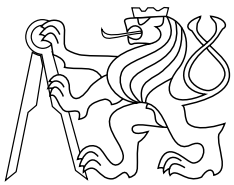
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

USER MANUAL

Rosbridge web interface

Martin Blaha, Michal Kreč, Petr Marek,
Tomáš Nouza, Tadeáš Lejsek

blahama7@fel.cvut.cz, krecmich@fel.cvut.cz, marekpe9@fel.cvut.cz,
nouzato1@fel.cvut.cz, lejsetad@fel.cvut.cz

May 28, 2013

Available at
https://cw.felk.cvut.cz/doku.php/misc/projects/nifti/sw/start

Supervisor: Ing. Michal Reinštein, Ph.D.,
Ing. Marek Polčák

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Rosbridge web interface

Martin Blaha, Michal Kreč, Petr Marek,
Tomáš Nouza, Tadeáš Lejsek

May 28, 2013

Abstract

A universal web GUI interface for mobile robots running ROS (Robot Opearating System) was developed as a semester project during the course A3M99PTO (in Czech called "Práce v týmu a její organizace"). The system allows to operate a mobile robot remotely using only a web browser. It does not matter, if an operator is next to the robot or on the opposite side of the planet. The system was successfully deployed on two different robotic platforms. This documents serves as a user manual to the application.

Contents

1 Software requirements

Although the system is designed to be platform independent for the user side, it has strict requirements for the robot side. It is not surprising that the robot must have installed ROS. Since the only officially supported OS for the ROS is currently the *Ubuntu*, the other required software has a *Linux* as a preferable platform. Even though *Windows* platform is labelled as experimentally supported on the ROS web site, no one from our team succeed in installing ROS on it.

For correct usage of our interface, the following must be installed on the robot:

- `Rosbridge` suite [1]
- `mjpeg_server` [2]
- `OpenSSH` server [3]
- `Shell In A Box` [4]
- `ros_pub_tf_echo` node (part of our system)

1.1 Rosbridge

The most important part of our software is based on the *rosbridge* version 2.0. It is an applications layer network protocol specification for the ROS emphasising the client/server model. Specifically, *rosbridge* allows clients to publish and subscribe topic messages and invoke services in the server's runtime environment. *Rosbridge* transports JSON-formatted messages over TCP sockets and web-sockets.

To install *rosbridge* it is recommended to follow the instructions here:
http://rosbridge.org/doku.php?id=downloading_rosbridge
It is also possible to install it as a debian package using:
`sudo apt-get install ros-groovy-rosbridge-server`
but this was not tested.

1.2 Mjpeg server

The *mjpeg_server* is a streaming server that subscribes to requested image topics in ROS and publishes those topics as MJPEG streams via HTTP. While *rosbridge* is capable of streaming video, as it is just another message type from ROS, the web browser is optimized to efficiently download images in binary format. Thus for increased performance benefits this additional

communication channel is used. In order to deal with a users specific requirements the video streams can be provided at a desired quality and size to accommodate different connection speeds and interface designs.

The *mjpeg_server* can be downloaded here:
https://github.com/RobotWebTools/mjpeg_server
It is a ROS package so simply copy it to your ROS workspace and `rosmake`.

1.3 OpenSSH server

The mobile robot has often no monitor and keyboard connected, when operating. If an operator wants to make some more difficult operation (e.g. copy some file, restart some application, etc.), he or she can connect to the robot remotely using SSH protocol and do what is needed. Commonly used *Linux* tool is the *OpenSSH*, which is distributed as a package with most of distribution.

For example on *Ubuntu* open terminal and type:

```
sudo apt-get install openssh-server
```

1.4 Shell In A Box

To enable the user to command a robot using SSH, the terminal window is a part of our web GUI. *Shell In A Box* is a open-source client/server tool that uses *OpenSSH server* on the server side and a javascript library on the client side. The functionality is the same as has any common terminal application.

The Shell In A Box can be downloaded here:
http://code.google.com/p/shellinabox/downloads/list
Download the latest version (source code) and install it using:

```
./configure
make
sudo make install
make clean
```

1.5 `ros_pub_tf_echo` node

Transformation messages are one of the most important messages in ROS. It provides transformations between every coordinates system on the robot. Our GUI uses them in painting the map, where the position of the laser scanner center is needed. Since the `/tf` messages are published 1000 times per second they must be filtered to prevent network overload. This filtration is made by our `ros_pub_tf_echo` node where only transformation between

map and laser frame is transferred. The output filtrated frequency can be set using rosparam `/freq_tf` (default value is 10) also as `/source_tf` (default `/map`) and `/target_tf` (default `/laser`).

This node is a part of our distribution package. It is a standard ROS package so simply copy it to your ROS workspace and `rosmake`.

1.6 Webserver requirements

All the above described software must run on the robot. The client only need to connect to a webserver where is our application stored. While only a javascript is used (all the computation is on the client), there are no special needs for the webserver (like PHP, etc.) when using our basic environment. Example can be found here: http://www.rosbridge.felk.cvut.cz/gui.

There is also an option to allow access only for the authorized users. For this feature we made an user management system based on the *WordPress* [5] that has an administration interface for managing the users. This system requires the *MySQL* [6] database installed on the server. Example can be found here: http://www.rosbridge.felk.cvut.cz/wordpress.

Source codes of our system can be downloaded here:
http://rosbridge.felk.cvut.cz/source_codes/source_codes.zip

2 Usage

On the machine where the *roscore* is running (commonly the robot) run:

1. the rosbridge: `roslaunch rosbridge_launch simple.launch`
2. the mjpeg server: `roslaunch mjpeg_server mjpeg_server.launch`
3. the Shell In A Box: `shellinaboxd -s /:SSH -t`
4. tf message filter: `roslaunch ros_pub_tf_echo web.launch`

Now the system is ready. Visit the site with our web gui[1], fill in the IP address of aforementioned machine and click the connect button.

The back-end of our website is powered by *WordPress*. Thus, it is very simple to manage different user accounts. For a simple user usage of the website type `user` as the login and `user` as the password. If you want to connect with the administrators privileges, type `admin` as the login and `heslo` as the password. You can operate the robot in both modes and the user interface

[1] login when using the *WordPress* version

will be the same. If you want to manage other user accounts, login as admin. An additional toolbar appears on the top of the page. Click "ROS user interface" to get to the administrator's interface. Click on users in the left menu to manage the user accounts. The *WordPress* theme that is used is called *Toolbox*. All *Javascript* files are stored in rosbridge/var/www/wordpress/wp-content/themes/mytoolbox/js, you can find the CSS style sheet in rosbridge/var/www/wordpress/wp-content/themes/mytoolbox. It is the file `style.css`. If you want to modify the HTML code of the page, stay in the same folder and open the file `page.php`. The code to change starts on the 21^{st} line.

3 GUI description

The GUI has five main parts and is in the Fig. 1. The usage is not limited to the one user so it can be opened multiple times on several computers/tablets/...

3.1 Camera window

Human gets most of information by his or her eyes thus the camera is the most important sensor for an operator. User can choose here any camera topic, that is published by robot and set its image size and quality. Because of the high network load, only one camera image is shown at the webpage.

3.2 Map window

Second most important information is how the robot senses the environment. For this purpose this windows shows the 2D map (message type `nav_msgs/OccupancyGrid` published on the topic `/map`), the laser scan (`sensor_msgs/LaserScan` published on the topic `/scan`), obstacles (`nav_msgs/GridCells` on the topic `/move_base/local_costmap/obstacles`), robot footprint (`geometry_msgs/PolygonStamped` on the topic `/move_base /local_costmap/robot_footprint`) and the laser center position as a red dot (obtained from the topic `/laser_web_tf` produced by the `ros_pub_tf_echo` node). As the robot moves in the environment, the map grows and is automatically downscaled to fit the window.

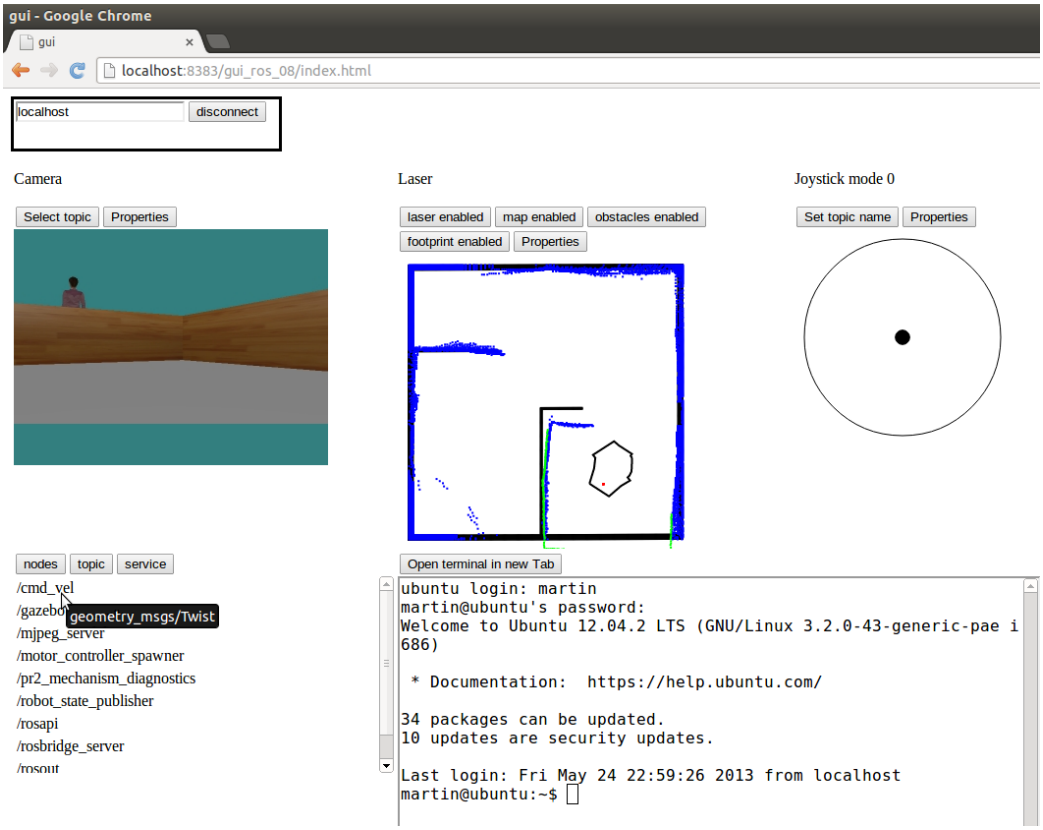


Figure 1: The web GUI with Gazebo simulator

3.3 Joystick window

For the basic moving with the robot, there is a software emulation of a joystick. It looks like a circle with dot in the middle. The dot is representing the zero position. By mouse clicking and dragging to this circle, the joystick commands are emulated. Before issuing any commands, the topic for sending `geometry_msgs/Twist` must be selected. Joystick has also two modes. Default mode reads finger position as the speed in the direction of the x-axis and rotation around the z axis. The second mode reads finger position as the velocity in the direction of the x-axis velocity in the y-direction.

3.4 Information window

This window works mainly for debugging. It is often the last part of the GUI that stops working because it has lowest network load. It has three function:

- Show all the nodes currently running on the robot
- Show all the topics currently running on the robot
- Show all the services currently running on the robot

3.5 Terminal window

The most powerful tool of our GUI is the terminal window. It automatically opens a SSH connection to the robot. From here, the user can do almost everything with not only ROS but even with the operating system on the robot computer.

References

[1] "rosbridge v2.0." http://rosbridge.org/doku.php?id=rosbridge_protocol_v2.0. Accessed: 25/05/2013.

[2] "Mjpeg server." http://www.ros.org/wiki/mjpeg_server. Accessed: 25/05/2013.

[3] "Openssh." http://www.openssh.org/. Accessed: 25/05/2013.

[4] "Shell in a box." http://code.google.com/p/shellinabox/. Accessed: 25/05/2013.

[5] "Wordpress." http://wordpress.org/. Accessed: 25/05/2013.

[6] "Mysql." http://www.mysql.com/. Accessed: 25/05/2013.