StickyFingers Plugin

What Is StickyFingers?

StickyFingers is a ROS plugin meant to assist in grasping objects or representing a magnetic, suction, or adhesive gripping element. It makes one link in a Gazebo model “sticky”- when enabled by a ROS message, if that link comes into contact with an object that object will become “stuck” to the link and follow its position until the disable message is sent.

Employing StickyFingers

To add a StickyFingers plugin to a link in your robot, add the following material to the link definition:

```
<sensor name='[NAME 1]' type='contact'>
     <plugin name="[NAME 2]" filename="libsticky_fingers.so">
          <capacity>[C]</capacity>
     </plugin>
     <contact>
          <collision>[COLLISION NAME]</collision>
     </contact>
</sensor>
```

[NAME 1] and [NAME 2] can be anything. [C] is the maximum mass the gripper should be able to lift- anything above this value, and any static objects, will not attach to the gripper. [COLLISION NAME] is the name of the collision object within the link that you want to make sticky.

IMPORTANT: The collision name *must* be of the form [link name]_collision, or the sensor plugin will return an error- for instance, if my link is named 'ftip', then the collision inside of it (and the contents inside the <collision> tags) must be named ftip_collision.

An example of this structure can be found in world/blocks_on_table.world.

Note that in a URDF file, like any plugin definition you will need to include the stickyfingers sensor outside of the link definition, inside of <gazebo> tags.

Controlling StickyFingers

On startup, every StickyFingers link offers an action server that communicates with StickyControl messages:

```
bool sticky_status
---
bool new_status
```

The messages are on topics named “sticky_finger/[NAME 1]”- for instance, if we had a finger named ftip_sticky, we would call the action server “sticky_finger/ftip_sticky”. The console will display the messages used by each sticky finger in the simulation whenever Gazebo starts up.

A True sticky_status will enable the finger, a False sticky_status will disable a finger and cause it to drop whatever it is holding.
A stand-alone executable to produce these messages is included as “finger_control_dummy_node”/