AI & Robotics

ROS Concepts

AI & ROBOTICS LAB

Goals

The **junior-colleague**

- can describe in own words roscore, master, parameter server and rosout in own words
- can explain in own words nodes, topics, pub/sub, service/client packages in context of ROS
- can start, list, inspect and ping existing nodes
- can describe the difference in communication characteristics of pub/sub and service/client in context of ROS
- can explain in own words the ROS architecture on one and multiple computers using schemes.
- can explain a complete ROS system from a given scheme
- can visualize a ROS system as a graph
- can inspect ROS topics
- can inspect ROS messages
- can plot ROS messages

ROS Framework

- Linux kernel (Ubuntu)
- Component oriented
- Each component = node
- Roscore = manager

Roscore

- Provides name service for a ROS system
- Manages communication for all the nodes
- Pre-requisites of a ROS-based system

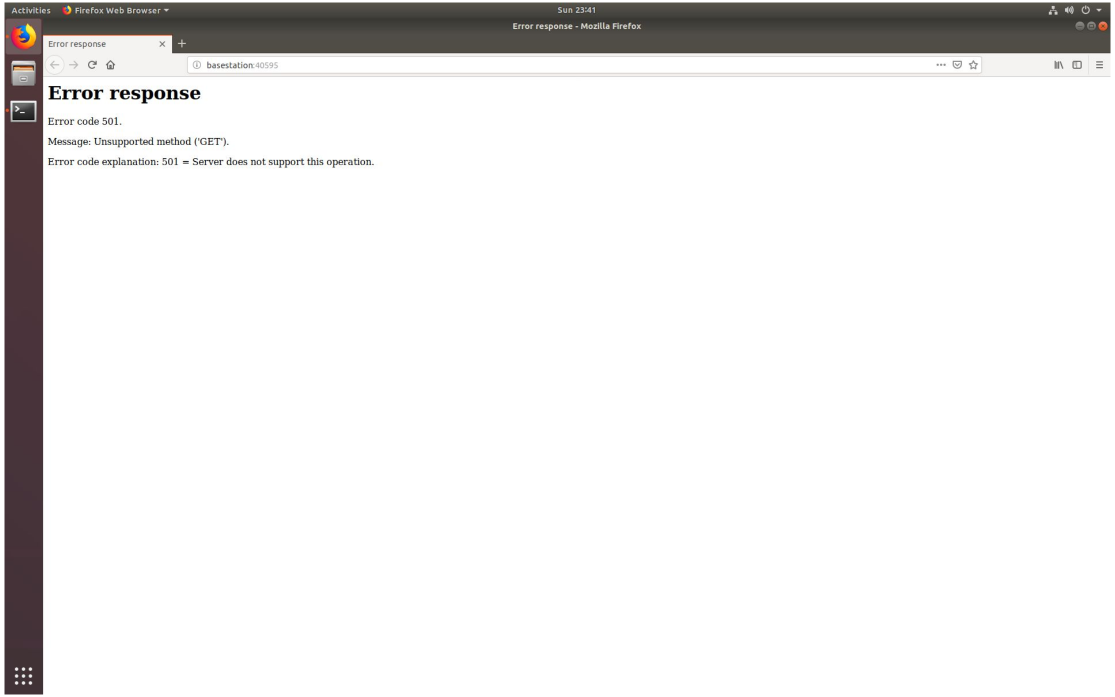
  → **Must be running!** (`$ roscore`)
- Consists out of a collection of nodes & programs
  - a ROS Master (http://wiki.ros.org/Master)
  - a ROS Parameter Server (http://wiki.ros.org/Parameter%20Server)
  - a rosout logging node (http://wiki.ros.org/rosout)
- `roslaunch` starts `roscore` if necessary

Roscore

Terminal 1

```
user@basestation:~$ roscore
... logging to /home/user/.ros/log/d75a18f4-d26b-11e5-8ec5-000c29ee3938/roslaunch-basestation-20409.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://basestation:40595/
ros_comm version 1.14.3

SUMMARY
========
PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.3

NODES
auto-starting new master
process[master]: started with pid [6021]
ROS_MASTER_URI=http://basestation:11311/

setting /run_id to 6c272ea8-4385-11e9-9fa2-000c29e9707b
process[rosout-1]: started with pid [6032]
started core service [/rosout]
```

Roscore != Webserver



Error response

Error code 501.

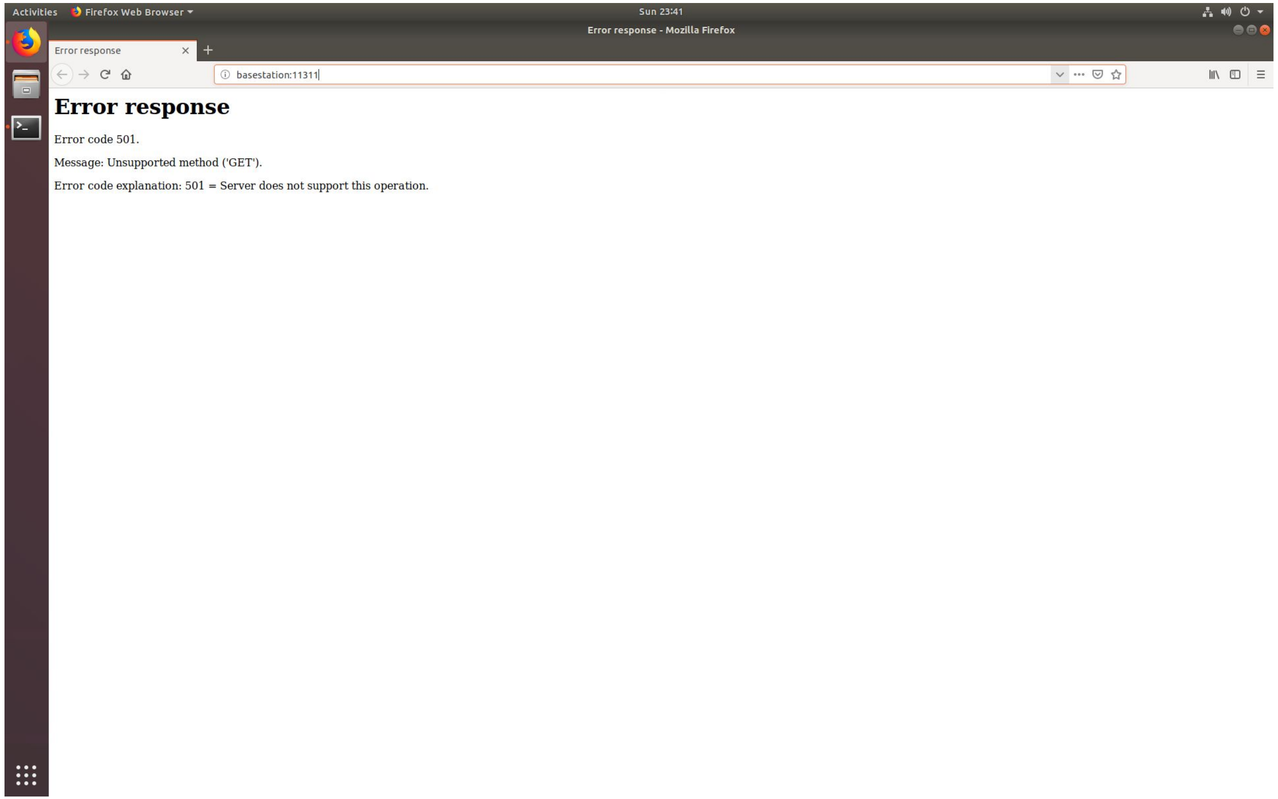
Message: Unsupported method ('GET').

Error code explanation: 501 = Server does not support this operation.

[INFO]

The URI http://basestation:11311 won’t work either.

Roscore != Webserver



Error response

Error code 501.

Message: Unsupported method ('GET').

Error code explanation: 501 = Server does not support this operation.

Node

- Base element (Component)
- Standalone unit
- Executable (Process)
- Communicates with other nodes

Node

- Uses Client library for communication
- Via ‘Topics’
- Publish/Subscribe (to) a topic
- Services
- Package groups functionality
- Package can contain multiple nodes
- Message broadcast

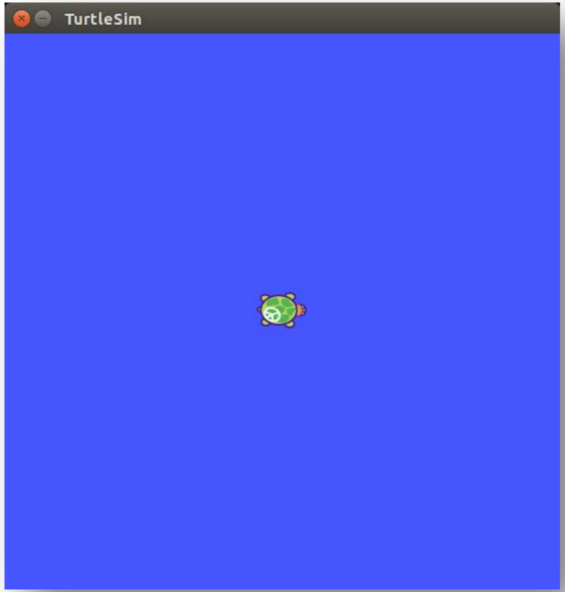
Example Node: turtlesim

Terminal 2

```
user@basestation:~$ rosrun turtlesim turtlesim_node
[ INFO] [1455392608.074807577]: Starting turtlesim with node name /turtlesim
[ INFO] [1455392608.084038816]: Spawning turtle [turtle1] at x=[5,544445], y=[5,544445], theta=[0,000000]
```

Terminal 3

```
user@basestation:~$ rosnode list
/rosout
/turtlesim
user@basestation:~$
```



[INFO]

```
$ rosrun [package_name] [node_name]
```

Node

Terminal 3

```
user@basestation:~$ rosnode info /turtlesim
--------------------------------------------------------------------------------
Node [/turtlesim]
Publications:
 * /turtle1/color_sensor [turtlesim/Color]
 * /rosout [rosgraph_msgs/Log]
 * /turtle1/pose [turtlesim/Pose]

Subscriptions:
 * /turtle1/cmd_vel [unknown type]

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

node http://basestation:33369/ ...
Pid: 21816
Connections:
 * topic: /rosout
    * to: /rosout
    * direction: outbound
    * transport: TCPROS
```

Node

Terminal 3

```
user@basestation:~$ rosnode ping /turtlesim
rosnode: node is [/turtlesim]
pinging /turtlesim with a timeout of 3.0s
xmlrpc reply from http://basestation:33369/    time=0.799894ms
xmlrpc reply from http://basestation:33369/    time=0.491858ms
xmlrpc reply from http://basestation:33369/    time=0.569105ms
xmlrpc reply from http://basestation:33369/    time=0.514030ms
xmlrpc reply from http://basestation:33369/    time=0.738144ms
xmlrpc reply from http://basestation:33369/    time=0.512838ms
xmlrpc reply from http://basestation:33369/    time=0.497103ms
^Cping average: 0.588996ms
user@basestation:~$
```

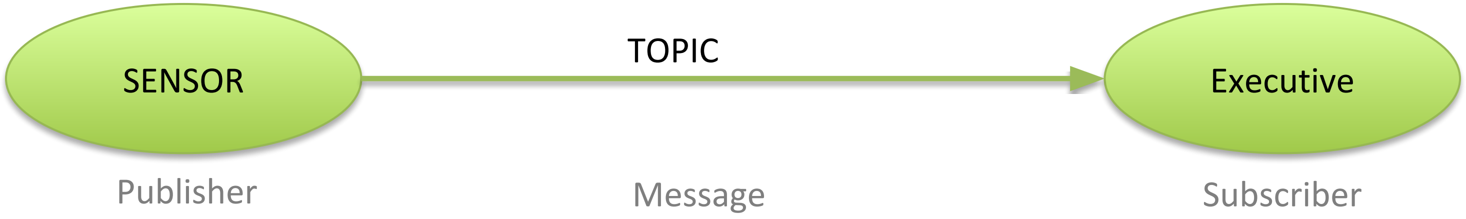
Topic

Terminal 3

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

Subscriptions:
 * /turtle1/cmd_vel [unknown type]
. . .
```

- Communication channel (Data stream)
- "many-to-many" one way communication
- "sender" Node → "publish messages to a topic"
- "receiver" Node → "subscribe to a topic"
- Rate (Hz) → system up-to-date

2 Nodes & 1 Topic



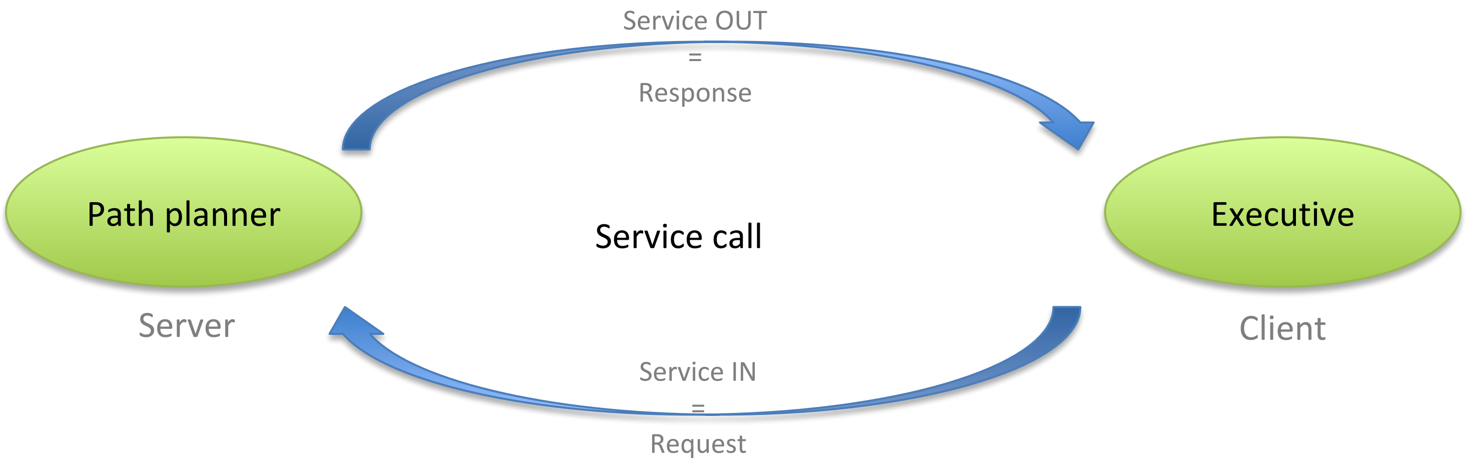
Services

Terminal 3

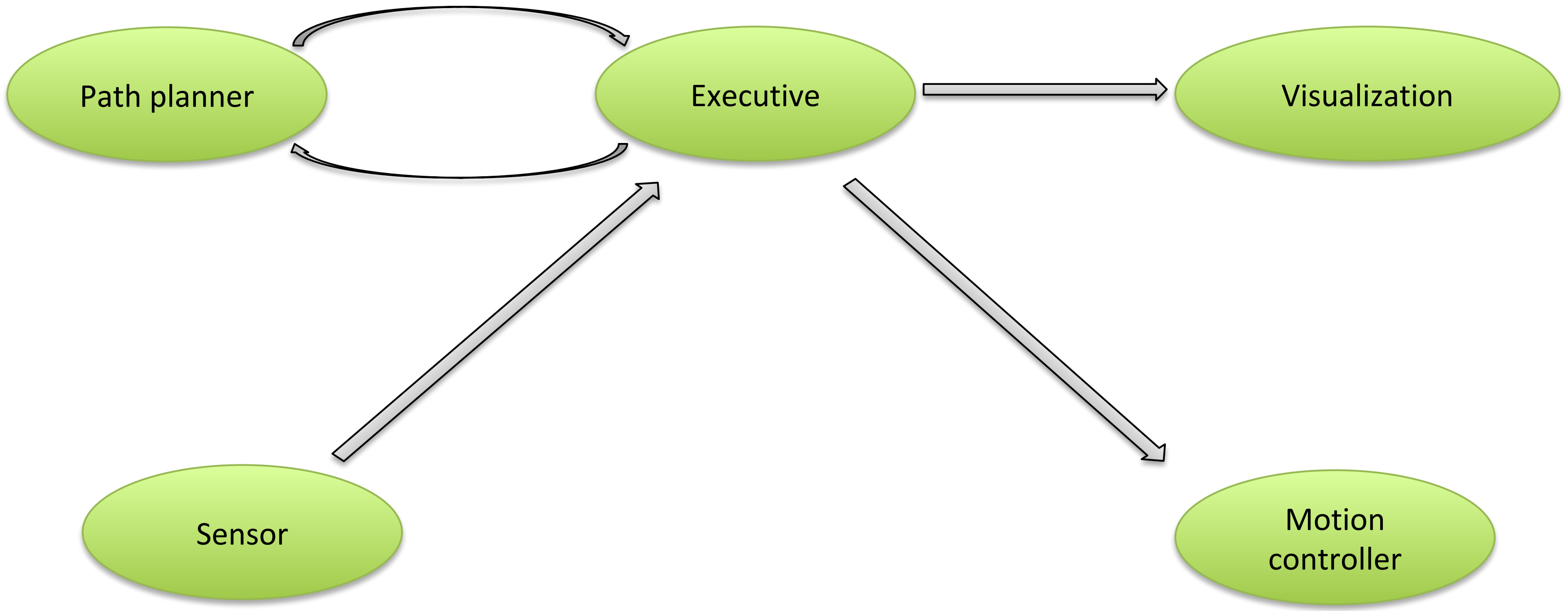
```
. . .
Services:
 * /turtle1/teleport_absolute
 * /turtlesim/get_loggers
 * /turtlesim/set_logger_level
 * /reset
 * /spawn
 * /clear
 * /turtle1/set_pen
 * /turtle1/teleport_relative
 * /killcontacting
```

- Communication channel
- ‘One-to-many’ two way communication
- Request / Reply
  (Like a function returning a value.)
- Slower than a topic

Services


Service OUT
=
Response
Path planner
Server
Service call
Executive
Client
Service IN
=
Request

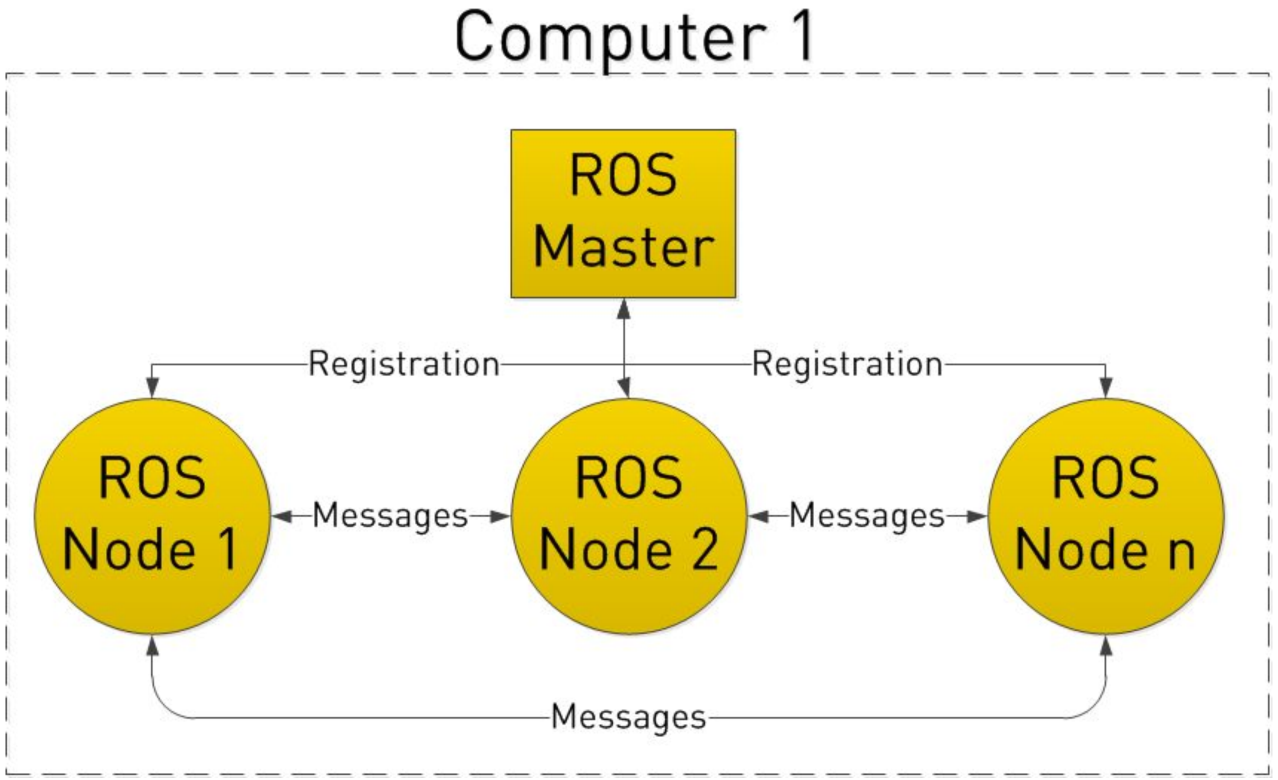
Complete Node Example



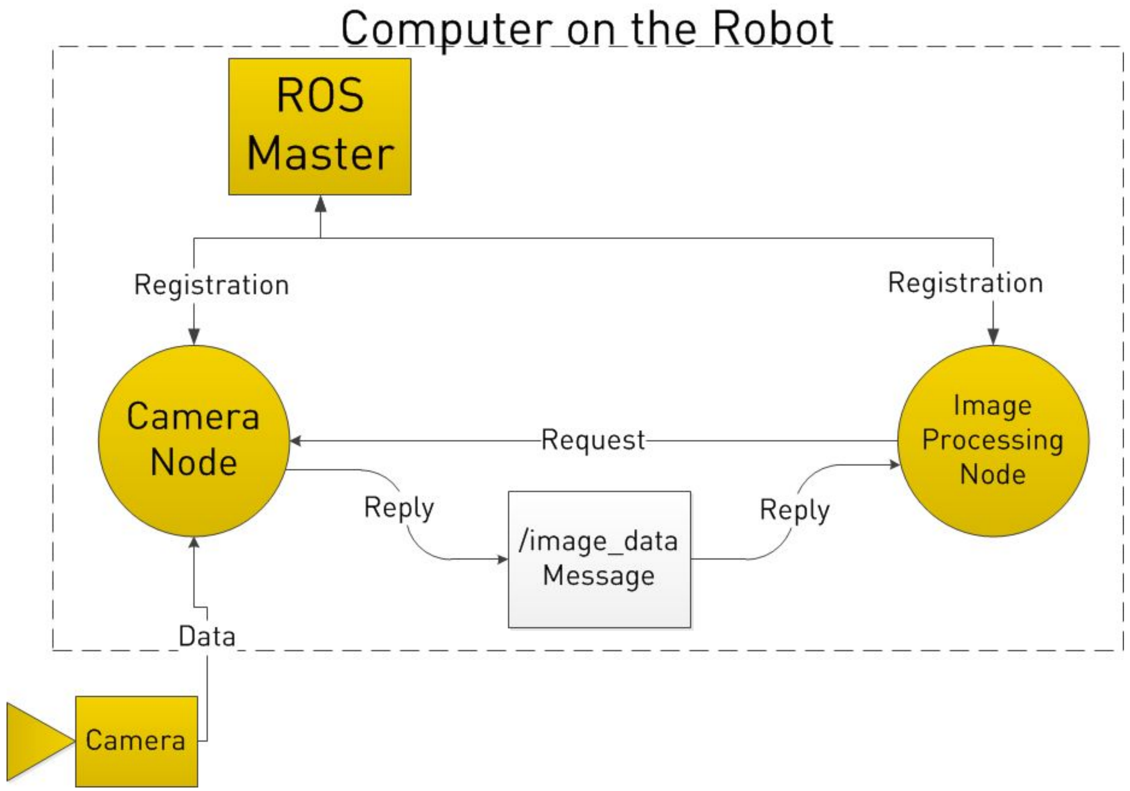
Camera, detection of objects in environment

Motions executions

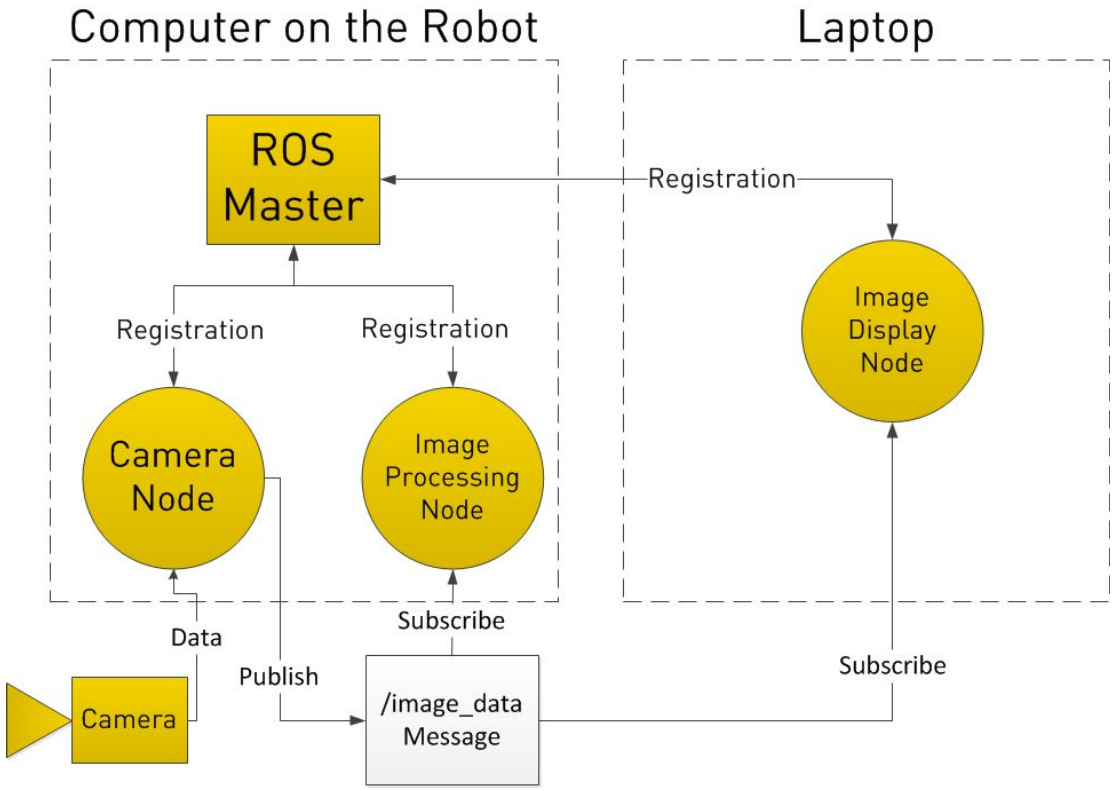
Architecture



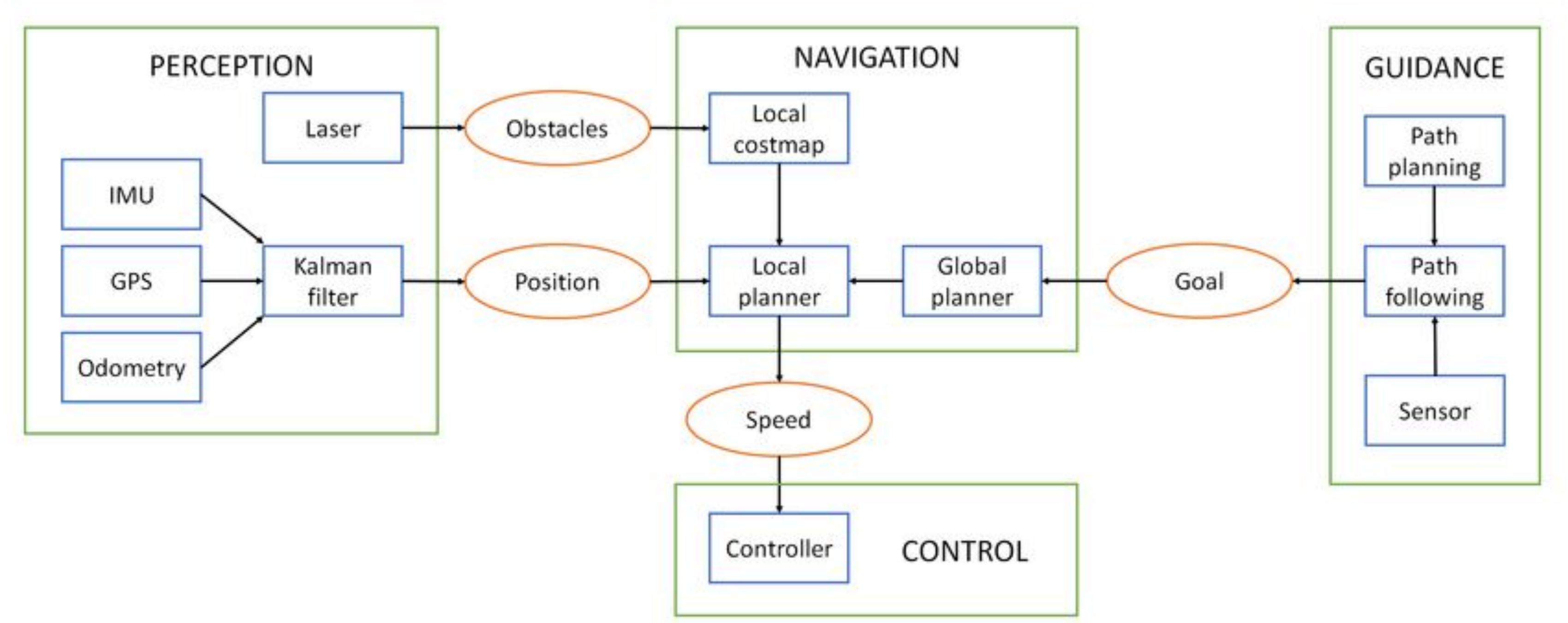
Architecture



Architecture



A complete system

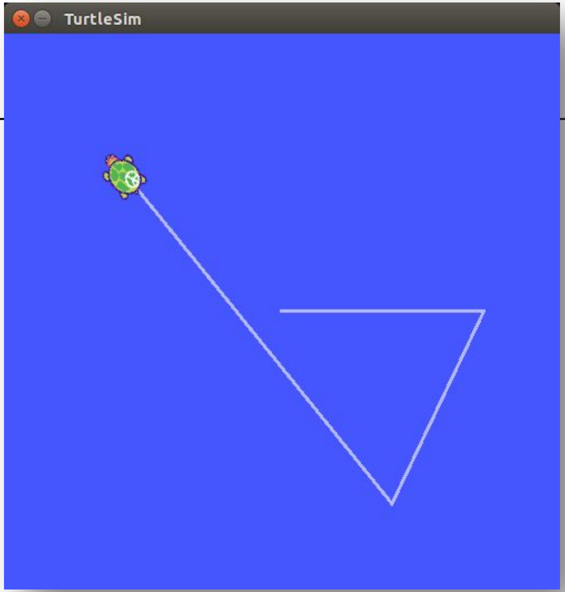


[SOURCE]
Ruiz-Larrea, Alberto & Roldán, Juan & Garzon, Mario & Cerro, Jaime & Barrientos, Antonio. (2016).
A UGV Approach to Measure the Ground Properties of Greenhouses.
https://www.researchgate.net/publication/290201414_A_UGV_Approach_to_Measure_the_Ground_Properties_of_Greenhouses

Example Node: turtlesim

Terminal 3

```
user@basestation:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle.
```



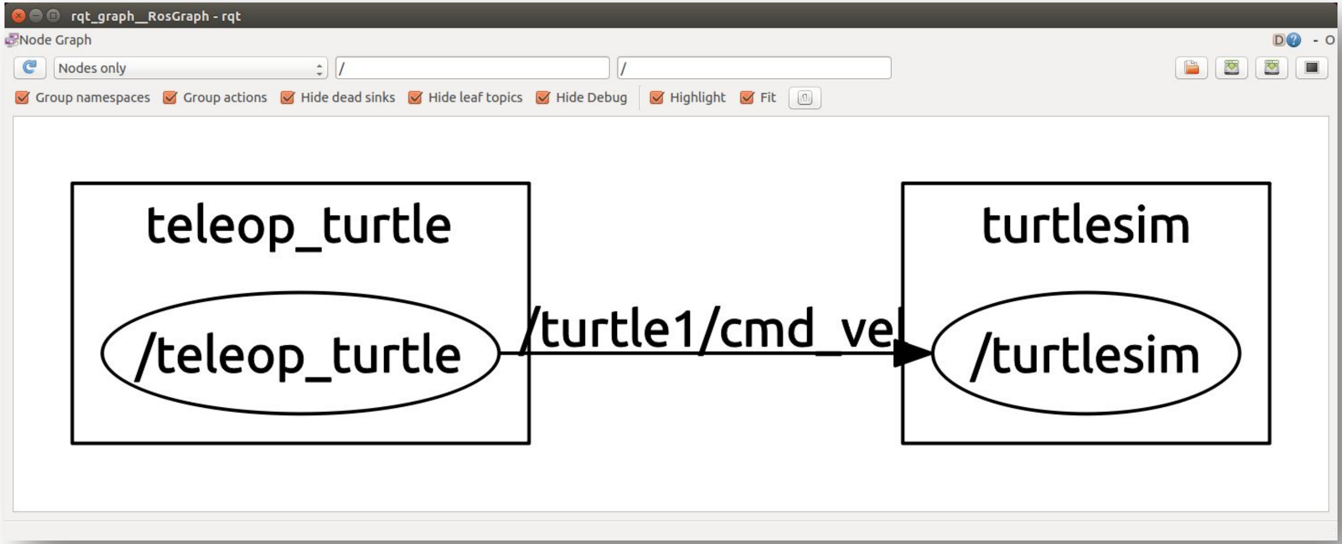
[INFO]

```
 $ rosrun [package_name] [node_name]
```

Visualization

Terminal 4

```
user@basestation:~$ rosrun rqt_graph rqt_graph &
```



Rostopic

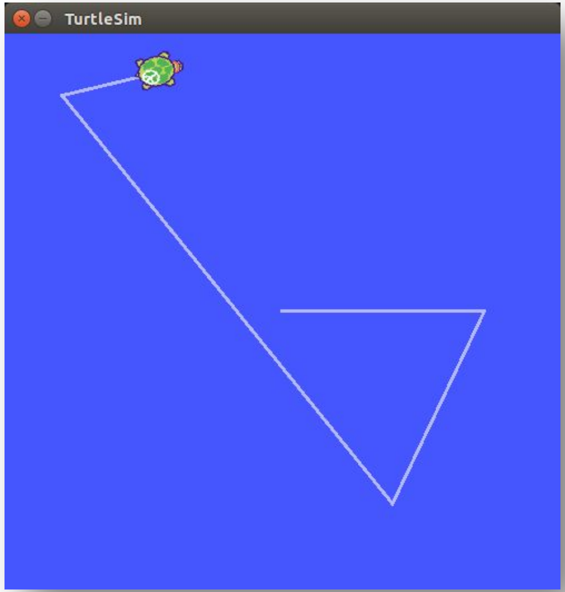
Terminal 4

```
user@basestation:~$ rostopic -h
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’

user@basestation:~$
```

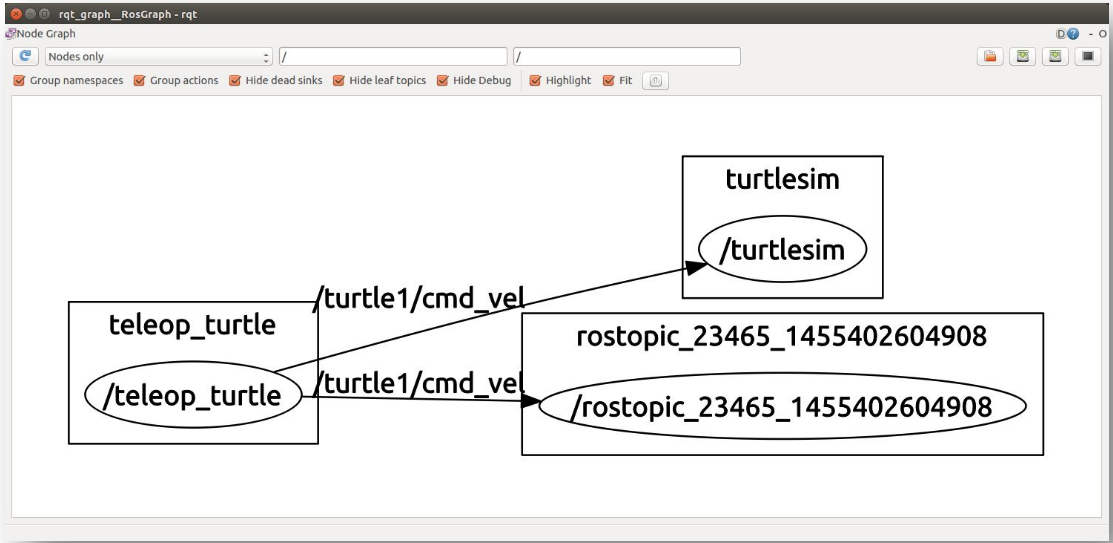
Rostopic

Terminal 4

```
user@basestation:~$ rostopic echo /turtle1/cmd_vel
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: -2.0
---
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
```



Visualization



Rostopic

Terminal 5

```
user@basestation:~$ rostopic list -v

Published topics:
 * /turtle1/color_sensor [turtlesim/Color] 1 publisher
 * /turtle1/cmd_vel [geometry_msgs/Twist] 1 publisher
 * /rosout [rosgraph_msgs/Log] 5 publishers
 * /rosout_agg [rosgraph_msgs/Log] 1 publisher
 * /turtle1/pose [turtlesim/Pose] 1 publisher

Subscribed topics:
 * /turtle1/cmd_vel [geometry_msgs/Twist] 2 subscribers
 * /rosout [rosgraph_msgs/Log] 1 subscriber
 * /statistics [rosgraph_msgs/TopicStatistics] 2 subscribers

user@basestation:~$
```

ROS Messages

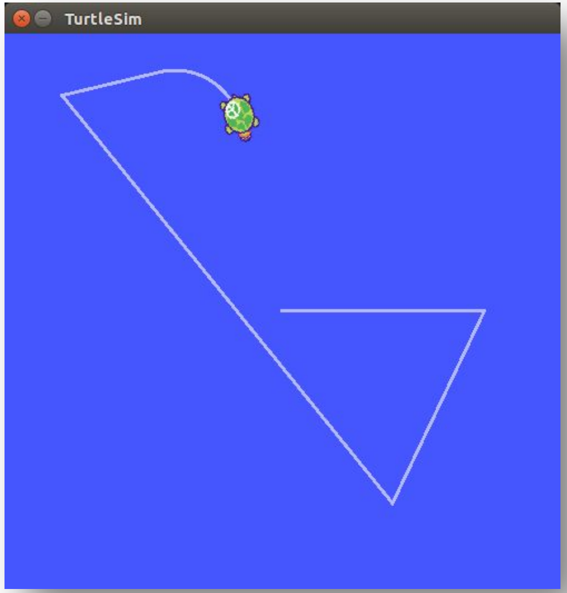
Terminal 5

```
user@basestation:~$ rostopic type /turtle1/cmd_vel
geometry_msgs/Twist
user@basestation:~$ rosmsg show geometry_msgs/Twist
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z

user@basestation:~$ rostopic pub -1 /turtle1/cmd_vel \
> geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, -1.5]'
publishing and latching message for 3.0 seconds
user@basestation:~$
```

[INFO]

```
$ rostopic type [topic]
$ rostopic pub [topic] [msg_type] [args]
```

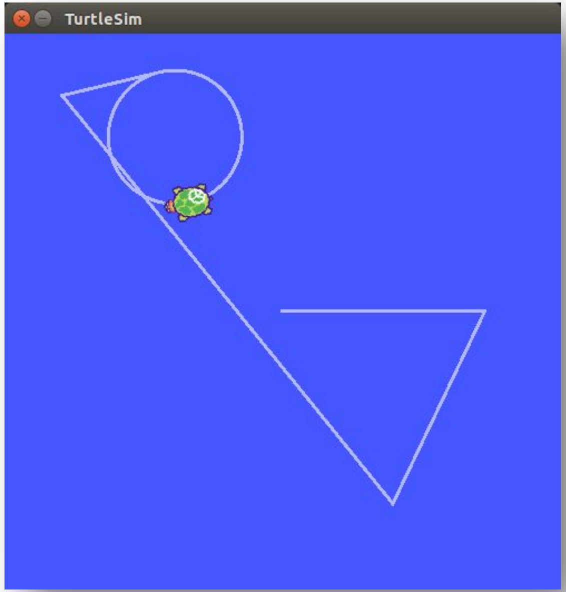
ROS Messages



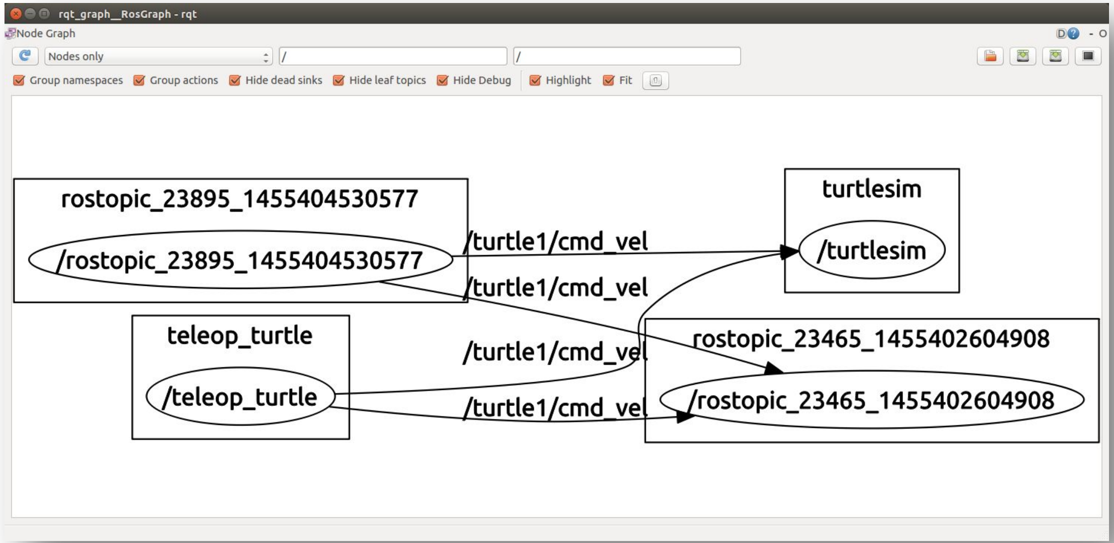
ROS Messages

Terminal 5

```
user@basestation:~$ rostopic pub -r 1 /turtle1/cmd_vel \
> geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, -1.5]'
```



Visualization



Rostopic

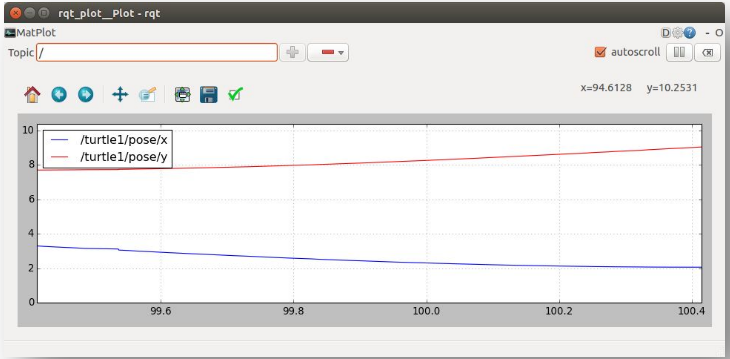
Terminal 6

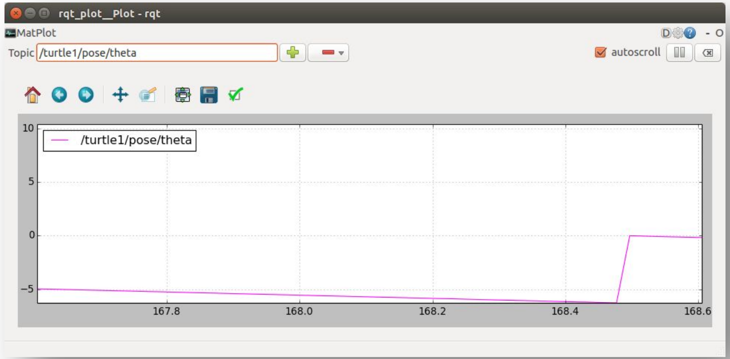
```
user@basestation:~$ rostopic hz /turtle1/pose
subscribed to [/turtle1/pose]
average rate: 62.500
	min: 0.015s max: 0.017s std dev: 0.00032s window: 62
average rate: 62.502
	min: 0.015s max: 0.017s std dev: 0.00028s window: 124
average rate: 62.500
	min: 0.015s max: 0.017s std dev: 0.00029s window: 187
average rate: 62.499
	min: 0.014s max: 0.018s std dev: 0.00033s window: 249
average rate: 62.500
	min: 0.014s max: 0.018s std dev: 0.00036s window: 312
average rate: 62.501
	min: 0.014s max: 0.018s std dev: 0.00035s window: 375
average rate: 62.499
	min: 0.014s max: 0.018s std dev: 0.00038s window: 437
average rate: 62.500
	min: 0.014s max: 0.018s std dev: 0.00039s window: 500
^C
```

Rostopic

Terminal 6

```
user@basestation:~$ rosrun rqt_plot rqt_plot
```





Terminology

Term	Description
Catkin	The official build system of ROS.
Node	An executable that uses ROS to communicate with other nodes.
Package	The main unit for organizing software in ROS.
Topic	Nodes can publish messages to a topic as well as subscribe to a topic to receive messages.
Publisher	“Talker” node which will continually broadcast a message.
Subscriber	“listener” node which will subscribe to a certain broadcasted message.
Service	Allow nodes to send a request and receive a response.
Master	Name service for ROS.
Rosout	ROS equivalent of stdout/stderr.
Roscore	Master + rosout + parameter server.
Parameter server	Nodes use this server to store and retrieve parameters at runtime.
Parameter	Can store strings, int, floats, …