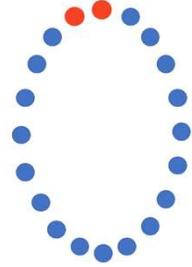




SEMTRON

ZERO

Robotics Development Framework, Workspace & Workflow



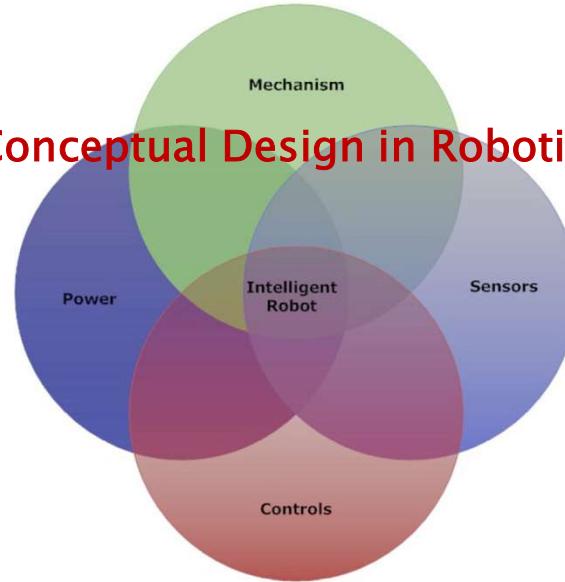
<https://tinyurl.com/5b5s9ham>

1



SEMTRON

Conceptual Design in Robotics



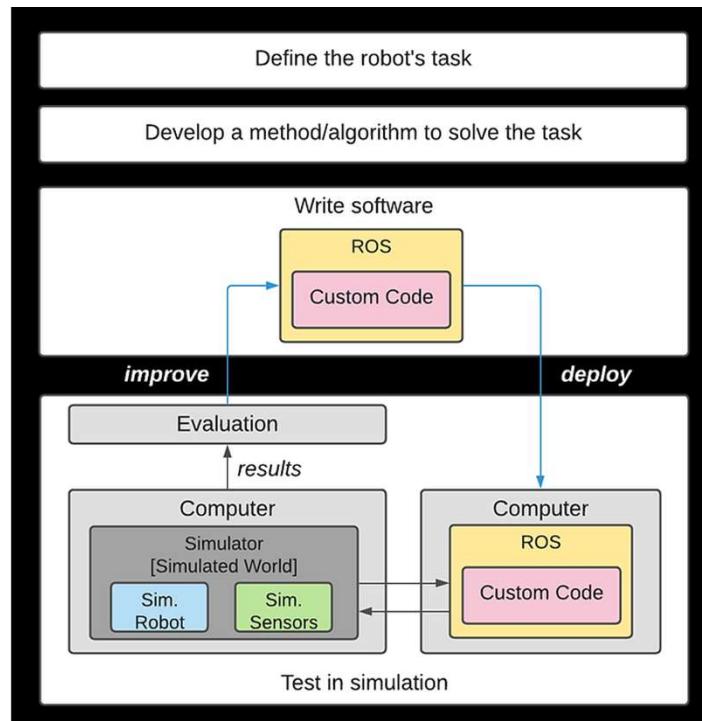
2



The requirements for a robot that collects table tennis balls:

- (a) It should be applicable to the setting of general table tennis courts, with a size of 8 m × 16 m;
- (b) It can collect at least 30 table tennis balls within a reasonable time as well as show the number of table tennis balls collected;
- (c) When no one is inside the table tennis court, it can be activated to start the process for collecting table tennis balls automatically within the designated area;
- (d) There should a basket to hold the collected table tennis balls and allow players to take out the balls easily to continue practising.

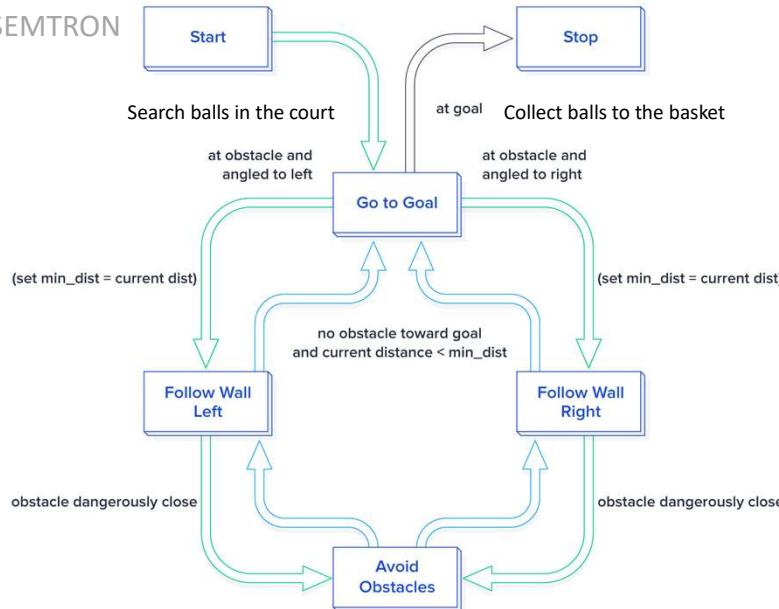
3



4



SEMTRON

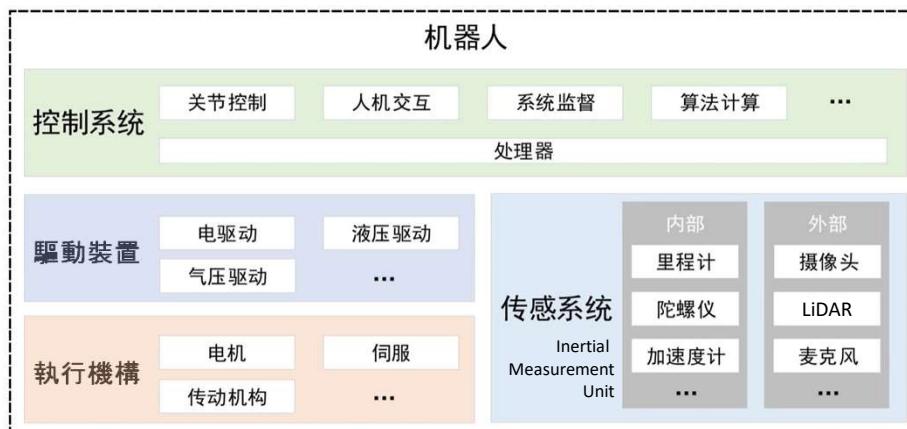


Define robot's tasks

5

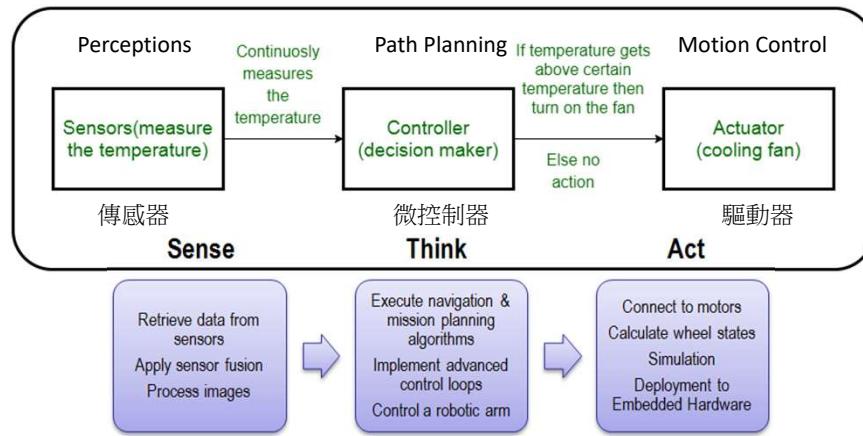


SEMTRON



Define robot's functions & behaviour

6

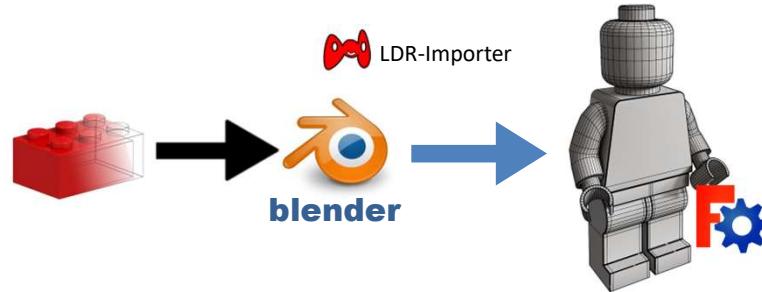


Robotics in physical computing

7

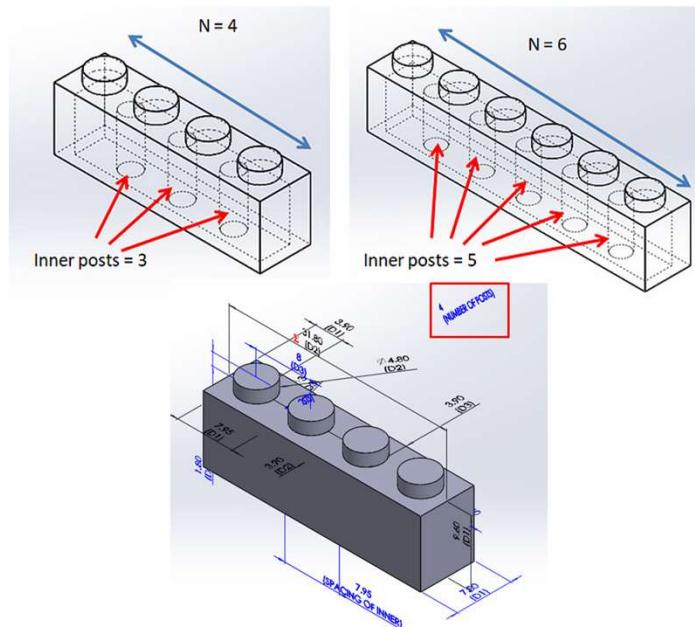


8



Parametric modeling

9

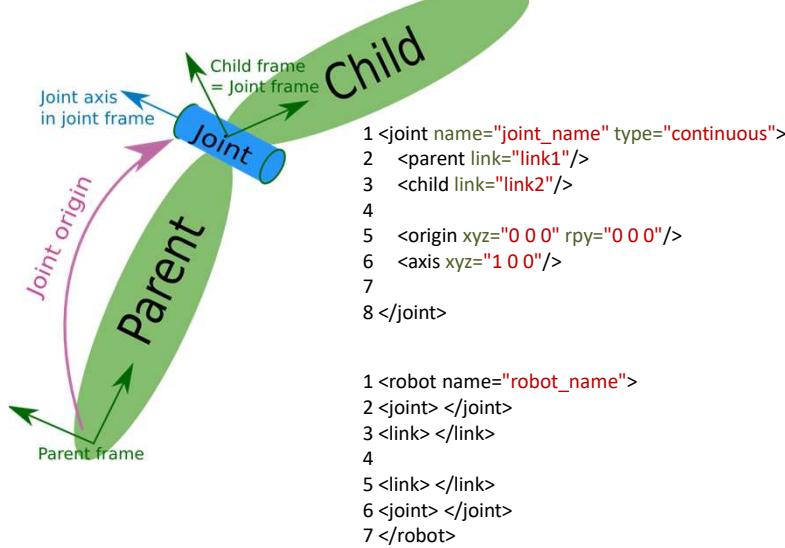


Parametric modeling

10



SEMTRON



URDF modeling

11

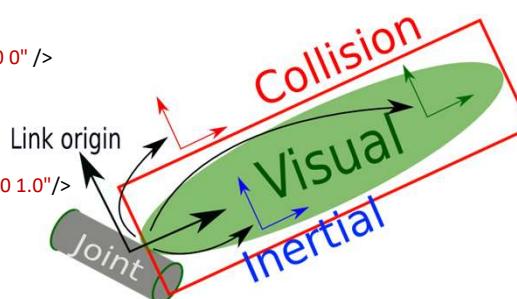
```

1 <link name="link_name">
2   <inertial>
3     <origin xyz="0 0 0.5" rpy="0 0 0" />
4     <mass value="1" />
5     <inertia ixx="100" ixy="0" ixz="0"
6       iyy="100" iyz="0"
7       izz="100" />
8   </inertial>
9
10  <visual>
11    <origin xyz="0 0 0" rpy="0 0 0" />
12    <geometry>
13      <box size="1 1 1" />
14    </geometry>
15    <material name="Cyan">
16      <color rgba="0 1 0 1.0" />
17    </material>
18  </visual>
19  <collision>
20    <origin xyz="0 0 0" rpy="0 0 0" />
21    <geometry>
22      <cylinder radius="1" length="0.5" />
23    </geometry>
24 </link>

```

$$\begin{pmatrix} \mathbf{ixx} & \mathbf{ixy} & \mathbf{ixz} \\ \mathbf{iyx} & \mathbf{iyy} & \mathbf{iyz} \\ \mathbf{izx} & \mathbf{izy} & \mathbf{izz} \end{pmatrix}$$

Inertia Tensor

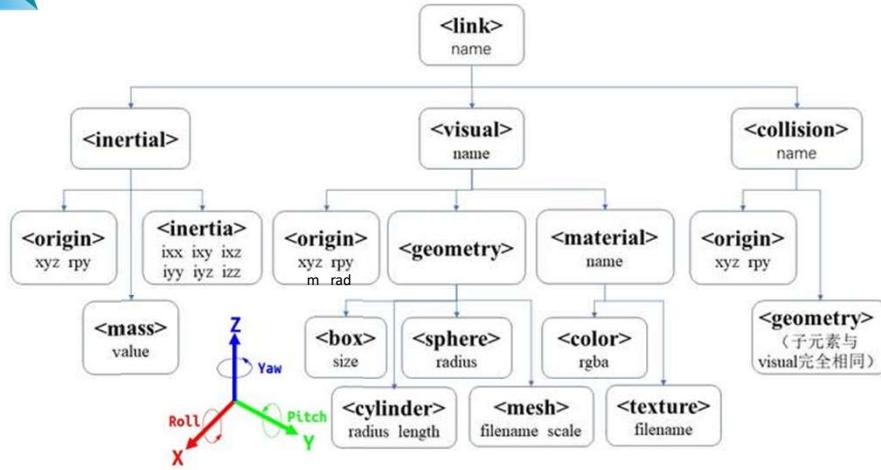


URDF modeling

12

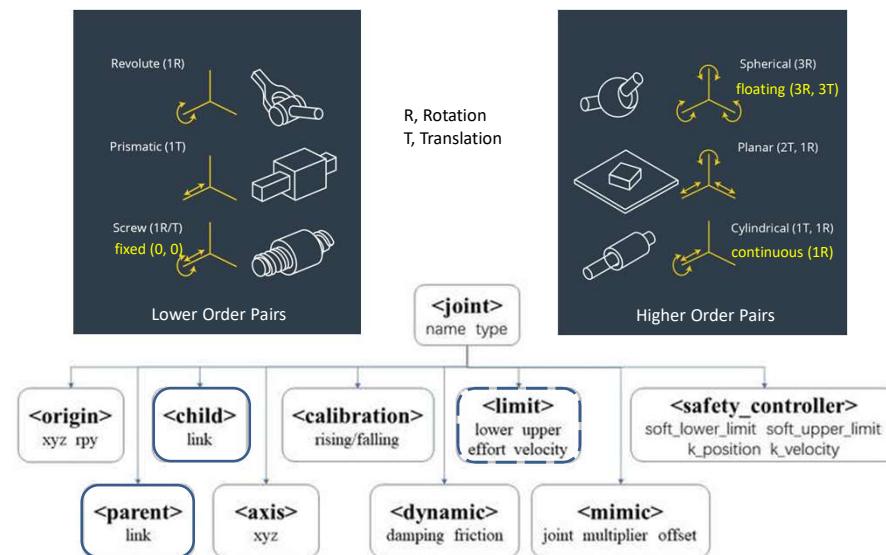


SEMTRO



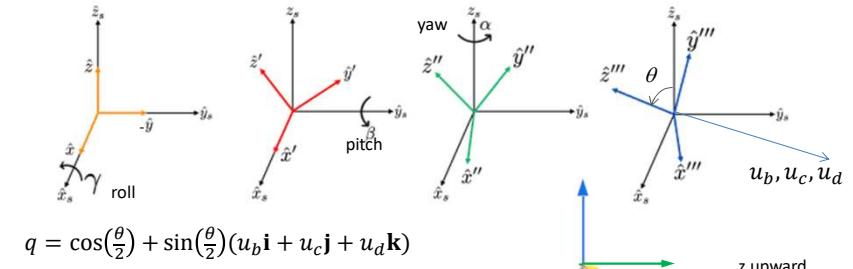
URDF modeling

13



URDF modeling

14



where θ is the angle of rotation
and $[u_b, u_c, u_d]$ is the axis of rotation

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = \begin{pmatrix} \frac{q_1}{\sin \frac{\theta}{2}} \\ \frac{q_2}{\sin \frac{\theta}{2}} \\ \frac{q_3}{\sin \frac{\theta}{2}} \end{pmatrix}$$

$$\theta = 2 \cos^{-1} q_0$$

ROS frame orientation - quaternion

15

<https://articulatedrobotics.xyz/category/concept-design>



16

- ❑ Most robotics work are in teams include one who develops a CAD model of robot.
- ❑ Instead of crafting an URDF by hand it is possible to export an URDF model.
- ❑ Below is a list of available URDF exporters for a variety of CAD and 3D modeling software systems.

CAD Exporters

- [Fusion 360 URDF Exporter](#)
- [SolidWorks URDF Exporter](#)
- [ROS Workbench for FreeCAD](#)
- [OnShape URDF Exporter](#)
- [CREO Parametric URDF Exporter](#)

Other URDF Export Tools

- [Copellia Sim URDF Exporter](#)
- [Isaac Sim URDF Exporter](#)
- [Blender URDF Exporter](#)
- [Gazebo SDFFormat to URDF Parser](#)
- [SDF to URDF Converter in Python](#)
- The [Blender Robotics Tools](#) includes a tool to export [URDF files from Blender](#)

URDF from CAD models

17

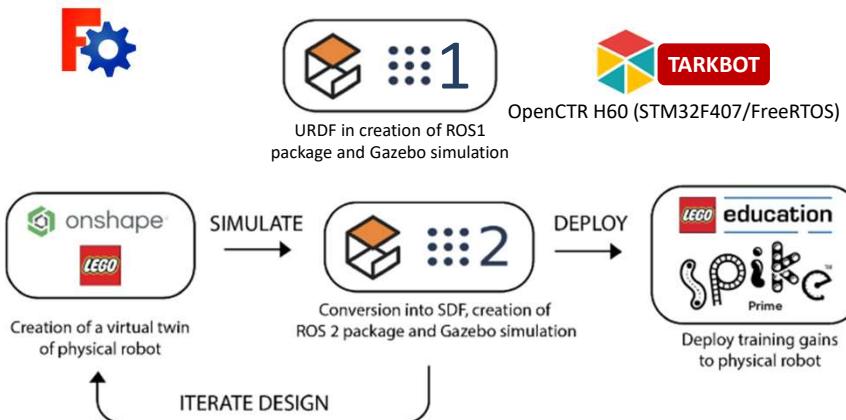


SEMTRON

Common Software

Common Hardware

https://download2.portableapps.com/portableapps/FreeCADPortable/FreeCADPortable_0.21.2.paf.exe



URDF from CAD models

18

Mode	Pan	Rotate	Zoom	Select
CAD (default)	or	or	or	 Multi select +CTRL
Touchpad	Hold Shift + drag	 Alt +	/	
Blender	Hold Shift +			

FreeCAD mouse navigation

19



SEMTRON

<https://github.com/USeebi/Lesson0><https://github.com/drfenixion/freecad.overcross/wiki>

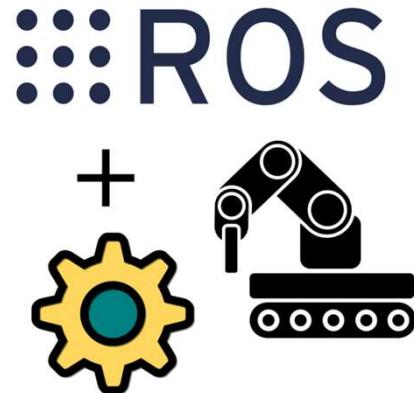
- **Ctrl** + **+** and **Ctrl** + **-** to zoom in and out, respectively.
- The arrow keys, **<** **>** **▲** **▼**, to shift the view left/right and up/down
- **Shift** + **<** and **Shift** + **>** to rotate the view by 90 degrees
- The numeric keys, **0** **1** **2** **3** **4** **5** **6**, for the seven standard views:
 Isometric, Front, Top, Right, Rear, Bottom, and Left
- **V** **O** will set the camera in Orthographic view.
- While **V** **P** sets it in Perspective view.
- **Ctrl** will allow you to select more than one object or element

FreeCAD keyboard navigation

20



Design Workspace in Robotics



21



- CALENDAR

Timeliness be aware of the due dates and complete the tasks in accordance with the schedule stipulated by their teachers.
- PIE CHART

Good Planning break an extended task into smaller sub-parts and complete them in stages.
- COMPUTER

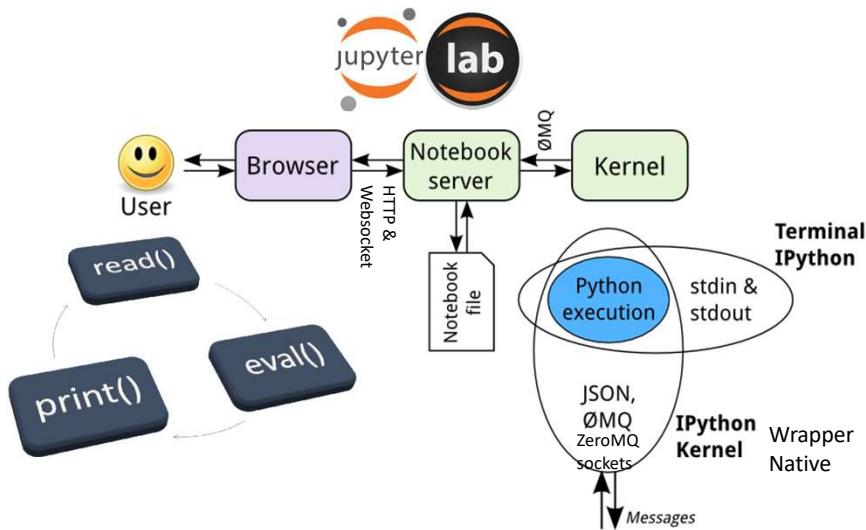
Good Record Keeping keep a complete record of their work, including drafts and backup copies of computer files.
- DRAFT

Originality ensure the work they submit is their own, record the sources of information used and acknowledge them properly in their work.

22



SEMTRO



Proposed Python workspace

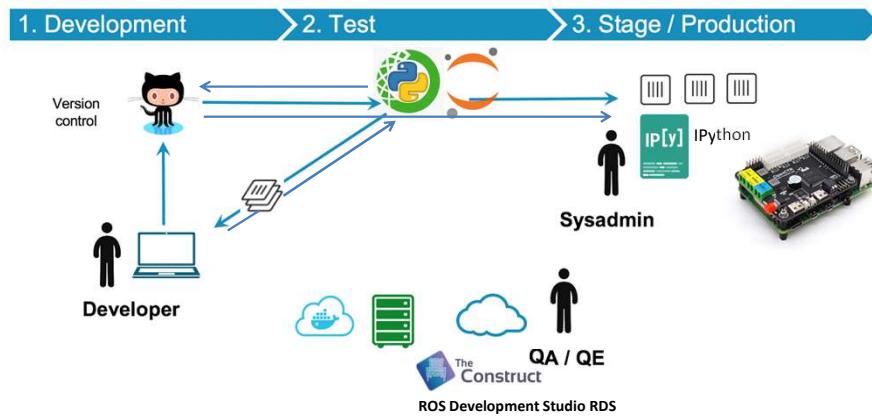
23

<https://robostack.github.io/index.html>

The screenshot shows a Jupyter Notebook interface. On the left, a sidebar displays a file tree with notebooks like 'Lorenz.ipynb', 'Data.ipynb', 'Fasta.ipynb', 'Julia.ipynb', 'Cpp.ipynb', 'audio', 'images', and 'R.ipynb'. The main area shows a code cell starting with `matplotlib inline` and `from ipywidgets import interactive, fixed`. Below the code, a plot titled 'The Lorenz Differential Equations' shows the chaotic attractor. To the right, there's an 'Output View' showing sliders for parameters sigma, beta, and rho, and a 'lorenz.py' code editor with the same code as the cell above. The bottom status bar indicates 'Ln 1, Col 1 Spaces: 4 lorenz.py'.

Proposed Python workspace

24



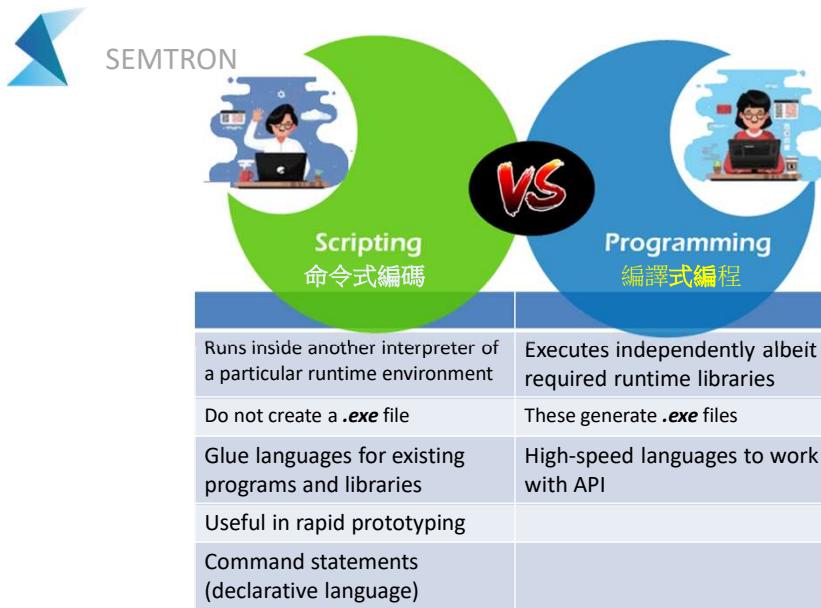
Proposed CI/CD workflow

25



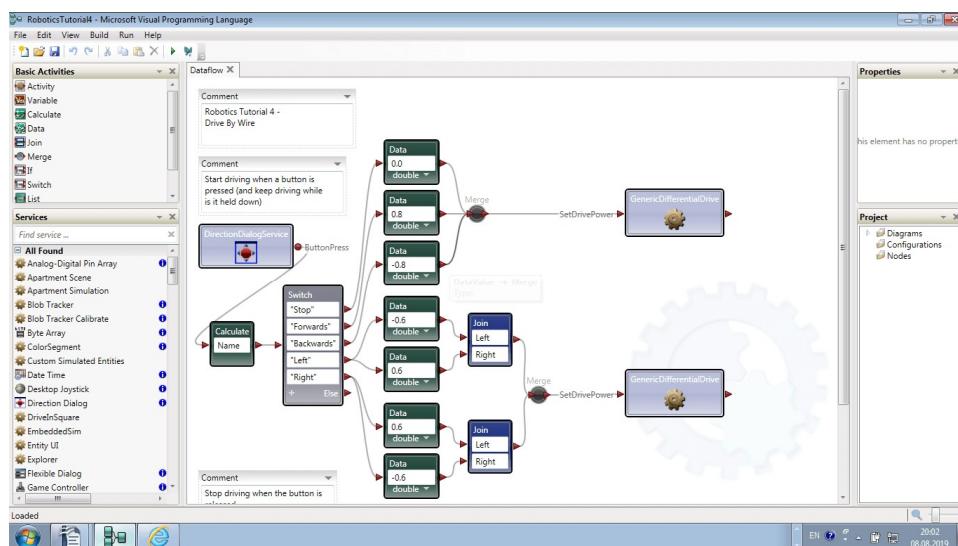
⌘ P #Show all commands	⌘ E #Explorer	⌘ D #Debug	⌘ H #Replace in files
⌘ P #Show files	⌘ B #Toggle sidebar	⌘ Y #Debug console	⌘ F #Find in files
⌘ M #Problems	⌘ U #Output	⌘ X #Extensions	⌃ ⌘ F #Find
VS CODE CHEATSHEET			
⌘ J #Toggle panel	F5 #Start	⌃ F5 #Stop	⌃ F5 #Restart
⌃ F5 #Start without debugging	F9 #Toggle breakpoint	⌃ D #Debug sidebar	F11 #Step into
⌘ K U #Close unmodified	F10 #Step over	⌃ F11 #Step out	⌃ Y #Debug panel
⌘ KW #Close all	⌘ KF #Close folder	⌃ V #Open Markdown prev.	⌃ K Z #Zen mode

26



Python scripting

27

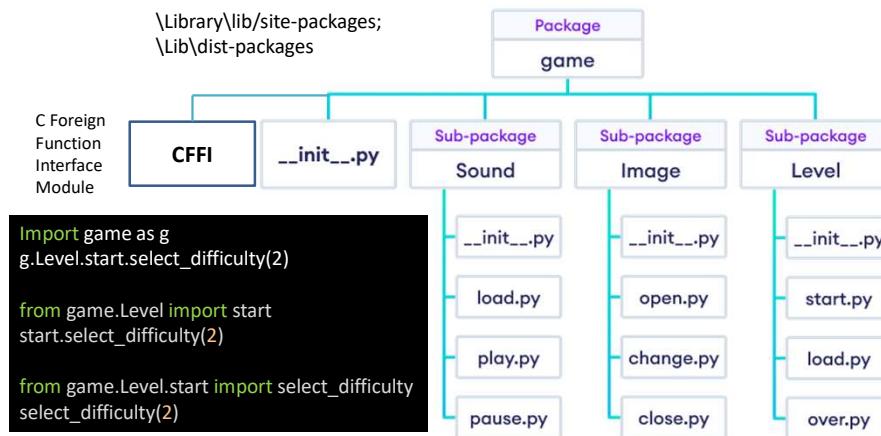


Visual scripting

28



SEMTRO



Packages are a way of structuring Python's module namespace by using "dotted module names".

Python packages

29

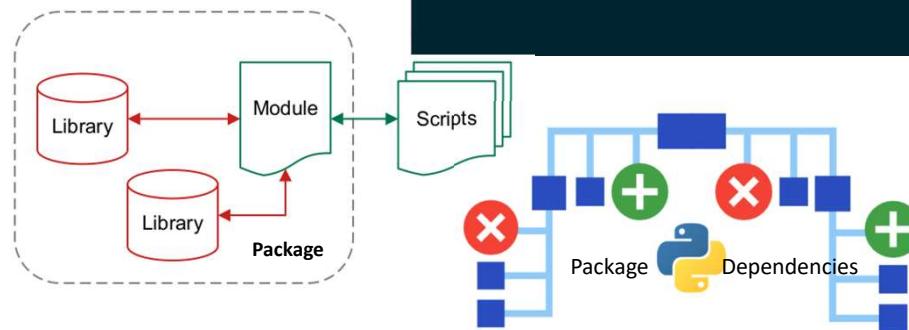


SEMTRO

A module is a file containing Python definitions & statements and use them in a script or in an interactive instance of the interpreter. The file name is the module name with the suffix .py appended.

```

$ cat hello_world.py
#!/bin/env python
print("Hello world!")
$ chmod +x hello_world.py
$ ./hello_world.py
Hello world!
$ 
$ python hello_world.py
Hello world!
$ 
  
```



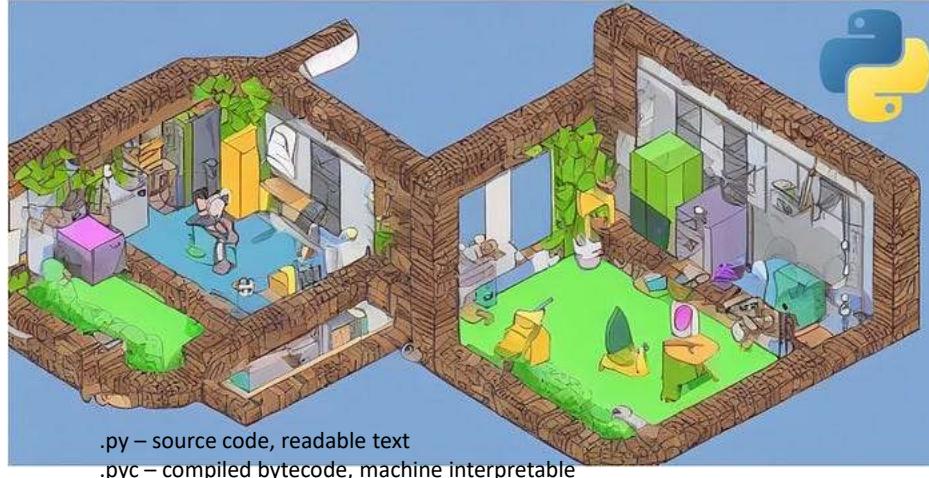
Python packages

30



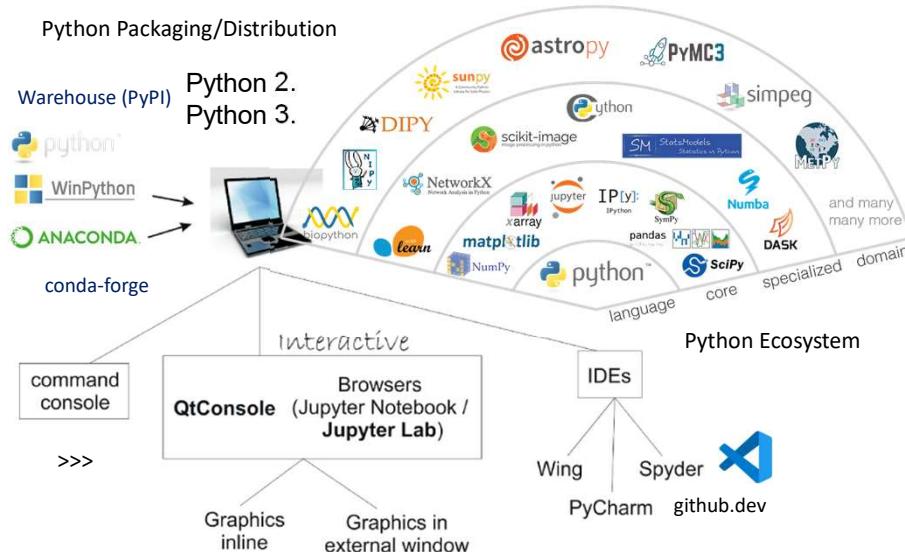
SEMTRO

```
$ conda env export > environment.yml
$ conda env create -f environment.yml
```



Python virtual environment

31



Python robotic development

32

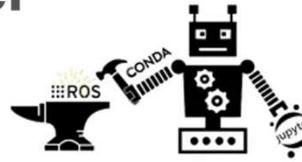


SEMTRO

binder

RoboStack

**Using the Robot Operating System alongside
the Conda and Jupyter Data Science Ecosystems**

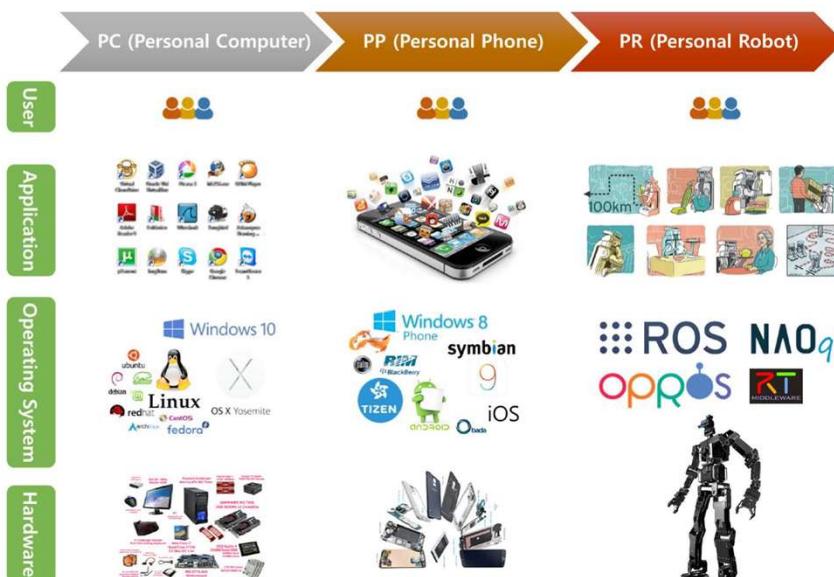


- ✓ ROS on *any* platform: + +
- ✓ Easy installation of ROS side-by-side with PyTorch , Tensorflow & many more
- ✓ No root access required – install ROS on shared workstations & high-performance computers
- ✓ Reproducible environments + reproducible robotics research
- ✓ RViz-like visualizations in the browser *without* a ROS installation
- ✓ Control your robot in Jupyter Notebooks & ROS integration with JupyterLab

0 1 Python 3 | Idle ROS

Python robotic development

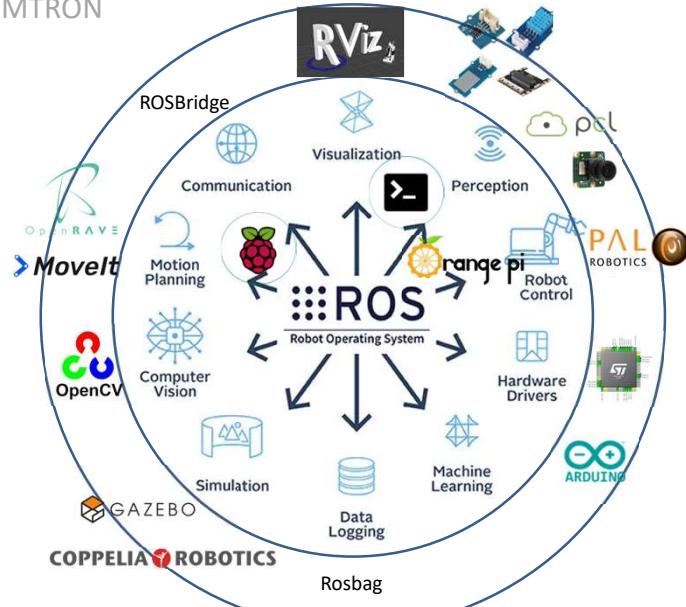
33

**Operating System**

34



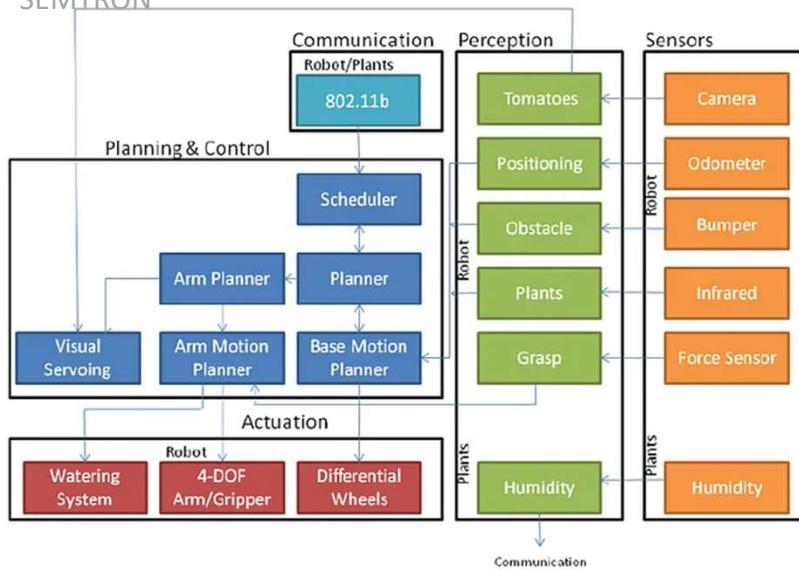
SEMTRON



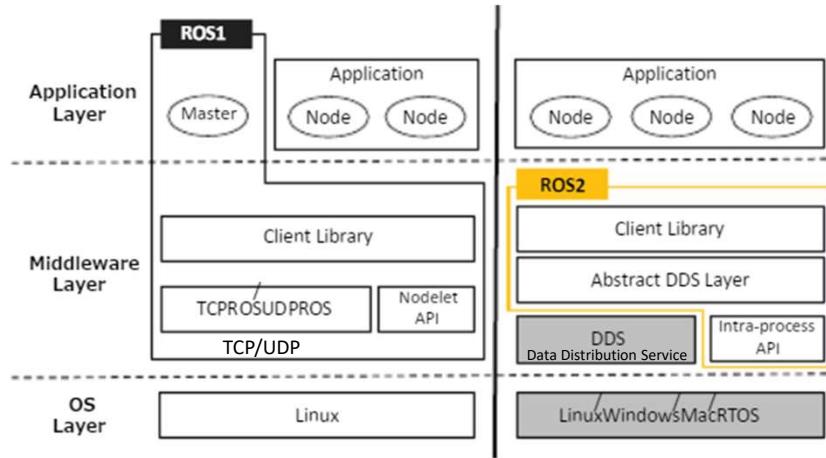
35



SEMTRON

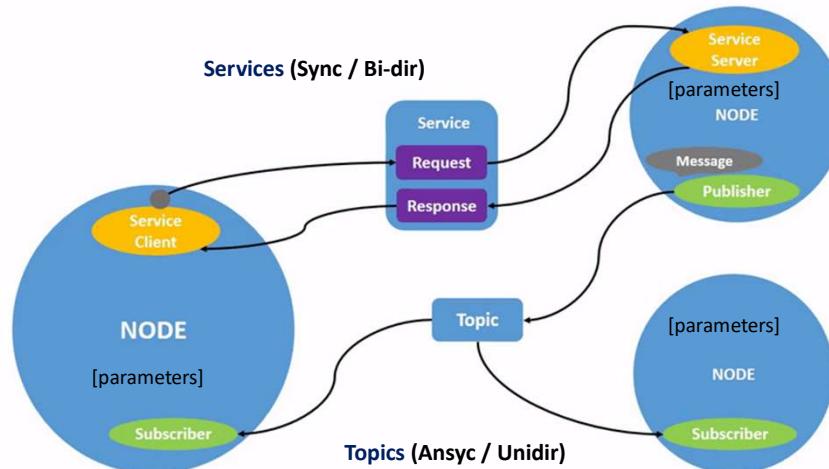


36



Robot Operating System

37

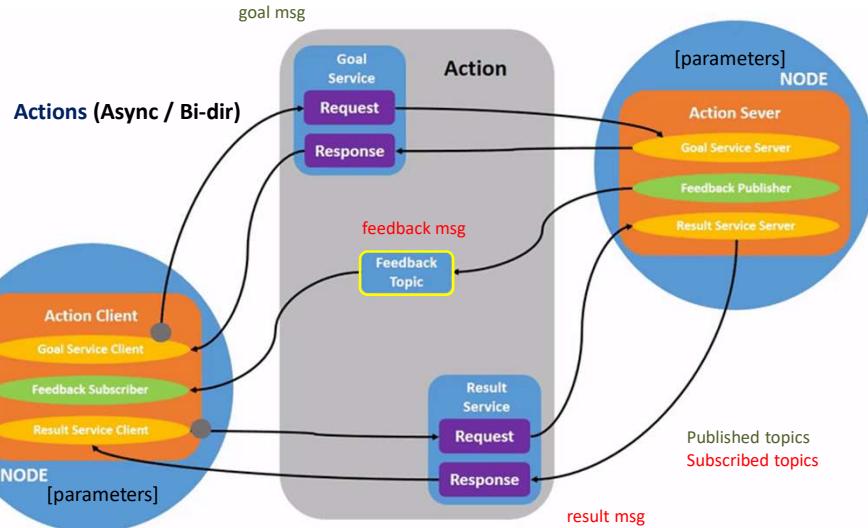


Robot Operating System

38



SEMTRO

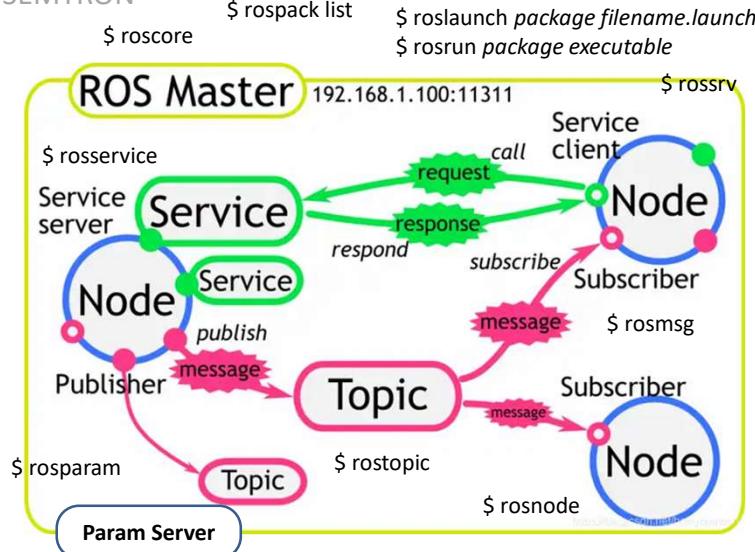


Robot Operating System

39



SEMTRO



Robot Operating System

40



SEMTRO

```

### activate ros_noetic env
$ conda activate K:\Miniconda3\envs\ros_noetic
$ roscore

### launch ros1 cli
$ rostopic list

### launch rviz GUI
$ rosrun rviz rviz

### launch gazebo GUI
$ gazebo --verbose

### launch a demo node talker
$ roscd rospy_tutorials
$ 001_talker_listener/talker

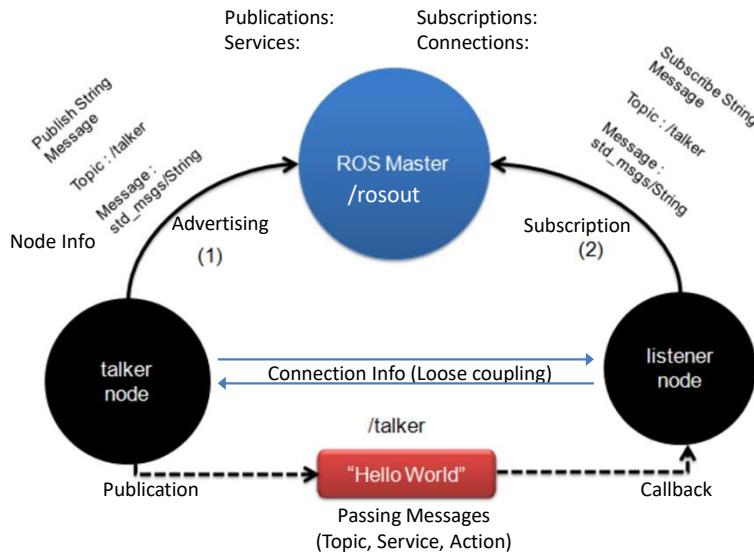
### from another terminal activate the ros_noetic env and launch a
demo listener
$ rosrun rospy_tutorials listener

```

41

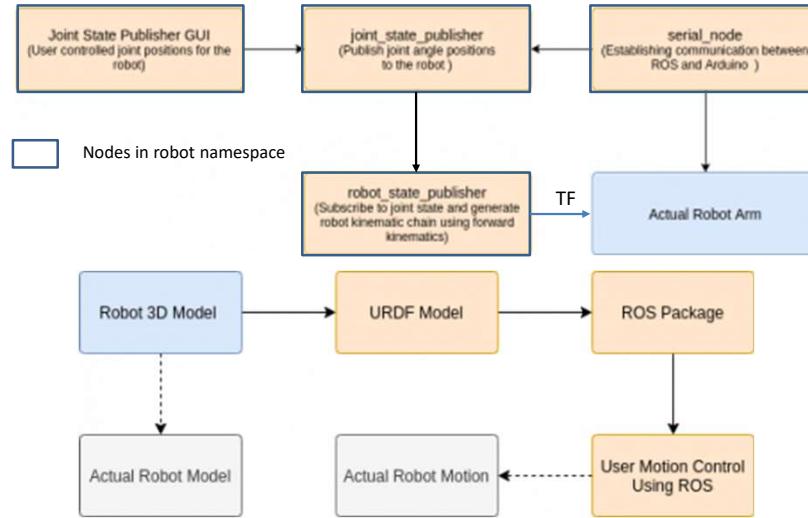


SEMTRO



ROS nodes and messages

42



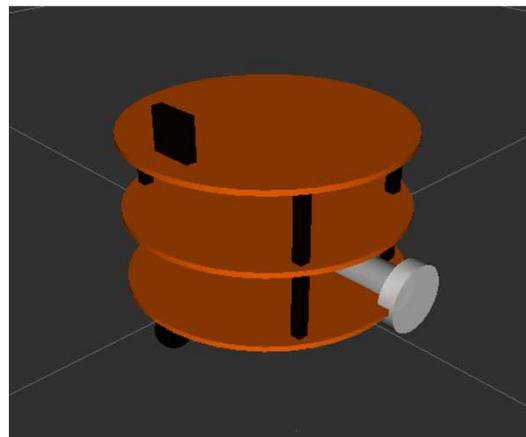
ROS URDF models

43



SEMTRO

ROS机器人开发实践 – 胡春旭
https://github.com/peijian1998/ros_exploring



ROS URDF models

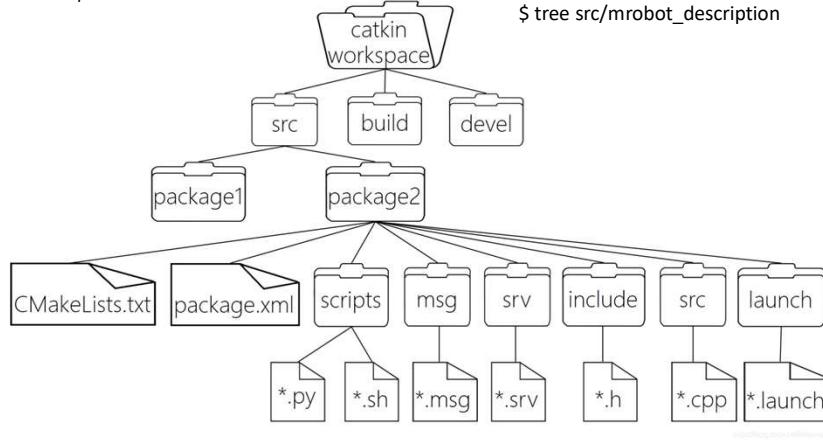
44



SEMTRO

```
$ rospack
$ roscl
```

```
$ cd ~/catkin_ws
$ tree src/mrobot_description
```



ROS applications

45



SEMTRO

SENSE
THINK
ACT
PACKAGE

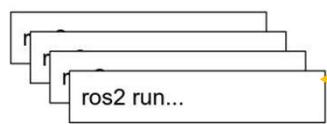
SENSOR NODE

COMPUTE NODE

MOTOR NODE

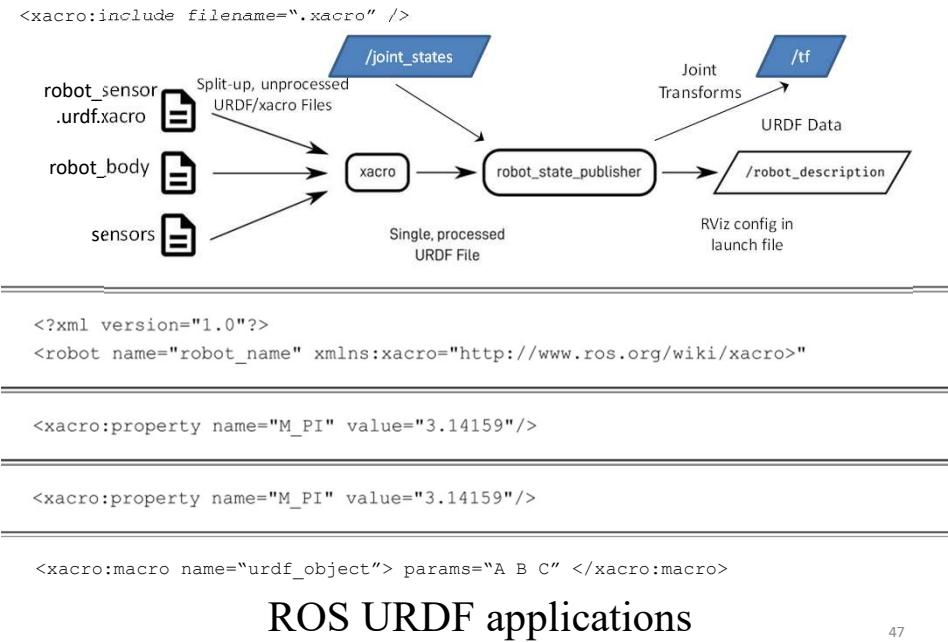
ROBOT NODE

PARAMS PACKAGE



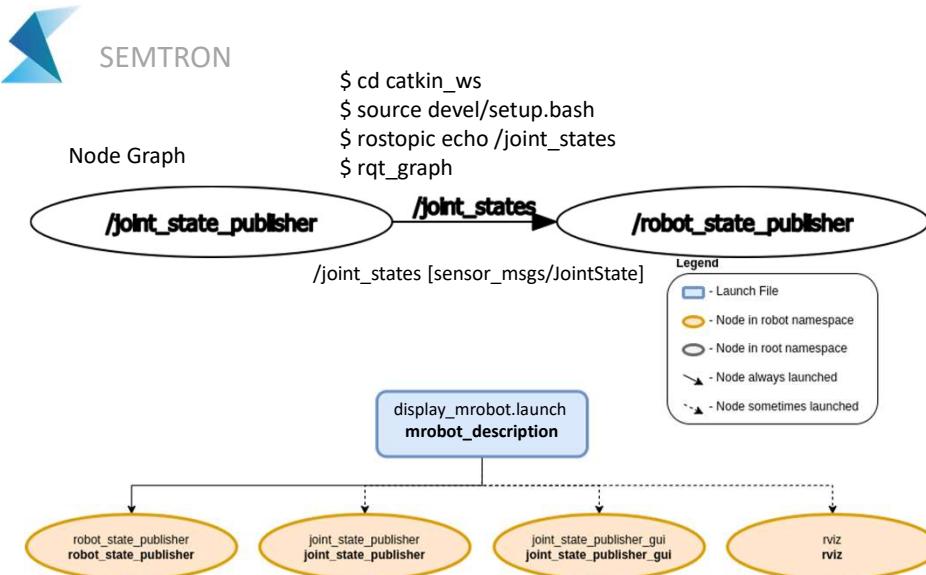
ROS applications

46



ROS URDF applications

47

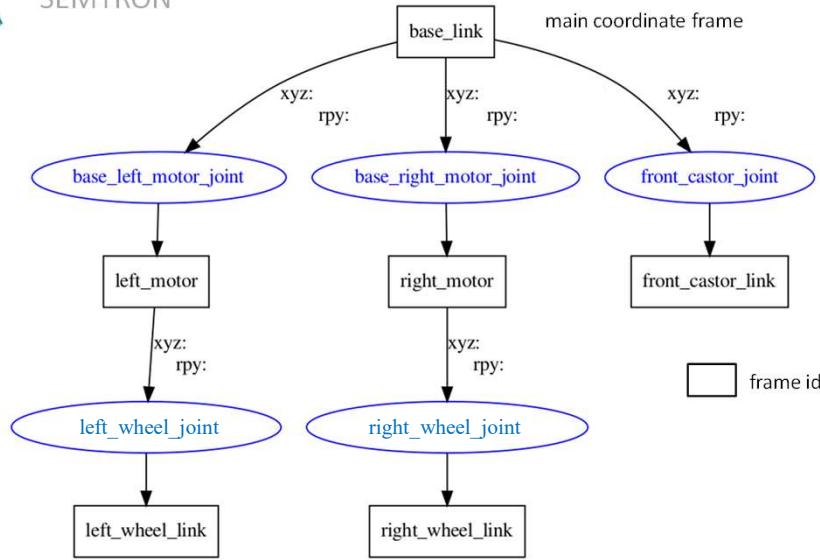


ROS visualizations

48



SEMTRO



ROS URDF models

49

```

<link name="base_link">
  <visual>
    <origin xyz=" 0 0 0" rpy="0 0 0" />
    <geometry>
      <cylinder length="0.005" radius="0.13"/>
    </geometry>
    <material name="yellow">
      <color rgba="1 0.4 0 1"/>
    </material>
  </visual>
  <collision>
    <origin xyz=" 0 0 0" rpy="0 0 0" />
    <geometry>
      <cylinder length="0.005" radius="0.13"/>
    </geometry>
  </collision>
  <inertial>
    <mass value="2" />
    <origin xyz="0 0 0" />
    <inertia ixx="0.01" ixy="0.0" ixz="0.0"
             iyy="0.01" iyz="0.0" izz="0.5" />
  </inertial>
</link>                                         display_mrobot_chassis_withPhy_urdf.launch
<launch>
  <param name="robot_description" textfile="$(find mrobot_description)/urdf/mrobot_chassis_withPhy.urdf" />

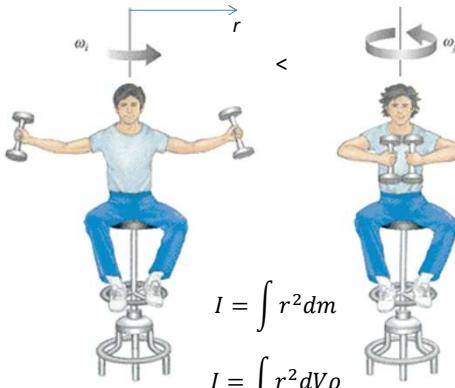
```



50



SEMTRON



$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

$$I_{xx} = \int \rho(\hat{y}^2 + \hat{z}^2)dV$$

$$I_{xy} = \int \rho(\hat{x}\hat{y})dV$$

I_{xx} **moment of inertia** about the x-axis
 I_{xy} **product of inertia** about a pair of given perpendicular axes

center of gravity $I \triangleq \iiint_V r^2 \rho(r) dV$

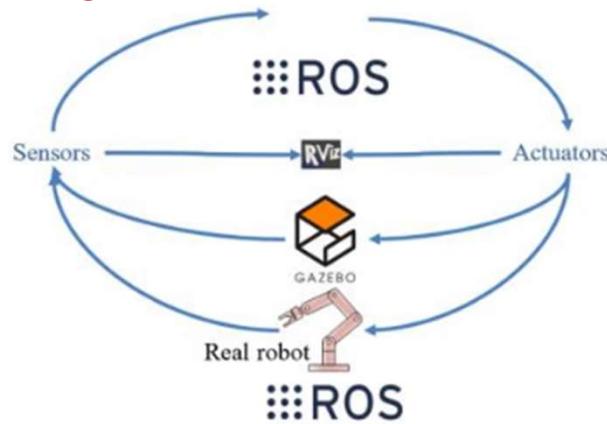
Physical properties in ROS URDF

51



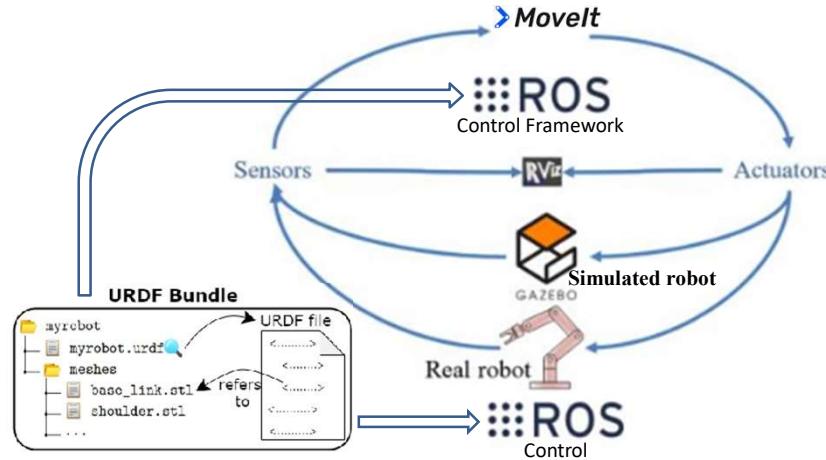
SEMTRON

Design Simulations in Robotics

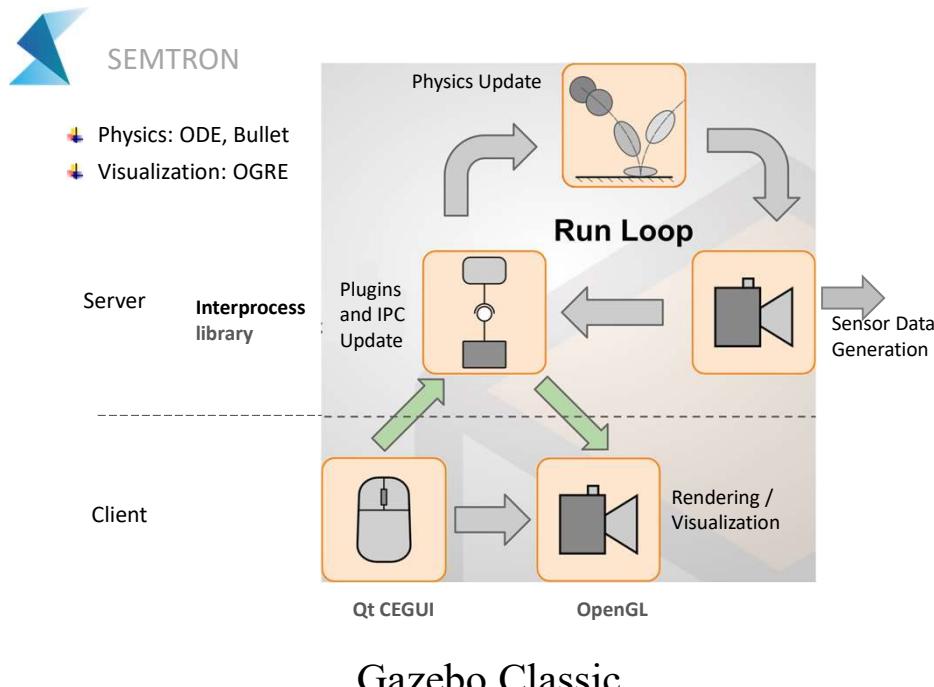


52

“ . . . RViz shows you what the robot thinks is happening, while Gazebo shows you what is really happening.”



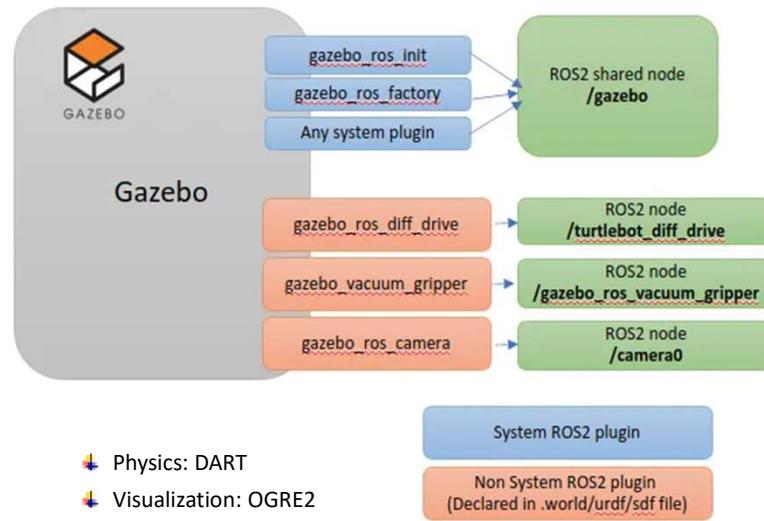
53



54

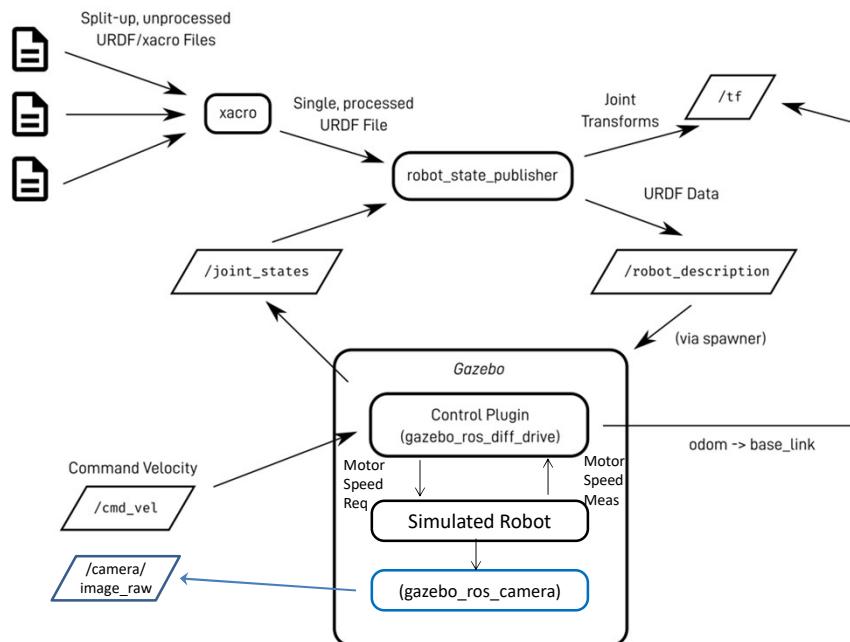


SEMTRO



Gazebo Ignition

55

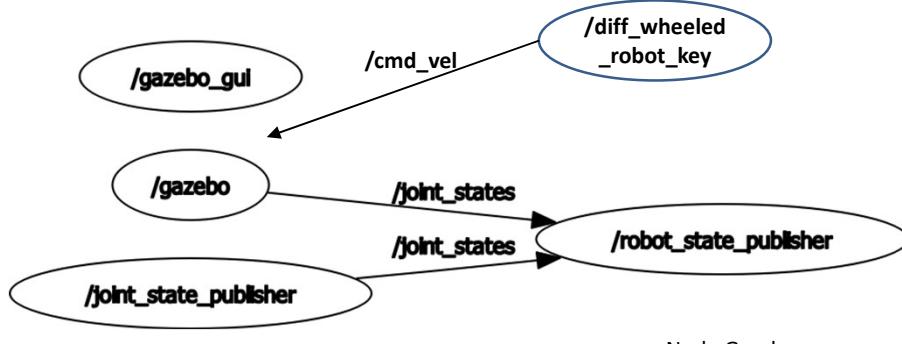


56



SEMTRO

```
$ roslaunch mrobot_gazebo view_mrobot_with_camera_gazebo.launch
$ roslaunch diff_wheeled_robot_control keyboard_teleop.launch
$ rostopic echo /cmd_vel
```



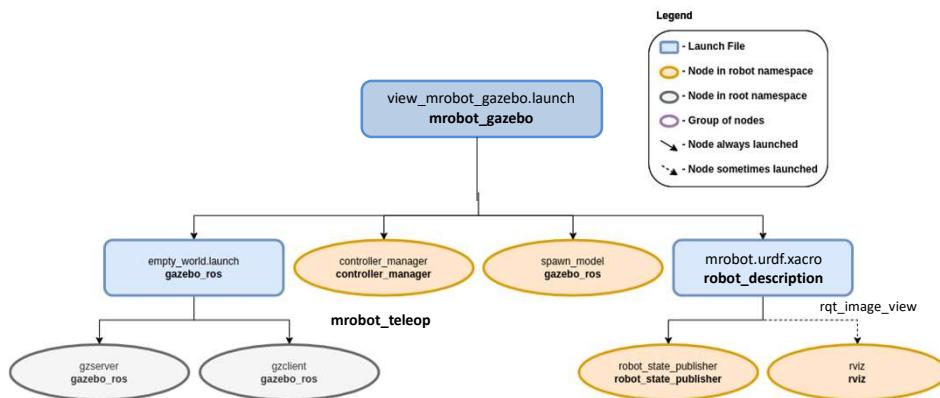
Node Graph

ROS simulation

57



SEMTRO



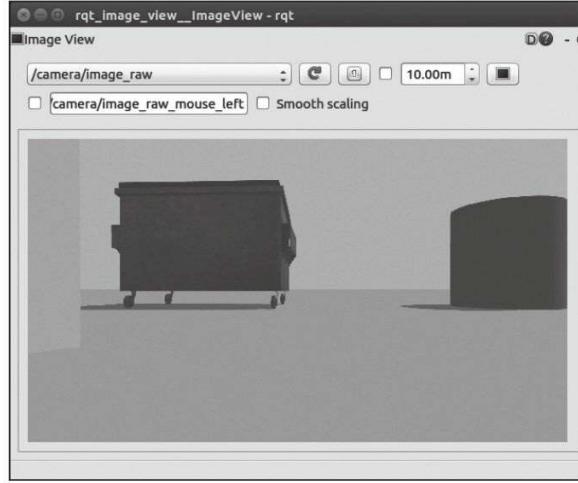
ROS simulation

58



SEMTRO

```
view_mrobot_with_camera_gazebo.launch
<node name="rviz" pkg="rviz" type="rviz" args="-d $(find
mrobot_description)/config/mrobot.rviz" required="true" />
```

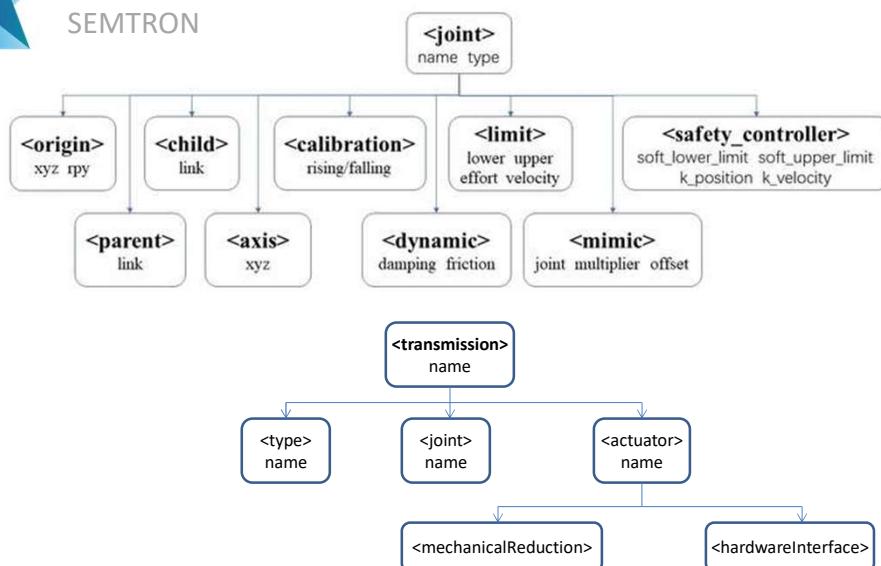


ROS simulation

59

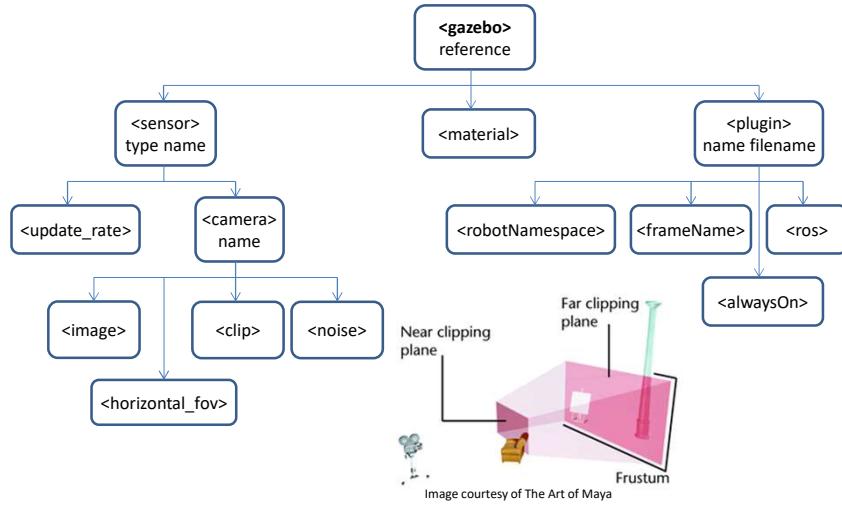


SEMTRO



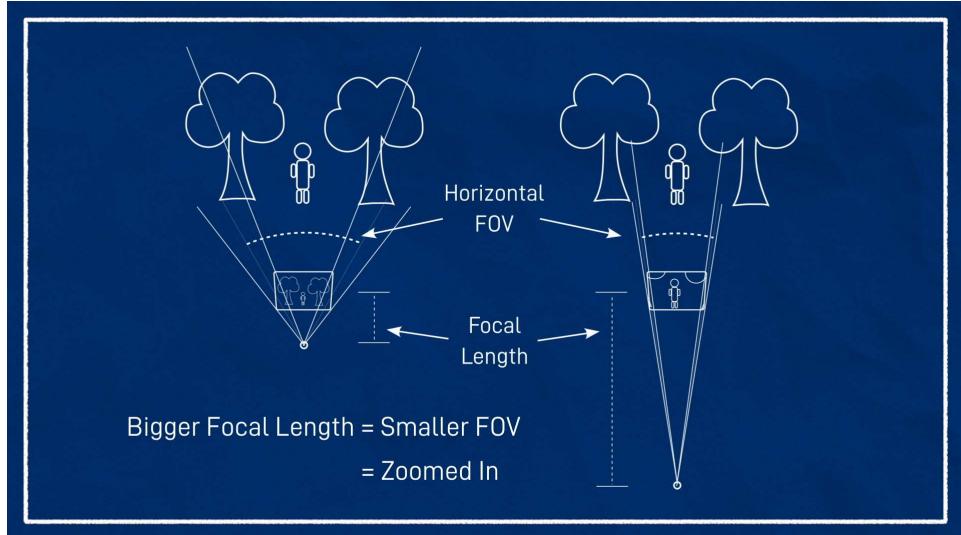
Gazebo simulation

60



Gazebo simulation

61



Gazebo simulation

62



Homework

63





ROS packages Gazebo Plugin Deprecated from simulator_gazebo

65

https://repo.anaconda.com/miniconda/Miniconda3-py311_24.5.0-0-Windows-x86_64.exe

```
>Downloads\Miniconda3-py311_24.5.0-0-Windows-x86_64.exe
/InstallationType=JustMe /AddToPath=0 /RegisterPython=0 /NoRegistry=1 /S
/D=K:\Miniconda3\py311
```

```
(base) > conda install mamba -c conda-forge
(base) > conda install git
(base) > mamba create -p K:\Miniconda3\envs\ros_noetic python=3.11
(base) > mamba activate K:\Miniconda3\envs\ros_noetic
(ros_noetic)
(ros_noetic) > conda config --env --add channels conda-forge
(ros_noetic) > conda config --env --add channels robostack-staging
(ros_noetic) > conda config --env --remove channels defaults
(ros_noetic) > mamba install ros-noetic-desktop ros-noetic-gazebo-ros
(ros_noetic) > mamba install ros-noetic-gazebo-plugins ros-noetic-gazebo-ros-pkgs ros-noetic-
gazebo-ros-control ros-noetic-rqt-controller-manager ros-noetic-xacro ros-
noetic-move-base
(ros_noetic) > mamba deactivate
(base) > mamba activate K:\Miniconda3\envs\ros_noetic
```

66

```
$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py
```

