# Design Guidelines

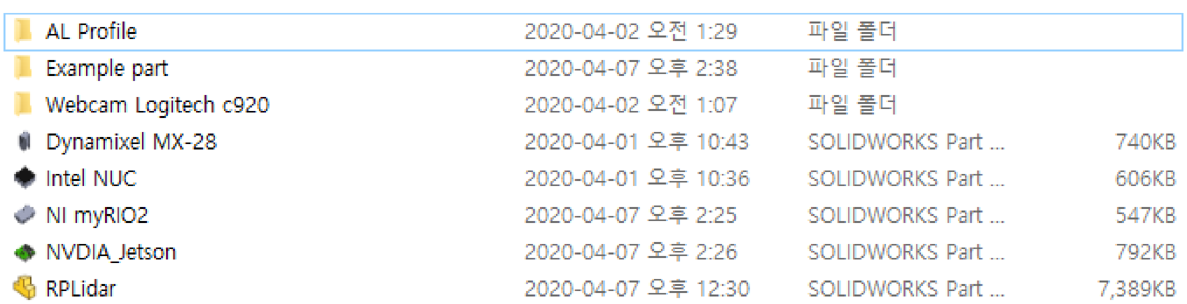
## 1. Design Constraints

- **Robot skeleton:** Made of aluminum profile.
- **Materials of additional parts:** ABS, Steel alloy, Aluminum alloy, Acrylic
- **Products (Sensors, PC, etc.):** Use at least one of the Products, following the maximum quantity constraints listed in the table below.

| Products | Dynamixel MX-28 | Intel NUC | NI myRIO | NVDIA Jetson | RPLidar | Webcam Logitech c920 |
|---|---|---|---|---|---|---|
| Quantity | No limit | 1 | 1 | 1 | 1 | 1~2 |

## 2. Starter-Kit

3D models of the system components (Solidworks Libs.zip) developed by our course TAs should be available from the course website.




| Folders/files | Description |
|---|---|
| AL Profile | the aluminum profile for constructing the robot skeleton |
| Example parts | Examples of additional parts |
| Webcam Logitech c920 | 3d model of Webcam. You can use the assembly file 'Webcam_C920_ASM.SLDASM' or just the part file 'Webcam_C920.SLDPRT' |
| Others | SLDPRT files of 3d model of the products including: |

| | |
|---|---|
| | Dynamixel MX-28 motor, Intel NUC, NI myRIO, NVIDIA Jetson, RPLidar and Logitech Webcam |

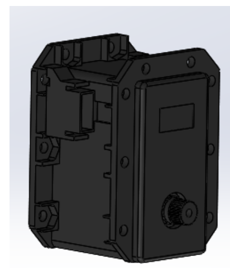
# 3. System Design

### Wheels and actuators


You can freely choose wheel dimensions as long as they do not exceed the maximum size allowed.

- Wheel diameter < 150mm
- Wheel thickness < 50 mm

The materials of the wheels should be either ABS or aluminum alloy.



Dynamixel MX-28 is the only option for the rotational actuator selection. Consider the actual motor configuration when generating a URDF file. (3.1Nm torque limit and 7.01622 rad/s angular velocity limit.)

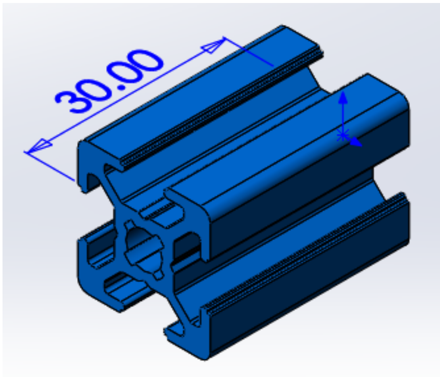
(http://emanual.robotis.com/docs/en/dxl/mx/mx-28/)

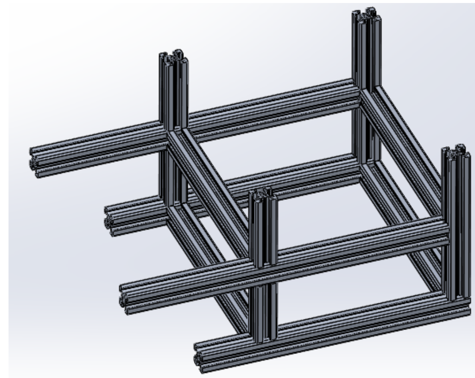
**** For design convenience, it is accepted to use a linear actuator. However, the actuator should have a 200N force limit and a 0.04m/s velocity limit.***

### Chassis skeleton

The robot base can be made of aluminum profiles (20x20 or 30x30 in 'AL Profile' folder), and the length can be adjusted freely. You can use other types of profiles if you want.

One example for designing the skeleton of robot is shown below. You do not need to add brackets for fastening the profile in this virtual design project, although fastening (bolting, bonding, etc.) is an important part of the structure for a real mechanical system.


30.00

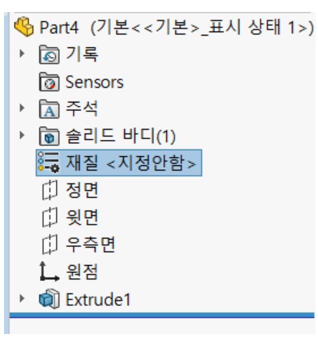


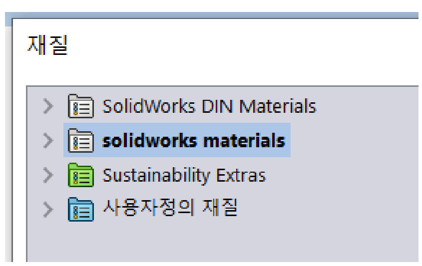
## Other parts

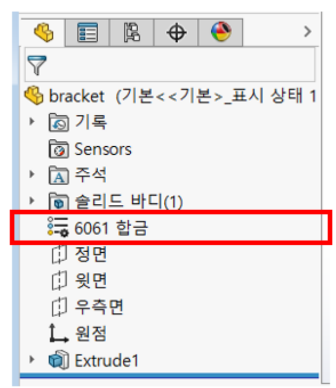
Assume that necessary parts have been already processed via 3d printing, milling machining, lathe machining, laser cutting, etc. and provided. As a result, it is free to make your own structure with given material options from the table shown below. If you want to use other materials, consult with TAs.

| Processing Method | Materials | Path |
|---|---|---|
| 3d printing | ABS | 플라스틱(Plastic) □ ABS |
| milling machining, lathe machining, laser cutting | Aluminum alloy | 알루미늄 합금(Aluminum Alloy) □ 6061 Alloy |
| | Steel alloy | 강(Steel) □ Alloy Steel |
| | Acrylic | 플라스틱(Plastic) □ 아크릴(acrylic) |

The material of the part can be set by clicking재질(Material) -> 재질 편집(Edit Material) -> solidworks materials (see below).

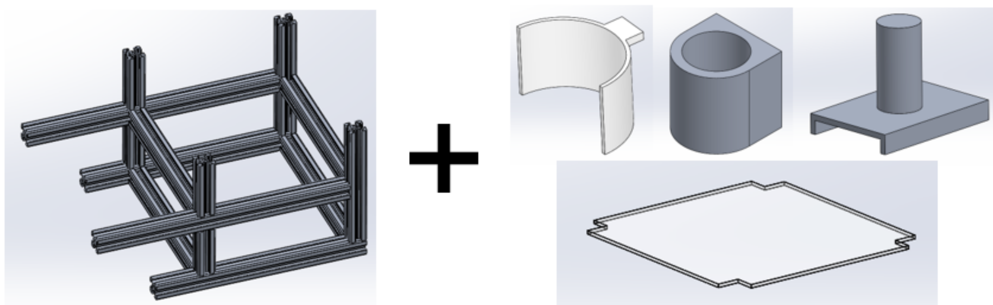


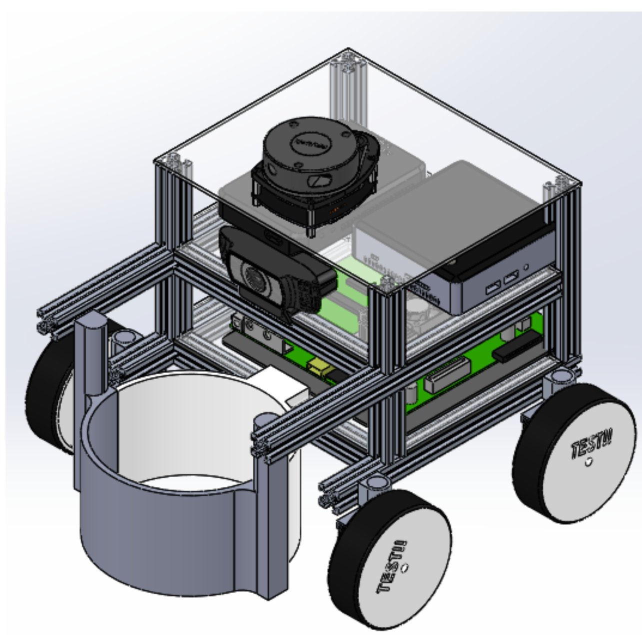
## 3.1. Assembly and integration

Additional parts should be attached to the skeleton parts. Again, there is no need for considering fastening components.



Robot skeleton Other parts(Link, brackets, floors)

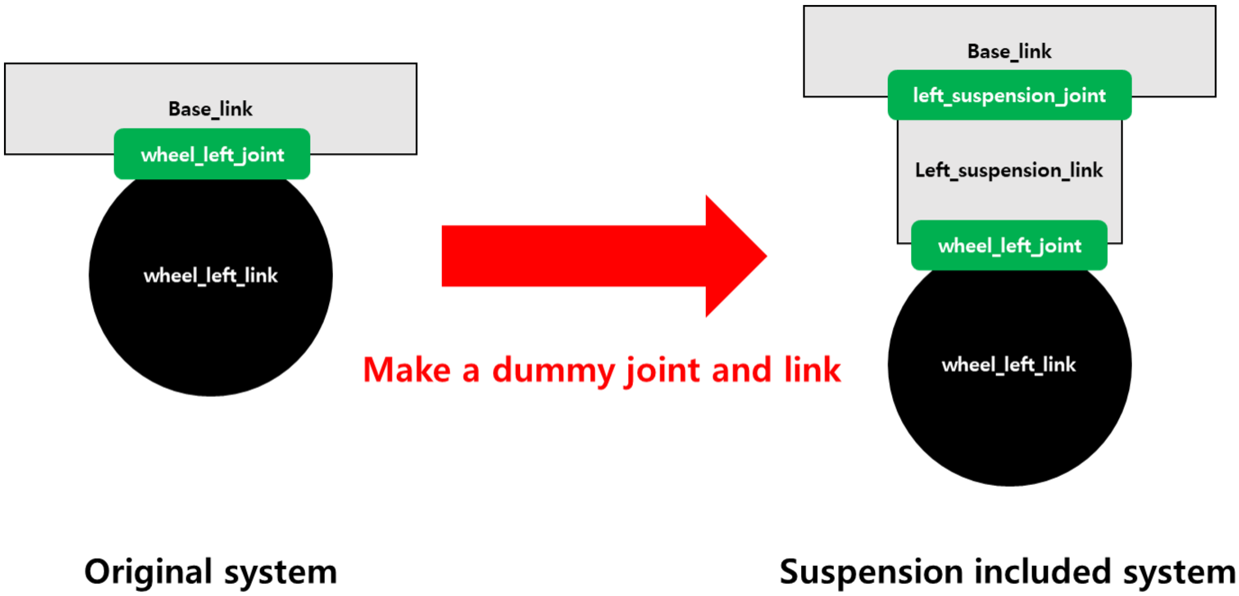
Install the system components on the mobile system. The size and mass of each component should be considered in finalizing the layout.



## 3.2. Suspension Design

You may want to design a suspension system to reduce vibration while passing over bumps in the entrance zone.

For example, if you want to add a suspension system to the left wheel.



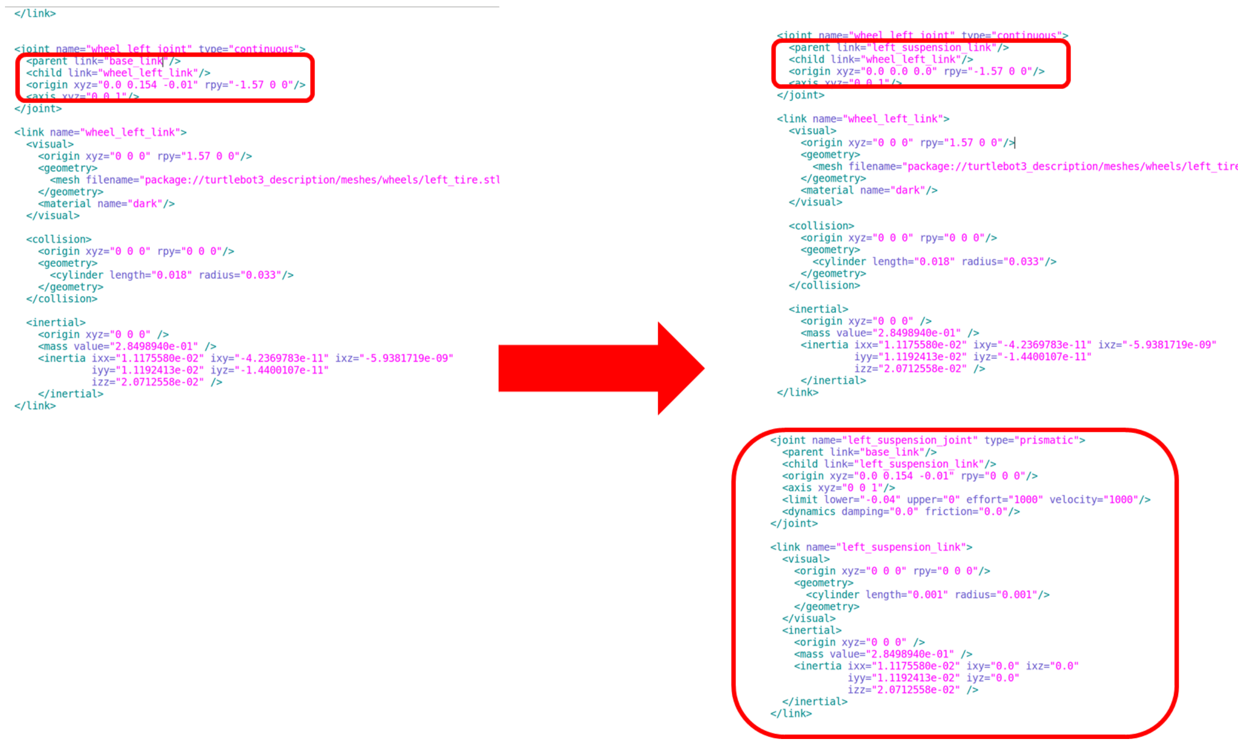

*Example files are given in the given zip file.

(*urdf/exex2.urdf in Capstone_Design_2021_Spring_Suspension_Design_Example.zip*)

(Important! You need to change the paths to the mesh files in the *urdf/exex2.urdf* file to your personal path.)

**Step 1:** Create a dummy joint and a link between the links where you want to insert the suspension system (between the base link and the wheel in the above-shown example) in your urdf file.

**Step 2:** Set this virtual suspension joint as “prismatic” or “revolute” and set the limit of the suspension system considering the maximum movement you want to allow.

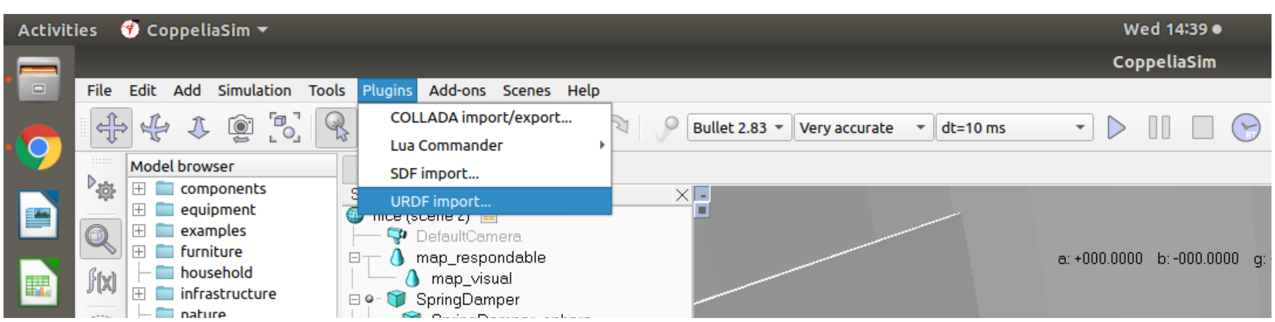


```xml
</link>

<joint name="wheel_left_joint" type="continuous">
  <parent link="base_link"/>
  <child link="wheel_left_link"/>
  <origin xyz="0.0 0.154 -0.01" rpy="-1.57 0 0"/>
  <axis xyz="0 0 1"/>
</joint>

<link name="wheel_left_link">
  <visual>
    <origin xyz="0 0 0" rpy="1.57 0 0"/>
    <geometry>
      <mesh filename="package://turtlebot3_description/meshes/wheels/left_tire.stl
    </geometry>
    <material name="dark"/>
  </visual>

  <collision>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <cylinder length="0.018" radius="0.033"/>
    </geometry>
  </collision>

  <inertial>
    <origin xyz="0 0 0" />
    <mass value="2.8498940e-01" />
    <inertia ixx="1.1175580e-02" ixy="-4.2369783e-11" ixz="-5.9381719e-09"
             iyy="1.1192413e-02" iyz="-1.4400107e-11"
             izz="2.0712558e-02" />
    </inertial>
</link>
```

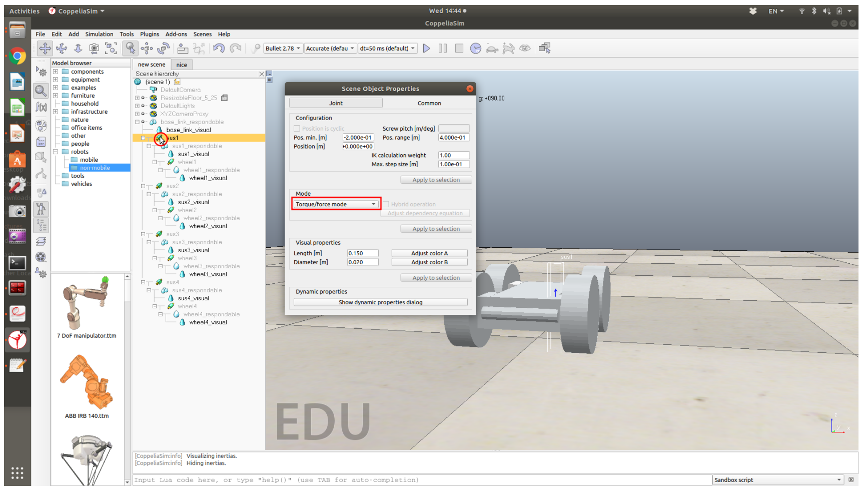
```xml
<joint name="wheel_left_joint" type="continuous">
  <parent link="left_suspension_link"/>
  <child link="wheel_left_link"/>
  <origin xyz="0.0 0.0 0.0" rpy="-1.57 0 0"/>
  <axis xyz="0 0 1"/>
</joint>

<link name="wheel_left_link">
  <visual>
    <origin xyz="0 0 0" rpy="1.57 0 0"/>
    <geometry>
      <mesh filename="package://turtlebot3_description/meshes/wheels/left_tire
    </geometry>
    <material name="dark"/>
  </visual>

  <collision>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <cylinder length="0.018" radius="0.033"/>
    </geometry>
  </collision>

  <inertial>
    <origin xyz="0 0 0" />
    <mass value="2.8498940e-01" />
    <inertia ixx="1.1175580e-02" ixy="-4.2369783e-11" ixz="-5.9381719e-09"
             iyy="1.1192413e-02" iyz="-1.4400107e-11"
             izz="2.0712558e-02" />
    </inertial>
</link>

<joint name="left_suspension_joint" type="prismatic">
  <parent link="base_link"/>
  <child link="left_suspension_link"/>
  <origin xyz="0.0 0.154 -0.01" rpy="0 0 0"/>
  <axis xyz="0 0 1"/>
  <limit lower="-0.04" upper="0" effort="1000" velocity="1000"/>
  <dynamics damping="0.0" friction="0.0"/>
</joint>

<link name="left_suspension_link">
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0"/>
    <geometry>
      <cylinder length="0.001" radius="0.001"/>
    </geometry>
  </visual>
  <inertial>
    <origin xyz="0 0 0" />
    <mass value="2.8498940e-01" />
    <inertia ixx="1.1175580e-02" ixy="0.0" ixz="0.0"
             iyy="1.1192413e-02" iyz="0.0"
             izz="2.0712558e-02" />
  </inertial>
</link>
```

***Or you can directly design the suspension joint and link in solidworks.**

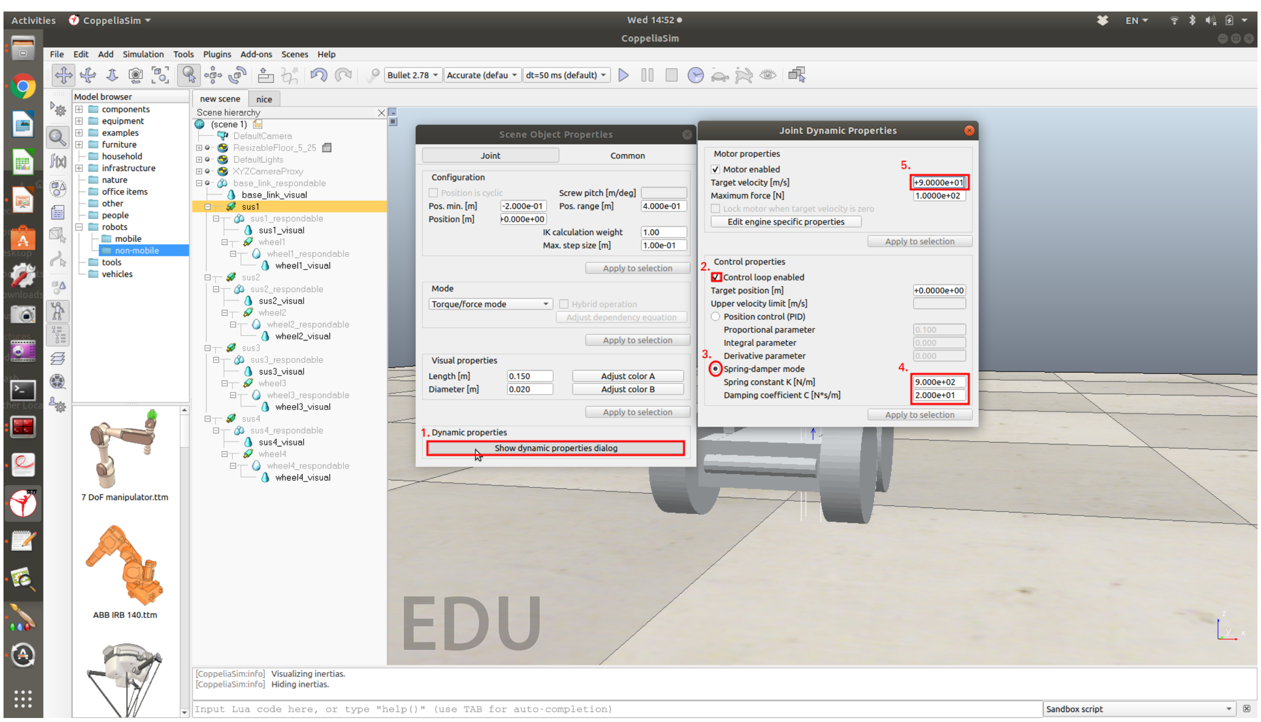
**Step 3:** Now load the robot in the Coppeliasim simulator using the URDF import plugin.



**Step 4:** Double click the suspension joint in 'Scene hierarchy' to setup the suspension joint. Change the Mode to Torque/force mode



**Step 5:** Get into 'Show dynamic properties dialog' and check 'control loop enabled'. Furthermore, check the 'spring-damper mode' since we are going to use the suspension joint as a spring and damper. Now, set the desired spring constant and damping coefficient as you want. Finally, set the 'Target Velocity' high enough. This is because Coppeliasim's torque/force controller's control structure. The controller outputs the torque/force until the joint meets the target velocity value. So if the user sets the 'Target Velocity' zero, the joint will not activate at all.



**Step 6:** Now do the same procedure for all suspension joints. Finally check the results by running the simulation! (Maybe make the robot fall from a certain height or try to make it go through bumpy environments!) If the suspension joint is too diverging, try to reduce the spring constant and damping coefficients.

*Video Examples of Suspension Test

(https://youtu.be/W7wRbnabqNg)

(https://youtu.be/xFBX8skYYls)