EXPERIMENT  
Test Gazebo/ROS Force Torque Sensor attached to Joint 6 of Fanuc LRMate 200id

# Background

Explanation of the force/torque sensor use in Gazebo is explained here:

<https://github.com/osrf/gazebo_tutorials/blob/master/force_torque_sensor/tutorial.md>

# Setup:

First, add Gazebo ROS force/torque sensor plugin to joint6 fanuc robot

We provide a URDF directly to the Gazebo roslaunch so in fanuc\_lrmate200id.urdf add the following XML code to enable the force/torque sensor plugin. Note

<gazebo reference="fanuc\_joint\_6">

<provideFeedback>true</provideFeedback>

</gazebo>

<gazebo>

<plugin name="ft\_sensor" filename="gazebo\_ros\_ft\_sensor.dll">

<updateRate>50.0</updateRate>

<topicName>ft\_sensor\_topic</topicName>

<jointName>fanuc\_joint\_6</jointName>

</plugin>

</gazebo>

One issue that was observed using the F/T Sensor plugin was never reading any values but 0,0,0 for f/t. This was found using rostopic echo to dump the FT values, even when the robot held peg was colliding with the pegarray on the table:

c:\opt\ros\noetic\gztaskboard>rostopic echo ft\_sensor\_topic

header:

seq: 0

stamp:

secs: 43

nsecs: 165000000

frame\_id: "fanuc\_link\_6"

wrench:

force:

x: 0.0

y: 0.0

z: 0.0

torque:

x: 0.0

y: 0.0

z: 0.0

Scanning the Internet provided the insight to add a Gazebo reference in the URDF to provide feedback as shown in the following::

<gazebo reference="fanuc\_joint\_6">

<provideFeedback>true</provideFeedback>

</gazebo>

Still nothing but zero F/T values were reported, so the joint 6 dependency (link 6 with mass and inertia contributing to the forces) was modified to allow physics to operate. So originally the robot joint 6 was assigned as kinematic (as were all other links so the robot won't immediately start drooping when no values are supplied to the joints).

Original Fanuc Link 6 URDF code with kinematic enabled and gravity disabled:

<gazebo reference="fanuc\_link\_6">

<static>0</static>

<gravity>0</gravity>

<kinematic>1</kinematic>

<material>Gazebo/Yellow</material>

</gazebo>

New Fanuc Link 6 URDF code with kinematic disabled and gravity enabled:

<gazebo reference="fanuc\_link\_6">

<static>0</static>

<gravity>1</gravity>

<kinematic>0</kinematic>

<material>Gazebo/Yellow</material>

</gazebo>

After this modification was tested, non-zero FT values were reported.

# SOFTWARE

Inside the peg in hole assembly code, test code was developed to slowly approach the peg board array top until the force indicates that the robot grasped peg is touching (colliding with) the peg board array.

Originally, it is uncertain what force value indicates contact so guessing was required. In addition, movement is slow as the communication mechanism to the gazebo server is slow.

The test code was a bit clunky as it approached the peg hole array board (with an offset I x so it would hit the board and not a hole). Previously dead reckoning insertion of the round pet into a round hole was coded and worked. Upon approach to the skewed offset hole position in X, a slow decremented approach in the -Z direction was undertaken until the FT force in the Z direction was significant (0.05 is the current guess.

double zdist = 0.002;

while (ros::ok() && robot.\_arm->ft.force.z < 0.05)

{

// Approach contact with table.

// decrement approach to surface

rHoletouch = robot.\_arm->retract(rHoletouch, tf::Vector3(0.0, 0.0, -0.01));

std::cout << "Holetouch Robot Coord Near = " << conversion::dumpPoseSimple(rHoletouch) << "\n";

robot.MoveTo(Convert<tf::Pose, robotPose>(rHoletouch), true);

ros::Rate(10).sleep();

ros::spinOnce();

}

// We are now in contact with the table

// move to hole centroid where force in Z should be zeroish again.

tf::Pose rStartPose = rHoletouch;

tf::Pose wHole1(qBend, hole.location);

tf::Pose rHoleTop = robot.\_arm->toRobotCoord(wHole1);

rHoleTop.getOrigin().setZ(rStartPose.getOrigin().z());

double n=0.0;

while (ros::ok() && fabs(robot.\_arm->ft.force.z) > 0.001)

{

std::cout << "Force =" << robot.\_arm->ft.force.x << ","

<< robot.\_arm->ft.force.y << ","

<< robot.\_arm->ft.force.z << std::endl;

tf::Vector3 diff;

n=n+0.1;

diff=lerp(rStartPose.getOrigin(),rHoleTop.getOrigin(), n);

std::cout << "Lerp End=" << conversion::dumpVectorSimple(rHoleTop.getOrigin())

<< " start=" << conversion::dumpVectorSimple(rStartPose.getOrigin())

<< " diff=" << conversion::dumpVectorSimple(diff) << "\n";

tf::Pose rNextPose (qBend, diff);

std::cout << "Guarded Move Robot Coord = " << conversion::dumpPoseSimple(rNextPose) << "\n";

robot.MoveTo(Convert<tf::Pose, robotPose>(rNextPose), true);

ros::Rate(10).sleep();

ros::spinOnce();

}

// Now we should move in -z until we detect a discernable positive z force.

std::cout << "Force Z=" << robot.\_arm->ft.force.z<< std::endl;

# References

FT Sensor shows zero values  
<https://answers.gazebosim.org/question/15465/ft-sensor-shows-zero-values/>

tutorial describes how to use a force/torque sensor on a joint. This sensor publishes force and torque readings to a topic.  
<https://github.com/osrf/gazebo_tutorials/blob/master/force_torque_sensor/tutorial.md>