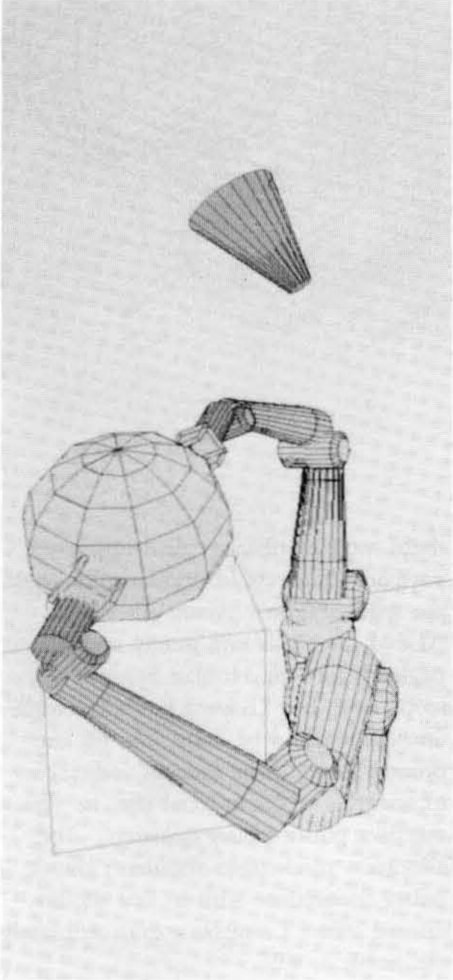


This graphic simulation shows JPL's robot arm in action. Many tasks take two arms working together. While Gutierrez was working on the problem of one moving arm, seniors Alvin Law and Ming Lee were simulating two stationary arms holding an object between them. Kenneth Kreutz, who, with Abhinda Jain, directed the two-arm work, notes that this project is harder than it appears because the arms must exert enough force on the object to keep it in their grasp, but not enough to crush it. Animation created by Mark Long.

Arms and the Robot

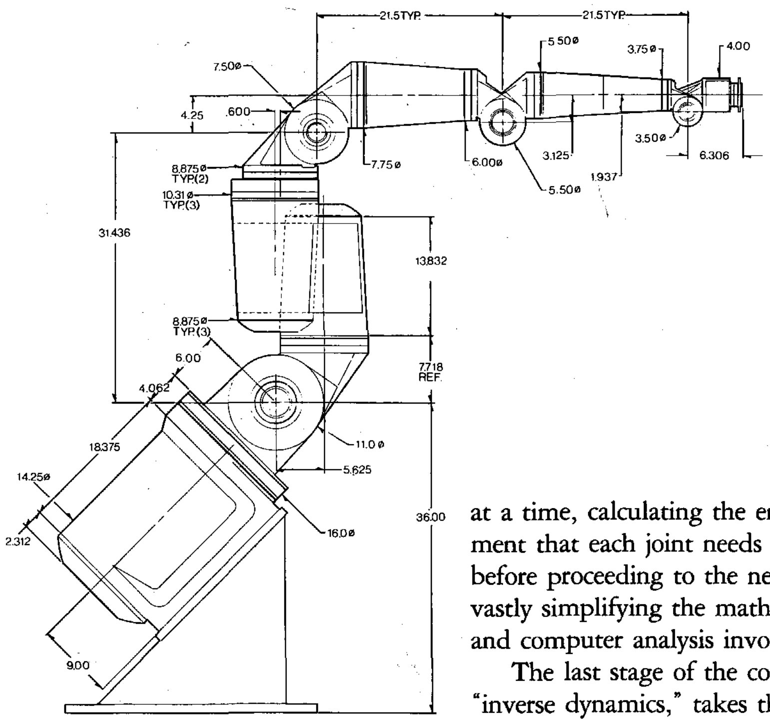
A robot arm will need flexible programming as well as flexible joints.

Imagine a job washing windows, wiping each pane clean with a smooth, circular motion. Pretty simple stuff, right? Not for a robot. This past summer Roman Gutierrez, now a junior in applied physics, used a Summer Undergraduate Research Fellowship (SURF) grant to work with Guillermo Rodriguez, a senior member of the technical staff at JPL, on a software package to control a robot arm's motion in two dimensions. Eventually, the program will be a part of the operating software for a robot arm that JPL is installing in one of their laboratories, a prototype for the ones that will be needed to service and repair satellites and assemble structures in earth orbit.

These arms will need much more sophisticated control programs than the ones used in earthbound factories. An arm welding fenders for Ford has a real no-brainer job. The arm and its assembly line are set in concrete, figuratively as well as literally. The assembly line brings each fender, in exactly the same orientation as the previous one, to within a millimeter or so of a given spot, and holds it there. Then the robot follows an explicit list of instructions to make a set of predetermined motions that touch the fender at designated spots. But there are no concrete floors in orbit, so a robot arm will need flexible programming as well as flexible joints, matching its motions to where things really are. The tasks to be performed in space are also more complex.

Telling a robot arm how to reach out and touch a desired spot isn't easy. First of all, the arm's mechanical elements must be defined for the computer—how many joints it has, each joint's range of motion, and the dimensions of the links connecting the joints—and the arm's initial position must be specified. Then the computer decides what position the arm would have to be in to reach the desired spot. Using a process called "inverse kinematics," the computer compares the arm's current position to the position it needs to assume and decides how to move it to that position. This analysis has generally been handled after the fashion of a series of still photos strung together to make a movie of the whole arm in motion, with each joint moving incrementally between frames. With everything moving at once, the mathematics becomes quite formidable.

Gutierrez used a new method, recently developed by Rodriguez, for designing robot movements based on state estimation theory. (State estimation theory allows the overall state of a complex system to be reconstructed from a limited sample of data.) The method essentially starts at the point to be touched and moves inward one joint



The robot arm. Dimensions are in inches.

at a time, calculating the entire movement that each joint needs to make before proceeding to the next joint, vastly simplifying the mathematics and computer analysis involved.

The last stage of the computation, "inverse dynamics," takes the motion information just calculated, factors in the forces at work—the load on the arm, friction, the arm's springiness, and so forth—and, working out from the shoulder, decides what force is needed at each joint to move the arm.

When the arm's operating software is complete, the inverse dynamic data will drive another set of programs controlling the electrical mechanisms that actually move the arm. For the immediate future, the data will be input to another computer simulation, which will calculate the arm's motion and display it on screen as it moves. (This second simulation package wasn't part of the summer's work, but is being created as part of JPL's Space Automation and Robotics Program.)

Most robot arms have six degrees of freedom, or ways in which they can move. The arm bends at the shoulder joint, rotates around it, and bends at the elbow. The other three degrees are in the wrist, which bends, rotates, and yaws—moving from side to side like a metronome or those mechanical waving hands sometimes seen in the rear windows of pickup trucks. JPL's arm will have one additional degree of freedom, a second elbow-like hinge in the forearm. "This will enable the arm to use the sort of bending motion a snake uses to lift the front of its body," says Rodriguez. "It will allow the arm to reach around obstacles that would otherwise block it." The arm is mechanically complete now, and is undergoing preliminary tests. It will become fully operational within a year.

The mathematical analysis needed to generate the arm's motion—even using state estimation theory—is no pushover, involving Newton-Euler recursive equations and the like. It's very difficult to interpret the data from complex arm motions, so at the moment Gutierrez's arm moves only in two dimensions instead of three, and doesn't have the mechanical arm's second elbow. "I was going to do an arm with three links," says Gutierrez, "but I ended up doing it with two. The computer doesn't mind how complicated it gets, but it got very complicated very quickly."

"It was a simplified arm, but it will still be very useful as a pilot project," says Rodriguez. "Roman developed a good motion analysis, and the programming is extendable." And if all else fails, its two-dimensional motions would be ideally suited for doing the space station's windows.□—*DS*