N88-17275

THE USE OF COMPUTER GRAPHIC SIMULATION IN THE DEVELOPMENT OF ROBOTIC SYSTEMS

Ken Fernandez

National Aeronautics and Space Administration
Marshall Space Flight Center
Information and Electronic Systems Laboratory
Huntsville, Alabama 35812

ABSTRACT

This paper describes the use of computer graphic simulation techniques to resolve critical design and operational issues for robotic systems. Use of this technology will result in greatly improved systems and reduced development costs. The major design issues in developing effective robotic systems are discussed and the use of ROBOSIM, a NASA developed simulation tool, to address these issues is presented. Three representative simulation case studies are reviewed: off-line programming of the robotic welding development cell for the Space Shuttle Main Engine (SSME); the integration of a sensor to control the robot used for removing the Thermal Protection System (TPS) from the Solid Rocket Booster (SRB); and the development of a teleoperator/robot mechanism for the Orbital Maneuvering Vehicle (OMV).

KEYWORDS

Robotics, Simulation, Computer-graphics, CAD, CAM, Off-line-programming, Robotic-welding, Robotic-spraying, Satellite-servicing.

INTRODUCTION

Robotic systems have become increasingly important to all facets of manufacturing: space is no exception. Perhaps the most publicized space robot is the Remote Manipulator System (RMS) which was built by Canada for the U.S. Space Shuttle. Prior to the RMS, robot manipulators were used on unmanned spacecraft to investigate soil properties on the moon and on Mars. Plans for the U.S. Space Station which will become operational in the early 1990's include the use of teleoperators and robots to perform routine station tasks e.g., inspection and maintenance. Earth-bound robots have also been used extensively to support the manufacturing of spacecraft components (Fernandez 1983,1985). Although the applications for space and earth seem radically different there remain many common issues in the procedures for design and testing of robot systems. Graphic simulation has proven to be extremely effective in the design of both types of system. In this paper we will examine: design issues for robots; ROBOSIM, a NASA developed computer graphic simulation tool; and three robotic systems that were developed using computer graphic simulation techniques.

Kinematic Design Issues

In designing a robot cell the selection of the robot's kinematic design is usually considered first. The number of robot joints, the type of joint (revolute or prismatic), and the physical configuration of each jointed segment are all elements of the robot's kinematic design. The position of the last reference frame (hand frame) is determined by the joint positions and the geometric relationships (kinematics). Minor changes in the kinematic design of a manipulator can greatly affect the volume through which the robot's hand may be moved. The design of the end-effector (tool) and the orientation of the part (workpiece) with respect to the robot (part positioning) also greatly affect the ability of a robot to perform a given task. For applications which will use an existing robot the designer must choose the appropriate robot, design the workcell layout and part fixturing. For systems which will use a custom-built robot, the task of designing the robot is added. A mistake in the design of a cell without the use of computer graphic simulation may not be detected until the hardware integration phase. This can result in costly schedule delays, procurement of incorrect components, and a greatly increased system cost.

PRECEDING PAGE BLANK NOT FILMED

Robot Motion Control

Robot control development is another area which can benefit from the use of computer graphic simulation techniques. Robot control algorithms may be viewed as existing at two levels: the kinematic control level; and the path planning level. Kinematic control algorithms are a function of the arm's kinematic design. These algorithms relate the position of the end-effector's reference frame to the joint position commands required to achieve the commanded position. These algorithms are a software implementation of the inverse kinematic equations. Prior to the use of graphic simulation, the control programs were debugged by observing the robot's motion subject to the commands of the experimental computer program. For robot systems with relatively low lifting capacity, a faulty program resulted in little more than embarrassment for the developer, however robot capacities have increased to the point where payloads are in the hundreds or thousands of pounds. Mistakes in programming can be serious. Another difficulty encountered in using the actual mechanism in the debugging process occurs for robots designed for use in zero-G which may not operate in a one-G environment. Again graphic simulation is the indicated procedure for this type of development.

Robot Path-Planning/Verification

Robot path-planning is the process of developing the sequential position, orientation and velocity commands that the robot's end-effector must execute in order to perform the desired function. Most current industrial robots are programmed using a teach pendant to manually command the robot to the desired points, this is the on-line manual programming method. Manual programming is highly in-efficient since the robot must be taken out of service, the path generated manually, replayed for verification and ultimately executed. On robots whose path programming is changed infrequently this is not significant, but for systems in which programming must be flexible manual programming is not satisfactory. Just as numerically controlled (NC) machine tools have become entirely programmed by off-line algorithms, the programming of robots will also eventually all be automated. Graphic simulation is a vital step that must be performed prior to the execution of an off-line generated robotic path program. Simulation will verify that: (1) the path specified is correct for the task; (2) the inverse kinematic equation may be solved at all points along the path program (controllability); and (3) the arm or other components will not collide accidentally with obstacles within the workcell.

Robot Dynamics

In industrial applications the primary dynamics issues are that the robot chosen for a task is capable of handling the required payload weights and transport velocities. Industrial robots are typically rated for lifting capacity only. An approximation of the robot's ability to perform a task dynamically can be made through dynamic simulation of the loaded robot. The maximum joint loads recorded during the dynamic simulation are compared to the loads that result if the manipulator were statically loaded per the manufacturer's specifications. If these joint loads are exceeded by the dynamic tests, then the robot may not be capable of performing the task. Since this is only an approximation, a safety margin should be used in making the final decision.

Although dynamic simulation is important for industrial robot systems, it is mandatory for systems used in space. Manipulator mechanisms and joint actuators are limited in weight due to launch considerations. Power supply limits reduce the size and rating of the mechanism's actuators. Dynamic studies will help to insure that the planned robotic tasks do not exceed the limits of the mechanism. The zero-G environment may be an advantage for handling larger payloads than would be possible on earth, but the dynamic interactions of the loaded manipulator and its mounting platform are significant for a space based robotic system. The possibility exists for parasitic oscillations to occur between the manipulator and the spacecraft's attitude control system. Simulation studies may reveal the existence of these or other undesirable effects.

ROBOSIM OVERVIEW

Simulation Procedure

ROBOSIM was developed over a three year period at the Marshall Space Flight Center (MSFC) to facilitate the design and development of robotic systems. Prior to ROBOSIM, robotic simulations were limited to the construction of scale models. Using ROBOSIM the kinematic design of the manipulator mechanism and other workcell components are modeled via a simulation language. The model consists of solid primitive shapes which approximate the robot's shape and mass properties. The joint configuration and type, either revolute, prismatic or fixed, are also specified. Once modeled, ROBOSIM computes

the standard linkage parameters (Hartenberg 1955), the inverse kinematics and the manipulator's dynamics. The designer may also specify the joint actuator transfer functions. Path motion is specified by position and velocity language constructs.

ROBOSIM Hardware Configuration

ROBOSIM is resident on a Digital Equipment Corporation (DEC) VAX11/780 processor. During simulation development the user may use a low cost terminal with TEK 4014 graphics compatibility. Although a simulation may be executed using a non-real-time terminal, the use of a real-time graphics display is preferred. Interfaces have been provided for several dynamic display systems including Evans & Sutherland PS330, GTI Poly 2000, Silicon Graphics IRIS with other interfaces planned. A limited Initial Graphics Exchange Standard (IGES) pre- and post-processor allows ROBOSIM to communicate graphics and tool motion commands with any CAD/CAM system adhering to the standard which was developed by the U.S. National Bureau of Standards.

The simulator's speed for non-dynamic studies is greater than real-time. This speed is decreased for very large models with multiple robots or robots with many degrees-of-freedom. Studies that required the modeling of dynamic effects also load the simulation processor. An Applied Dynamics AD10 parallel processor is used to improve the simulator's response in these situations.

ROBOSIM Software System Structure

ROBOSIM's software structure may be characterized as a hierarchy of three levels of software utilities. This structure is typical of large software systems. At the core or kernel of this system are routines that provide support for the most rudimentary of simulation tasks. Included among these functions are vector and matrix arithmetic and display control. The typical user of ROBOSIM interacts with these routines indirectly through his use of higher level utilities. A characteristic of routines at this level is their inflexibility in their interfacing requirements i.e., data must be provided in specific formats. By interfacing via the higher levels a user avoids these requirements, however direct access is available when needed. Typically, a ROBOSIM user who is performing simulation studies involving externally supplied mechanism control algorithms must communicate directly with the kernel routines.

The second level within ROBOSIM integrates the lower level routines into more complex algorithms that perform often needed tasks in display management and robot control. Examples of graphics routines that function at this level include subroutines to perform viewpoint and perspective transformations. Examples of routines that service robot kinematics and control issues include those which perform end-effector position computations and formulations of the manipulator's Jacobian matrix.

The highest level within ROBOSIM provides the human interface. At this level robots, workpieces, and fixturing assemblies may be modeled, placed within a workcell, programmed, dynamically simulated and viewed using fewer than forty distinct language instructions. The simplicity of this software interface greatly increases ROBOSIM's use and it is this interface that is perhaps the most important feature of ROBOSIM.

SIMULATION EXAMPLES

ROBOSIM V1.0 became operational in July 1985. In the year since, ROBOSIM has been applied to numerous robot simulation studies, the three listed below are typical. The studies include: the development of an off-line programming algorithms for welding on the SSME; the development of vision sensor guided control for a robot used to refurbish the SRB; and the design of a robot manipulator for the Orbital Maneuvering Vehicle. For each study a discussion of the application, the simulation goals, and the results will be presented.

Downhand Control for SSME Robotic Welding

The Space Shuttle Main Engine is constructed of stainless steel using over 2000 welded seams. At the present time 30% of these welds are performed by fixed automation while the remaining 70% are performed manually. A study performed of the manufacturing operation indicates that an additional 30% could be automated in a cost-effective manner using robotic welding techniques. The primary goal of this effort is the improvement of weld quality and reliability. A further improvement in manufacturing efficiency could be obtained by using automatic off-line robot programming techniques with downhand welding control. Downhand welding is the term applied to arc welding with the part in an orientation that maintains the weld puddle in a horizontal plane. This allows increased puddle size with a resulting greater deposition rate, fewer passes and reduced welding times.

Manual robot programming to perform downhand welding is extremely tedious and the results are only approximate. The algorithm for automatic off-line programming of the downhand position (Fernandez 1986) was developed using ROBOSIM as a test bed. The algorithm programs the robot and part positioner so that their coordinated motion results in a constant weld travel speed while maintaining the downhand position. Figure 1 depicts the robot cell with the six degree-of-freedom robot and a two degree-of-freedom part positioner. The part in figure 1 is a corrugated metal sheet. The part geometry may be read from a CAD data base using IGES format, or it may have been inferred from a manually generated path program sent to the downhand algorithm via the robot's communication interface. In either case the algorithm computes the desired local vertical in a reference frame moving along the weld seam at the specified weld velocity. Weld positioner commands are computed so that the desired downhand orientation is achieved. Robot position and velocity commands are also generated to keep the torch moving in the weld seam at a constant surface feed rate. Figure 2 depicts several frames from the simulation of the downhand welding algorithm. In figure 2 we note that the algorithm is functioning since the tangent to the weld seam remains horizontal at the point where the torch is in contact.

Vision Guided Off-Line Programming for SRB Refurbishment

The Solid Rocket Boosters used to assist in launching the Space Shuttle are designed to be re-used. To achieve this the Thermal Protection System (TPS) prevents the erosion of the booster's casing during the heat of re-entry. The main component of the TPS is the Marshall Sprayable Ablator (MSA) which reduces the booster's skin temperature by controlled evaporation. After recovery at sea the SRB is returned to the booster processing facility at the Kennedy Space Center (KSC).

High-pressure waterblast (20000 psi) is used to remove the partially burned ablative material prior to its re-application. Due to the difficulty in performing the cleaning operations manually, robotic workcells were developed (Fernandez 1983). Prototypes of these workcells were implemented at the MSFC Industrial Productivity Facility in Huntsville, Alabama. A computer graphic simulation of the prototype robotic cell is shown in figure 3. The robot, a Cincinnati Milacron HT3, is equipped with the high pressure nozzle. The aft booster section is shown mounted on a computer controlled rotary positioning table. In the initial implementation of this cell, manual robot programming methods were employed. The current operation includes both manual and off-line programming techniques. One problem in the operation of this cell occurs when the water blast fails to remove the MSA in the first cleaning pass. At this point the robot and turntable must be re-programmed manually to perform the touch-up cleaning.

A solution to the problem of programming the robot to perform touch-up cleaning of the TPS residue is the development of a vision sensor and off-line programming utilities. Graphic simulation via ROBOSIM was used to develop these programming utilities without the danger of damaging the actual workcell during initial development and de-bugging procedures. In operation the vision sensor will scan sections of the SRB that are presented by rotating the turntable. Due to the spray and debris real-time visual inspection is not possible, instead the inspection is performed after the entire cleaning pass is completed. The vision algorithm within the sensor provides information on the location of the MSA residues as x and y-coordinate locations referenced to the image plane of the sensor camera. Although the vision routines were developed under a separate effort, the camera is simulated in the graphic system by placing an "eye-point" in the same location and orientation as the hardware system. The focal length of the perspective transformation (Duda 1973) associated with the "eye-point" is adjusted to match the field-of-view of the sensor camera's lens. In evaluating the off-line programming algorithm a simulated MSA residue is placed on the modeled SRB. The residue is placed within the field-of-view of the "eye-point" by rotating the turntable in the graphics model. To simulate the sensor's output the screen x and y-coordinates are noted and passed as input to the off-line programming utilities in a manner similar to the actual sensor. The resulting turntable and robot motion commands were executed by the graphic model, and the resulting operation was viewed in graphics to determine if proper cleaning motion would have occurred. This result is established when the graphic representation of the spray (dotted line in figure 3) impinges on the simulated residue. The use of graphic simulation will continue when the sensor is integrated into the cleaning workcell. During operations the simulation will serve as a preview verification of the off-line generated cleaning paths.

Design of A Robot for the Orbital Maneuvering Vehicle

The Orbital Maneuvering Vehicle is designed as a re-useable, remotely controlled, free-flying vehicle capable of performing a wide range of on-orbit services in support of orbiting assets. It is projected as an important element of the Space Transportation System (STS), designed to operate from either the Shuttle , the Space Station or from the ground. The descriptions of the OMV or manipulator mechanism contained in this paper are not specific to any designs which may be currently under consideration by the U.S. National Aeronautics and Space Administration, however, the functional concepts described are correct and have been published elsewhere (Huber 1984).

The concept of the OMV includes the ability to accept mission kits to allow it to perform a variety of tasks in addition to its role as recoverable booster. One such kit is a manipulator/teleoperator, the "Smart Front-End" (SFE), which will allow remotely controlled manipulation to accomplish satellite and Space Station service tasks on-orbit. Figure 4 illustrates this concept. The OMV is shown equipped with a generic SFE manipulator. The SFE pictured consists of a bi-lateral pair of six degree-of-freedom (DOF) manipulators and a manipulator transport mechanism. The transport system provides three DOF: a rotary track which encircles the docking adapter; a hinged boom; and a sliding joint allowing the bi-lateral pair to traverse the boom. The generic satellite which is being serviced in figure 4 is shown detached from the OMV/SFE cluster for clarity. In normal operation a solid connection would be established by a docking mechanism.

ROBOSIM will be used extensively to assist in the development and evaluation of concepts for the SFE manipulator. Kinematic studies will reveal whether the SFE mechanism can be folded and stored within the space allocated on-board the Space Shuttle. Other kinematic studies will be required to determine if the OMV/SFE cluster can be successfully deployed from the cargo bay by the Space Shuttle's RMS. In figure 5 our generic OMV/SFE cluster is shown with the SFE folded in the stowable configuration. Further kinematic studies will determine if collisions between the SFE manipulator and satellite appendages occur during the execution of planned motion paths.

The implementation of an SFE manipulator will also require the development of several modes of mechanism control. An algorithm to control the SFE during deployment or un-folding will be developed. Although this type of algorithm usually involves a predetermined sequence of joint motions, provision must be included to override this sequence, if necessary, and execute new motions to correct or avoid anomalies. During docking operations the mechanism can take a passive or an active role. If a passive role is assumed, control algorithms for the SFE can improve the maneuverability of the OMV by arranging the arm's configuration to minimize inertial imbalance, avoid obstruction of the target satellite and prevent the reaction control system (RCS) thruster plumes from impinging on the SFE. Strategies of controlled compliance in the SFE joint servo control loops may further improve the controllability of the OMV during fine docking maneuvers by de-coupling the SFE's mass or actively using the SFE's momentum to affect additional control.

Once the OMV is docked with the target satellite a variety of different control issues must be resolved. As previously mentioned, algorithms that use mechanisms with kinematic redundancy to avoid collisions and minimize disturbance torques could significantly improve the system's performance. Real-time computer graphic simulation coupled to prototype teleoperator workstations can aid in resolving many issues relating to man-in-the-loop control. The placement of cameras may be simulated to insure that the field-of-view (FOV) is not obstructed. If a dual arm SFE design is chosen, graphic simulation could help to determine the most effective human interface for controlling the bi-lateral mechanism. Graphic simulation will not end with the successful SFE design, during servicing activities, a graphic display will allow the human operator to preview service tasks in simulation. Since communication delays in the man-in-the-loop control system may be large and varying, the use of a "predictive graphic display" to supplement the delayed visual feedback may improve the efficiency in performing operations remotely. When semi-autonomous or "supervisor control" methods are developed, the graphics display would allow the human to verify mechanism motions that are proposed by the controller. One final note relates to the design of the satellite rather than the OMV itself. Current satellite design philosophy is oriented toward multiple redundancy and no post-launch servicing, the advent of on-orbit service techniques will relax some of these design constraints, but satellite design must change to take advantage of these new possibilities. Hardware simulations of servicing missions on modular satellites have been performed (Fernandez 1980a,1980b,1984 and Scott 1985a,1985b),

but computer graphic simulation provides a cost-effective means of preliminary evaluation of the compatibility between a satellite and the servicer.

CONCLUSIONS

The experience gained at the Marshall Space Flight Center indicates that the use of computer graphic simulation in support of robot systems development is extremely important. Although hardware implementation is not replaced by these simulators, a considerable cost savings is experienced by delaying hardware implementation until the designs have matured. Once a robot system becomes operational the value of graphic simulation continues as a means of previewing planned task execution. It is expected that as the performance of computer graphic simulators increases and as hardware costs decrease the use of graphic methods will become widespread.

REFERENCES

Duda, R. O. and Hart, P. E. (1973). Pattern Classification and Scene Analysis. John Wiley & Sons, New York. Ch. 10, pp. 379-404.

Fernandez, K. R. (1980a). Computer Control of a Robotic Satellite Servicer. Proc. of the IEEE3 Southeastcon '80. Nashville, TN. pp.237-240.

Fernandez, K. R. (1980b). Application of a Computer Controlled Robot to Remote Equipment Maintenance. Proc. of the IEEE IASCON80. Cincinnati, OH. pp.1180-1184.

Fernandez, K. R., Jones, C. S. III, and Roberts, M. L. (1983). NASA's Use of Robotics in Building the Space Shuttle. Proc. of 13th ISIR/ROBOTS 7 of the SME. Vol. 1. Chicago, IL. pp.11-35 to 11-43.

Fernandez, K. R., Purinton, S. C., and Bryan, T. (1984). Simulation of a Robot System Used to Remotely Service Satellites. Proc. of the ANS Robotics and Remote Handling in Hostile Environments Nat. Topical Meeting. Gatlingburg, TN. pp.317-321.

Fernandez, K. R., et al (1985). In Robert L. Vaughn (Ed.) Space Shuttle: A Triumph in Manufacturing. SME Dearborn, Michigan. Chapter 5, pp.229-248.

Fernandez, K. R. and Cook, G. E. (1986). Computer Graphic Simulation of An Algorithm for Controlling Downhand Position in Robotic Welding. Proc. of the SME Conf. on Robotic Solutions in Aerospace Manufacturing. Orlando, FL. 10 pages.

Hartenberg, R. S. and Denavit, J. (1955). A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. ASME Jour. of Applied Mechanics June, pp.215-221.

Huber, W. G. (1984). User's Guide for Orbital Maneuvering Vehicle. NASA, MSFC. 12 pages.

Scott, D. R. (1985a). Remote Satellite Servicing. Proc. of the NASA Workshop on Proximity Operations in Earth Orbit. Houston, TX. 14 pages.

Scott, D. R. (1985b). Concepts in Remote Satellite Servicing. Proc. of the NASA Workshop on Satellite Servicing. Houston, TX. 16 pages.

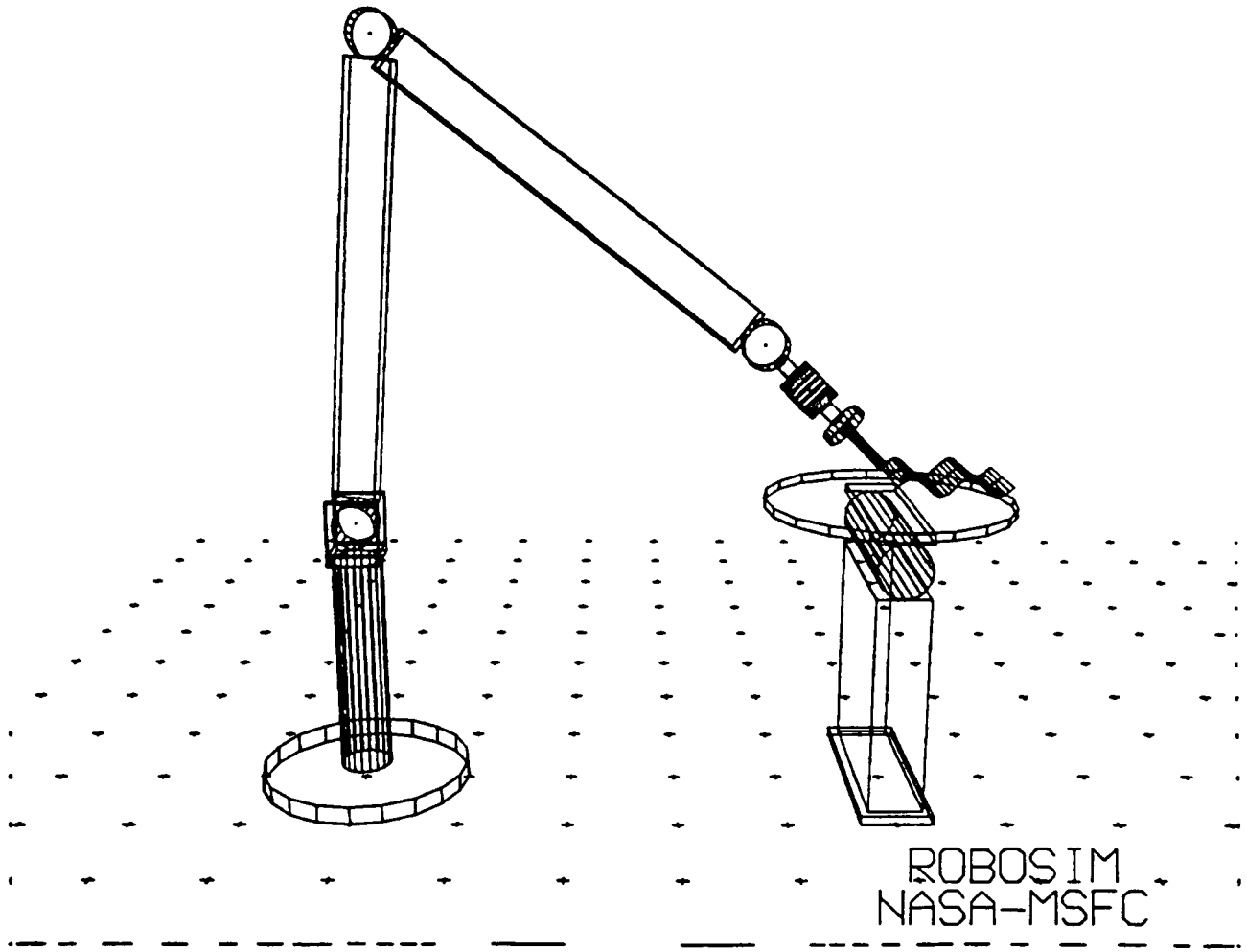

Fig. 1. Simulated Six Axis Robot and Two Axis Positioner

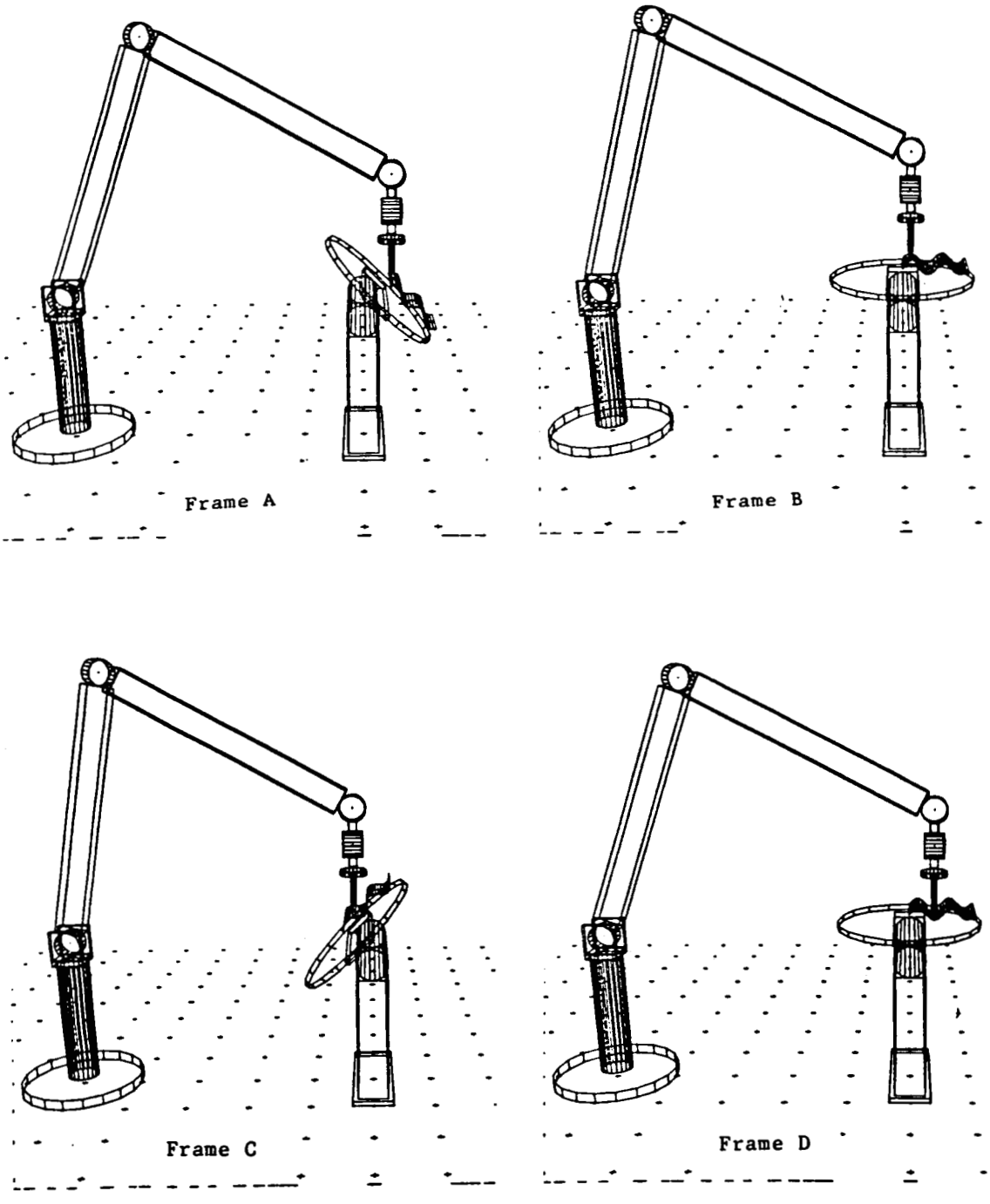

Fig. 2. Simulated Weld Operation with Automatic Downhand Position

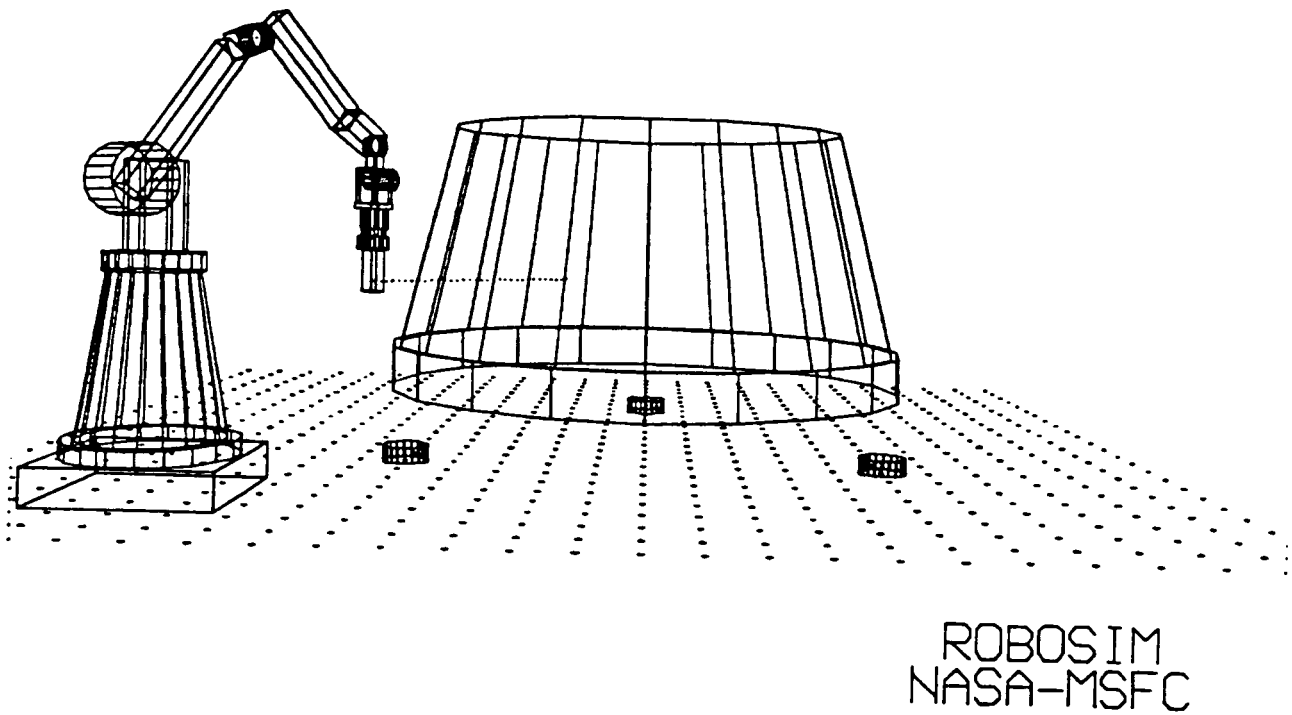

Fig. 3. Simulated SRB Refurbishment Workcell

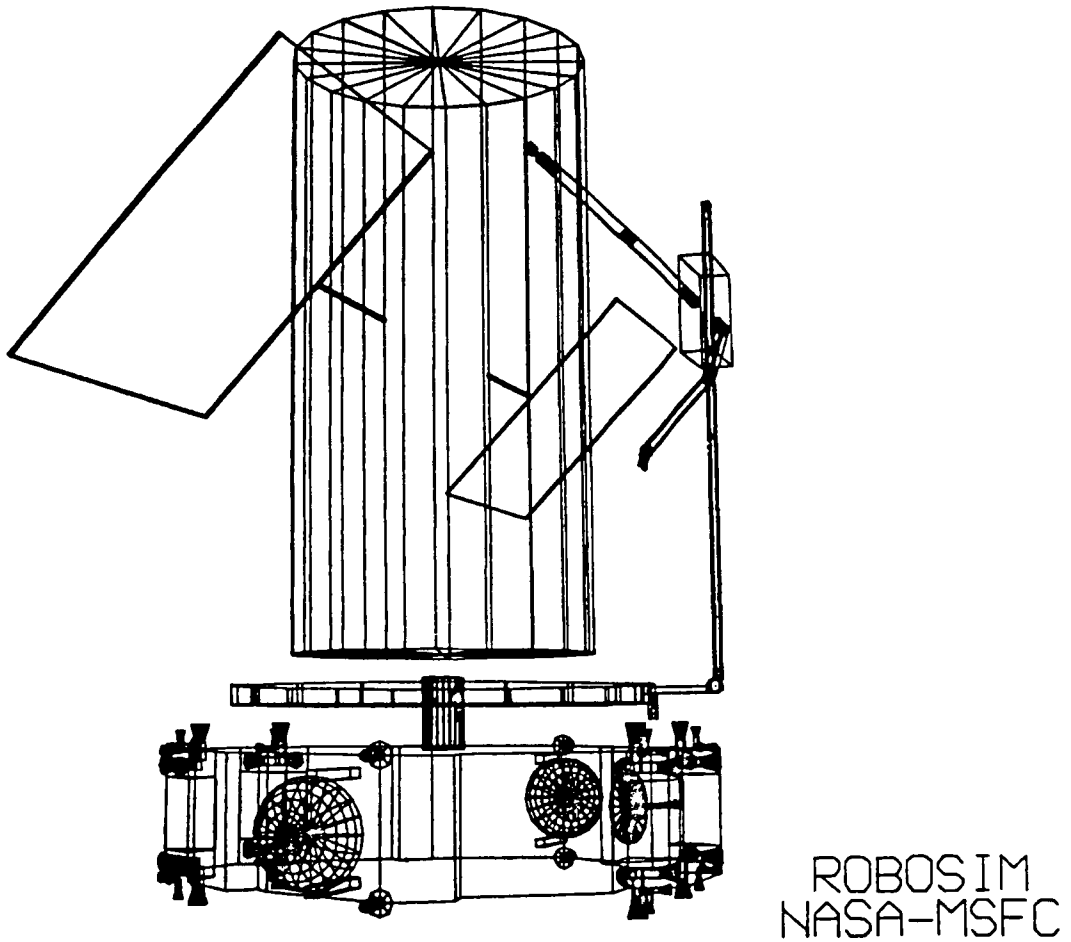

Fig. 4. Generic OMV Performing Satellite Servicing

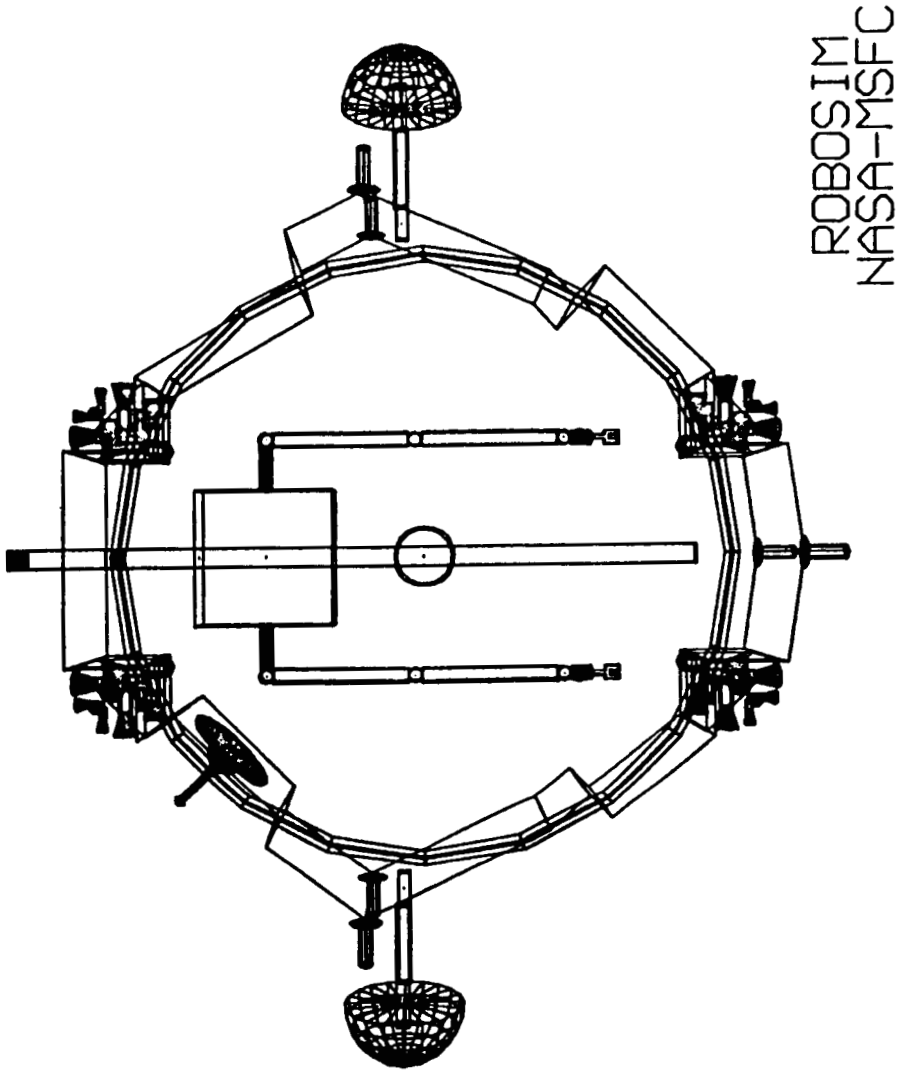

Fig. 5. OMV Shown with SFE Manipulator in Stowed Position