Motion Generator & GAZEBO & Action Engine

LEE GEON HEE
gunhee6392@gmail.com

Manual List :

1) ROS Package
2) Docker
3) Jeckins
4) Robocare data

1. ROS Package.

Package name :

1. kist_ws(in NAS)
2. action_engine(in git hub)

Packages lists(kist_ws):

Silbor3_2dnav
Silbot3_description
Silbot3_feasible_moveit
Silbot3_gazebo
Silbot3_motion_generation
Silbot3_msgs
Silbot3_omniwheels(incomplete)
Silbot3_slam
Silbot3_silbot3_teleoperation
Silbot3_tutorial
Silbot3_xmlparsing

### highlight is important function and used frequently.

Description(kist_ws):

1. Silbor3_2dnav

This is a package for implementing Navigation function.
It is basically used to using move_base function, therefore launch files and yaml files is described according to default format.

2. Silbot3_description

It is a package for silbot3 to have its modeling data which include stl file with mesh and URDF file(XACRO) describing relationship between joint and link.
It also includes launch file to run in the GAZEBO which means 3D virtual environment.

3. Silbot3_feasible_moveit

It is used to utilizing MoveIt API, but now it isn’t used .

4. Silbot3_gazebo

It is a package to deploying various functions in the GAZEBO.
For using ros_control, It is added to yaml file regarding controller(ex. Position, trajectory controller) and launch file to run.

5. Silbot3_motion_generation

It is used to making motion generation program based on MoveIt!, but it is considerably transformed for implementing motion generation program and linked silbot3_xmlparsing package.

6. Silbot3_msgs

It is a package provided by the Robocare.
It included msg and service file for working Silbot3.

7. Silbot3_omniwheels(incomplete)

It is a package for other researcher to implement holonomic constraint which means omni-wheels mobility here and added for working navigation functions in the GAZEBO.

8. Silbot3_slam

It is a package for working mapping and map server.

9. Silbot3_teleoperation

It is a package for looking movement of Silbot3 in the GAZEBO and means “Teleoperation” function using Keyboard.

10. Silbot3_tutorial

It is a package for working silbot3 in the Robocare.
It is written by Python and C(++) according to various function, for example LED and movement… , It is just added to take possibility for working Silbot3 in real.

11. Silbot3_xmlparsing

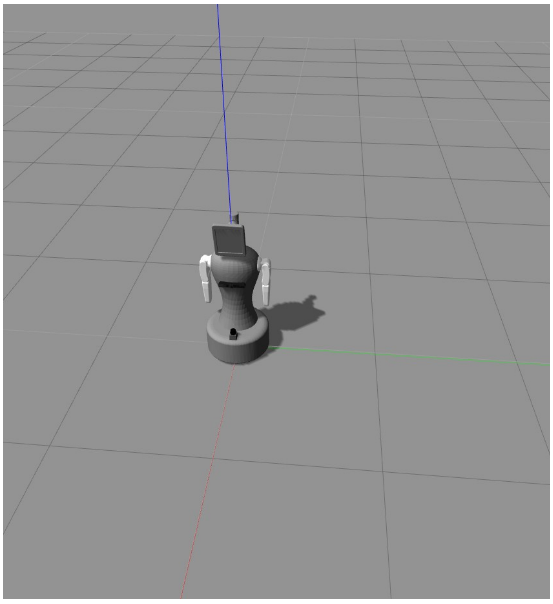
It is a package which has Tinyxml library which helps to make xml files and makes xml files for working Silbot3.
Using Xml files, you can work Silbot3 in real and GAZEBO.

How to launch each function.

1. GAZEBO.

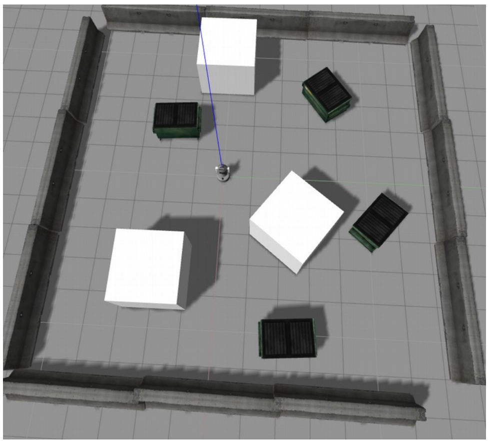
Load the model of Silbot3 in the GAZEBO.

```
$ roslauch silbot3_description only_silbot.launch
```



Load the model of Silbot3 with obstacles in the GAZEBO.

```
$ roslaunch silbot3_description silbot3_collision_xacro.launch
```



Load the model of Silbot3 with obstacle in the GAZEBO and controller.

Once you load the model,

```
$ roslaunch silbot3_description silbot3_collision_xacro.launch
```

if you want to implement position controller, you can write

```
$ roslaunch silbot3_gazebo silbot_arm_position_controller.launch
```

### if you want to check xml file, you should load the position controller.

if you want to implement position controller, you can write

```
$ roslaunch silbot3_gazebo silbot_arm_trajectory_controller.launch
```

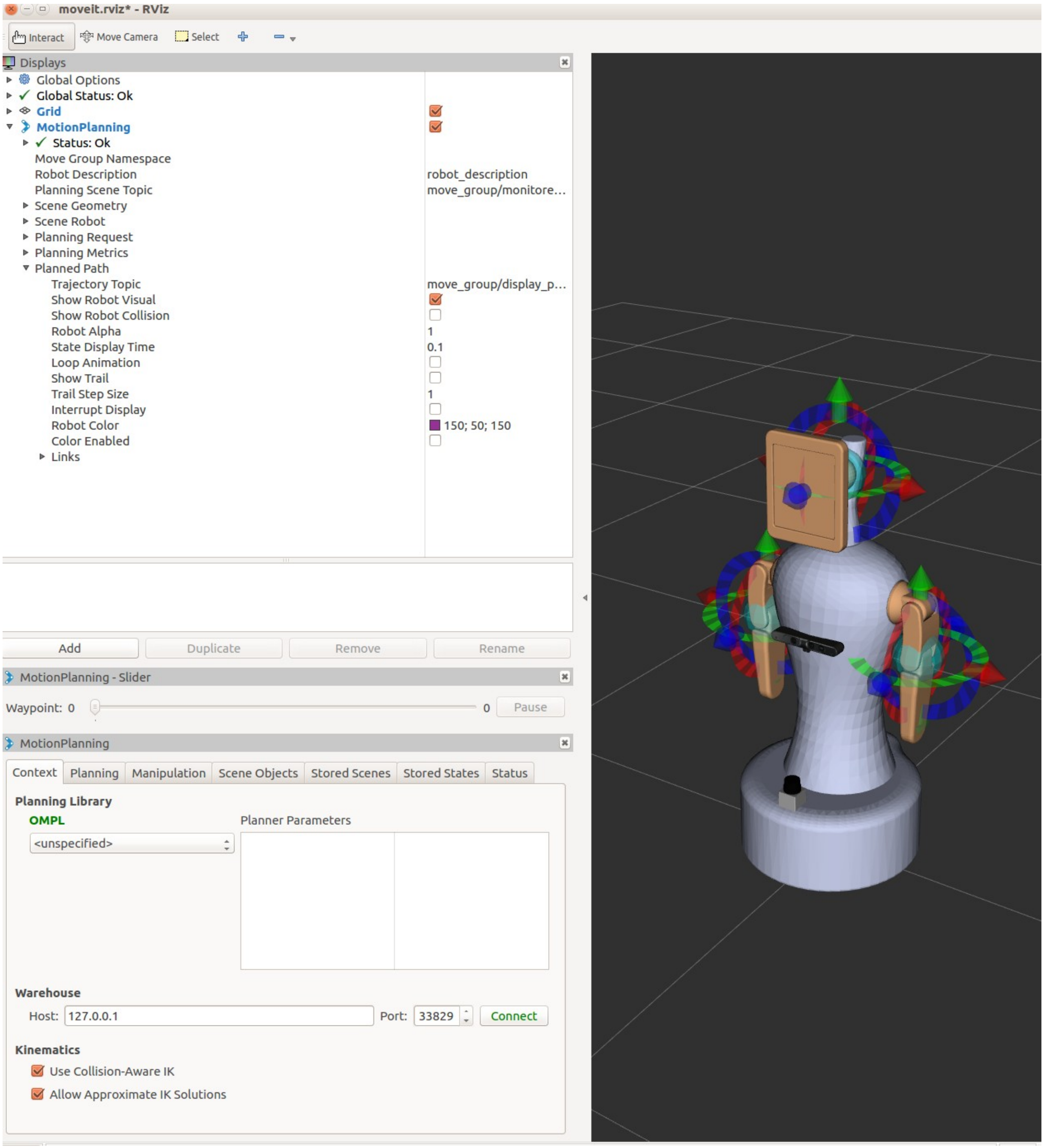
2. Motion Generation program.

It can make xml file in the Rviz using Silbot3.

First, you can see this GUI when you command as:

```
roslaunch silbot3_motion_generation motionGenerationProgram.launch
```

### Please check "Allow Approximate IK Solution" checkbox in left-bottom since it can give to control arm of silbot3 easily.



Displays
- Global Options
- Global Status: Ok
- Grid ☑
- MotionPlanning ☑
  - Status: Ok
  - Move Group Namespace
  - Robot Description: robot_description
  - Planning Scene Topic: move_group/monitore…
  - Scene Geometry
  - Scene Robot
  - Planning Request
  - Planning Metrics
  - Planned Path
    - Trajectory Topic: move_group/display_p…
    - Show Robot Visual ☑
    - Show Robot Collision ☐
    - Robot Alpha: 1
    - State Display Time: 0.1
    - Loop Animation ☐
    - Show Trail ☐
    - Trail Step Size: 1
    - Interrupt Display ☐
    - Robot Color: 150; 50; 150
    - Color Enabled ☐
    - Links

MotionPlanning - Slider

Waypoint: 0 0

MotionPlanning

Planning Library

OMPL <unspecified>

Planner Parameters

Warehouse

Host: 127.0.0.1 Port: 33829

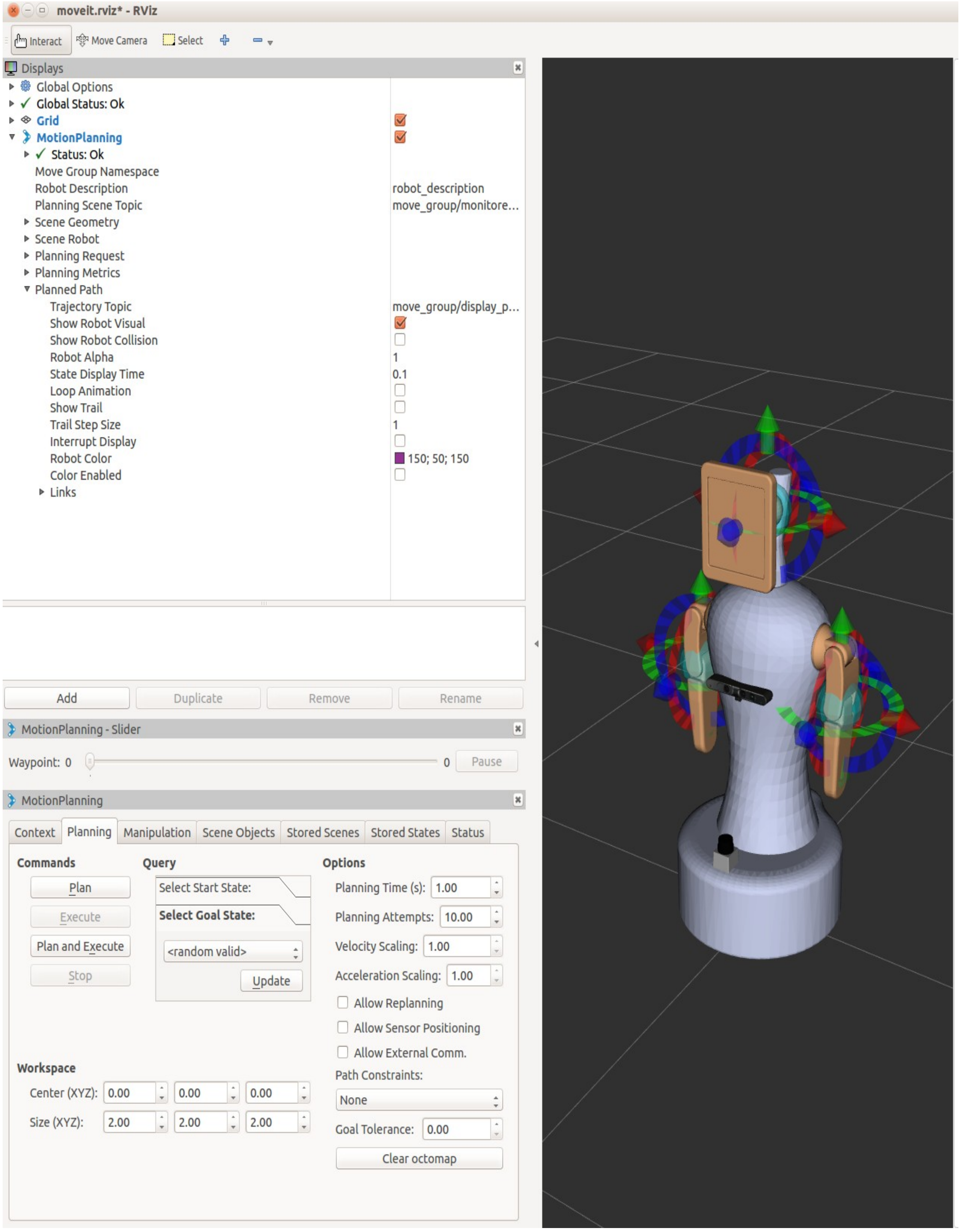
Kinematics

☑ Use Collision-Aware IK

☑ Allow Approximate IK Solutions

Second, You can choice Planning tab in left-bottom for making xml file
In this case, you can control silbot3 using interactive markers

When you click the “Plan” button, you can see trajectory to which arm and head of silbot3 move according.
### Its process isn’t affect making a xml file.
When you click the “Execute” or “Plan and Execute” button, making a xml file immediately works according to the trajectory of “Plan” and depends on previous operation.



Global Options
✓ Global Status: Ok
Grid ☑
MotionPlanning ☑
✓ Status: Ok
Move Group Namespace
Robot Description: robot_description
Planning Scene Topic: move_group/monitore...
Scene Geometry
Scene Robot
Planning Request
Planning Metrics
Planned Path
Trajectory Topic: move_group/display_p...
Show Robot Visual: ☑
Show Robot Collision: □
Robot Alpha: 1
State Display Time: 0.1
Loop Animation: □
Show Trail: □
Trail Step Size: 1
Interrupt Display: □
Robot Color: 150; 50; 150
Color Enabled: □
Links

Waypoint: 0 0

Select Start State:
Select Goal State:
<random valid>

Planning Time (s): 1.00
Planning Attempts: 10.00
Velocity Scaling: 1.00
Acceleration Scaling: 1.00
□ Allow Replanning
□ Allow Sensor Positioning
□ Allow External Comm.
Path Constraints: None
Goal Tolerance: 0.00

Workspace
Center (XYZ): 0.00 0.00 0.00
Size (XYZ): 2.00 2.00 2.00

Finally, you can modulate many parameters and determine file name and directory.

INITIALIZATION_FROM_YAML param confirms whether it is initialized using yaml file.

FILE_NAME param is file name which you want to take.

FILE_DIRECTORY is saved directory of file which you want to take.

rate param is the rate of /joint_state topic.

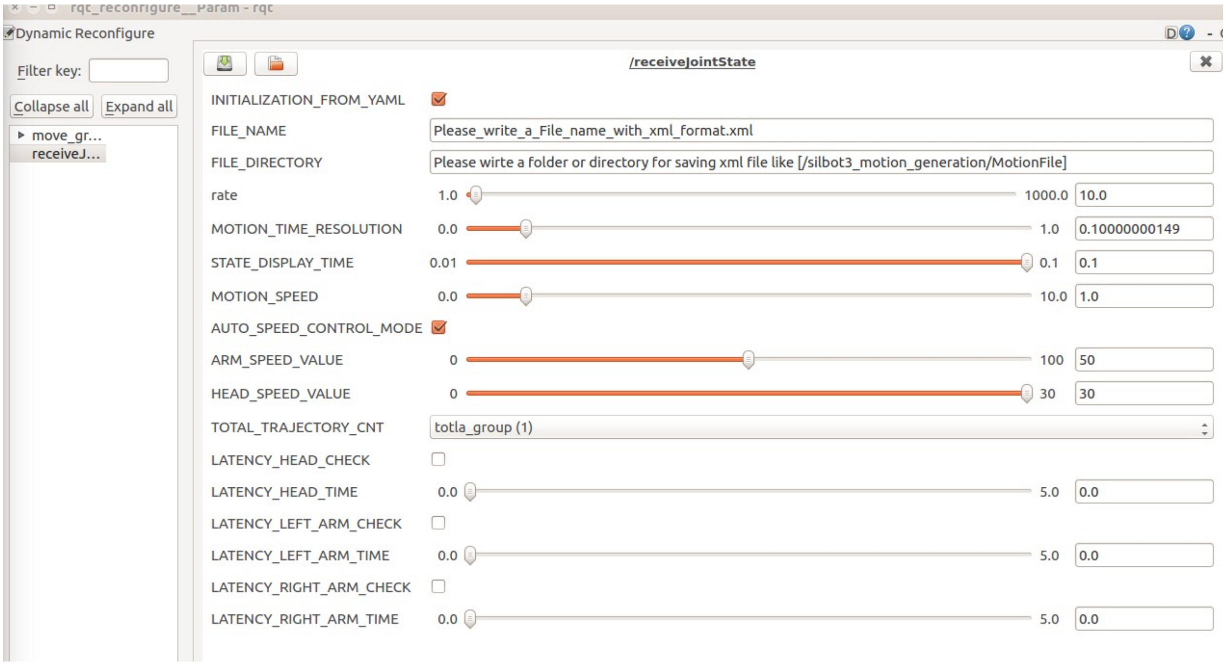
MOTION_TIME_RESOLUTION param is value of time gap of xml file.

AUTO_SPEED_CONTROL_MODE param is autonomous mode, if you select it, it autonomously decide param of speed value.

ARM_SPEED_VALUE, HEAD_SPEED_VALUE param is value to be described with speed value in xml file.

TOTAL_TRAJECTORY_CNT param is count how to work frequently, but it don’t need to control.

LATENCY params provide latency function, it can be selected with group and should have amount of latency time in order to work



INITIALIZATION_FROM_YAML	☑		
FILE_NAME	Please_write_a_File_name_with_xml_format.xml		
FILE_DIRECTORY	Please wirte a folder or directory for saving xml file like [/silbot3_motion_generation/MotionFile]		
rate	1.0	1000.0	10.0
MOTION_TIME_RESOLUTION	0.0	1.0	0.10000000149
STATE_DISPLAY_TIME	0.01	0.1	0.1
MOTION_SPEED	0.0	10.0	1.0
AUTO_SPEED_CONTROL_MODE	☑		
ARM_SPEED_VALUE	0	100	50
HEAD_SPEED_VALUE	0	30	30
TOTAL_TRAJECTORY_CNT	totla_group (1)		
LATENCY_HEAD_CHECK	□		
LATENCY_HEAD_TIME	0.0	5.0	0.0
LATENCY_LEFT_ARM_CHECK	□		
LATENCY_LEFT_ARM_TIME	0.0	5.0	0.0
LATENCY_RIGHT_ARM_CHECK	□		
LATENCY_RIGHT_ARM_TIME	0.0	5.0	0.0

3. Playing the created motion in GAZEBO.

In order to play motion of slilbot3 with xml file in GAZEBO, you will mostly use the silbot3_xmlparsing package.
Once you finish previous task such as loading model in GAZEBO and working position controller, you can check playing motion of Silbot3.

You should parse the xml file.
${prefix} : the directory of xml file you want
($(arg file name)) : the file name of xml file you want

```
rosrun silbot3_xmlparsing silbot3_xmlparsing ${prefix}/($(arg file name))
```

You can see motion of Silbot3.

```
rosrun silbot3_xmlparsing trXml2Cmd
```

2. Docker.

3. Jeckins.

4. Robocare data

saved in the NAS of Cjsteam
(smb://161.122.114.157/cjsteam/Users/이건희/2018 New Silbot File)