SEMTRON

ZERO

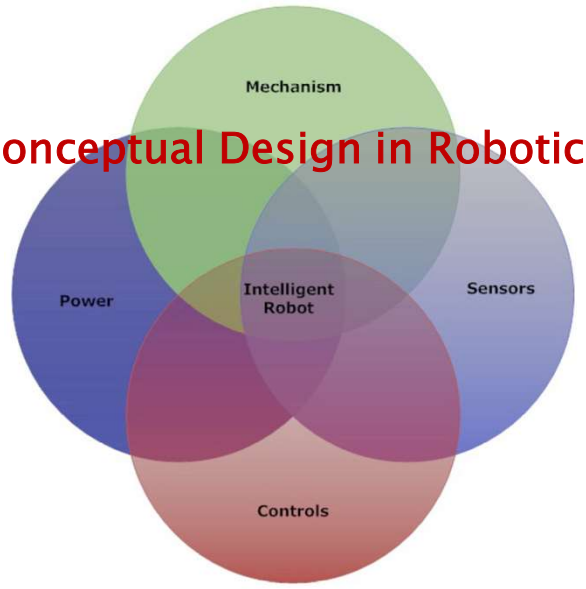
Robotics Development Framework, Workspace & Workflow

https://tinyurl.com/5b5s9ham

1

SEMTRON

Conceptual Design in Robotics



2

SEMTRON

The requirements for a robot that collects table tennis balls:

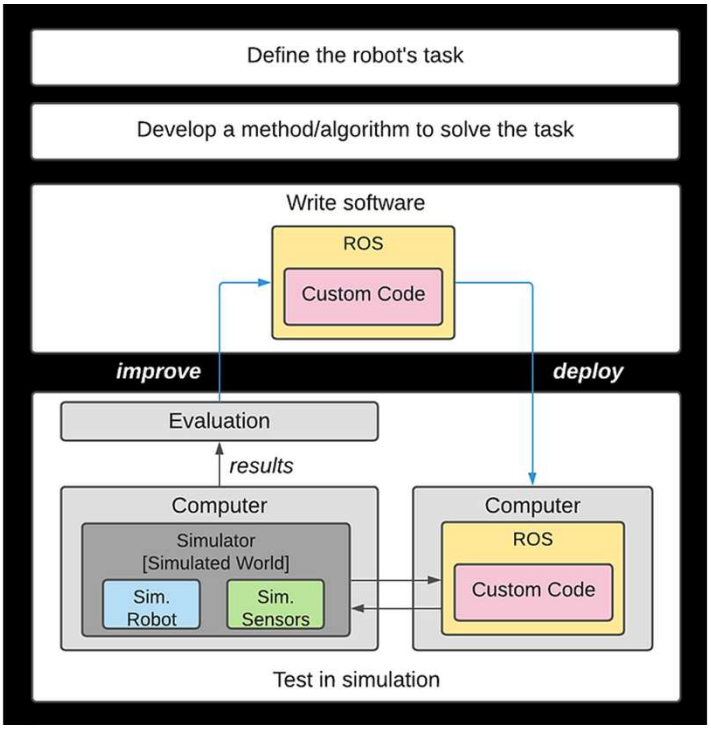
(a) It should be applicable to the setting of general table tennis courts, with a size of 8 m × 16 m;

(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;

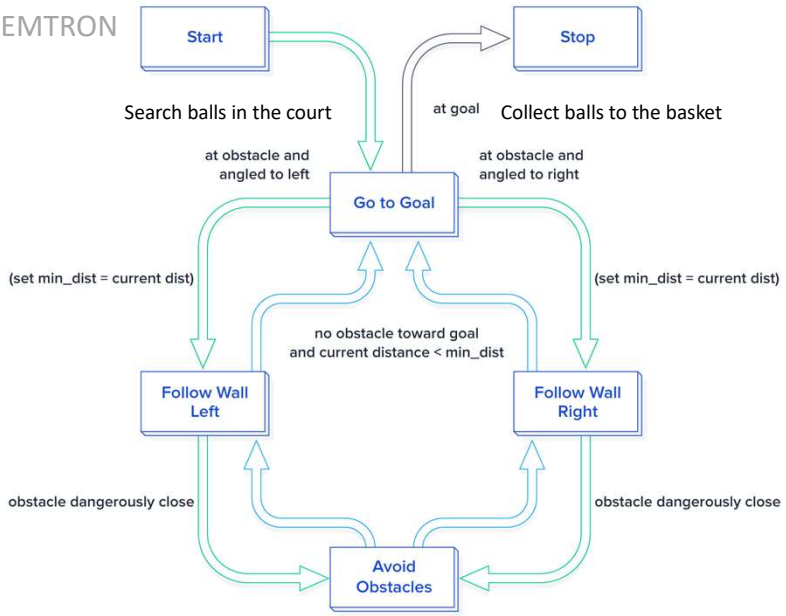
(d) There should a basket to hold the collected table tennis balls and allow players to take out the balls easily to continue practising.

3



4

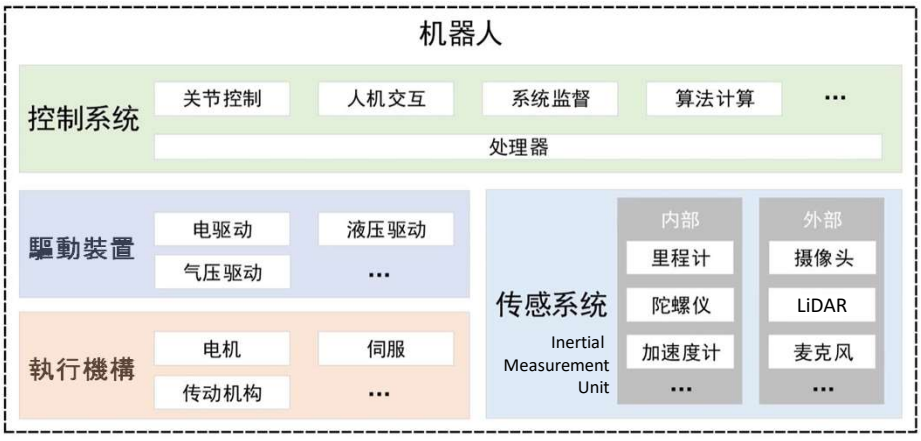
SEMTRON



Define robot's tasks

5

SEMTRON

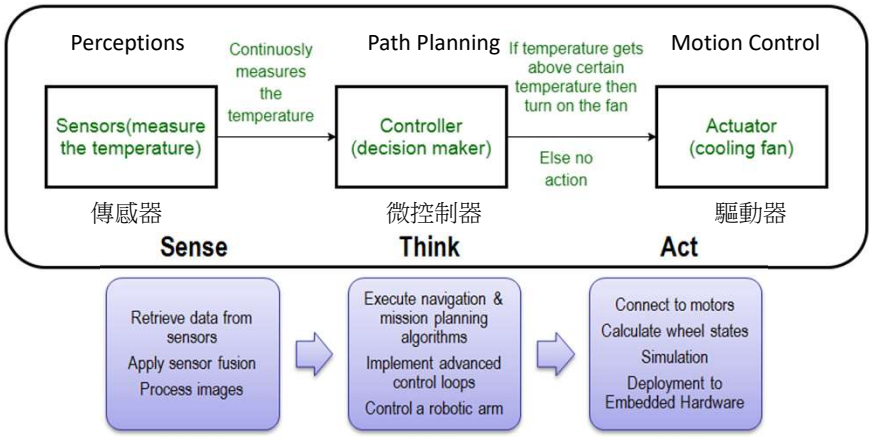


Define robot's functions & behaviour

6

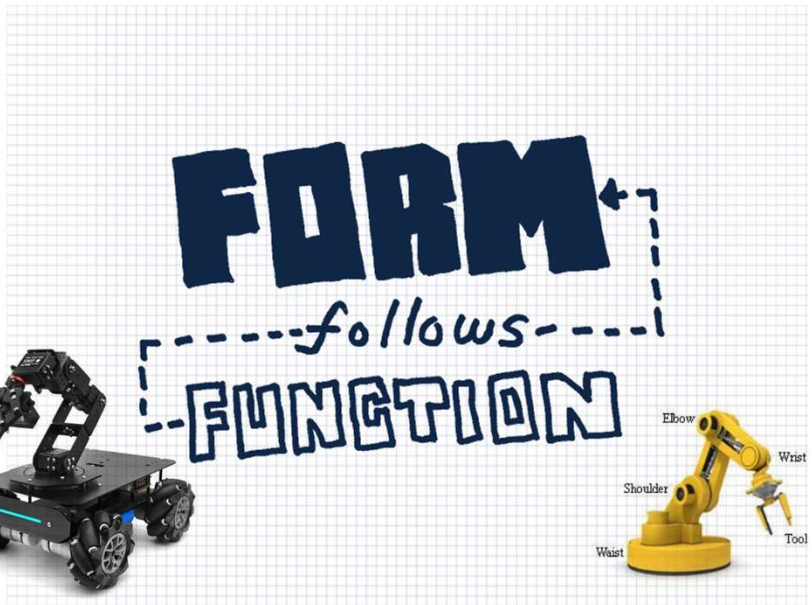
SEMTRON

NVIDIA JETSON Raspberry pi



Robotics in physical computing

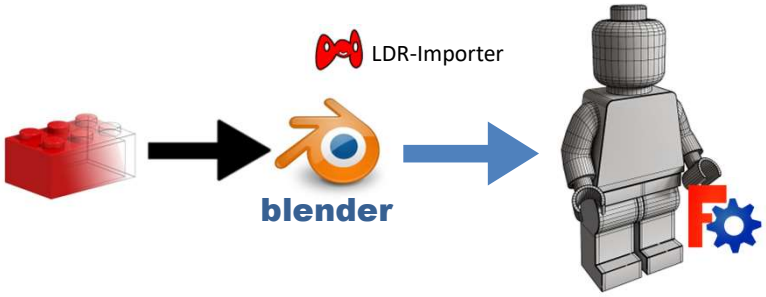
7



8

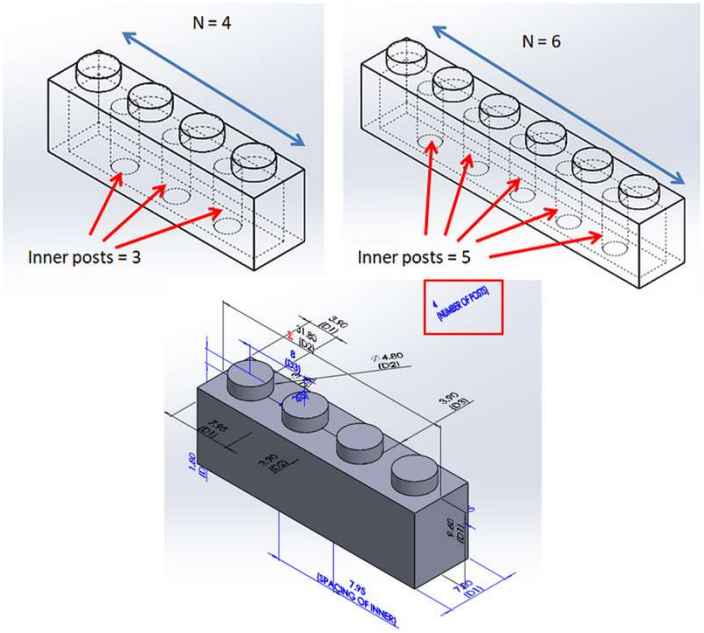
SEMTRON

LDraw.org
Centralised LDraw Resources



Parametric modeling

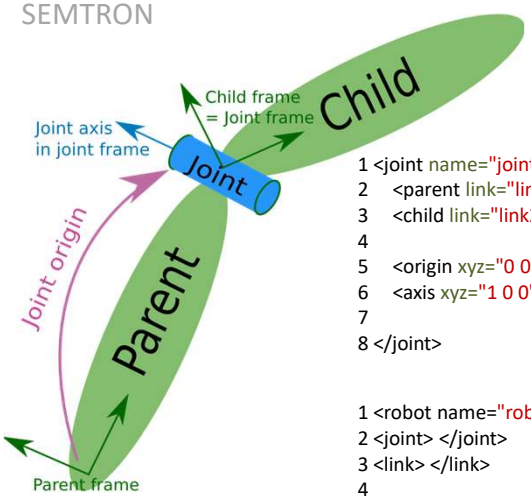
9



Parametric modeling

10

SEMTRON




```
1 <joint name="joint_name" type="continuous">
2    <parent link="link1"/>
3    <child link="link2"/>
4
5    <origin xyz="0 0 0" rpy="0 0 0"/>
6    <axis xyz="1 0 0"/>
7
8 </joint>
```

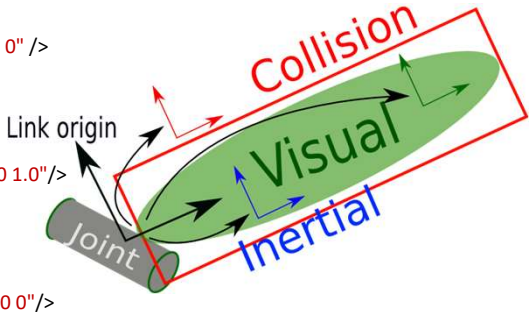
```
1 <robot name="robot_name">
2 <joint> </joint>
3 <link> </link>
4
5 <link> </link>
6 <joint> </joint>
7 </robot>
```

URDF modeling


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

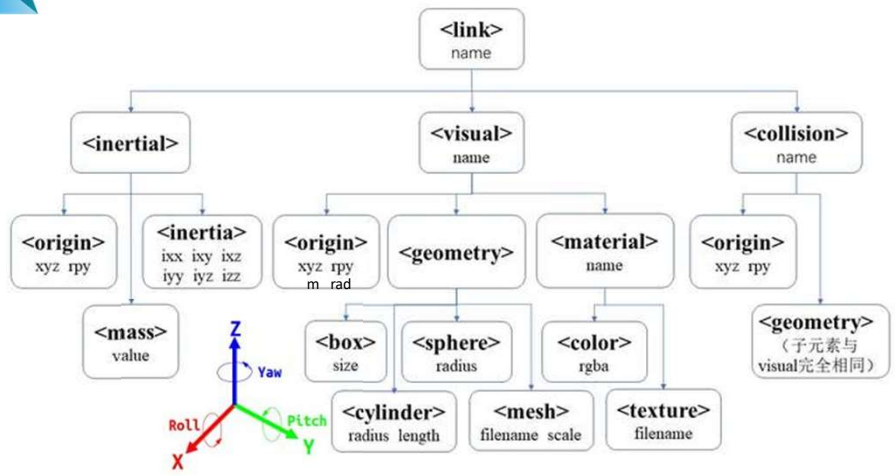
$$\begin{pmatrix} \mathbf{ixx} & \mathbf{ixy} & \mathbf{ixz} \\ ixy & \mathbf{iyy} & \mathbf{iyz} \\ ixz & izy & \mathbf{izz} \end{pmatrix}$$

Inertia Tensor



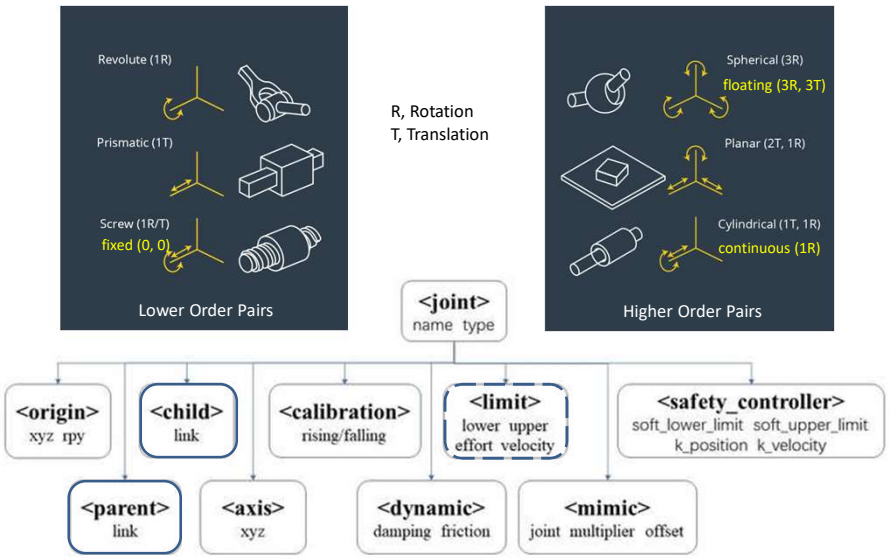
URDF modeling

SEMTRON



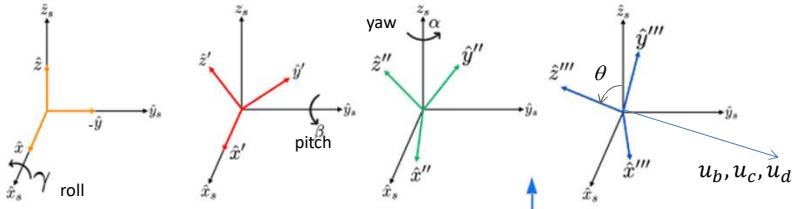
URDF modeling

13



URDF modeling

14

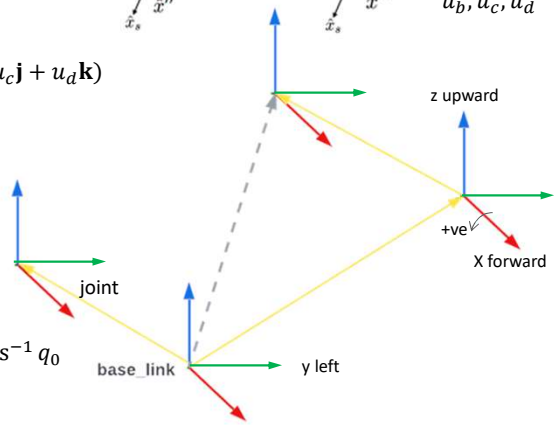


$$q = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(u_b\mathbf{i} + u_c\mathbf{j} + u_d\mathbf{k})$$

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

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

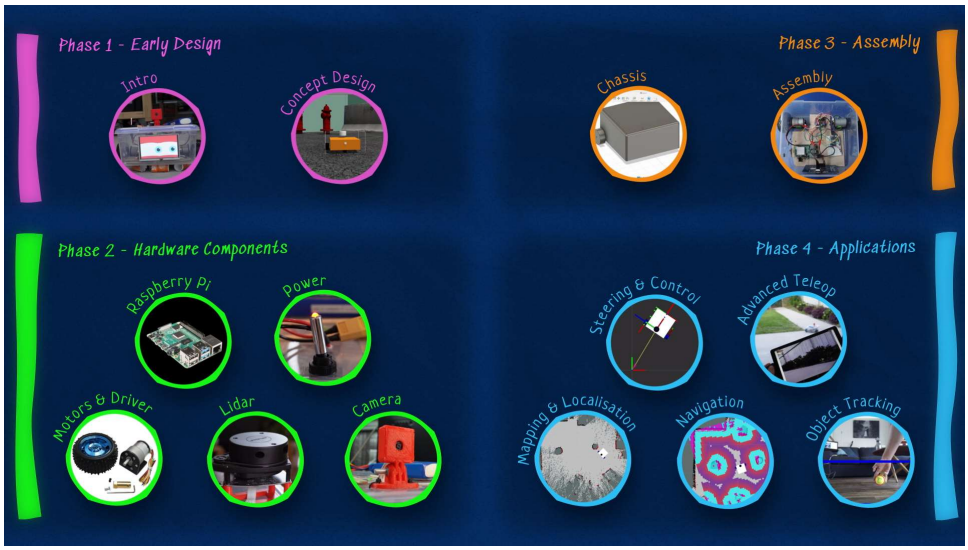
$$\theta = 2\cos^{-1} q_0$$



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
- SolidWorkds 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 SDFormat to URDF Parser
- SDF to URDF Converter in Python
- The Blender Robotics Tools includes a tool to export URDF files from Blender

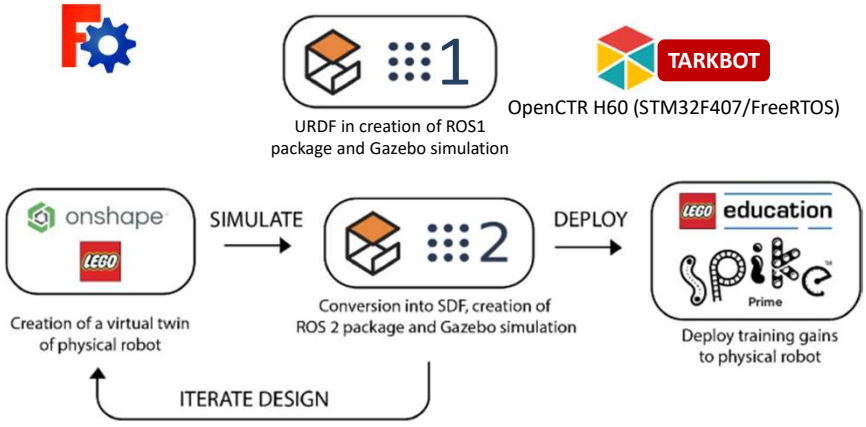
URDF from CAD models

17

SEMTRON

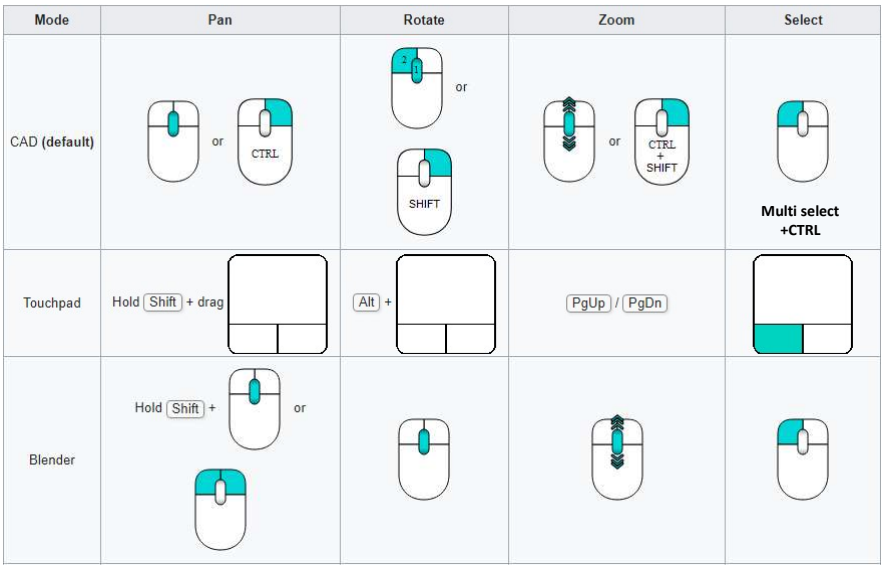
Common Software Common Hardware

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



URDF from CAD models

18



FreeCAD mouse navigation

SEMTRON

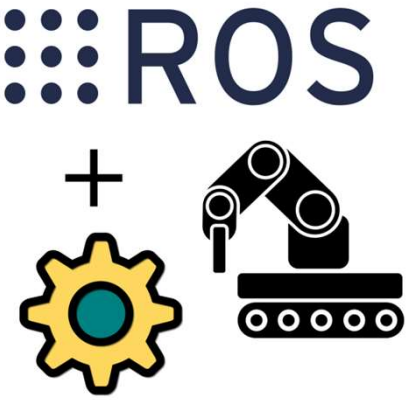
https://github.com/USeebi/Lesson0
https://github.com/drfenixion/freecad.overcross/wiki

- 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

SEMTRON

Design Workspace in Robotics



21

SEMTRON

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

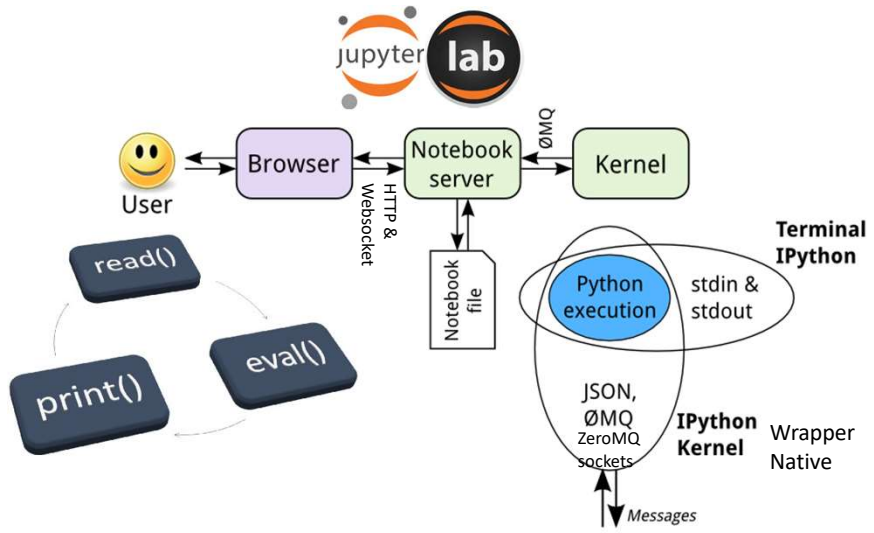
Good Planning — break an extended task into smaller sub-parts and complete them in stages.

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

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

22

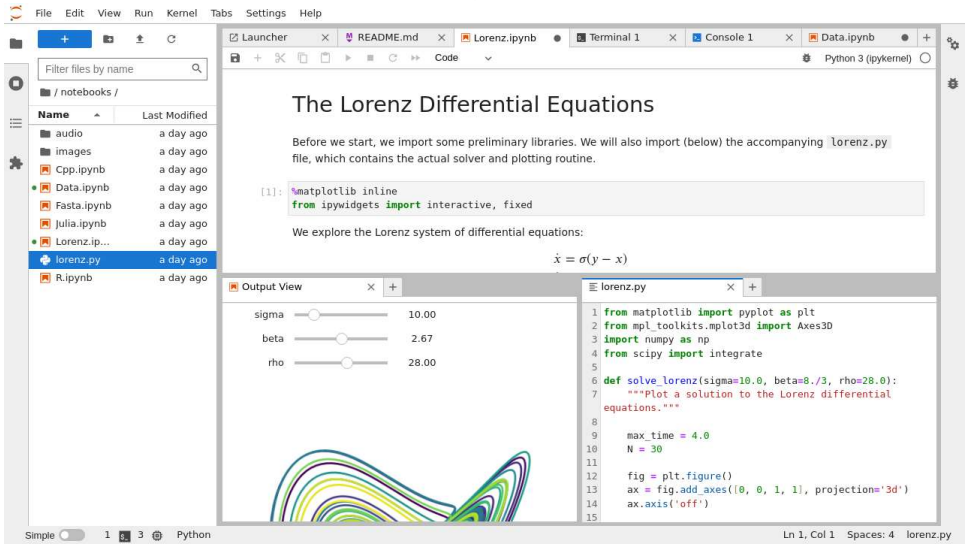
SEMTRON



Proposed Python workspace

23

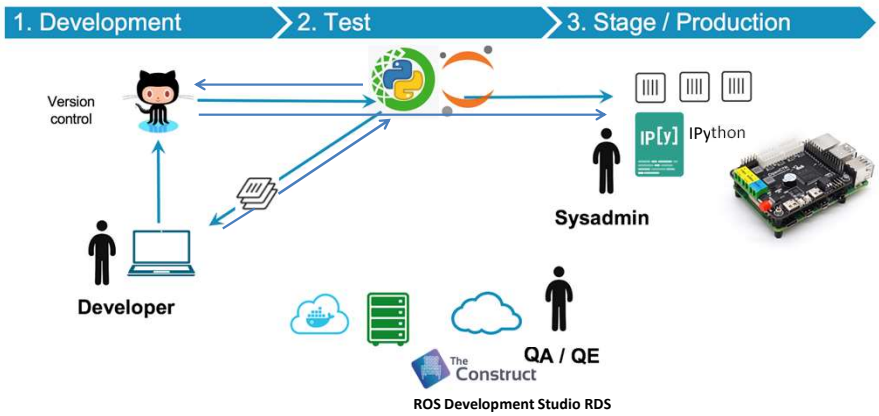
https://robostack.github.io/index.html



Proposed Python workspace

24

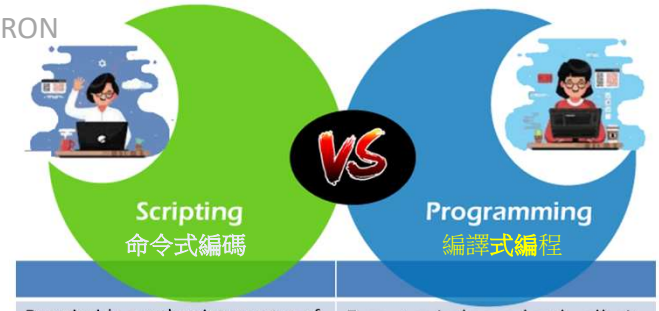
SEMTRON



Proposed CI/CD workflow

SEMTRON

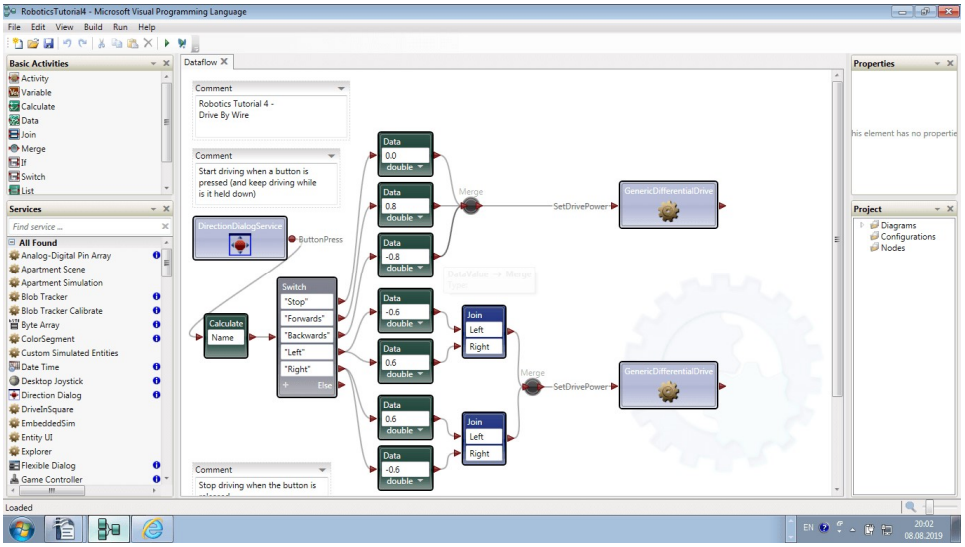
SEMTRON



Runs inside another interpreter of a particular runtime environment	Executes independently albeit required runtime libraries
Do not create a ***.exe*** file	These generate ***.exe*** files
Glue languages for existing programs and libraries	High-speed languages to work with API
Useful in rapid prototyping	
Command statements (declarative language)	

Python scripting

27

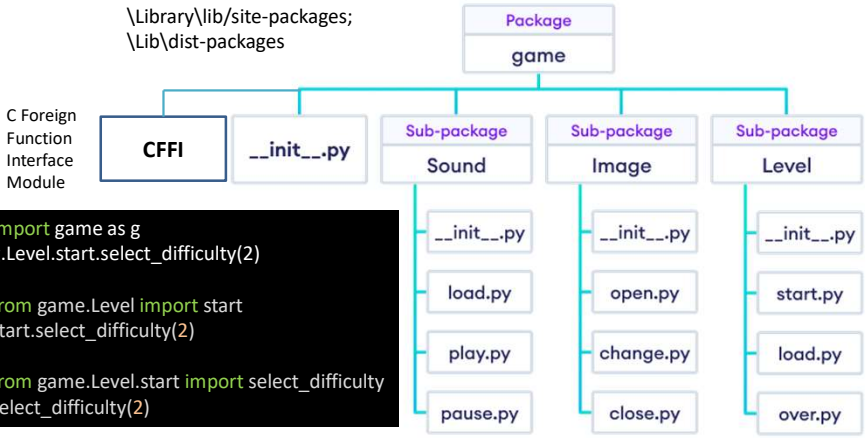


Visual scripting

28

SEMTRON

\Library\lib/site-packages;
\Lib\dist-packages




```
Import game as g
g.Level.start.select_difficulty(2)

from game.Level import start
start.select_difficulty(2)

from game.Level.start import select_difficulty
select_difficulty(2)
```

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

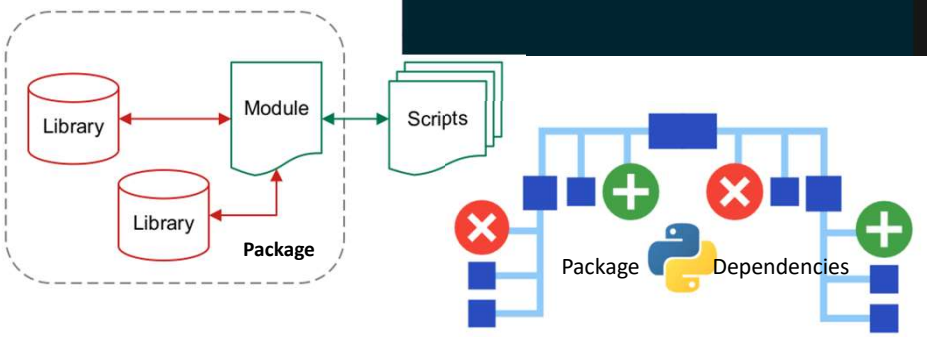
Python packages

SEMTRON

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!
$
```

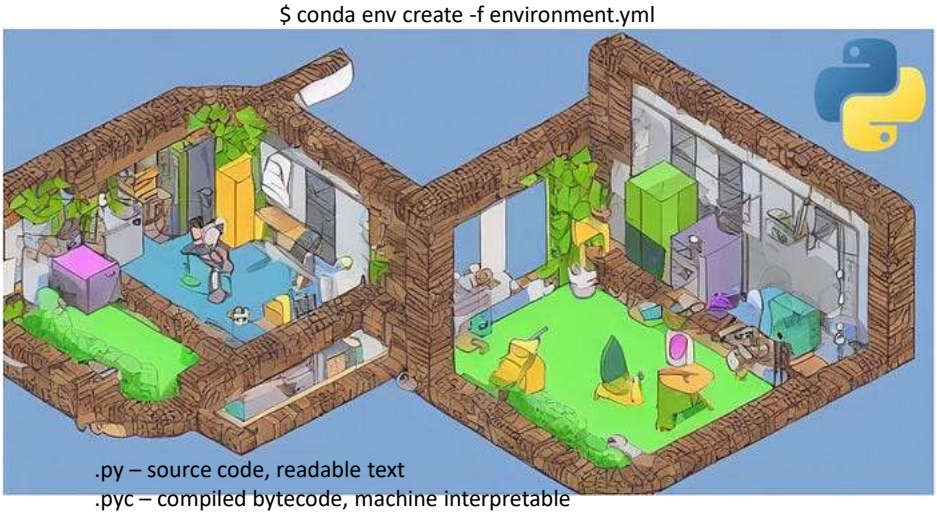
shebang
Path to interpreter



Python packages

SEMTRON

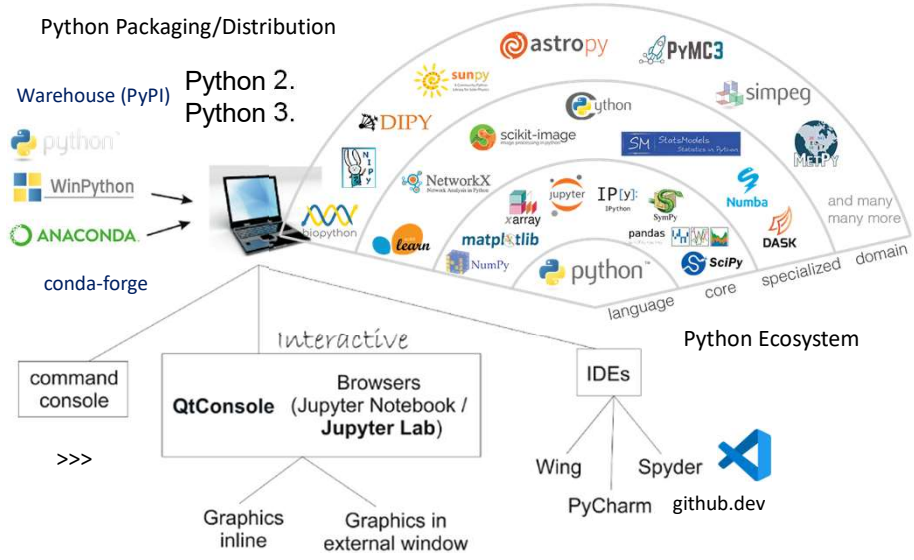
$ conda env export > environment.yml
$ conda env create -f environment.yml



.py – source code, readable text
.pyc – compiled bytecode, machine interpretable

Python virtual environment

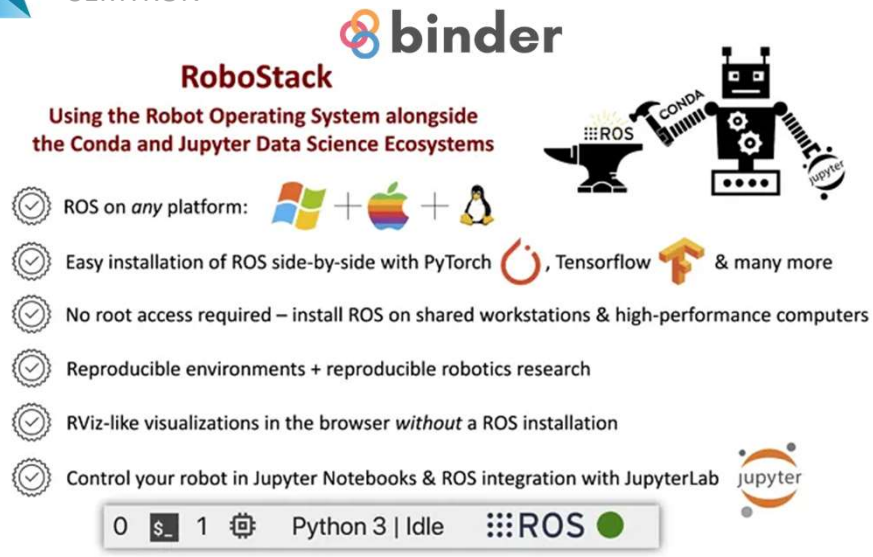
31



Python robotic development

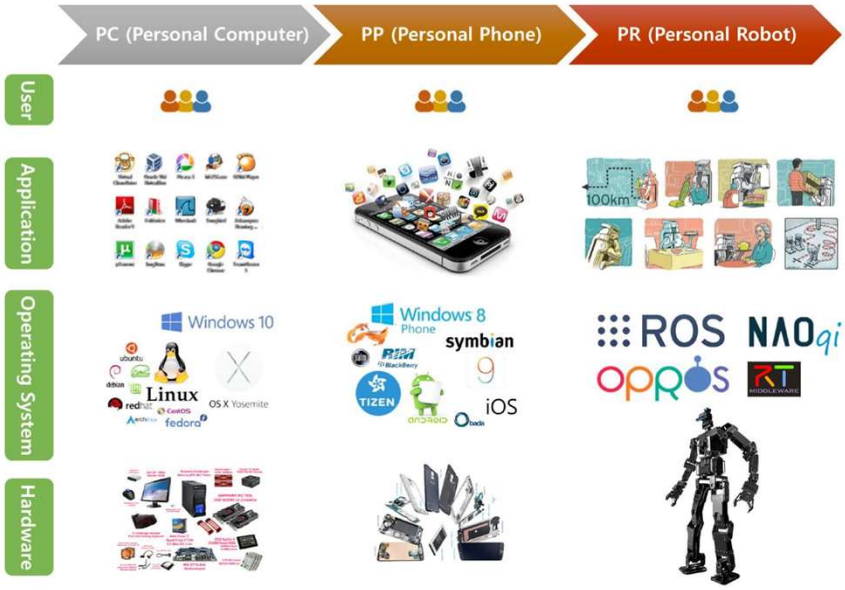
32

SEMTRON



Python robotic development

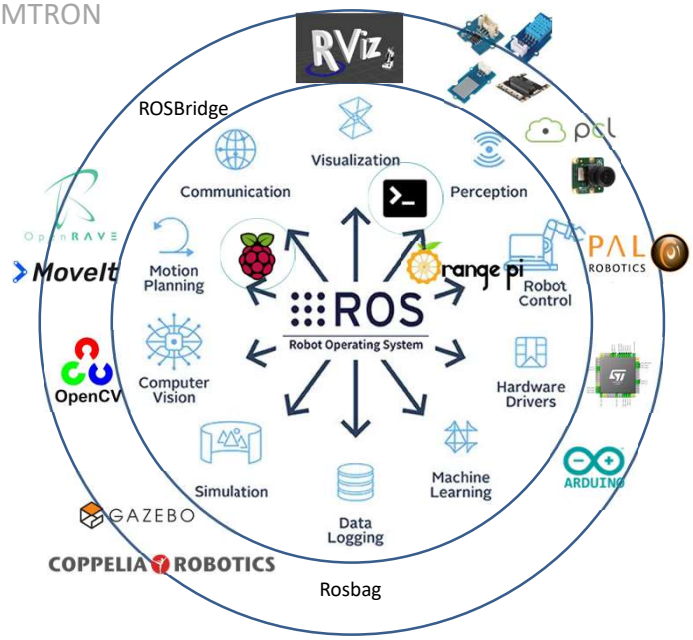
33



Operating System

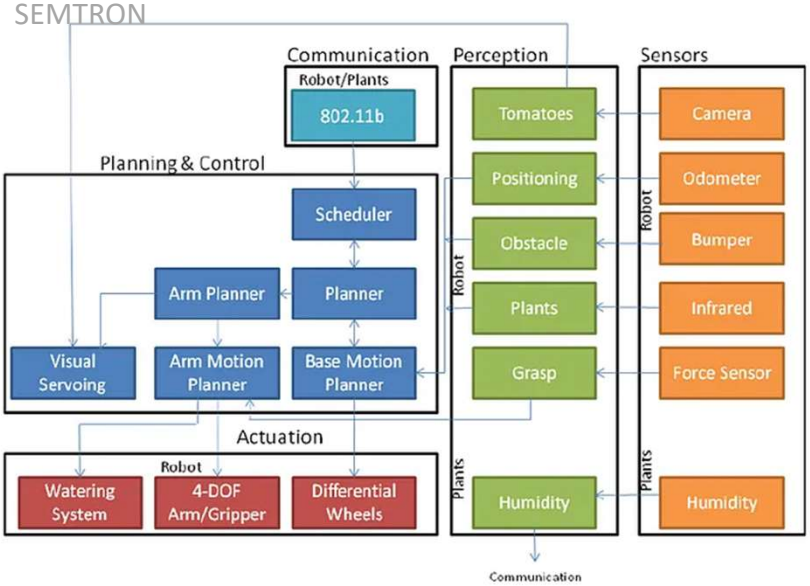
34

SEMTRON



35

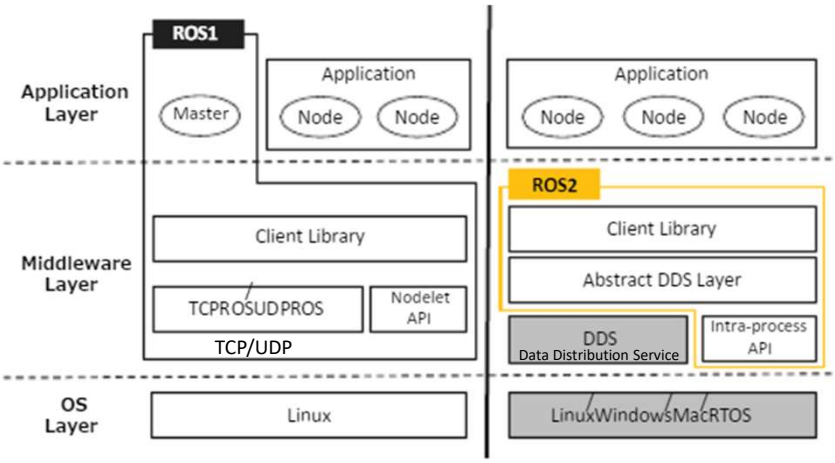
SEMTRON



Robot Operating System

36

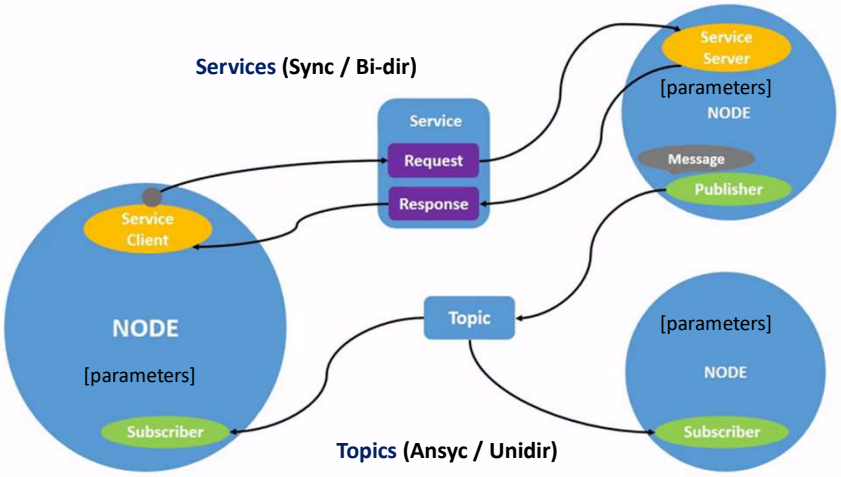
SEMTRON



Robot Operating System

37

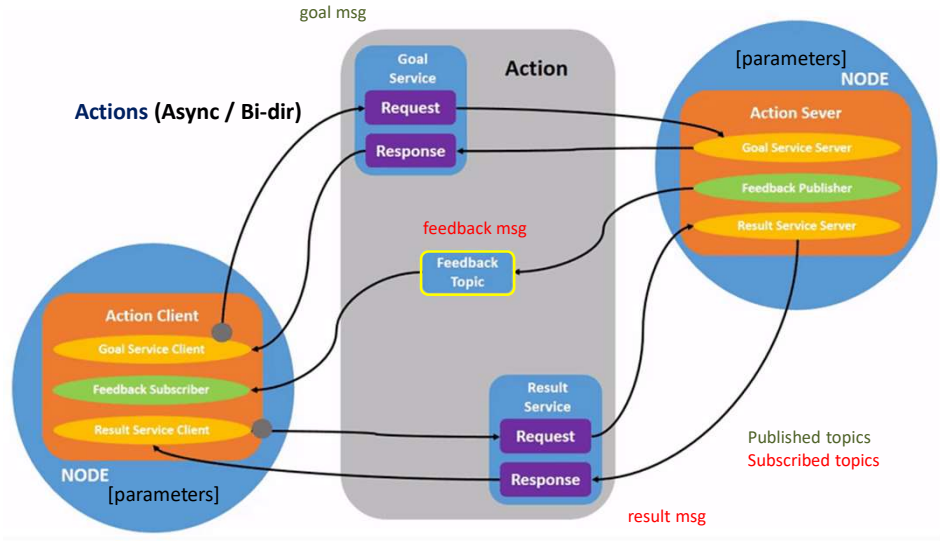
SEMTRON



Robot Operating System

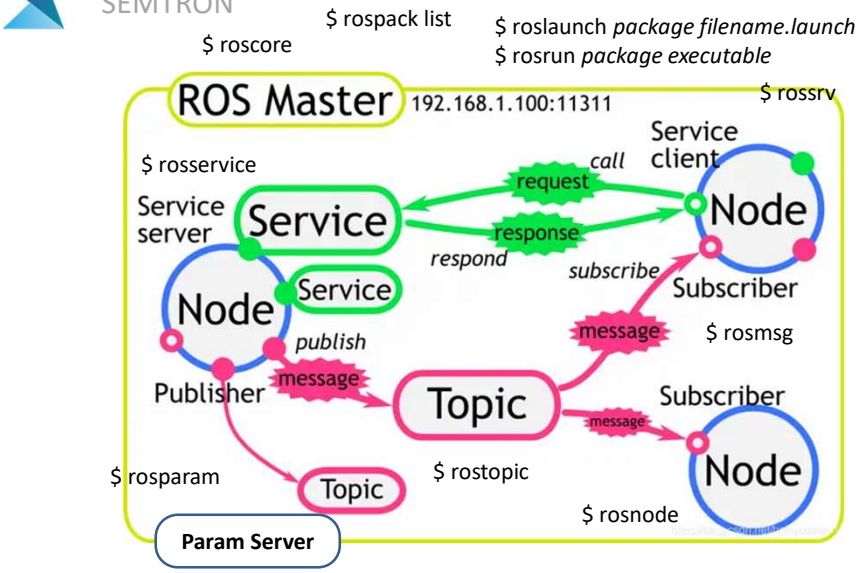
38

SEMTRON



Robot Operating System

SEMTRON



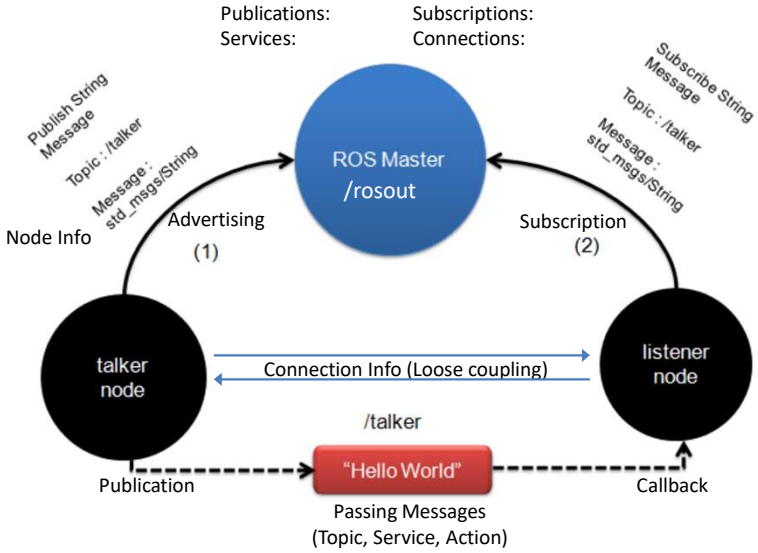
Robot Operating System

SEMTRON

```
### 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
```

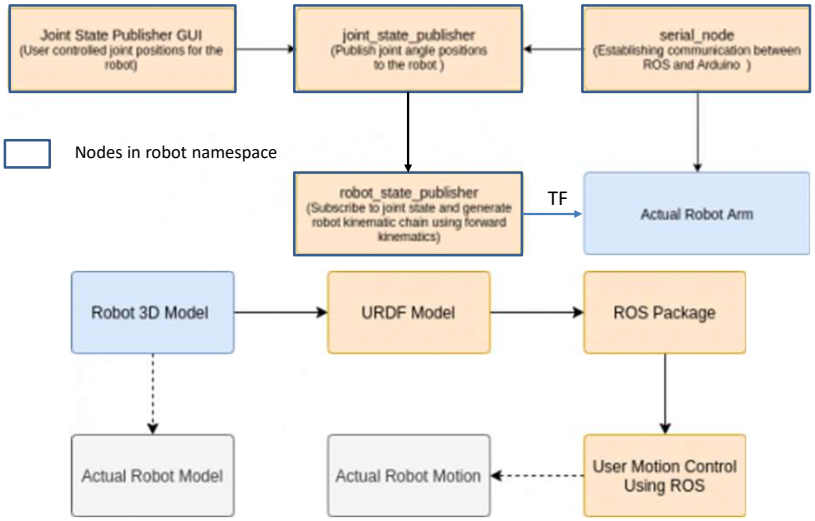
41

SEMTRON



ROS nodes and messages

42

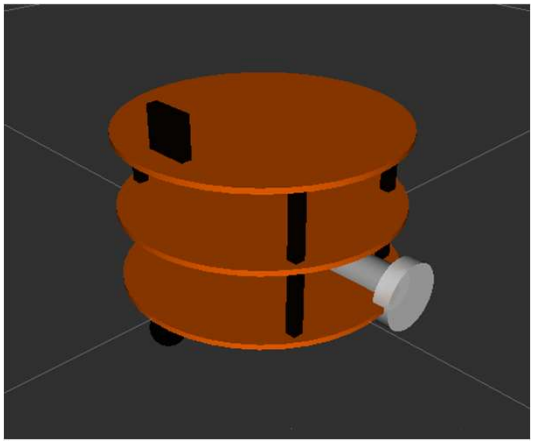


ROS URDF models

43

SEMTRON

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



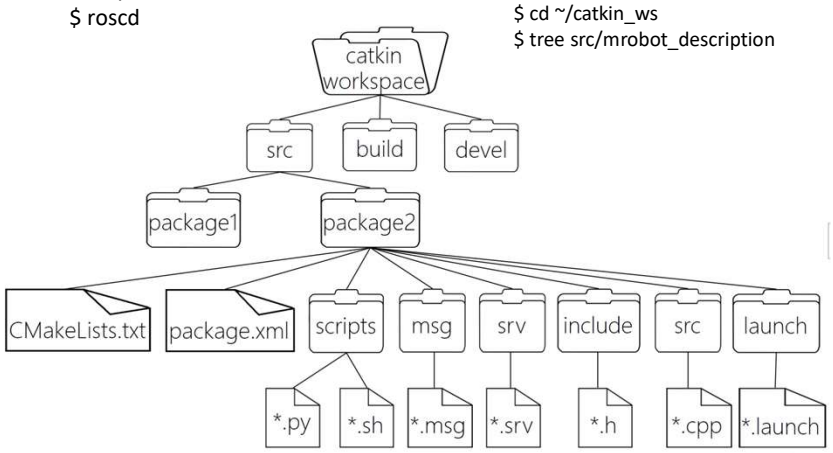
ROS URDF models

44

SEMTRON

$ rospack
$ roscd

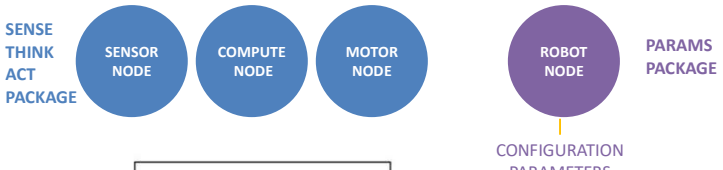
$ cd ~/catkin_ws
$ tree src/mrobot_description

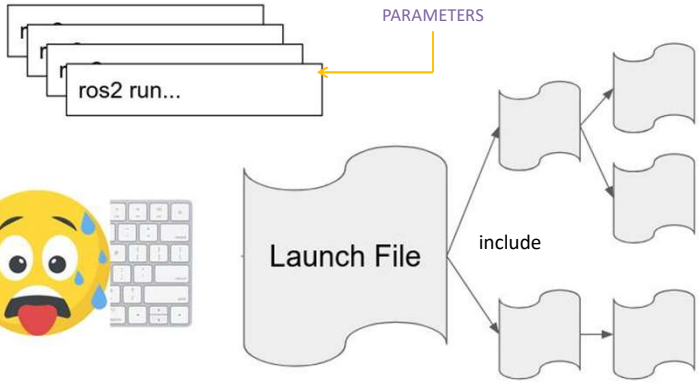


ROS applications

45

SEMTRON

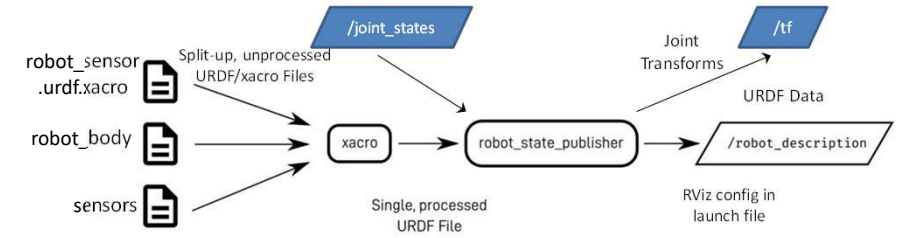




ROS applications

46

```
<xacro:include filename=".xacro" />
```




```
<?xml version="1.0"?>
<robot name="robot_name" xmlns:xacro="http://www.ros.org/wiki/xacro>"
```

```
<xacro:property name="M_PI" value="3.14159"/>
```

```
<xacro:property name="M_PI" value="3.14159"/>
```

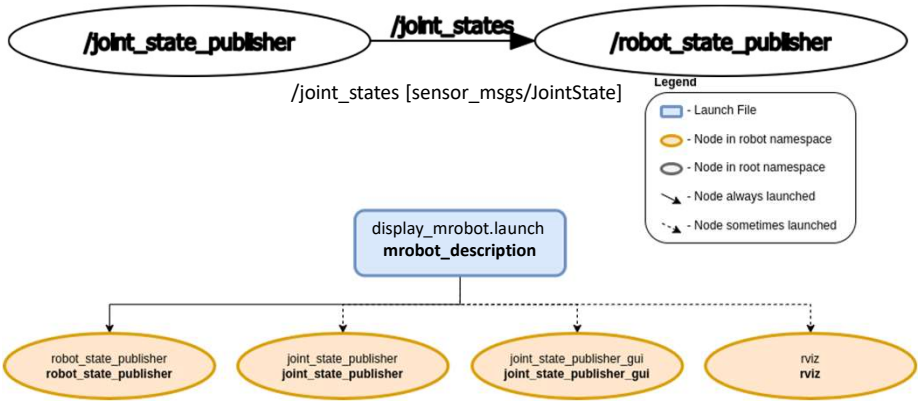
```
<xacro:macro name="urdf_object"> params="A B C" </xacro:macro>
```

ROS URDF applications

SEMTRON

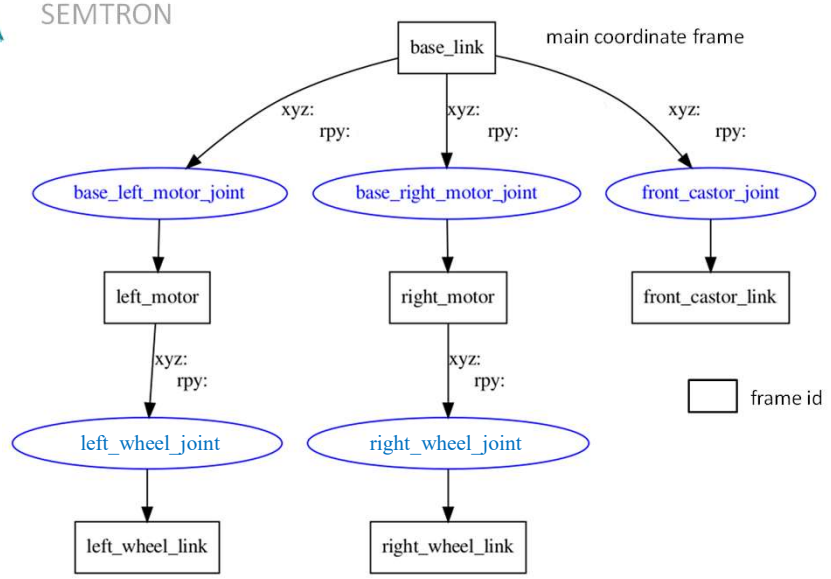
```
$ cd catkin_ws
$ source devel/setup.bash
$ rostopic echo /joint_states
$ rqt_graph
```

Node Graph



ROS visualizations

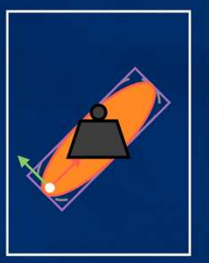
SEMTRON



ROS URDF models


```
<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>
```

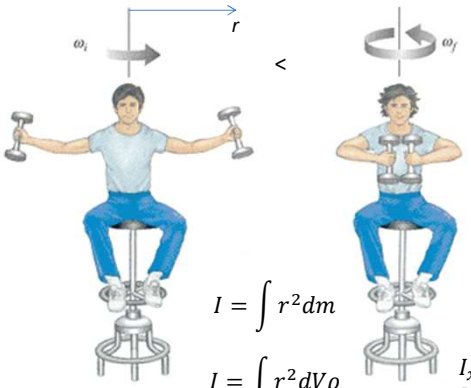
```
<link name="arm_link">
    <visual>
        <geometry>
        <origin>
        <material>
    </visual>
    <collision>
        <geometry>
        <origin>
    </collision>
    <inertial>
        <mass>
        <origin>
        <inertia>
    </inertial>
</link>
```



display_mrobot_chassis_withPhy_urdf.launch

```
<launch>
    <param name="robot_description" textfile="$(find
mrobot_description)/urdf/mrobot_chassis_withPhy.urdf" />
```

SEMTRON



$$I = \int r^2 dm$$

$$I = \int r^2 dV\rho$$

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

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

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

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

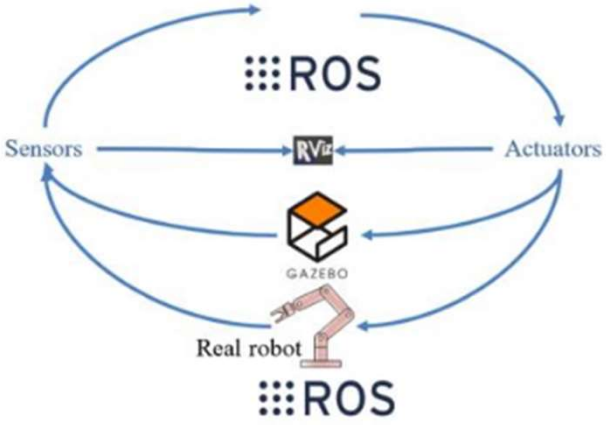
center of gravity $I \triangleq \iiint_V r^2 \rho(r) dV$

Physical properties in ROS URDF

51

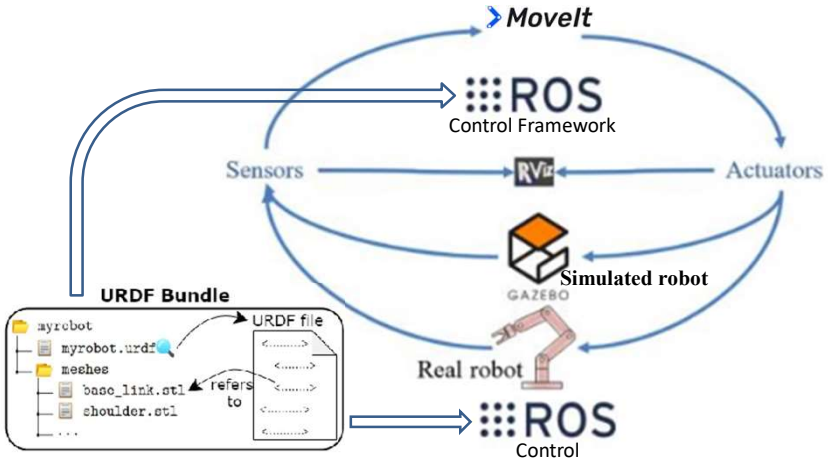
SEMTRON

Design Simulations in Robotics



52

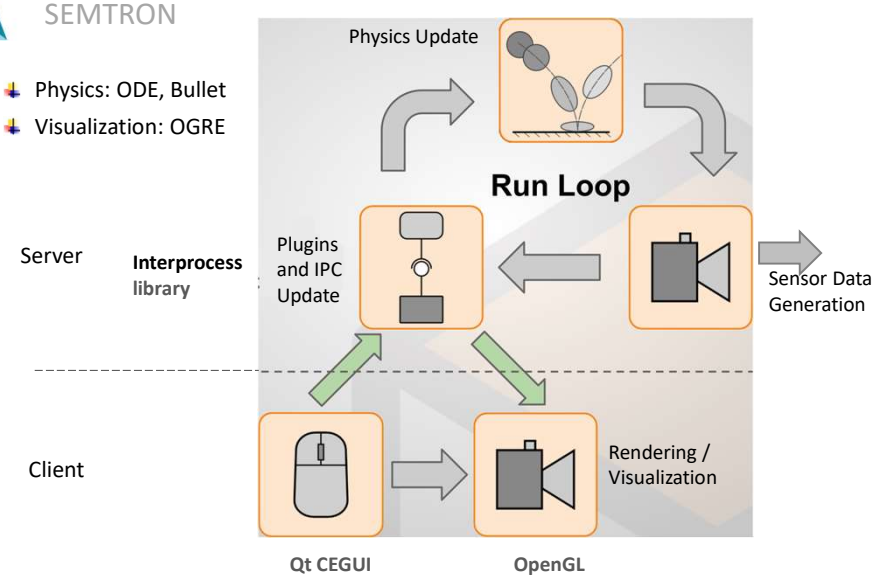
". . . RViz shows you what the robot thinks is happening, while Gazebo shows you what is really happening."



53

SEMTRON

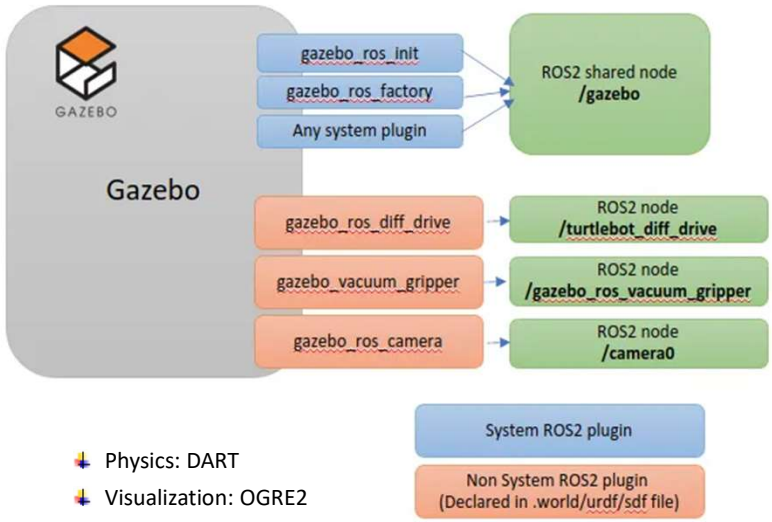
- Physics: ODE, Bullet
- Visualization: OGRE



Gazebo Classic

54

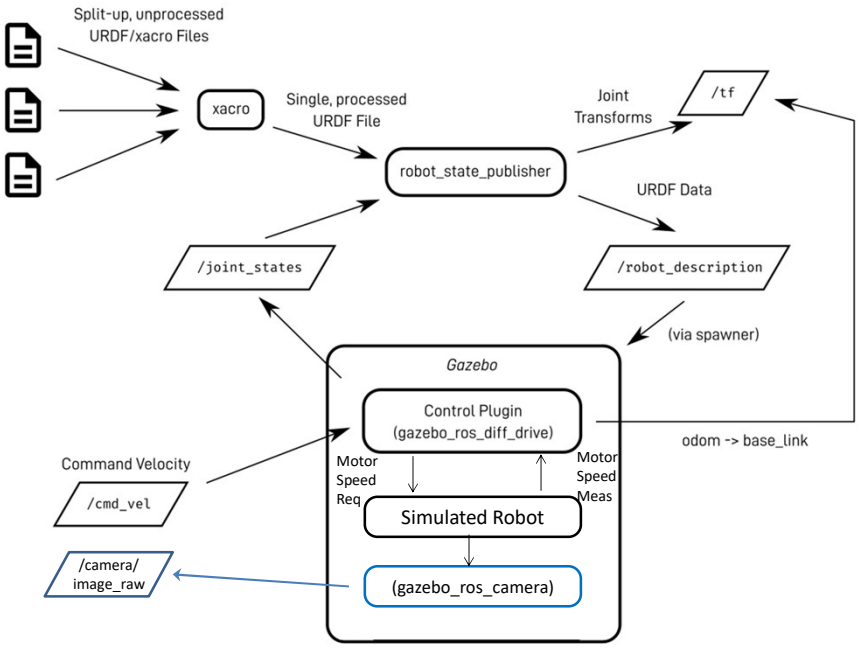
SEMTRON



- Physics: DART
- Visualization: OGRE2

Gazebo Ignition

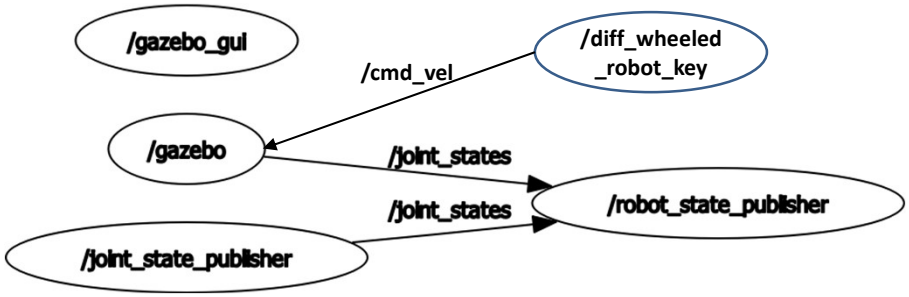
55



56

SEMTRON

```
$ roslaunch mrobot_gazebo view_mrobot_with_camera_gazebo.launch
$ roslaunch mrobot_teleop mrobot_teleop.launch
$ rostopic echo /cmd_vel
```

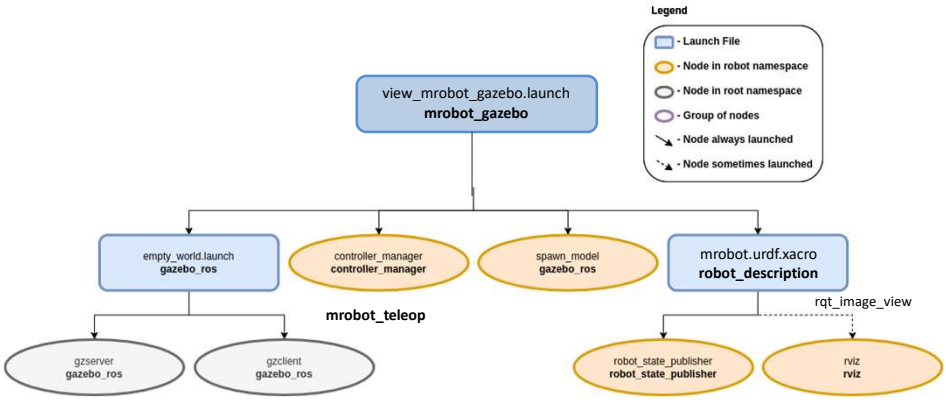


Node Graph

ROS simulation

57

SEMTRON



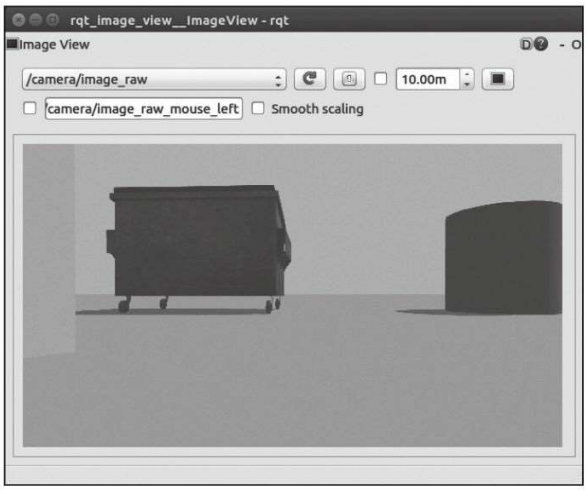
ROS simulation

58

SEMTRON

view_mrobot_with_camera_gazebo.launch

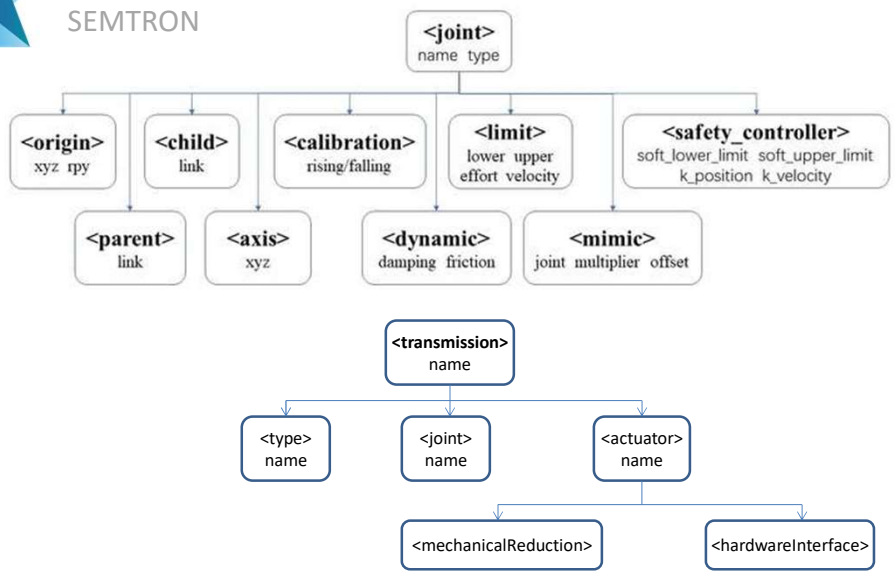
```
<node name="rviz" pkg="rviz" type="rviz" args="-d $(find
mrobot_description)/config/mrobot.rviz" required="true" />
```



ROS simulation

59

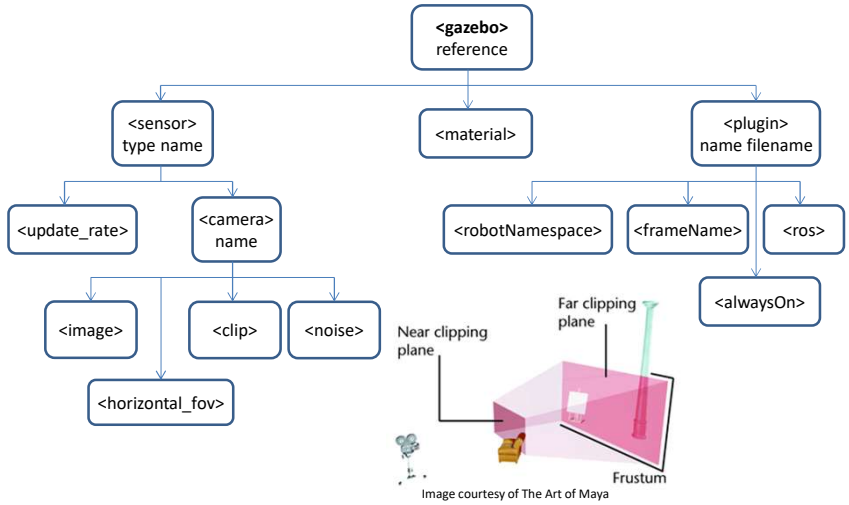
SEMTRON



Gazebo simulation

60

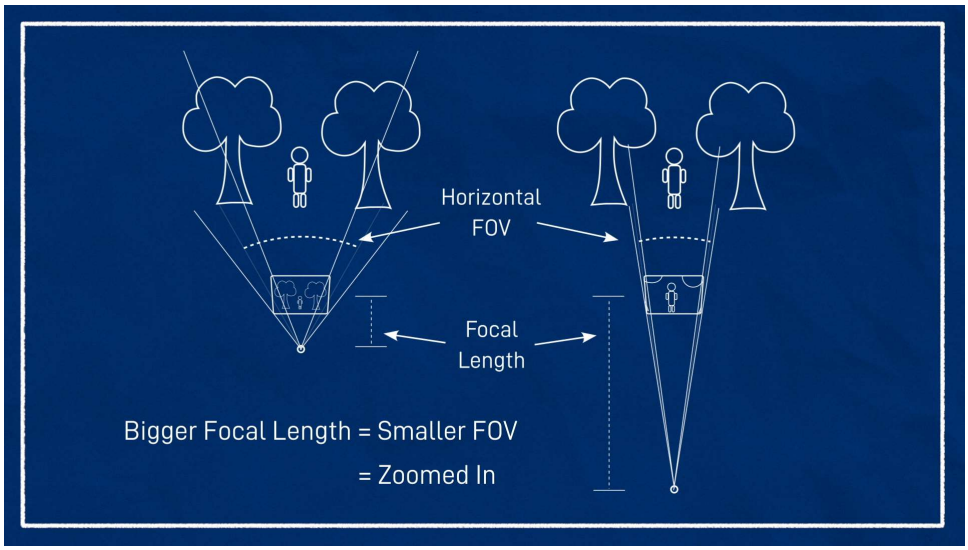
SEMTRON



Gazebo simulation

61

SEMTRON



Gazebo simulation

62

Homework

63

ROS Versions History

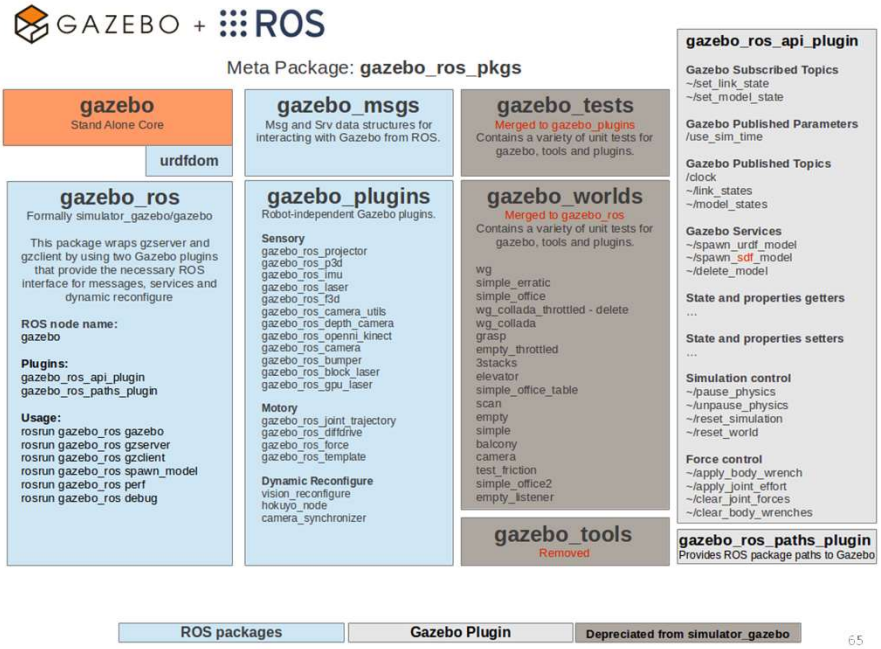
Jan 2010	Mar 2010	Aug 2010	Mar 2011	Aug 2011	Apr 2012	Dec 2012	Sep 2013	Jul 2014
ROS 1.0 Release!	Box Turtle	C Turtle	Diamondback	Electric Emys	Fuerte Turtle	Groovy Galapagos	Hydro Medusa	Indigo Igloo

May 2015	May 2016	May 2017	May 2018	2020	Dec 2017	Jul 2018	Dec 2018	Jun 2019
Jade Turtle	Kinetic Kame	Lunar Loggerhead	Melodic Morenia	Noetic Ninjemys	(ROS2) Ardent Apalone	(ROS2) Bouncy Bolson	(ROS2) Crystal Clemmys	(ROS2) Dashing Diademata

* Images sources: https://en.wikipedia.org/wiki/Robot_Operating_System
* Background picture by Negative Space

The Construct
Learn and develop for robots with ROS

64



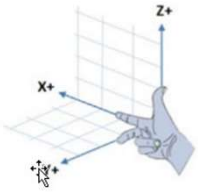
https://repo.anaconda.com/miniconda/Miniconda3-py311_24.5.0-0-Windows-x86_64.exe

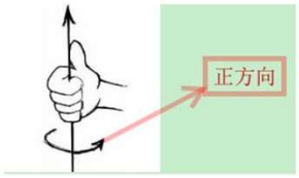
```
>Downloads\Miniconda3-py311_24.5.0-0-Windows-x86_64.exe
  /InstallationType=JustMe /AddToPath=0 /RegisterPython=0 /NoRegistry=1 /S
  /D=K:\Miniconda3\py311
```

```
(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





67