Documentation: Jackal robot simulation in Gazebo using ROS

Author: Dennis Benders
Last edited: 17.11.2019

This document contains the installation procedure for:

- ROS Kinetic
- Gazebo 7 (as default Gazebo along with ROS Kinetic)
- Gazebo 9 (from source)
- Gazebo 9 when Gazebo 7 was already installed (from source)
- Jackal simulation in Gazebo 7
- Jackal simulation in Gazebo 9 (from source)
- Matlab R2019a

Each installation procedure is described starting on a new page.

This document is written as part of a project where the IMU data of the Jackal robot in the simulation was analyzed. If you encounter other issues with sensors/actuators on the Jackal robot in the simulation, please send it to the author.

It is recommended to follow this document from beginning to end (depending on which Gazebo version you want to use, you can skip the other Gazebo version installation procedures). Sometimes certain settings are assumed to be known, such including the setup.bash files in the .bashrc file. Sourcing of these files manages overlaying of packages, thereby being able to reuse package names. The lastly sourced file comes from the Catkin workspace in which the ROS packages will first be found.

It is also recommend to always back-up your correctly working setup!

All content in this document is tested on Ubuntu 16.04!

ROS Kinetic

See http://wiki.ros.org/kinetic/Installation/Ubuntu.

1. Configure your Ubuntu repositories

See https://help.ubuntu.com/community/Repositories/Ubuntu.

1.1 Open Software and Updates (e.g. by pressing the “Windows” on your keyboard and typing in “Software and Updates”)
1.2 Ensure that main, universe, restricted and multiverse are allowed

2. Setup your sources.list

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

In case this command is not working, use this command: echo 'deb *http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main' | sudo tee -a /etc/apt/sources.list.d/ros-latest.list*

3. Set up your keys

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' –recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

If you experience issues connecting to the keyserver, you can try substituting *hkp://pgp.mit.edu:80* or *hkp://keyserver.ubuntu.com:80* in the previous command.

Alternatively, you can use curl instead of the apt-key command, which can be helpful if you are behind a proxy server:
curl -sSL 'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0xC1CF6E31E6BADE8868B172B4F42ED6FBAB17C654' | sudo apt-key add -

4. Installation

4.1 Update your Debian package:
sudo apt-get update

4.2 Full installation of ROS Kinetic:
sudo apt-get install ros-kinetic-desktop-full

If you don’t want the full installation, use:
sudo apt-get install ros-kinetic-desktop

If you just want the ROS-base (bare bones), use:
sudo apt-get install ros-kinetic-ros-base

If you want to install an individual package, use:
sudo apt-get install ros-kinetic-PACKAGE

To find available packages, use:
apt-cache search ros-kinetic

5. Initialize rosdep
Before you can use ROS, you will need to initialize rosdep. rosdep enables you to easily install system dependencies for source you want to compile and is required to run some core components in ROS.
sudo rosdep init
rosdep update

6. Environment setup
It's convenient if the ROS environment variables are automatically added to your bash session every time a new shell is launched:
echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
source ~/.bashrc

TAKE CARE: ~/.bashrc must only source the setup.bash for the ROS version you are currently using!
~/.bashrc can be opened using the file manager. Type CTRL+H when you are located in Home. The .bashrc file should appear and is automatically opened in your default text editor.
You can also open a new terminal (automatically starting at the system root) and typing:
vim .bashrc

7. Dependencies for building packages
Up to now you have installed what you need to run the core ROS packages. To create and manage your own ROS workspaces, there are various tools and requirements that are distributed separately. For example, rosinstall is a frequently used command-line tool that enables you to easily download many source trees for ROS packages with one command.
To install this tool and other dependencies for building ROS packages, run:
sudo apt install python-rosinstall python-rosinstall-generator python-wstool build-essential

8. RECOMMENDED: look up the ROS tutorials to learn the basic concepts of ROS (i.e. to create your own catkin workspace and ROS packages): http://wiki.ros.org/ROS/Tutorials.

Gazebo 7 TODO: test only from source method

See http://gazebosim.org/tutorials?tut=ros_installing&cat=connect_ros.

There are two ways of installing Gazebo 7

- Install from Pre-Built Debians (recommended when using ROS Kinetic)
- Install from source

Install from pre-built Debians (recommended)

1. Install the gazebo_ros_pkgs

sudo apt-get install ros-kinetic-gazebo-ros-pkgs ros-kinetic-gazebo-ros-control

Install from source

1. Clone the Github repositories with source code in your (Catkin) workspace

1.1 If you haven't installed git yet:
sudo apt-get install git

1.2 Kinetic is using the gazebo 7.x series, start by installing it:
sudo apt-get install -y libgazebo7-dev

1.3 Download the source code from the gazebo_ros_pkgs Github repository:
cd PATH_TO_YOUR_WORKSPACE/src (e.g. *cd ~/jackal_ws/src*)
git clone https://github.com/ros-simulation/gazebo_ros_pkgs.git -b kinetic-devel

1.4 Check for any missing depencies using rosdep:
rosdep update
rosdep check --from-paths . --ignore-src --rosdistro kinetic

1.5 You can automatically install the missing depencies using rosdep via debian install:
rosdep install --from-paths . --ignore-src --rosdistro kinetic -y

2. Build your workspace

cd PATH_TO_YOUR_WORKSPACE/ (e.g. *~/jackal_ws/*)
catkin_make
source devel/setup.bash

3. Test Gazebo integrated in ROS

Be sure that your workspace setup.bash is included in your .bashrc file, so that it is automatically sourced every time you start a new terminal (include the line, e.g.: *source ~/jackal_ws/devel/setup.bash* at the end of the .bashrc file).
Now test Gazebo with ROS:
roscore

Open a new tab in your terminal using CTRL+SHIFT+T and enter:
rosrun gazebo_ros gazebo (or use: *roslaunch gazebo_ros empty_world.launch*)

If you get the following error: "gzserver: symbol lookup error: /usr/lib/x86_64-linux-gnu/libsdformat.so.4: undefined symbol:

_ZN8ignition4math15SemanticVersionC1ERKNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEE", please try the following command:
sudo apt upgrade libignition-math2

You can also look up the available ROS topics in a new terminal tab:
rostopic list

IMPORTANT NOTE:
Take care of the fact that, when having build packages from source, your workspace will change. After removing these packages and building the workspace again, the build files of the removed packages will remain. In almost all cases your workspaces is the lastly sourced file, so all other packages with the same name will be overwritten with your (deleted) packages. The code is now broken, because you deleted the packages, so your simulation may also be broken (particularly, this can happen when you source the hector_gazebo_plugins to get IMU data from you simulation. This IMU data will not be shown when typing the command "rostopic echo imu/data").
In this case, the best thing you can do is to build a new workspace. Therefore, you need a back-up of your correctly working workspace before building the packages from source. First, delete your current workspace, then create a new workspace (http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment). Create new packages (http://wiki.ros.org/ROS/Tutorials/CreatingPackage) corresponding to your old packages and replace the package.xml and CmakeLists.txt files of those packages with the previous ones (if you created messages in your old package, for instance, these settings in the package.xml are included again, without going through the whole tutorial on creating custom messages). Now copy-paste all custom directories (e.g. msg, launch, scripts, etc.) from your old packages into your new packages and build your workspace ("catkin_make" when having your workspace path active in the terminal). This should finish successfully and you have re-created your workspace.

Gazebo 9 (from source)

TAKE CARE: ROS Kinetic is not officially compatible with Gazebo 9: see http://gazebosim.org/tutorials/?tut=ros_wrapper_versions)

See https://www.theconstructsim.com/all-about-gazebo-9-with-ros/.
See http://gazebosim.org/tutorials/?tut=ros_wrapper_versions.

1. Update the repository

sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb_release -cs` main" /etc/apt/sources.list.d/gazebo-stable.list'
wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -

In case the first command is not working, use this command: echo 'deb *http://packages.osrfoundation.org/gazebo/ubuntu-stable lsb_release -cs main' | sudo tee -a /etc/apt/sources.list.d/gazebo-stable.list*

2. Update the repository of packages

sudo apt-get update

3. Install Gazebo 9

sudo apt-get install ros-kinetic-gazebo9-*

This command will usually install:

- ros-kinetic-gazebo9-dev
- ros-kinetic-gazebo9-ros-control
- ros-kinetic-gazebo9-msgs
- ros-kinetic-gazebo9-ros
- ros-kinetic-gazebo9-ros-pkgs

To install one of these package apart (e.g. ros-kinetic-gazebo9-dev), use:
sudo apt-get install ros-kinetic-gazebo9-dev

4. Test whether Gazebo 9 is working properly

gazebo

This command should give you the Gazebo 9 GUI showing an empty world and a started simulation (see bottom of the screen for the current simulation status).

Gazebo 9 when Gazebo 7 was already installed (from source)

See https://www.theconstructsim.com/all-about-gazebo-9-with-ros/.

1. Uninstall the default Gazebo

*sudo apt-get remove ros-ROS_DISTRO-gazebo**
*sudo apt-get remove libgazebo**
*sudo apt-get remove gazebo**

2. Follow all steps listed under Gazebo 9 (from source)

Jackal simulation in Gazebo 7 TODO: test from source only

See http://wiki.ros.org/jackal_simulator.
See http://docs.ros.org/indigo/api/jackal_tutorials/html/simulation.html.
NOTE: these links show the setup for ROS Indigo. However, the same commands can be used for ROS Kinetic!

There are two ways of installing the Jackal files

- Install from Pre-Built Debians (recommended when using ROS Kinetic)
- Install from source

Install from pre-built Debians (recommended)

1. Install Jackal packages

sudo apt-get install ros-kinetic-jackal-simulator ros-kinetic-jackal-desktop

Install from source

1. Clone the Github repositories with source code in your (Catkin) workspace

cd PATH_TO_YOUR_WORKSPACE/src (e.g. *cd ~/jackal_ws/src*)
git clone https://github.com/ros-simulation/gazebo_ros_pkgs.git -b kinetic-devel

2. Build and source

cd PATH_TO_YOUR_WORKSPACE/ (e.g. *~/jackal_ws/*)
catkin_make
source devel/setup.bash

3. Test the Jackal simulation in Gazebo 7 using ROS

roscore
roslaunch jackal_gazebo jackal_world.launch

You can also look up the available ROS topics in a new terminal tab:
rostopic list

This should show more topics than just the default Gazebo topics, see the picture below. For example, the /imu/data topic should provide you the IMU data of the Jackal robot. You can test the simulation to show the data being published to the topic using the following command (e.g. to show the IMU data of the Jackal robot):
rostopic echo imu/data

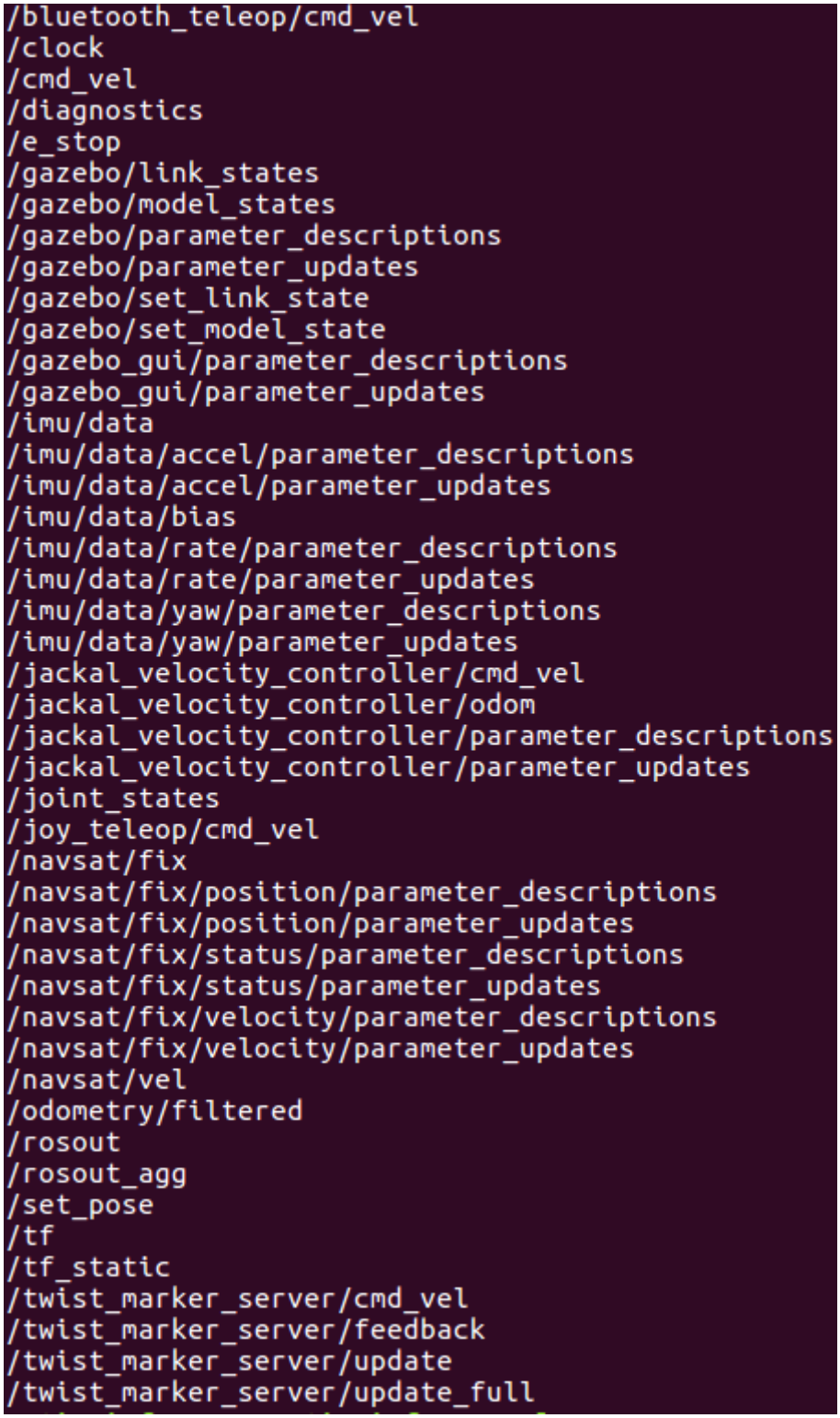
4. Visualize ROS topic data

Tip: in order to easily visualize data from different ROS topics, record a rosbag. The following command shows you the help function of rosbag record:
rosbag record -h

In order to record the /imu/data ROS topic, save it with filename imu_data_CURRENT_DATE_AND_TIME.bag with total recording time 10 seconds, execute the following command:
rosbag record imu/data -o test.bag --duration 10

Recorded rosbags can also easily be visualized with PlotJuggler (https://github.com/facontidavide/PlotJuggler). Install PlotJuggler by executing the following command:
sudo apt-get install ros-kinetic-plotjuggler

To run PlotJuggler, enter:
rosrun plotjuggler PlotJuggler



Jackal simulation in Gazebo 9 (from source) TODO: test this!

See https://gist.github.com/vfdev-5/57a0171d8f5697831dc8d374839bca12.

1. Install the necessary packages

sudo apt-get install ros-kinetic-robot-localization ros-kinetic-controller-manager ros-kinetic-joint-state-controller ros-kinetic-diff-drive-controller ros-kinetic-gazebo-ros ros-kinetic-gazebo-ros-control ros-kinetic-gazebo-plugins ros-kinetic-lms1xx ros-kinetic-pointgrey-camera-description ros-kinetic-roslint ros-kinetic-amcl ros-kinetic-gmapping ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-message-runtime ros-kinetic-topic-tools ros-kinetic-teleop-twist-joy

This command ensures that all necessary packages are installed. However, when just reading out IMU data of the Jackal robot, for example, not every package is needed. If you have enough space on your computer, just install every package.

2. Clone the Github repositories with source code in your (Catkin) workspace

2.1 Download the source code from the gazebo_ros_pkgs Github repository:
cd PATH_TO_YOUR_WORKSPACE/src (e.g. *cd ~/jackal_ws/src*)
git clone https://github.com/jackal/jackal.git
git clone https://github.com/jackal/jackal_simulator.git
git clone https://github.com/jackal/jackal_desktop.git
git clone https://github.com/ros-visualization/interactive_marker_twist_server.git

2.2 You can automatically install the missing depencies using rosdep via debian install:
rosdep install --from-paths . --ignore-src --rosdistro kinetic -y

3. Install sensor simulation package

When only installing the Gazebo source code for the Jackal robot, various kinds of sensor will not get published. Therefore, it is good to install the hector_gazebo package (http://rosindex.github.io/p/hector_gazebo_plugins/ and https://github.com/tu-darmstadt-ros-pkg/hector_gazebo):
cd PATH_TO_YOUR_WORKSPACE/src (e.g. *cd ~/jackal_ws/src*)
git clone https://github.com/tu-darmstadt-ros-pkg/hector_gazebo.git

TODO: maybe add alternative: gazebo_ros_pkgs with gazebo_ros_imu.cpp (https://github.com/ros-simulation/gazebo_ros_pkgs/blob/kinetic-devel/gazebo_plugins/src/gazebo_ros_imu.cpp) → to be tested!

4. Build and source

cd PATH_TO_YOUR_WORKSPACE/ (e.g. *~/jackal_ws/*)
catkin_make
source devel/setup.bash

5. Test whether the Jackal simulation in Gazebo is working properly

roslaunch jackal_gazebo jackal_world.launch

MATLAB R2019a TODO: extend + test this!

See https://help.ubuntu.com/community/MATLAB.
See https://nl.mathworks.com/videos/how-to-install-matlab-1525083586145_b.html?adobe_mc_ref=https%3A%2F%2Fwww.google.com%2F.
See https://www.youtube.com/watch?v=X_PNVr01F8k.

1. Download MATLAB R2019a

1.1 Go to https://nl.mathworks.com/downloads/.
1.2 Click “Click here” below the “Get Latest Release” box
1.3 Log in with your Mathworks account
1.4 Select R2019a below “Download earlier release”
1.5 Select Linux
1.6 Select “Save file”

Your MATLAB R2019a download file will be written to the Downloads folder.

2. Unzip the downloaded file

2.1 Right-click on downloaded directory and select “Extract here”
2.2 Right-click on unzipped directory and select “Open in terminal”

3. Install MATLAB R2019a

3.1 Run the installer as super user:
sudo ./install

3.2 Now the MathWorks Installer should pop up
3.3 Select “Log in with a MathWorks Account” and click “Next”
3.4 Select “Yes” to accept the terms of the license agreement and click “Next”
3.5 Select “Log in to your MathWorks Account, fill in your Email Address and Password and click “Next”
3.6 Select “Select a license”, select your license to be used and click “Next”
3.7 Choose “/usr/local/MATLAB/R2019a” and click “Next”
3.8 Select the desired products you want to install and click “Next”
3.9 Select “Create symbolic links to MATLAB scripts in:”, check whether the links are created in the “/usr/local/bin” directory and click “Next”
3.10 Check the license, installation folder and products and click “Install”

4. Activate MATLAB

4.1 After installation, select “Activate MATLAB” and click “Next”
4.2 In the “Activate MathWorks Software” window, click “Next”
4.3 Fill in your Ubuntu username (which will thereafter only be allowed to use MATLAB) and click “Next”
4.4 Check your license number, email address and username under “Confirm selection:” and click “Confirm”
4.5 In the “Activation is complete” window, click “Finish”

6. Create a MATLAB launcher (recommended) TODO

7. Change initial working folder (recommended) TODO

8. Install new toolboxes (if needed) TODO

9. Read out custom ROS messages
NOTE: each time you change the content of your custom ROS messages, you should repeat this procedure.

9.1 Generate the ROS messages of all your packages in the workspace
*rosgenmsg(“PATH_TO_YOUR_WORKSPACE/src”) (*e.g. *rosgenmsg("~/jackal_ws/src")*)

If this command gives warnings, delete the previously generated matlab_gen directory (present in *PATH_TO_YOUR_WORKSPACE/src) and open the javaclasslist.txt?? TODO*