NASA Contractor Report—178050

Evaluation of Automated Decisionmaking Methodologies and Development of an Integrated Robotic System Simulation

Martin Marietta Aerospace
Denver Aerospace
P.O. Box 179
Denver, Colorado 80201

Contract NAS1-16759

March 1986

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

EVALUATION OF AUTOMATED DECISIONMAKING METHODOLOGIES AND DEVELOPMENT OF AN INTEGRATED ROBOTIC SYSTEM SIMULATION

Prepared by:

Dennis C. Haley
Bonni J. Almand
Mark M. Thomas
Linda D. Krauze
Keith D. Gremban
James C. Sanborn
Joy H. Kelly
Thomas M. Depkovich
William J. Wolfe
Thuy Nguyen

This work was performed for NASA Langley Research Center under Contract NAS1-16759.

MCR-85-665

MARTIN MARIETTA
DENVER AEROSPACE
P.O. Box 179
Denver, Colorado 80201

CONTENTS

SUMMARY

The computer simulation called ROBSIM, developed under the four phases of this contract provides the capability to perform kinematic and dynamic analyses of user defined, rigid link manipulators. The kinematic analysis provides positions, velocities, and accelerations of all parts of the manipulator for a prescribed motion. The dynamic analysis includes requirements analysis which calculates system loads for specific motions, and response simulation which calculates the motion resulting from a prescribed set of driving torques or feedback control law. This report documents the fourth phase of the contract which built on the basic capabilities developed in the phases one through three.

The ability to utilize CAD/CAM generated data to model system components was added. This feature allows the user to access a file of CAD/CAM data written according to the Initial Graphics Exchange Specification, version 2.0. The ROBSIM preprocessor function can read this file and then write it in another format that can be read in and used for the detailed graphics modeling of system components. A graphics display is available during the reformatting procedure.

The capability to simulate multiple arms with movable bases was also added for this phase of the contract. A system may be modeled with all of the manipulators attached to a single movable base, or each robotic arm may be attached to its own independent base. With this capability, the user defines motion trajectories for both the manipulator joints or end effectors and the movable bases. Position and/or force calculations are then carried out for the bases as well as for the joints and links of the arms. This addition has great utility in light of the emphasis being placed on telerobotics and space station and satellite servicing.

To make the motion trajectory specification process easier, a task oriented motion specification module (or task command module) was added. This software allows the user to define a manipulator's motion by choosing command from a list or menu. Each command in the menu is a combination of some of the lower level commands currently in place in ROBSIM and is separated into these commands when actually implemented to control motion.

The fourth addition gives the user another option for controlling a manipulator's motion. This option simulates a video camera mounted on the end-effector of an arm tracking a target. Currently, only stationary targets are modeled.

INTRODUCTION

Background

This report documents the results of work performed in Tasks 21 through 24 of contract NAS1-16759, Evaluation of Automated Decisionmaking Methodologies and Development of an Integrated Robotic System Simulation. It was prepared by Martin Marietta Denver Aerospace for the NASA Langley Research Center in accordance with the contract statement of work. These tasks constitute Phase IV of an ongoing activity addressing technologies relevant to the design and operation of advanced manipulator systems. Phase I concentrated on the identifying and evaluating applicable artificial intelligence techniques and on developing a framework and mathematical models for the computer simulation. These results (Phase I) were documented in 1982 in NASA contractor reports 165975, 165976, and 165977. Phases II and III developed and implemented the software necessary to model a complete manipulator system and then perform kinematic and/or dynamic analyses. The results of work done under Phases II and II were documented in 1984 in NASA contractor reports 172401, 172402, and 172403.

This project is motivated by the realization that NASA advanced missions require the increasing use of automation technologies for both economic and performance reasons. Development of these complex technologies in a timely and cost efficient manner requires the extensive use of computer simulation tools that allow options to be evaluated and compared before building hardware prototypes.

Contract Objectives

The objective of this phase of the contract was to build on the simulation capabilities developed in Phases I through III, specifically by adding the following enhancements:

1) The ability to utilize geometry data from a CAD/CAM database to model manipulator system components;

2) The capability to simulate multiple arms on a single movable base or multiple movable bases;

3) The addition of a task command module to make manipulator motion specification easier. Each task command is broken down into a series of lower level primitives previously used to define a motion;

4) The ability to control a manipulator's motion by simulating a video camera mounted on the end-effector tracking a stationary target.

Report Organization

This report consists of three volumes--the task or study results and two appendices. The study result volume documents the technical aspects of the software developed and implemented for Tasks 21 through 24 of contract NAS1-16759. Each task is reviewed in a separate section of the report, along with any impact it might have had on other parts of the program. Appendix A is the updated users guide, which reflects any changes caused by implementation of Tasks 21 through 24. Any page that was changed will be marked Rev A, October 1985 on the upper outside corner. Appendix B is the programmers guide and contains Visual Control Logic Representations (VCLRs) of all the subroutines as well as an outline of the program structure. As in Appendix A, any pages that were changed or added as a result of incorporation of the of the four new tasks are marked Rev A, October 1985.

TASK 21, USE OF CAD/CAM DATA

Completion of Task 21 required the design of an interface between ROBSIM- and CAD/CAM-generated models. This interface enables the ROBSIM user to model the detailed graphics of a system more easily by accessing and using data converted from a CAD/CAM model file. This can be used for the detailed graphics modeling of manipulator components as well as for modeling the system environment, target objects, and load objects.

The interface designed for this task reads CAD/CAM data formatted according to the Initial Graphics Exchange Specification (IGES) version 2.0. This specification (IGES) is supported by several vendors of CAD/CAM systems to facilitate the exchange of information between different systems. IGES version 2.0 supports a number of geometric entities; however, for this implementation of ROBSIM the following entities are translated:

1) Points;
2) Lines;
3) Arc Segments;
4) Transformations.

These entities are decomposed into individual (x, y, z) points that represent their polynomial curves, and are written to files that may be read in as obstacle entities for detailed graphics modeling during system definition or simply displayed on a graphics terminal. A line is defined by its two endpoints, whereas a circular arc segment is divided into 10 linear segments. Transformations include both linear and rotational components and may be applied to any of the other geometric entities. Figure 1 shows a display of a manipulator generated on a Computer-Vision CAD/CAM system, and converted using ROBSIM for display on an Evans and Sutherland graphics system. Figures 2 through 4 show the use of the interface software to create the detailed representations of the first three links of the manipulator.

The work done for this task demonstrates the usefulness of this effort, and the need for work in this area to continue. However, certain concerns should be noted before extensive development is conducted.

During development of the software for this task, files were generated on three different CAD/CAM systems that all wrote files to the IGES version 2.0 specification. In all of the files the data needed to define the geometric entities were there, but the formats varied. This indicates a need for more work in the area of standardization. In addition, CAD/CAM systems are still closer to automated drafting tools, than a tool for parts definition. Geometric features such as flanges, holes or fillets are not defined, only low-level entities such as arcs, splines, and lines are. To accomplish feature definition solids definition, and the definition of other intrinsic properties, further revisions to IGES and evolution in the product definition database format are needed.

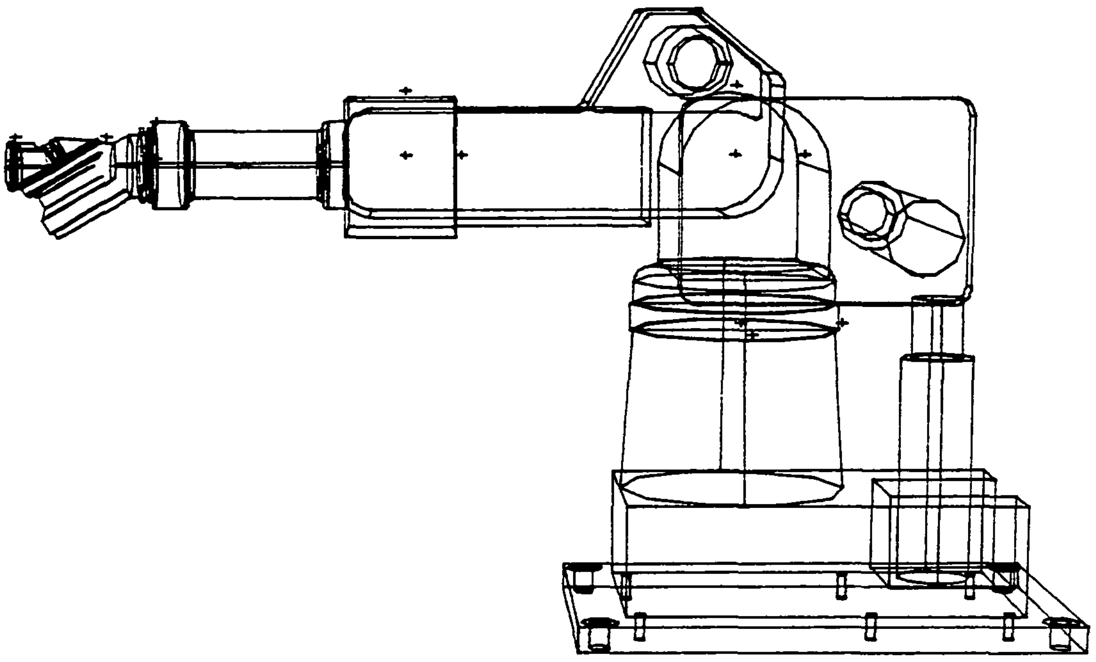


Figure 1. - CAD/CAM-generated representation of T3 manipulator.

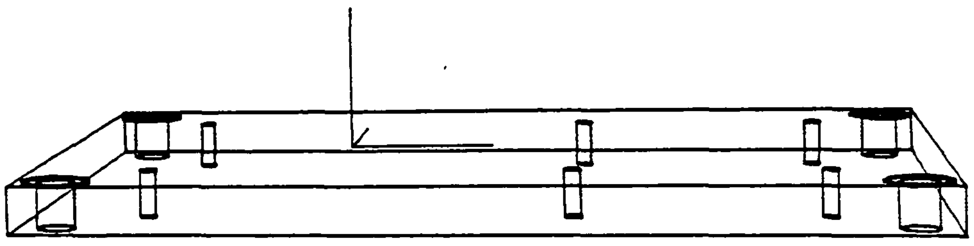


Figure 2. - Base link modeled with CAD/CAM data.

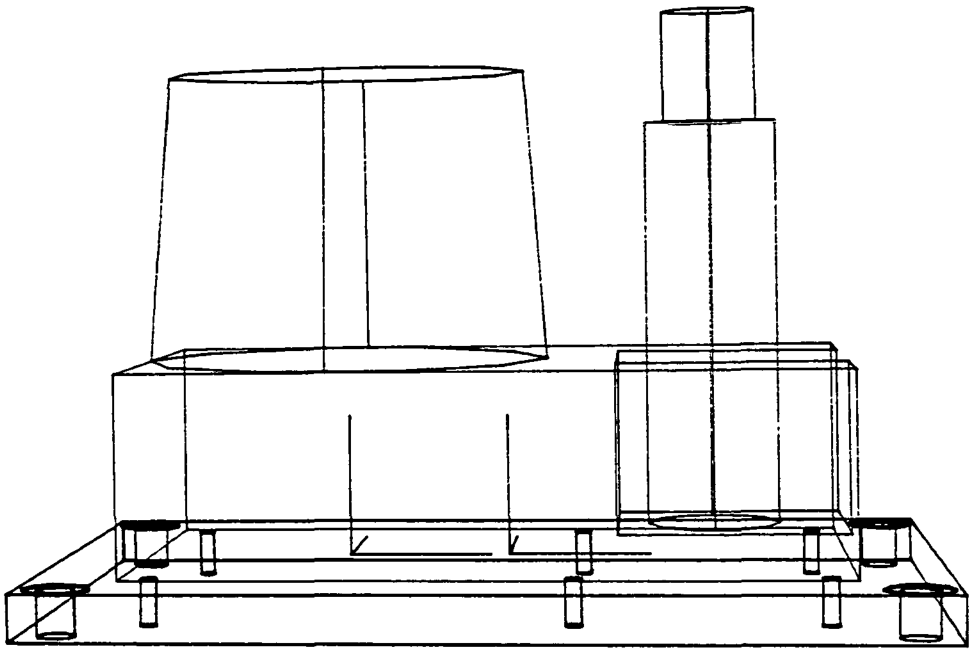


Figure 3. - Base and link 1 modeled with CAD/CAM data.

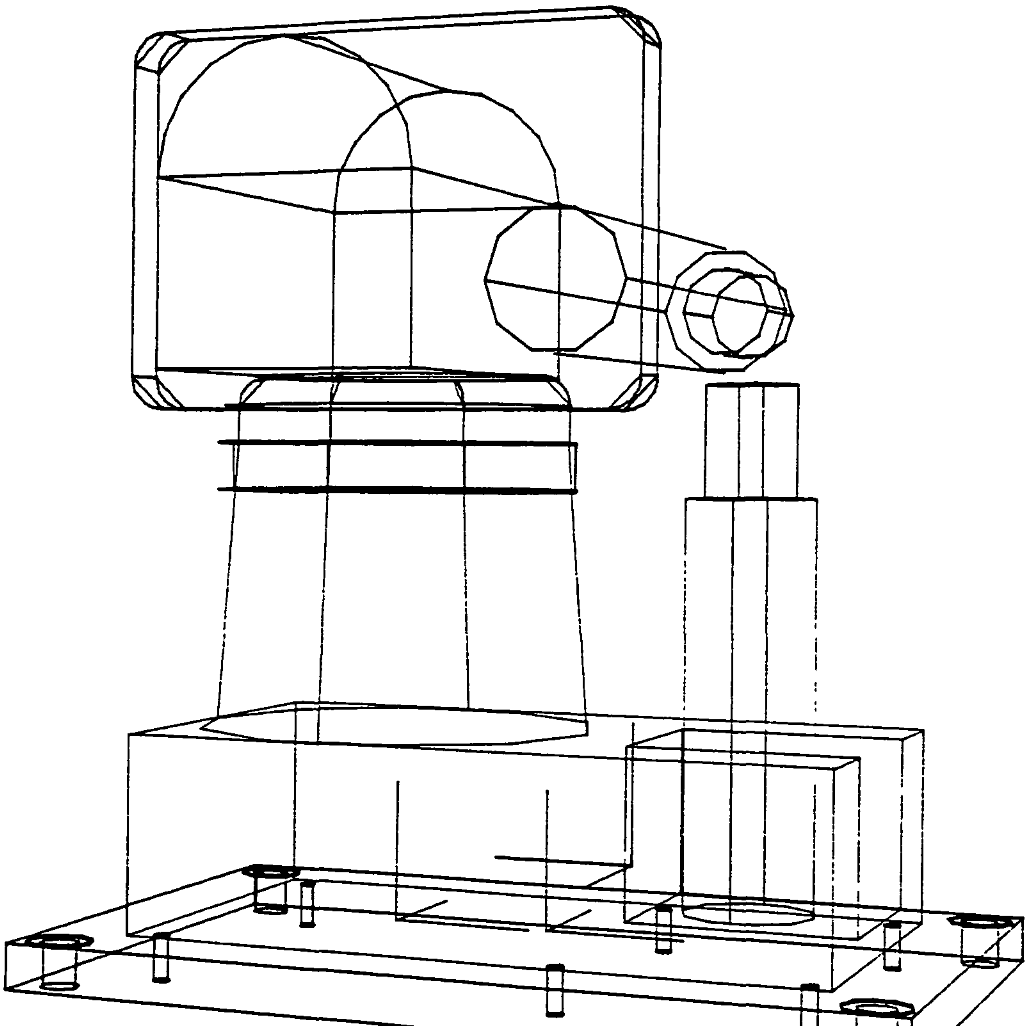


Figure 4 - Base, link 1 and link 2 modeled with CAD/CAM data

In an effort to address some of the shortcomings of the IGES specification, the McDonnell Aircraft Company has been working on a Product Definition Data Interface (PDDI) under contract to the Air Force Wright Aeronautical Laboratories (AFWAL). PDDI attempts to fill in some gaps and provide another step in the evolution to a more universal standard that would be an interface between all engineering, design, development, and manufacturing functions. Some of the milestones of this evolution are shown in Figure 5. More information on PDDI may be obtained from the ICAM CM Library, Wright-Patterson AFB, OH 45433-6533.

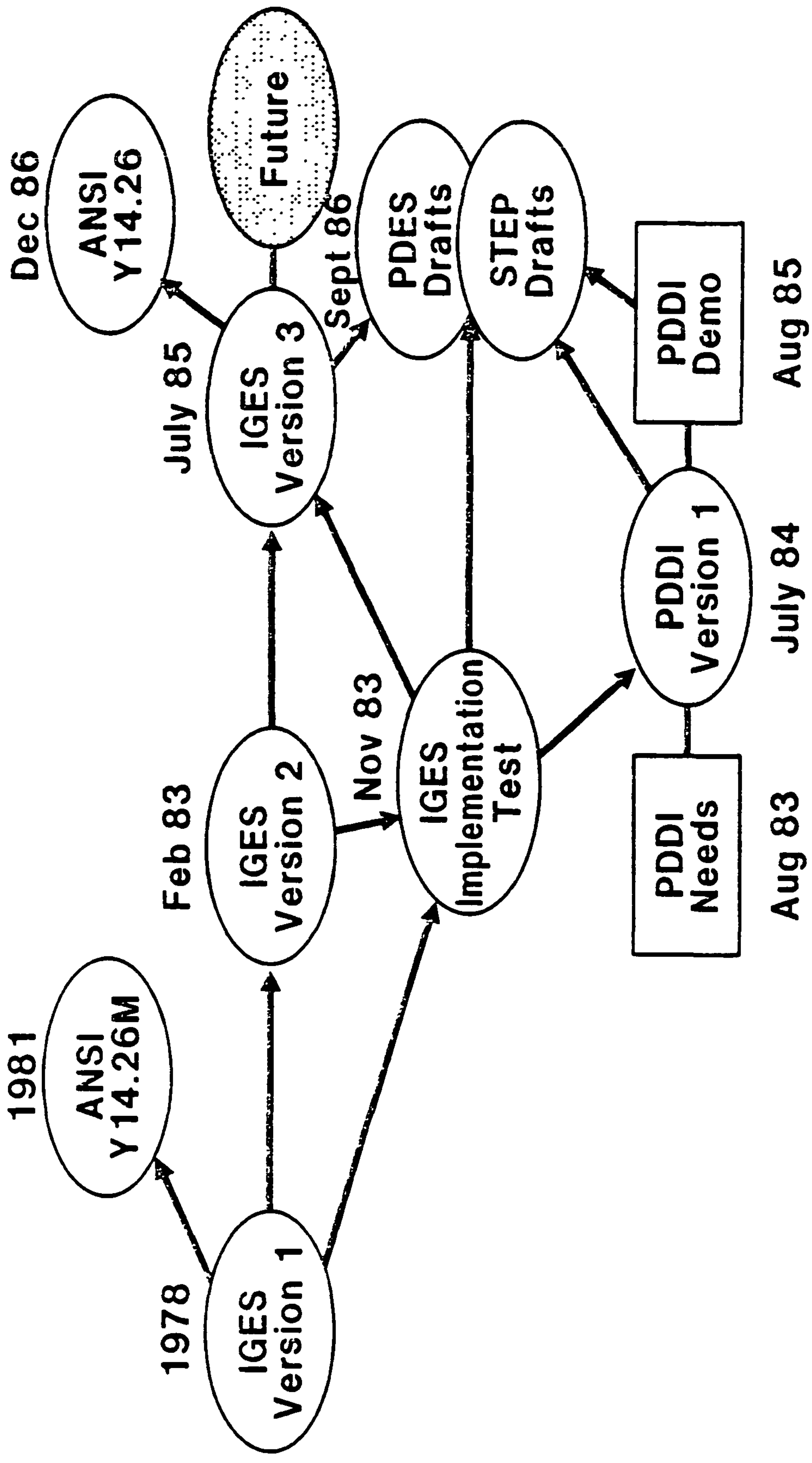


Figure 5. - Product database evolution.

TASK 22, MULTIPLE ARMS WITH MOVING BASE

Task 22 extended the analysis capabilities of the ROBSIM program to include multiple manipulator systems with moving bases. The ability to determine forces and moments at the base of each arm in the system is included with the moving base capability. This section of the report is divided into the following three subsections:

1) Kinematic analysis
2) Requirement analysis
3) Response simulation

Kinematic analysis

The forward kinematic solution for a manipulator with a moving base is identical to that described in NASA contractor report 172401. That is, starting with the specified velocities and accelerations for the base, the velocities and accelerations of each link are obtained by successive application of the following recursive relations:

$$\underline{\omega}_{i+1} = \underline{\omega}_i + \dot{\theta}_{i+1} \underline{S}_{i+1} \text{ (rotating joint i + 1)}$$

$$\underline{\omega}_{i+1} = \underline{\omega}_i \text{ (sliding joint i + 1)}$$

$$\underline{V}_{i+1} = \underline{V}_i + \underline{\omega}_i \times (\underline{R}_{i+1} - \underline{R}_i) \text{ (rotating joint i + 1)}$$

$$\underline{V}_{i+1} = \underline{V}_i + \underline{\omega}_i \times (\underline{R}_{i+1} - \underline{R}_i) + \dot{\theta}_{i+1} \underline{S}_{i+1} \text{ (sliding joint i + 1)}$$

$$\underline{\alpha}_{i+1} = \underline{\alpha}_i + \underline{\omega}_i \times \dot{\theta}_{i+1} \underline{S}_{i+1} + \ddot{\theta}_{i+1} \underline{S}_{i+1} \text{ (rotating joint i + 1)}$$

$$\underline{\alpha}_{i+1} = \underline{\alpha}_i \text{ (sliding joint i + 1)}$$

$$\underline{a}_{i+1} = \underline{a}_i + \underline{\omega}_i \times [\underline{\omega}_i \times (\underline{R}_{i+1} - \underline{R}_i)] + \underline{\alpha}_i \times (\underline{R}_{i+1} - \underline{R}_i) \text{ (rotating joint i + 1)}$$

$$\underline{a}_{i+1} = \underline{a}_i + \underline{\omega}_i \times [\underline{\omega}_i \times (\underline{R}_{i+1} - \underline{R}_i)] + \underline{\alpha}_i \times (\underline{R}_{i+1} - \underline{R}_i) + 2\underline{\omega}_i \times \dot{\theta}_{i+1} \underline{S}_{i+1} + \ddot{\theta}_{i+1} \underline{S}_{i+1} \text{ (sliding joint i + 1)}$$

where:

$\underline{\omega}_{i+1}$:	angular velocity of link i + 1
$\underline{\omega}_i$:	angular velocity of link i
$\dot{\theta}_{i+1}$:	joint i + 1 relative angular velocity
$\underline{S}_{i+1}$:	unit vector along joint i + 1 rotational axis
$\underline{V}_{i+1}$:	translational velocity of joint i + 1
$\underline{V}_i$:	translational velocity of joint i
$\underline{R}_{i+1}$:	vector locating joint i + 1 in the world frame
$\underline{R}_i$:	vector locating joint i in the world frame
$\underline{\alpha}_{i+1}$:	angular acceleration of link i + 1
$\underline{\alpha}_i$:	angular acceleration of link i
$\ddot{\theta}_{i+1}$:	joint i + 1 relative angular acceleration
$\underline{a}_{i+1}$:	translational acceleration of joint i + 1 in the world frame
$\underline{a}_i$:	translational acceleration of joint i in the world frame

All vectors used in the preceeding equations are expressed in terms of the world reference frame, and the origin of the $[X_i]$ frame is placed at joint i (Fig. 6).

To accommodate base motion analysis, an option was added to allow the user to define a base time history profile. This new option is very similar to end-effector control of a manipulator using the time history profile specification. The two options available for defining base motion within each time segment are:

1) Rate control;
2) Position control.

These options and their implementations are discussed in NASA contractor report 172401. The outputs generated from this option are the base position, velocities and accelerations that are used as inputs to the recursive equations previously stated to determine the link velocities and accelerations.

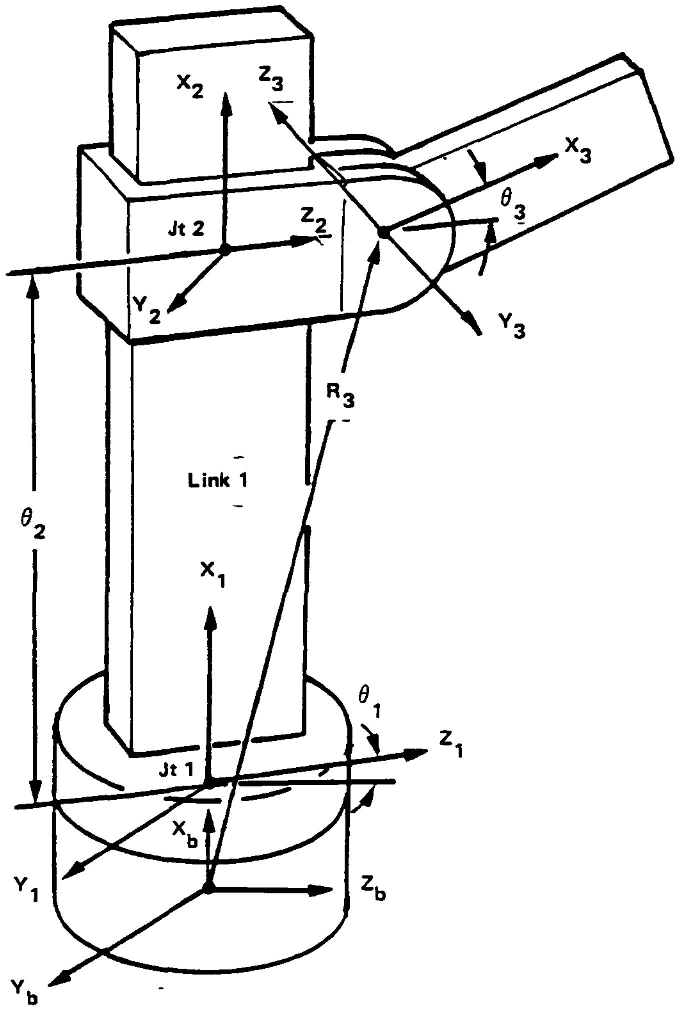


Figure 6. -
Kinematic representation of a serial manipulator

Requirement analysis.

The reaction forces and torques at the manipulator joints are evaluated as discussed in NASA contractor report 172401. The forces and torques are successively computed starting from the terminal link of the arm and proceeding back to the base. Assuming the only external load applied to the intermediate links are the gravity forces acting through their centroids, the recursive equations for the joint reactions are

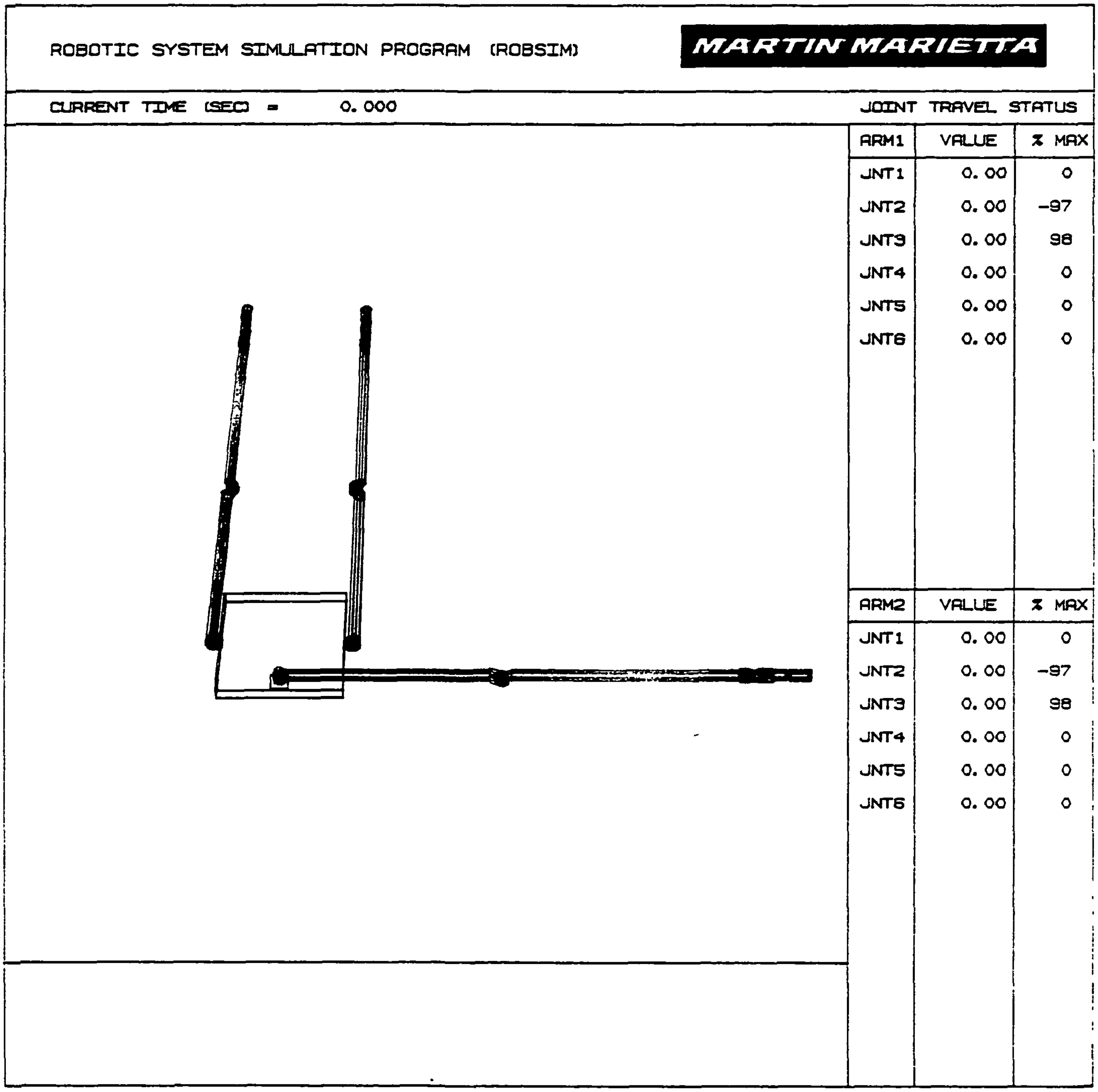
$$\underline{f}_i = \underline{f}_{i+1} + m_i (\underline{a}_{cgi} + \underline{g})$$

$$\underline{t}_i = \underline{t}_{i+1} + (\underline{R}_{i+1} - \underline{R}_i) \times \underline{f}_{i+1} + \underline{h}_{ii} \times m_i (\underline{a}_{cgi} + \underline{g})$$
$$+ [I_i] \alpha_i + \underline{\omega}_i \times [I_i] \underline{\omega}_i$$

where:

$\underline{f}_i$:	force at joint i
$\underline{f}_{i+1}$:	force at joint i + 1
m_i:	mass of link i
$\underline{a}_{cgi}$:	acceleration of centroid of link i
$\underline{g}$:	acceleration of vector due to gravity
I_i:	inertia matrix of link i in world frame
$\underline{h}_{ii}$:	vector from origin of link i to the centroid of that link
$\underline{t}_i$:	reaction torque at joint i
t_{i+1}:	reaction torque at joint i + 1

All of the vector quantities in these equations are expressed in the world coordinate system. Another capability added for this task, is the ability to put multiple arms on one moving base (see Fig 7). In this case, the reaction forces and torques at the base are computed as a sum of all the individual base reaction forces and torques contributed by each arm that is attached to the moving base.



ROBOTIC SYSTEM SIMULATION PROGRAM (ROBSIM) MARTIN MARIETTA

CURRENT TIME (SEC) = 0.000

JOINT TRAVEL STATUS

ARM1	VALUE	% MAX
JNT1	0.00	0
JNT2	0.00	-97
JNT3	0.00	98
JNT4	0.00	0
JNT5	0.00	0
JNT6	0.00	0

ARM2	VALUE	% MAX
JNT1	0.00	0
JNT2	0.00	-97
JNT3	0.00	98
JNT4	0.00	0
JNT5	0.00	0
JNT6	0.00	0

Figure 7 - Multiple arms on a single movable base.

A preferred method for modeling a system containing a moving base with multiple arms attached is to model one of the arms with the moving base geometric and mass properties, and the other arms with zero base properties. The locations of all first joints will then be referenced to a common base frame.

Response simulation

The difference between the new modification and the simulation discussed in NASA contractor report 172401 is the inclusion of the base motion in the dynamic equations of motion. These equations are solved for the base accelerations and the joint accelerations at each processing time step. To calculate the base and joint accelerations, the equations of motion are reformulated to add the six rigid-body base accelerations (three rotational, three translational). The controlling equation for an N-joint manipulator with a moving base is

$$\underline{\tau} = [A\ (\underline{\theta})]\ \underline{\ddot{\theta}} + \underline{b}\ (\underline{\theta}, \underline{\dot{\theta}}) \qquad [1]$$

where

$\underline{\tau}$:	(6 + N, 1) generalized torque vector containing 3 base reaction torques, 3 base reaction forces and N driving torques at N joints
$\underline{\ddot{\theta}}$:	(6 + N, 1) acceleration vector containing 3 base angular accelerations, 3 base translational accelerations and N joint accelerations
$[A(\underline{\theta})]$:	(6 + N, 6 + N) effective inertia matrix referenced to the base and joint coordinates, accordingly
$\underline{b}(\underline{\theta}, \underline{\dot{\theta}})$:	(6 + N, 1) vector of position- and velocity-related effective torques, including external loads, gravity, velocity-related inertia terms and friction

The calculation of these terms is described in NASA contractor report 172401, except for the new modifications to the calculations of $\underline{\tau}$ and $[A\ (\theta)]$.

Generalized torque, $\underline{\tau}$ - Besides the N-joint actuator torques this generalized torque also includes the three reaction torques and 3 reaction forces at the base. These base reaction torques and forces can be generated by two methods:

1) Read a file of base reaction torques and forces versus time;

2) Read a base motion profile (not currently implemented).

The base reaction torques and forces file which can be generated during a requirements analysis run could be used along with the joint actuator torque file to drive the response simulation , thereby validating that the requirements analysis and response simulation.

If a base motion is specified, part of the equations [1] will be solved for the N-joint accelerations using the known base motion profile and the N-joint torques. Then the backward recursive formulas are solved next for the base reaction forces and torques.

Effective inertia matrix, $[A^i]$ - As discussed in NASA contractor report 172401, the effective inertia matrix $[A^i]$ due to the link's mass distribution is given by

$$[A^i] = [J^i]^T \left[\begin{array}{c|c} [M_i] & [0] \\ \hline [0] & [I_i] \end{array}\right] [J^i]^T$$

where

$[M_i]$: (3, 3) diagonal matrix with link mass $[M_i]$ along the diagonal

$[I_i]$: (3, 3) link i inertia matrix about the link i centroid, referenced to the world frame

$[A^i]$: (6 + N, 6 + N) link i effective inertia matrix referenced to the base and joint coordinates

$[J^i]$: (6, 6 + N) Jacobian matrix relating the motion of link i to the joints. Because the base is modeled as a 6-DOF rigid body, the expressions:

$$\underline{J}_i^i = \begin{Bmatrix} \underline{S}_j \times (\underline{r}_{cgi} - \underline{R}_j) \\ \underline{S}_j \end{Bmatrix} \qquad \text{Rotating joint } j \leq i$$

and

$$\underline{J}_i^i = \begin{Bmatrix} \underline{S}_j \\ 0 \\ 0 \\ 0 \end{Bmatrix} \qquad \text{Sliding joint } j \leq i$$

can be applied to the base degrees of freedom $1 \leq j \leq 6$ with

$\underline{R}_j$ = vector locating the origin of the base frame, , referenced to the world system, for $1 \leq j \leq 6$

$\underline{r}_{cgi}$ vector locating the base cg in world system,

$\underline{S}_j$ = *unit vector along base axes, referenced to the world system.*

$$\underline{S}_1 = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \; i \quad \underline{S}_2 = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \; i \quad \underline{S}_3 = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \qquad \text{rotating base joints } 1 \leq j \leq 3$$

$$\underline{S}_5 = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \; i \quad \underline{S}_5 = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \; i \quad \underline{S}_6 = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \qquad \text{sliding base joints } 4 \leq j \leq 6$$

Therefore, for i = base

$$[J^b] = \left[\begin{array}{c|c|c|c|c|c|c|c|c}
\begin{Bmatrix}1\\0\\0\end{Bmatrix} \times (\underline{r}_{CGb} - \underline{R}_b) & \begin{Bmatrix}0\\1\\0\end{Bmatrix} \times (\underline{r}_{CGb} - \underline{R}_b) & \begin{Bmatrix}0\\0\\1\end{Bmatrix} \times (\underline{r}_{CGb} - \underline{R}_b) & \begin{matrix}1\\0\\0\end{matrix} & \begin{matrix}0\\1\\0\end{matrix} & \begin{matrix}0\\0\\1\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} \\
\begin{matrix}1\\0\\0\end{matrix} & \begin{matrix}0\\1\\0\end{matrix} & \begin{matrix}0\\0\\1\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix} & \begin{matrix}0\\0\\0\end{matrix}
\end{array}\right]$$

Similarly, for joints i from 1 to N, a component vector of $[J^i]$ can be written

$$\underline{J}^i_i = \begin{Bmatrix} \underline{S}_j \times (\underline{r}_{CGb} - \underline{R}_j) \\ \hline \underline{S}_j \end{Bmatrix} \qquad \text{Rotating joint } j \leq i$$

$$\underline{J}^i_i = \begin{Bmatrix} \underline{S}_j \\ 0 \\ 0 \\ 0 \end{Bmatrix} \qquad \text{Sliding joint } j \leq i$$

where: $1 \leq i \leq N$ and $1 \leq j \leq 6 + N$

For $j > i$ the components of $\underline{J}^i_j$ are zero because a displacement at joint j has no effect on the absolute motion of link i.

The total inertia matrix is calculated (as described in NASA contractor report 172401) using the symmetry of the inertia matrix, and a recursive procedure to complete the mass properties of the composite system of link i through N.

In the full simulation case where a torque file is input, the equations of motion are then solved for the base accelerations and joint accelerations in terms of the state (base positions and velocities, joint positions and velocities), and the generalized torque $\underline{T}$.

TASK 23, TASK COMMAND MODULE

Task 23 added a task command module to the motion specification section of ROBSIM. The intent of this task was to simplify the motion specification process by implementing a set of task commands, in which each task command is a combination of some of the motion commands previously used.

The commands for motion specification previously in place and described in NASA contractor report 172401 are now called primitives or primitive-level commands. The task commands added for this task are essentially just groupings of primitives. Both command levels are now in place and the user may define a motion specification file using either level. The primitive-level commands are grouped into two sections--motion and nonmotion. Table I lists these commands.

TABLE I. - PRIMITIVE-LEVEL COMMANDS

Motion	Nonmotion
joint rate control	grasp a load object
joint position control	release a load object
end effector rate control	change tool reference point
end effector position control	time delay or wait
	force/torque control on/off
	active compliance control on/off

The task level commands added in this phase of the contract were:

1) Pick up an object;
2) Place an object;
3) Normal or guarded arm movement;
4) Hold current position;
5) Change the end effector reference point;
6) Use operator control;
7) Set response simulation control mode;
8) Sensor control of end effector position.

When implemented, each of these commands is separated into primitives that are then written to a motion history file and used for motion control the same way as before. Sensor control of the end-effector position was also added to the primitive command level to keep consistancy between primitive and task level commands.

Picking up an object combines position control of the end-effector to move the arm to the load object and the nonmotion primitive grasp to combine the load object's mass properties with those of the end effector for subsequent motion. The position to which the end-effector moves is defined with the load object as the grasp point. This point is transformed to desired end-effector position at the end of the move by multiplying the grasp point by the load-to-world transformation matrix and adding this to the load object location vector

$$\underline{r}_p(t_f) = [P_1]\ {}_1\underline{r}_g + \underline{L}_1$$

where

$\underline{r}_p$ is the grasp point vector in world coordinates

t_f is the time at the end of the move

$[P_1]$ is the load to world transformation matrix

${}_1\underline{r}_g$ is the grasp point vector in the load coordinates

$\underline{L}_1$ is the location of the load object in world coordinates

The desired orientation of the end-effector at the end of the move is found from the approach vector and the end effector y-axis orientation

$$\underline{Z}_e = \underline{X}_e \times \underline{Y}_e \qquad [T_e] = [\underline{X}_e\ \underline{Y}_e\ \underline{Z}_e]$$

where

$\underline{X}_e$ is the load object approach vector (unit vector) in the world coordinate system (also a vector defining the direction of the end-effector x-axis in world coordinates)

$\underline{Y}_e$ is a unit vector defining the direction of the end-effector y-axis in world coordinates

$\underline{Z}_e$ is a unit vector defining the direction of the end-effector z-axis in world coordinates

$[T_e]$ is the end-effector-to-world transformation matrix

Placing an object combines position control of the end-effector to move the object to a desired location and the release command to return the end-effector mass properties to those it had before grasping the object. Implementation of this option uses the algorithms currently in place.

The move arm task command allows the user to specify either a normal or guarded move. A normal move is simply position control of the end-effector. A guarded move is also position control of the end-effector but is used when in close proximity to another object to avoid collision. The time allowed for the move is doubled (slowing down the motion) and force/torque control is turned on to detect if any collisions occur. No sensed force would indicate that the move went smoothly and a sensed force or torque would mean a collision had occurred and the path of motion must be altered.

The wait or time delay command holds the arm in its current position for a length of time specified by the user.

Changing the end-effector reference point changes the point for which motion is defined when the user controls the end effector instead of the individual joints.

The operator control command has not been implemented yet.

Setting the control mode allows the user to choose a control method other than proportional-integral-derivative control of each joint to be used in the response simulation mode of ROBSIM. The control options currently implemented include:

a) Hybrid force/torque control;
b) Active compliance control;
c) Dual arm coordinated control.

The hybrid force/torque control and active compliance control are discussed in the previous ROBSIM report, NASA contractor report 172401. Although dual-arm control was not a requirement of this phase of the contract, initial work in this area was done in support of the ITA program. When analyzing a single manipulator arm, the ROBSIM program solves the dynamic equations of motion for the joint accelerations at each timestep and numerically integrates these accelerations using a fourth-order Runge-Kutta integration algorithm. The dynamic motion equation solved has the form

$$\underline{\tau} = [A(\underline{\theta})]\underline{\ddot{\theta}} + \underline{b}(\underline{\theta},\underline{\dot{\theta}})$$

where

$\underline{\theta}$ $\underline{\dot{\theta}}$ $\underline{\ddot{\theta}}$ are the vectors of joint displacements, velocities, and accelerations

$\underline{\tau}$ is the vector of actuator drive torques and is calculated using a feedback control law

$\underline{b}$ is the vector of position- and velocity-related torques and is solved for using the recursive Newton-Euler equations with joint accelerations set to zero

[A] is the instantaneous effective inertia matrix for the manipulator and is calculated using methods described by Thomas [1982] and Walker [1982].

As mentioned earlier, this equation is solved for $\underline{\ddot{\theta}}$ and a numerical integration algorithm obtains $\underline{\dot{\theta}}$ and $\underline{\theta}$

For this initial implementation of the dual-arm control simulation, an attempt was made to couple the dynamic solution equations and restrict the translational and rotational velocities of the end-effector to be the same.

Sensor control of the end-effector position simulates a video-type sensor mounted on the end effector that can locate a target in the manipulator workspace and then move the arm toward it.

The target is created during the system definition portion of ROBSIM in a manner very similar to creating a load object. The target, however, is drawn as four dots within the system (see Fig 8).

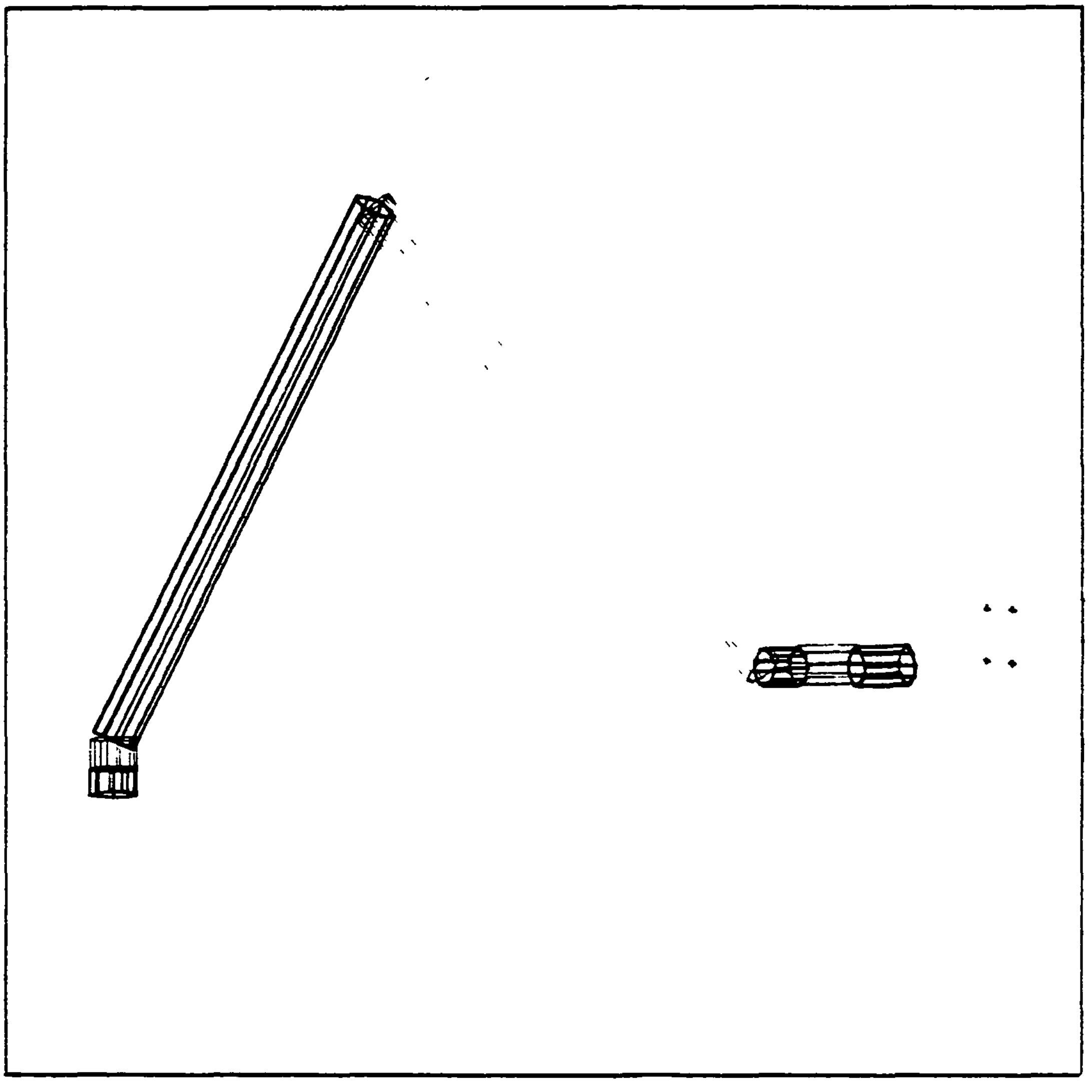


Figure 8. - Manipulator system with target.

The addition of the task command module necessitated a slight modification of the load object creation/modification process. The process has changed minimally from that described in the previous ROBSIM report (NASA contractor report 172401). The same options and parameters still need to be defined by the user with the addition of three more pieces of data. This additional information requested of the user during the object creation process has been added in conjunction with the creation of a set of task level commands.

The additional data needed for each load object are a name for the object, a grasp point on the object, and an approach vector for grasping the load object.

The load name is a unique eight-character alphanumeric string identifying the load object.

The grasp point is the x, y, z coordinates in the load local coordinate system that define where a manipulator end-effector must grasp the object to move or lift it.

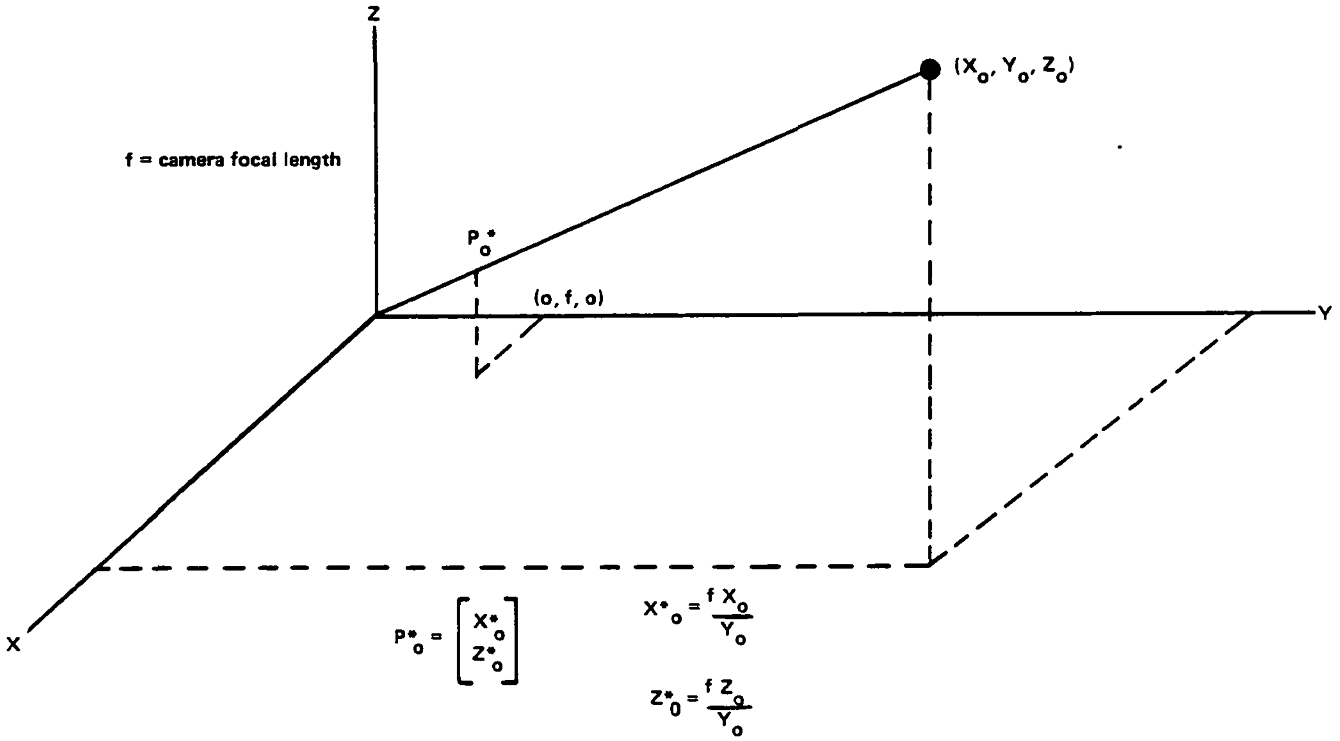
The approach vector is used to define the direction or side of the object that the end effector must be coming from to grasp the object. This vector should be defined in the load local coordinate system and corresponds to the direction of end-effector x-axis.

TASK 24, RANGING SYSTEM SIMULATION

Task 24 added the capability to simulate a manipulator-mounted sensor system that provides accurate relative position and orientation determination between the manipulator end-effector and a specified target. This information is then used to control the motion of the manipulator. The algorithm implemented is based on a paper by Robert M. Haralick [Haralick, 1980]. This algorithm calculates the camera (sensor) pointing angles and distances to a target based on the perspective projection of the four corner dots of the target.

Given an x-y-z coordinate frame, the perspective projection of the point (x_o, y_o, z_o) is defined in Figure 9.

Note that by convention the y-axis is along the line of sight (LOS) of the sensor and the image plane. Where starred coordinates (X^*, Z^*) exist, the LOS is placed one focal length, f, in front of the camera lens, which is at the origin.



$$P^*_o = \begin{bmatrix} X^*_o \\ Z^*_o \end{bmatrix} \qquad X^*_o = \frac{f X_o}{Y_o} \qquad Z^*_0 = \frac{f Z_o}{Y_o}$$

Figure 9. - Definition of the perspective projection.

The camera pointing angles (note that on the camera line of sight points in type positive y-axis direction) are defined as:

Counterclockwise rotation about the z-axis. (theta).

$$-90° < \theta \leq 90°$$

Counterclockwise rotation about the x-axis (phi).

$$-90° < \phi \leq 90°$$

Clockwise rotation about the y-axis (xi).

$$-180° < \xi \leq 180°$$

Given a reference frame to start from, a sequence of pointing angles can arbitrarily orient a sensor (Fig. 10).

A direction cosine matrix (sometimes called the "attitude" matrix) is a 3x3 matrix that provides the mathematical connection between the coordinates of a given point in the reference frame and the coordinates of the same point in a rotated frame.

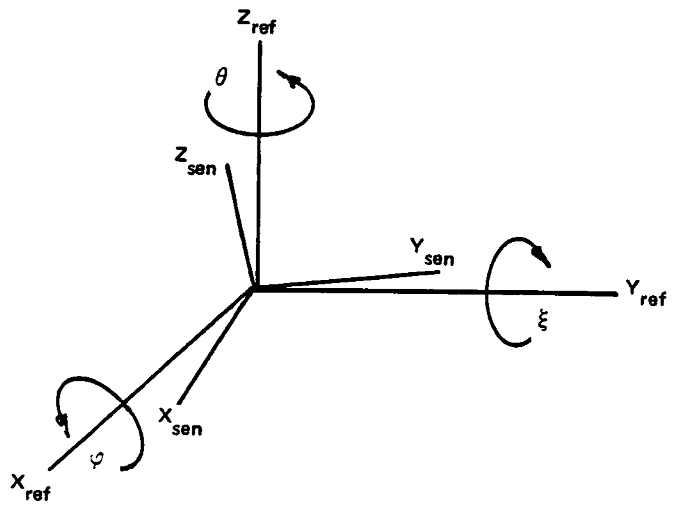


Figure 10. - Sensor coordinate frame rotated relative to the reference coordinate frame.

The entries in the 3x3 matrix [A] depend on the $\theta\ \phi\ \xi$ angles of the particular rotation.

For a counterclockwise rotation θ about the z-axis, the direction cosine matrix is given by

$$[A_z] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

For a counterclockwise rotation ϕ about the x-axis, the direction cosine matrix is given by

$$[A_x] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

For a clockwise rotation ξ about the y-axis, the direction cosine matrix is given by

$$[A_y] = \begin{bmatrix} \cos\xi & 0 & \sin\xi \\ 0 & 1 & 0 \\ -\sin\xi & 0 & \cos\xi \end{bmatrix}$$

Now, given an arbitrary orientation, the direction cosine matrix is obtained by multiplying the sequence of matrices for $\theta\ \phi\ \xi$ with the θ matrix on the right and the ξ matrix on the left. Therefore

$$[A] = [A_y]\,[A_x]\,[A_z]$$

$$= \begin{bmatrix} \cos\xi\cos\theta + \sin\xi\sin\phi\sin\theta & \cos\xi\sin\theta - \sin\xi\sin\phi\cos\theta & \sin\xi\cos\phi \\ -\cos\phi\sin\theta & \cos\phi\cos\theta & \sin\phi \\ -\sin\xi\cos\theta + \cos\xi\sin\phi\sin\theta & -\sin\xi\sin\theta - \cos\xi\sin\phi\cos\theta & \cos\xi\cos\phi \end{bmatrix}$$

To calculate the perspective projection in a rotated sensor frame of a point $\underline{P}$ given reference frame coordinates (x_o, y_o, z_o), we first convert to sensor frame coordinates. Given the sensor pointing angles $\theta\ \phi\ \xi$ we have

$$[A]\underline{P}_{ref} = \underline{P}_{sen} = \begin{bmatrix} (x_o)_{sen} \\ (y_o)_{sen} \\ (z_o)_{sen} \end{bmatrix}$$

Now applying the definition of perspective projection, the perspective projection of $\underline{P}$ in the sensor frame is given by

$$x^* = f\,\frac{(x_0)\,\text{sen}}{(y_0)\,\text{sen}} \qquad z^* = f\,\frac{(z_0)\,\text{sen}}{(y_0)\,\text{sen}}$$

and ,substituting, we get

$$x^* = f\,\frac{x_0(\cos\xi\cos\theta + \sin\xi\sin\phi\sin\theta) + y_0(\cos\xi\sin\theta - \sin\xi\sin\phi\cos\theta) + z_0(\sin\xi\cos\phi)}{x_0(-\cos\phi\sin\theta) + y_0(\cos\phi\cos\theta) + z_0(\sin\phi)}$$

$$z^* = f\,\frac{x_0(-\sin\xi\cos\theta + \cos\xi\sin\phi\sin\theta) + y_0(-\sin\xi\sin\theta - \cos\xi\sin\phi\cos\theta) + z_0(\cos\xi\cos\phi)}{x_0(-\cos\phi\sin\theta) + y_0(\cos\phi\cos\theta) + z_0(\sin\phi)}$$

Furthermore, these equations can be neatly expressed as

$$\begin{bmatrix} x^* \\ z^* \end{bmatrix} = \begin{bmatrix} \cos\xi & \sin\xi \\ -\sin\xi & \cos\xi \end{bmatrix} \begin{bmatrix} x' \\ z' \end{bmatrix}$$

$$x' = f\,\frac{x_0\cos\theta + y_0\sin\theta}{x_0(-\cos\phi\sin\theta) + y_0(\cos\phi\cos\theta) + z_0(\sin\phi)}$$

$$z' = f\,\frac{x_0\sin\phi\sin\theta - y_0\sin\phi\cos\theta + z_0\cos\phi}{x_0(-\cos\phi\sin\theta) + y_0(\cos\phi\cos\theta) + z_0(\sin\phi)}$$

This result is obtained by factoring out cos ξ and sin ξ from the first equation for x^* and z^*. The advantage to these expressions is that the dependence on ξ has been neatly separated from θ and ϕ.

Now we attempt to go in the reverse direction. That is, assuming knowledge of the perspective projection, what can one say about the possible points in three-dimensional space that produced it? The set of points in three-dimensional space that could produce the perspective projection (x^*, z^*) are given by the ray of reference frame coordinate defined by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda \begin{bmatrix} x'\cos\theta - f\sin\theta\cos\phi + z'\sin\theta\sin\phi \\ x'\sin\theta + f\cos\theta\cos\phi - z'\cos\theta\sin\phi \\ f\sin\phi + z'\cos\phi \end{bmatrix}$$

for some λ and where

$$\begin{bmatrix} x' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\xi & -\sin\xi \\ \sin\xi & \cos\xi \end{bmatrix} \begin{bmatrix} x^* \\ z^* \end{bmatrix}$$

This result comes from the observation that any point on the ray has coordinates (x^*, f, z^*) in the sensor frame. To express this in the reference frame we multiply by $[A]^{-1}$

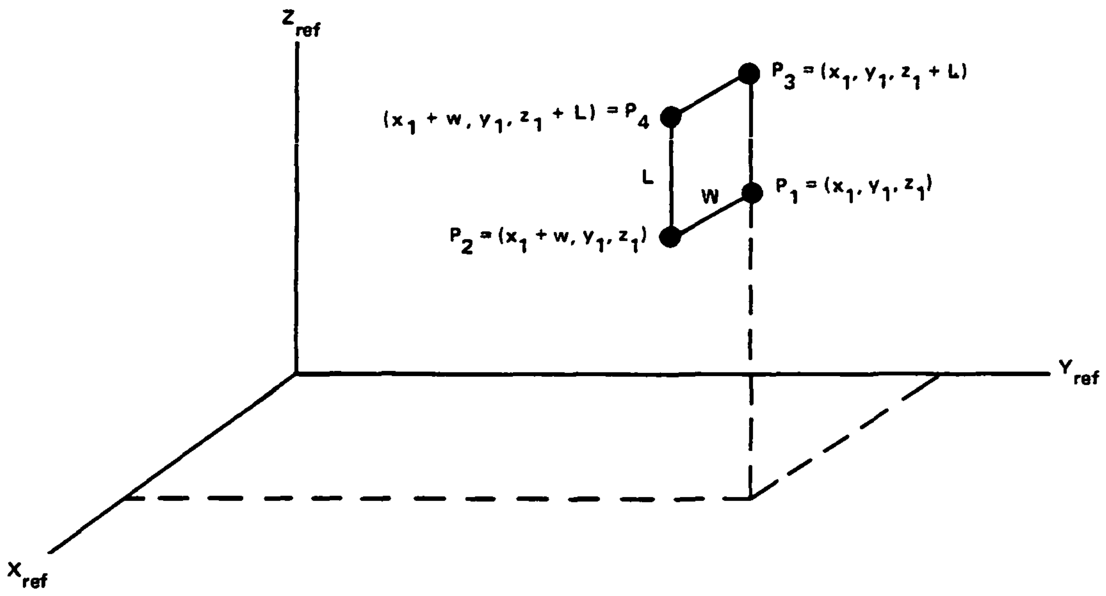
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda \; [\,A\,]^{-1} \begin{bmatrix} x^* \\ f \\ z^* \end{bmatrix}$$

$$= \lambda \begin{bmatrix} \cos\theta & -\sin\theta\cos\phi & \sin\theta\sin\phi \\ \sin\theta & \cos\theta\cos\phi & -\sin\phi\cos\theta \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x^*\cos\xi - z^*\sin\xi \\ f \\ x^*\sin\xi + z^*\cos\xi \end{bmatrix}$$

Note that $[A]^{-1}$ is obtained by doing the negative of the original rotation angles in the reverse sequence.

The perspective projection of a target can then be accomplished by finding the projection of each of the four corners.

Similarly, the reverse calculations may also be carried out (i.e., given the perspective projections of the corners P_1^* P_2^* P_3^* P_4^*, find the pointing angles θ ϕ and ξ) (see Fig. 11 and 12).



Z_{ref}
$P_3 = (x_1, y_1, z_1 + L)$
$(x_1 + w, y_1, z_1 + L) = P_4$
L
W
$P_1 = (x_1, y_1, z_1)$
$P_2 = (x_1 + w, y_1, z_1)$
Y_{ref}
X_{ref}

Figure 11. -
The coordinates of the corners of the target rectangle in the reference frame parallel to the rectangle sides.

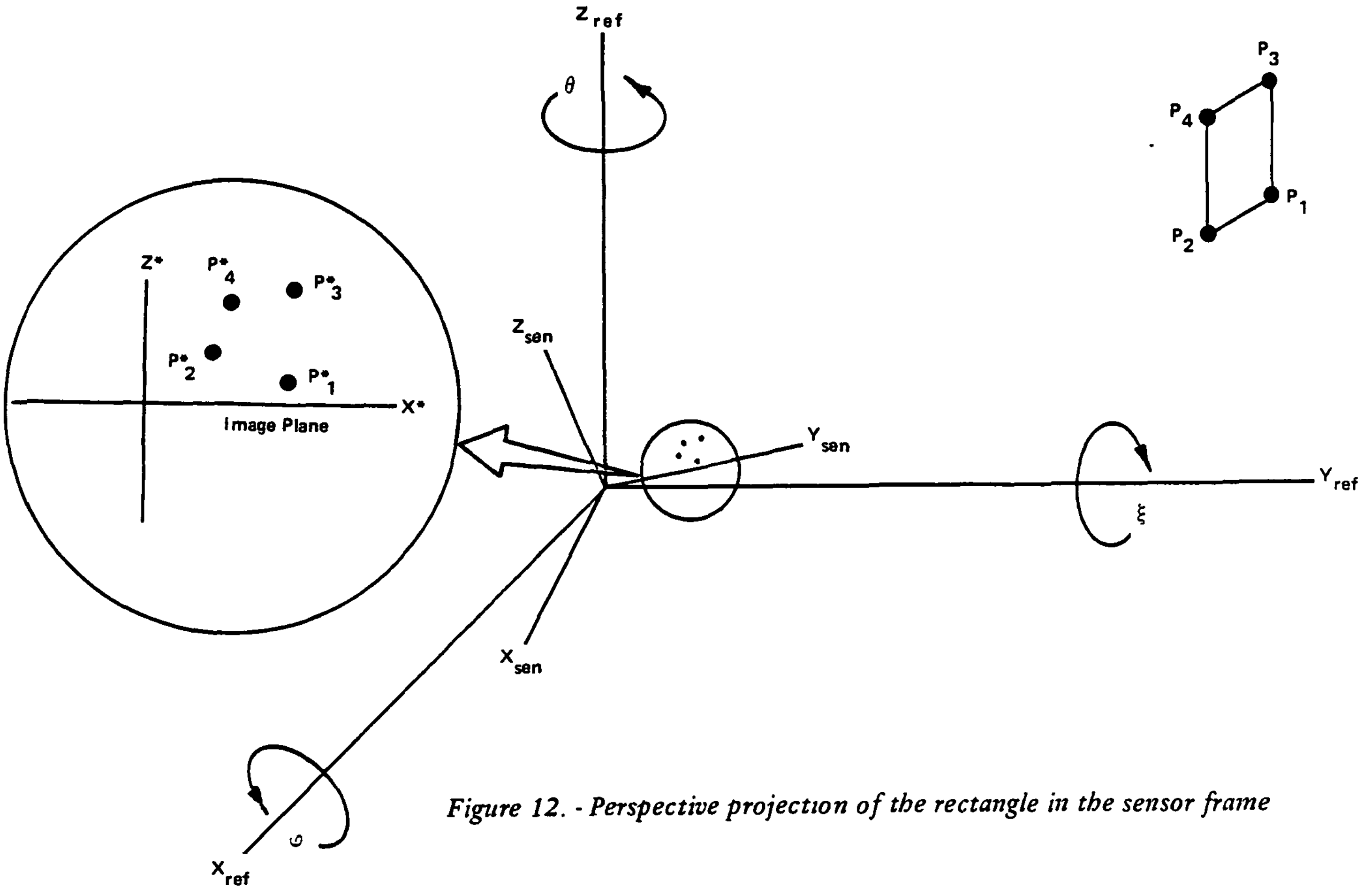


Figure 12. - Perspective projection of the rectangle in the sensor frame

Establish the notation

$$P_1^* = \begin{bmatrix} x_1^* \\ z_1^* \end{bmatrix} \quad P_2^* = \begin{bmatrix} x_2^* \\ z_2^* \end{bmatrix} \quad P_3^* = \begin{bmatrix} x_3^* \\ z_3^* \end{bmatrix} \quad P_4^* = \begin{bmatrix} x_4^* \\ z_4^* \end{bmatrix}$$

$$P_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad P_2 = \begin{bmatrix} x_2 = x_1 + w \\ y_2 = y_1 \\ z_2 = z_1 \end{bmatrix} \quad P_3 = \begin{bmatrix} x_3 = x_1 \\ y_3 = y_1 \\ z_3 \quad z_1 + L \end{bmatrix} \quad P_4 = \begin{bmatrix} x_4 = x_1 + w \\ y_4 = y_1 \\ z_4 \quad z_1 + L \end{bmatrix}$$

We can apply the results of the previous section to each of the corners. For K=1, 2, 3, 4

$$P_K^* = \begin{bmatrix} x_K^* \\ z_K^* \end{bmatrix} = \begin{bmatrix} \cos\xi & \sin\xi \\ -\sin\xi & \cos\xi \end{bmatrix} \begin{bmatrix} x_K' \\ z_K' \end{bmatrix}$$

where

$$x_K' = f \frac{x_K \cos\theta + y_K \sin\theta}{x_K(-\cos\phi\sin\theta) + y_K(\cos\phi\cos\theta) + z_K(\sin\phi)}$$

$$z_K' = f \frac{x_K \sin\phi\sin\theta - y_K \sin\phi\cos\theta + z_K \cos\phi}{x_K(-\cos\phi\sin\theta) + y_K(\cos\phi\cos\theta) + z_K(\sin\phi)}$$

and going in the reverse direction we know that there are four constants $\lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_4$ such that if

$$\begin{bmatrix} x_K' \\ z_K' \end{bmatrix} = \begin{bmatrix} \cos\xi & -\sin\xi \\ \sin\xi & \cos\xi \end{bmatrix} \begin{bmatrix} x_K^* \\ z_K^* \end{bmatrix}$$

and

$$\begin{bmatrix} x_K \\ y_K \\ z_K \end{bmatrix} = \lambda_K \begin{bmatrix} x_K' \cos\theta - f\sin\theta\cos\phi + z_K' \sin\theta\sin\phi \\ x_K' \sin\theta + f\cos\theta\cos\phi - z_K' \cos\theta\sin\phi \\ f\sin\phi + z_K' \cos\phi \end{bmatrix}$$

then (x, y, z) are the reference coordinates of a point on the ray emanating from the origin through the point P_K^* . This gives us a starting point to go from the perspective projections P_1^* P_2^* P_3^* P_4^* to the sensor pointing angles θ ϕ ξ . Now we use the fact that these perspective projections are of a rectangle.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \lambda_1 \begin{bmatrix} x_1'\cos\theta - f\sin\theta\cos\phi + z_1'\sin\theta\sin\phi \\ x_1'\sin\theta + f\cos\theta\cos\phi - z_1'\cos\theta\sin\phi \\ f\sin\phi + z_1'\cos\phi \end{bmatrix} \qquad [1]$$

$$\begin{bmatrix} x_1 + w \\ y_1 \\ z_1 \end{bmatrix} = \lambda_2 \begin{bmatrix} x_2'\cos\theta - f\sin\theta\cos\phi + z_2'\sin\theta\sin\phi \\ x_2'\sin\theta + f\cos\theta\cos\phi - z_2'\cos\theta\sin\phi \\ f\sin\phi + z_2'\cos\phi \end{bmatrix} \qquad [2]$$

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 + L \end{bmatrix} = \lambda_3 \begin{bmatrix} x_3'\cos\theta - f\sin\theta\cos\phi + z_3'\sin\theta\sin\phi \\ x_3'\sin\theta + f\cos\theta\cos\phi - z_3'\cos\theta\sin\phi \\ f\sin\phi + z_3'\cos\phi \end{bmatrix} \qquad [3]$$

$$\begin{bmatrix} x_1 + w \\ y_1 \\ z_1 + L \end{bmatrix} = \lambda_4 \begin{bmatrix} x_4'\cos\theta - f\sin\theta\cos\phi + z_4'\sin\theta\sin\phi \\ x_4'\sin\theta + f\cos\theta\cos\phi - z_4'\cos\theta\sin\phi \\ f\sin\phi + z_4'\cos\phi \end{bmatrix} \qquad [4]$$

These equations give enough information to solve for θ ϕ ξ without knowing L, W, x_1, y_1, z_1, λ_1, λ_2, λ_3, λ_4. To begin, notice that the first and second coordinates of equations [1] and [3] are the same (x_1 and y_1)

$$\lambda_1(x_1'\cos\theta - f\sin\theta\cos\phi + z_1'\sin\theta\sin\phi)$$
$$= \lambda_3(x_3'\cos\theta - f\sin\theta\cos\phi + z_3'\sin\theta\sin\phi)$$

$$\lambda_1(x_1'\sin\theta + f\cos\theta\cos\phi - z_1'\cos\theta\sin\phi)$$
$$= \lambda_3(x_3'\sin\theta + f\cos\theta\cos\phi - z_3'\cos\theta\sin\phi)$$

Cross multiplying, canceling $\lambda_1 \cdot \lambda_3$ and combining like terms

$$(x_1' - x_3')f\cos\phi + (x_3'z_1' - x_1'z_3')\sin\phi = 0$$

$$\tan\phi = f\frac{(x_3' - x_1')}{(x_3'z_1' - x_1'z_3')}$$

Now, notice that the first and second coordinates of equations [2] and [4] are the same. Apply the same steps as for [1] and [3] and get

$$(x_2' - x_4')f\cos\phi + (x_4'z_2' - x_2'z_4')\sin\phi = 0$$

This gives an alternate expression for tan ϕ

$$\tan\phi = f\frac{(x_4' - x_2')}{(x_4'z_2' - x_2'z_4')}$$

Thus ϕ can be calculated two ways

$$\phi = \tan^{-1}\left[\frac{f(x_3' - x_1')}{x_3'z_1' - x_1'z_3'}\right] = \tan^{-1}\left[\frac{f(x_4' - x_2')}{(x_4'z_2' - x_2'z_4')}\right] \qquad [5]$$

This gives us a way to calculate ϕ once the primed coordinates are known. Now to solve for θ, notice that the second and third coordinates of [1]and [2] are the same (y_1 and z_1)

$$\lambda_1(x_1'\sin\theta + f\cos\theta\cos\phi - z_1'\cos\theta\sin\phi)$$

$$= \lambda_2(x_2'\sin\theta + f\cos\theta\cos\phi - z_2'\cos\theta\sin\phi)$$

$$\lambda_1(f\sin\phi + z_1'\cos\phi) = \lambda_2(f\sin\phi + z_2'\cos\phi)$$

Cross multiplying, cancel $\lambda_1 \cdot \lambda_2$ and combine like terms

$$(x_1' - x_2')f\sin\theta\sin\phi + (x_1'z_2' - x_2'z_1')\sin\theta\cos\phi + (z_2' - z_1')f\cos\theta = 0$$

and by symmetry (the second and third coordinates of [3] and [4] are the same: y_1 and $z_1 + L$)

$$(x_3' - x_4')f\sin\theta\sin\phi + (x_3'z_4' - x_4'z_3')\sin\theta\cos\phi + (z_4' - z_3')f\cos\theta = 0$$

Take these two equations and eliminate the first term by multiplying the first one by $(x_3' - x_4')$ and the second one by $(x_1' - x_2')$ and then subtracting

$$[(x_3' - x_4')(x_1'z_2' - x_2'z_1') - (x_1' - x_2')(x_3'z_4' - x_4'z_3')]\sin\theta\cos\phi$$

$$+ [(x_3' - x_4')(z_2' - z_1') - (x_1' - x_2')(z_4' - z_3')] f\cos\theta = 0$$

$$\tan\theta = f\frac{(x_1' - x_2')(z_4' - x_3') - (x_3' - x_4')(z_2' - z_1')}{[(x_3' - x_4')(x_1'z_2' - x_2'z_1') - (x_1' - x_2')(x_3'z_4' - x_4'z_3')]\cos\phi}$$

This gives a way to calculate θ once ϕ and the primed coordinates are known.

Now lets go back to equation [5]. Because these are both equal to $\tan\phi$

$$\frac{(x_1' - x_3')f}{x_1'z_3' - x_3'z_1'} = \frac{(x_2' - x_4')f}{x_2'z_4' - x_4'z_2'}$$

From this and the definition of the primed coordinates we have

$$\frac{(x_1^*\cos\xi - z_1^*\sin\xi) - (x_3^*\cos\xi - z_3^*\sin\xi)}{x_1^* z_3^* - x_3^*z_1^*} = \frac{(x_2^*\cos\xi - z_2^*\sin\xi) - (x_4^*\cos\xi - z_4^*\sin\xi)}{x_2^* z_4^* - x_4^* z_2^*}$$

Note that the denominators result from the rotationally invariant

$$x_1^*z_3^* - x_3^*z_1^* = x_1'z_3' - x_3'z_1' \qquad \text{etc.}$$

$$\frac{(x_1^* - x_3^*)\cos\xi + (z_3^* - z_1^*)\sin\xi}{x_1^*z_3^* - x_3^*z_1^*} = \frac{(x_2^* - x_4^*)\cos\xi + (z_4^* - z_2^*)\sin\xi}{x_2^*z_4^* - x_4^*z_2^*}$$

Divide both sides by cos ξ and solve for tan ξ

$$\tan\xi = \frac{(x_1^* - x_3^*)(x_2^*z_4^* - x_4^*z_2^*) - (x_2^* - x_4^*)(x_1^*z_3^* - x_3^*z_1^*)}{(z_4^* - z_2^*)(x_1^*z_3^* - x_3^*z_1^*) - (z_3^* - z_1^*)(x_2^*z_4^* - x_4^*z_2^*)}$$

Now we have a complete procedure for starting with $P_1^* \; P_2^* \; P_3^* \; P_4^*$ and arriving at $\theta \; \phi \; \xi$.

CONCLUDING REMARKS

This report has documented the work done in Phase IV of contract NAS7-16759 Evaluation of Automated Decisionmaking Methodologies and Development of an Integrated Robotic System Simulation. The tasks (21 through 24) added for this phase extend the capabilities and versatility of the ROBSIM program as follows:

1) Access to CAD/CAM data - This software module addressed task 21 of the contract. It allows the user to access and display CAD/CAM generated data that is written to a file in accordance with the Initial Graphics Exchange Specification (IGES) version 2.0. The user may also use this data for the detailed graphics modeling of system components.

2) Multiple arms with movable bases - This work addressed task 22 of the contract. The user may now model and do analyses on systems that include up to five manipulator arms. In addition, the arm bases may have motion specified for them. Each arm amy be mounted on an independent movable base, or a single base may be attached to more than one arm.

3) Task command module - This addresses task 23 of the contract and was added to make the procedure for motion trajectory specification of a manipulator easier. Each of the task commands is a combination of one or more of the lower level motion specification commands currently in place.

4) Ranging system simulation - This addresses task 24 of the contract. This gives the user another way of controlling the robot motion. The software developed simulates a video camera mounted on the end-effector of a manipulator. The camera looks for a user specified target and moves the end-effector towards it.

The ROBSIM program is a useful tool for the analysis and design of manipulator system components. There are, however, areas for future enhancement than would make the program even more versatile and useful. The areas for expansion could be directed toward the support of teleoperator systems and other robotic systems capable of remote space operations such as satellite servicing or space station construction.

The addition of hand controllers to drive the simulation would have great utility in the area of telerobotics and teleoperator control. The system would be useful for manipulator operator training and for use as a predictive display in servicing operations having large time delays.

Interfacing the analysis software to multiple graphics systems to try and make it somewhat device independent would increase interest in using the program. Other interfaces of interest would be to existing software packages in various areas like controls analysis, finite element analysis, and flexibility.

Work in the area of trajectory planning should be continued. Also, the hardware/software validation efforts should be carried on to better define the strong points of the simulation and the areas requiring more work. In order to validate the software running on different systems, a library of test cases should be built and documented. Another useful library would be one containing validated models of several popular manipulators.

Lastly, the developments needed to implement real-time control could be investigated.

REFERENCES

Haley, Almand, Thomas, Krauze, Gremban, Sanborn, Kelly, and Depkovich: Evaluation of Automated Decisionmaking Methodologies and Development of An Integrated Robotic System Simulation. NASA Contractor Report 172401, September 1984.

R. M. Haralick: Determining Camera Parameters from the Perspective Projection of a Rectangle. Manuscript, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, June 1980.

M. Thomas and D. Tesar: "Dynamic Modeling of Serial Manipulator Arms." ASME Transactions, Journal of Dynamic Systems, Measurement, and Control, Vol 104, 1982, pp 218-228.

M. W. Walker and D. E. Orin: "Efficient Dynamic Computer Simulation of Robotic Mechanisms." ASME Transactions, Journal of Dynamic Systems, Measurement, and Control, Vol 104, 1982, pp 205-211.

Standard Bibliographic Page

1 Report No. NASA CR-178050	2 Government Accession No	3. Recipient's Catalog No
4 Title and Subtitle Evaluation of Automated Decisionmaking Methodologies and Development of an Integrated Robotic System Simulation		5 Report Date March 1986
		6 Performing Organization Code
7 Author(s) D. C. Haley, B. J. Almand, M. M. Thomas, L. D. Krauze, K. D. Gremban, J. C. Sanborn, J. H. Kelley, T. M. Depkovich, W. J. Wolfe, and T. Nguyen		8 Performing Organization Report No MCR-85-665
9 Performing Organization Name and Address Martin Marietta Aerospace Denver Aerospace P. O. Box 179 Denver, CO 80201		10 Work Unit No
		11. Contract or Grant No. NAS1-16759
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		13. Type of Report and Period Covered Contractor Report
		14 Sponsoring Agency Code 506-45-21-01

15. Supplementary Notes

This report consists of three volumes: Study Results, NASA CR-178050; Appendix A - User's Guide, NASA CR-178051; and Appendix B - Programmer's Guide, NASA CR-178052. Supersedes NASA CR-172401, NASA CR-172402, and NASA CR-172403.

16. Abstract

The report describes the implementation of a generic computer simulation for manipulator systems (ROBSIM). The program is written in FORTRAN, and allows the user to:

1. Interactively define a manipulator system consisting of multiple arms, load objects, targets, and an environment.
2. Request graphic display or replay of manipulator motion.
3. Investigate and simulate various control methods including manual force/torque and active compliance control.
4. Perform kinematic analysis, requirements analysis, and response simulation of manipulator motion.

Previous reports (NASA CR 172401-172403) have described the algorithms and procedures for using ROBSIM. This report supersedes these reports and describes additional features which have been added:

1. The ability to define motion profiles and compute loads on a common base to which manipulator arms are attached.
2. Capability to accept data describing manipulator geometry from a Computer Aided Design (CAD) data base using the Initial Graphics Exchange Specification (IGES) format.
3. A manipulator control algorithm derived from processing the TV image of known reference points on a target.
4. A vocabulary of simple high-level task commands (e.g., move, grasp, approach) which can be used to define task scenarios.

17. Key Words (Suggested by Author(s)) ROBSIM Robotics Simulation		18. Distribution Statement Unclassified - Unlimited Subject Category 63	
19 Security Classif (of this report) Unclassified	20 Security Classif (of this page) Unclassified	21. No of Pages 38	22. Price