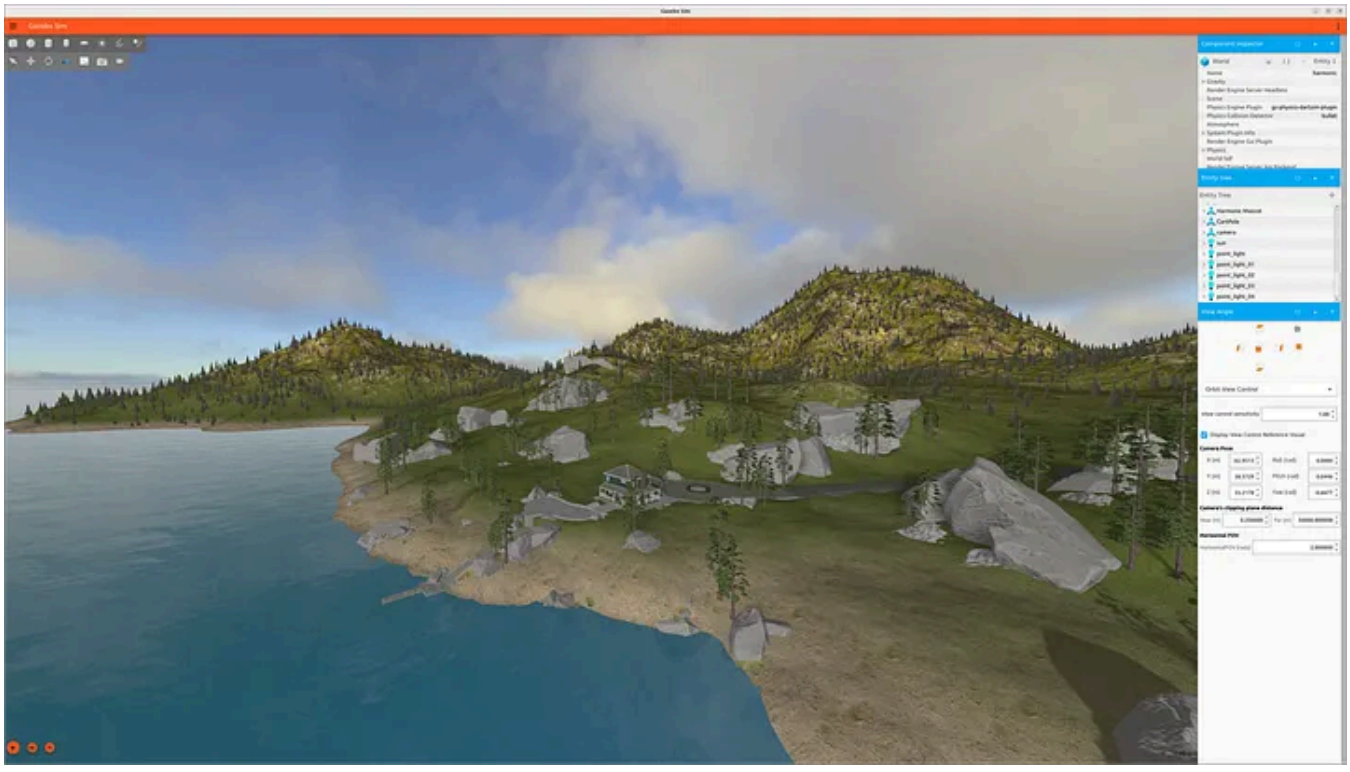
Gazebo Sim (Harmonic) Plugins and Sensors for ROS2

6 min read · Oct 11, 2024

Ali Tekeş Follow

Listen Share

You can access code examples from this GitHub repository. The repository includes a variety of sensor applications for simulation. I'm thinking about adding content on how to use GazeboSim sensor data with **ROS2** and **RViz2**. If you're interested in integrating with ROS2, let me know with a like or comment. Stay tuned, and happy reading!



Gazebo Sim Harmonic

Sensors

GPU Lidar

```
<sensor name="gpu_lidar" type="gpu_lidar">
        <lidar>
          <scan>
            <horizontal>
              <samples>1800</samples>
              <resolution>1</resolution>
              <min_angle>-3.1459</min_angle>
              <max_angle>3.1459</max_angle>
            </horizontal>
            <vertical>
              <samples>16</samples>
              <resolution>1</resolution>
              <min_angle>-0.261799</min_angle>
              <max_angle>0.261799</max_angle>
            </vertical>
          </scan>
          <range>
            <min>0.05</min>
            <max>100</max>
            <resolution>0.01</resolution>
          </range>
          <noise>
            <type>gaussian</type>
            <mean>0</mean>
            <stddev>0.01</stddev>
          </noise>
        </lidar>
        <gz_frame_id>saye/sensor_link</gz_frame_id>
        <topic>laser_scan</topic>
        <visualize>true</visualize>
        <update_rate>10</update_rate>
      </sensor>
```

IMU

```
      <!-- IMU  -->
      <sensor name="imu_sensor" type="imu">
        <always_on>1</always_on>
        <update_rate>50</update_rate>
        <imu>
          <enable_orientation>0</enable_orientation>
          <angular_velocity>
            <x>
              <noise type="gaussian">
```

```
          <mean>0</mean>
          <stddev>0.009</stddev>
          <bias_mean>0.00075</bias_mean>
          <bias_stddev>0.005</bias_stddev>
          <dynamic_bias_stddev>0.00002</dynamic_bias_stddev>
          <dynamic_bias_correlation_time>400.0</dynamic_bias_correlation_
          <precision>0.00025</precision>
        </noise>
      </x>
      <y>
        <noise type="gaussian">
          <mean>0</mean>
          <stddev>0.009</stddev>
          <bias_mean>0.00075</bias_mean>
          <bias_stddev>0.005</bias_stddev>
          <dynamic_bias_stddev>0.00002</dynamic_bias_stddev>
          <dynamic_bias_correlation_time>400.0</dynamic_bias_correlation_
          <precision>0.00025</precision>
        </noise>
      </y>
      <z>
        <noise type="gaussian">
          <mean>0</mean>
          <stddev>0.009</stddev>
          <bias_mean>0.00075</bias_mean>
          <bias_stddev>0.005</bias_stddev>
          <dynamic_bias_stddev>0.00002</dynamic_bias_stddev>
          <dynamic_bias_correlation_time>400.0</dynamic_bias_correlation_
          <precision>0.00025</precision>
        </noise>
      </z>
    </angular_velocity>
    <linear_acceleration>
      <x>
        <noise type="gaussian">
          <mean>0</mean>
          <stddev>0.021</stddev>
          <bias_mean>0.05</bias_mean>
          <bias_stddev>0.0075</bias_stddev>
          <dynamic_bias_stddev>0.000375</dynamic_bias_stddev>
          <dynamic_bias_correlation_time>175.0</dynamic_bias_correlation_
          <precision>0.005</precision>
        </noise>
      </x>
      <y>
        <noise type="gaussian">
          <mean>0</mean>
          <stddev>0.021</stddev>
          <bias_mean>0.05</bias_mean>
          <bias_stddev>0.0075</bias_stddev>
          <dynamic_bias_stddev>0.000375</dynamic_bias_stddev>
          <dynamic_bias_correlation_time>175.0</dynamic_bias_correlation_
          <precision>0.005</precision>
```

```
            </noise>
          </y>
          <z>
            <noise type="gaussian">
              <mean>0</mean>
              <stddev>0.021</stddev>
              <bias_mean>0.05</bias_mean>
              <bias_stddev>0.0075</bias_stddev>
              <dynamic_bias_stddev>0.000375</dynamic_bias_stddev>
              <dynamic_bias_correlation_time>175.0</dynamic_bias_correlation_
              <precision>0.005</precision>
            </noise>
          </z>
        </linear_acceleration>
      </imu>
    </sensor>
```

Intel Real Sense D435 RGBD Camera

```
    <!-- Based on Intel realsense D435 (intrinsics and distortion not modelec
    <sensor name="rs_front" type="rgbd_camera">
      <pose>0.122 0 0.257 0 0.46 0</pose>
      <always_on>1</always_on>
      <update_rate>30</update_rate>
      <camera name="rs_front">
        <horizontal_fov>1.50098</horizontal_fov>
        <lens>
          <intrinsics>
            <!-- fx = fy = width / ( 2 * tan (hfov / 2 ) ) -->
            <fx>343.159</fx>
            <fy>343.159</fy>
            <!-- cx = ( width - 1 ) / 2 -->
            <cx>319.5</cx>
            <!-- cy = ( height - 1 ) / 2 -->
            <cy>179.5</cy>
            <s>0</s>
          </intrinsics>
        </lens>
        <distortion>
          <k1>0.0</k1>
          <k2>0.0</k2>
          <k3>0.0</k3>
          <p1>0.0</p1>
          <p2>0.0</p2>
          <center>0.5 0.5</center>
        </distortion>
        <image>
          <width>640</width>
```

```
        <height>360</height>
        <format>R8G8B8</format>
      </image>
      <clip>
        <near>0.01</near>
        <far>300</far>
      </clip>
      <depth_camera>
        <clip>
          <near>0.1</near>
          <far>10</far>
        </clip>
      </depth_camera>
      <noise>
        <type>gaussian</type>
        <mean>0</mean>
        <stddev>0.007</stddev>
      </noise>
    </camera>
  </sensor>
```

> *Front Camera*

```
  <sensor name="camera_front" type="camera">
    <pose>0.111 0.03 0.235 0 0 0</pose>
    <always_on>1</always_on>
    <update_rate>25</update_rate>
    <camera name="camera_front">
      <horizontal_fov>1.658</horizontal_fov>
      <lens>
        <intrinsics>
          <!-- fx = fy = width / ( 2 * tan (hfov / 2 ) ) -->
          <fx>329.9</fx>
          <fy>329.9</fy>
          <!-- cx = ( width - 1 ) / 2 -->
          <cx>359.5</cx>
          <!-- cy = ( height - 1 ) / 2 -->
          <cy>539.5</cy>
          <s>0</s>
        </intrinsics>
      </lens>
      <distortion>
        <k1>0.0</k1>
        <k2>0.0</k2>
        <k3>0.0</k3>
        <p1>0.0</p1>
        <p2>0.0</p2>
        <center>0.5 0.5</center>
```

```
      </distortion>
      <image>
        <width>720</width>
        <height>1080</height>
        <format>R8G8B8</format>
      </image>
      <clip>
        <near>0.01</near>
        <far>300</far>
      </clip>
      <noise>
        <type>gaussian</type>
        <mean>0</mean>
        <stddev>0.007</stddev>
      </noise>
    </camera>
  </sensor>
```

Rear Camera

```
  <sensor name="camera_rear" type="camera">
    <pose>0.069 -0.03 0.235 0 0 3.141592653589793</pose>
    <always_on>1</always_on>
    <update_rate>25</update_rate>
    <camera name="camera_rear">
      <horizontal_fov>1.658</horizontal_fov>
      <lens>
        <intrinsics>
          <!-- fx = fy = width / ( 2 * tan (hfov / 2 ) ) -->
          <fx>329.9</fx>
          <fy>329.9</fy>
          <!-- cx = ( width - 1 ) / 2 -->
          <cx>359.5</cx>
          <!-- cy = ( height - 1 ) / 2 -->
          <cy>539.5</cy>
          <s>0</s>
        </intrinsics>
      </lens>
      <distortion>
        <k1>0.0</k1>
        <k2>0.0</k2>
        <k3>0.0</k3>
        <p1>0.0</p1>
        <p2>0.0</p2>
        <center>0.5 0.5</center>
      </distortion>
      <image>
        <width>720</width>
```

```
        <height>1080</height>
        <format>R8G8B8</format>
      </image>
      <clip>
        <near>0.01</near>
        <far>300</far>
      </clip>
      <noise>
        <type>gaussian</type>
        <mean>0</mean>
        <stddev>0.007</stddev>
      </noise>
    </camera>
  </sensor>
```

Camera Left


```
  <sensor name="camera_left" type="camera">
    <pose>0.085 0.09 0.27 0 0 1.5707963267948966</pose>
    <always_on>1</always_on>
    <update_rate>25</update_rate>
    <camera name="camera_left">
      <horizontal_fov>1.658</horizontal_fov>
      <lens>
        <intrinsics>
          <!-- fx = fy = width / ( 2 * tan (hfov / 2 ) ) -->
          <fx>329.9</fx>
          <fy>329.9</fy>
          <!-- cx = ( width - 1 ) / 2 -->
          <cx>359.5</cx>
          <!-- cy = ( height - 1 ) / 2 -->
          <cy>539.5</cy>
          <s>0</s>
        </intrinsics>
      </lens>
```

```
      <distortion>
        <k1>0.0</k1>
        <k2>0.0</k2>
        <k3>0.0</k3>
        <p1>0.0</p1>
        <p2>0.0</p2>
        <center>0.5 0.5</center>
      </distortion>
      <image>
        <width>720</width>
        <height>1080</height>
        <format>R8G8B8</format>
      </image>
      <clip>
        <near>0.01</near>
        <far>300</far>
      </clip>
      <noise>
        <type>gaussian</type>
        <mean>0</mean>
        <stddev>0.007</stddev>
      </noise>
    </camera>
  </sensor>
```

Camera Right

```
  <sensor name="camera_right" type="camera">
    <pose>0.085 -0.09 0.27 0 0 -1.5707963267948966</pose>
    <always_on>1</always_on>
    <update_rate>25</update_rate>
    <camera name="camera_right">
      <horizontal_fov>1.658</horizontal_fov>
      <lens>
        <intrinsics>
          <!-- fx = fy = width / ( 2 * tan (hfov / 2 ) ) -->
          <fx>329.9</fx>
          <fy>329.9</fy>
          <!-- cx = ( width - 1 ) / 2 -->
          <cx>359.5</cx>
          <!-- cy = ( height - 1 ) / 2 -->
          <cy>539.5</cy>
          <s>0</s>
        </intrinsics>
      </lens>
      <distortion>
        <k1>0.0</k1>
        <k2>0.0</k2>
```

```
            <k3>0.0</k3>
            <p1>0.0</p1>
            <p2>0.0</p2>
            <center>0.5 0.5</center>
          </distortion>
          <image>
            <width>720</width>
            <height>1080</height>
            <format>R8G8B8</format>
          </image>
          <clip>
            <near>0.01</near>
            <far>300</far>
          </clip>
          <noise>
            <type>gaussian</type>
            <mean>0</mean>
            <stddev>0.007</stddev>
          </noise>
        </camera>
      </sensor>
```

Magnetometer

```
      <sensor name="magnetometer" type="magnetometer">
        <always_on>1</always_on>
        <update_rate>50</update_rate>
        <magnetometer>
          <x>
            <noise type="gaussian">
              <mean>0.000000080</mean>
              <bias_mean>0.000000400</bias_mean>
            </noise>
          </x>
          <y>
            <noise type="gaussian">
              <mean>0.000000080</mean>
              <bias_mean>0.000000400</bias_mean>
            </noise>
          </y>
          <z>
            <noise type="gaussian">
              <mean>0.000000080</mean>
              <bias_mean>0.000000400</bias_mean>
            </noise>
          </z>
```

```
    </magnetometer>
  </sensor>
```

Thermal Camera

```
<sensor name="thermal_camera" type="thermal">
  <pose>0.44 -0.17 0.574 0 0 0</pose>
  <camera name="thermal_camera">
    <horizontal_fov>0.628</horizontal_fov>
    <lens>
      <intrinsics>
        <!-- fx = fy = width / ( 2 * tan (hfov / 2 ) ) -->
        <fx>317.173</fx>
        <fy>317.173</fy>
        <!-- cx = ( width - 1 ) / 2 -->
        <cx>102.5</cx>
        <!-- cy = ( height - 1 ) / 2 -->
        <cy>77.5</cy>
        <s>0</s>
      </intrinsics>
    </lens>
    <image>
      <width>206</width>
      <height>156</height>
      <format>L8</format>
    </image>
    <clip>
      <near>0.1</near>
      <far>100</far>
    </clip>
  </camera>
  <always_on>1</always_on>
  <update_rate>9</update_rate>
  <plugin
    filename="gz-sim-thermal-sensor-system"
    name="gz::sim::systems::ThermalSensor">
    <min_temp>253.15</min_temp>
    <max_temp>673.15</max_temp>
    <resolution>1.6</resolution>
  </plugin>
</sensor>
```

NavSat Sensor

```
        <sensor name="navsat_sensor" type="navsat">
          <always_on>1</always_on>
          <update_rate>30</update_rate>
        </sensor>
```

Air Pressure

```
    <sensor name="air_pressure" type="air_pressure">
          <always_on>1</always_on>
          <update_rate>20</update_rate>
          <air_pressure>
            <reference_altitude>0</reference_altitude>
            <noise type="gaussian">
              <mean>0.00000008</mean>
            </noise>
          </air_pressure>
        </sensor>
```

. . .

Contact Sensor

```
          <sensor name='sensor_contact' type='contact'>
            <contact>
              <collision>collision_sphere</collision>
              <topic>/contact_example</topic>
            </contact>
            <always_on>1</always_on>
            <update_rate>100</update_rate>
          </sensor>
```

Plugins

```
<plugin
  filename="gz-sim-physics-system"
  name="gz::sim::systems::Physics">
```

```
</plugin>
<plugin
  filename="gz-sim-air-pressure-system"
  name="gz::sim::systems::AirPressure">
</plugin>
<plugin
  filename="gz-sim-altimeter-system"
  name="gz::sim::systems::Altimeter">
</plugin>
<plugin
  filename="gz-sim-imu-system"
  name="gz::sim::systems::Imu">
</plugin>
<plugin
  filename="gz-sim-magnetometer-system"
  name="gz::sim::systems::Magnetometer">
</plugin>
<plugin
  filename="gz-sim-forcetorque-system"
  name="gz::sim::systems::ForceTorque">
  <gz::system_priority>10</gz::system_priority>
</plugin>
<plugin
  filename="gz-sim-user-commands-system"
  name="gz::sim::systems::UserCommands">
</plugin>
<plugin
  filename="gz-sim-scene-broadcaster-system"
  name="gz::sim::systems::SceneBroadcaster">
</plugin>

<plugin
  filename="gz-sim-joint-controller-system"
  name="gz::sim::systems::JointController">
  <joint_name>j1</joint_name>
  <initial_velocity>1.0</initial_velocity>
</plugin>

<plugin
  filename="gz-sim-pose-publisher-system"
  name="gz::sim::systems::PosePublisher">
  <publish_link_pose>true</publish_link_pose>
  <use_pose_vector_msg>true</use_pose_vector_msg>
  <static_publisher>true</static_publisher>
  <static_update_frequency>1</static_update_frequency>
</plugin>

<plugin
  filename="gz-sim-odometry-publisher-system"
  name="gz::sim::systems::OdometryPublisher">
  <odom_frame>vehicle/odom</odom_frame>
  <robot_base_frame>vehicle</robot_base_frame>
</plugin>
```

```
<plugin
  filename="gz-sim-diff-drive-system"
  name="gz::sim::systems::DiffDrive">
  <left_joint>front_left_wheel_joint</left_joint>
  <left_joint>rear_left_wheel_joint</left_joint>
  <right_joint>front_right_wheel_joint</right_joint>
  <right_joint>rear_right_wheel_joint</right_joint>
  <wheel_separation>1.25</wheel_separation>
  <wheel_radius>0.3</wheel_radius>
  <topic>cmd_vel</topic>
</plugin>

<plugin filename="gz-sim-triggered-publisher-system"
  name="gz::sim::systems::TriggeredPublisher">
  <input type="gz.msgs.Empty" topic="/reset_robot"/>
  <output type="gz.msgs.Twist" topic="/cmd_vel">
      linear: {x: 0}
  </output>
  <service
    name="/world/triggered_publisher/set_pose"
    reqType="gz.msgs.Pose"
    repType="gz.msgs.Boolean"
    timeout="3000"
    reqMsg="name: 'blue_vehicle', id: 8, position: {x: -3, z: 1}">
  </service>
</plugin>

<world>
  ...
  <plugin filename="gz-sim-altimeter-system"
    name="gz::sim::systems::Altimeter">
  </plugin>
  ...
</world>

<plugin
  filename="gz-sim-diff-drive-system"
  name="gz::sim::systems::DiffDrive">
  <left_joint>left_wheel_joint</left_joint>
  <right_joint>right_wheel_joint</right_joint>
  <wheel_separation>1.25</wheel_separation>
  <wheel_radius>0.3</wheel_radius>
  <odom_publish_frequency>1</odom_publish_frequency>
  <max_linear_acceleration>1</max_linear_acceleration>
  <min_linear_acceleration>-1</min_linear_acceleration>
  <max_angular_acceleration>2</max_angular_acceleration>
  <min_angular_acceleration>-2</min_angular_acceleration>
  <max_linear_velocity>0.5</max_linear_velocity>
  <min_linear_velocity>-0.5</min_linear_velocity>
  <max_angular_velocity>1</max_angular_velocity>
  <min_angular_velocity>-1</min_angular_velocity>
</plugin>
```

```
<plugin
  filename="gz-sim-ackermann-steering-system"
  name="gz::sim::systems::AckermannSteering">
  <left_joint>front_left_wheel_joint</left_joint>
  <left_joint>rear_left_wheel_joint</left_joint>
  <right_joint>front_right_wheel_joint</right_joint>
  <right_joint>rear_right_wheel_joint</right_joint>
  <left_steering_joint>front_left_wheel_steering_joint</left_steering_joint>
  <right_steering_joint>front_right_wheel_steering_joint</right_steering_joint>
  <kingpin_width>1.0</kingpin_width>
  <steering_limit>0.5</steering_limit>
  <wheel_base>1.0</wheel_base>
  <wheel_separation>1.25</wheel_separation>
  <wheel_radius>0.3</wheel_radius>
  <min_velocity>-1</min_velocity>
  <max_velocity>1</max_velocity>
  <min_acceleration>-3</min_acceleration>
  <max_acceleration>3</max_acceleration>
</plugin>

<plugin
  filename="gz-sim-mecanum-drive-system"
  name="gz::sim::systems::MecanumDrive">
  <front_left_joint>front_left_wheel_joint</front_left_joint>
  <front_right_joint>front_right_wheel_joint</front_right_joint>
  <back_left_joint>rear_left_wheel_joint</back_left_joint>
  <back_right_joint>rear_right_wheel_joint</back_right_joint>
  <wheel_separation>1.25</wheel_separation>
  <wheelbase>1.511</wheelbase>
  <wheel_radius>0.3</wheel_radius>
  <min_acceleration>-5</min_acceleration>
  <max_acceleration>5</max_acceleration>
</plugin>

<plugin filename="ImageDisplay" name="Visible 2D">
  <topic>your_image_topic</topic>
  <gz-gui>
    <title>Visible 2D</title>
    <property key="state" type="string">docked</property>
    <property type="double" key="height">400</property>
    <property type="double" key="width">600</property>
  </gz-gui>
</plugin>

<gui fullscreen="0">

  <!-- 3D scene -->
  <plugin filename="MinimalScene" name="3D View">
    <gz-gui>
      <title>3D View</title>
      <property type="bool" key="showTitleBar">false</property>
      <property type="string" key="state">docked</property>
```

```
    </gz-gui>

    <engine>ogre2</engine>
    <scene>scene</scene>
    <ambient_light>1.0 1.0 1.0</ambient_light>
    <background_color>0.8 0.8 0.8</background_color>
    <camera_pose>-6 0 6 0 0.5 0</camera_pose>
  </plugin>

  <!-- Plugins that add functionality to the scene -->
  <plugin filename="EntityContextMenuPlugin" name="Entity context menu">
    <gz-gui>
      <property key="state" type="string">floating</property>
      <property key="width" type="double">5</property>
      <property key="height" type="double">5</property>
      <property key="showTitleBar" type="bool">false</property>
    </gz-gui>
  </plugin>
  <plugin filename="GzSceneManager" name="Scene Manager">
    <gz-gui>
      <property key="resizable" type="bool">false</property>
      <property key="width" type="double">5</property>
      <property key="height" type="double">5</property>
      <property key="state" type="string">floating</property>
      <property key="showTitleBar" type="bool">false</property>
    </gz-gui>
  </plugin>
  <plugin filename="InteractiveViewControl" name="Interactive view control">
    <gz-gui>
      <property key="resizable" type="bool">false</property>
      <property key="width" type="double">5</property>
      <property key="height" type="double">5</property>
      <property key="state" type="string">floating</property>
      <property key="showTitleBar" type="bool">false</property>
    </gz-gui>
  </plugin>
  <plugin filename="CameraTracking" name="Camera Tracking">
    <gz-gui>
      <property key="resizable" type="bool">false</property>
      <property key="width" type="double">5</property>
      <property key="height" type="double">5</property>
      <property key="state" type="string">floating</property>
      <property key="showTitleBar" type="bool">false</property>
    </gz-gui>
  </plugin>

  <!-- World control -->
  <plugin filename="WorldControl" name="World control">
    <gz-gui>
      <title>World control</title>
      <property type="bool" key="showTitleBar">false</property>
      <property type="bool" key="resizable">false</property>
      <property type="double" key="height">72</property>
```

```
      <property type="double" key="z">1</property>

      <property type="string" key="state">floating</property>
      <anchors target="3D View">
        <line own="left" target="left"/>
        <line own="bottom" target="bottom"/>
      </anchors>
    </gz-gui>

    <play_pause>true</play_pause>
    <step>true</step>
    <start_paused>true</start_paused>
    <use_event>true</use_event>

  </plugin>

  <!-- World statistics -->
  <plugin filename="WorldStats" name="World stats">
    <gz-gui>
      <title>World stats</title>
      <property type="bool" key="showTitleBar">false</property>
      <property type="bool" key="resizable">false</property>
      <property type="double" key="height">110</property>
      <property type="double" key="width">290</property>
      <property type="double" key="z">1</property>

      <property type="string" key="state">floating</property>
      <anchors target="3D View">
        <line own="right" target="right"/>
        <line own="bottom" target="bottom"/>
      </anchors>
    </gz-gui>

    <sim_time>true</sim_time>
    <real_time>true</real_time>
    <real_time_factor>true</real_time_factor>
    <iterations>true</iterations>
  </plugin>

  <plugin filename="ImageDisplay" name="Full 2D">
    <topic>topic_name</topic>
    <gz-gui>
      <title>Full 2D</title>
      <property key="state" type="string">docked</property>
      <property type="double" key="height">400</property>
      <property type="double" key="width">600</property>
    </gz-gui>
  </plugin>

  <plugin filename="ImageDisplay" name="Visible 2D">
    <topic>topic_name</topic>
    <gz-gui>
      <title>Visible 2D</title>
```

```
      <property key="state" type="string">docked</property>
      <property type="double" key="height">400</property>
      <property type="double" key="width">600</property>
    </gz-gui>
  </plugin>
```



```
      <property key="state" type="string">docked</property>
      <property type="double" key="height">400</property>
      <property type="double" key="width">600</property>
    </gz-gui>
  </plugin>

</gui>

<plugin filename="gz-sim-contact-system" name="gz::sim::systems::Contact"/>
```


Gazebosim Plugin | Gazebosim Sensor | Harmonic | Ros2 | Gazebo

Written by Ali Tekeş

Sakarya University - Computer Engineering Student (3/4) - TUBITAK Intern Researcher

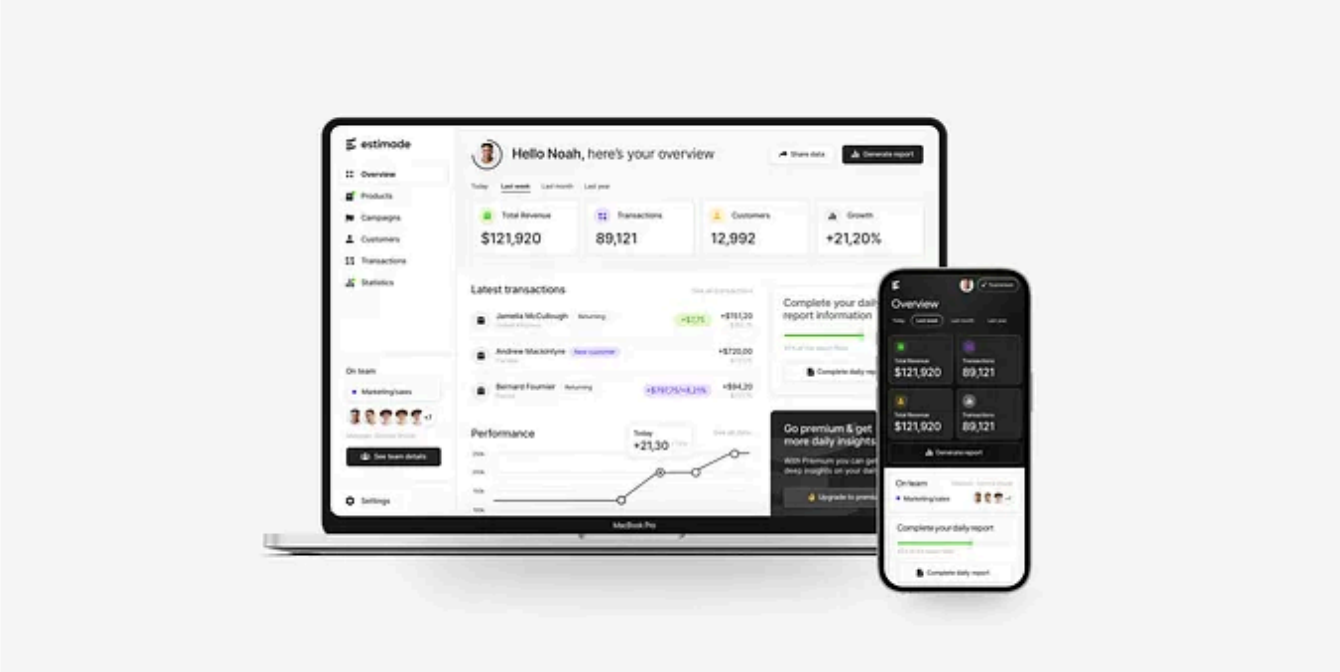
Responses (1)

Christyj
Feb 1

great content. can you explain how to choose the plugin and integrate it to a robot such that it can be visualized with rviz2 and the latest gz sim. Thanks

 1 reply Reply

Recommended from Medium

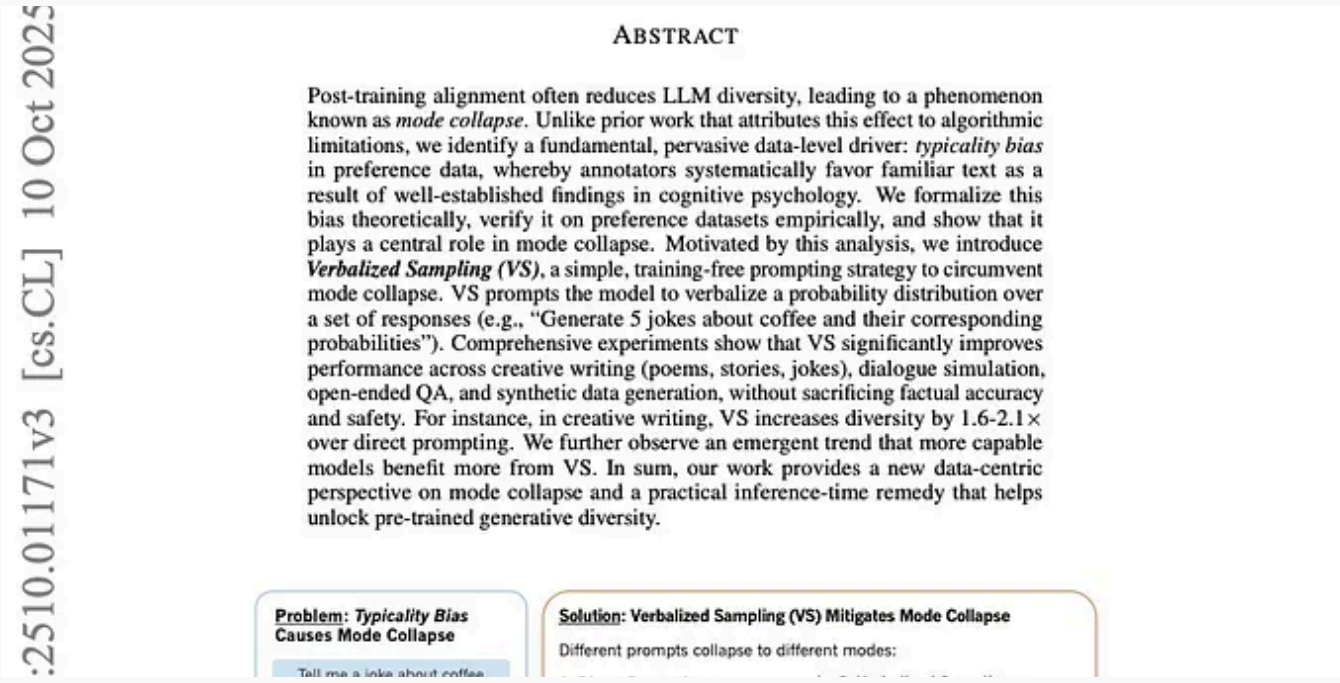


Michal Malewicz

The End of Dashboards and Design Systems

Design is becoming quietly human again.

Nov 26 2.3K 79

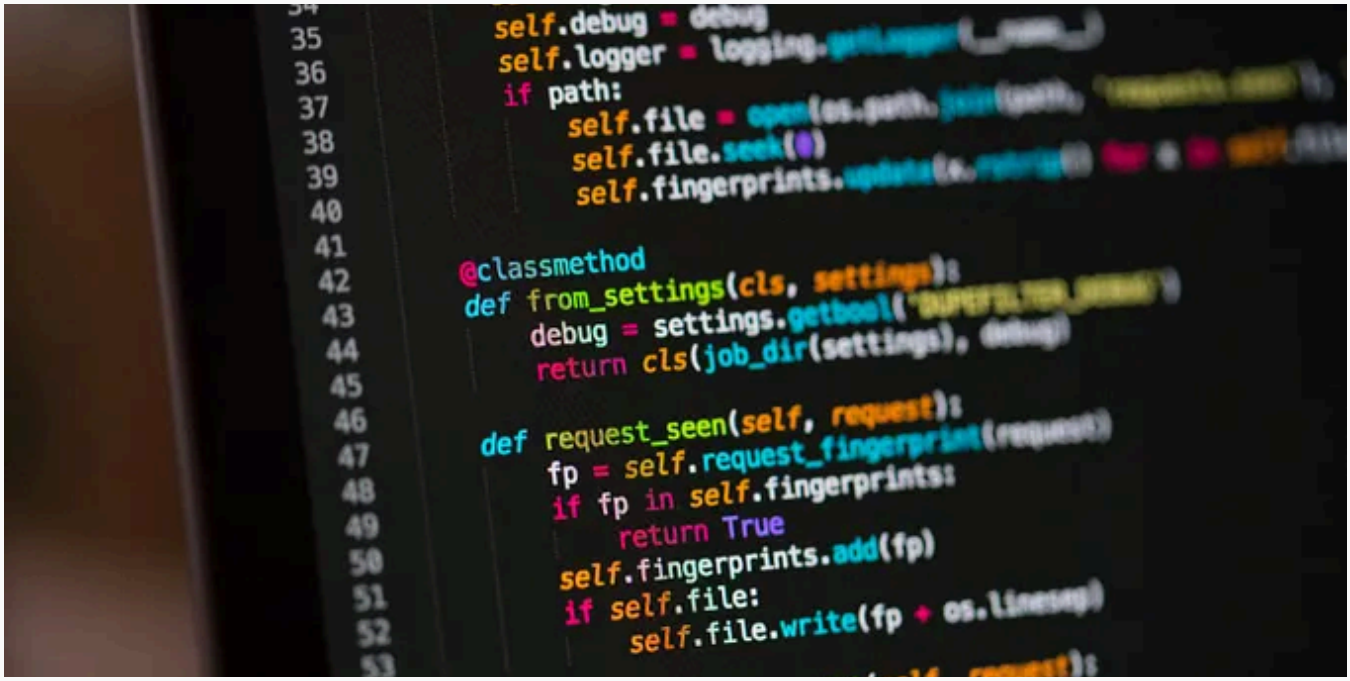


In Generative AI by Adham Khaled

Stanford Just Killed Prompt Engineering With 8 Words (And I Can't Believe It Worked)

ChatGPT keeps giving you the same boring response? This new technique unlocks 2× more creativity from ANY AI model—no training required...

Oct 20 19.5K 476

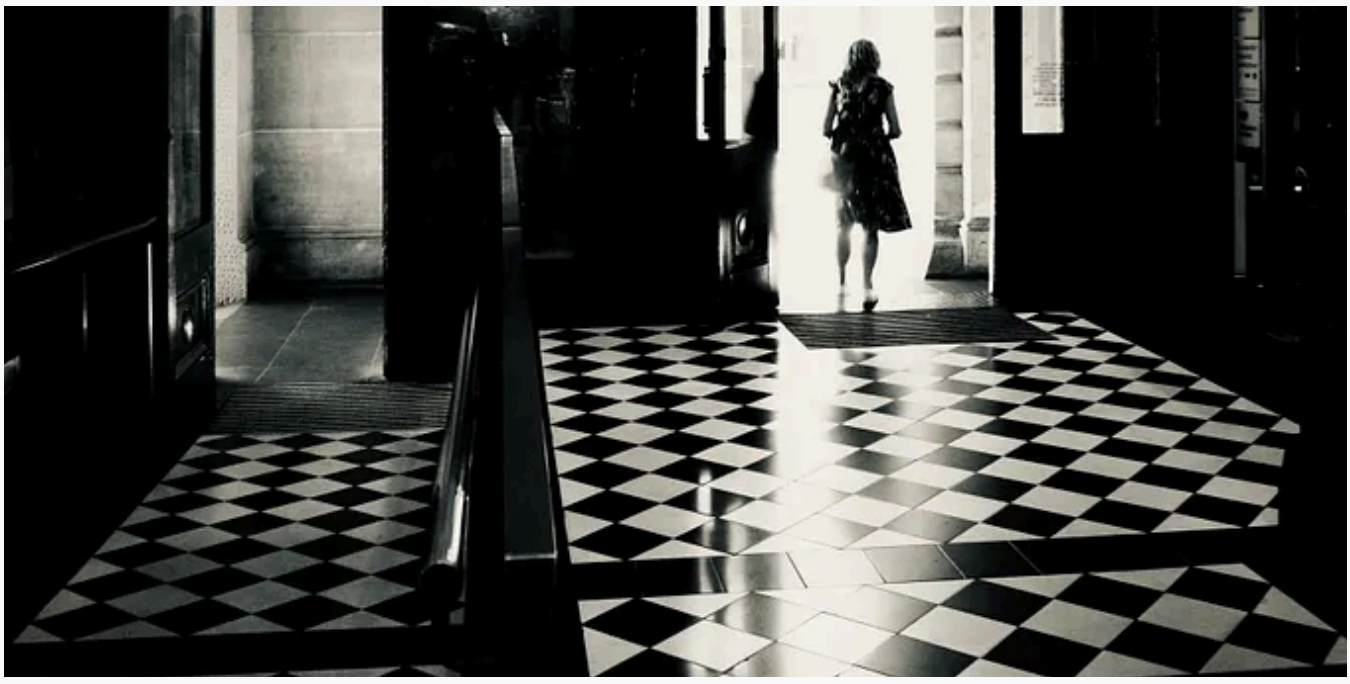


In Data And Beyond by Partha Mishra

A Beginner's Guide to Gradio in Python

Part 3: Basics of GUI Development with Gradio

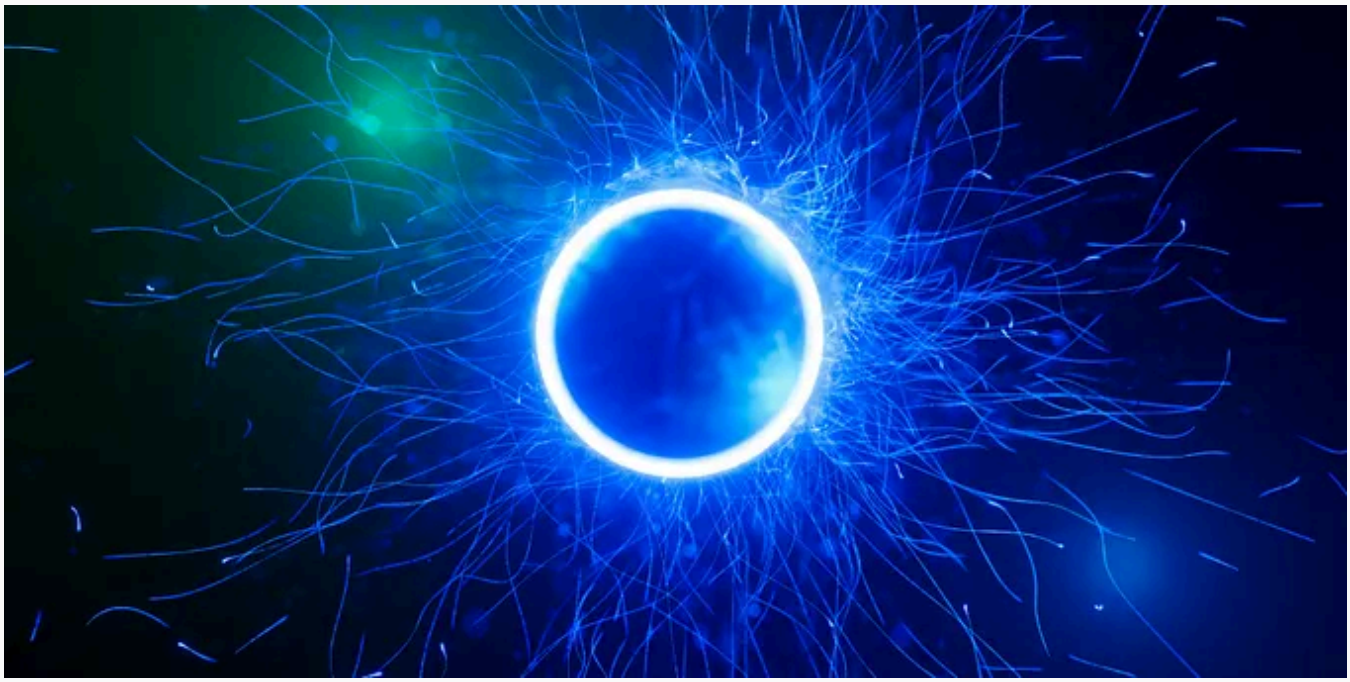
Nov 8 109



In /powerbrick by Jesse Kim

Power Apps and Power Automate, synchronised

Jul 6 4

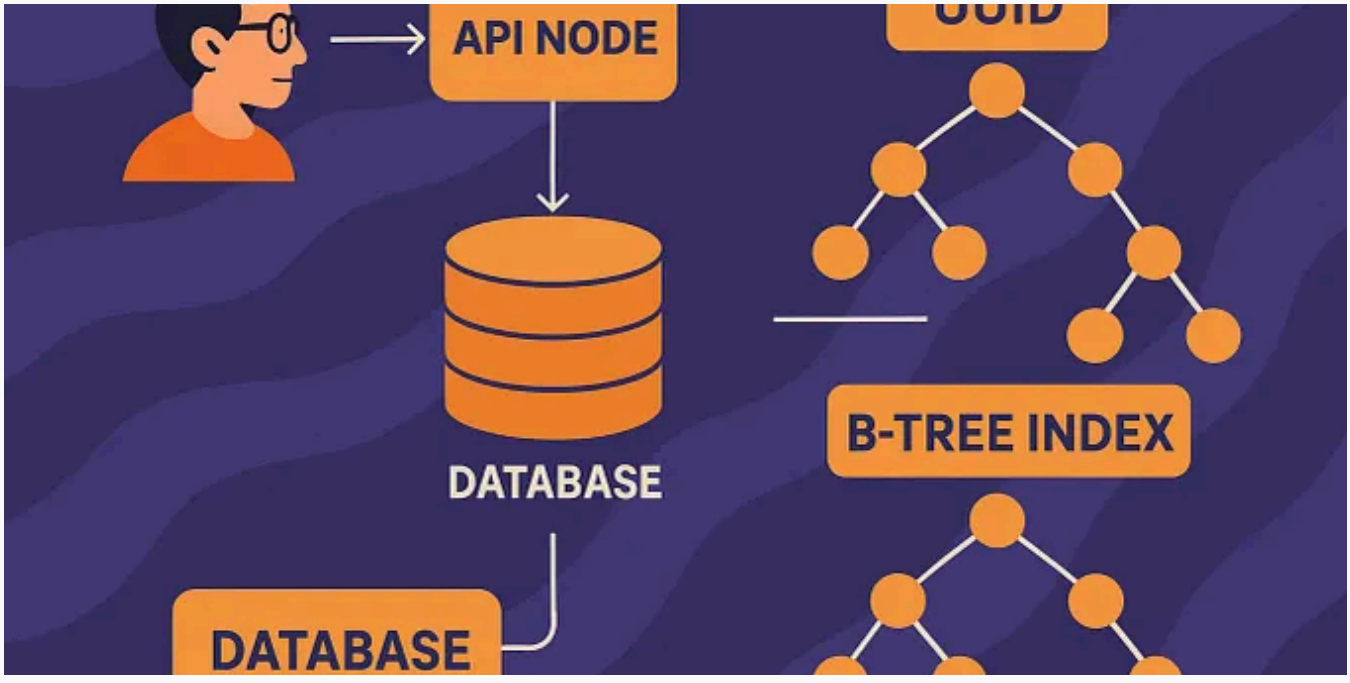


Will Lockett

The AI Bubble Is About To Burst, But The Next Bubble Is Already Growing

Techbros are preparing their latest bandwagon.

Sep 14 19K 706



The Concurrent Mind

Stop Using UUIDs as Primary Keys. It Is Killing Your Database Throughput

Your database is not slow because it is old. It is slow because every new row lands in a random place that your CPU cannot predict.

Nov 24 563 28

See more recommendations