Crcl 2 Ros Workspace

# Contents

This repository contains C++ code that provides a CRCL XML streaming and parsing component, that maps command and status motion primitives from CRCL to ROS, then uses ROS moveit to plan motion trajectories that are then simulated in Gazebo.

# Status

## Requirements

* Ubuntu 16.04 Trusty
* Gazebo 7 (to allow gazebo\_ros\_api packages)
* ROS 1 Kinetic
* CodeSynthesis and Xerces XML tools
* Gnu C++ compiler
* moveit

# Installation

Clone the crcl2ros repository from the github repository. You may need to install the following packages if you have not done so already:

* ros-indigo-desktop-full
* gazebo7
* libgazebo7-dev
* gazebo-ros-pkgs
* moveit-full

You will need Xerces C library installed, as the catkin uses it to link against for CodeSynthesis. The CodeSynthesis xsd compilation of the CRCL xsd has already been done and the cpp/h files are included in the repository.

If a missing ROS dependency causes the build to fail, run rosdep.

## Building

Change to the crcl2ros main subfolder to build.

$ cd $HOME

# retrieve the latest development version.

$ git clone -b master https://github.com/usnistgov/crcl2ros.git

# change folder

$ cd crcl2ros

# build the workspace (using catkin\_tools)

$ catkin build -DCMAKE\_BUILD\_TYPE=Debug

It is suggested to use a Debug compilation so you can then attach to the running crclapp in Qt IDE to debug. In the wiki is a description on how to debug the crclapp using QT IDE.

## Running

$ source devel/setup.bash

$ roslaunch fanuc\_lrmate200id\_support top.launch

# Disclaimer

This software was produced by the National Institute of Standards and Technology (NIST), an agency of the U.S. government, and by statute is not subject to copyright in the United States. Recipients of this software assume all responsibility associated with its operation, modification, maintenance, and subsequent redistribution.

See NIST Administration Manual 4.09.07 b and Appendix I.

# Background

The repository contains a workspace for ROS to handle CRCL streaming, CRCL to ROS conversion, and ROS to Gazebo simulation.

Here are some notes regarding the implementation:

* Only the Fanuc LRMate 200id has been thoroughly tested.
* The CRCL has been extended to provide Gazebo or Sensor feedback regarding model information. In addition, inferences about the kitting model is performed and included in the extended CRCL status model report as properties.
* There is no Rviz.
* roslaunch files were merged and stripped of unused components so as to understand why multiple robot namespaces caused issues.
* The gazebo\_ros\_api plugin is loaded upon launch of gazebo
* A lot of effort went into allowing multiple robots with separate URDF. ROS makes it rather straightforward to bundle multiple robots into one URDF (like a 2 armed robot) and then controlled in world coordinate space. In the end, existing SDF was used as the Gazebo world and identical URDF with a qualified namespace was used as ROS param.

|  |  |
| --- | --- |
| ROS node | Description |
| crcl | metapackage |
| crclapp | Crcl 2 ROS workspace node |
| fanuc\_lrmate200id\_support | configuration, URDF, launch and other files for Fanuc LRMate 200id robot |
| gzdatabase | Gazebo database of SDF models, meshes, etc. |
| gzparallelgripperplugin | Gazebo C++ plugin for parallel gripper grasping. |
| motoman\_sia20d\_support | configuration, URDF, launch and other files for Motoman Sia 20d robot |

## Configuring

The configuration for the crcl2ros workspace is defined