Multi-Robot Control (MRC)

W2: ROS Topics, Services, roslaunch and rosbag
MATLAB and ROS
Plan: ROS Tutorials

Tutorial Topics

- Environment and filesystem
- Packages
- Nodes
- Topics
- Services and Parameters
HW2 Overview

1. Make a ROS Package
2. Use rospack utility
3. Driving Turtlesim
   • One letter
   • Two letters
ROS Packages

A Package

- Is how ROS software is organized and distributed
- Contains things like nodes, messages, libraries, configuration, launch files, etc.
- Is built using catkin

1. Common Files and Directories

ROS packages tend to follow a common structure. Here are some of the directories and files you may notice.

- include/package_name: C++ include headers (make sure to export in the CMakeLists.txt)
- msg/: Folder containing Message (msg) types
- src/package_name/: Source files, especially Python source that are exported to other packages.
- srv/: Folder containing Service (srv) types
- scripts/: executable scripts
- CMakeLists.txt: CMake build file (see catkin/CMakeLists.txt)
- package.xml: Package catkin/package.xml
- CHANGES.rst: Many packages will define a changelog which can be automatically injected into binary packaging and into the wiki page for the package

http://wiki.ros.org/Packages
ROS Package Locations
A topic is a **name** for a stream of messages with a defined **type**

```bash
bsb@aku:~$ rostopic list
/rostopic_30932_1554745975639
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
/turtle2/cmd_vel
/turtle2/color_sensor
/turtle2/pose
bsb@aku:~$  
```
ROS Topics

“A topic is a **name** for a stream of messages with a defined **type**”
“A topic is a name for a stream of messages with a defined type”
bsb@aku:~$ rosnode list
/rosout
/rqt_gui_py_node_10197
/turtlesim

rosrun rqt_graph rqt_graph

/turtlesim
bsb@aku:~$ rosnode info /turtlesim

Node [/turtlesim]

Publications:
* /rosout [rosgraph_msgs/Log]
* /turtle1/color_sensor [turtlesim/Color]
* /turtle1/pose [turtlesim/Pose]

Subscriptions:
* /turtle1/cmd_vel [unknown type]

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

contacting node http://aku:38857/ ...
Pid: 10088
Connections:
* topic: /rosout
  * to: /rosout
  * direction: outbound
* transport: TCPROS
Driving Turtle Using Topics

How to generate

• Msg type: Twist
• Topic: /turtle1/cmd_vel

1. Keyboard teleop

```
bsb@aku:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
Use arrow keys to move the turtle.
```

2. rostopic command

• From command line
• As a Bash script

3. MATLAB/Simulink
Driving Turtle Using Topics

rostopic

`bsb@aku:~$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[5.0, 0.0, 0.0]' '[0.0, 0.0, -4.5]'`

`/rostopic_12324_1554759129752
/turtle1/cmd_vel
/turtlesim`

`bsb@aku:~/catkin_ws/src/mrc_hw2/scripts$ bash turtleletter.sh`

```
rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[4.0, 0.0, 0.0]' '[0.0, 0.0, 0.0]'
rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[5.0, 0.0, 0.0]' '[0.0, 0.0, -4.5]'
```
ROS Services

Topics:
- Publish/Subscribe,
- Many-to-many
- One-way

Services:
- Request/Reply, e.g.,
  - Get/set interactions
  - Queries
  - Remote procedure call
- Client ↔ Server
- Service =
  - pair of messages
    (request/reply)
  - service name

Examples:
- Planner: find path from A-B
- Spawn a simulated robot
- Set camera calibration
- Request camera image
- IMU Self test
- IMU calibrate
- UAV, set mission

http://wiki.ros.org/Services
2.1 turtlesim_node

turtlesim_node provides a simple simulator for teaching ROS concepts.

2.1.1 Subscribed Topics

turtleX/cmd_vel (geometry_msgs/Twist)
   The linear and angular command velocity for turtleX. The turtle will execute a velocity command for 1 second then time out.

2.1.2 Published Topics

turtleX/pose (turtlesim/Pose)
   The x, y, theta, linear velocity, and angular velocity of turtleX.

2.1.3 Services

clear (std_srvs/Empty)
   Clears the turtlesim background and sets the color to the value of the background parameters.

reset (std_srvs/Empty)
   Resets the turtlesim to the start configuration and sets the background color to the value of the background.

kill (turtlesim/Kill)
   Kills a turtle by name.

spawn (turtlesim/Spawn)
   Spawns a turtle at (x, y, theta) and returns the name of the turtle. Also will take name for argument but will fail if a duplicate name.

turtleX/set_pen (turtlesim/SetPen)
   Sets the pen's color (r g b), width (width), and turns the pen on and off (off).

turtleX/teleport_absolute (turtlesim/TeleportAbsolute)
   Teleports the turtleX to (x, y, theta).

turtleX/teleport_relative (turtlesim/TeleportRelative)
   Teleports the turtleX a linear and angular distance from the turtles current position.
2.1 turtlesim_node

TurtleX. The turtle will execute a velocity command for 1 second then

Duplicate name.

turtleX/set_pen (turtlesim)
Sets the pen's color (r g b) and thickness.

turtleX/teleport_absolute
Teleports the turtleX to (x y)

turtleX/teleport_relative
Teleports the turtleX a linear and angular distance from the turtles current position.

bsb@aku:/catkin_ws/src/mrc_hw3/launch$ rosservice list
/mclear
/keyboard_control/get_loggers
/keyboard_control/set_logger_level
/kill
/reset
/rosout/get_loggers
/rosout/set_logger_level
/simulated_turtle/get_loggers
/simulated_turtle/set_logger_level
/spawn
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
2.1 turtlesim_node

turtlesim_node provides a simple simulator for teaching ROS concepts.

2.1.1 Subscribed Topics

$ rosservice call /turtle1/teleport_absolute 1 1 1.57

Node: /turtle1
URI: rosrpc://aku:39165
Type: turtlesim/TeleportAbsolute
Args: x y theta

- The color to the value of the background parameters.

- Sets the background color to the value of the background.

- The name of the turtle. Also will take name for argument but will fail if a

- The turtleX_teleport_relative (turtlesim/TeleportRelative)

- Teleports the turtleX a linear and angular distance from the turtles current position.
HW2 Overview

1. Make a ROS Package
2. Use rospack utility
3. Driving Turtlesim
   • One letter
   • Two letters