



UNIVERSIDAD
NACIONAL
DE COLOMBIA



INTRODUCCIÓN ROS + MATLAB

Universidad Nacional de Colombia

Robótica 2020 - II

Prof. Pedro Cárdenas

Autor: Jurgen Krejci Muñoz

Octubre - 2020

¿Que vamos a trabajar?

- Manejo de ROS desde Matlab
- Creación / Manejo de paquetes
- Modelado de Robots (Formato URDF)
- Simulación en Gazebo
- ROS en múltiples máquinas (Conexión Remota)

Monitor - Jorgen Krejci Muñoz

Contacto - ihkrejci@unal.edu.co / Foro Moodle

Herramientas

- Ubuntu 18.04 LTS (Bionic Beaver) -> Recomendado
- ROS Melodic (Full Desktop)
- Emulador de terminal - Terminator
- Catkin Tools
- Visual Studio Code
 - Extensiones: ROS / ROS Snippets / URDF
- Matlab
 - Toolbox: Robotics Toolbox / ROS Toolbox

ROS desde Matlab

Las funciones de Matlab se llaman de forma similar a los comandos típicos de terminal de ROS:

Función	Descripción
<u>rosinit</u>	Connect to ROS network
<u>rostopic</u>	Retrieve information about ROS topics
<u>rosmmsg</u>	Retrieve information about ROS messages and message types
<u>roswode</u>	Retrieve information about ROS network nodes
<u>roswparam</u>	Access ROS parameter server values
<u>roswshutdown</u>	Shutdown ROS system

Source: Network Connection and Exploration

Inicialización de ROS

Comandos

roscpp

roscpp(hostname)

roscpp(hostname,port)

roscpp(URI)

roscpp(___,Name,Value)

Cerrar red

```
>> roscppshutdown
```

Inicia el nodo maestro (local)

```
>> roscpp
```

Conexión a nodo maestro en una IP específica

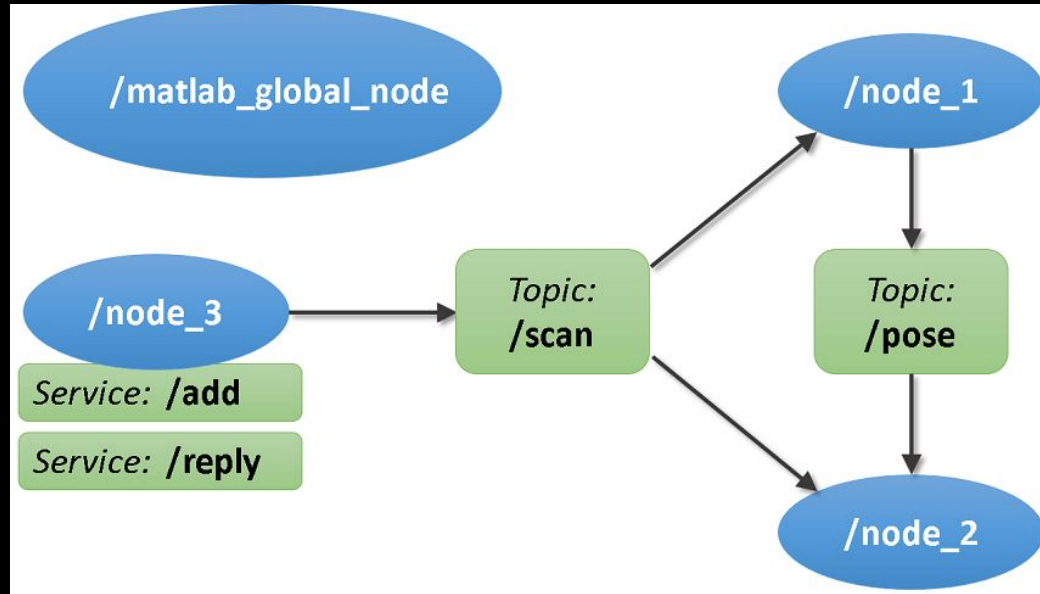
```
>> roscpp('192.168.17.128')
```

Conexión a nodo maestro en una IP específica
junto con nombre de nodo específico

```
>> roscpp('192.168.17.128',  
'NodeHost','192.168.17.1','NodeName','/test_node')
```

Red ejemplo

```
>> roslaunch
>> exampleHelperROSCreateSampleNetwork
>> exampleHelperROSShutdownSampleNetwork
```



Comandos de ROS en Matlab

TÓPICOS

Forma 1
(Estilo Terminal)

```
>> rostopic list  
>> rostopic echo topicname  
>> rostopic info topicname  
>> rostopic type topicname
```

Forma 2
(Script)

```
>> topiclist = rostopic("list")  
>> msg = rostopic("echo", topicname)  
>> topicinfo = rostopic("info", topicname)  
>> msgtype = rostopic("type", topicname)
```

Comandos de ROS en Matlab

NODOS

```
>> rosnode list  
>> rosnode info nodename  
>> rosnode ping nodename
```

```
>> nodelist = rosnode("list")  
>> nodeinfo = rosnode("info",nodename)  
>> rosnode("ping",nodename)
```

SERVICIOS

```
>> rosservice list  
>> rosservice info svcname  
>> rosservice type svcname  
>> rosservice uri svcname
```

```
>> svclist = rosservice("list")  
>> svcinfo = rosservice("info",svcname)  
>> svctype = rosservice("type",svcname)  
>> svcuri = rosservice("uri",svcname)
```


Comandos de ROS en Matlab

MENSAJES

```
>> rosmmsg show msgtype  
>> rosmmsg md5 msgtype  
>> rosmmsg list
```

```
>> msginfo = rosmmsg("show", msgtype)  
>> msgmd5 = rosmmsg("md5", msgtype)  
>> msglist = rosmmsg("list")
```

PARÁMETROS

```
>> rosparam list  
>> rosparam get pname  
>> rosparam set pname  
>> rosparam delete pname
```

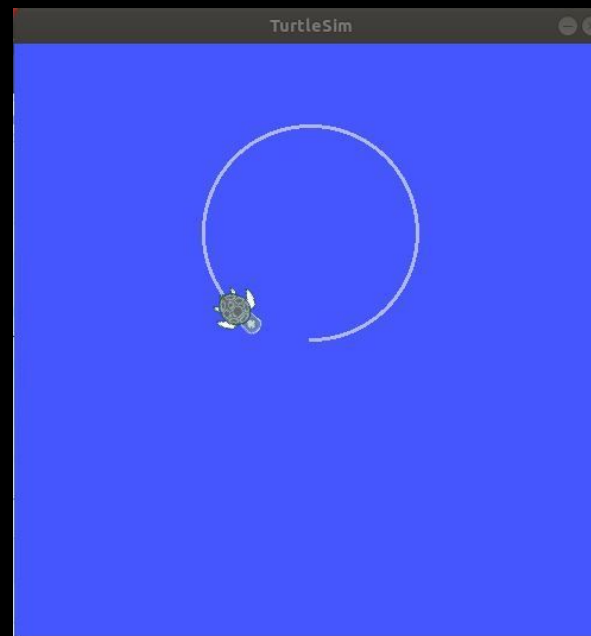
```
>> list = rosparam("list")  
>> pvalOut = rosparam("get",pname)  
>> rosparam("set",pname,pval)  
>> rosparam("delete",pname)
```

Intercambio de información

Para hacer intercambio de datos en ROS, usando Matlab utilizamos publicadores y suscriptores, esto con las herramientas rossubscriber y rospublisher

Ejemplo con TurtleSim

```
$ rosrunc turtlesim turtlesim_node
```



Intercambio de información

Ejemplo de creación de subscriptor:

```
%% Creating topic subscriber  
pose = rossubscriber('/turtle1/pose');  
pause(1);  
fprintf('\n pose,');  
disp(pose);  
  
%% Message reception  
actual_pose = receive(pose,3);  
disp(actual_pose);
```

Suscripción usando funciones 'Callback'

```
>> robotpose = rossubscriber('/pose',@exampleHelperROSPoseCallback)
```

Intercambio de información

Ejemplo de creación de publicador:

Twist Info

```
%% Creating topic publisher
disp('Topic info for /turtle1/cmd_vel:')
rostopic info /turtle1/cmd_vel;

commander = rospublisher('/turtle1/cmd_vel', 'geometry_msgs/Twist');
pause(1);
fprintf('\n commander,');
disp(commander);

%% Publishing Msgs
speed_command = rosmesssage(commander);

speed_command.Linear.X = 1;
speed_command.Angular.Z = 1;

send(commander,speed_command);
pause(2);
```

Llamada a servicios

Desde matlab podemos llamar servicios de nodos en disponibles en la res, para ello se maneja una estructura de cliente servidor (rossvcclient / rossvcserver)

% Creating service client and calling service

```
turtle_client = rossvcclient('/turtle1/set_pen');  
fprintf('\nClient: \n');  
disp(turtle_client);  
fprintf('\nRequest message: \n');  
request_msg = rosmessage(turtle_client);  
disp(request_msg);  
  
request_msg.R = 0;  
request_msg.G = 255;  
request_msg.B = 255;  
request_msg.Width = 4;  
  
fprintf('\nMessage details: \n');  
request_msg.showdetails;  
  
response = call(turtle_client,request_msg,'Timeout',3);  
fprintf('\nMessage call: \n');  
disp(response);
```

Servidor de parámetros

%% Parameter server values

```
fprintf("\nROS parameter list: \n");  
rosparam list;
```

```
fprintf("\nBackground blue: \n");  
background_b = rosparam("get", "/turtlesim/background_b");  
disp(background_b);
```

```
fprintf("\nChanging background color: \n");  
rosparam("set", "/turtlesim/background_r", 0);  
rosparam("set", "/turtlesim/background_g", 255);  
rosparam("set", "/turtlesim/background_b", 255);
```

```
background_r = rosparam("get", "/turtlesim/background_r");  
disp(background_r);
```

% Call for /clear service to update background color

```
clear_client = rossvcclient('/clear');  
clear_msg = rosmessage(clear_client);  
response = call(clear_client, clear_msg, 'Timeout', 3);
```

Referencias y recursos

- [ROS Toolbox \(Matlab\)](#)
- [Tutoriales de ROS \(Beginner\)](#)
- [Tutoriales URDF](#)
- [URDF Tutorial \(The Construct\)](#)
- [Git Crash Course](#)