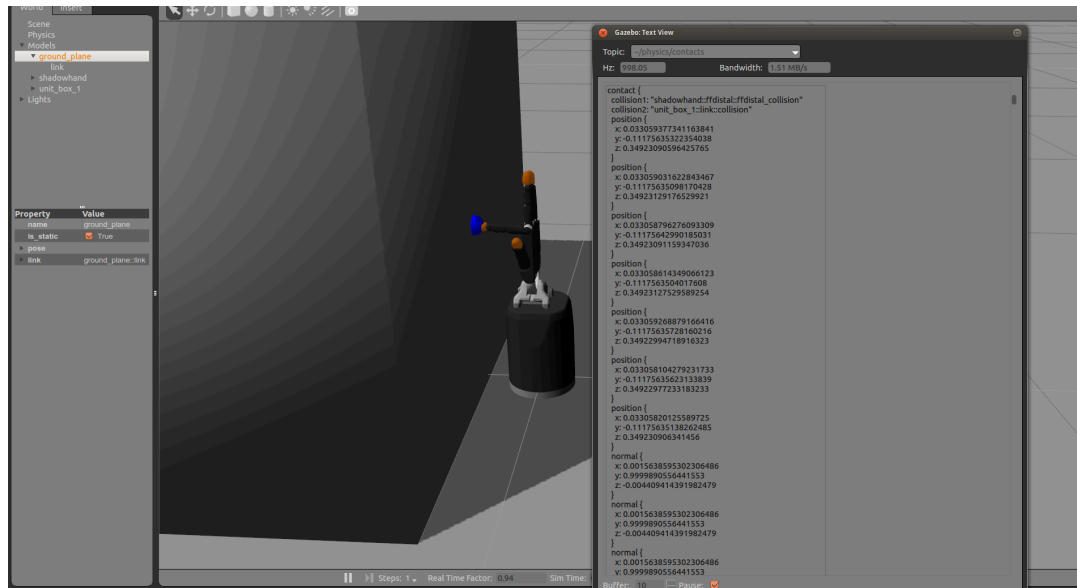


Samuel Carlos

Bumper Sensor en Gazebo Classic

Por qué es una tarea imposible



Introducción

Gazebo es un entorno difícil de manejar y sin dudas tiene una curva de aprendizaje altísima. En este entorno hostil existe una comunidad de desarrolladores y usuarios que juntos han colaborado para hacer de gazebo una plataforma OpenSource dedicada a pruebas de robótica. Sin embargo, no todo es perfecto. En este informe se explicará el caso particular del Bumper Sensor.

Bumper Sensor

El bumper sensor es el collider por defecto que trae Gazebo para detectar choques. Su implementación, de hecho, viene dentro de la librería *libgazebo_ros*, la cual se encarga de conectar ROS con Gazebo y trae plugins como el del IMU, control de robot diferencial, diversos PIDs, etc. Es esta librería la que debe instalarse al usar ROS y Gazebo juntos.

Ya que viene en la librería por defecto, no sería descabellado asumir que funciona perfectamente una vez se implementa dentro del URDF. Pero lastimosamente, la realidad es otra.

El problema

Si bien el bumper sensor puede implementarse y está listo para detectar colisiones dentro del mundo de gazebo (lo cual puede confirmarse corriendo en la terminal códigos como

```
gz topic -e /gazebo/default/<nombre de tu paquete>/<nombre de tu link asociado al bumper sensor>/my_contact/contacts
```

Es a la hora de integrarlo con ROS donde se producen problemas. No sólo no parece que no detecte ninguna colisión, sino que la herramienta urdf to sdf da problemas una vez más al tratar de convertir al urdf (lenguaje de programación que se usa para la descripción del robot) a sdf (lenguaje de programación interno de gazebo).

Comunidad

La comunidad ofrece una explicación al segundo problema, y ofrece una manera de “engañar” al intérprete de urdf a sdf al jugar con los nombres de las variables que se usan.

You need apply the name attribute to your collision tag such as:

```
<collision name="bump_sensor_collision">
...
</collision>
```

This should make it work. However if specifying in urdf or xacro you may still have an issue in that the sensor doesn't output anything. In that case suffix the

```
<contact>
  <collision>bump_sensor_collision</collision>
</contact>
```

with a `_collision`, such that it reads:

```
<contact>
  <collision>bump_sensor_collision_collision</collision>
</contact>
```

This trick was due to how the sdf is generated from the urdf file, in that it was appending a `_collision` to the name attribute of the collision specification.

La solución consiste en usar la herramienta de conversión de gazebo para ver qué nombre le pone el intérprete dentro de Gazebo automáticamente a las colisiones, y ponerlo en tags específicos dentro del mismo plugin implementado. El intérprete se usa con (en la misma carpeta donde esté el urdf):

gz sdf -p <tu .urdf> > <tu archivo .sdf para debugging>

Nos daremos cuenta que hemos solucionado este error cuando los nombres en el archivo sdf generado por el intérprete sean los mismos. Lo cual nos da líneas de código muy exóticas como:

```
<selfCollide>true</selfCollide>
<sensor name= "TibiaRU_bumper" type="contact">
  <alwaysOn>true</alwaysOn>
  <updateRate>15.0</updateRate>
  <plugin name="gazebo_ros_bumper_controller" filename="libgazebo_ros_bumper.so">
    <bumperTopicName>bumper_TibiaRU</bumperTopicName>
    <contact>
      <collision>TibiaRU_link_fixed_joint_lump_TibiaRU_Bump_link_collision_1</collision>
    </contact>
    <frameName>world</frameName>
  </plugin>
</sensor>
```

Sin embargo, la comunidad no da una solución al problema número 1 y principal: La detección de colisiones a la hora de integrarlo con ROS.

Comments

Hi jatirm. I am having these same issues. Did you ever get this figured out? Thanks.
Asked by [mequi](#) on 2020-07-27 12:07:53 UTC

Same issue here. Googled everything I can...
Asked by [Sission](#) on 2021-04-29 10:48:10 UTC

It looks like this the generated sdf name of the collision:
robot_footprint_fixed_joint_lump_bumper_link_collision_3
but this is what is being set in the <sensor> contact:
robot_footprint_fixed_joint_lump_bumper_collision_collision_3
Asked by [lucasw](#) on 2021-09-28 07:01:44 UTC

Answers

Did anybody make any progress on that? Despite using the correct collision-tag read from the sdf-file, the state argument of the published messages remains empty with obvious collisions.
Asked by [lukasschueler](#) on 2021-11-29 10:44:07 UTC

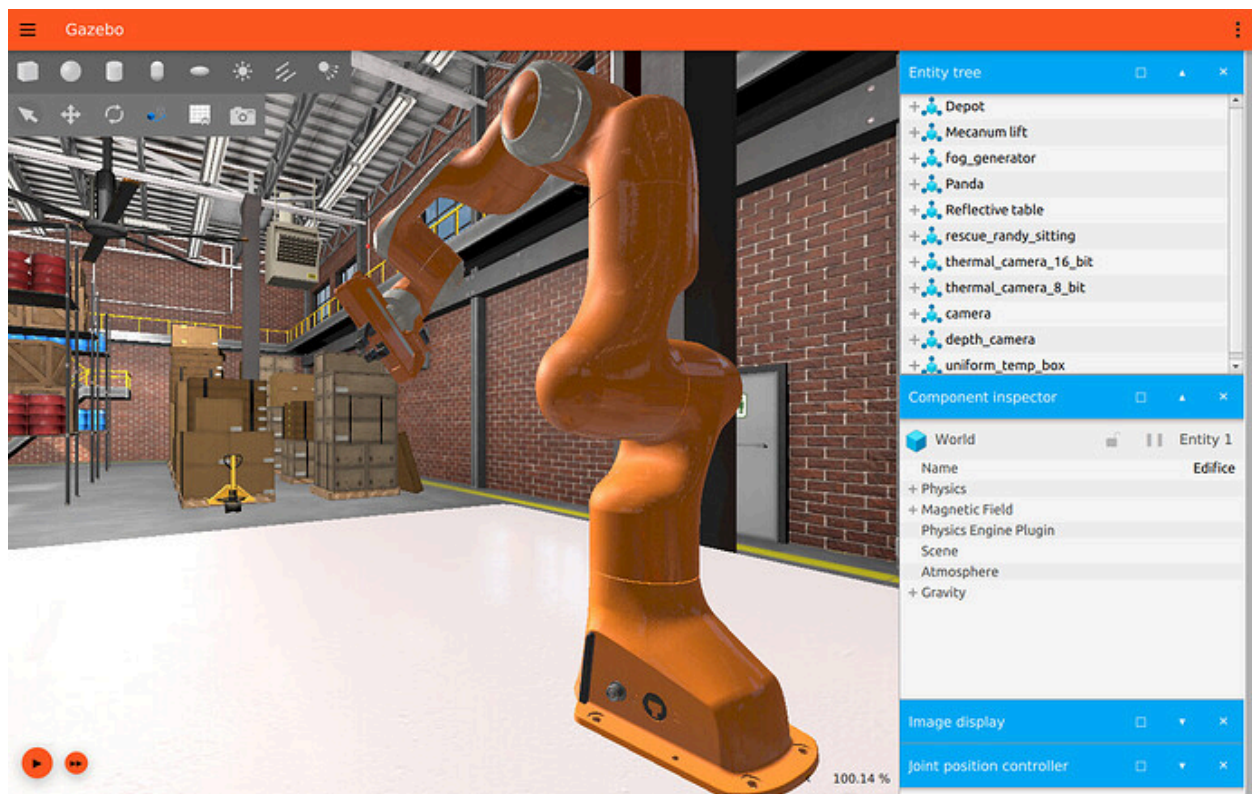
Comments

Al parecer todos tenemos el mismo problema: Cuando logras que el sdf tenga los mismos nombres en las etiquetas ya puedes aparecer el bumper sensor en gazebo, incluso crear tópicos y darle una geometría. Pero al suscribirse a estos tópicos NUNCA verás un mensaje diferente a un stamp de tiempo con el frame del mundo.

El POST más antiguo que encontré en este problema es del 2016. El más reciente fue de 2021. (Puede parecer que entonces este problema se soluciona entonces en 2022, ya que dejaron de aparecer post al respecto, y ciertamente se resolvió de forma parcial.)

SOLUCIÓN EN 2022

En el año 2022 se lanza la nueva versión de Gazebo para ese momento: Gazebo SIM, la cual mejora con respecto a su predecesor la calidad y facilidad en la integración de plugins, mejoras en la estética y funcionalidad general.



No es coincidencia que los post relacionados al problema del bumper sensor desaparezcan en esta fecha, pues con el lanzamiento de Gazebo Ignition (rama de gazebo sim) se implementa una clase DEDICADA al manejo de distintas colisiones. Dando no sólo una solución al problema sino también haciendo que sea mucho más fácil y amigable con los usuarios.

El nombre de la clase es ContactSensor, y se pueden encontrar cantidad de tutoriales en Youtube sobre su manejo.

Reflexión Final

En conclusión, podemos decir que la comunidad de ROS junto con sus desarrolladores realmente nunca encontró una solución a este problema desde Gazebo Classic, sino que dentro de su nuevo proyecto potenciaron este sensor junto con muchos otros. Por esto podemos decir que este problema de hace casi 10 años no es un problema de curva de aprendizaje alta, sino un error de los desarrolladores en su librería pensada para usar con ROS.

¿Es una tarea imposible? Sería arrogante decir que sólo porque como grupo no lo pudimos lograr es porque no se puede hacer, pero viendo las evidencias sí se puede concluir que el usuario promedio de ROS no puede hacer esta tarea. Para cumplirla, se tendría que como mínimo tener conocimientos al nivel de los desarrolladores de Gazebo y de ROS, para así crear una librería o un plugin desde cero que pueda suplir estas necesidades. Así mismo, aunque no encontré ningún plugin funcional, puede que en algún foro chino se hallará la solución en un foro del 2016 al cual no pude llegar. Diré entonces que es **imposible con nuestros conocimientos actuales** y debido a que hay una manera mucho más fácil de implementar, también sería una pérdida de tiempo seguir tratando de implementar el bumper sensor dentro del entorno de gazebo classic.