VIRTUAL COOPERATIVE ROBOTS

Duane Driggs[1], Jeron Williams[2], Sakawa Baylor[2], Sanjay Kodiyalam[2], Venkata Rama Reddy Sabbella[3], Kazim Sekeroglu[1], Amitava Jana[2]
Department of Electrical Engineering[1] / Mechanical Engineering[2] / Computer Science[3], Southern University, Baton Rouge, LA 70813
duanedriggs@engr.subr.edu, jeronwilliams504@yahoo.com, sakawabaylor@engr.subr.edu, sanjaykodiyalam@engr.subr.edu, venkatram1235@gmail.com, ksekeroglu@gmail.com, jana@engr.subr.edu,

Abstract
Motivated by the desire for virtual task planning for cooperating robot systems a generic CAVE-library based Visual Studio C++ program is developed to stereoscopically visualize multiple robots in a Cave Automatic Virtual Environment (CAVE) at Southern's College of Engineering. Robotic links are first conceptualized in AUTOCAD. Currently two robots, with three 2-axes links each, are represented while they jointly hold an object. OpenGL transformations are used to display and rotate the robot's links. A master-slave configuration of the robots is targeted: The procedures for direct movement of the master's links based on user input and automatic movement of the slave's links have been formulated. Future work will implement these procedures in the C++ program and tailor this design to drive the movement of real robots in the Mechanical Engineering Department's Mechatronics Laboratory.

Keywords: Task planning, Cooperative Robots, Affine transformations, CAVE.

1. Introduction

Robotic systems are of particular interest in areas not conducive for human presence – as in bomb disposal or in handling radioactive materials. These tasks may require more than one robot to increase the load carrying capacity [1]. Task planning for such cooperating robot systems has been a subject of research involving kinematics, dynamics, and control. Of these, kinematics is the most fundamental with the two cooperating robots forming a closed kinematic loop [1,2]. The current work stereoscopically visualizes two robots in a CAVE and targets virtual task planning with the end goal of recording the movements of the robots for subsequent use in driving real robots in the Mechatronics laboratory. A generic program to stereoscopically visualize multiple robots with multi-axis links is being developed: Currently the display routine specializes to the case of two robots – to be operated in a master-slave configuration. While the routines to move robots are not yet implemented, the movement procedures have been formulated.

2. Conceptualizing robotic links

The geometry of the robotic links is first conceptualized in a solid model of the base and first link built in AUTOCAD (Fig. 1). For simplicity, this model is built using only cuboids, cylinders and hemispheres.

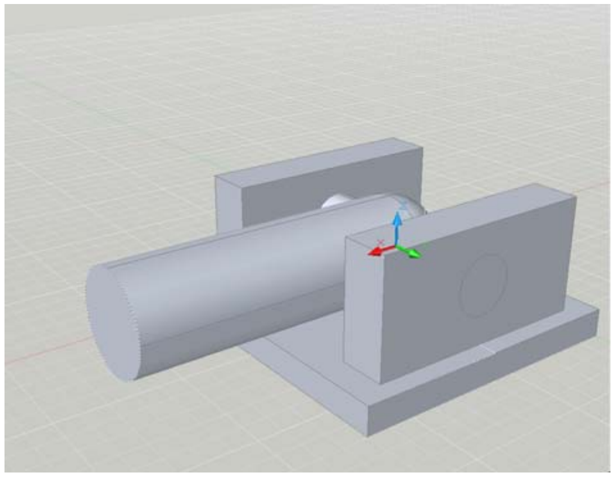

Fig. 1. Solid model of the robot's base and first link. The base and posts are cuboids while the link and the shaft holding it are cylindrical. The link is capped by a hemisphere at one end. Subsequent links are envisioned to be similar with grippers at the other end of the previous link serving as the posts for the shaft holding the next link.

3. Stereoscopic display in the CAVE

While the data structure used in the program allows for multiple robots with any number of multi-axes links, the display routine specializes to the case of two robots jointly holding a bar (Fig. 2). In order to be able to translate and rotate in any desired way the bar must have the all of its 6 degrees of freedom. This is accomplished by each of the robots having three 2-axes links that are constructed and put together as conceptualized in the AUTOCAD model (Fig. 1): using cylinders and cuboids – with further simplification by eliminating the hemispheres capping the links. These links can rotate around an axis along their length and also a perpendicular axis defined by the shaft holding them. The display routine uses OpenGL primitives of triangles and quadrilaterals to construct the surfaces of the cylinders and cuboids constituting the links. OpenGL operations of translation and rotation are used to position and orient the links appropriately. The post-multiplication feature of OpenGL transformation accumulation, that

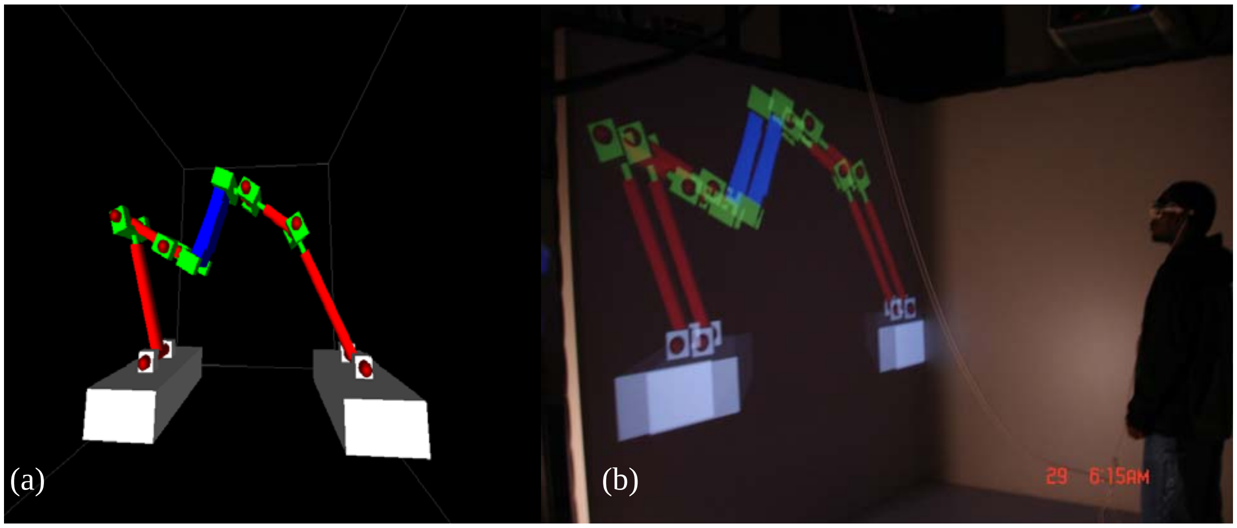

Fig. 2. (a) Non-stereo display of the two robots in "simulator mode" on a desktop. (b) Left and right eye images created automatically by the CAVE library [3] enabling the user to see a stereoscopic image of the robots in the CAVE.

corresponds to transforming coordinate systems, simplifies this task – as parts of links are created in appropriately shifted and rotated coordinate systems. Figure 2a shows the display of the two robots from the program running in "simulator mode" producing a single non-stereoscopic image on a desktop. Each of the robotic links can rotate around an axis along their length and also along the perpendicular shaft-axis. In the configuration shown the two robots are seen holding the different bars – shown in blue – with nearly the same orientations but significantly different positions. If cooperation were perfect there would be no difference in the orientations and positions of the two bars - they would overlap thereby defining the single bar jointly held by both the robots. The closed loop formed by the both of the robot's links together with this bar would then imply that the net OpenGL transformation is zero.

Figure 2b shows a user stereoscopically visualizing the robots in the CAVE using shutter glasses. As the user moves, the CAVE library [3] automatically computes the correct stereoscopic perspective projections for each display using the information from the six-degrees-of-freedom head-tracking device attached to these glasses. This library creates a new thread for each of the displays in the CAVE that operate synchronously while the main thread continues to operate asynchronously.

4. Procedures for moving the robots

While the procedures for moving the two robots are not yet implemented in the C++ program they have been formulated with the two robots in a master-slave configuration.

The robot that is to serve as the master will move by user input via a wand - a second 6-degrees-of-freedom sensor with buttons that provides interaction with the virtual environment [3]. The display is augmented with a straight line emanating from the wand. Any of the master's links may be selected on intersection with this line by holding down a button on the wand. The selected link is highlighted in the display with a change in color. Subsequent changes in the orientation of the joystick drive the changes in the orientation of the link – by correspondingly changing all of its degrees of freedom. The routine for moving the master follows the navigation routine that translates, rotates, and scales all of the objects in the display and is within the display loop.

The robot that is to serve as the slave will move automatically in response to the master's movement such that the bar that is held by both is not strained. The automatic movement of the slave is driven by a routine that iteratively minimizes a dimensionless measure Δ of the bar's strain. Δ is constructed from the difference between affine (A) transformations [4] acting on the bar due to its positioning by the master (^{M}A) and slave (^{S}A) robots. With $^{M/S}A_{4x4} = \begin{bmatrix} ^{M/S}R_{3x3} & ^{M/S}t_{3x1} \\ 0_{1x3} & 1_{1x1} \end{bmatrix}$ (R is the rotation and t is the translation of the bar's center), Δ is chosen as: $\Delta = w_1 \frac{\|^{S}t - ^{M}t\|^2}{(v_{bar})^{2/3}} + w_2 \|^{S}R - ^{M}R\|^2$, where v_{bar} is the volume of the bar that w_1 and w_2 are weight factors (w1+w2=1) chosen to be equal as an initial attempt. For perfect cooperation Δ=0. Using OpenGL, the affine transformations are constructed by applying the operations of translation and rotation in the exact sequence as they appear in the routine for the display of the robots holding the bar.

The iterative procedure to minimize Δ involves increasing or decreasing each of the slave's link angles as needed to reduce Δ. This is done in sequence over all of the link angles and repeated until Δ is sufficiently small (10^{-3}). The magnitude of angle change attempted is specific to each link angle and is large initially (5°). Subsequently it is reduced by a factor of 2 whenever both increment and decrement of the link angle does not reduce Δ. This procedure is an explicit search for the slave's link angles that minimize Δ. It avoids the need to compute angular derivatives of Δ that may be required if other minimization algorithms such as the Newton-Raphson [5] are used. As the overall procedure to move the slave is expected to be computationally intensive the routine is run asynchronously in the main thread of the C++ program.

Ongoing and future work

Implementation of the procedures for moving the master and slave robots in the C++ program is in progress. Future work will explore optimizing the routine for moving the slave and explore if it can be executed within the display loop with sufficient speed so as to have a display frame rate of at least 30 per second. Both robots will eventually function as slaves with the user moving the jointly held object to avoiding obstacles in the virtual environment. The path followed by the robots – recorded as a time sequence of all the link angles – may then be use to drive real robots performing the same task in the Mechatronics laboratory.

Acknowledgement

This work is supported by NSF/EPSCoR EPS-0701491 (CyberTools). The Center for Energy and Environmental Studies at Sothern University and Scott Woodall are acknowledged for supporting and designing hardware upgrades to the CAVE.

References

1. A. Jana, D. M. Auslander, R. N. Dave, S. S. Chehl: Task Planning for Cooperating Robot Systems.
2. H. P. Huang, and R. S. Chen: Modelling and Adaptive Coordiantion control of a Two-robot System. Journal of Robotic systems, **9** (1), pp 65-92 (1992)
3. Dave Pape, Carolina Cruz-Neira, Marek Czernuszenko: CAVE Library version 2.6. Electronic Visualization Laboratory University of Illinois at Chicago, (1997)
4. http://en.wikipedia.org/wiki/Affine_transformation
5. http://en.wikipedia.org/wiki/Newton's_method