21 nodes. Clear the screen:

```
tlharmanphd@D125-43873:~/ros_ws$ clear
```

```
tlharmanphd@D125-43873:/$ rostopic help
rostopic is a command-line tool for printing information about ROS Topics.

Commands:
        rostopic bw       display bandwidth used by topic
        rostopic echo     print messages to screen
        rostopic find     find topics by type
        rostopic hz       display publishing rate of topic
        rostopic info     print information about active topic
        rostopic list     list active topics
        rostopic pub      publish data to topic
        rostopic type     print topic type

Type rostopic <command> -h for more detailed usage, e.g. 'rostopic echo -h'
```

rosparam help

```
tlharmanphd@D125-43873:/$ rosparam help
rosparam is a command-line tool for getting, setting, and deleting parameters from the ROS Parameter Server.

Commands:
        rosparam set       set parameter
        rosparam get       get parameter
        rosparam load      load parameters from file
        rosparam dump      dump parameters to file
        rosparam delete    delete parameter
        rosparam list      list parameter names

tlharmanphd@D125-43873:~$ rosparam list | more   (To see on screen)
```

Save to a file:

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rosparam list >
rosparameters.txt
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ ls
baxter_old.sh  baxter.sh~  CarolsPics  install           run_baxter
baxter.sh      build       devel       rosparameters.txt  src
```

Examples:

```
tlharmanphd@D125-43873:~$ rosparam get /rosversion
'1.11.8

tlharmanphd@D125-43873:~$ rosparam get /rosdistro
'indigo
```

```
tlharmanphd@D125-43873:/$ rosmsg help
rosmsg is a command-line tool for displaying information about ROS Message types.
Commands:
        rosmsg show      Show message description
        rosmsg list      List all messages
        rosmsg md5       Display message md5sum
        rosmsg package   List messages in a package
        rosmsg packages  List packages that contain messages
Type rosmsg <command> -h for more detailed usage
```

```
tlharmanphd@D125-43873:/$ rosservice help
Commands:
        rosservice args    print service arguments
        rosservice call    call the service with the provided arguments
        rosservice find    find services by service type
        rosservice info    print information about service
        rosservice list    list active services
        rosservice type    print service type
        rosservice uri     print service ROSRPC uri
Type rosservice <command> -h for more detailed usage, e.g. 'rosservice call -h'
```

Use the $**rosservice list** command to see the services for the active node. For Baxter, the list is long.

rosbag help

```
tlharmanphd@D125-43873:/$ rosbag help
Usage: rosbag <subcommand> [options] [args]

A bag is a file format in ROS for storing ROS message data. The rosbag command can record, replay and manipulate
bags.

Available subcommands:
   check              Determine whether a bag is playable in the current system, or if it can be migrated.
   compress           Compress one or more bag files.
   decompress         Decompress one or more bag files.
   filter   Filter the contents of the bag.
   fix      Repair the messages in a bag file so that it can be played in the current system.
   help
   info     Summarize the contents of one or more bag files.
   play     Play back the contents of one or more bag files in a time-synchronized fashion.
   record             Record a bag file with the contents of specified topics.
   reindex            Reindexes one or more bag files.

For additional information, see http://wiki.ros.org/rosbag
```

Table 2 ROS Help Information

BAXTER-ROS NODES, TOPICS, AND MESSAGES USING /robot/state

```
tlharmanphd@D125-43873:~$ cd ~/ros_ws
tlharmanphd@D125-43873:~/ros_ws$ ls
baxter_old.sh  baxter.sh~  CarolsPics  install           rosparameters.txt~  src
baxter.sh      build       devel       rosparameters.txt  run_baxter
tlharmanphd@D125-43873:~/ros_ws$ ./baxter.sh
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ . run_baxter untuck
        Today is Sat Apr  4 13:12:07 CDT 2015
        [INFO] [WallTime: 1428171127.370967] Untucking arms
        [INFO] [WallTime: 1428171127.637309] Moving head to neutral position
        [INFO] [WallTime: 1428171127.637770] Untucking: One or more arms Tucked; Disabling
        Collision Avoidance and untucking.
        [INFO] [WallTime: 1428171135.067889] Finished tuck
```

Node Information for topic /robot/state

```
 [baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic echo
/robot/state | more
        enabled: True
        stopped: False
        error: False
        estop_button: 0
        estop_source: 0
        ---
        enabled: True
        stopped: False
        error: False
        estop_button: 0
        estop_source: 0
        ---
        enabled: True
        stopped: False
        error: False
```

CNTL+c to stop or use | more to see one screen at a time.

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ clear
```

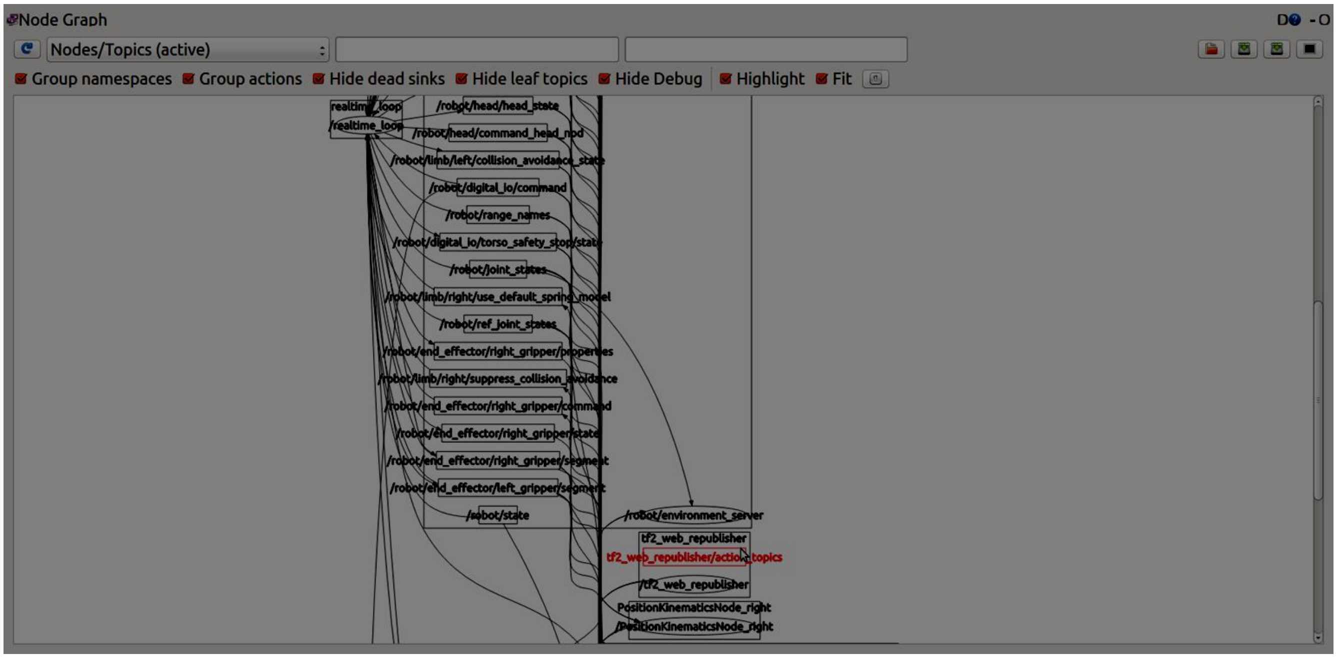
ROSNODE /robot/state WITH BAXTER
Open a new window. Most ROS commands have help screens that are usually helpful.

Let the /robot/state topic execute so it can be viewed with rqt_graph:

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic echo /robot/state
```

In another terminal window, let rqt_graph run in the **background.**

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rqt_graph&
```



Node Graph

Nodes/Topics (active)

☑ Group namespaces ☑ Group actions ☑ Hide dead sinks ☑ Hide leaf topics ☑ Hide Debug ☑ Highlight ☑ Fit

realtime_loop
/realtime_loop
/robot/head/head_state
/robot/head/command_head_nod
/robot/limb/left/collision_avoidance_state
/robot/digital_io/command
/robot/range_names
/robot/digital_io/torso_safety_stop/state
/robot/joint_states
/robot/limb/right/use_default_spring_model
/robot/ref_joint_states
/robot/end_effector/right_gripper/properties
/robot/limb/right/suppress_collision_avoidance
/robot/end_effector/right_gripper/command
/robot/end_effector/right_gripper/state
/robot/end_effector/right_gripper/segment
/robot/end_effector/left_gripper/segment
/robot/state
/robot/environment_server
tf2_web_republisher
tf2_web_republisher/action_topics
/tf2_web_republisher
PositionKinematicsNode_right
/PositionKinematicsNode_right

Figure 2 rqt_graph for /realtime_loop and /robot_state

In the graph, /realtime_loop is publishing the robot state.

NODE INFORMATION FOR /realtime_loop

Save very long printout to a file:

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rosnode info /realtime_loop > realtimeloop.txt
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ ls
baxter_old.sh  baxter.sh~  CarolsPics  install          rosparameters.txt   run_baxter
baxter.sh      build       devel       realtimeloop.txt  rosparameters.txt~  src
```

EDITED OUTPUT

```
Node [/realtime_loop]
Publications:
 * /robot/itb/right_itb/state [baxter_core_msgs/ITBState]
 * /robot/limb/left/stiffness [MotorControlMsgs/Stiffness]
 * /robot/analog_io/left_hand_range/value_uint32 [std_msgs/UInt32]
 * /robot/digital_io/torso_brake/state [baxter_core_msgs/DigitalIOState]
.
* /robot/joint_states [sensor_msgs/JointState]
 * /robot/sonar_states [MotorControlMsgs/SonarStates]
 * /robot/head/head_state [baxter_core_msgs/HeadState]
 * /robot/analog_io/torso_fan/state [baxter_core_msgs/AnalogIOState]
 * /robot/ref_joint_names [MotorControlMsgs/StringArray]
 * /robot/analog_io_names [MotorControlMsgs/StringArray]
 * /robot/analog_io/torso_fan/value_uint32 [std_msgs/UInt32]
 * /robot/digital_io/left_itb_light_outer/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/limit_switch_1/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/left/commanded_endpoint_state [baxter_core_msgs/EndpointState]
 * /robot/digital_io/torso_left_itb_button3/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_right_itb_button0/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/torso_right_itb_wheel/state [baxter_core_msgs/AnalogIOState]


Subscriptions:
 * /robot/end_effector/right_gripper/rsdk/set_state [unknown type]
 * /robot/end_effector/left_gripper/segment [MotorControlMsgs/URDFSegment]
 * /robot/end_effector/right_gripper/command [baxter_core_msgs/EndEffectorCommand]
 * /robot/limb/right/set_feed_forward_weights [unknown type]
.
 * /robot/set_super_enable [unknown type]
 * /robot/sonar/head_sonar/lights/set_green_level [unknown type]
 * /robot/limb/left/command_joint_position [unknown type]
 * /robot/limb/right/suppress_gravity_compensation [unknown type]

Services:
 * /load_controller
 * /reload_controller_libraries
 * /switch_controllers
 * /robot/limb/left/set_gc_enable
 * /list_controllers
 * /robot/limb/right/set_gc_enable
 * /robot/limb/left/set_gc_tare
 * /robot/limb/right/set_gc_tare
 * /realtime_loop/set_logger_level
 * /realtime_loop/get_loggers
 * /list_controller_types
 * /unload_controller

contacting node http://uhclbaxter.local:41055/ ...
```

```
Pid: 18579
Connections:
 * topic: /rosout
   * to: /rosout
   * direction: outbound
   * transport: TCPROS
 * topic: /diagnostics
   * to: /diagnostic_aggregator
   * direction: outbound
   * transport: TCPROS
 * topic: /diagnostics
   * to: /rosbag_diagnostics
   * direction: outbound
   * transport: TCPROS
 * topic: /robot/state
   * to: /rostopic_16006_1428173899591
   * direction: outbound
* transport: TCPROS
```

TOPIC /robot/state

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic info /robot/state
        Type: baxter_core_msgs/AssemblyState

        Publishers:
         * /realtime_loop (http://uhclbaxter.local:41055/)

        Subscribers:
        * /rostopic_16006_1428173899591 (http://172.29.64.201:59193/)
```

NODE INFORMATION

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rosnode info
/rostopic_16006_1428173899591
        --------------------------------------------------------------------------------
        Node [/rostopic_16006_1428173899591]
        Publications:
         * /statistics [rosgraph_msgs/TopicStatistics]
         * /rosout [rosgraph_msgs/Log]

        Subscriptions:
         * /robot/state [baxter_core_msgs/AssemblyState]

        Services:
         * /rostopic_16006_1428173899591/set_logger_level
         * /rostopic_16006_1428173899591/get_loggers


        contacting node http://172.29.64.201:59193/ ...
        Pid: 16006
        Connections:
         * topic: /rosout
            * to: /rosout
            * direction: outbound
            * transport: TCPROS
         * topic: /statistics
            * to: /rqt_gui_py_node_15976              (rqt_graph)
            * direction: outbound
            * transport: TCPROS
         * topic: /robot/state
            * to: /realtime_loop (http://uhclbaxter.local:41055/)
            * direction: inbound
            * transport: TCPROS
```

MESSAGES

If a topic publishes a message, we can determine the message type and read the message. This is shown in the example to determine the color of the background for the turtle.

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic type /robot/state
        baxter_core_msgs/AssemblyState
```

The message type in the case of the topic /robot/state is
baxter_core_msgs/AssemblyState .

The word “type” in this context is referring to the concept of a
data type . It’s important to understand message types because they determine the content of the messages. That is, the message type of a topic tells you what information is included in each message on that topic, and how that information is organized.

> From the message *type* we can fine the format of the message. Be sure to note that Color in the message type starts with a capital letter. http://wiki.ros.org/rostopic

rosmsg show

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rosmsg show baxter_core_msgs/AssemblyState
        uint8 ESTOP_BUTTON_UNPRESSED=0
        uint8 ESTOP_BUTTON_PRESSED=1
        uint8 ESTOP_BUTTON_UNKNOWN=2
        uint8 ESTOP_BUTTON_RELEASED=3
        uint8 ESTOP_SOURCE_NONE=0
        uint8 ESTOP_SOURCE_USER=1
        uint8 ESTOP_SOURCE_UNKNOWN=2
        uint8 ESTOP_SOURCE_FAULT=3
        uint8 ESTOP_SOURCE_BRAIN=4
        bool enabled
        bool stopped
        bool error
        uint8 estop_button
        uint8 estop_source
```

http://wiki.ros.org/msg

To understand the format of the message it is necessary to find the message type. The types include integers of 8, 16, 32, or 64 bits, floating point numbers, strings and other formats. The structure of the message type is:

```
        <field>  <constant>
```

where the field defines the type of data and the constant is the name. For example, the message format has **uint8 for information about the emergency stop button.** This indicates that if we wish to modify the value, an 8-bit unsigned integer is needed. The value is thus in the range of 0-255. The states enabled, stopped, or error are Boolean values – true or false.

To see the data being published, use the rostopic echo command:

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic echo /robot/state
enabled: True
stopped: False
error: False
estop_button: 0
estop_source: 0
---
```

Only one of the continuous outputs shown. The output indicates that Baxter is enabled and is not stopped or does not have an error.

For these exercises, keep the /robot/state topic publishing. Open other terminal windows as needed.

rostopic options

Determine the options for rostopic:

rostopic help

```
tlharmanphd@D125-43873:~$ rostopic help
        rostopic is a command-line tool for printing information about ROS Topics.

        Commands:
                rostopic bw     display bandwidth used by topic  (publications per second)
                rostopic echo   print messages to screen
                rostopic find   find topics by type
                rostopic hz     display publishing rate of topic
                rostopic info   print information about active topic
                rostopic list   list active topics
                rostopic pub    publish data to topic
                rostopic type   print topic type

        Type rostopic <command> -h for more detailed usage, e.g. 'rostopic echo -h'
```

We have already used rostopic **echo, info, list, and type.**

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic hz
/robot/state
        subscribed to [/robot/state]
        average rate: 100.483
                min: 0.005s max: 0.011s std dev: 0.00059s window: 86
        average rate: 100.227
                min: 0.005s max: 0.014s std dev: 0.00059s window: 186
        average rate: 100.149
                min: 0.005s max: 0.014s std dev: 0.00049s window: 286
        average rate: 100.121
                min: 0.001s max: 0.019s std dev: 0.00078s window: 385
```

CNTL+c to stop.

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic find
baxter_core_msgs/AssemblyState
                /robot/limb/left/state
                /robot/assembly/head/stat
                /robot/assembly/left/state
                /robot/assembly/right/state

                /robot/state
                /robot/head/state
                /robot/limb/right/state
                /robot/assembly/torso/state
```

These options will all yield the robot state as shown before. For example,

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic echo
/robot/limb/left/state
        enabled: True
        stopped: False
        error: False
        estop_button: 0
        estop_source: 0
        ---
```

BAXTER CORE MESSAGES PYTHON

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ cd
/opt/ros_ws/install/lib/python2.7/dist-packages/baxter_core_msgs/msg

[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:/opt/ros_ws/install/lib/python2.7/dist-
packages/baxter_core_msgs/msg$ ls
        _AnalogIOState.py         _CollisionAvoidanceState.py   _EndEffectorState.py   _ITBStates.py
        _AnalogIOState.pyc        _CollisionAvoidanceState.pyc  _EndEffectorState.pyc  _ITBStates.pyc
        _AnalogIOStates.py        _CollisionDetectionState.py   _EndpointState.py
        _JointCommand.py
        _AnalogIOStates.pyc       _CollisionDetectionState.pyc  _EndpointState.pyc
        _JointCommand.pyc
        _AnalogOutputCommand.py   _DigitalIOState.py            _EndpointStates.py
        _NavigatorState.py
        _AnalogOutputCommand.pyc  _DigitalIOState.pyc           _EndpointStates.pyc
        _NavigatorState.pyc
        _AssemblyState.py         _DigitalIOStates.py           _HeadPanCommand.py
        _NavigatorStates.py
        _AssemblyState.pyc        _DigitalIOStates.pyc          _HeadPanCommand.pyc
        _NavigatorStates.pyc
        _AssemblyStates.py        _DigitalOutputCommand.py      _HeadState.py
        _RobustControllerStatus.py
        _AssemblyStates.pyc       _DigitalOutputCommand.pyc     _HeadState.pyc
        _RobustControllerStatus.pyc
        _CameraControl.py         _EndEffectorCommand.py        __init__.py            _SEAJointState.py
        _CameraControl.pyc        _EndEffectorCommand.pyc       __init__.pyc
        _SEAJointState.pyc
        _CameraSettings.py        _EndEffectorProperties.py     _ITBState.py
        _CameraSettings.pyc       _EndEffectorProperties.pyc    _ITBState.pyc
```

ENDPOINT STATES

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic echo /robot/limb/right/endpoint_state
header:
  seq: 747739
  stamp:
    secs: 1428265132
    nsecs: 369195141
  frame_id: "
pose:
  position:
    x: 0.695893267361
    y: -0.0699519381125
    z: 0.241383037347
  orientation:
    x: -0.0799101178444
    y: 0.978632327302
    z: 0.105964529508
    w: 0.15704986314
twist:
  linear:
    x: -0.0025545689611
    y: -0.00591651247959
    z: -0.00312210371769
  angular:
    x: 0.000532478387338
    y: 0.0284426535359
    z: -0.030825033121
wrench:
  force:
    x: -1.21743285656
    y: 0.526600658894
    z: 3.0805721283
  torque:
    x: 0.226078554988
    y: 0.192814916372
    z: 0.407820761204
---
```

Continuous output of the position, orientation, and velocities. Compare to the position on the turtle window. CNTL+c to stop output.

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic echo /robot/limb/left/endpoint_state
header:
  seq: 760053
  stamp:
    secs: 1428265255
    nsecs: 508759513
  frame_id: "
pose:
  position:
    x: 0.688082138738
     y: 0.329212550401
    z: 0.305543966434
  orientation:
    x: 0.0431297746156
    y: 0.998513811224
    z: 0.0274511197475
    w: 0.0188792838943
twist:
  linear:
    x: 0.00588143782412
    y: -0.00797073232737
    z: 0.00642156644791
  angular:
    x: 0.00195915492857
    y: -0.0141418956114
    z: -0.0233123913764
wrench:
  force:
    x: -1.27401793003
    y: -0.154179736972
    z: 1.06045055389
  torque:
    x: -0.134043604136
    y: -0.129927709699
    z: -0.0974206402898
---
```

MOVE BAXTER'S HEAD

Head Joints

Head Position and State

The head state topic will give you the current `pan` angle (side-to-side) of the head and report boolean status flags if the robot is currently moving its head or nodding. *Note: Flags may not report 'true' values until after the first respective movement command is sent.*

Component IDs:

`head_nod`, `head_pan`

Head State

`/robot/head/head_state` (baxter_core_msgs-HeadState)

- `pan` field gives you the current angle (radians) of the head. 0 is forward, `-pi/2` to Baxter's right, and `+pi/2` to Baxter's left.
- `isPanning` and `isNodding` are boolean fields that will switch to True while the robot is executing a command. *Note: The `isPanning` field is initialized to True upon startup and will update thereafter.*

Head (Joint) State

`/robot/joint_states` (sensor_msgs-JointState)

- The position of the head may also be determined from the `joint_state` message. *Note: The 'nod' joint will never update, as it is only a binary state.*

Head Movement Control

Pan Head

`/robot/head/command_head_pan` (baxter_core_msgs-HeadPanCommand)

- `target` sets the target angle. 0.0 is straight ahead.
- `speed` is an integer from [0-100], 100 = max speed.
- Setting an angle in the command_head_pan topic does not gurantee the head will get to that position. There is a small deband around the reference angle around the order of +/- 0.12 radians.

Nod Head

`/robot/head/command_head_nod` (std_msgs-Bool)

- Send True to nod!

Example:

```
# Check head position/state:
$ rostopic echo /robot/head/head_state
# Move (pan) head side-to-side:
$ rostopic pub /robot/head/command_head_pan baxter_core_msgs/HeadPanCommand -- 0.0 100
```

```
# Make head nod up-down:
$ rostopic pub /robot/head/command_head_nod std_msgs/Bool True
```

Find type of message for HEAD TOPICS:

```
        /robot/head/command_head_nod
        /robot/head/command_head_pan
        /robot/head/head_state
        /robot/head/state
```

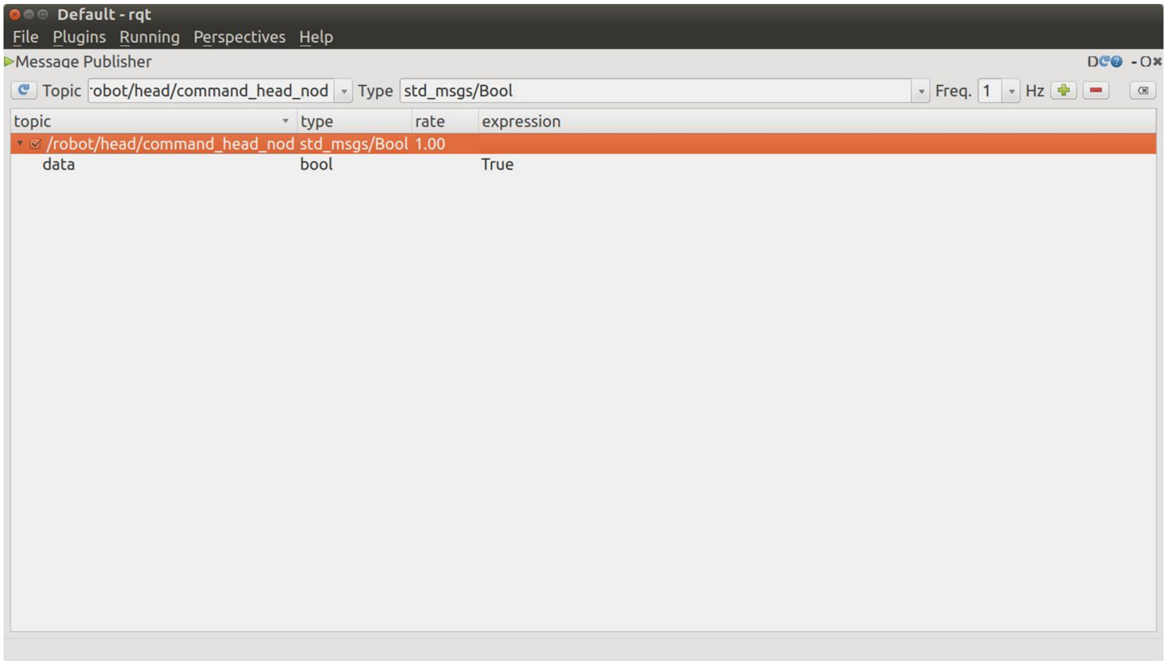
```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic type
/robot/head/command_head_pan
        baxter_core_msgs/HeadPanCommand
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rosmsg show
baxter_core_msgs/HeadPanCommand
        float32 target
        int32 speed
```

RQT PUBLISHER

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rqt
```

Select publisher and publish head nod at 1 Hz. Select the topic and add to list +, then make sure the expression is T**rue** since the expression here is Boolean. The head will nod at a rate determined by the selection of **Hz.** Be sure to check the box and right click on the topic and select **Expand Selected.**



Topic ·obot/head/command_head_nod Type std_msgs/Bool Freq. 1 Hz

```
topic                          type          rate   expression
☑ /robot/head/command_head_nod std_msgs/Bool 1.00
    data                       bool                 True
```

Figure 3 rqt Publisher

http://wiki.ros.org/rqt_publisher

Note the other options for $rqt by selecting Plugins in the menu bar.

rqt_common_plugins: *rqt_action* | *rqt_bag* | *rqt_bag_plugins* | *rqt_console* | *rqt_dep* | *rqt_graph* | *rqt_image_view* | *rqt_launch* | *rqt_logger_level* | *rqt_msg* | *rqt_plot* | *rqt_publisher* | *rqt_py_common* | *rqt_py_console* | *rqt_reconfigure* | *rqt_service_caller* | *rqt_shell* | *rqt_srv* | *rqt_top* | *rqt_topic* | *rqt_web*

Terminal Publishing to Move Baxter's Head

The following command will send a single message to Baxter that caused the head to nod once.

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic pub
/robot/head/command_head_nod std_msgs/Bool 1
        publishing and latching message. Press ctrl-C to terminate
```

(From Rob Linsalata on Google forum)

COMBINE TWO COMMANDS

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic type
/robot/head/command_head_pan | rosmsg show
        float32 target
        int32 speed
```

ERROR HEAD PAN?

TRY

```
  $ rostopic pub /robot/head/command_head_pan baxter_core_msgs/HeadPanCommand -- 0.0 100
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic pub
/robot/head/command_head_pan baxter_core_msgs/HeadPanCommand "0.8 50.0"
ERROR: Not enough arguments:
 * Given: ['0.8 50.0']
 * Expected: ['target', 'speed']

Args are: [target speed]
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic pub
/robot/head/command_head_pan baxter_core_msgs/HeadPanCommand ['0.8',' 50.0']


[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic pub
/robot/head/command_head_pan baxter_core_msgs/HeadPanCommand [0.8,  50]
Usage: rostopic pub /topic type [args...]

rostopic: error: Argument error: while parsing a flow node
expected the node content, but found '<stream end>'
  in "<string>", line 1, column 6:
    [0.8,
         ^

[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic pub
```

```
/robot/head/command_head_pan baxter_core_msgs/HeadPanCommand [0.8 50]
Usage: rostopic pub /topic type [args...]

rostopic: error: Argument error: while parsing a flow sequence
  in "<string>", line 1, column 1:
    [0.8
    ^
expected ',' or ']', but got '<stream end>'
  in "<string>", line 1, column 5:
    [0.8
        ^

[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic pub
/robot/head/command_head_pan baxter_core_msgs/HeadPanCommand '[0.8, 50]'
ERROR: Not enough arguments:
 * Given: [[0.8, 50]]
 * Expected: ['target', 'speed']

Args are: [target speed]


[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rostopic pub
/robot/head/command_head_pan baxter_core_msgs/HeadPanCommand '[0.8, 50]'
ERROR: Not enough arguments:
 * Given: [[0.8, 50]]
 * Expected: ['target', 'speed']

Args are: [target speed]
```

(We can use ROS services to manipulate the turtle and perform other operations.)

Services: (**he format is $rosservice call <service> <arguments>)**

```
* /turtle1/teleport_absolute
 * /turtlesim/get_loggers
 * /turtlesim/set_logger_level
 * /reset
 * /spawn
 * /clear
 * /turtle1/set_pen
 * /turtle1/teleport_relative
 * /kill
```

The node **/turtlesim** publishes three topics and subscribes to the **/turtle1/cmd_vel** topic**.** The services for the node are listed also.

PARAMETER SERVER

As the node publishes, the color of the background for example, it is possible to change the parameters. The command format is

rosparam help

```
tlharmanphd@D125-43873:/$ rosparam help
    rosparam is a command-line tool for getting, setting, and deleting parameters from the ROS
    Parameter Server.
    Commands:
        rosparam set        set parameter
        rosparam get        get parameter
        rosparam load       load parameters from file
        rosparam dump       dump parameters to file
        rosparam delete     delete parameter
        rosparam list       list parameter names
```

```
tlharmanphd@D125-43873:~$ rosparam get /
```

ROS SERVICES

Services: **(We can use ROS services to manipulate the turtle and perform other operations - the format is $rosservice call <service> <arguments>)**

* /turtle1/teleport_absolute
* /turtlesim/get_loggers
* /turtlesim/set_logger_level
* /reset
* /spawn
* /clear
* /turtle1/set_pen
* /turtle1/teleport_relative
* /kill

NODE TOPIC POSE ----TO DO

Another topic for turtlesim node is the turtle’s **pose.** This is the x, y position, angular direction, and the linear and angular velocity. In this example, the turtle is not moving as shown in Figure 5.

To get a list of messages for ROS of **geometry_msgs**
http://wiki.ros.org/geometry_msgs

This displays a verbose list of topics to publish to and subscribe to and their type:

```
tlharmanphd@D125-43873:~$ rostopic list -v
```

```
$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

Where is the turtle?

```
tlharmanphd@D125-43873:~$ rostopic echo /turtle1/pose
```

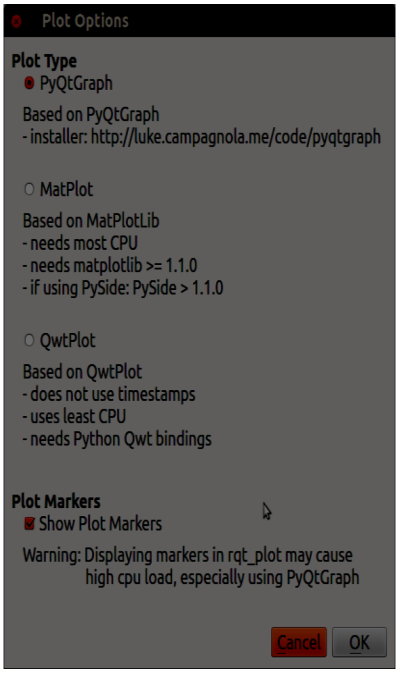
To make the turtle move in a circle

```
harman@Laptop-M1210:~$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

rqt_plot
We can plot information about the nodes and topics.

tlharmanphd@D125-43873:~$ **rqt_plot**

Select plotting type:



Plot Type
● PyQtGraph
Based on PyQtGraph
- installer: http://luke.campagnola.me/code/pyqtgraph

○ MatPlot
Based on MatPlotLib
- needs most CPU
- needs matplotlib >= 1.1.0
- if using PySide: PySide > 1.1.0

○ QwtPlot
Based on QwtPlot
- does not use timestamps
- uses least CPU
- needs Python Qwt bindings

Plot Markers
☒ Show Plot Markers
Warning: Displaying markers in rqt_plot may cause high cpu load, especially using PyQtGraph

Figure 7 Selection of Plotting for rqt_plot

Experiment with different plot types and controls allowed for the plot such as changing the scales, etc.

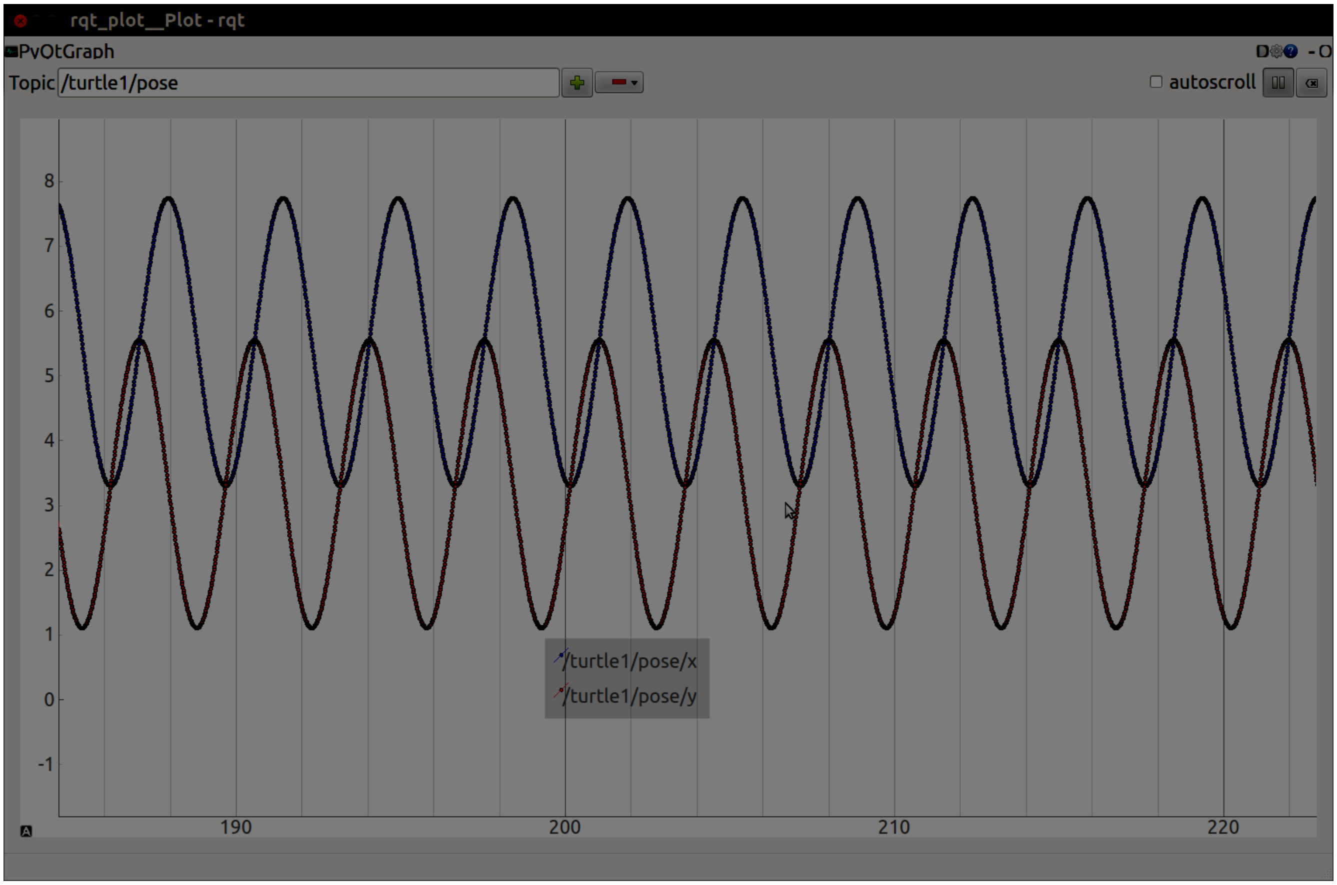


Figure 8 Plot of /turtle1/pose/x and /pose/y

Period of just over 3 seconds for 360 degree rotation. Note the periodic motion in x and y. Right click to change values for axes, etc.

With this plot, right click to set the axes ranges and other aspects of the plot. The pose has five values as shown before, but we have chosen to only plot the x and y variations as the turtle moves in a circle.

Choosing only x and y positions and experimenting with scales and autoscroll. See the tutorial for further help.

http://wiki.ros.org/rqt_plot

To plot from the command line, both of the following lines plot the same topics according to the wiki.

```
$ rqt_plot /turtle1/pose/x:y:z

$ rqt_plot /turtle1/pose/x /turtle1/pose/y /turtle1/pose/z
```

Obviously, if you want to change the topics to plot, you need to restart the program and give the new topic names.

ENABLE KEYBOARD CONTROL

TERMINATOR

Figure 10 Four Turtlesim Windows using Terminator

The screen with four windows was created using Terminator. It is downloaded in Ubuntu from the Software Center Icon on the launcher: http://en.wikipedia.org/wiki/Ubuntu_Software_Center

The terminator is described at this site: https://apps.ubuntu.com/cat/applications/terminator/

1. List the ROS parameters to get information about the ROS nodes. The nodes are generally the executable scripts in ROS.
2. Determine what information you can get for the node turtlesim.

(Publications and Subscriptions)

```
tlharmanphd@D125-43873:~$ rostopic list
        /rosout
        /rosout_agg
        /turtle1/cmd_vel
        /turtle1/color_sensor
        /turtle1/pose
```

One important topic is `/turtle1/cmd_vel` which will be **published** using the keyboard or by publishing the topic with the `rostopic pub` command.

$rostopic pub -r 1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]' for Indigo

The command will publish at a rate (`-r`) of once a second (1 Hz). The topic `/turtle1/command_velocity` is followed by the message type `turtlesim/Velocity` that commands the turtle to turn with linear velocity 2.0 and angular velocity 1.8 according to the ROS tutorial:

http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics

As noted before, a turtlesim/Velocity message has two floating point elements : `linear` and `angular`. In this case, `2.0` becomes the linear value, and `1.8` is the `angular` value. These arguments are actually in YAML syntax, which is described more in the YAML command line documentation.

Clear the screen

When you want to CLEAR THE SCREEN

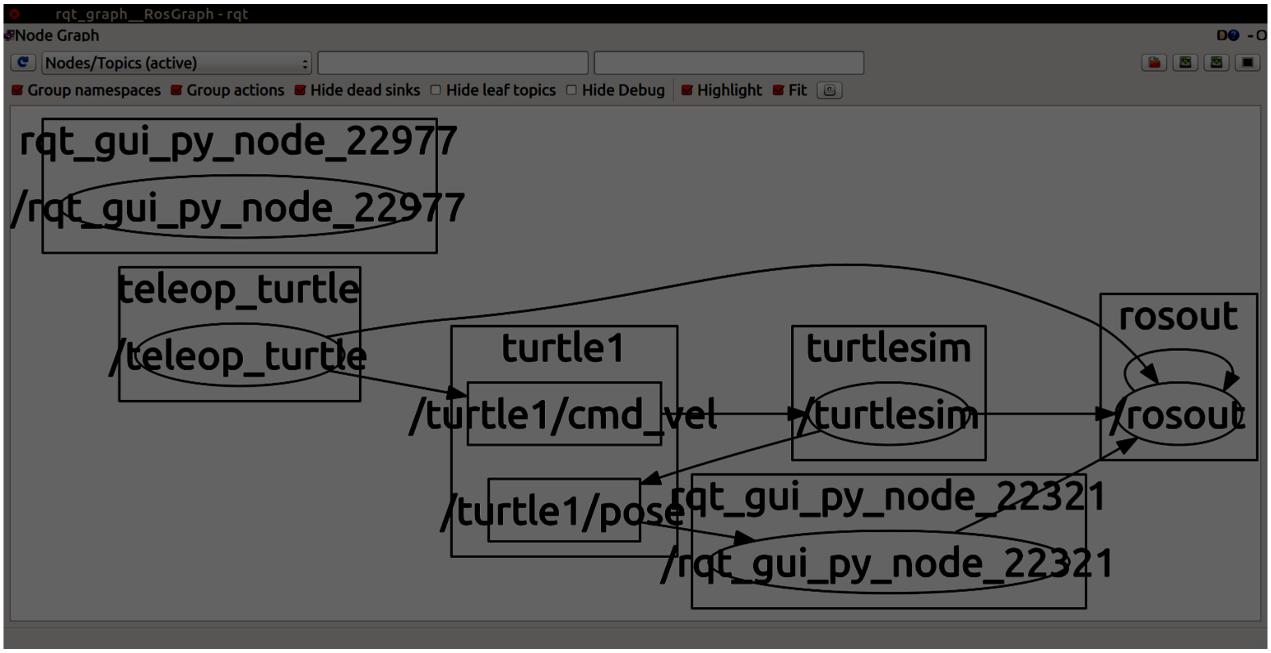
```
tlharmanphd@D125-43873:~$ rosservice call /clear
```

rqt_graph

There is another feature of ROS that is useful for those who wish to see a graphical view of the communication between nodes. We know that `/teleop_turtle` node **publishes** a message on the topic called `/turtle1/command_velocity` and the node `/Turtlesim` **subscribes** to those messages.

This can be shown in a graphical form with the command:

```
tlharmanphd@D125-43873:~$ rqt_graph
```



Nodes/Topics (active)

☑ Group namespaces ☑ Group actions ☑ Hide dead sinks ☐ Hide leaf topics ☐ Hide Debug ☑ Highlight ☑ Fit

Figure 12 Turtlesim graph showing communication

Figure 13 RQT_GRAPH for /turtlesim

ROSBAG

A bag is a file format in ROS for storing ROS message data. The rosbag command can record, replay and manipulate bags.

rosbag help

```
Usage: rosbag <subcommand> [options] [args]
Available subcommands:
   check          Determine whether a bag is playable in the current system, or if it can be migrated.
   compress       Compress one or more bag files.
   decompress  Decompress one or more bag files.
   filter         Filter the contents of the bag.
   fix            Repair the messages in a bag file so that it can be played in the current system.
   help
   info           Summarize the contents of one or more bag files.
   play           Play back the contents of one or more bag files in a time-synchronized fashion.
   record         Record a bag file with the contents of specified topics.
   reindex        Reindexes one or more bag files.
```

We will use the record and play option to learn how to save and replay messages.

For additional information, see http://wiki.ros.org/rosbag

References that describe the rosbag commands in more detail:

http://wiki.ros.org/rosbag/Tutorials/Recording%20and%20playing%20back%20data

http://wiki.ros.org/rosbag/Commandline

Figure 14 Windows for turtlesim

```
tlharmanphd@D125-43873:~$ pwd
        /home/tlharmanphd
tlharmanphd@D125-43873:~$ mkdir bagfilesturtle
tlharmanphd@D125-43873:~$ ls -d b*
        backup  bagfilesturtle  baxter.sh~
```

Here we are making a temporary directory to record data.
Then running rosbag record command with the option –a indicates that all published topics will be accumulated in a bag file.

Start to record the topics with the **rosbag record -a** command:

```
tlharmanphd@D125-43873:~/bagfilesturtle$ rosbag record -a
        [ INFO] [1427220792.012510086]: Recording to 2015-03-24-13-13-12.bag.
        [ INFO] [1427220792.012714289]: Subscribing to /turtle1/color_sensor
        [ INFO] [1427220792.015024218]: Subscribing to /turtle1/cmd_vel
        [ INFO] [1427220792.017232168]: Subscribing to /rosout
        [ INFO] [1427220792.019675036]: Subscribing to /rosout_agg
        [ INFO] [1427220792.021687650]: Subscribing to /turtle1/pose
```

Now change the focus to the teleop_key window move turtle with arrow keys for 10 or so seconds.

In the window running rosbag record, exit with a Ctrl-C when you have finished moving the turtle.
Now examine the contents of the directory **bagfilesturtle**. You should see a file with a name that begins with the year, data, and time and the suffix .bag. This is the bag file that contains all topics published by any node in the time that rosbag record was running.

Now that we've recorded a bag file using rosbag record option we can examine it and play it back using the commands rosbag info and rosbag play. First we are going to see what's recorded in the bag file.

rosbag info

```
tlharmanphd@D125-43873:~/bagfilesturtle$ ls
        2015-03-24-13-13-12.bag
```

Here the name is the date and time.

```
tlharmanphd@D125-43873:~/bagfilesturtle$ rosbag info 2015-03-24-13-13-12.bag
        path:        2015-03-24-13-13-12.bag
        version:     2.0
        duration:    1:22s (82s)
        start:       Mar 24 2015 13:13:12.02 (1427220792.02)
        end:         Mar 24 2015 13:14:34.58 (1427220874.58)
        size:        823.2 KB
        messages:    10736
        compression: none [1/1 chunks]
        types:    geometry_msgs/Twist [9f195f881246fdfa2798d1d3eebca84a]
                  rosgraph_msgs/Log   [acffd30cd6b6de30f120938c17c593fb]
                  turtlesim/Color     [353891e354491c51aabe32df673fb446]
                  turtlesim/Pose      [863b248d5016ca62ea2e895ae5265cf9]
        topics:   /rosout               160 msgs    : rosgraph_msgs/Log   (2 connections)
                  /rosout_agg           156 msgs    : rosgraph_msgs/Log
                  /turtle1/cmd_vel      130 msgs    : geometry_msgs/Twist
                  /turtle1/color_sensor   5145 msgs    : turtlesim/Color
                  /turtle1/pose         5145 msgs    : turtlesim/Pose
```

This tells us topic names and types as well as the number (count) of each message topic contained in the bag file. We can see that of the topics being advertised that we saw in the rostopic output, four of the five were actually published over our recording interval. As we ran rosbag record with the -a flag it recorded all messages published by all nodes.

The next step in this tutorial is to replay the bag file to reproduce behavior in the running system. First kill the teleop program that may be still running from the previous section - Ctrl-c in the terminal where you started turtle_teleop_key.

rosbag play

Leave turtlesim running or restart with a "fresh" turtle.

```
tlharmanphd@D125-43873:~$ rosrun turtlesim turtlesim_node
        [ INFO] [1427221332.211909961]: Starting turtlesim with node name /turtlesim
        [ INFO] [1427221332.225487283]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445],
        theta=[0.000000]
```

In a terminal window run the following command in the directory where you took the original bag file:

```
tlharmanphd@D125-43873:~/bagfilesturtle$ rosbag play 2015-03-24-13-13-12.bag
        [ INFO] [1427221486.993700128]: Opening 2015-03-24-13-13-12.bag

        Waiting 0.2 seconds after advertising topics... done.

        Hit space to toggle paused, or 's' to step.
         [RUNNING]  Bag Time: 1427220874.545656   Duration: 82.521750 / 82.553575
```

```
Done.
```

Turtle begins executing messages from its last location.

In its default mode rosbag play will wait for a certain period (.2 seconds) after advertising each message before it actually begins publishing the contents of the bag file. Waiting for some duration allows any subscriber of a message to be alerted that the message has been advertised and that messages may follow. If rosbag play publishes messages immediately upon advertising, subscribers may not receive the first several published messages. The waiting period can be specified with the -d option.

Eventually the topic /turtle1/command_velocity will be published and the turtle should start moving in turtlesim in a pattern similar to the one you executed from the teleop program. The duration between running rosbag play and the turtle moving should be approximately equal to the time between the original rosbag record execution and issuing the commands from the keyboard in the beginning part of the tutorial. You can have rosbag play not start at the beginning of the bag file but instead start some duration past the beginning using the -s argument. A final option that may be of interest is the -r option, which allows you to change the rate of publishing by a specified factor. If you execute:

```
rosbag play -r 2 <your bagfile>
```

You should see the turtle execute a slightly different trajectory - this is the trajectory that would have resulted had you issued your keyboard commands twice as fast.

After - the motion will start on playback from the current position of the turtle.

```
tlharmanphd@D125-43873:~/bagfilesturtle$ rosbag play -r2 2015-03-24-13-13-12.bag
        [ INFO] [1427221716.127268792]: Opening 2015-03-24-13-13-12.bag

        Waiting 0.2 seconds after advertising topics... done.

        Hit space to toggle paused, or 's' to step.
         [RUNNING]  Bag Time: 1427220874.545836   Duration: 82.521930 / 82.553575
        Done.
```

Recording a subset of the data

When running a complicated system, such as the pr2 software suite, there may be hundreds of topics being published, with some topics, like camera image streams, potentially publishing huge amounts of data. In such a system it is often impractical to write log files consisting of all topics to disk in a single bag file. The rosbag record command supports logging only particular topics to a bag file, allowing a user to only record the topics of interest to them.

To name the bag file and selectively record

(This option is the letter O)

```
tlharmanphd@D125-43873:~/bagfilesturtle$ rosbag record -O cmdvel /turtle1/cmd_vel
/turtle1/pose
        [ INFO] [1427222327.911823890]: Subscribing to /turtle1/cmd_vel
        [ INFO] [1427222327.914523800]: Subscribing to /turtle1/pose
        [ INFO] [1427222327.917503556]: Recording to cmdvel.bag.

tlharmanphd@D125-43873:~/bagfilesturtle$ ls
        2015-03-24-13-13-12.bag  cmdvel.bag
```

Move the turtle with the keys with focus on the teleop window. The -O argument tells rosbag record to log to a file named subset.bag, and the topic arguments cause rosbag record to only subscribe to these two topics. Move the turtle around for several seconds using the keyboard arrow commands, and then Ctrl-c in the rosbag window to stop the rosbag record.

```
tlharmanphd@D125-43873:~/bagfilesturtle$ rosbag info cmdvel.bag
        path:        cmdvel.bag
        version:     2.0
        duration:    1:01s (61s)
        start:       Mar 24 2015 13:38:48.20 (1427222328.20)
        end:         Mar 24 2015 13:39:49.94 (1427222389.94)
        size:        311.4 KB
        messages:    3972
        compression: none [1/1 chunks]
        types:       geometry_msgs/Twist [9f195f881246fdfa2798d1d3eebca84a]
                     turtlesim/Pose      [863b248d5016ca62ea2e895ae5265cf9]
        topics:      /turtle1/cmd_vel    112 msgs    : geometry_msgs/Twist
                     /turtle1/pose      3860 msgs    : turtlesim/Pose
```

```
tlharmanphd@D125-43873:~/bagfilesturtle$ rosbag play cmdvel.bag
        [ INFO] [1427222827.531968073]: Opening cmdvel.bag

        Waiting 0.2 seconds after advertising topics... done.

        Hit space to toggle paused, or 's' to step.
         [RUNNING]  Bag Time: 1427222389.908203   Duration: 61.712115 / 61.743916
        Done.
```

WATCH THE TURTLE MOVE!

The limitations of rosbag record/play

In the previous section you may have noted that the turtle's path may not have exactly mapped to the original keyboard input - the rough shape should have been the same, but the turtle may not have exactly tracked the same path. The reason for this is that the path tracked by turtlesim is very sensitive to small changes in timing in the system, and rosbag is limited in its ability to exactly duplicate the behavior of a running system in terms of when messages are recorded and processed by rosrecord, and when messages are produced and processed when using rosplay. For nodes like turtlesim, where minor timing changes in when command messages are processed can subtly alter behavior, the user should not expect perfectly mimicked behavior.

APPENDIX I REFERENCES

ROS CONCEPTS

ROS has three levels of concepts: the Filesystem level, the Computation Graph level, and the Community level. These levels and concepts are summarized below and later sections go into each of these in greater detail.

The filesystem level concepts mainly cover ROS resources that you encounter on disk, such as packages, metapackages, manifests, repositories, messages, and services

The *Computation Graph* is the peer-to-peer network of ROS processes that are processing data together. The basic Computation Graph concepts of ROS are *nodes*, *Master*, *Parameter Server*, *messages*, *services*, *topics*, and *bags*, all of which provide data to the Graph in different ways.

The ROS Community Level concepts are ROS resources that enable separate communities to exchange software and knowledge. These resources include distributions, repositories, ROS wiki, ROS answers, and a Blog.

In addition to the three levels of concepts, ROS also defines two types of names -- Package Resource Names and Graph Resource Names -- which are discussed below.

http://wiki.ros.org/ROS/Concepts

ROSCORE

From the ROS tutorial http://wiki.ros.org/roscore

roscore is a collection of nodes and programs that are pre-requisites of a ROS-based system. You **must** have a roscore running in order for ROS nodes to communicate. It is launched using the `roscore` command.

ROS MASTER

The ROS Master provides naming and registration services to the rest of the nodes in the ROS system. It tracks publishers and subscribers to topics as well as services. The role of the Master is to enable individual ROS nodes to locate one another. Once these nodes have located each other they communicate with each other peer-to-peer.
http://wiki.ros.org/Master

Clearpath diagram of Master

http://www.clearpathrobotics.com/blog/how-to-guide-ros-101/

ROS NODES AND TURTLESIM

http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes

ROS TOPICS AND TURTLESIM

http://wiki.ros.org/ROS/Tutorials/UnderstandingTopics

ROSSERVICE

rosservice contains the rosservice command-line tool for listing and querying ROS Services
http://wiki.ros.org/rosservice

ROSSERVICE AND ROS SERVICE PARAMETERS

This tutorial introduces ROS services, and parameters as well as using the rosservice and rosparam commandline tools.
http://wiki.ros.org/ROS/Tutorials/UnderstandingServicesParams

http://wiki.ros.org/Parameter%20Server

http://wiki.ros.org/rosparam

http://www.cse.sc.edu/~jokane/agitr/agitr-small-param.pdf (Chapter 7 of O'Kane)

ROSSERVICE AND ROS TELEPORT PARAMETER

Let's bring the turtle to a known starting point using absolute teleportation. Its inputs are [x y theta]. The origin [0 0 0] is offscreen so we will start with [1 1 0]. The turtle should be facing to the right (0*).

```
rosservice call /turtle1/teleport_absolute 1 1 0
```

https://sites.google.com/site/ubrobotics/ros-documentation

USING RQT_PLOT, RQT_CONSOLE AND ROSLAUNCH WITH TURTLESIM

http://wiki.ros.org/rqt_plot

This tutorial introduces ROS using rqt_console and rqt_logger_level for debugging and roslaunch for starting many nodes at once.
http://wiki.ros.org/ROS/Tutorials/UsingRqtconsoleRoslaunch

ROSBAG TURTLESIM EXAMPLE

This tutorial will teach you how to record data from a running ROS system into a .bag file, and then to play back the data to produce similar behavior in a running system.

Keywords: data, rosbag, record, play, info, bag

TURTLESIM EXAMPLE

http://wiki.ros.org/rosbag/Tutorials/Recording%20and%20playing%20back%20data/

DATA LOGGING USING ROSBAG

http://www.fer.unizg.hr/_download/repository/p08-rosbag.pdf

INTRODUCTION TO TF AND TURTLESIM

This tutorial will give you a good idea of what tf can do for you. It shows off some of the tf power in a multi-robot example using turtlesim. This also introduces using tf_echo, view_frames, rqt_tf_tree, and rviz.
http://wiki.ros.org/tf/Tutorials/Introduction%20to%20tf/

YAML Command LINE

Several ROS tools (rostopic, rosservice) use the YAML markup language on the command line. YAML was chosen as, in most cases, it offers a very simple, nearly markup-less solution to typing in typed parameters.

For a quick overview of YAML, please see YAML Overview.

http://wiki.ros.org/ROS/YAMLCommandLine

APPENDIX II MANIFEST (PACKAGE.XML)

```
tlharmanphd@D125-43873:~$ gedit /opt/ros/indigo/share/turtlesim/package.xml
```

APPENDIX III DIRECTORIES AND FILES

```
tlharmanphd@D125-43873:~$ locate turtlesim
```

03/25/15

```
tlharmanphd@D125-43873:~$ cd /opt/ros/indigo/lib/turtlesim
tlharmanphd@D125-43873:/opt/ros/indigo/lib/turtlesim$ ls -la
total 400
drwxr-xr-x   2 root root   4096 Mar 16 20:56 .
drwxr-xr-x 105 root root  20480 Mar 18 15:47 ..
-rwxr-xr-x   1 root root  72248 Feb 20 13:18 draw_square
-rwxr-xr-x   1 root root  59832 Feb 20 13:18 mimic
-rwxr-xr-x   1 root root 220920 Feb 20 13:18 turtlesim_node
-rwxr-xr-x   1 root root  27112 Feb 20 13:18 turtle_teleop_key
tlharmanphd@D125-43873:/opt/ros/indigo/lib/turtlesim$
```

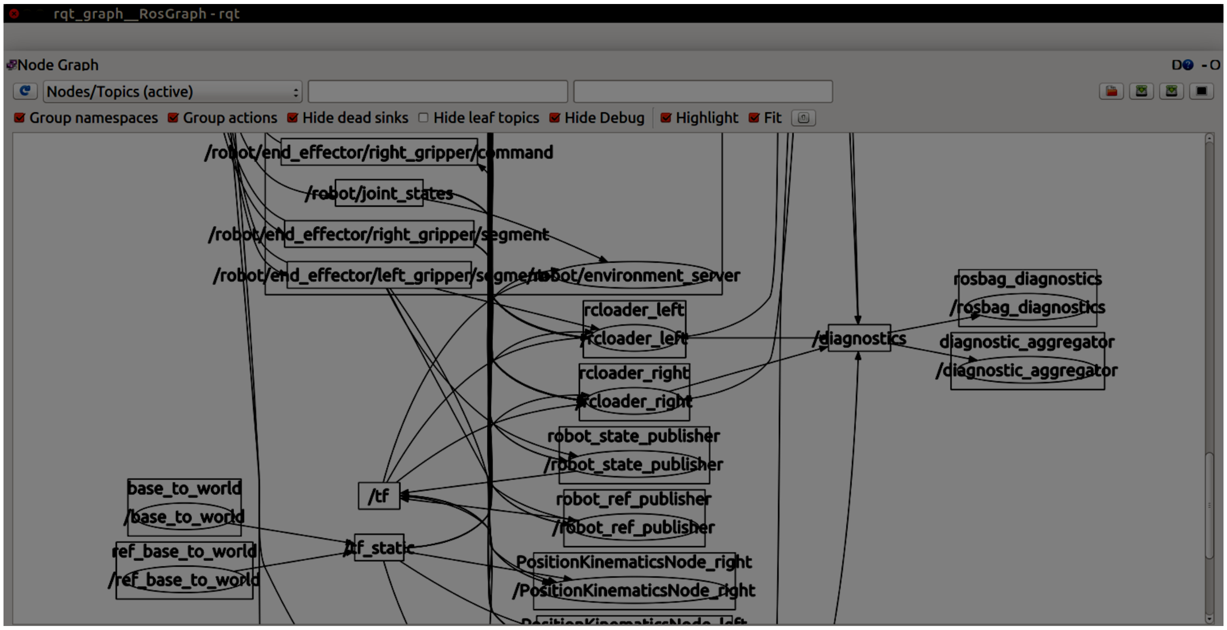
/robot/state/publisher

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rosnode info
/robot_state_publisher
--------------------------------------------------------------------------------
Node [/robot_state_publisher]
Publications:
 * /tf [tf2_msgs/TFMessage]
 * /rosout [rosgraph_msgs/Log]

Subscriptions:
 * /robot/end_effector/left_gripper/segment [MotorControlMsgs/URDFSegment]
 * /robot/end_effector/right_gripper/segment [MotorControlMsgs/URDFSegment]
 * /robot/joint_states [sensor_msgs/JointState]

Services:
 * /robot_state_publisher/get_loggers
 * /robot_state_publisher/set_logger_level


contacting node http://uhclbaxter.local:34950/ ...
Pid: 18583
Connections:
 * topic: /rosout
    * to: /rosout
    * direction: outbound
    * transport: TCPROS
 * topic: /tf
    * to: /robot/environment_server
    * direction: outbound
    * transport: TCPROS
 * topic: /tf
    * to: /PositionKinematicsNode_left
    * direction: outbound
    * transport: TCPROS
 * topic: /tf
    * to: /PositionKinematicsNode_right
    * direction: outbound
    * transport: TCPROS
 * topic: /tf
    * to: /tf2_web_republisher
    * direction: outbound
    * transport: TCPROS
 * topic: /tf
    * to: /rcloader_left
    * direction: outbound
    * transport: TCPROS
 * topic: /tf
    * to: /rcloader_right
    * direction: outbound
    * transport: TCPROS
```


rqt_graph__RosGraph - rqt

Node Graph

Nodes/Topics (active)

☒ Group namespaces ☒ Group actions ☒ Hide dead sinks ☐ Hide leaf topics ☒ Hide Debug ☒ Highlight ☒ Fit

```
 * topic: /robot/joint_states
    * to: /realtime_loop (http://uhclbaxter.local:47064/)
    * direction: inbound
    * transport: TCPROS
 * topic: /robot/end_effector/left_gripper/segment
    * to: /realtime_loop (http://uhclbaxter.local:47064/)
    * direction: inbound
    * transport: TCPROS
 * topic: /robot/end_effector/right_gripper/segment
    * to: /realtime_loop (http://uhclbaxter.local:47064/)
    * direction: inbound
    * transport: TCPROS
```

Clear the screen:

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ clear
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rqt_graph
```

Figure

We will concentrate on the topic /robot/state

Kill A Node

You can close the window with the node /hello defined or kill the node with **rosnode kill <node>** command**.**

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rosnode kill -h
Usage: rosnode kill [node]...

Options:
  -h, --help  show this help message and exit
  -a, --all   kill all nodes
```

To check running process use **$ps –ef** to see all the processes running.

(A very long list.)

Python Script for _AssemblyState.py

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:/opt/ros_ws/install/lib/python2.7/dist-packages/baxter_core_msgs/msg$ gedit _AssemblyState.py
```

```
"""autogenerated by genpy from baxter_core_msgs/AssemblyState.msg. Do not edit."""
import sys
python3 = True if sys.hexversion > 0x03000000 else False
import genpy
import struct


class AssemblyState(genpy.Message):
  _md5sum = "d7ff2b9fa7d5f688665ce44db4ee2af8"
  _type = "baxter_core_msgs/AssemblyState"
  _has_header = False #flag to mark the presence of a Header object
  _full_text = """bool enabled           # true if enabled
bool stopped           # true if stopped -- e-stop asserted
bool error             # true if a component of the assembly has an error
#
# The following are specific to the robot top-level assembly:
uint8  estop_button      # One of the following:
  uint8   ESTOP_BUTTON_UNPRESSED = 0   # Robot is not stopped and button is not pressed
  uint8   ESTOP_BUTTON_PRESSED   = 1
  uint8   ESTOP_BUTTON_UNKNOWN   = 2   # STATE_UNKNOWN when estop was asserted by a non-
user source
  uint8   ESTOP_BUTTON_RELEASED  = 3   # Was pressed, is now known to be released, but robot is still
stopped.
#
uint8  estop_source      # If stopped is true, the source of the e-stop.  One of the following:
  uint8  ESTOP_SOURCE_NONE      = 0   # e-stop is not asserted
  uint8  ESTOP_SOURCE_USER      = 1   # e-stop source is user input (the red button)
  uint8  ESTOP_SOURCE_UNKNOWN   = 2   # e-stop source is unknown
  uint8  ESTOP_SOURCE_FAULT     = 3   # MotorController asserted e-stop in response to a joint fault
  uint8  ESTOP_SOURCE_BRAIN     = 4   # MotorController asserted e-stop in response to a lapse of the
brain heartbeat

"""
  # Pseudo-constants
  ESTOP_BUTTON_UNPRESSED = 0
  ESTOP_BUTTON_PRESSED = 1
  ESTOP_BUTTON_UNKNOWN = 2
  ESTOP_BUTTON_RELEASED = 3
  ESTOP_SOURCE_NONE = 0
  ESTOP_SOURCE_USER = 1
  ESTOP_SOURCE_UNKNOWN = 2
  ESTOP_SOURCE_FAULT = 3
```

```
ESTOP_SOURCE_BRAIN = 4

__slots__ = ['enabled','stopped','error','estop_button','estop_source']
_slot_types = ['bool','bool','bool','uint8','uint8']

def __init__(self, *args, **kwds):
  """
  Constructor. Any message fields that are implicitly/explicitly
  set to None will be assigned a default value. The recommend
  use is keyword arguments as this is more robust to future message
  changes.  You cannot mix in-order arguments and keyword arguments.

  The available fields are:
     enabled,stopped,error,estop_button,estop_source

  :param args: complete set of field values, in .msg order
  :param kwds: use keyword arguments corresponding to message field names
  to set specific fields.
  """
  if args or kwds:
    super(AssemblyState, self).__init__(*args, **kwds)
    #message fields cannot be None, assign default values for those that are
    if self.enabled is None:
      self.enabled = False
    if self.stopped is None:
      self.stopped = False
    if self.error is None:
      self.error = False
    if self.estop_button is None:
      self.estop_button = 0
    if self.estop_source is None:
      self.estop_source = 0
  else:
    self.enabled = False
    self.stopped = False
    self.error = False
    self.estop_button = 0
    self.estop_source = 0

def _get_types(self):
  """
  internal API method
  """
  return self._slot_types

def serialize(self, buff):
  """
```

```
    serialize message into buffer
    :param buff: buffer, ``StringIO``
    """
    try:
      _x = self
      buff.write(_struct_5B.pack(_x.enabled, _x.stopped, _x.error, _x.estop_button, _x.estop_source))
    except struct.error as se: self._check_types(struct.error("%s: '%s' when writing '%s'" % (type(se),
str(se), str(_x))))
    except TypeError as te: self._check_types(ValueError("%s: '%s' when writing '%s'" % (type(te),
str(te), str(_x))))

  def deserialize(self, str):
    """
    unpack serialized message in str into this message instance
    :param str: byte array of serialized message, ``str``
    """
    try:
      end = 0
      _x = self
      start = end
      end += 5
      (_x.enabled, _x.stopped, _x.error, _x.estop_button, _x.estop_source,) =
_struct_5B.unpack(str[start:end])
      self.enabled = bool(self.enabled)
      self.stopped = bool(self.stopped)
      self.error = bool(self.error)
      return self
    except struct.error as e:
      raise genpy.DeserializationError(e) #most likely buffer underfill


  def serialize_numpy(self, buff, numpy):
    """
    serialize message with numpy array types into buffer
    :param buff: buffer, ``StringIO``
    :param numpy: numpy python module
    """
    try:
      _x = self
      buff.write(_struct_5B.pack(_x.enabled, _x.stopped, _x.error, _x.estop_button, _x.estop_source))
    except struct.error as se: self._check_types(struct.error("%s: '%s' when writing '%s'" % (type(se),
str(se), str(_x))))
    except TypeError as te: self._check_types(ValueError("%s: '%s' when writing '%s'" % (type(te),
str(te), str(_x))))

  def deserialize_numpy(self, str, numpy):
    """
    unpack serialized message in str into this message instance using numpy for array types
    :param str: byte array of serialized message, ``str``
    :param numpy: numpy python module
    """
    try:
      end = 0
      _x = self
      start = end
      end += 5
      (_x.enabled, _x.stopped, _x.error, _x.estop_button, _x.estop_source,) =
_struct_5B.unpack(str[start:end])
      self.enabled = bool(self.enabled)
      self.stopped = bool(self.stopped)
      self.error = bool(self.error)
      return self
    except struct.error as e:
      raise genpy.DeserializationError(e) #most likely buffer underfill

_struct_I = genpy.struct_I
_struct_5B = struct.Struct("<5B")
```

ReadMe for Baxter

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter$ ls -la
        total 48
        drwxrwxr-x 4 tlharmanphd tlharmanphd 4096 Mar 16 21:45 .
        drwxrwxr-x 9 tlharmanphd tlharmanphd 4096 Mar 18 15:46 ..
        drwxrwxr-x 2 tlharmanphd tlharmanphd 4096 Mar 16 21:45 baxter_sdk
        -rw-rw-r-- 1 tlharmanphd tlharmanphd  589 Mar 16 21:45 baxter_sdk.rosinstall
        -rwxrwxr-x 1 tlharmanphd tlharmanphd 6122 Mar 16 21:45 baxter.sh
        -rw-rw-r-- 1 tlharmanphd tlharmanphd 1058 Mar 16 21:45 CHANGELOG.rst
        drwxrwxr-x 8 tlharmanphd tlharmanphd 4096 Mar 16 22:11 .git
        -rw-rw-r-- 1 tlharmanphd tlharmanphd  466 Mar 16 21:45 .gitignore
        -rw-rw-r-- 1 tlharmanphd tlharmanphd 1504 Mar 16 21:45 LICENSE
        -rw-rw-r-- 1 tlharmanphd tlharmanphd 2206 Mar 16 21:45 README.rst
        -rw-rw-r-- 1 tlharmanphd tlharmanphd 2838 Mar 16 21:45 RELEASE.rst
```

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws/src/baxter$ gedit
README.rst
```

```
Baxter SDK
==============

The Baxter SDK provides a platform for development of custom applications for the Baxter Research Robot

This repository contains metapackages and files for installation/use of the Baxter SDK

Installation
------------
| Please follow the Getting Started wiki page for instructions on installation of the Baxter SDK:
| http://sdk.rethinkrobotics.com/wiki/Workstation_Setup

Code & Tickets
--------------

+-----------------+-------------------------------------------------------------+
| Documentation   | http://sdk.rethinkrobotics.com/wiki                         |
+-----------------+-------------------------------------------------------------+
| Issues          | https://github.com/RethinkRobotics/baxter/issues            |
+-----------------+-------------------------------------------------------------+
| Contributions   | http://sdk.rethinkrobotics.com/wiki/Contributions           |
+-----------------+-------------------------------------------------------------+

Baxter Repository Overview
-------------------------

::

     .
     |
     +-- baxter_sdk/             baxter metapackage containing all baxter sdk packages
     |
     +-- baxter_sdk.rosinstall   rosinstall script containing all baxter sdk packages
     |
     +-- baxter.sh               convenient environment initialization script
```

Other Baxter Repositories

```
+------------------+---------------------------------------------------+
| baxter_interface | https://github.com/RethinkRobotics/baxter_interface |
+------------------+---------------------------------------------------+
| baxter_tools     | https://github.com/RethinkRobotics/baxter_tools     |
+------------------+---------------------------------------------------+
| baxter_common    | https://github.com/RethinkRobotics/baxter_common    |
+------------------+---------------------------------------------------+
| baxter_examples  | https://github.com/RethinkRobotics/baxter_examples  |
+------------------+---------------------------------------------------+
```

Latest Release Information
----

http://sdk.rethinkrobotics.com/wiki/Release-Changes

BAXTER JOINT STATES

Joints

Baxter has 7 joints (DoF) in each of its two arms and two more joints in its head (side-to-side panning, and binary, up-down nodding). The control for the head is done separately from the arms; however, you can read the current joint states (position, velocity, and effort) for all the joints on both arms and head by subscribing to one topic:

`/robot/joint_states` (`sensor_msgs/JointState`)
where the units for the position of a joint are in (rad), the units of velocity are in (rad/s) and the units of effort in each joint is in (Nm).

The following sections cover the individual joint sensing and control in more detail:

- Arm Joints
- Head Joints

Arm Joints

Component IDs: `left_e0`, `left_e1`, `left_s0`, `left_s1`, `left_w0`, `left_w1`, `left_w2`, `right_e0`, `right_e1`, `right_s0`, `right_s1`, `right_w0`, `right_w1`, `right_w2`

Arm Joint States

`/robot/joint_states` (`sensor_msgs/JointState`)

- `name[i]`: '<component_id>' of `i`-th joint in message value arrays.
- `position[i]`: position of joint `i` rad
- `velocity[i]`: velocity of joint `i` in rad/s
- `effort[i]`: effort applied in joint `i` in Nm

Joint states are published by the `robot_state_publisher` and are updated with information from the sensors for every cycle.
Set Joint State Publishing Rate
`/robot/joint_state_publish_rate` (`std_msgs/UInt16`)

- The rate at which the joints are published can be controlled by publishing a frequency on this topic.
- Default rate is 100Hz; Maximum is 1000Hz.
- *Note: In current version of SDK (v0.6.0), there is a known bug where high rates will cause many topics to lag and become unresponsive.*

Arm Joint Control

There are currently three modes for controlling the arms: Position Mode, Velocity Mode, and Torque Mode.

Alternatively, a joint trajectory action server has been created in support of timestamped joint position trajectories using the ROS standard joint trajectory action.

For more information on joint control, see the following movement example programs.

Joint Control Mode

Each arm can be in independent control modes by publishing the desired control mode to the topic for the appropriate arm.

Position Mode

The first is Position Mode, in which users publish target joint angles for given joints and the internal controller drives the arm to the published angles.

Velocity Mode

Warning: *Advanced Control Mode.*
In Velocity Control Mode, users publish velocities for given joints and the joints will move at the specified velocity.

Torque Mode

Warning: *Advanced Control Mode.*
USE WITH EXTREME CAUTION: THIS CONTROL MODE BYPASSES COLLISION AVOIDANCE AND CAN RESULT IN POTENTIALLY HARMFUL MOTIONS
In Torque Control Mode, users publish torques for given joints and the joints will move at the specified torque.

Switching Modes
`/robot/limb/<side>/joint_command` (`baxter_core_msgs/JointCommand`)

- Mode is set implicitly by specifying the mode in the command message. Publish a `JointCommand` message to the `joint_command` topic for a given arm to set the arm into the desired control mode.
- Constants for each mode are defined in the `JointCommand` message type.

Joint Trajectory Action

The Joint Trajectory Action provides an ROS action interface for tracking trajectory execution.

The joint trajectory action server provides an action interface for execution of trajectories requested by the client, known as a *Goal* request. The joint trajectory action server then executes the request trajectory communicating the *Result* response. The actionlib (action server/client interface) package differs from ROS services, simple request/response interface, in that actions allow mid-execution cancellation, they can also provide feedback during execution as to the progress of the Goal request.

Joint Trajectory Action Server
`/robot/limb/<limb>/follow_joint_trajectory/cancel` `/robot/limb/<limb>/follow_joint_trajectory/feedback` `/robot/limb/<limb>/follow_joint_trajectory/goal` `/robot/limb/<limb>/follow_joint_trajectory/result` `/robot/limb/<limb>/follow_joint_trajectory/status`

Usage:

```
    # Verify that the robot is enabled:

    $ rosrun baxter_tools enable_robot.py

    # Start the joint trajectory action server:

    $ rosrun baxter_interface joint_trajectory_action_server.py
```

Please see the simple joint trajectory example or joint trajectory playback example for examples of creating a client of this action server and requesting a joint trajectory.

Parameters:

The joint trajectory action server provides a number of parameters which describe it's behavior during the trajectory execution. These were largely designed to follow these standards.

Note: All of these parameters will be initialized on startup of the trajectory_controller.py if they were not previously specified. `/rethink_rsdk_joint_trajectory_controller/<joint_name>__kp` The proportional gain with which the joint trajectory controller will track the commanded trajectory for the specified joint.
`/rethink_rsdk_joint_trajectory_controller/<joint_name>__kd` The derivative gain with which the joint trajectory controller will track the commanded trajectory for the specified joint.
`/rethink_rsdk_joint_trajectory_controller/<joint_name>__ki` The integral gain with which the joint trajectory controller will track the commanded trajectory for the specified joint.
`/rethink_rsdk_joint_trajectory_controller/goal_time (double, default: 0.0)` The amount of time (in seconds) that the controller is permitted to be late to the goal. If goal_time has passed and the controller still has not reached the final position (within the parameters described by `/rethink_rsdk_joint_trajectory_controller/<joint_name>_goal`, then the goal is aborted.
`/rethink_rsdk_joint_trajectory_controller/<joint>_goal (double, default: -1.0)` The maximum final error for for the trajectory goal to be considered successful. Negative numbers indicate that there is no constraint. Given in units of joint position (radians). If this constraint is violated, the goal is aborted.
`/rethink_rsdk_joint_trajectory_controller/trajectory (double, default: -1.0)` The maximum error for at any point during execution for the trajectory goal to be considered successful. Negative numbers indicate that there is no constraint. Given in units of joint position (radians). If this constraint is violated, the goal is aborted.
Dynamic Reconfigure GUI is suggest for use with ROS Distributions >=Groovy for setting these parameters.

```
    # Start the dynamic reconfigure GUI:

    $ rosrun rqt_reconfigure rqt_reconfigure
```

Expand the joint trajectory controller's parameters by choosing `rethink_rsdk_joint_trajectory_controller` from the left menu. Use the sliders/input fields to specify these parameters dynamically.
Alternatively, these parameters can be set via a YAML file, command line, or programmatically (rospy, roscpp).

sensor_msgs/JointState Message

File: sensor_msgs/JointState.msg

Raw Message Definition

```
# This is a message that holds data to describe the state of a set of torque controlled joints.
#
# The state of each joint (revolute or prismatic) is defined by:
#  * the position of the joint (rad or m),
#  * the velocity of the joint (rad/s or m/s) and
#  * the effort that is applied in the joint (Nm or N).
#
# Each joint is uniquely identified by its name
# The header specifies the time at which the joint states were recorded. All the joint states
# in one message have to be recorded at the same time.
#
# This message consists of a multiple arrays, one for each part of the joint state.
# The goal is to make each of the fields optional. When e.g. your joints have no
# effort associated with them, you can leave the effort array empty.
#
# All arrays in this message should have the same size, or be empty.
# This is the only way to uniquely associate the joint name with the correct
# states.


Header header

string[] name
float64[] position
float64[] velocity
float64[] effort
```

Compact Message Definition

std_msgs/Header header
string[] name
float64[] position
float64[] velocity
float64[] effort

autogenerated on Fri, 03 Apr 2015 13:14:30

Cartesian (Endpoint State)

Published at 100 Hz, the endpoint state topic provides the current Cartesian Position, Velocity and Effort at the endpoint for either limb.

Endpoint State

Endpoint State

`/robot/limb/<side>/endpoint_state` (`baxter_core_msgs/EndpointState`)

- The endpoint state message provides the current `position/orientation pose`, `linear/angular velocity`, and `force/torque effort` of the robot end-effector at 100 Hz. Pose is in Meters, Velocity in m/s, Effort in Nm.

---

5 lines (4 sloc) 0.095 kb

```
Header header

geometry_msgs/Pose   pose
geometry_msgs/Twist  twist
geometry_msgs/Wrench wrench
```

baxter_core_msgs/EndpointState Message

File: `baxter_core_msgs/EndpointState.msg`

Raw Message Definition

```
Header header
geometry_msgs/Pose    pose
geometry_msgs/Twist   twist
geometry_msgs/Wrench wrench
```

Compact Message Definition

std_msgs/Header header
geometry_msgs/Pose pose
geometry_msgs/Twist twist
geometry_msgs/Wrench wrench

autogenerated on Thu, 08 Jan 2015 14:11:37

Hi Chris,

Internal to the robot we use KDL as our forward kinematics solver. KDL also uses Baxter's dynamic model to produce our gravity compensation torques for the limbs. These are simply loaded from our URDF available here.

We then publish this endpoint pose (the FK solution), as well as the velocity differentiated from that position, and effort found from the jacobian and measured joint torques on the topic: `/robot/limb/<side>/endpoint_state` of message type baxter_core_msgs/EndpointState.

I did some work recently on recreating the FK/IK KDL solutions in python using PyKDL and the new urdf_parser_py in the urdfdom package, but accidentally deleted :(.. I have been meaning to recreate this work, so I will do it this week and share.

Maybe others can chime in/share their FK/ID solvers they've created. Do you have any other requirements for the dynamics model that are not encapsulated in the urdf?

Hope this helps.

Regards,
Kyle

REALTIME_LOOP   04/04/15

```
[baxter - http://172.29.64.200:11311] tlharmanphd@D125-43873:~/ros_ws$ rosnode info /realtime_loop | more
--------------------------------------------------------------------------------
Node [/realtime_loop]
Publications:
 * /robot/itb/right_itb/state [baxter_core_msgs/ITBState]
 * /robot/limb/left/stiffness [MotorControlMsgs/Stiffness]
 * /robot/analog_io/left_hand_range/value_uint32 [std_msgs/UInt32]
 * /robot/digital_io/torso_brake/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/left_vacuum_sensor_analog/value_uint32 [std_msgs/UInt32]
 * /robot/limb/left/state [baxter_core_msgs/AssemblyState]
 * /robot/digital_io/head_lcd_auto_config/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_itb_button3/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/right/endpoint_state [baxter_core_msgs/EndpointState]
 * /robot/digital_io/torso_right_itb_button2/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/left/limb_state [MotorControlMsgs/LimbState]
 * /robot/analog_io/left_vacuum_sensor_analog/state [baxter_core_msgs/AnalogIOState]
 * /robot/digital_io/left_itb_button_right/state [baxter_core_msgs/DigitalIOState]
 * /robot/itb/left_itb/state [baxter_core_msgs/ITBState]
 * /robot/sonar/head_sonar/lights/green_level [std_msgs/Float32]
 * /robot/digital_io/torso_digital_input0/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_camera_power/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io_states [baxter_core_msgs/AnalogIOStates]
 * /robot/analog_io/right_itb_wheel/value_uint32 [std_msgs/UInt32]
 * /robot/limb/left/gravity_compensation_torques [baxter_core_msgs/SEAJointState]
 * /robot/digital_io/torso_left_itb_button_left/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_itb_button1/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_ui_output2/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_brake_sensor/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/left_suck/state [baxter_core_msgs/DigitalIOState]
 * /robot/end_effector/right_gripper/properties [baxter_core_msgs/EndEffectorProperties]
 * /robot/digital_io/torso_right_itb_light_inner/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/left_itb_wheel/value_uint32 [std_msgs/UInt32]
 * /robot/limb/right/limb_state [MotorControlMsgs/LimbState]
 * /robot/digital_io/torso_ui_output0/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb_names [MotorControlMsgs/StringArray]
 * /robot/accelerometer_names [MotorControlMsgs/StringArray]
 * /robot/limb/left/velocity_controller_state [MotorControlMsgs/VelocityControllerState]
 * /robot/digital_io/left_itb_button3/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_itb_button2/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/right/stiffness_constraint [MotorControlMsgs/StiffnessConstraint]
 * /robot/analog_io/torso_lighting/state [baxter_core_msgs/AnalogIOState]
 * /robot/assembly_names [MotorControlMsgs/StringArray]
 * /robot/digital_io/torso_right_itb_button_down/state [baxter_core_msgs/DigitalIOState]
 * /robot/assembly/head/state [baxter_core_msgs/AssemblyState]
 * /robot/sonar/head_sonar/sonars_enabled [std_msgs/UInt16]
 * /robot/digital_io/right_suck/state [baxter_core_msgs/DigitalIOState]
 * /robot/end_effector/left_gripper/state [baxter_core_msgs/EndEffectorState]
 * /robot/digital_io/left_shoulder_button/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/right/velocity_controller_state [MotorControlMsgs/VelocityControllerState]
 * /robot/digital_io/right_itb_button0/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/right/stiffness [MotorControlMsgs/Stiffness]
 * /robot/digital_io/right_lower_button/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/torso_right_itb_wheel/value_uint32 [std_msgs/UInt32]
 * /robot/digital_io/left_itb_light_inner/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/torso_left_itb_wheel/state [baxter_core_msgs/AnalogIOState]
 * /robot/accelerometer_states [MotorControlMsgs/AccelerometerStates]
 * /robot/digital_io/left_itb_button1/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_right_itb_button1/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_itb_button_down/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_right_itb_button_up/state [baxter_core_msgs/DigitalIOState]
 * /robot/sonar/head_sonar/state [sensor_msgs/PointCloud]
 * /robot/digital_io/motor_fault_signal/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_lower_cuff/state [baxter_core_msgs/DigitalIOState]
 * /robot/accelerometer/right_accelerometer/state [sensor_msgs/Imu]
 * /robot/digital_io/right_itb_button_left/state [baxter_core_msgs/DigitalIOState]
 * /robot/assembly/left/state [baxter_core_msgs/AssemblyState]
 * /robot/limb/left/twist_speed_constraint [MotorControlMsgs/SpeedConstraint]
 * /robot/digital_io/torso_right_itb_button_right/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/limit_switch_2/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_right_itb_button_left/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_left_itb_button1/state [baxter_core_msgs/DigitalIOState]
 * /robot/range_names [MotorControlMsgs/StringArray]
 * /robot/digital_io/right_hand_camera_power/state [baxter_core_msgs/DigitalIOState]
 * /robot/ref_joint_states [sensor_msgs/JointState]
 * /robot/digital_io/right_itb_button_up/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/right/gravity_compensation_torques [baxter_core_msgs/SEAJointState]
 * /robot/assembly/right/state [baxter_core_msgs/AssemblyState]
 * /robot/digital_io/torso_left_itb_light_inner/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_safety_stop/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/left_hand_camera_power/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_left_itb_button2/state [baxter_core_msgs/DigitalIOState]
 * /robot/state [baxter_core_msgs/AssemblyState]
 * /robot/analog_io/right_hand_range/value_uint32 [std_msgs/UInt32]
 * /robot/head/state [baxter_core_msgs/AssemblyState]
 * /robot/end_effector/right_gripper/state [baxter_core_msgs/EndEffectorState]
 * /robot/analog_io/left_hand_range/state [baxter_core_msgs/AnalogIOState]
 * /robot/itb/torso_left_itb/state [baxter_core_msgs/ITBState]
 * /robot/digital_io/torso_left_itb_button_up/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_right_itb_button3/state [baxter_core_msgs/DigitalIOState]
 * /robot/end_effector_names [MotorControlMsgs/StringArray]
 * /robot/limb/left/stiffness_constraint [MotorControlMsgs/StiffnessConstraint]
 * /robot/limb/right/collision_avoidance_state [baxter_core_msgs/CollisionAvoidanceState]
 * /robot/analog_io/right_itb_wheel/state [baxter_core_msgs/AnalogIOState]
 * /robot/digital_io/left_itb_button_down/state [baxter_core_msgs/DigitalIOState]
 * /robot/accelerometer/left_accelerometer/state [sensor_msgs/Imu]
 * /robot/digital_io/torso_ui_output3/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/left_itb_button_up/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/right/state [baxter_core_msgs/AssemblyState]
 * /robot/limb/left/collision_detection_state [baxter_core_msgs/CollisionDetectionState]
 * /diagnostics [diagnostic_msgs/DiagnosticArray]
 * /robot/end_effector/right_gripper/segment [MotorControlMsgs/URDFSegment]
 * /robot/sonar/head_sonar/lights/red_level [std_msgs/Float32]
 * /robot/digital_io/left_blow/state [baxter_core_msgs/DigitalIOState]
 * /robot/range/left_hand_range/state [sensor_msgs/Range]
 * /robot/digital_io/left_itb_button_left/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_left_itb_light_outer/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_pneumatic/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/left/collision_avoidance_state [baxter_core_msgs/CollisionAvoidanceState]
 * /robot/assembly/torso/state [baxter_core_msgs/AssemblyState]
 * /robot/digital_io/right_blue_light/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_shoulder_button/state [baxter_core_msgs/DigitalIOState]
 * /robot/range/right_hand_range/state [sensor_msgs/Range]
 * /robot/sonar/head_sonar/lights/state [std_msgs/UInt16]
 * /robot/itb_states [baxter_core_msgs/ITBStates]
 * /robot/sonar_names [MotorControlMsgs/StringArray]
 * /robot/analog_io/torso_lighting/value_uint32 [std_msgs/UInt32]
 * /robot/joint_names [MotorControlMsgs/StringArray]
 * /robot/itb_names [MotorControlMsgs/StringArray]
 * /robot/digital_io/torso_left_itb_button_down/state [baxter_core_msgs/DigitalIOState]
 * /robot/itb/torso_right_itb/state [baxter_core_msgs/ITBState]
 * /robot/digital_io/right_itb_button_right/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_blow/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_process_sense0/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/left_pneumatic/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/right_vacuum_sensor_analog/value_uint32 [std_msgs/UInt32]
 * /robot/digital_io/right_itb_light_outer/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/right_hand_range/state [baxter_core_msgs/AnalogIOState]
 * /robot/digital_io/left_itb_button2/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_right_itb_light_outer/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/right_itb_light_inner/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/right/commanded_endpoint_state [baxter_core_msgs/EndpointState]
 * /robot/limb/right/twist_speed_constraint [MotorControlMsgs/SpeedConstraint]
 * /robot/digital_io_states [baxter_core_msgs/DigitalIOStates]
 * /robot/digital_io/head_red_light/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_process_sense1/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_left_itb_button_right/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io_names [MotorControlMsgs/StringArray]
 * /robot/end_effector/left_gripper/properties [baxter_core_msgs/EndEffectorProperties]
 * /robot/limb/left/endpoint_state [baxter_core_msgs/EndpointState]
 * /robot/digital_io/torso_foot_pedal/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/limit_switch_3/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_ui_output1/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/head_green_light/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/right/collision_detection_state [baxter_core_msgs/CollisionDetectionState]
 * /robot/analog_io/right_vacuum_sensor_analog/state [baxter_core_msgs/AnalogIOState]
```

```
 * /robot/analog_io/torso_left_itb_wheel/value_uint32 [std_msgs/UInt32]
 * /robot/digital_io/right_upper_button/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_left_itb_button0/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/left_itb_button0/state [baxter_core_msgs/DigitalIOState]
 * /rosout [rosgraph_msgs/Log]
 * /robot/digital_io/left_lower_button/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/left_upper_button/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/head_yellow_light/state [baxter_core_msgs/DigitalIOState]
 * /robot/end_effector/left_gripper/segment [MotorControlMsgs/URDFSegment]
 * /robot/digital_io/left_lower_cuff/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/left_itb_wheel/state [baxter_core_msgs/AnalogIOState]
 * /robot/joint_states [sensor_msgs/JointState]
 * /robot/sonar_states [MotorControlMsgs/SonarStates]
 * /robot/head/head_state [baxter_core_msgs/HeadState]
 * /robot/analog_io/torso_fan/state [baxter_core_msgs/AnalogIOState]
 * /robot/ref_joint_names [MotorControlMsgs/StringArray]
 * /robot/analog_io_names [MotorControlMsgs/StringArray]
 * /robot/analog_io/torso_fan/value_uint32 [std_msgs/UInt32]
 * /robot/digital_io/left_itb_light_outer/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/limit_switch_1/state [baxter_core_msgs/DigitalIOState]
 * /robot/limb/left/commanded_endpoint_state [baxter_core_msgs/EndpointState]
 * /robot/digital_io/torso_left_itb_button3/state [baxter_core_msgs/DigitalIOState]
 * /robot/digital_io/torso_right_itb_button0/state [baxter_core_msgs/DigitalIOState]
 * /robot/analog_io/torso_right_itb_wheel/state [baxter_core_msgs/AnalogIOState]

Subscriptions:
 * /robot/end_effector/right_gripper/rsdk/set_state [unknown type]
 * /robot/end_effector/left_gripper/segment [MotorControlMsgs/URDFSegment]
 * /robot/end_effector/right_gripper/command [baxter_core_msgs/EndEffectorCommand]
 * /robot/limb/right/set_feed_forward_weights [unknown type]
 * /collision/right/collision_detection [MotorControlMsgs/CollisionDetection]
 * /robot/sonar/head_sonar/lights/set_lights [unknown type]
 * /robot/limb/left/command_twist_speed_limit [unknown type]
 * /robot/limb/right/suppress_contact_safety [std_msgs/Empty]
 * /robot/sonar/head_sonar/lights/set_red_level [unknown type]
 * /robot/limb/right/command_stiffness [unknown type]
 * /robot/limb/left/suppress_cuff_interaction [unknown type]
 * /robot/head/command_head_nod [std_msgs/Bool]
 * /robot/end_effector/left_gripper/rsdk/set_properties [unknown type]
 * /robot/sonar/head_sonar/set_sonars_enabled [unknown type]
 * /collision/left/collision_detection [MotorControlMsgs/CollisionDetection]
 * /robot/end_effector/left_gripper/command [baxter_core_msgs/EndEffectorCommand]
 * /robot/assembly/right/set_components_enable [unknown type]
 * /robot/limb/left/suppress_contact_safety [std_msgs/Empty]
 * /robot/limb/left/use_default_spring_model [std_msgs/Empty]
 * /robot/limb/right/use_default_spring_model [std_msgs/Empty]
 * /robot/limb/left/command_stiffness [unknown type]
 * /robot/limb/left/set_damping_correction_weights [unknown type]
 * /robot/limb/left/set_dominance [unknown type]
 * /robot/limb/left/command_twist_stamped [unknown type]
* /robot/limb/left/set_feed_forward_weights [unknown type]
* /robot/limb/right/command_joint_position [unknown type]
* /robot/joint_state_publish_rate [unknown type]
* /robot/end_effector/left_gripper/rsdk/set_state [unknown type]
* /robot/limb/left/command_twist_speed_limit_scale [unknown type]
* /robot/assembly/torso/set_components_enable [unknown type]
* /robot/limb/right/command_twist_speed_limit [unknown type]
* /robot/limb/right/inverse_dynamics_command [unknown type]
* /robot/head/command_head_pan [baxter_core_msgs/HeadPanCommand]
* /robot/sonar/head_sonar/approach_alert [unknown type]
* /robot/limb/left/command_velocity_tozero [unknown type]
* /robot/limb/left/set_speed_ratio [unknown type]
* /robot/digital_io/command [baxter_core_msgs/DigitalOutputCommand]
* /robot/limb/right/command_stiffness_limit [unknown type]
* /robot/assembly/head/set_components_enable [unknown type]
* /robot/limb/right/joint_command [unknown type]
* /robot/analog_io/command [unknown type]
* /robot/end_effector/right_gripper/segment [MotorControlMsgs/URDFSegment]
* /robot/end_effector/left_gripper/set_object_mass [unknown type]
* /robot/limb/left/suppress_collision_avoidance [std_msgs/Empty]
* /robot/limb/left/inverse_dynamics_command [unknown type]
* /robot/limb/right/command_nullspace_setpoint_and_twist_stamped [unknown type]
* /robot/end_effector/right_gripper/set_object_mass [unknown type]
* /robot/limb/right/suppress_collision_avoidance [std_msgs/Empty]
* /robot/limb/right/command_twist_stamped [unknown type]
* /robot/set_super_reset [unknown type]
* /robot/limb/right/weight_integral_terms [unknown type]
* /robot/limb/left/command_nullspace_setpoint_and_twist_stamped [unknown type]
* /robot/end_effector/right_gripper/rsdk/set_properties [unknown type]
* /robot/limb/right/set_damping_correction_weights [unknown type]
* /robot/limb/right/set_dominance [unknown type]
* /robot/limb/left/weight_integral_terms [unknown type]
* /robot/limb/right/joint_command_timeout [unknown type]
* /robot/limb/left/command_stiffness_limit [unknown type]
* /robot/limb/right/command_velocity_tozero [unknown type]
* /robot/limb/right/suppress_cuff_interaction [unknown type]
* /robot/end_effector/right_gripper/configure [unknown type]
* /robot/limb/left/joint_command_timeout [unknown type]
* /robot/limb/right/set_speed_ratio [unknown type]
* /robot/limb/right/command_twist_speed_limit_scale [unknown type]
* /robot/end_effector/left_gripper/configure [unknown type]
* /robot/assembly/left/set_components_enable [unknown type]
* /robot/set_super_stop [unknown type]
* /robot/limb/left/suppress_gravity_compensation [unknown type]
* /robot/limb/left/joint_command [unknown type]
* /robot/set_super_enable [unknown type]
* /robot/sonar/head_sonar/lights/set_green_level [unknown type]
* /robot/limb/left/command_joint_position [unknown type]
* /robot/limb/right/suppress_gravity_compensation [unknown type]
```

```
Services:
 * /load_controller
 * /reload_controller_libraries
 * /switch_controllers
 * /robot/limb/left/set_gc_enable
 * /list_controllers
 * /robot/limb/right/set_gc_enable
 * /robot/limb/left/set_gc_tare
 * /robot/limb/right/set_gc_tare
 * /realtime_loop/set_logger_level
 * /realtime_loop/get_loggers
 * /list_controller_types
 * /unload_controller


contacting node http://uhclbaxter.local:41055/ ...
Pid: 18579
Connections:
 * topic: /rosout
    * to: /rosout
    * direction: outbound
    * transport: TCPROS
 * topic: /diagnostics
    * to: /diagnostic_aggregator
    * direction: outbound
    * transport: TCPROS
 * topic: /diagnostics
    * to: /rosbag_diagnostics
    * direction: outbound
    * transport: TCPROS
 * topic: /robot/state
    * to: /rostopic_16006_1428173899591
    * direction: outbound
    * transport: TCPROS
 * topic: /robot/limb_names
    * to: /rcloader_right
--More--
```