



Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences



Project SetUp Instruction Manual

Collision Monitoring for a Mobile Manipulator

Urvashi Negi

Zain Ul Haq

Sreenivasa Hikkal Venugopala

Supervised by

Djordje Vukcevic

March 8, 2021

Following are the prerequisites (tested) for installing the collision monitoring library.

- Linux Ubuntu OS (16.04, 18.04)
- ROS (Kinetic, Melodic)

1 Installation of Dependency Packages from Source

1. Install CMAKE ($\geq 3.18.4$)
 - (a) Download CMAKE source code (cmake-*.tar.gz)
 - (b) Extract the compressed files.
 - (c) Open a terminal and navigate into cmake folder by using 'cd' command.
 - (d) Create a folder named 'build' and navigate into into it (mkdir build and cd build).
 - (e) Execute the command './bootstrap'.
 - (f) Execute the command 'make'.
 - (g) Execute the command 'sudo make install'.
2. Install EIGEN ($\geq 3.3.7$)
 - (a) Download EIGEN source code (eigen-*.tar.gz)
 - (b) Extract the compressed files.
 - (c) Open a terminal and navigate into eigen folder by using 'cd' command.
 - (d) Create a folder named 'build' and navigate into into it (mkdir build and cd build).
 - (e) Execute the command 'cmake ..'.
 - (f) Execute the command 'sudo make install'.
3. Install KDL
 - (a) Clone from the github page ([Link](#))

- (b) Open a terminal and navigate into 'orocos_kinematics_dynamics/orocos_kdl' folder by using 'cd' command.
- (c) Create a folder named 'build' and navigate into into it (mkdir build and cd build).
- (d) Execute the command 'cmake ..'.
- (e) Execute the command 'make'.
- (f) Execute the command 'sudo make install'.

4. Install KDL Parser

- (a) Clone the source code branch based on your ROS from the github page (Link)
- (b) Open a terminal and navigate into 'kdl_parser/kdl_parser' folder by using 'cd' command.
- (c) Create a folder named 'build' and navigate into into it (mkdir build and cd build).
- (d) Execute the command 'cmake ..'.
- (e) Execute the command 'make'.
- (f) Execute the command 'sudo make install'.

5. Install ROS Kortex

- (a) Clone the source code from the github page (Link)
- (b) Follow the instructions provided under 'build' section on github page.

2 Instructions for Compiling the Collision Monitoring Library

1. Clone the source code from the project github page (Link).
2. Compile 'Narko_description' available in 'catkin_workspace/src' folder by following below instructions.

- (a) Navigate into the folder 'Narko_description' available in 'catkin_workspace/src'.
 - (b) Create a folder named 'build' and navigate into it (mkdir build and cd build).
 - (c) Execute the command 'cmake ..'.
 - (d) Execute the command 'make'.
 - (e) Execute the command 'sudo make install'.
3. Compile the Collision Monitoring Library.
- (a) Navigate into 'sdp_ws20_collision_monitoring_for_robotic_manipulators' folder.
 - (b) Create a folder named 'build' and navigate into it (mkdir build and cd build).
 - (c) Execute the command 'cmake ..'.
 - (d) Execute the command 'make'.
4. Compile the Library and ROS packages for simulation and testing.
- (a) Navigate into 'catkin_workspace' folder.
 - (b) Execute the command 'catkin_make'. If 'catkin_make' does not work, then execute 'catkin build'.
 - (c) Execute the command 'source devel/setup.bash'.
5. For visualization of the output launch the launch files present under folders '/kinova_arm/launch' for visualizing arms and for visualizing the mobile base and arms use 'narko_kinova_base_collision/launch', using the command 'roslaunch *.launch'

In case if you come across an error as stated in below image, execute the following command 'chmod +x marker.py' and recompile the catkin workspace as given under Section 2 Point 4.

Collision Monitoring for a Mobile Manipulator

```
auto-starting new master
process[master]: started with pid [7634]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to fadf97d4-7ea7-11eb-81cb-74e5f9699849
process[rosout-1]: started with pid [7647]
started core service [/rosout]
process[gazebo-2]: started with pid [7662]
process[spawn_urdf-3]: started with pid [7676]
process[arm_controller-4]: started with pid [7677]
process[base_controller-5]: started with pid [7678]
process[arm_simulator-6]: started with pid [7684]
process[robot_state_publisher-7]: started with pid [7698]
process[custom_rviz-8]: started with pid [7712]
ERROR: cannot launch node of type [narko_kinova_base_collision/marker.py]: can't locate node [marker.py] in package [narko_kinova_base_collision]
process[world_frame_publisher-10]: started with pid [7719]
[ INFO] [1615054550.842662685]: Finished loading Gazebo ROS API Plugin.
[ INFO] [1615054550.843229999]: waitForService: Service [/gazebo/set_physics_properties] has not been advertised, waiting...
[ INFO] [1615054551.336698203]: waitForService: Service [/gazebo/set_physics_properties] is now available.
[ INFO] [1615054551.400412217, 0.041000000]: Physics dynamic reconfigure ready.
[ INFO] [1615054551.799967297, 0.149000000]: PlanarMovePlugin missing <enableYAxis>, defaults to "1"
[ INFO] [1615054551.813582154, 0.149000000]: Starting plugin DiffDrive(ns = //)
[ WARN] [1615054551.813686657, 0.149000000]: DiffDrive(ns = //): missing <rosDebugLevel> default is na
[ INFO] [1615054551.815800305, 0.149000000]: DiffDrive(ns = //): <tf_prefix> =
[ WARN] [1615054551.816083838, 0.149000000]: DiffDrive(ns = //): missing <publishWheelTF> default is false
```

Figure 1: error