Robot Practical Course

Assignment #1

This assignment is supposed to familiarize yourself with the ROS and the `itr_rpc` package.

Task 1.1 Set up ROS and your own workspace: Git is installed on all our machines. For now, you will only need to check out the package. If you are unfamiliar with git and would like to know more, please read a guide on how to use git (e.g. `https://githowto.com/`) after the course. ROS is also installed with all required components on all our machines. The documentation can be found at `https://wiki.ros.org`

1.1.1: First of all, you need to set up your environment and workspace for ROS. This only has to be done once for your user account.

- Login to the pool workstations with your informatics account (`Xname`)
- Open a command shell (`Ctrl+Alt+T`)
- Load the ROS environment `source /opt/ros/kinetic/setup.bash`
- Create a folder for your workspace `mkdir -p ~/catkin_ws/src` (The ~ is an alias for your home-folder in Linux, `mkdir -p` creates a new folder with all required intermediate folders.)
- Change into the workspace `cd ~/catkin_ws/` (`cd` stands for change directory.)
- Initialize the workspace `catkin_make` (This is the command for building your workspace. It has to be called from the root of your workspace. You will need it again later on.)

Your personal workspace is now set up. Nevertheless, you would have to tell each shell which you open about your workspace and the ROS environment. We will now make this permanent, so you do not forget to load it.

- In your shell, open your shell profile either in a graphical or a command line editor
  `vim ~/.bashrc` (command line)
  `gedit ~/.bashrc` (graphical)
- At the end of the file, add `source ~/catkin_ws/devel/setup.bash`. Save and close.
- In all open shells, enter `exec bash`

1.1.2: Check out the project from the git repository. The will be updates to the project during the practical course. In order to avoid conflicts, only edit the files which you are instructed to edit.

- Navigate to the `src` folder in your workspace.
- Check out the git repository `git clone https://gogs.mafiasi.de/einig/itr_rpc.git` using your informatics account (with a two digit number).
- Build your workspace to generate all required message and service definitions.
- Update your package index `rospack profile`

Now you are all set with your ROS environment and the workspace. For the future, there are some ROS specific command which might be useful:
`roscd`, `rosrun`, `roslaunch`, `roscore`, `rostopic`, `rosservice`, `rosclean`, `rosmsg`, `rosnode`.
All these commands support `tab completion`, be sure to use it as it will avoid typing mistakes.

Task 1.2 Launching ROS nodes: In this task, you will learn how to launch ROS nodes and how to interact with ROS.

1.2.1: Launch the graphical display `roslaunch itr_rpc task_1.launch`. Inspect the available messages, topics and services.

1.2.2: Run the default forward kinematics script `rosrun itr_rpc dummy_fk.py`. Discuss the behavior.

1.2.3: ROS supports dynamically reconfiguring parameters of running nodes. Run the configuration GUI `rosrun rqt_reconfigure rqt_reconfigure` and switch off the velocity limits. Explain the difference in the behavior.

Task 1.3 Write your first node: In this task, you will write your first own node which performs a circular motion with the TCP. **Ensure, that the velocity limits are off**. After relaunching the GUI, you will have to change the parameter again.

When writing code in Python, ensure that you are using the correct amount of spaces for indentation. In case you use `gedit` as editor, on the bottom bar, set the tab width to `4` and to `Use Spaces` (Check box must be checked).

1.3.1: The circular motion must be performed around $\left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$ with a radius of l_2. Inspect the GUI to find out the orientation of the coordinate system. Write your script in the `nodes/script.py` file and inspect `nodes/dummy_fk.py` for assistance.

1.3.2: Turn on the velocity limits. Explain what is happening and why. Can you fix your code to work with the velocity limits?

1.3.3: The circular motion is supposed to be only around $\left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$. Verify that your circular motion does not start before reaching that position by relaunching the GUI.