

Lab 7 – Report

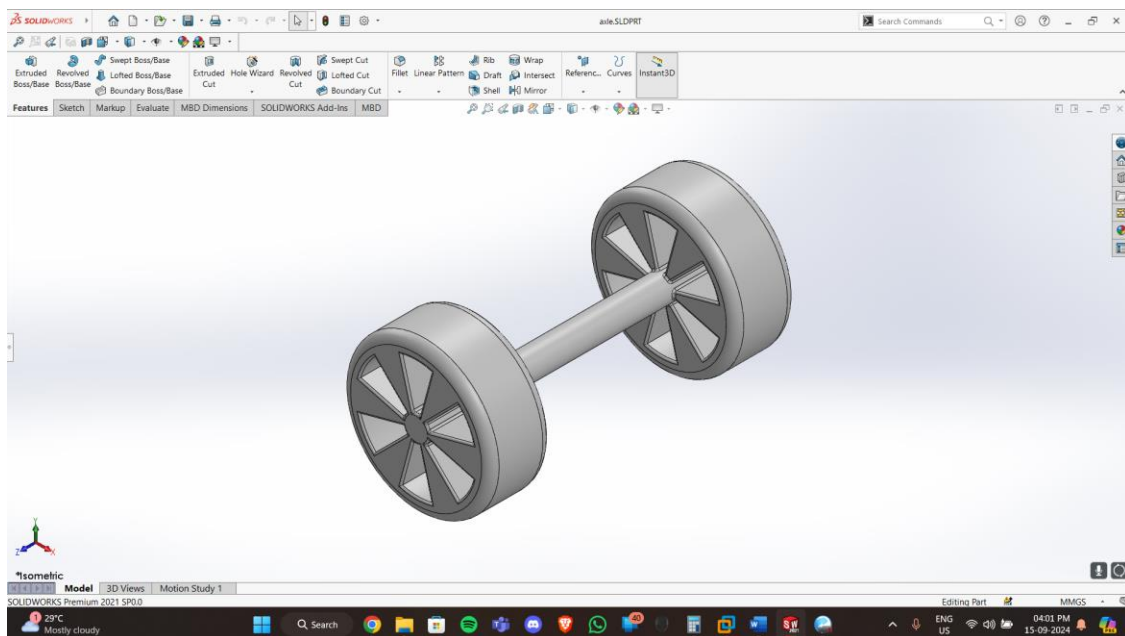
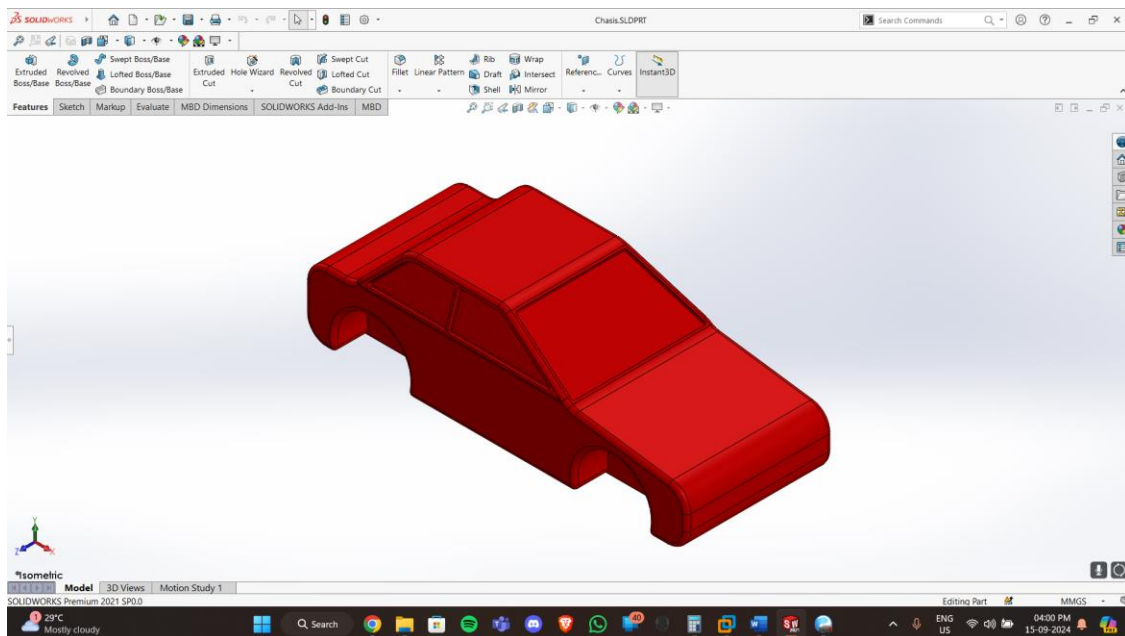
Meet Kansara – 220929270 Roll no. 54

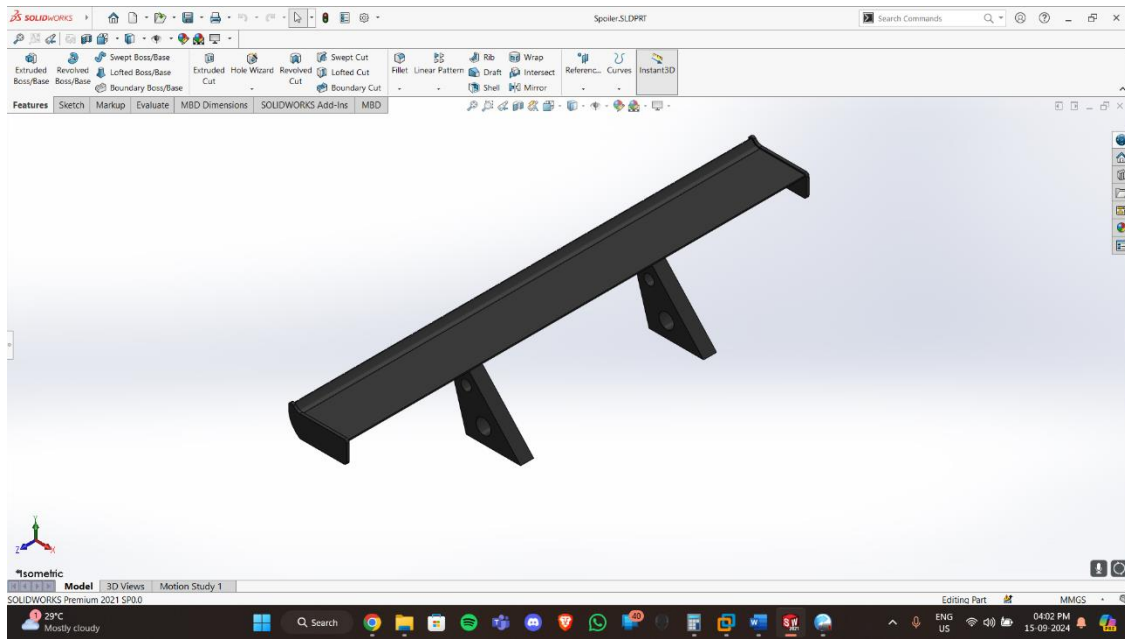
Aim: To design a model in SolidWorks, export it as a URDF, and then simulate it in RViz and Gazebo.

Code Execution and analysis:

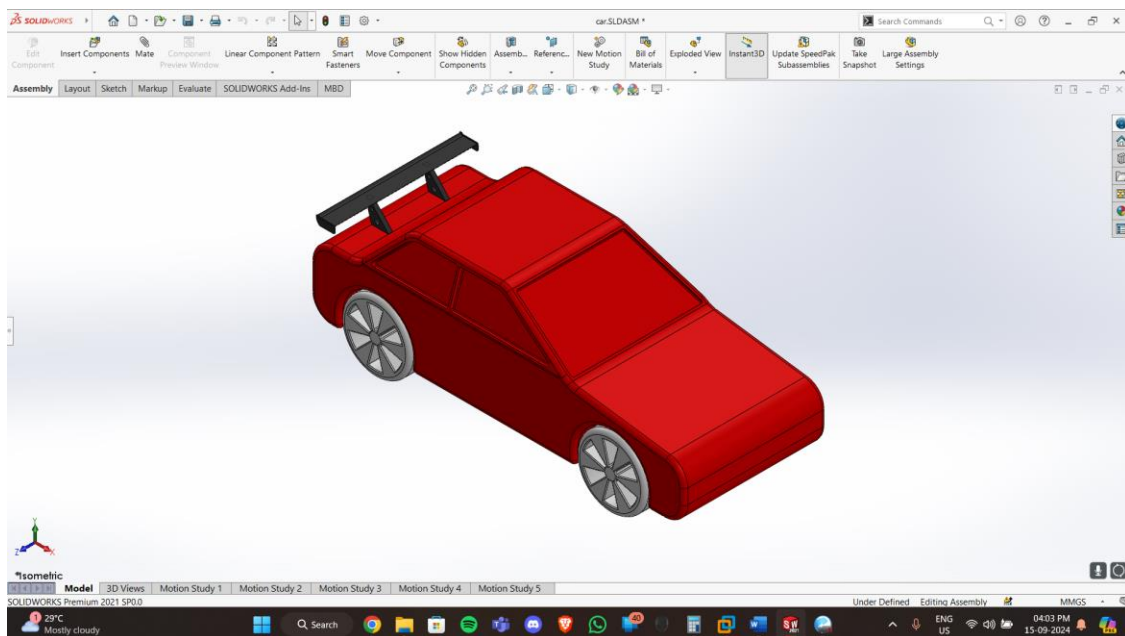
1. Creating a Car Model in SolidWorks

- Parts





- Assembly



2. Exporting model as URDF

```
<?xml version="1.0"?>
<robot name="car">
  <link name="chasis">
    <inertial>
      <origin xyz="0.080665 1.1872E-10 0.16605" rpy="0 0 0" />
      <mass value="56.78" />
      <inertia
        ixx="0.7602" ixy="2.8095E-09" ixz="0.15295"
        iyy="3.3323" iyz="-5.5178E-10" izz="3.5837" />
    </inertial>
    <visual>
```

```

    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://lab7/meshes/chasis.STL" />
    </geometry>
    <material name="Red">
      <color rgba="0.8 0.1 0.1 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://lab7/meshes/chasis.STL" />
    </geometry>
  </collision>
</link>

<link name="front_axle">
  <inertial>
    <origin xyz="4.2194E-13 6.6368E-12 -1.7421E-12" rpy="0 0 0" />
    <mass value="2.3273" />
    <inertia
      ixx="0.048559" ixy="7.7679E-13" ixz="4.7993E-14"
      iyy="0.0089055" iyz="-8.4669E-14" izz="0.048559" />
    </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://lab7/meshes/front_axle.STL" />
    </geometry>
    <material name="White">
      <color rgba="0.9 0.9 0.9 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://lab7/meshes/front_axle.STL" />
    </geometry>
  </collision>
</link>

<joint name="front_axle_joint" type="continuous">
  <origin xyz="0.3897 0 0.085" rpy="0 0 0" />
  <parent link="chasis" />
  <child link="front_axle" />
  <axis xyz="0 -1 0" />
</joint>

<link name="rear_axle">
  <inertial>
    <origin xyz="4.2191E-13 6.6371E-12 -1.742E-12" rpy="0 0 0" />
    <mass value="2.3273" />
    <inertia
      ixx="0.048559" ixy="7.768E-13" ixz="4.7991E-14"
      iyy="0.0089055" iyz="-8.4669E-14" izz="0.048559" />
    </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://lab7/meshes/rear_axle.STL" />
    </geometry>
    <material name="White">
      <color rgba="0.9 0.9 0.9 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://lab7/meshes/rear_axle.STL" />
    </geometry>
  </collision>
</link>

```

```

    </geometry>
  </collision>
</link>

<joint name="rear_axle_joint" type="continuous">
  <origin xyz="-0.18814 0 0.085" rpy="0 0 0" />
  <parent link="chasis" />
  <child link="rear_axle" />
  <axis xyz="0 -1 0" />
</joint>

<link name="spoiler">
  <inertial>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <mass value="0" />
    <inertia
      ixx="0" ixy="0" ixz="0"
      iyy="0" iyz="0" izz="0" />
  </inertial>
  <visual>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://lab7/meshes/spoiler.STL" />
    </geometry>
    <material name="Black">
      <color rgba="0.2 0.2 0.2 1" />
    </material>
  </visual>
  <collision>
    <origin xyz="0 0 0" rpy="0 0 0" />
    <geometry>
      <mesh filename="package://lab7/meshes/spoiler.STL" />
    </geometry>
  </collision>
</link>

<joint name="spoiler_joint" type="fixed">
  <origin xyz="-0.29658 0 0.22246" rpy="0 0 0" />
  <parent link="chasis" />
  <child link="spoiler" />
  <axis xyz="0 0 0" />
</joint>

<!-- Gazebo plugins -->
<gazebo>
  <plugin filename="libgazebo_ros2_control.so" name="gazebo_ros2_control">
    <parameters>/home/meet/ros2_ws/src/lab7/config/control.yaml</parameters>
    <robot_param>robot_description</robot_param>
    <robot_param_node>robot_state_publisher</robot_param_node>
  </plugin>
</gazebo>

<gazebo reference="chasis">
  <material>Gazebo/Red</material>
</gazebo>

<gazebo reference="front_axle">
  <material>Gazebo/White</material>
</gazebo>

<gazebo reference="rear_axle">
  <material>Gazebo/White</material>
</gazebo>

<gazebo reference="spoiler">
  <material>Gazebo/Black</material>
</gazebo>

<!-- ROS 2 Differential Drive Controller -->

```

```

<gazebo>
  <plugin name="diff_drive" filename="libgazebo_ros_diff_drive.so">
    <ros>
      <namespace>/</namespace>
    </ros>

    <!-- Wheel Information -->
    <left_joint>front_axle_joint</left_joint>
    <right_joint>front_axle_joint</right_joint>
    <wheel_separation>0.577</wheel_separation>
    <wheel_diameter>0.17</wheel_diameter>

    <!-- Limits -->
    <max_wheel_torque>20</max_wheel_torque>
    <max_wheel_acceleration>2.0</max_wheel_acceleration>

    <!-- Output -->
    <odometry_frame>odom</odometry_frame>
    <robot_base_frame>chasis</robot_base_frame>

    <publish_odom>true</publish_odom>
    <publish_odom_tf>true</publish_odom_tf>
    <publish_wheel_tf>true</publish_wheel_tf>

    <odometry_source>world</odometry_source>

    <!-- Update Rate -->
    <update_rate>50.0</update_rate>
  </plugin>
</gazebo>

</robot>

```

3. Visualizing the Car Model in RViz

```

from launch import LaunchDescription
from launch_ros.actions import Node

def generate_launch_description():
    urdf_file = '/home/meet/ros2_ws/src/lab7/urdf/car_rviz.urdf'

    joint_state_publisher_node = Node(
        package="joint_state_publisher_gui",
        executable="joint_state_publisher_gui",
        name="joint_state_publisher_gui",
        output="screen",
        arguments=[urdf_file]
    )

    robot_state_publisher_node = Node(
        package="robot_state_publisher",
        executable="robot_state_publisher",
        name="robot_state_publisher",
        output="screen",
        arguments=[urdf_file]
    )

    rviz_node = Node(
        package="rviz2",
        executable="rviz2",
        name="rviz2",
        output="screen"
    )

    nodes_to_run = [
        joint_state_publisher_node,

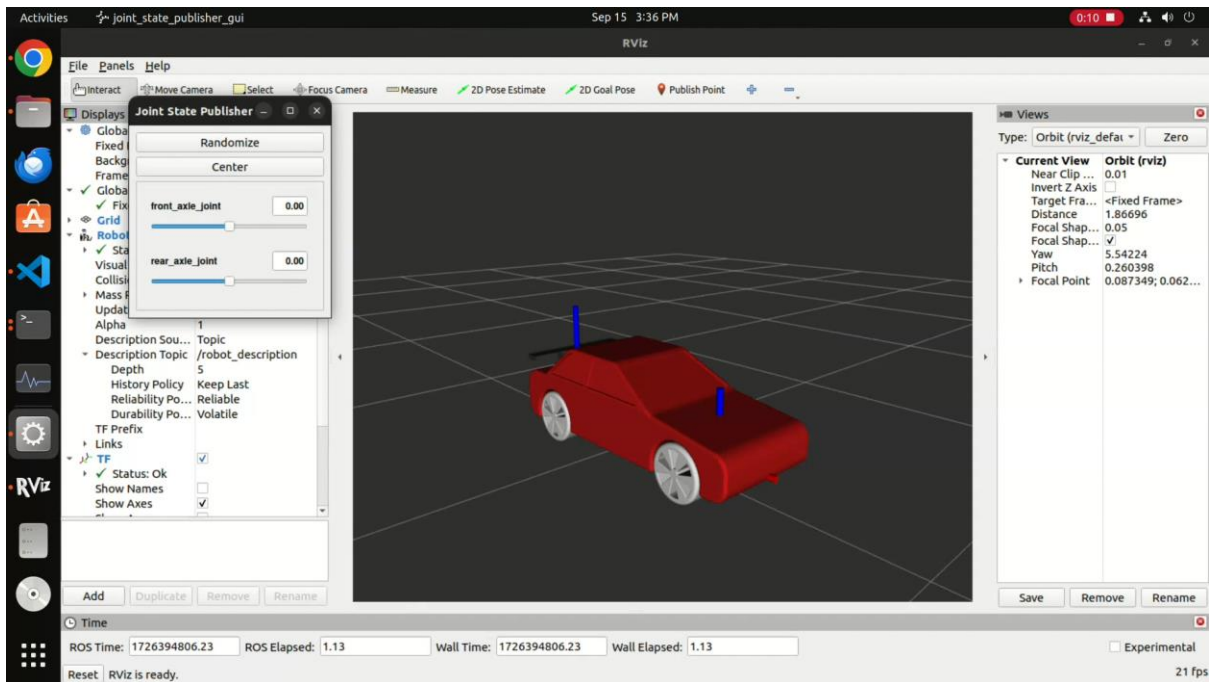
```

```

    robot_state_publisher_node,
    rviz_node
]

return LaunchDescription(nodes_to_run)

```



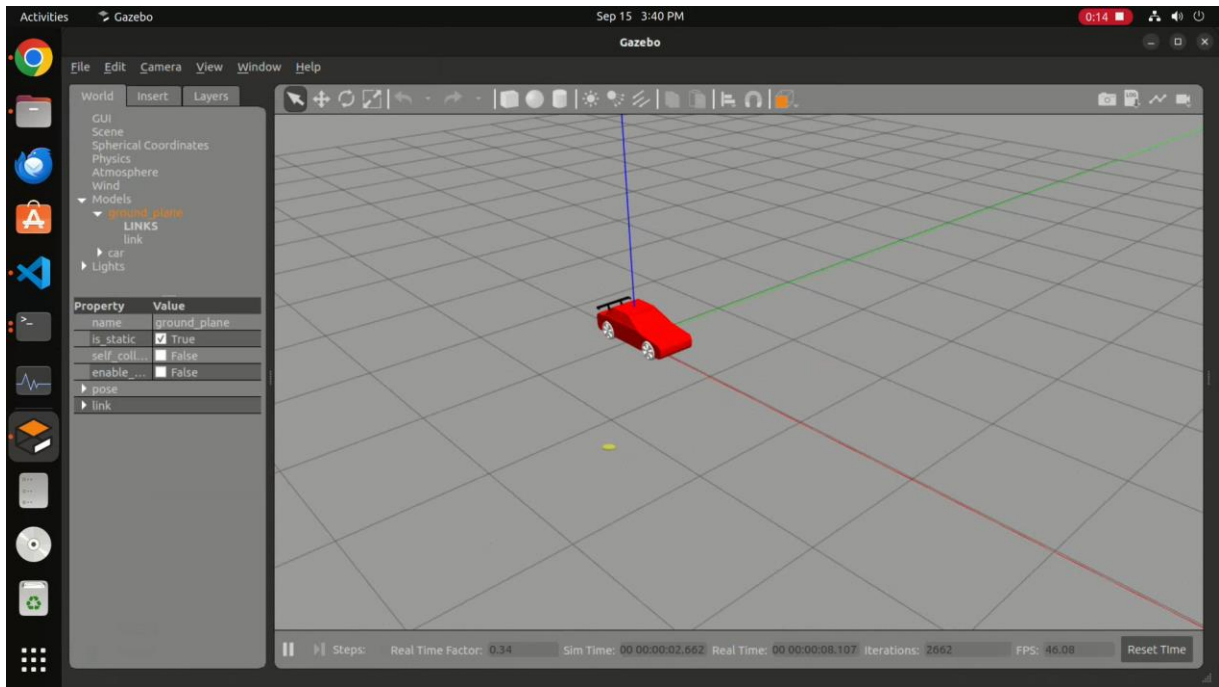
4. Simulating the Car Model in Gazebo

```

import os
from launch import LaunchDescription
from launch.actions import ExecuteProcess
from launch_ros.actions import Node

def generate_launch_description():
    urdf_file = '/home/meet/ros2_ws/src/lab7/urdf/car.urdf'
    return LaunchDescription(
        [
            ExecuteProcess(
                cmd=["gazebo", "-s", "libgazebo_ros_factory.so"],
                output="screen",
            ),
            Node(
                package="gazebo_ros",
                executable="spawn_entity.py",
                arguments=["-entity", "car", "-b", "-file", urdf_file],
                output="screen",
            ),
            Node(
                package="robot_state_publisher",
                executable="robot_state_publisher",
                arguments=[urdf_file],
                output="screen",
            ),
        ]
    )

```



Conclusion By successfully designing any model in SolidWorks, exporting it as a URDF, and simulating it in RViz and Gazebo, we can effectively test and visualize its performance in a virtual environment.