RVinci: RViz Interaction and Navigation with da Vinci

Logan M. Ellis[1] and Dr. Simon Leonard[2]

The Robotic Refueling Mission underway at NASA is seeing the development of robotic applications for on-orbit satellite servicing and refueling. The robots will be required to dock with the satellite, cut and peel back the insulation layer, and pump fuel into the fuel port.

Signal delays between the robot and ground station can reach 3-5 seconds, making teleoperation prohibitively difficult but constraining the motion of both master and slave robots increases efficiency and accuracy.

A virtual workspace for the da Vinci Surgical Robot Console, linked to the physical robot environment, provides a versatile platform to aid in the alignment of virtual holonomic constraints. One use is to align a virtual plane with a planar surface in a video stream to provide positional awareness and allow for motion constraint.

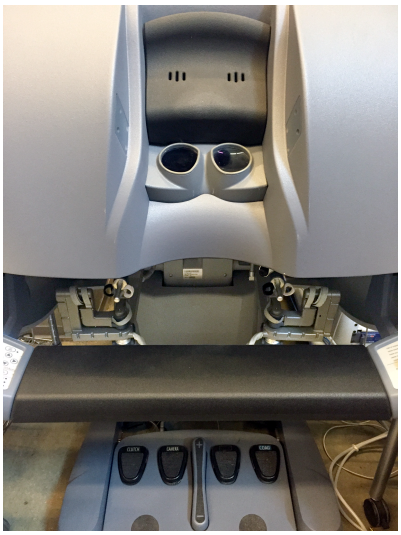

Figure 1 – da Vinci Console

A plugin was created for RViz, a visualization tool for ROS, utilizing C++, Qt, and OGRE libraries. This plugin created a separate RViz window with two viewports for the da Vinci's stereoscopic display. Two 3D are produced and controlled by the da Vinci manipulators. This allows the user to manipulate interactive objects within the world. The user can navigate the world by pressing the camera pedal and moving the manipulators, similar to camera control with a physical da Vinci Surgical System camera setup. Figure 2 is what the user sees when peering into the da Vinci display.

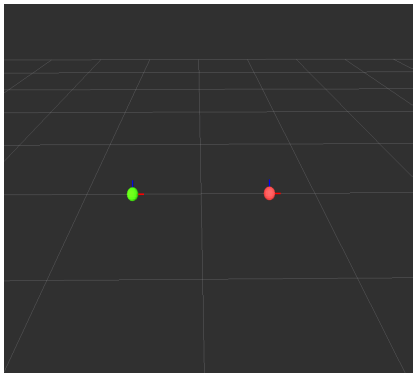

Figure 2 – RVinci Display

The RVinci plugin provides a means of navigating and interacting with objects within RViz utilizing the da Vinci console. The control and navigation of the virtual workspace is similar to the experience of using a physical da Vinci Surgical System. This virtual workspace can be linked with a physical robotic environment to produce a versatile and intuitive control platform for a variety of applications.

---

1) Department of Mechanical Engineering, University of Missouri – Kansas City
2) Department of Computer Science, Johns Hopkins University;

The funding for this project was provided by the National Science Foundation's Research Experience for Undergraduates (REU) program under grant number EEC-1460674.