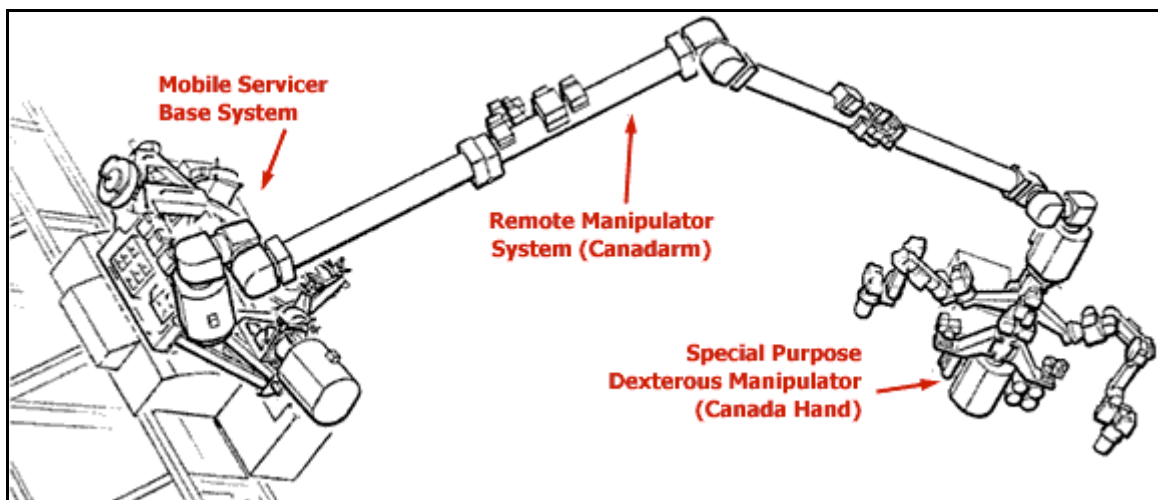


Project Proposal

Virtual Control of the
Space Station Remote Manipulator System
(SSRMS)

Jeffrey Sugar
August 27, 1998



Introduction

Current research involving the International Space Station (ISS), robotics, and virtual reality has led a group at NASA Langley Research Center (LaRC) to develop a collaborative engineering environment. This environment will someday allow users at LaRC to interact with other users at remote sites to demonstrate the use of the SSRMS (shown on the cover as the Canadarm).

The goal of this project is to take the three dimensional model of the SSRMS into a virtual environment to allow the user to move the robotic arm in a prescribed path.

Problem

The research done at LaRC to this point has taken the three dimensional model of the SSRMS into an environment where the robot can move about all joints. This robot animator program allows the user to toggle between all joints on the robot arm while the angle of rotation loops from 0 to 360 degrees. Recent work, that is not yet complete, involves the kinematics of the robot. Two points, A and B, are identified by giving the x, y, and z position coordinates as well as the i, j, and k orientation coordinates for the robot end effector to start and end at. These points are then entered into a kinematics solver and the appropriate angles per time period are then entered into the robot animator program. The animator program then moves the robot through the sequence of angles to move the end effector from point A to point B.

The next step is to immerse the user in a virtual environment where he/she can “grasp” the end effector and move the arm from any point in space to any other point. The animator program does not allow for this as it only asks for the starting and end point of the end effector. The virtual control of the SSRMS proposed here will give the user the ability to determine the path of the end effector.

In addition to moving the end effector, the robot arm will be able to grasp objects and move them in the virtual environment. To move an object, the proximity of the end effector to the object will have to be determined. If the end effector is within proximity of the object, then it will be assumed that the end effector has captured the object. Once the object is captured, it can be moved. Moving objects in a virtual environment will allow the user to see if the motion is possible and determine if any problems will occur.

A final problem to examine is the use of different computer environments where this system can be used. A user of the system should be able to maneuver the SSRMS if they are viewing it on a computer screen, a single wall projection system, or in the CAVE¹ environment. Combining all of these tasks in the collaborative engineering environment allows several users, locally and remotely, to view the same motion.

Objectives

The solution to this problem involves several items. The first is to get the model of the SSRMS into the virtual environment. Once the model is in this environment, different methods can be used to determine how the user will move the arm. Moving the

arm involves determining the angles of the joints, which will be provided by the inverse kinematics software from Analytical Mechanics Association (AMA), Inc. The closed form equations will be programmed into the environment to provide the inverse kinematics.

The arm will be able to move when it is in an empty (moving only itself) state and loaded (moving an object) state. When the arm is moving an object, a method will have to be developed to determine when the object is grasped.

The software written should be platform independent allowing the user to work in front of a computer screen, a single wall projection system, or in the CAVE environment.

Product

The model of the SSRMS is in the form of several object files. Each object file contains one link of the robot arm and one axis of rotation. There are eight files in all each of which was created using Wavefront and must be read and stripped of its proper information. Each object file contains information pertaining to the number of vertices, lines, and faces. Currently, the robot animator program reads each Wavefront object file, strips away the proper information, and brings this information into OpenGL using arrays. From these arrays, the model can be generated.

Once the model is available, the coding for the virtual environment begins. The coding for the virtual reality aspect of this project can take place using either the OpenGL graphics library or the IRIS Performer application development environment. There are advantages and disadvantages associated with each method of programming. A determination will be made as to which programming language will be used.

Once the model is in the virtual environment, different approaches can be used to have the user interact with the robot arm. An example of how the user can interact is through the wand. The wand is similar to a computer mouse except the wand works in three dimensions allowing for interactive input. The wand has three buttons and a pressure-sensitive joystick. The user can move in the virtual environment by pointing in the direction of movement and clicking a button. The robot arm could be moved when the user highlights a link of the arm, clicks on it, and then drags it to a new location. Once the approach for interaction has been determined, the end result will allow the user to move the robot arm.

To move an object, a determination will be made as to how close the end effector is to the object in question. Once the proximity limit has been reached, the object will be attached to the end effector and can be moved. A method to release the object will have to be made so the user can move another object. Also, depending on the software provided by AMA, other links of the robot could be moved if the proper closed form equations are included. The primary link the user will have access to will be the end effector.

The choice of the coding language could have an effect on the ability of the user to work in different environments. Performer is somewhat more lenient in this area than OpenGL. In either case, the goal is to have the software understand which environment

the user is in (computer screen, single wall projection system, or CAVE) and respond accordingly.

Resources

An understanding of C++ and OpenGL are necessary for the coding of this project. I have an understanding of C++ and am in the process of learning OpenGL. If Performer is to be used, that will have to be learned.

AMA, Inc. will provide either the closed form equations for the inverse kinematics solution or the software used to solve the equations. Without these equations, the movement of the SSRMS will not mimic the real motion of the robot arm.

Virginia Tech can provide access to the CAVE, several Unix workstations, as well as technical support from John Kelso and Ron Kriz.

Additional help can be obtained from Michelle Garn, Pat Troutman, and Scott Angster at NASA LaRC as well as other NASA centers. It is my understanding that NASA currently uses virtual reality as well as telerobotics for training missions involving the Space Shuttle's Remote Manipulator System. In addition, several universities and companies are using virtual reality and telerobotics to solve their own problems as well as doing collaborative work with NASA.

To gain help with this project, several on-site trips will be necessary at NASA LaRC located in Hampton, Virginia. More trips could be planned to visit other sites where similar research is being conducted. These sites include the University of Maryland Space Systems Laboratory (SSL), NASA Johnson Space Center, and the Jet Propulsion Laboratory.

Schedule

The end product will be delivered to NASA LaRC on or before December 31, 1998. The end product will be a software deliverable containing the code for the SSRMS in a virtual environment. The software should be able to run in the LaRC collaborative engineering environment.

Costs

There are no expected software or hardware costs involved. Potential research travel costs include: gas, airfare, housing, and meals. If more trips to NASA LaRC are required for extended periods of time, the costs of housing, gas, and meals could limit the ability of the student to complete the project.

¹ Description of the CAVE provided by the CAVE User's Guide located at <http://www.ev1.uic.edu/pape/CAVE/prog/CAVEGuide.html#description>.

The CAVE (CAVE Automatic Virtual Environment) is a projection-based VR system that surrounds the viewer with 4 screens. The screens are arranged in a cube made up of three rear-projection screens for walls

and a down-projection screen for the floor; that is, a projector overhead points to a mirror, which reflects the images onto the floor. A viewer wears stereo shutter glasses and a six-degrees-of-freedom head-tracking device. As the viewer moves inside the CAVE, the correct stereoscopic perspective projections are calculated for each wall. A second sensor and buttons in a wand held by the viewer provide interaction with the virtual environment.