ELECTRONICS LABORATORY • SYRACUSE, N.Y.

Final Report

REMOTE MANIPULATOR DYNAMIC SIMULATION

April 1972

E. C. Wild
P. K. Donges
W. A. Garand

Contract No. NAS 9-11948

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

OFFICE OF PRIME RESPONSIBILITY

GENERAL ELECTRIC

Final Report

REMOTE MANIPULATOR DYNAMIC SIMULATION

April 1972

E. C. Wild
P. K. Donges
W. A. Garand

Contract No. NAS 9-11948

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

The implemented simulation generates the real time visual scenes required to perform "man in the loop" investigations of remote manipulator application and design concepts for the space shuttle. The simulated remote manipulator - space shuttle system consists of a manipulator (in which shoulder, elbow, wrist and end effector movements are represented), a space shuttle, a payload and a space station central assembly element. The simulation is now in operation at the Guidance and Control Simulation Laboratory, NASA's Manned Spacecraft Center, Houston, Texas. It is implemented on a computed display system that consists of an XDS Sigma-5 digital computer, the Electronic Scene Generator, an operators station and the associated interface hardware. A block diagram of the system is shown in Figure 1.

The major effort on this contract involved the development of the required mathematical solutions and the real time DYNAMIC COLLISION and VISUAL SCENE PROGRAMS and the integration of these programs to yield an operating simulation. A description of the capabilities of the implemented simulation is presented in Section II, CAPABILITIES AND CHARACTERISTICS OF THE SIMULATION. The work performed during the contract is summarized in Section III. Sections IV and V describe the mathematical models and the programs developed on the contract. Section VI, OPERATIONAL MODIFICATIONS, provides the required information for modifying the physical characteristics of the various components of the simulated manipulator - space shuttle system. Section VII lists some of the recommendations for future expansion and refinement of the simulation; these were a result of a series of informal discussions between NASA and contractor engineers during the period of the contract.

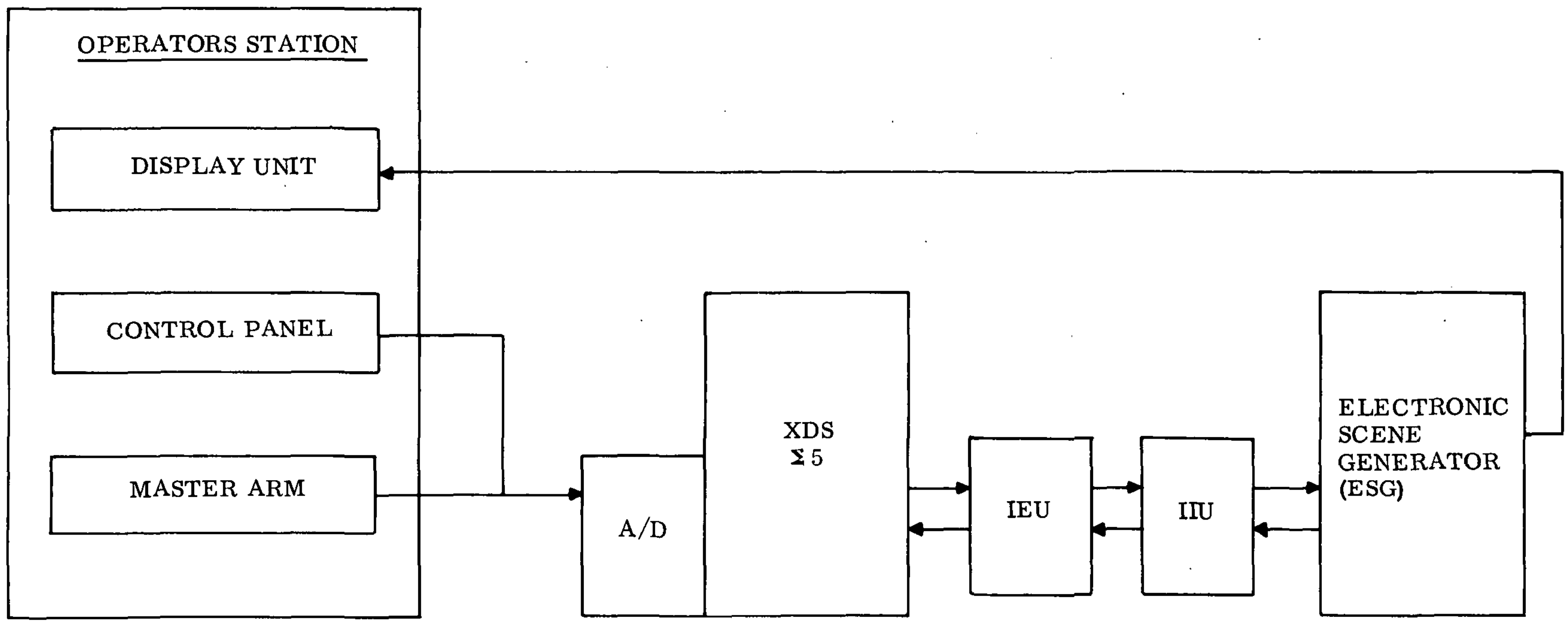


Figure 1. Simulation System: Block Diagram

II. CAPABILITIES AND CHARACTERISTICS OF THE SIMULATION

The operational characteristics are the set of tasks that the operator of the remote manipulator system can perform with the simulation. These are:

1. Maneuver the remote manipulator to acquire a payload that resides in the shuttle cargo bay (Acquisition Phase).

2. Remove the payload from the cargo bay (Extraction Phase).

3. Maneuver the payload and dock it with a central assembly element, CAE (Maneuvering Phase).

The visual-dynamic characteristics, which were developed and implemented, are those that provide the visual cues encountered in performing the above tasks. These visual-dynamic characteristics are provided by the two operating programs developed and implemented on the contract. The VISUAL SCENE PROGRAM generates the information required to depict the dynamic relationships of the remote manipulator system, shuttle, payload and central assembly element during the simulation. The DYNAMIC COLLISION PROGRAM computes the relative spatial location of the physical components of the simulated system and detects collisions, payload acquisitions and docking. It employs this information to control the motion depicted in the visual scene. The visual-dynamic characteristics of the simulation are:

1. Capability to view the scene from any of eight observer (camera) positions during all phases of the simulation.

2. Detection of collision between the manipulator end effector and the payload. Detection of collision between the manipulator end effector and the end effector receptacle.

3. Detection of a successful acquisition of the payload.

4. Detection of collision between the payload and the shuttle cargo bay.

5. Detection of collision between the payload and the central assembly element and detection of a successful docking of the payload.

In addition to controlling the slave arm of the manipulator the operator can:

1. Select in real time one of eight pre-arranged "camera" stations from which to view the environment. He can further control the pan and tilt and can "zoom" the selected camera in and out.

2. Change, in real time, the length of the vertical - fixed arm; i. e., the vertical position of the shoulder joint, with respect to the top deck of the shuttle.

3. Control the opening and closing of a movable finger of the end effector.

4. Reset the simulation to the Acquisition Phase.

Some of the tasks, which a space shuttle remote manipulator system would be used to perform, cannot be simulated on the implemented simulation because deliberate limitations were imposed on the operational characteristics due to the limited funding available for the job. Section VII delineates the effort required to overcome these limitations.

III. SUMMARY OF THE WORK COMPLETED

The major effort on this contract was expended on the development and implementation of mathematical solutions, computing algorithms and the solution of software integration and hardware interface problems. This effort consisted of five major tasks:

1. The derivation of mathematical solutions and algorithms for detecting collisions between the simulated space shuttle - remote manipulator system components.

2. The derivation of dynamics equations and numerical models for the physical and visual simulations.

3. The development of a technique, which uses dynamic separating planes, to solve the visual priority problem.

4. The development, integration and implementation of two real time programs.

5. The design and installation of a hardware modification to the Input Expansion Unit, (IEU). This modification increased the data transmission capability from the XDS Sigma 5 to the Electronic Scene Generator (ESG) from 16 twenty-four bit words to 64 twenty-four bit words per frame.

IV. THE DYNAMIC COLLISION PROGRAM

The operational tasks that can be simulated include acquisition of the payload, extraction of the payload from the shuttle cargo bay and the maneuvering and docking of the payload. To perform these tasks the operator relies on the visual information provided by physical collisions between various components of the remote manipulator system. The development of the simulation required that two types of collisions be considered. The first type of collision is that which is detrimental to performing the desired task and, thus, must be relieved by the operator. The second type are those which are essential to the performance of a given task and must be maintained by the operator. Two examples of detrimental collisions are 1) collision between the end effector and the payload surface and 2) collisions between the payload and the walls of the cargo bay. An example of an essential collision is one that occurs when a finger of the end effector contacts the end effector receptacle in the physical relationship required to achieve acquisition of the payload.

The DYNAMIC COLLISION PROGRAM solves the mathematical and logical decision problems encountered in detecting collision, determining acquisition and the docking of the payload. Its solutions are employed to control the VISUAL SCENE PROGRAM. It also computes three dynamic separating planes employed by the VISUAL SCENE PROGRAM to compute the visual priorities between the manipulator end effector and the end effector receptacle.

A. MATHEMATICS AND ALGORITHMS

The mathematical solutions and the computing algorithms developed are:

1. Collision Detection - Mathematical solution to the problem of detecting when physical components of the manipulator-space shuttle system collide and detecting when a payload acquisition or docking is successfully completed.

2. Separating Plane Computation - Selection of three separating planes and a method of computing the data required to specify their location and orientation to the VISUAL SCENE PROGRAM.

3. Coordinate Systems - Selection of a system of coordinates to mathematically describe the manipulator - payload-shuttle-central assembly element geometry and derivation of the equations required to compute the relative spacial locations of the components.

1. <u>Collision Detection and Separating Plane Computation</u>

The collision detection solution is based on the collision of two hexahedrons. There are three possible ways in which two hexahedrons, Hex A and Hex B, can physically collide.

1. A vertex of Hex A collides with a face of Hex B.

2. A vertex of Hex B collides with a face of Hex A.

3. An edge of Hex A (Hex B) collides with an edge of Hex B (Hex A).

The mathematical tests for collision consist of testing the eight vertices of each hexahedron for collision with the six faces of the other hexahedron and tesing the twelve edges of one hexahedron for collision with the twelve edges of the other.

The theoretical basis for these tests and the algorithms that were developed to implement them are best explained by considering the two hexahedrons, Hex A and Hex B. To determine if a vertex of Hex A collides with a face of Hex B the position of all eight vertices of Hex A with respect to the six faces of Hex B is found. For each vertex a six bit word, the vertex location word, is formed. The location word for vertex i has the form, $E_i = (a, b, c, d, e, f)$, where a 1 in bit position x indicates that vertex i is on the TRUE side of face x and a zero in bit position x indicates that vertex i is on the FALSE side of face x. A standard nomenclature for labeling the faces and vertices of a hexahedron is used throughout this report. This relationship between faces and vertices is shown in Figure 2. A location word E_i is formed for all eight vertices of Hex A. i.e., location words $E_1 \ldots E_8$. A vertex of Hex A collides with a face of Hex B if the vertex is inside the volume defined by the six faces of Hex B. By definition this situation exists if and only if the vertex is on the FALSE side of all six faces of Hex B. Thus, a location word, E_i, consisting of six zeros indicates that vertex i has collided. Conversely, if a location word contains at least one "one", a collision does not exist. Furthermore the information required to determine whether a face of Hex B can be used to visually separate the two hexahedrons is available from the eight location words formed for Hex A. If all eight vertices of Hex A lie on the TRUE side of the same face of Hex B, a plane co-planar to that face physically separates, and thus visually separates, the two hexahedrons. Thus, if all eight location words have a 1 in bit position x, face x of Hex B can be used as a separating plane by the VISUAL SCENE PROGRAM.

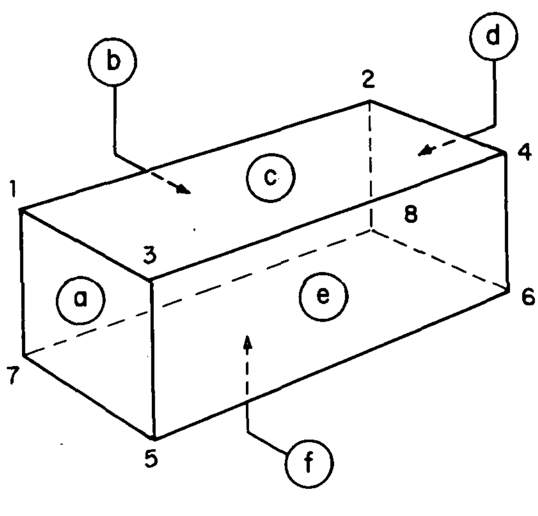


Figure 2. Hexahedron Face-Vertex Standard Notation

The collision of a vertex of Hex B with a face of Hex A is determined by the same algorithm with the roles of the two hexahedrons reversed. If, during these two vertex-face collision tests, a collision is detected or a separating plane is found the test is terminated. If, however, both vertex-face collision tests yield neither a collision nor a separating plane a potential edge-to-edge collision may exist.

The conditions for a potential edge-to-edge collision can be determined by examining the eight vertex location words, $E_1 \ldots E_8$. In Figure 3 the two hexahedrons have a potential edge-to-edge-collision. Edge $v_i - v_{i+1}$ of Hex A potentially collides with edge $b_j - b_{j+1}$ of Hex B, the edge defined by faces c and e of Hex B. A potential collision of these two edges is revealed by the vertex location words for vertices v_i and v_i+1. A potential collision of edge $v_i - v_i+1$ with edge $b_i - b_i+1$ exists if v_i is on the TRUE side of face c and the FALSE side of face e while v_{i+1} is simultaneously on the TRUE side of face e and the FALSE side of face c. Logically this condition is revealed by "ones" in the bit positions corresponding to faces c and e of the modulo two sum of the vertex location words for v_i and v_{i+1}. Thus, to determine if a potential edge-to-edge collision exists, the modulo two sum of the vertex location words, which define the twelve edges of Hex A, are examined. If the modulo two sum for an edge of Hex A contains two or more "ones", that edge is further examined to determine which edges of Hex B it potentially collides with. The potential collision edges of Hex B are found from the vertex locations words for v_i and v_{i+1}. The vertex location word for v_i contains ones in the bit positions corresponding to the faces of Hex B that v_i is on the TRUE side of. The edges bounding these faces are those that must be tested for collision. The maximum number of Hex B edges that must be tested is six, which occurs when v_i is on the TRUE side of either two or three Hex B faces. The minimum number to be tested is four, which is the situation when v_i is on the TRUE side of only one Hex B face. A list of bounding edges for each possible combination of faces, which a vertex may find itself on the TRUE side of, is compiled. This list is used to set up a sequential edge-to-edge collision test. The test consists of forming the successive scalar triple products of a vector representing the Hex A edge $v_i - v_{i+1}$ and vectors specified by the vertices of the bounding edges of the Hex B faces and the vertex v_i. If all of these scalar triple products are less than zero the edge, $v_i - v_{i+1}$, pierces one of the faces that v_i is on the TRUE side of and, thus, edge $v_i - v_{i+1}$ has collided. If, in the process of computing the scalar triple products, one of them is greater than or equal to zero the edge, $v_i - v_{i+1}$, cannot collide and the plane defined by the two vectors from v_i to the vertices of Hex B can be used as a visual separating plane. Figure 4 illustrates the geometry of the scalar triple product formation for the case where v_i is on the true side of face c of Hex B. Using the nomenclature defined in this figure the edge-to-edge collision test is expressed mathematically as follows: The vector defining the edge, $v_i - v_{i+1}$, is:

$$V_i = v_{i+1} - v_i$$

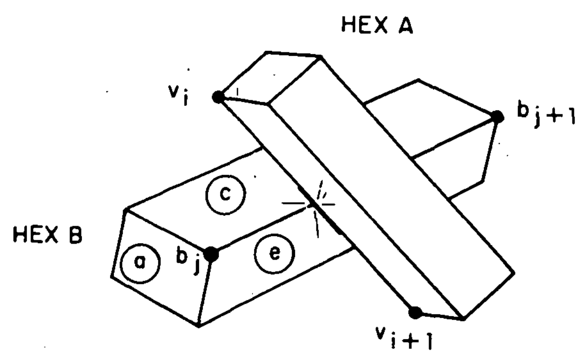


Figure 3. Potential Edge-to-edge Collision

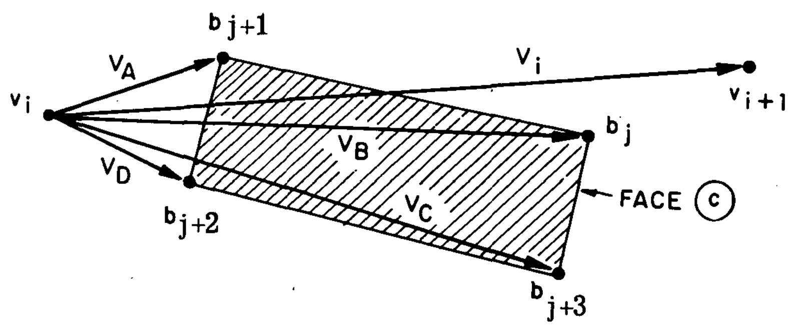


Figure 4. Edge-to-edge Scalar Triple Product Test

The vectors between v_i and the vertices defining the bounding edges of face c:

$$V_A = b_{j+1} - v_i$$

$$V_B = b_j - v_i$$

$$V_C = b_{j+3} - v_i$$

$$V_D = b_{j+2} - v_i$$

Collision of edge v_i - v_{i+1} with Hex B occurs if and only if the following scalar triple products equations are satisfied:

$$V_B \times V_A \cdot V_i \leq 0$$

$$V_C \times V_B \cdot V_i \leq 0$$

$$V_D \times V_C \cdot V_i \leq 0$$

$$V_A \times V_D \cdot V_i \leq 0$$

If any of the scalar triple products are greater than zero a collision does not exist and a visual separating plane can be specified for Hex A and Hex B. For example, if the scalar triple product, $V_C \times V_B \bullet V_i$, was greater than zero the plane defined by the points, v_i, b_j and b_{j+3}, physically and visually separates the two hexahedrons.

The collision and separating plane computing algorithms described above are the basic building block of the DYNAMIC COLLISION PROGRAM.

2. Coordinate Systems and Model Computation

To perform the computations for detecting collisions and obtaining separating planes the vertices defining the mathematical models of the various system components must be in the same coordinate system. Figure 5 defines the coordinate systems required to specify the dynamics of the remote manipulator system.

During the Acquisition Phase (Mode 1) the encounter of the end effector and the end effector receptacle of the payload are of concern. The payload and its end effector receptacle are specified in the Reference Coordinate system (X_R, Y_R, Z_R), while the end effector is specified in System Two coordinates (x_2, y_2, z_2). The vertices of the three hexahedrons (EFF HEX 1, EFF HEX 2 and EFF HEX 3), which form the end effector model, are transformed into Reference Coordinates. The method employed is to obtain the reference coordinates of the three System-Two unit vectors; UX2, UY2 and UZ2, using the sequence of rotations defined in the Appendix. The rotation of a single unit vector, for example UX2, produces three Reference System vectors, UX2XR, UX2YR and UX2ZR. The Reference Coordinates for the vertices of the hexahedrons employed to model the end effector are found by multiplying the Reference System vectors of the appropriate System-Two unit vectors by dimensional constants. To illustrate this process of model

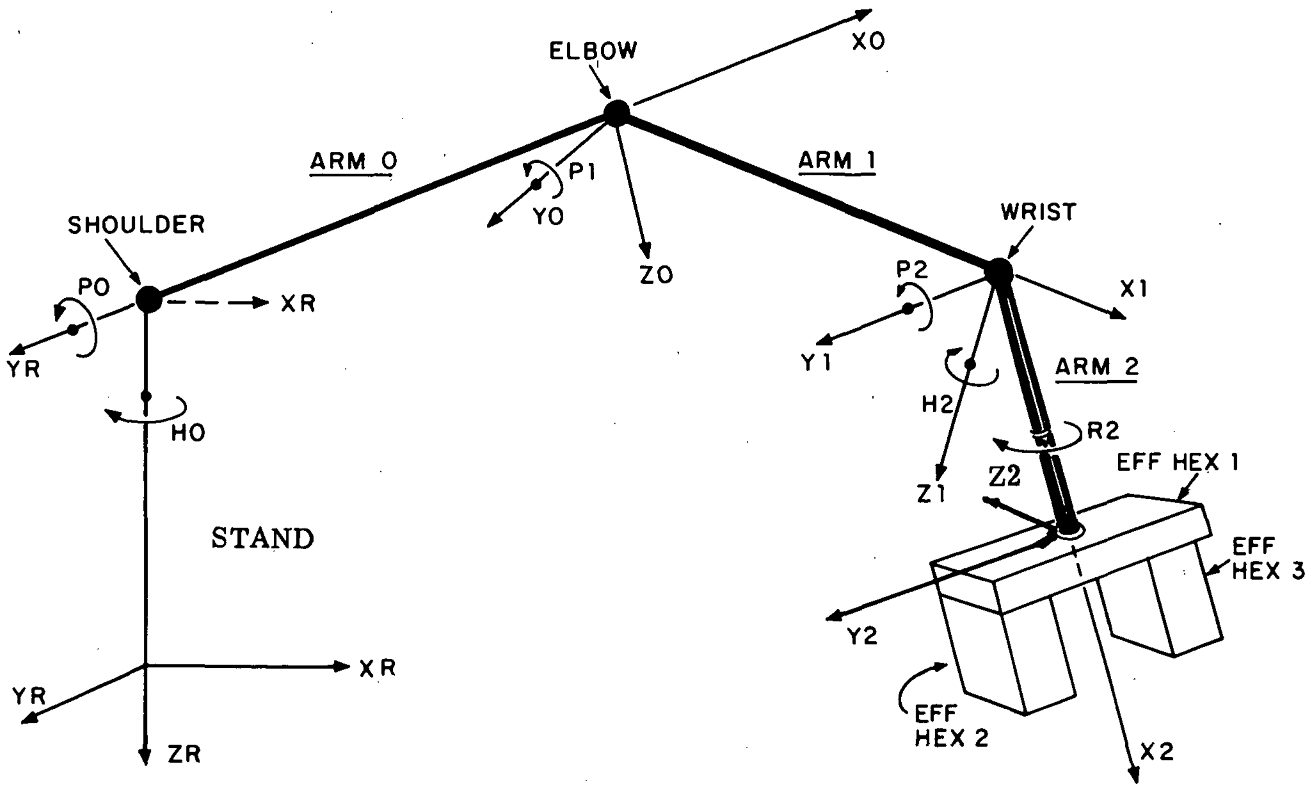


Figure 5. Manipulator and Coordinate Systems

generation in Reference Coordinates consider the first four vertices of EFF HEX 1, as shown in Figure 6. Assume that the reference system coordinates of a point in the center of the top of the hexahedron are known (XRA, YRA, ZRA) and that the Reference System Coordinates for vertices v1, v2, v3 and v4 of the hexahedrons are to be calculated. The physical dimensions of the hexahedron are specified by length (L) in the UX2 direction, width (W) in the UY2 direction, and height (H) in the UZ2 direction.

The reference system coordinates of v1 are:

v1XR = XRA -W(UY2XR) - H (UZ2XR)
v1YR = YRA -W(UY2YR) - H(UZ2YR)
v1ZR = ZRA -W(UY2ZR) - H(UZ2ZR)

The reference system coordinates of v2 are:

v2XR = XRA - W(UY2XR) + H(UZ2XR)
v2YR = YRA - W(UY2YR) + H(UZ2YR)
v2ZR = ZRA - W(UY2ZR) + H(UZ2ZR)

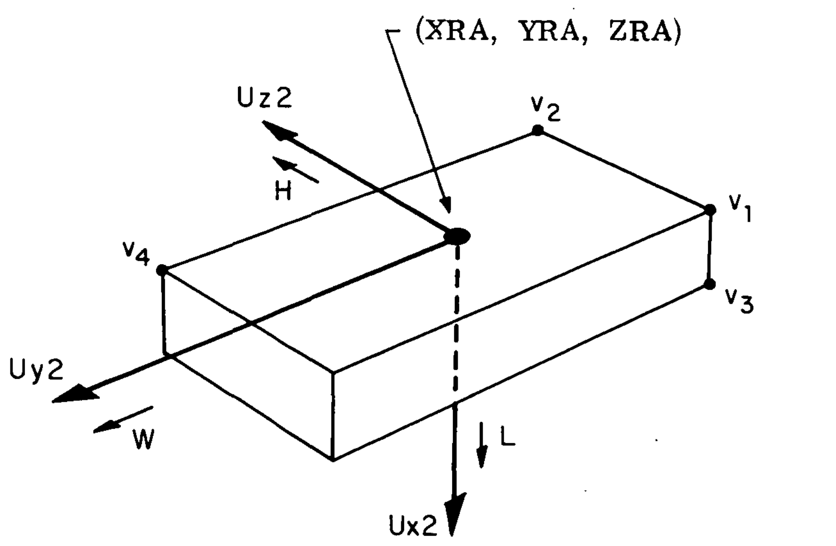


Figure 6. EFF HEX 1: System Two Coordinates and Dimensions

The reference system coordinates of v3 are:

v3XR = XRA - W(UY2XR) - H(UZ2XR) + L(UX2XR)
v3YR = YRA - W(UY2YR) - H(UZ2YR) + L(UX2YR)
v3ZR = ZRA - W(UY2ZR) - H(UZ2ZR) + L(UX2ZR)

The reference system coordinates of v4 are:

v4XR = XRA + W(UY2XR) + H(UZ2XR)
v4YR = YRA + W(UY2YR) + H(UZ2YR)
v4ZR = ZRA + W(UY2ZR) + H(UZ2ZR)

The vertices are calculated in this manner, rather than first specifying each vertex in System Two coordinates and transforming each to Reference Coordinates by the sequence of rotations, to save computing time. With this approach only three unit vectors must be transformed and advantage can be taken of the symmetry of hexahedron geometry. This latter advantage is illustrated by the equations for v_1 and v_4 above. The quantity W(UY2XR) + H(UZ2XR) appears in both equations; only the sign differs.

In the Extraction Phase (Mode 2) and the Maneuvering Phase (Mode 3) the calculation of the vertices of the payload model in Reference Coordinates is performed because the collision of the payload with the cargo bay walls of the space shuttle and with the central assembly element are of interest. The procedure is essentially the same. A subroutine, which uses the diameter of the payload, length of the payload and the diameter of the payload docking probe as input data, computes the reference system coordinates of the payload vertices.

3. Acquisition, Extraction and Docking

The Acquisition Phase (Mode 1) concerns itself with the operational task of manipulating the end effector to acquire the payload residing in the space shuttle cargo bay. The physical components of the remote manipulator system, which enter into the DYNAMIC COLLISION PROGRAM during the Acquisition Phase, are the three hexahedrons that comprise the end effector model (EFF HEX 1, EFF HEX 2, EFF HEX 3) and the single hexahedron (RECPT) that is the end effector bar receptacle. Acquisition of the payload requires that the end effector be positioned over the receptacle with the moving finger of the end effector being brought to a "fully closed" position. Figure 7 shows the end effector and receptacle models that were implemented. Acquisition of the payload occurs when the following relative conditions between the models.

- All BAR vertices are located on the TRUE side of face d of EFF HEX 1
- All BAR vertices are located on the TRUE side of face b of EFF HEX 2
- All BAR vertices are located on the TRUE side of face e of EFF HEX 3
- The distance between the interior faces of EFF HEX 2, face b, and EFF HEX 3, face e, is less than some selected value .*
- The distance between the "bottom" face of EFF HEX 1, face d, and the "top" face of the BAR, face c, is less than some selected value. **
- The end effector is between the "ends" of the bar receptacle.

When these six conditions are all satisfied an acquisition has occurred and the program automatically enters the Extraction Phase (Mode 2).

During the Extraction Phase (Mode 2) the vertices of the payload and payload docking probe are tested for collision with the four interior walls and the bottom of the cargo bay. This collision-detection algorithm is identical to that employed for effector hex vertex - bar face collision. The payload and probe vertices assume the role of the effector hex vertices and the cargo bay walls and bottom faces assume the role of the bar faces.

---

*In the implementation this "selected value" is D3Y and may be changed by the experimenter.

**In the implementation this "selected value" is TPOB and may be changed by the experimenter.

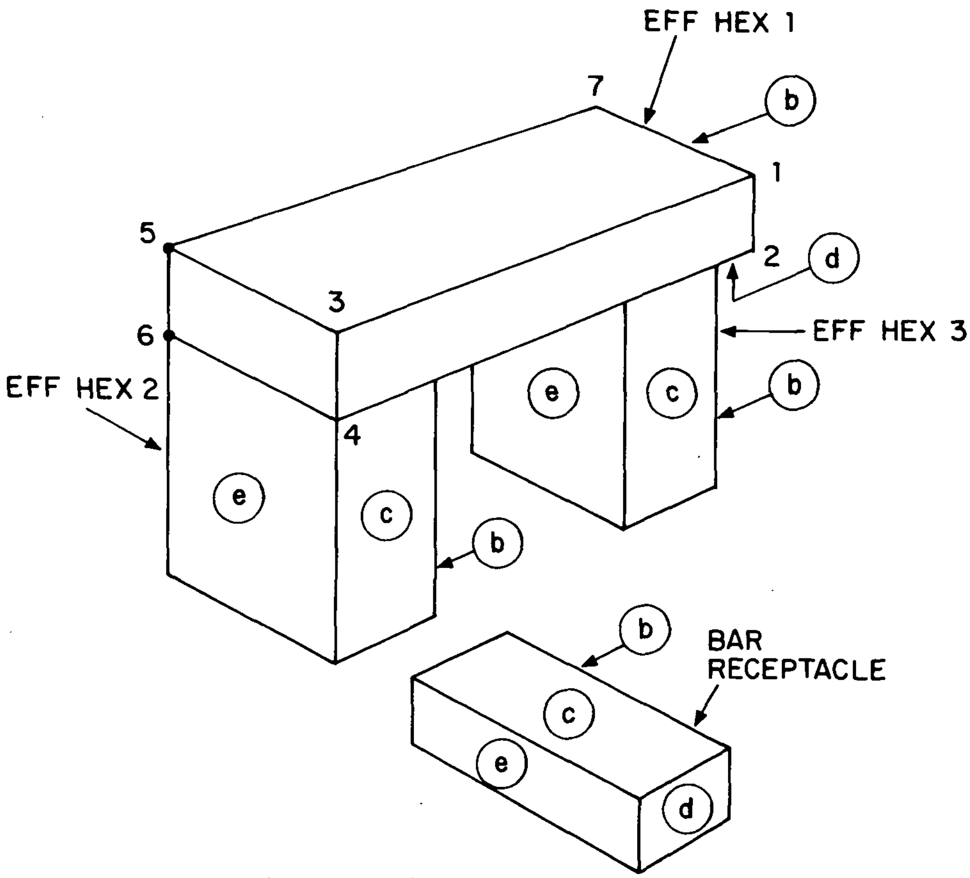
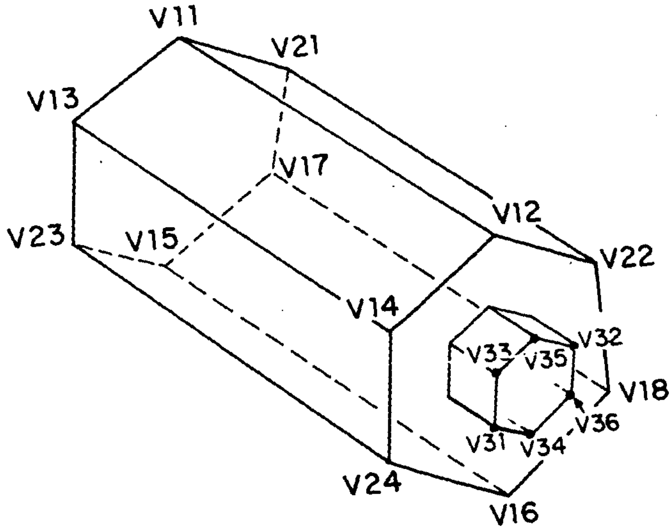


Figure 7. End Effector - Bar Receptacle Models

Only the collision of payload vertices and cargo bay walls are tested, no cargo bay vertex - payload face or edge-to-edge testing is done. For this test the payload is considered to consist of three "hexahedrons". Figure 8 shows the payload vertices that have grouped into the three "hexahedrons"; (V11, V12, V13, V14, V15, V16, V17, V18), (V21, V22, V23, V24) and (V31, V32, V33, V34, V35, V36). It also shows how the cargo bay face names are assigned. Payload and probe vertex positions with respect to the "top face", face c, of the cargo bay are also tested. When all vertices are above face c the payload is clear of the cargo bay and the maneuvering Phase (Mode 3) is automatically entered.

During the Maneuvering Phase the payload and payload probe vertices are tested for collision with the central assembly element, CAE. This collision-detection algorithm is also identical to that employed for effector hex vertex - bar face collisions. The payload and probe vertices again assume the role of the effector hex vertices. The CAE faces now play the role of the bar faces. The CAE, however, has eight faces where the bar had only six. Therefore, the CAE model for the Dynamic Collision Program consists of only the "front" half of the octahedron used to model the CAE. These faces are identified in Figure 9.



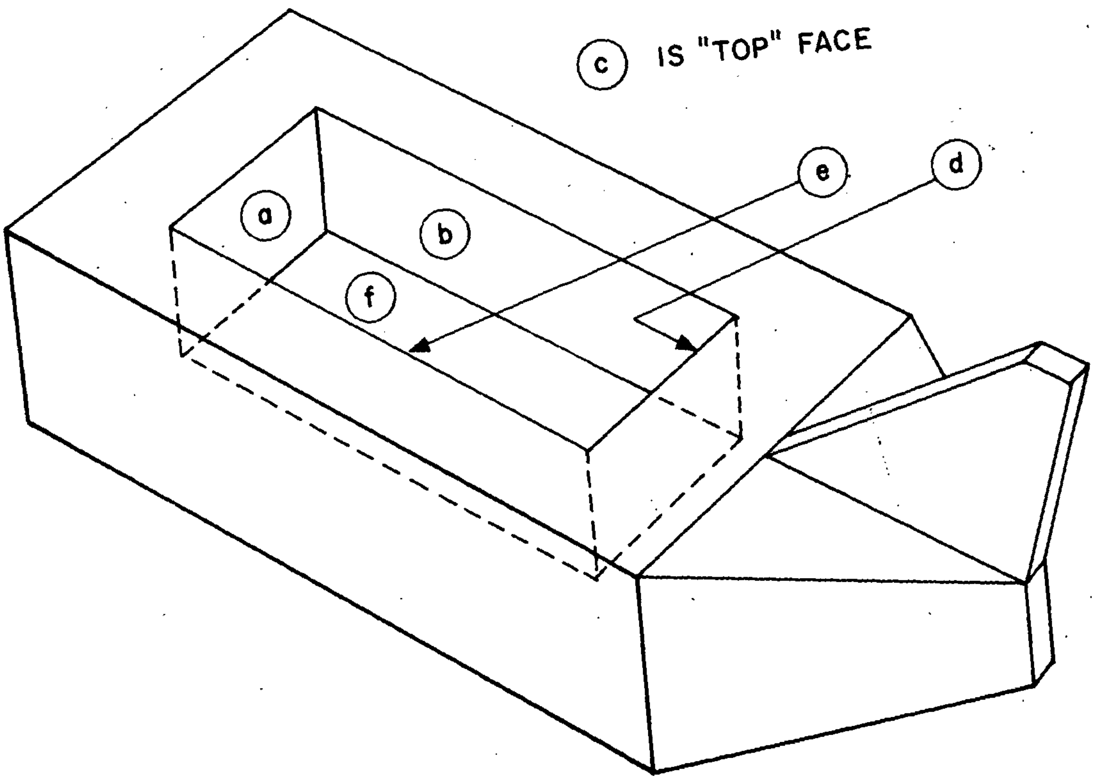


Figure 8. Payload and Cargo Bay Models

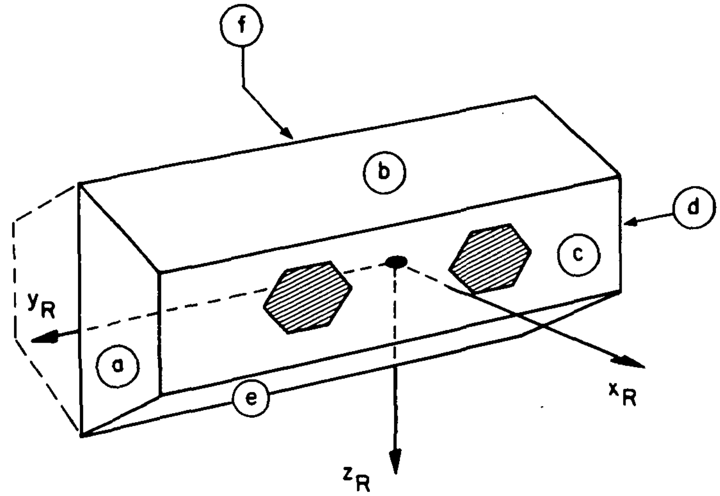


Figure 9. CAE: Front Half of the Octahedron

The mathematical detection of a successful docking is based upon the definition of a volume in space, which is labeled the docking envelope. This is the spatial volume immediately to the true side of CAE face c; it is bounded by that face and the mathematical extensions of CAE faces b, e, a, and d. Thus the docking envelope consists of all points lying on the true side of CAE face c, and on the false sides of CAE faces a, b, d and e. The test employed examines all the vertices of the payload docking probe. If, and only if, all these vertices are in the docking envelope a further test is performed to determine if the payload probe and CAE docking port are properly aligned within some assigned tolerance. This latter test consists of testing the distance between the center of the port and probe and the angle between the y-axis of the probe and the y-axis of the port. The test distance is set at 0.25 feet and the test angle at 2^{o} 32'.

B. CONTROL OF THE VISUAL SCENE PROGRAM

The DYNAMIC COLLISION PROGRAM controls the VISUAL SCENE PROGRAM. All of the system inputs, which originate at the operators station, are processed by the DYNAMIC COLLISION PROGRAM. These analog inputs are listed in Table I.

The sines and cosines of the six master arm angles are computed, used in the DYNAMIC COLLISION program computations, and transmitted to the ESG as a block of 12 twenty-four bit words.

The remaining analog input signals are processed by the DYNAMIC COLLISION program to form the scene generator control word. The format of this control word is shown in Table II.

The scene generator control word also contains six bits of information, which are generated by the DYNAMIC COLLISION PROGRAM.

TABLE I

ANALOG INPUTS

ANALOG INPUT CHANNEL	INPUT SIGNAL	CHARACTERISTIC	
1	Initialize	Two voltage levels: Run (OV) Initialize (+V)	
2	Initialize Select	Three voltage levels: Full Problem (OV) Mode 1 (+V) Mode 2 or 3 (-V)	
3	Heading 0 (HO)	Sholder Heading:	
4	Pitch 0 (PO)	Sholder Pitch	scaling
5	Pitch 1 (P1)	Elbow Pitch	set
6	Heading 2 (H2)	Wrist Heading	by analog
7	Pitch 2 (P2)	Wrist Pitch	adjustments
8	Roll 2 (R2)	Wrist Roll	
9	Telescope Arm	Three voltage levels Arm Stop (OV) Increase Length (+V) Decrease Length (-V)	
10	Camera Pan or Tilt	Five voltage levels: +Pan (+V) -Pan (+½V) No Pan, No Tilt (OV) +Tilt (-½V) -Tilt (-V)	
11	Camera Zoom	Three voltage levels: No Zoom (OV) Zoom In (+V) Zoom Out (-V)	
12	Camera Position Select	Eight voltage levels: Position A (+V) B (+3/4V) C (+½V) D (+1/4V) E (OV) F (-1/4V) G (-½V) H (-V)	
13	Movable Effector Prong Control	Three voltage levels: Prong Stop (OV) Close Prong (+V) Open Prong (-V)	

TABLE II

SCENE GENERATOR CONTROL WORD FORMAT

BIT POSITION	CONTROL FUNCTION
0	Initialize
1	Telescope Arm
2	Telescope Pos/Neg[1]
3	Tilt
4	Tilt or Pan Pos/Neg[1]
5	Pan
6	Zoom
7	Zoom In/Out
8	
9	Station Point Code[2]
10	
11	
12	Open Prong
13	Close Prong
14	Spare
15	Spare
16	Spare
17	Spare
18	Finger Motion Inhibit Flag[3]
19	Arm Motion Inhibit Flag[3]
20	Mode 4
21	Mode 3
22	Mode 2
23	Mode 1

1. Pos = "1"
   Neg = "0"

3. Motion Inhibit Flags

   Inhibit Motion = "1"

   Allow Motion = "0"

All other functions:
Enable function = "1"
Disable function = "0"

2. Station Point Codes are:

Station Point	Bit Number 8	9	10	11
A	0	0	0	0
B	0	0	0	1
C	0	0	1	0
D	0	0	1	1
E	0	1	0	0
F	1	0	0	0
G	1	0	0	1
H	1	0	1	0

Two of these bits, the Finger Motion Inhibit Flag and the Arm Motion Inhibit Flag are used to control the motion of manipulator components in the visual scene. The DYNAMIC COLLISION PROGRAM computes the spatial position of selected components of the manipulator system for each frame time and detects collisions. If the collision exists for two successive frames a motion inhibit flag is sent to the VISUAL SCENE program on the second frame, and on each succeeding frame until the collision is relieved. Two motion inhibit flags are required in the Acquisition Phase (Mode 1), where the moveable finger of the end effector is permitted to move independently of the other components of the effector. Thus, in MODE 1, the Arm Motion Inhibit Flag is set to a "one" to instruct the VISUAL SCENE program to inhibit all motion if any of the effector hexahedrons are in collision. The Finger Motion Inhibit Flag is set to a "one" only if the movable finger (EFF HEX 3) is in collision. During the other modes both Motion Inhibit Flags are set if the payload collides with the cargo bay (Mode 2) or the payload collides with the CAE (Mode 3).

The other four information bits generated by the DYNAMIC COLLISION PROGRAM, which are transmitted to the VISUAL SCENE PROGRAM, are the four mode bits, 20-23, of the Scene Generator Control Word.

In addition to the Scene Generator Control Word and the 12 sines and cosines the separating plane information is sent to the VISUAL PROGRAM. Each of the three separating planes is defined by the three components of a normalized normal and the perpendicular distance from the separating plane to the Reference Coordinate System origin. This perpendicular distance is the quality, D, in the equation of a plane ($AX+BY+CZ-D=0$), where A, B and C are the respective X, Y and Z components of the normalized normal. This perpendicular distance is calculated by taking the dot product of a point on the plane and the normalized normal.

C. PROGRAM FLOW CHART AND LISTING DESCRIPTION

The flow diagram of the DYNAMIC COLLISION PROGRAM is shown in Figure 10. The labels contained in the rectangular boxes correspond to the statement labels that appear in the label field of the program listing. The numbers to the right of the rectangles are the program listing line number. Those line numbers that are underlined refer to the second subroutine, MODE 2 and 3, of the assembly. The rectangles, with triangular ends indicate decision logic routines that have labels. The diamonds indicate decision logic that do not have a label assigned.

The program is written in XDS Symbol and operates on the XDS Sigma 5.

The program is a closed loop. It requires less than one frame time (50 msec) to complete one circuit of the loop. The exact time required is a function of the physical relationships that exist between the components of the manipulator system. The computational flow for a given picture frame begins when a WRITE DIRECT command is presented to the Sigma 5 by the IEU. The XMITESG routine then transmits a block of output data to the ESG and the computation for the next frame begins. After the computation is completed the program is "looping" at the first write loop of XMITESG awaiting the next WRITE DIRECT command from the IEU.

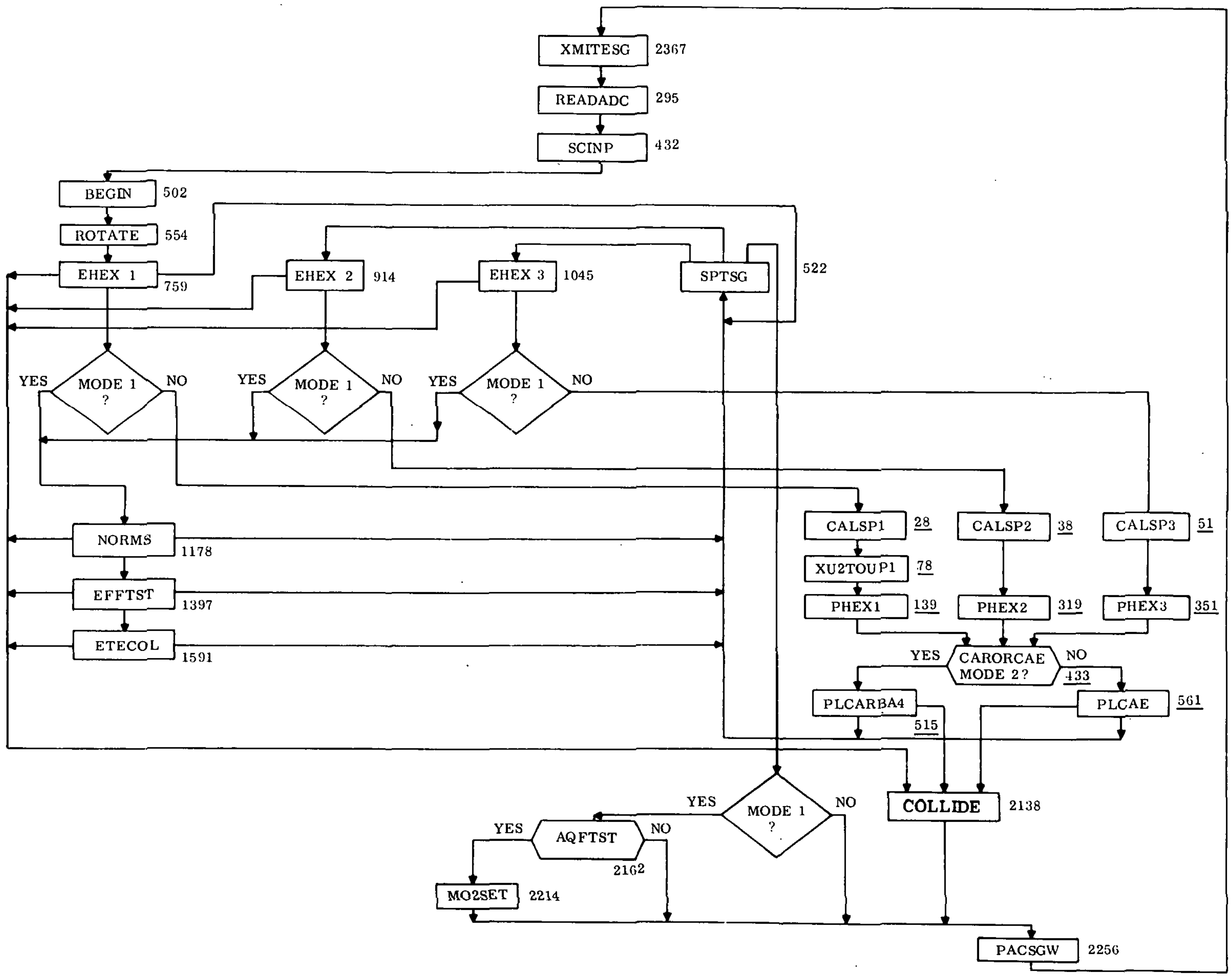


Figure 10. Dynamic Collision Program

1. Functional Description of the Subroutine

XMITESG

This routine consists of a sequence of WRITE DIRECT loops, which cause the output data block to be transmitted to the ESG. The output data block consists of 32 twenty-four bit words. The output data sequence consists of the scene generator control word, SGWRD, the twelve sines and cosines of the master arm angles, the three normals and the perpendicular distance for each of the three separating planes followed by seven blank words.

READADC

The first 13 input channels of the analog-to-digital converter are read in and stored in the INPUT Table (Line numbers 34-47 of the listing). The channel assignment matches the sequence of data as listed in the INPUT TABLE. The remainder of this routine is devoted to the processing of the input data to form the 14 most significant bits of the scene generator control word, SGWRD. (See Table II).

SCINP

The sine and cosine of the six master arm angles are calculated and stored in the SINE/COS data table; lines 52 thru 63 of the listing.

BEGIN

This is the initialization routine for the program. The three motion inhibit flags HIMF, HIMF+1 and HIMF+2 are reset. The effector hexahedron number, HEXM, and the greater than eight feet flag, ECTVF, are reset. * The MODE flag is examined. The Acquisition Phase (Mode 1) is denoted by a "-1", Extraction Phase (Mode 2) by a "0" and the Maneuvering Phase (Mode 3) by a "+1" in MODE. If MODE is greater than or equal to "0" the BAR data table, listing line number 248, contains either the cargo bay or CAE vertex data and the program branches to ROTATE. If MODE is "-1" the Bar receptacle vertex data (RECT data table, line 72 of the listing) is loaded into the BAR data table before the program branches to ROTATE.

ROTATE

This routine performs the rotations required to calculate the Reference System Coordinates of the effector (System Two) unit vectors, calculates the Reference System Coordinates of arm 1 and of arm 0 and adds these to find the Reference System Coordinates of the end point of arm 1.

EHEX 1

This routine generates the numerical model of EFF HEX 1 in reference system coordinates. As each vertex is computed it is also tested for collision with the top of the payload and if a collision is detected the program branches to COLLIDE. Another test is performed in this routine. The distance from the center of face d of EFF HEX 1 to the center of the bar receptacle is found.

---

*The use of the ECTVF flag is described below in the EHEX 1 subroutine description.

If this distance is greater than eight feet no collision between the end effector and the bar receptacle can occur. Thus there is no need to test the other two effector hexahedron. The ECTVF flag is set to a one and the same bar receptacle face is chosen for the separating plane for all three effector hexahedrons. The program branches to SPTSG. If the ECTVF flag is not set and no collision occurs the program branches to NORMS if MODE is "-1" and to CALSP1 if MODE is "0" or "1".

NORMS

This routine is used only in MODE 1. The vertices of the bar receptacle are tested for possible collision with the faces of the effector hex under test and to see if one of the effector hexahedron faces can be used as a separating plane. If a collision does occur the program branches to COLLIDE. If no collision occurs and an effector face is found to be a separating plane that face is further examined. It is examined to determine if it is face d, when EFF HEX 1 is under test, face b, when EFF HEX 2 is under test, or face e, when EFF HEX 3 is under test. If one of these conditions is true the appropriate acquisition flag, AQUIS, AQUIS+1 or AQUIS+2, is set and the program branches to SPTSG. If none of these conditions is true the appropriate AQUIS flag is reset. If there is no collision and no separating plane is found the program exits to EFFTST.

EFFTST

This routine is similar to NORMS. The difference being that the vertices of the effector hexahedron under test are now tested for collision with a bar face and to determine if a bar face can be chosen for a separating plane. There is no acquisition testing in this routine. If a collision occurs the program branches to COLLIDE. If a separating plane is found the branch is to SPTSG. If neither of these two events occurs the routine exits to ETECOL.

ETECOL

This is the edge-to-edge collision test routine. The effector vertex location words formed in EFFTST are examined to determine which edges of the effector hexahedron under test are to be tested for collision with the bar receptacle edges. Again if a collision is detected the branch is to COLLIDE. If no collision is detected a separating plane is found and the branch is to SPTSG.

EHEX 2 and EHEX 3

These two routines are similar to EHEX 1. They calculate the reference system coordinates of EFF HEX 2 and EFF HEX 3 respectively. They both exit to COLLIDE, if a collision with the top of the payload occurs, to NORMS if the mode is MODE 1. EHEX 2 exits to CALSP2 and EHEX 3 exits to CALSP3 if the mode is MODE 2 or MODE 3.

SPTSG

This is the computational control routine. The number of the hexahedron under test, HEXM, is updated. The separating plane normals and the coordinates of a point on the separating plane for the hexahedron that was under test is transferred from data table NSP, line number 254 of the listing to storage area NSPTSG, line 261 of the listing. If the updated HEXM is "2" the program branches to EHEX2, if it is "3" the branch is to EHEX 3, and if it is 4 MODE is examined. If MODE is "-1" the program branches to the acquisition final test AQFTST. If MODE is "0" or "1" the branch is to PACSGW to format the output data for transfer to the ESG.

COLLIDE

Whenever a collision is detected this routine is entered. The collision flags, H1CF, H1CF+1 and H1CF+2, are examined to see if a collision occurred during the previous frame. If none occurred only the collision flag for the hexahedron under test is set. If a collision occurred during the previous frame both the collision flag and the motion inhibit flag, H1MF or H1MF+1 or H1MF+2, are set for the hexahedron under test. The acquisition flag, AQUIS or AQUIS+1 or AQUIS+2, for the hexahedron under test is then examined. If this flag is not set the remaining hexahedrons do not require testing and the branch is to PACSGW, If it is set the branch is to SPTSG and the remaining hexahedrons are tested.

AQFTST

The purpose of this routine is to determine if all the conditions necessary for a successful acquisition of the payload bar receptacle by the end effector have been satisfied. These are the conditions listed above in section IV. B. 3. If, at anytime in the testing process, one of these conditions is found to be unsatisfied the program branches to PACSGW. Incorporated in this routine is a test to determine if an edge of the bottom face, face d, of EFF HEX 1, has collided with the top of the bar receptacle. This test requires checking the distance from the center of edges 2-4 and 6-8 of EFF HEX 1 and the top surface of the bar. If a collision does occur H1CF is examined to determine if a collision occurred during the previous frame. If so, H1MF and H1CF are both set; if not, only H1CF is set. If one of the edges has collided the program branches to PACSGW. If neither edge has collided and all the acquisition conditions are satisfied a successful acquisition has been made and the program branches to M02SET.

M02SET

This routine is entered if a successful acquisition of the payload bar receptacle has been accomplished. The program is to be be commanded to enter MODE 2 in the next picture frame by setting MODE to "0". Two other calculations are required before entering MODE 2. First the unit vector of System Two, expressed in Reference System coordinates, UZ2XR is examined to determine if the end effector moveable finger, EFF HEX 3, was in contact with bar receptacle face b or bar receptacle face e at acquisition.

If it was in contact with face b, UZ2XR is negative; if it was in contact with face e, UZ2XR is positive. The sign of UZ2XR determines an equivalence between the payload unit vectors, UXP, UYP and UZP, and the unit vectors of System Two that will exist in MODES 2 and 3. This equivalence, for the two possible signs of UZ2XR, is as follows:

PAYLOAD UNIT VECTORS	NEGATIVE UZ2XR	POSITIVE UZ2XR
UXPXR	-UZ2XR	UZ2XR
UXPYR	-UZ2 YR	UZ2YR
UXPZR	-UZ2ZR	UZ2ZR
UYPXR	UY2XR	-UY2XR
UYPYR	UY2YR	-UY2YR
UYPZR	UY2ZR	-UY2ZR
UZPXR	UX2XR	UX2XR
UZPYR	UY2XR	UY2XR
UZPZR	UZ2XR	UZ2XR

The second calculation is performed to find the vector between the end of arm 1 and the center of the payload. At acquisition, the Reference Coordinates of the end point of arm 1 one are subtracted from the Reference Coordinates of the center of the payload to obtain this vector whose components are VP1XR, VP1YR and VP1ZR. This vector is expressed in the System Two coordinate system as a function of the sign of UZ2XR as follows:

ARM 1 TO PAYLOAD CTR.	NEGATIVE UZ2XR	POSITIVE UZ2XR
VP1X2	VP1ZR	VP1ZR
VP1Y2	VP1YR	-VP1YR
VP1Z2	-VP1XR	VP1XR

All three collision flags and all three motion inhibit flags are reset and M02SET exits to PACSGW.

PACSGW

The ten least significant bits of the scan generator control word, SGWRD, are found, the perpendicular distances for the three separating planes are calculated and the sines and cosines of the master arm angles rescaled by this routine. It exits to XMITESG.

The routines described up to this point are employed in MODE 1. The routines that follow are employed in MODE 2 and MODE 3, where the collisions to be detected occur between the payload and the cargo bay or between the payload and CAE.

CALSP1

In MODE 2 and 3 the payload is under acquisition by the end effector. Face d of EFF HEX 1 is assigned as the separating plane between EFF HEX 1 and the bar receptacle by this routine. The exit is to XU2TOUPL.

XU2TOPUL

This routine assigns the System Two unit vectors (UX2, UY2 and UZ2) to their equivalent payload unit vectors (UXP, UYP and UZP) as a function of the sign of UZ2XR at acquisition. This assignment is that listed above in the M02SET routine. Exit is to PHEX1.

PHEX 1

The Reference System Coordinates of the vector between the end of arm 1 and the payload center, (VP1XR, VP1YR and VP1ZR) are calculated. The Reference System Coordinates of the vertices of hexahedron one of the payload are calculated and stored in the BAR table. Exit is to CARORCAE.

CARORCAE

MODE is examined.

If MODE is "0" (MODE 2) the cargo bay vertices, which are in the cargo bay vertex table, CARBAY table, line 103 of the listing, are put into the Effector Vertex, Table, EE table, line 240. The normals to the cargo bay faces are assigned and put into the Effector Normal table, NE table, line 245 of the listing. Exit is to PLCARBAY.

If MODE is "1" (MODE 3) a branch to CAEINEE occurs where the Central Assembly Element vertices, CAE table, line 129 of the listing, are put into the EE table and the normals to the CAE faces calculated and put into the NE table. Exit is to PLCAE.

PLCARBAY

This routine is used in MODE 2 to test the payload for collision with the walls and bottom of the cargo bay and to detect whether all the vertices are above the top face, face c, of the cargo bay. The vertex collision test employs the subroutine CALLWR. If a collision occurs the program branches to COLLIDE. If no collision occurs for payload hexahedron one and two the program branches to SPTSG. If no collision occurs for payload hexahedron three, and all the vertices of the three payload hexahedrons are above face c of the cargo bay, MODE is incremented before the branch to SPTSG occurs. The program will then enter MODE 3.

PLCAE

This routine is employed in MODE 3 to test if payload vertices collide with a CAE face or if a successful docking is accomplished. The vertex collision test also employs the subroutine CALLWR. If a collision occurs the program immediately branches to COLLIDE. If no collision occurs for payload hexahedrons one and two the branch is to SPTSG. Payload hexahedron three consists of the docking probe vertices. These vertices are tested to see if they are within the docking envelope. If they are the docking test is performed. This is the test that checks the distance between the center of the docking probe and the docking port and the angle between the y-axis of the port and the probe. If the conditions for a successful docking are satisfied MODE is incremented before the branch to SPTSG is made. MODE equal to "3" (MODE 4) indicates that a successful docking has occurred.

CALSP2 and CALSP3

These routines respectively assign face b of EFF HEX 2 and face e of EFF HEX 3 as separating planes between payload hexahedron three and the bar receptacle.

PHEX2 and PHEX 3

The two routines are similar to PHEX1. They calculate the reference system coordinates of payload hexahedron two and payload hexahedron three.

V. THE VISUAL SCENE PROGRAM

The function of the VISUAL SCENE PROGRAM is to perform the frame by frame calculations related to data input processing, coordinate rotations, and priority determination. The results of these calculations generate the visual scenes of the space shuttle, remote manipulator, payload, and central assembly element, (CAE).

The scene can be viewed from eight possible station or camera positions. Three dynamic coordinate systems and a reference coordinate system are used. The shuttle and central assembly element are located in the reference coordinate system. The three manipulator arm segments are each located in a dynamic coordinate system of their own. The shoulder joint has two degrees of freedom (heading and pitch), the elbow joint has one degree of freedom (pitch), and the wrist joint has three degrees of freedom (heading, pitch and roll).

The payload, modeled as a octahedron that can be a maximum 60 feet long and 15 feet in diameter, is located in the shuttle cargo bay and described in Reference System coordinates for the Acquisition Phase. The payload remains in Reference System coordinates until it is acquired by the end effector. At this time, and throughout any subsequent maneuvering operations, the payload is described in the same dynamic coordinate system, System Two, as the manipulator end effector. This change from one coordinate system to another allows the payload to be moved without a separate and special coordinate system and control logic of its own.

The visual scene is made up from a number of objects which are in turn assembled to build the desired model. As the manipulator is moved or the view point changed, the relative visual priority between objects changes.

Separating planes are specified for the model to visually separate the objects. The priority routine determines the position of the view point relative to these separating planes and calculates the correct object priority list for each frame.

The VISUAL SCENE PROGRAM is implemented on the R520 of the ESG. It consists of three main sections, the Source Input Routine, the Dynamic Routine, and the Priority Determination Routine. Several associated subroutines are utilized by each of the three main sections. The main calling sequence was developed for each of the main sections, along with a number of special routines as dictated by Remote Manipulator requirements. These include arm telescope and finger movement, camera zoom, camera pitch and pan. The calculation sections were developed under contract No. NAS 9-3916 and are used as is or with slight modifications.

A. SOURCE INPUT ROUTINE

The Source Input Routine selects the IIU, for data transfer to the R520, once during each frame. This data consists of the scene generator control word, master arm angles in the form of sin/cos, and bar receptacle - end effector separating plane data in the form of unit normals to the planes and the perpendicular distance to the planes from reference coordinate system origin.

If bit 19 (arm motion flag) of the scene generator control word is a "1", the R520 disconnects from the IIU and does not input new data for the current frame. The data received during the previous frame is used by the R520 for its current frame calculations. If bit 19 of the control word is a "0", the transmission of data from the IIU to the R520 is completed and then the disconnect is performed.

The Source Input Routine, after disconnecting from the IIU, checks the appropriate bits of the control word and performs the functions shown in Figure 11 the Source Input Routine flow diagram. The Source Input Routine also transfers to a typewriter input-output routine, which is used to input or print out data via the typewriter.

Upon completion of its appointed tasks, the Source Input Routine transfers control to the Dynamic Calculation Routine.

B. DYNAMICS ROUTINE (See Dynamics Routine flow diagram, Figure 12 a and b and the coordinate systems, Figure 13)

The main functionof the Dynamic Routine is to calculate the required L and P vectors, which are used in the VCU to form the intercept (A) and slope (B) quantities for environment edges. The L and P vectors are formed first in the coordinate system in which they are located. They are then resolved into all other coordinate systems in which objects of the environment appear. Then they are stored for future use in the Priority Routine and for transmission to the VCU. The equations for the formation of the L and P vectors are:

1. EQUATIONS FOR THE L & P VECTORS - STATION POINTS (CAMERA) A, B, C, D, E, H, & I.

$$LR = \begin{pmatrix} -SPX \\ -SPY \\ -SPZ \end{pmatrix}$$

Where proper station point components have previously been loaded into SPX, SPY, & SPZ.

$$LO = (\theta_0)(\psi_0)\left[TR+LR\right]$$

$$L1 = (\theta_1)\left[TO+LO\right]$$

$$L2 = (\phi_2)(\theta_2)(\psi_2)\left[T1+L1\right]$$

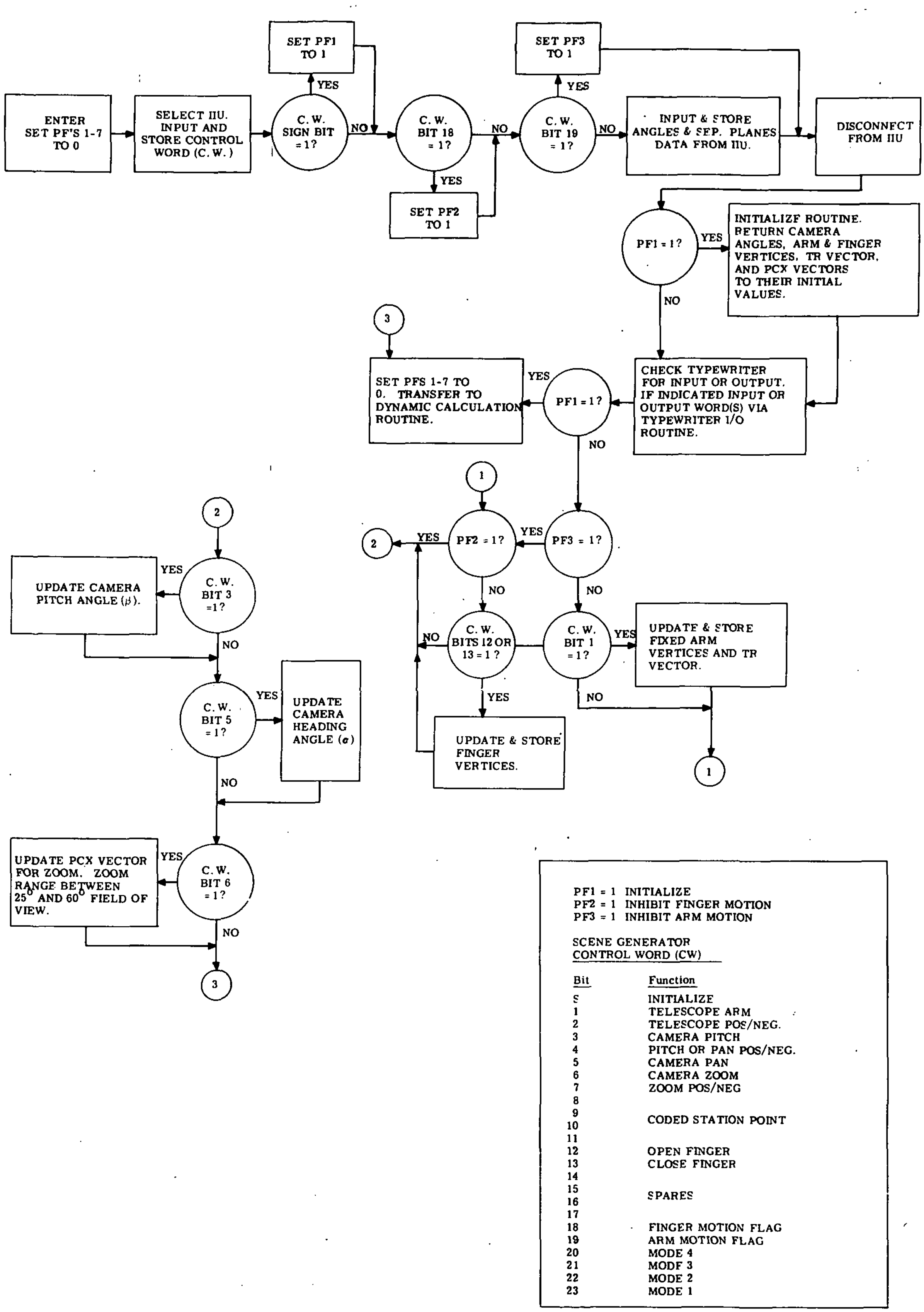


Figure 11. Source Input Routine

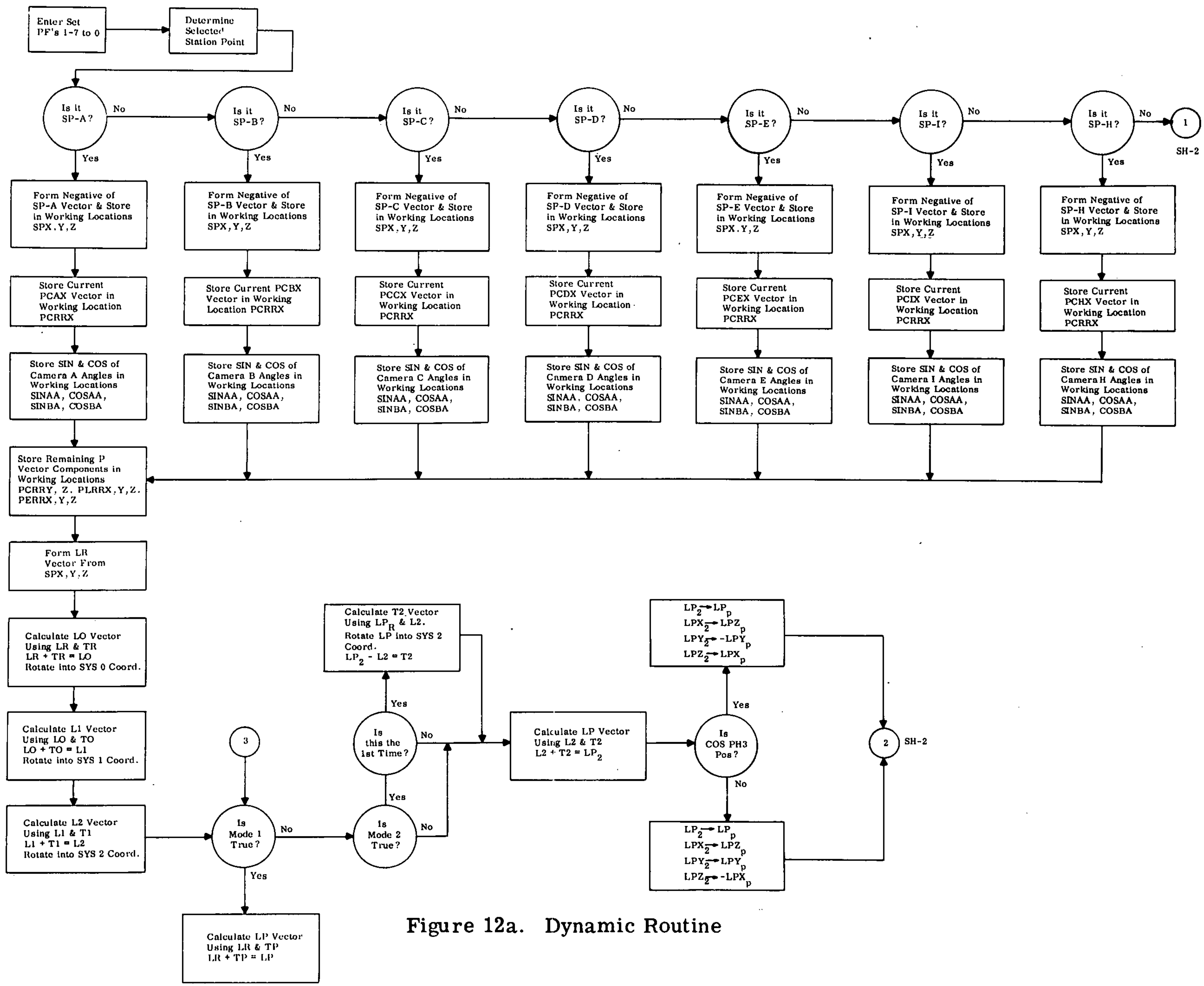


Figure 12a. Dynamic Routine

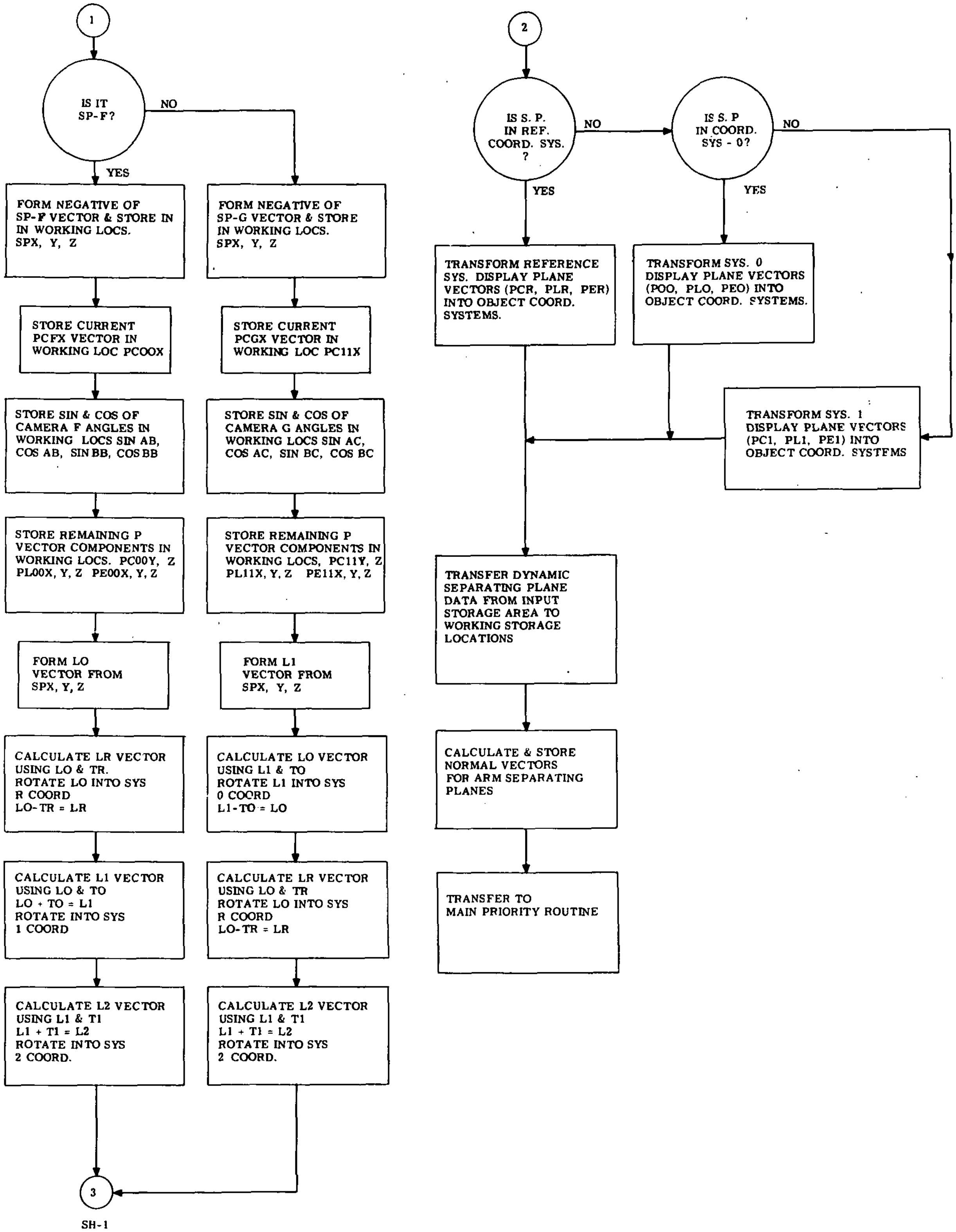
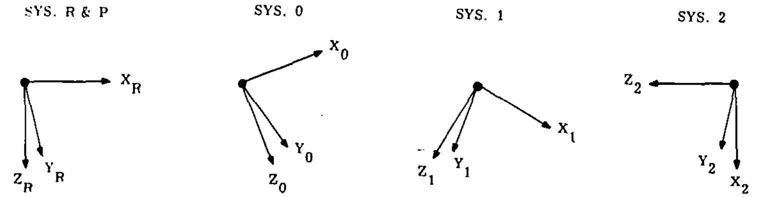


Figure 12b. Dynamic Routine

ORIENTATION OF COORDINATE SYSTEMS FOR ARM POSITIONS SHOWN BELOW AND PAYLOAD IN SHUTTLE.



THE X-AXES OF COORDINATE SYSTEMS 0, 1, & 2 ALWAYS LIES ALONG ARM 0, 1, & 2.

COORDINATE SYSTEM	DEGREES OF FREEDOM	WHAT THEY ARE	PROGRAM NOMENCLATURE
0	2	HEADING ψ_0 PITCH θ_0	ψ_1 PS1 θ_1 TH1
1	1	PITCH θ_1	θ_2 TH2
2	3	HEADING ψ_2 PITCH θ_2 ROLL ϕ_2	ψ_3 PS3 θ_3 TH3 ϕ_3 PH3

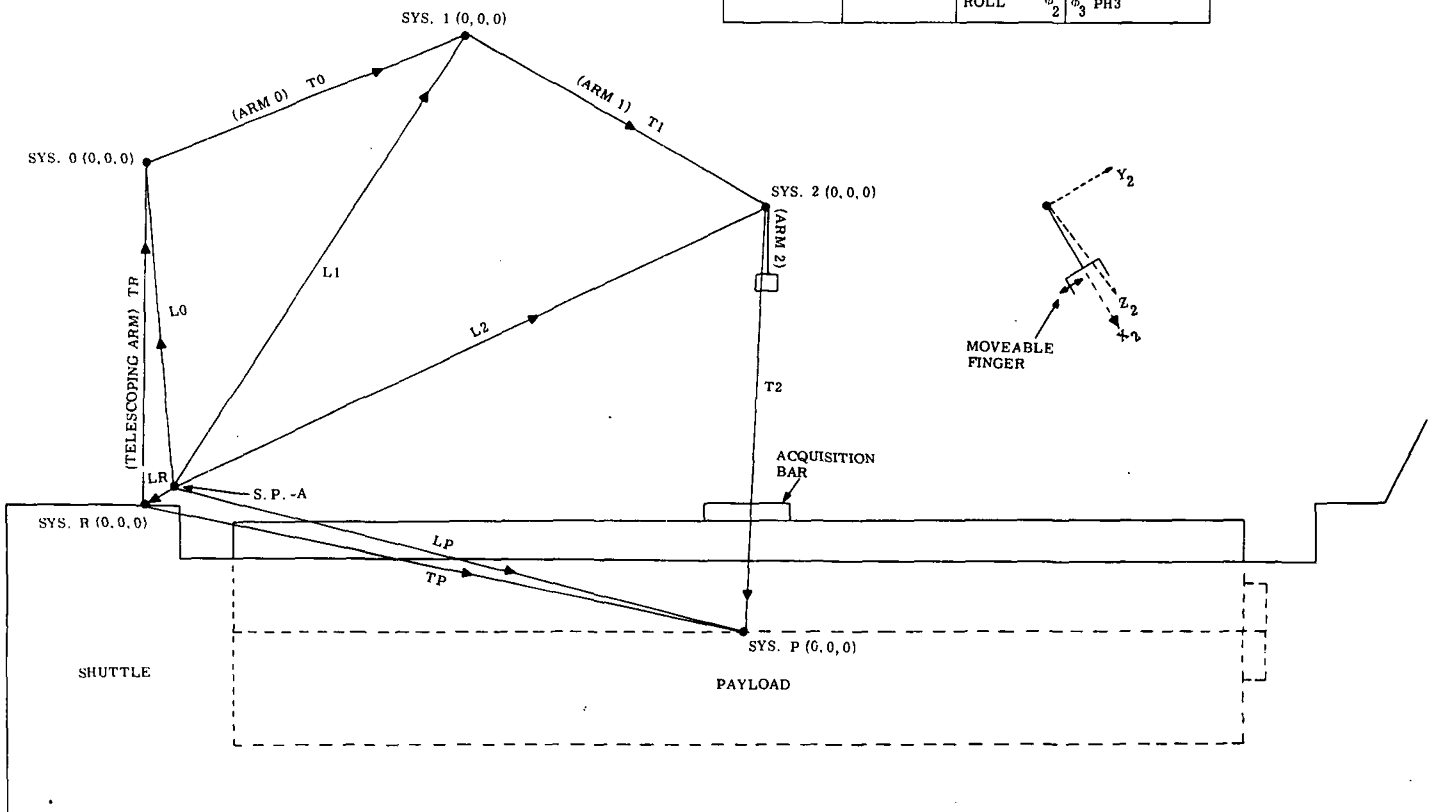


Figure 13. Coordinate Systems

$$LP = \begin{cases} [LR+TP] & \text{If Mode 1 is True} \\ \begin{pmatrix} -[L2Z+T2Z] \\ L2Y+T2Y \\ L2+T2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Positive.} \\ \begin{pmatrix} L2Z+T2Z \\ -[L2Y+T2Y] \\ L2X+T2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Negative.} \end{cases}$$

$$PCR = (-\alpha_A)(-\beta_A) \begin{matrix} PCRRX \\ PCRRY \\ PCRRZ \end{matrix}$$

$$PCO = (\theta_0)(\psi_0)(PCR)$$

$$PC1 = (\theta_1)(PCO)$$

$$PC2 = (\phi_2)(\theta_2)(\psi_2)(PC1)$$

$$PCP = \begin{cases} PCR & \text{If Mode 1 is True} \\ \begin{pmatrix} -PC2Z \\ PC2Y \\ PC2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Positive.} \\ \begin{pmatrix} PC2Z \\ -PC2Y \\ PC2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Negative.} \end{cases}$$

PL & PE VECTOR EQUATIONS ARE THE SAME AS PC ABOVE WITH L, THEN E, SUBSTITUTED FOR C IN EQUATIONS, i.e.,

$$PLR = (-\alpha_A)(-\beta_A) \begin{pmatrix} PLRRX \\ PLRRY \\ PLRRZ \end{pmatrix}$$

2. EQUATIONS FOR THE L & P VECTORS - STATION POINT (CAMERA) F. (ARM 0).

$$LO = \begin{pmatrix} -SPX \\ -SPY \\ -SPZ \end{pmatrix}$$

Where proper station point components have previously been loaded into SPX, SPY, & SPZ

$$LR = -TR + (-\psi_o)\,(-\theta_o)\,(L0)$$

$$L_1 = (\theta_1)\ TO + L0$$

$$L_2 = (\phi_2)\,(\theta_2)\,(\psi_2)\,\left[T1 + L1\right]$$

$$LP = \begin{cases} \left[LR \quad LR + TP\right] & \text{If Mode 1 is True} \\ \begin{pmatrix} -\left[L2Z+T2Z\right] \\ L2Y+T2Y \\ L2X+T2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Positive.} \\ \begin{pmatrix} L2Z+T2Z \\ -\left[L2Y+T2Y\right] \\ L2X+T2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Negative.} \end{cases}$$

$$PCO = (-\alpha_B)\,(-\beta_B)\begin{pmatrix} PC00X \\ PC00Y \\ PC00Z \end{pmatrix}$$

$$PCR = (-\psi_o)\,(-\theta_o)\,(PC0)$$

$$PC1 = (\theta_1)\,(PC0)$$

$$PC2 = (\phi_2)\,(\theta_2)\,(\psi_2)\,(PC1)$$

$$PCP = \begin{cases} PCR & \text{If Mode 1 is True} \\ \begin{pmatrix} -PC2Z \\ PC2Y \\ PC2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Positive.} \\ \begin{pmatrix} PC2Z \\ -PC2Y \\ PC2Z \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ if Negative.} \end{cases}$$

PL & PE VECTOR EQUATIONS ARE THE SAME AS PC ABOVE WITH L, THEN E, SUBSTITUTED FOR C IN EQUATIONS.

3. EQUATIONS FOR L & P VECTORS FOR STATION POINT (CAMERA) G. (ARM 1)

$$L1 = \begin{pmatrix} -SPX \\ -SPY \\ -SPZ \end{pmatrix}$$

Where proper station point components have previously been loaded into SPX, SPY, SPZ

$$LO = -TO + (-\theta_1)(L1)$$

$$LR = -TR + (-\psi_o)(-\theta_o)(L0)$$

$$L2 = (\phi_2)(\theta_2)(\psi_2)\left[T1 + L1\right]$$

$$LP = \begin{cases} \left[LR + TP\right] & \text{If Mode 1 is True.} \\ \begin{pmatrix} -\left[L2Z+T2Z\right] \\ L2Y+T2Y \\ L2X+T2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Positive.} \\ \begin{pmatrix} L2Z+T2Z \\ -\left[L2Y+T2Y\right] \\ L2X+T2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Negative.} \end{cases}$$

$$PC1 = (-\alpha_c)(-\beta_c)\begin{pmatrix} PC11X \\ PC11Y \\ PC11Z \end{pmatrix}$$

$$PC0 = (-\theta_1)(PC1)$$

$$PCR = (-\psi_o)(-\theta_o)(PC0)$$

$$PC2 = (\phi_2)(\theta_2)(\psi_2)(PC1)$$

$$PCP = \begin{cases} PCR & \text{If Mode 1 is True.} \\ \begin{pmatrix} -PC2Z \\ PC2Y \\ PC2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Positive.} \\ \begin{pmatrix} PC2Z \\ -PC2Y \\ PC2X \end{pmatrix} & \text{If Mode 1 is False \& COS } \phi_2 \text{ is Negative.} \end{cases}$$

PL & PE VECTOR EQUATIONS ARE THE SAME AS PC ABOVE WITH L, THEN E, SUBSTITUTED FOR C IN EQUATIONS.

The initial vertex data for the top of the telescoping arm and the moveable finger of the end effector are stored in the R520. This initial vertex data is used, along with a fixed incremental value, to update the vertex position of the telescoping arm or the movable finger when dictated by the control word. The updated vertex data is stored in the VCU output block for subsequent transmission to the VCU. This data is stored in the correct vertex location by the VCU such that the vertex pair list is correct for formation of the edges involved.

The dynamics routine also calculates the normal vectors to the separating planes between arm 0/arm 1 and arm 1/arm 2 for later use in the priority routine (SP 11 and SP 12 of Figure 14). These normals are calculated to insure that the separating plane bisects the angle between the adjacent arms.

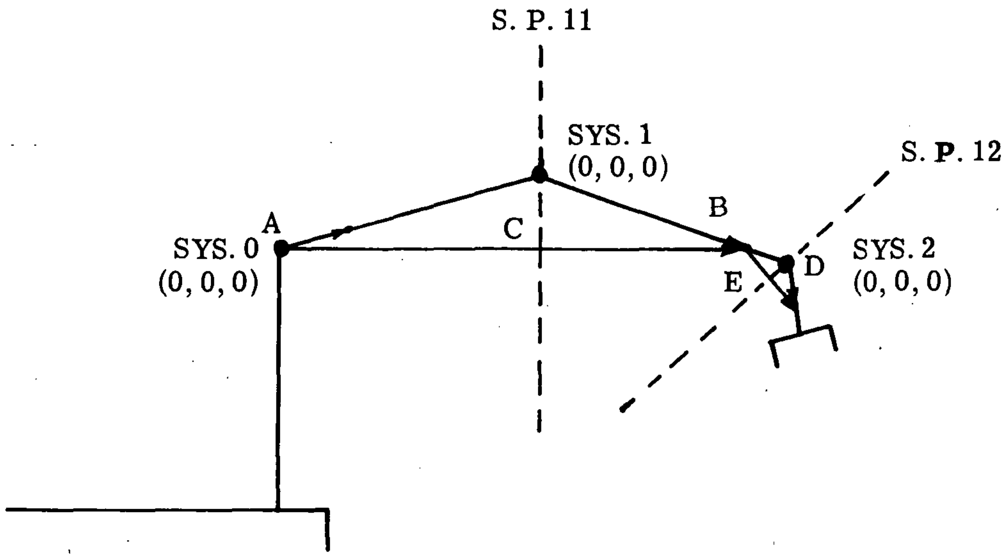
Table III lists some important quantities, such as camera position and camera heading & pitch, along with their associated memory storage location and scaling information. This table is an operator aid for use in changing physical parameters, if and when the need arises.

The relative positioning of cameras (station points) with respect to fixed object separating planes is used by the Priority Routine in its frame-by-frame priority determination sequence. It is important to recognize that there are limitations in repositioning cameras (station points) because of this relationship. If a camera position change is desired, it is suggested that a careful study be made to guarantee that it will not cause a priority conflict between objects due to its new position with respect to fixed separating planes in the system.

C. PRIORITY DETERMINATION ROUTINE

The solution of visual priority in the Remote Manipulator Simulation requires a determination of relative location, with respect to the observer, of collections of model faces called clusters. A cluster is a group of faces whose relative visual priority may be specified by a fixed list. This list specifies a fixed order in which the images of the faces must be overlaid to insure that, for any view of the group of faces, the visual priorities are always correct. The image of the overall model is constructed by overlaying images of the constituent clusters. This overlay order is determined by observer position with respect to the clusters and by the relative cluster locations.

The relative location of clusters and the subsequent order in which their images should be overlaid changes due to motion dynamics exhibited by the manipulator arm and payload. When the relative location of clusters is fixed, model space may be subdivided by a decision tree of fixed separating planes such that the order in which cluster images are overlaid is determined simply from observer position (see Instruction Manual for Electronic Scene Generator Expansion System, Vol. II, and interim specifications outlined in Off-Line Software for Expanded Electronic Scene Generator, delivered under Contract No. NAS 9-11065). However, when relative cluster motion is introduced via multiple, chained coordinate systems, repositioning of separating planes or restructuring of the decision tree or of the model itself may be necessary.



1) Form Normal to S. P. 11

a) Resolve A into SYS. 1 Coordinates

$$A = (\theta_1) \begin{pmatrix} X_o \\ 0 \\ 0 \end{pmatrix}$$

b) Normal to S. P. 11 = A + B = C

2. Form Normal to S. P. 12

a) Resolve B into SYS. 2 Coordinates

$$B = (\phi_2)\,(\theta_2)\,(\psi_2) \begin{pmatrix} X_1 \\ 0 \\ 0 \end{pmatrix}$$

b) Normal to S. P. 12 = B + D = E

Figure 14. Formation of Normals to Dynamic Arm Separating Planes

TABLE III

MEMORY LOCATION AND SCALING

FUNCTION			COORDINATE SYSTEM	SCALING	R520 MEMORY LOCATION
CAMERA	A	X POSITION	REF.	B11	16000
	A	Y POSITION	REF.	B11	16001
	A	Z POSITION	REF.	B11	16002
	A	HEADING α_A	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16033
CAMERA	A	PITCH β_A	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16034
CAMERA	B	X POSITION	REF.	B11	16003
	B	Y POSITION	REF.	B11	16004
	B	Z POSITION	REF.	B11	16005
	B	HEADING α_B	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16035
CAMERA	B	PITCH β_B	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16036
CAMERA	C	X POSITION	REF.	B11	16006
	C	Y POSITION	REF.	B11	16007
	C	Z POSITION	REF.	B11	16010
	C	HEADING α_C	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16037
CAMERA	C	PITCH β_C	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16040
CAMERA	D	X POSITION	REF.	B11	16011
	D	Y POSITION	REF.	B11	16012
	D	Z POSITION	REF.	B11	16013
	D	HEADING α_D	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16041
CAMERA	D	PITCH β_D	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16042
CAMERA	E	X POSITION	REF.	B11	16014
	E	Y POSITION	REF.	B11	16015
	E	Z POSITION	REF.	B11	16016
	E	HEADING α_E	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16043
CAMERA	E	PITCH β_E	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16044
CAMERA	F	X POSITION	ZERO (0)	B11	16017
	F	Y POSITION	ZERO (0)	B11	16020
	F	Z POSITION	ZERO (0)	B11	16021
	F	HEADING α_F	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16045
CAMERA	F	PITCH β_F	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16046
CAMERA	G	X POSITION	ONE (1)	B11	16022
	G	Y POSITION	ONE (1)	B11	16023
	G	Z POSITION	ONE (1)	B11	16024
	G	HEADING α_G	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16047
CAMERA	G	PITCH β_G	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16050
CAMERA	H	X POSITION	REF.	B11	16025
	H	Y POSITION	REF.	B11	16026
	H	Z POSITION	REF.	B11	16027
	H	HEADING α_H	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16051
CAMERA	H	PITCH β_H	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16052
CAMERA	I	X POSITION	REF.	B11	16030
	I	Y POSITION	REF.	B11	16031
	I	Z POSITION	REF.	B11	16032
	I	HEADING α_I	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16053
CAMERA	I	PITCH β_I	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16054
CENTER OF PAYLOAD POSITION					
	X	UPPER WORD	REF.	B11	16121
	X	LOWER WORD	REF.	B11	16122
	Y	UPPER WORD	REF.	B11	16123
	Y	LOWER WORD	REF.	B11	16124
	Z	UPPER WORD	REF.	B11	16125
	Z	LOWER WORD	REF.	B11	16126

TABLE III - CONCLUDED

MEMORY LOCATION AND SCALING

FUNCTION	COORDINATE SYSTEM	SCALING	R520 MEMORY LOCATION
INITIAL VERTEX POSITIONS FOR TOP OF TELESCOPING ARM			
V1Z	REF.	B16	16170
V2Z	REF.	B16	16171
V3Z	REF.	B16	16172
V4Z	REF.	B16	16173
V5Z	REF.	B16	16174
INITIAL VERTEX POSITIONS FOR MOVEABLE FINGER			
V1Y	TWO (2)	B16	16203
V2Y	TWO (2)	B16	16204
V3Y	TWO (2)	B16	16205
V4Y	TWO (2)	B16	16206
V5Y	TWO (2)	B16	16207
V6Y	TWO (2)	B16	16210
V7Y	TWO (2)	B16	16211
V8Y	TWO (2)	B16	16212
ARM TELESCOPE RATE FOR ARM VERTICES			
PDV	N. A.	B16	16306
NDV	N. A.	B16	16307
ARM TELESCOPE RATE FOR TR VECTOR			
NDT	N. A.	B11	16213
PDT	N. A.	B11	16214
FINGER MOTION RATE			
NDF	N. A.	B16	16215
PDF	N. A.	B16	16216
CAMERA PITCH/PAN RATE			
PDA	N. A.	MSB = 180°, NEXT MSB = 90°, Etc.	16227
NDA	N. A.		16230
CAMERA ZOOM RATE			
PDPCX	N. A.	B3	16231
NDPCX	N. A.	B3	16232
INITIAL TR VECTOR COMPONENTS			
TRXU	REF.	B11 ON LOWER WORD	16077
TRXL	REF.	B11 ON LOWER WORD	16100
TRYU	REF.	B11 ON LOWER WORD	16101
TRYL	REF.	B11 ON LOWER WORD	16102
TRZU	REF.	B11 ON LOWER WORD	16103
*TRZL	REF.	B11 ON LOWER WORD	16104

The problem of dynamic visual priority in the Remote Manipulator Simulation can be simplified by taking advantage of inherent restrictions on model motion and observer positioning. This approach avoids restructuring of the model, permits a limited set of decision trees to suffice in solving dynamic priority, and utilizes known fixed priority relationships wherever possible. In particular, observer positioning is limited to nine fixed locations on the shuttle, manipulator arm, and central assembly element (CAE). Changing cluster relationships are classified by operational mode, i. e., Model: Acquisition Phase, Mode 2: Extraction Phase and Mode 3: Manuvering Phase. Arm and payload motion is assumed to be constrained to the space inside the cargo bay, above the top surface of the shuttle and to the "front" or positive XR side of the CAE.

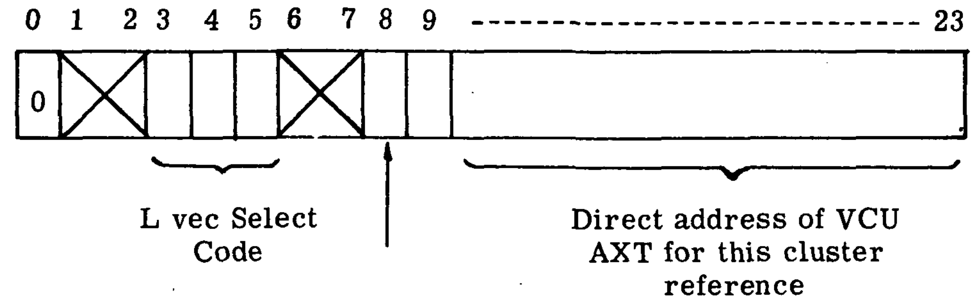
Relative cluster ordering for priority overlay is determined via three main devices: separating plane decision trees, fixed priority lists whose entries reference clusters or decision trees, and special sorting of a cluster list on the basis of separating plane decisions that are not amenable to a tree structure. Dynamic separating planes are used in situations where observer position, with respect to fixed planes and operational mode, does not provide sufficient information for correctly resolving relative cluster position. For example, dynamic separating planes are used to determine the priority between the end effector and the bar receptacle.

1. Operational of Visual Priority Routine, MAINPRIOR

The purpose of the priority routine, MANIPRIOR, is to compute a list of VCU instructions, called the Priority Output Table (POT) , to be used to the output to the VCU once per frame. In the VCU, the POT directs changes in the L vector used in aspect testing and implicity establishes the order in which clusters are to be overlaid in a given frame. In MAINPRIOR, L vector initialization is added to the POT whenever subsequent cluster references must be associated with a new dynamic coordinate system. Cluster references are added to the POT as the relative ordering of clusters for priority overlay is determined. The highest-to-lowest priority is from top down in the POT.

Each camera or observer position is associated with a Permanent Priority List (PPL) whose entries are Direct Cluster References or Tree Root Node References (see Figure 15). Each pass through MAINPRIOR begins by selecting the PPL associated with the current camera position. Then individual PPL entries are processed. Each entry is checked for L vector initialization. If such is called for, the L vector select code (see Table IV) in the entry is used to (1) form a corresponding L vector reference that is added to the POT, and (2) initialize the bottom of the separating plane L vector pushdown stack used during tree processing. Then, if the entry is a Direct Cluster Reference, corresponding cluster instructions are added to the POT. If the entry is a Tree Root Node Reference, the referenced decision tree is traced for its effects on the POT. The determination of which tree is referenced is made directly from the PPL entry address field if no mode test is called for. If mode test is specified, then the current mode and the mode upper limit and tree type in the PPL entry are used to determine which tree is referenced (see Figure 15). The PPL is exhausted on encounter with a zero entry.

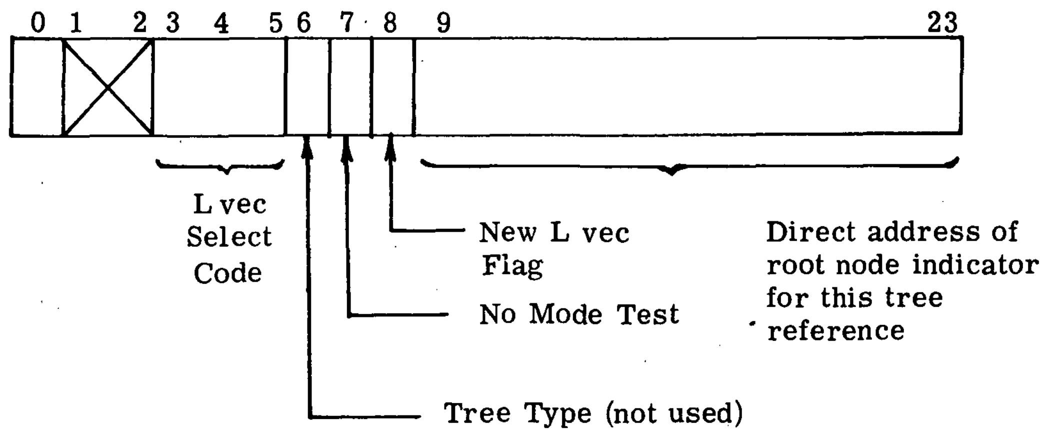
1. Direct Cluster Reference (Sign = 0)



2. Tree Root Node Reference (Sign = 1)

a Formats:

(a) No Mode Test (Bit 7 = 0)



(b) Mode Test (Bit 7 = 1)

Figure 15. Permanent Priority List Entries

TABLE IV

L VEC SELECT CODE FOR PPL ENTRIES, INDICATORS, AND SONS

Bits 3-5	Lvec referenced
0 0 0	LR (Reference)
0 0 1	L0 (Sys. 0)
0 1 0	L1 (Sys. 1)
0 1 1	L2 (Sys. 2)
1 0 0	LP (Payload)

TABLE V

L VECTOR NOMENCLATURE

Mnemonic	Coordinate System	Remarks
LR	Reference	Fixed
L0	0	Upper arm chained to ref. sys.
L1	1	Lower arm chained to sys. 0
L2	2	End effector chained to sys. 1
LP	Payload	Fixed or chained to sys. 2

TABLE VI

GENERAL STATION POINT POSITIONING

Station Point	Location
A	Forward End of Cargo Bay
B	Left Side of Cargo Bay
C	Aft End of Cargo Bay
D	Right Side of Cargo Bay
E	Vertical Telescoping Arms
F	Shoulder To Elbow Arm (ARMO)
G	Elbow and Wrist Arm (ARM 1)
H	Central Assembly Element (CAE)
I	Shuttle Nose (forward of CAE)

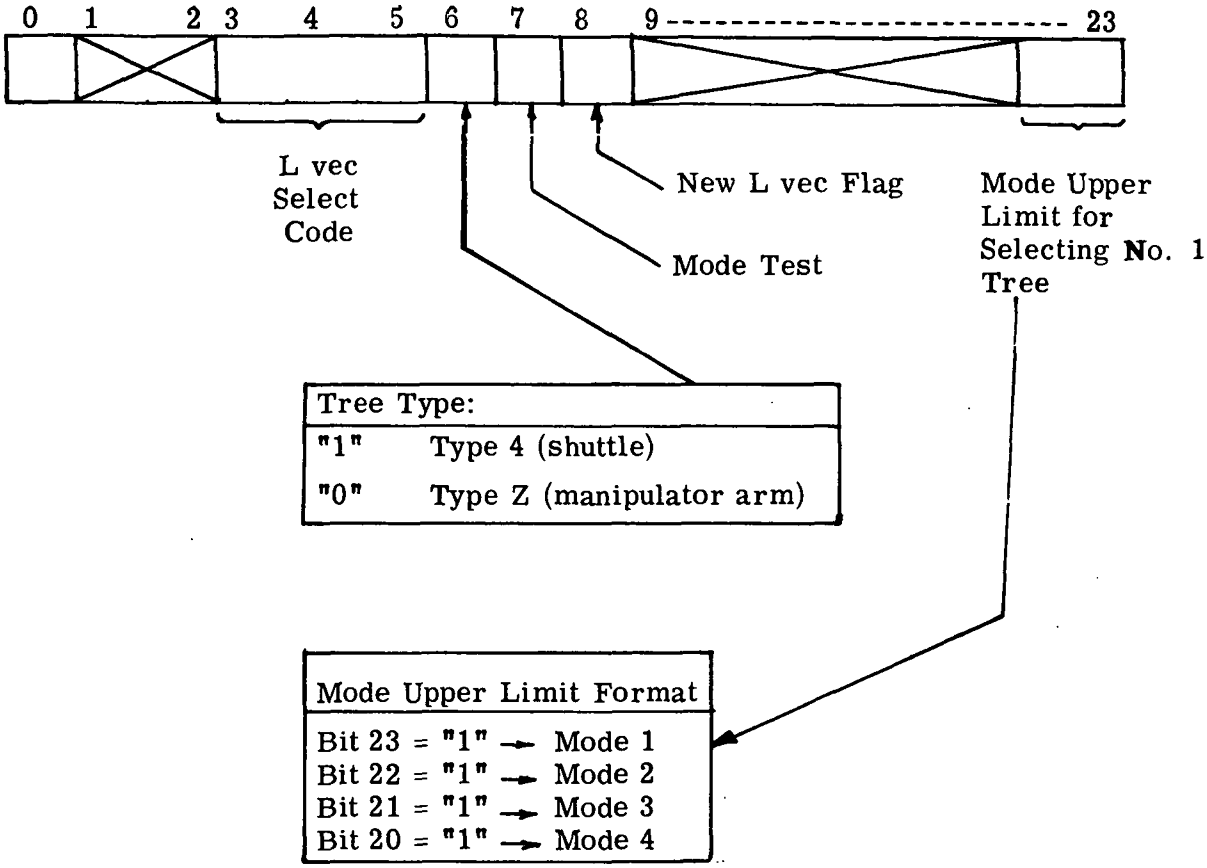
The tracing of separating plane decision trees is similar to that used in operating software delivered under Contract No. NAS 9-11065 and documented in the literature previously mentioned. Certain additions have been made to the tracing algorithm to accommodate (1) changing L vectors for separating plane testing and for insertion in the POT, (2)) multiple clusters at tree leaves, and (3) special priority required for the end effector and acquisition bar (see Figures 16, 17 and 18). Capability is also provided to trace subtrees of a given tree.

The Node Indicator (see Figures 16 and 17) serves to control the three passes through a given mode, as before. In addition, the Node Indicator can specify a new L vector for separating plane testing. When such a new L vector is indicated all further separating plane testing, within the subtree that the current node heads, will use the new L vector unless other new L vectors are specified in nodes within the subtree. A pushdown stack is used to (1) accumulate references to L vectors as the active separating plane L vector is successively modified on the first pass through modes whose indicators call for new L vectors, and (2) reinstall the most previous L vector for plane testing on the third pass through any node whose indicator previously called for a new L vector. This L vector selection capability permits mixing of coordinate systems within a tree.

The Tree Node Son may be a VCU cluster instruction or reference to another node, as before. It also serves three other purposes. A new L vector for the POT may be specified, where it is known that clusters referenced by the son are associated with an L vector (or coordinate system) different from the default POT L vector established by the most recent PPL entry L vector initialization. Multiple clusters may be referenced by the son, wherever it is convenient to provide a fixed cluster list as a leaf of a tree. Finally, special priority for the end effector and acquisition bar may be specified.

The special priority for end effector and bar receptacle first traces a tree of fixed separating planes to order the clusters within the end effector. Then the bar receptacle cluster is sorted upward from the bottom of the end effector list on the basis of observer position with respect to the special dynamic separating planes that separate the three clusters of the end effector claw from the acquisition bar. The process is depicted in Figure 19 under the X1 tree description.

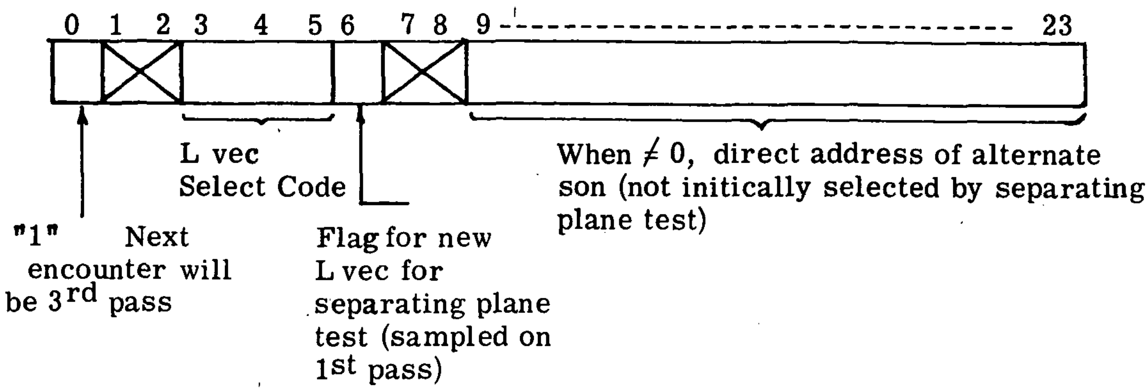
Model layout, coordinate systems, and location of separating planes are shown in Figure 20. Note that separating planes 11, 12, D1, D2, and D3 are dynamically computed; that is, normals and D values change from frame to frame. Planes 11 and 12 are formed in the Visual Scene Program by the routine that computes L and P vectors, DYCALC, and are each positioned to bisect the angle between the adjacent manipulator arm sections. Planes D1 through D3 are computed within the Dynamic Collision Program in the Sigma 5 computer and are transmitted to the R520 once per frame. L vectors nomenclature is covered in Table V. Camera or station point positioning within model space is covered in Table VI.



Separating Plane Tree Node

Father Pointer
Trees Son (see son format)
Indicator (see indicator format)
False Son (see son format)
NX NY NZ Separating plan normal components, scale = BO D = P·N, scale = B11

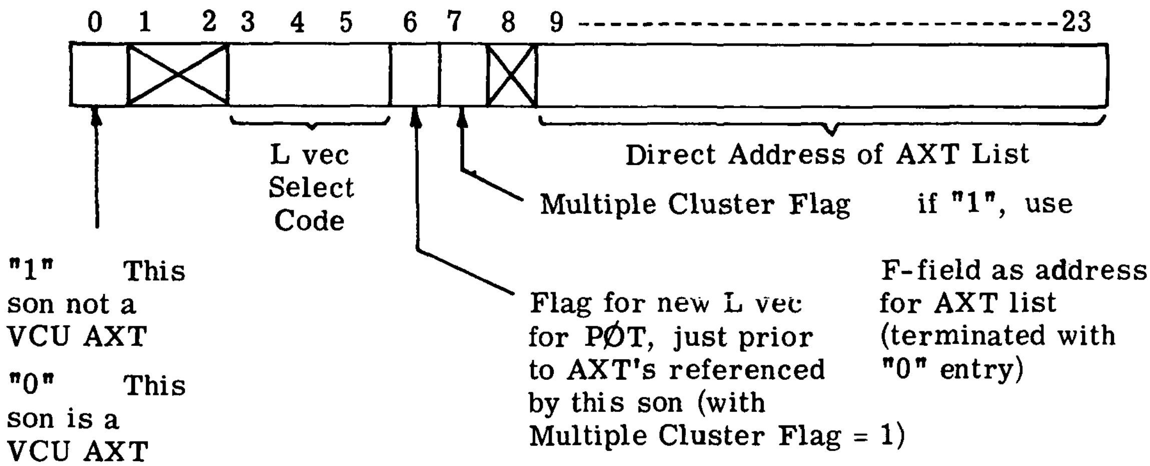
Figure 16. Node Indicator



Summary of indicator control states:

Indicator Fields	Interpretation
S = F = 0	Next encounter will be 1st pass through node. Test for new separating plane L vector, then perform separating test. Refult of test loads alternate son address into F - field for 2nd pass. Winning son address passed in operating register.
S = 0 β F ≠ 0	Next encounter will be 2nd pass through node. Use F-field to determine address of alternate son. Set S = 1 for 3rd pass.
S = 1	Next encounter will be 3rd and last pass through node. Check this indicator address against root node indicator address. If root node, done. Else father pointer is address of indicator for next node. In any case, set S = 0 for 1st pass.

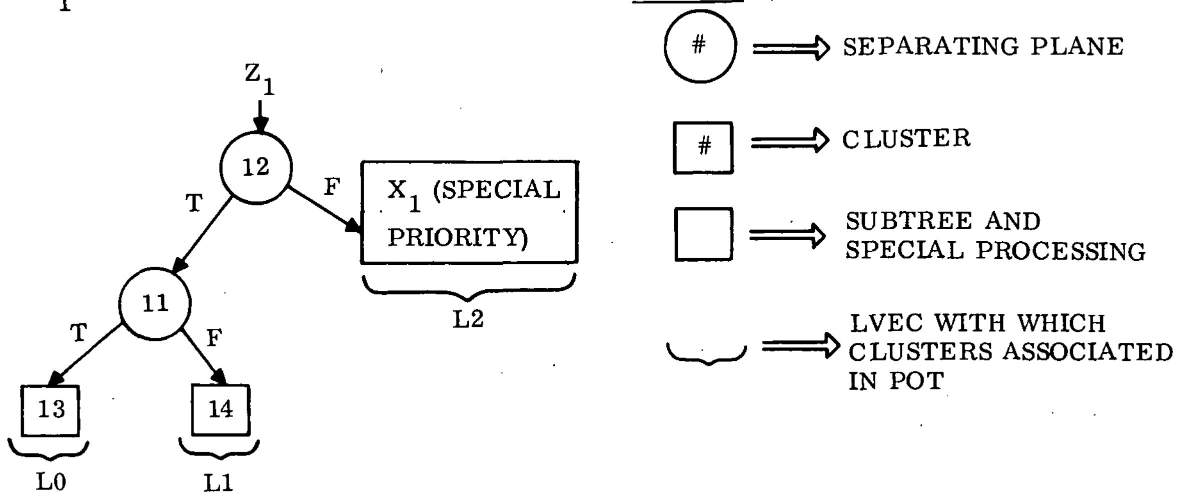
Figure 17. Node Indicator



Summary of son control states:	
Son Fields	Interpretation
S = 0	This son is VCU AXT.
S ≠ 0 X = F = 0	This son specifies branchs to special priority for end effector and payload aquisition bar.
S ≠ 0	If Bit 7 only = 1, use F-field as address of AXT bit terminated by zero entry. If Bit 6 also = 1, use L vec Select Code (Bits 3-5) to obtain VCU L vec instructions for insertion in PØT before any subsequent cluster instructions referenced by Multiple Cluster List Pointer in F-field

Figure 18. Tree Node Son

Z_1: ARM WITHOUT PAYLOAD



Z_2: ARM WITH PAYLOAD

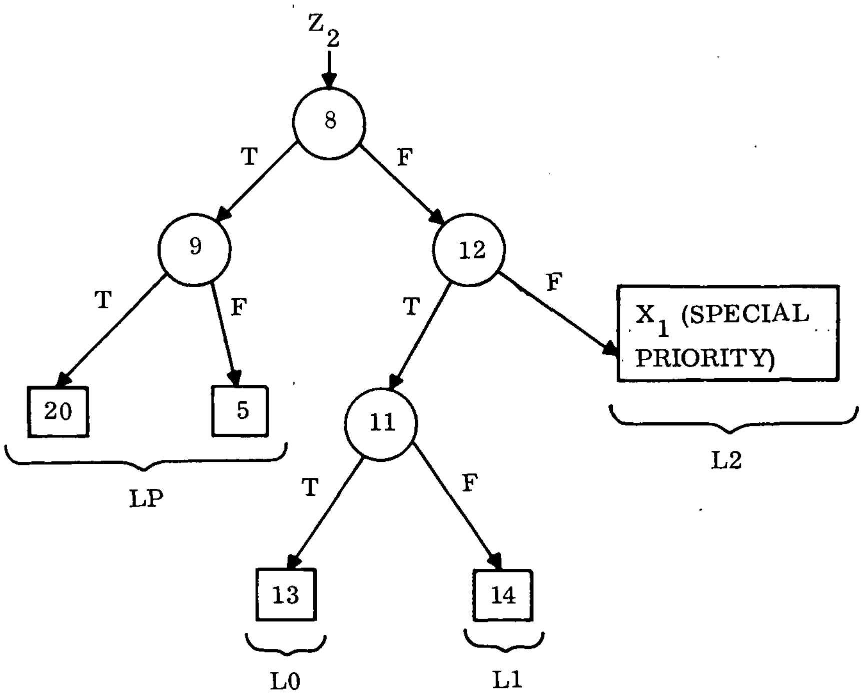


Figure 19. Priority Trees

(Page 1 of 4)

Y_1: SHUTTLE WITH PAYLOAD

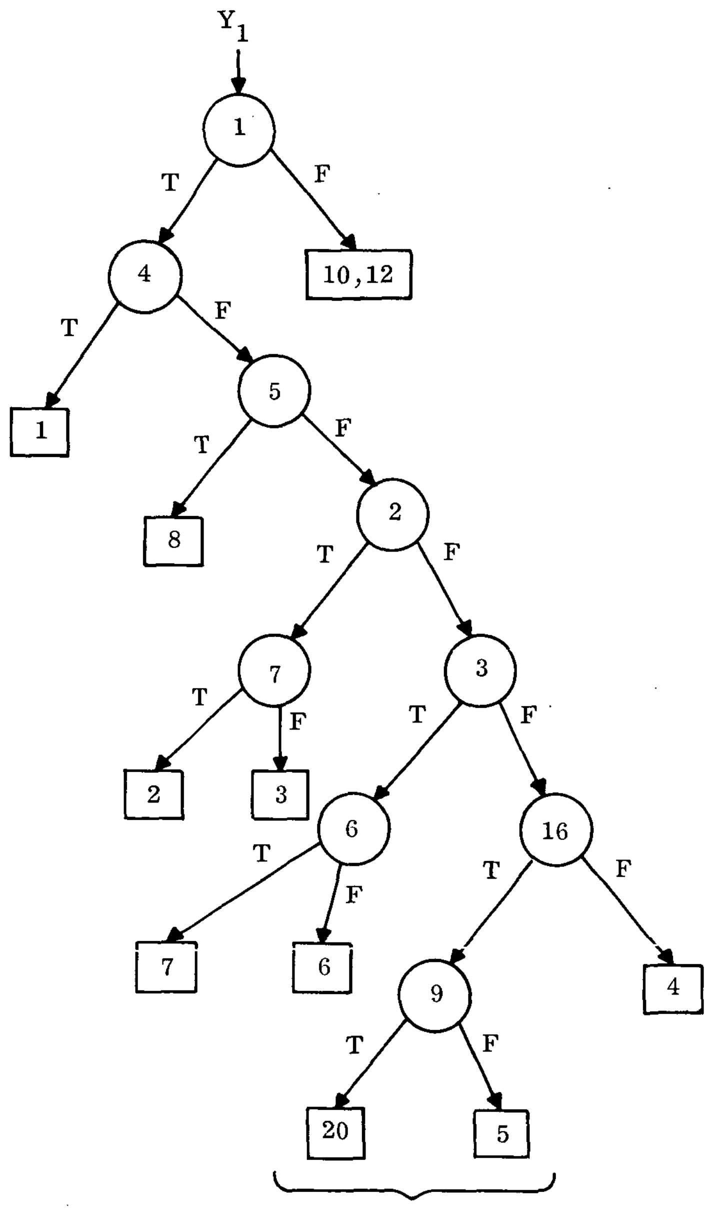


Figure 19. Priority Trees

(Page 2 of 4)

Y_2: SHUTTLE WITHOUT PAYLOAD

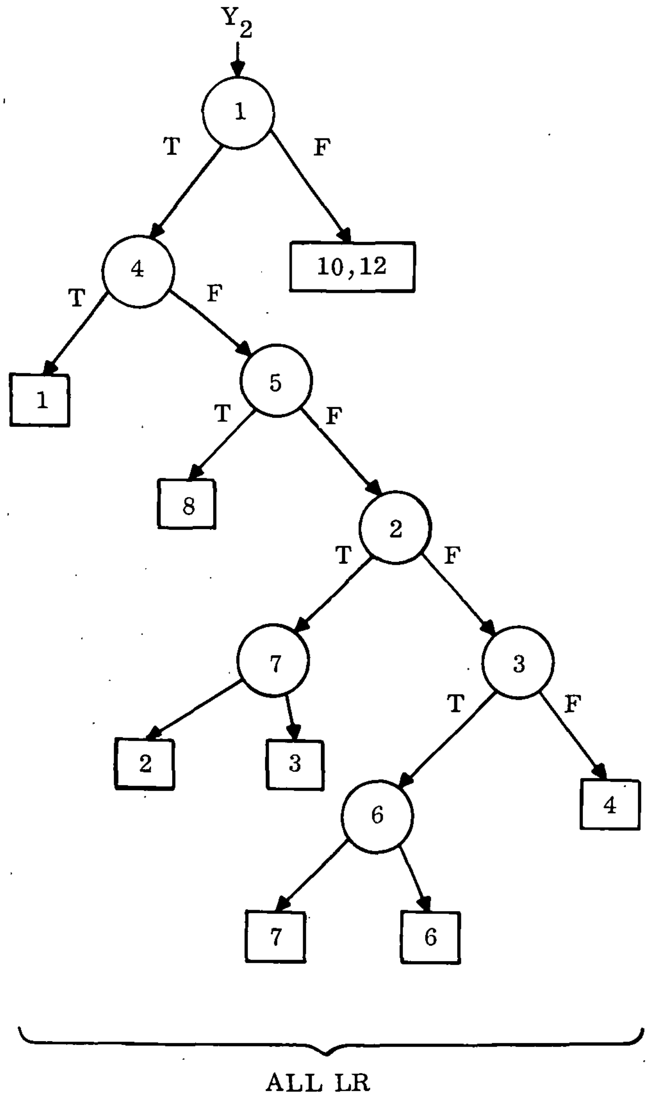
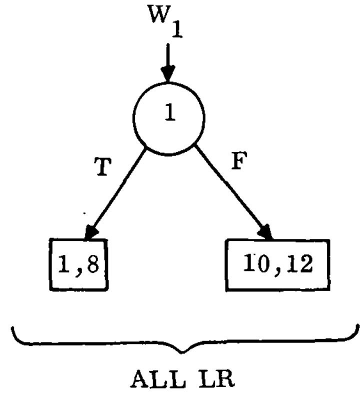


Figure 19. Priority Trees

(Page 3 of 4)

W_1: SHUTTLE FRONT AND REAR



X_1: END EFFECTOR AND ACQUISITION BAR

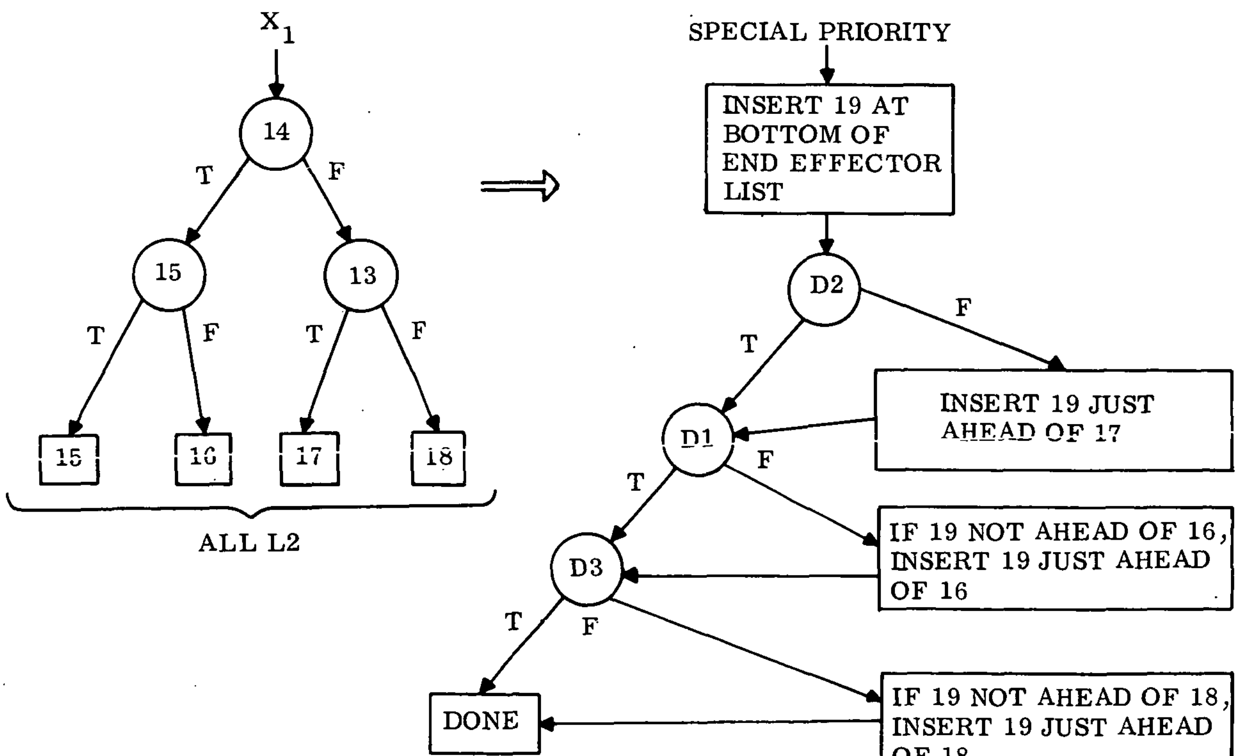


Figure 19. Priority Trees

(Page 4 of 4)

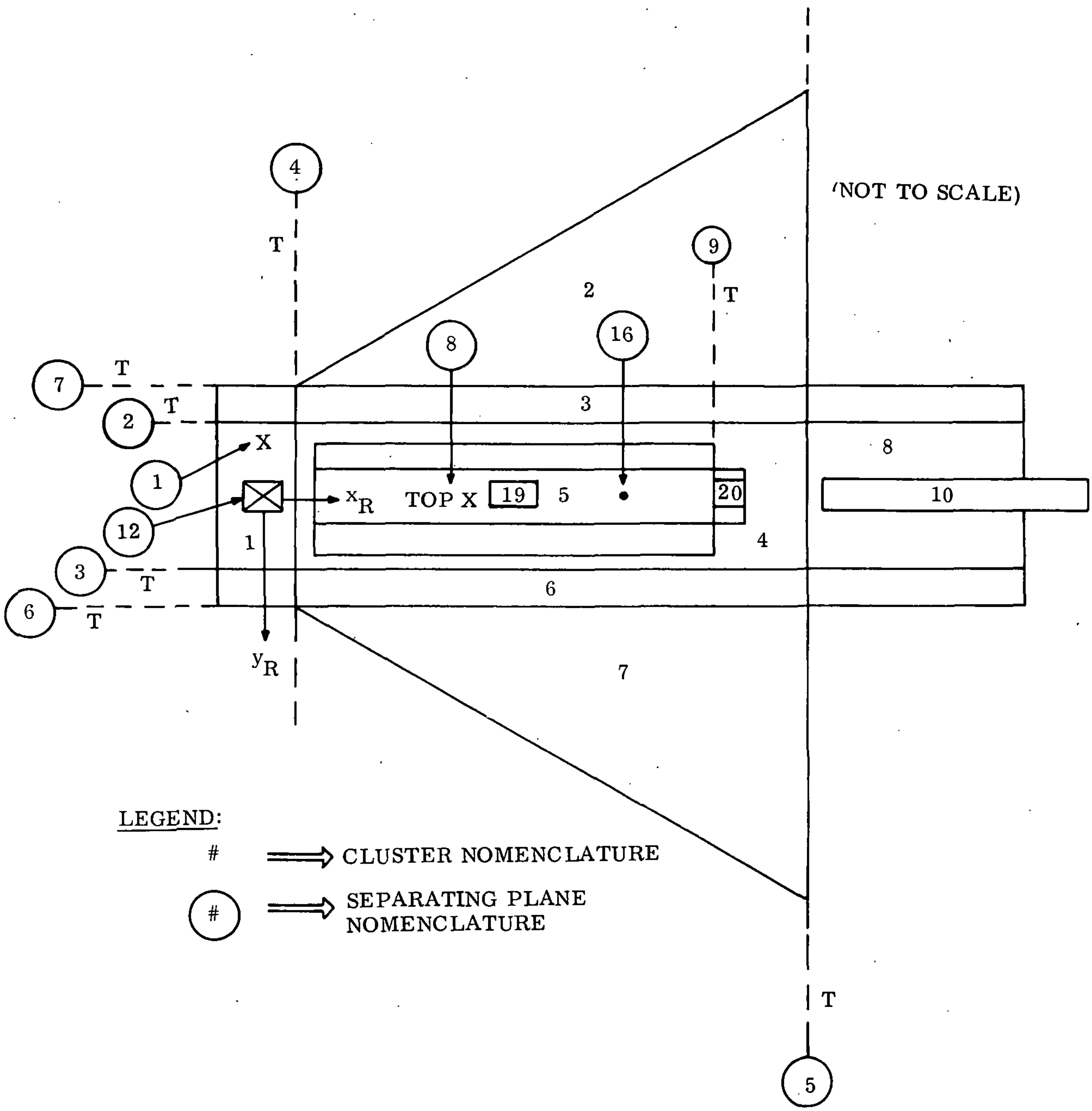


Figure 20. Model and Separation Planes

(Page 1 of 2)

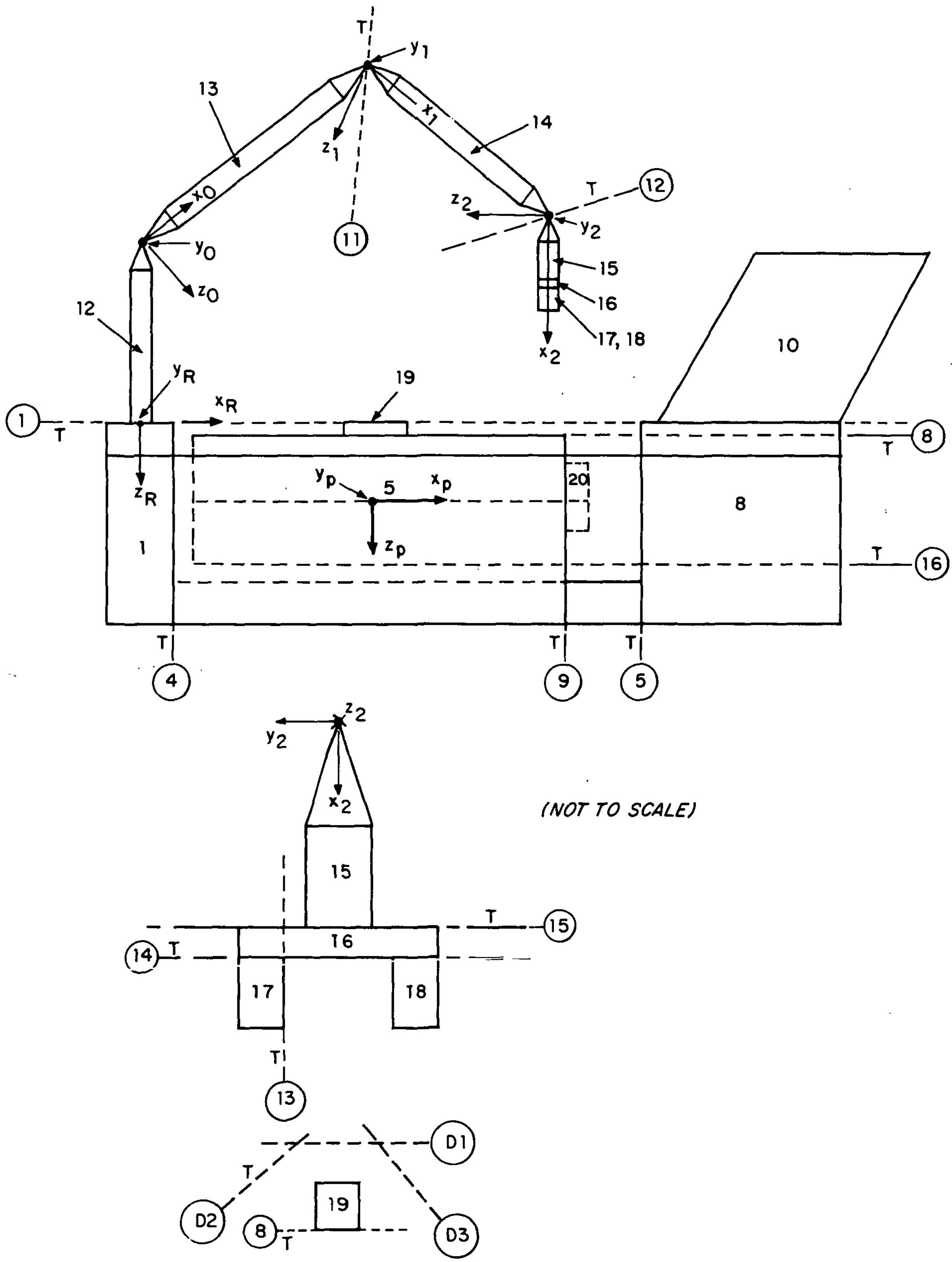


Figure 20. Model and Separation Planes

(Page 2 of 2)

The Permanent Priority Lists are shown in Table VII. Only station points F, G, and H utilize mode considerations for selecting alternative trees. In Table VIII the system in which each of the separating planes is tested is indicated. Figure 19 documents the trees used in various parts of the priority processing. The Z1 tree is a proper subtree of the Z2 tree. The X1 tree is traced within the special priority program to form the end effector cluster list into which cluster 19 is sorted.

The flowchart and listing for MAINPRIOR is given in Figure 21.

2. Visual Scene Program Output to VCU

After dynamics and priority processing within the Visual Scene Program, resulting VCU data and instructions are output to the VCU by the VCU XMT routine. L and P vectors are transmitted first, then the Priority Output Table, and then the dynamic vertices associated with the telescoping arm and end effector moveable finger. If Sense Switch 4 is set, control is passed to the Octal Utility Program (DUP). Otherwise control is returned to Source Input.

The flowchart and listing for VCU XMT is presented in Figure 22.

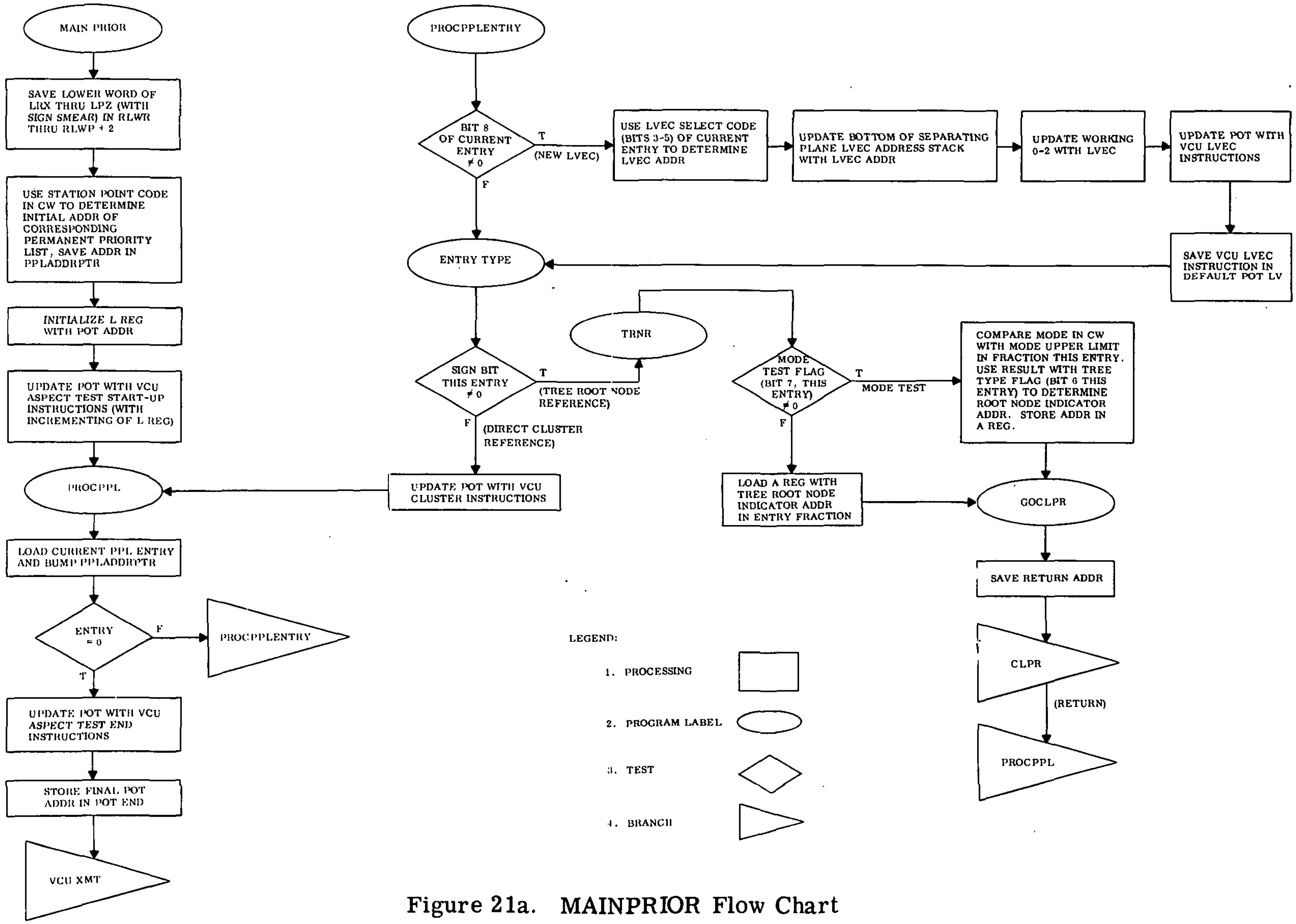


Figure 21a. MAINPRIOR Flow Chart

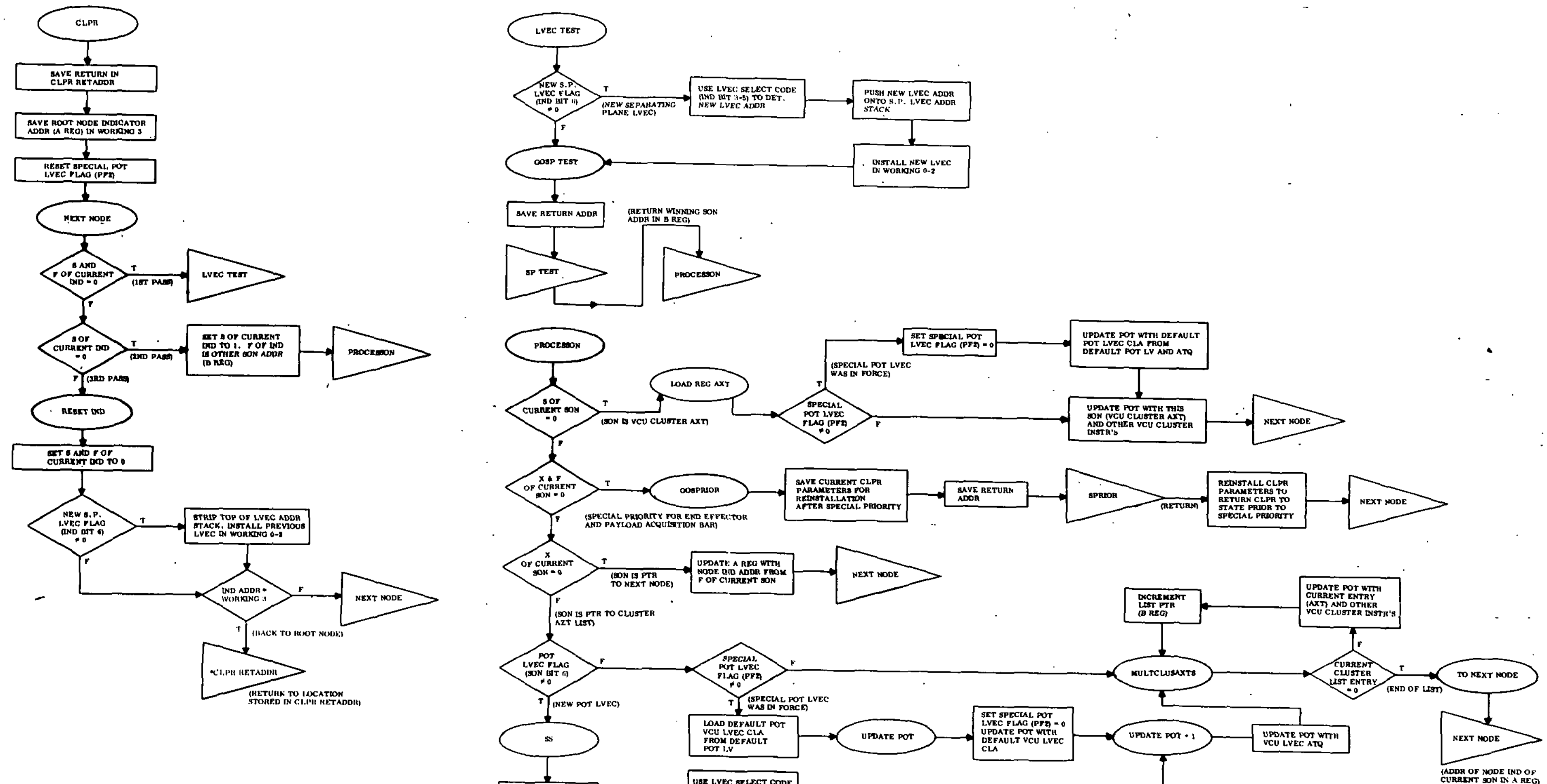


Figure 21b. MAINPRIOR Flow Chart

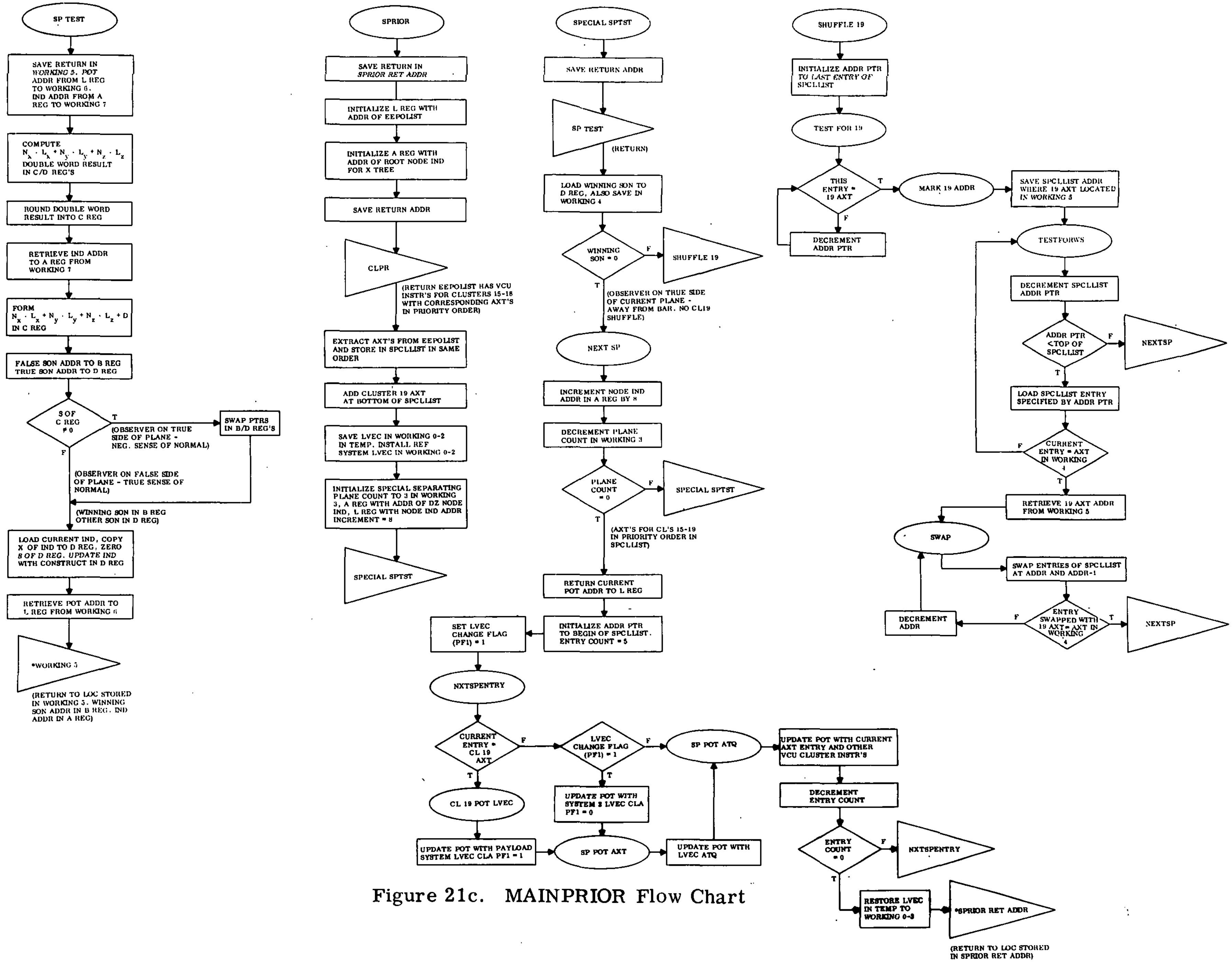


Figure 21c. MAINPRIOR Flow Chart

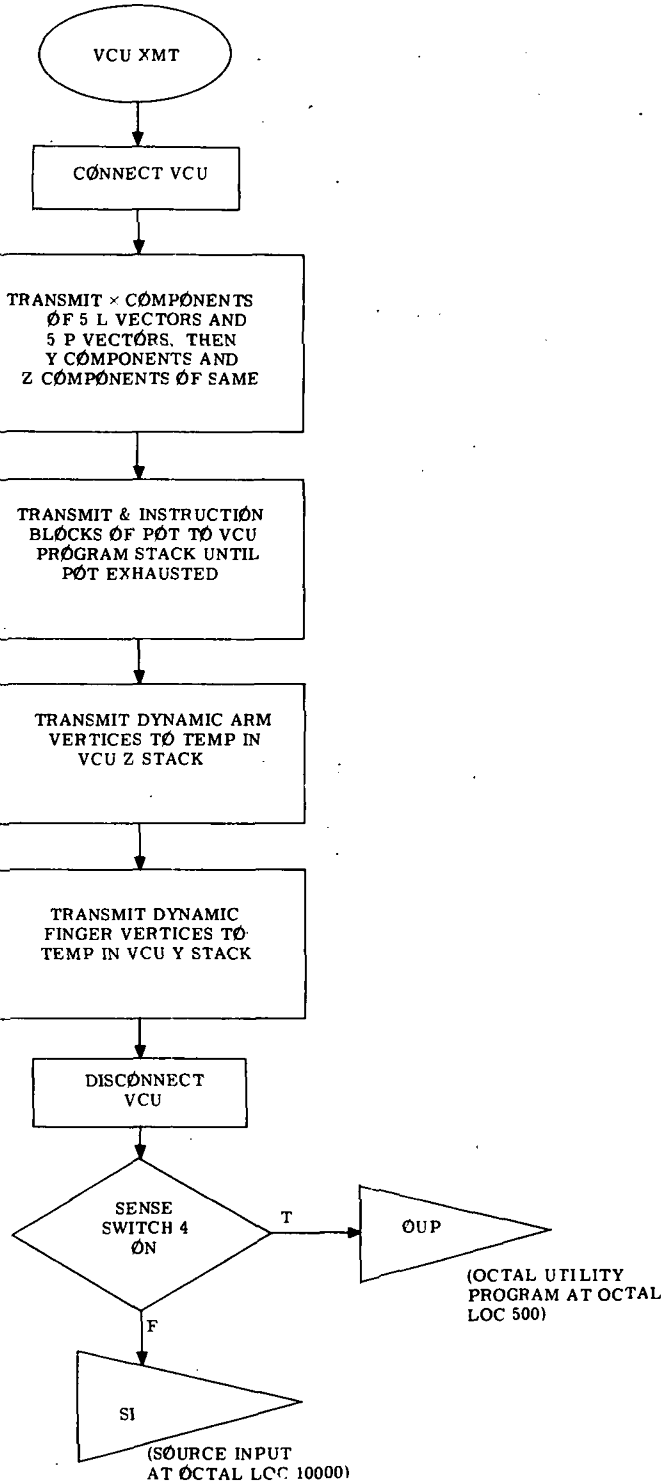


Figure 22. VCU XMT Flow Chart

TABLE VII

PERMANENT PRIORITY LISTS
(numbers denote cluster nomenclature, letters denote separating plane trees)

Station Point	List	Remarks
A, E	12, 1, Z, 4, 3, 2, 6, 7, 10, 8, 21	
B	3, Z, 2, 4, 6, 7, W_1, 21	$Z = Z_2$ All Modes
C	10, 8, Z, 12, 4, 3, 2, 6, 7, 1, 21	
D	6, Z, 7, 4, 3, 2, W_1, 21	
F	Z, Y, 21	Mode 1 $Z=Z_1, Y=Y_1$
G	Z, 4, 21	Mode 2, 3 $Z=Z_2, Y=Y_2$
H	21, Z, Y	Mode 1, 2 $Z=Z_1$, $Y=Y_1$ Mode 3 $Z=Z_2$, $Y=Y_2$
I	12, 1, 21, Z, 4, 3, 2, 6, 7, 10, 8	$Z=Z_2$ All Modes

TABLE VIII.

L VECTORS FOR SEPARATING PLANE TESTS

Plane	L Vector
1	LR
2	
3	
4	
5	
6	
7	LR
8	LP
9	LP
11	L1
12	L2
13	
14	
15	L2
16	LR
D1	
D2	
D3	LR

VI. OPERATIONAL MODIFICATIONS

The characteristics of the implemented simulation that can be modified include the geometry and relative locations of physical components, arm telescoping and prog open/close rates, camera tilt/pan and zoom rates, camera location and payload acquisition tolerances.

A. COMPONENT GEOMETRY AND LOCATION

The geometry and relative locations of the physical components of the system must have a one-to-one correspondence in the DYNAMIC COLLISION and VISUAL SCENE programs. Hence, if these parameters are to be changed, the numerical models must be modified in both programs. Some of the physical characteristics that can be changed and the parameters that must be considered are listed below.

CHARACTERISTIC TO BE MODIFIED	VISUAL SCENE PROGRAM	DYNAMIC COLLISION PROGRAM
1. Payload		
a) location	Center of Payload Position Vector See Table III	Payload Center Coordinates PLCTRX, PLCTRY, PLCTRZ listing lines 187, 188, 189
b) dimensions	Requires off-line modeling program.	Payload Dimensions LP, DPL, DDP listing lines 183, 184, 185.
2. Central Assembly Element (CAE)		
a) location and b) dimensions	Requires off-line modeling program.	CENTRAL ASMB ELE VERTICES Table CAE, listing line 129 DOCKING PORT CENTER Table PORCTRX, PORCTRY, PORCTRZ listing lines 154-156

3. Manipulator Arm Dimensions and End Effector Dimensions	Requires off-line modeling program.	Vertical Arm - STANDZ listing line 181 Arm 0 - LA0 listing line 176 Arm 1 - LA1 listing line 177 Arm 2 - LA2 listing line 178 EFF HEX 1 - L1, W1, H1 listing lines 165, 166, 167 EFF HEX 2 - L2, W2, H2 listing and EFF HEX 3 lines 169, 170, 171
4. Cargo Bay Dimensions	Requires off-line modeling program.	CARGO BAY VERTICES Table CARBARY listing line 103
5. Bay Receptacle Dimensions and Location on Payload	Requires off-line modeling program.	BAR EFFECTOR RECEPTACLE Table RECPT listing line 72 Bar EFFECTOR RECEPTACLE CENTER XBCTR, YBCTR, ZBCTR and BARBOT listing lines 97-100

B. RATE MODIFICATIONS

Three rate type parameters can be changed. These rates and their locations in the two programs are listed below. The rates employed in the simulation are frame rates; ie., ft per 1/50 second.

RATE TO BE MODIFIED	VISUAL SCENE PROGRAM	DYNAMIC COLLISION PROGRAM
1. Telescoping Arm	Arm telescope rate for Arm Vertices and TR Vector. Table III	TELRATE listing line 194
2. Prong Open/Close	Finger Motion Rate. Table IV	PRGRATE listing line 193
3. Camera Pan/Tilt and Zoom	Camera Pitch/Pan Rate and Camera Zoom Rate Table III	Does not appear in this program.

C. CAMERA POSITION

The camera position and initial angle data locations for the VISUAL SCENE PROGRAM are listed in Table III. No changes are required in the DYNAMIC COLLISION PROGRAM.

D. ACQUISITION TOLERANCES

The Acquisition testing function is performed by the DYNAMIC COLLISION PROGRAM. The tolerances that can be varied are used in the Acquisition Final test (AQFTST). They are found in the BAR ACQUISITION LIMITS Table, lines 198-201 of the listing. They are:

LABEL	FUNCTION
FRONT	An XR coordinate +0.5 feet from the "front end" of the bar receptacle.
REAR	An XR coordinate -0.5 feet from the "rear end" of the bar receptacle. The geometric center of the bottom face, face d, of EFF HEX 1 must be between these two coordinates for a successful acquisition to occur.
TOPB	The ZR coordinate of the top surface, face c, of the bar receptacle. For acsuisition testing the top of the bar receptacle is assumed to be located 1/16 of a foot in the -Z direction, above, the top surface of the VISUAL SCENE PROGRAM and DYNAMIC COLLISION PRO GRAM numerical modesl of the bar receptacle.
PRGSPC	This is the prong spacing tolerance used in the acquisition test. Physically it represents the distance between the center line of EFF HEX 1 and EFF HEX 3 (the prong) when the prong is in the fully closed position.

E. PRONG AND FIXED ARM LIMITS

The movable finger of the end effector (prong) and the telescoping arm have motion limits that can be changed by the following parameters:

	VISUAL SCENE PROGRAM	DYNAMIC COLLISION PROGRAM
1. Prong Motion Limits		
a) closed limit	No change required	PRGFC - listing line 207
b) fully open limit	Initial Vertex Positions for Moveable Finger Table III	PRGFO - listing line 208

2. Telescoping Arm Limits		
a) Upper Limit	No change required	STANUL - listing line 205
b) Lower Limit	Initial Vertex Positions for Top of Telescoping Arm and Initial TR Vector Components TRZL only See Table III	STANLL - listing line 206

The detection of limits for both the moving finger and the telescoping arm is performed in the DYNAMIC COLLISION program. The initial fully open limit and initial lower telescoping arm position limit must be modified in the VISUAL SCENE program to insure that these lower limits agree in both programs when the simulation is initialized to MODE 1.

VII. EXTENSION OF THE OPERATIONAL CAPABILITY

The implemented simulation is unidirectional. The three phases of operation, Acquisition, Extraction and Maneuvering, can only be performed in this sequence. Hence, some of the operational tasks that a remote manipulator system would be called upon to perform cannot be simulated. These tasks include the ability to acquire a free flying payload and insert it into the cargo bay and the ability to position a payload in space and release the manipulator end effector from its "acquired" state.

The addition of the following operational characteristics to the simulation will provide the ability to simulate most of the tasks that are required of a remote manipulator. That is:

1. Capability to acquire a free flying payload that is limit cycling. Dock the acquired payload with the central assembly element (CAE) and/or insert it in the space shuttle cargo bay.

2. Capability to translate the space shuttle with respect to the CAE and the free flying payload.

3. Capability to release manipulator end effector from an acquired payload that is docked with the CAE, inserted in the shuttle cargo bay, or placed in a free flying condition.

4. Permit the payload to move in a vertical direction (up or down), when in contact with side walls of cargo bay during extraction of payload from cargo bay.

5. Add four (4) additional camera positions to the simulation; two of which are to be located on the payload, one on CAE, (which increases the number of camera positions on CAE to a total of two) and one on the shuttle wing.

The addition of these operational characteristics can be accomplished by applying the mathematical solutions and algorithms developed on this contract to the increased number of physical encounters and relationships between system components that will occur.

APPENDIX

COORDINATE TRANSFORMATION MATHEMATICS

In order to perform the tests required to detect collisions, acquisition and docking in the DYNAMIC COLLISION PROGRAM the vertices of the numerical models, which describe the physical components of the manipulator-shuttle system, must be expressed in the same coordinate system. The three arms, ARMO 0, ARM 1 and ARM 2, of the manipulator are each defined in their own coordinate systems. The coordinate system for ARM 2, System-Two coordinates, has its origin at the wrist joint, the Arm 1 coordinate system origin is at the elbow joint and the Arm 0 coordinate system origin is at the shoulder joint. There are six angles used to maneuver the manipulator arm. The shoulder joint has heading (HO) and pitch (PO), the elbow joint has only pitch (P1) and the wrist joint (end effector) has heading (H2), pitch (P2) and roll (R2). The space shuttle, and central assembly element numerical models vertices are all defined in the Reference Coordinate System, XR, YR, ZR, which has its origin at the foot of the fixed vertical arm, the arm labeled STAND in Figure 5. In the Acquisition Phase, MODE 1, collision and acquisition testing of the end effector with respect to the top surface of the payload and the bar receptacle of the payload is performed. The payload top surface and bar receptacle are expressed in Reference System Coordinates in MODE 1 and thus the vertices of EFF HEX 1, EFF HEX 2, and EFF HEX 3, must be transformed to Reference System Coordinates. During the Extraction and Maneuvering Phases, MODE 2 and MODE 3, collision testing of the payload with respect to the shuttle cargo bay and the CAE is performed. Thus the vertices of the payload numerical model must be expressed in Reference System Coordinates. In Section IV it was shown how effector hexahedron vertices are found by multiplying the transformed unit vectors by dimensional constants, and it was stated that the payload vertices are found by a similar process. The following equations are employed to obtain the Reference System Coordinates of the System-Two unit vectors.

The transformation equation for obtaining the Reference System Coordinates (XR, YR, ZR) of the System-Two vector (X2, Y2, Z2) is:

$$\begin{pmatrix} XR \\ YR \\ ZR \end{pmatrix} = R(-HO)\ R\ (-PO)\ R\ (-P1)\ R\ (-H2)\ R(-P2)\ R\ (-R2) \begin{pmatrix} X2 \\ Y2 \\ Z2 \end{pmatrix}$$

where:

a rotational through the roll angle - RX is:

$$R(-RX) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & COS\ RX & -SIN\ RX \\ 0 & SIN\ RX & COS\ RX \end{pmatrix}$$

a rotation through the pitch angle -PX is:

$$R(-PX) = \begin{pmatrix} COS\ PX & 0 & SIN\ PX \\ 0 & 1 & 0 \\ -SIN\ PX & 0 & COS\ PX \end{pmatrix}$$

a rotation through the heading angle -HX is:

$$R(-HX) = \begin{pmatrix} COS\ HX & 0 & -SIN\ HX \\ 0 & 1 & 0 \\ SIN\ HX & 0 & COS\ HX \end{pmatrix}$$

To find the Reference System Coordinates of the System Two unit vectors each is operated on by the above rotation sequence i. e., this unit vectors:

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{are transformed.}$$

The Reference System Coordinates of manipulator Arms 1 and 2 are found by the following transformation equations:

for Arm 1:

$$\begin{pmatrix} LA1XR \\ LA1YR \\ LA1ZR \end{pmatrix} = R(-HO)\ R(-PO)\ R(-P1) \begin{pmatrix} LA1 \\ 0 \\ 0 \end{pmatrix}$$

for Arm 0:

$$\begin{pmatrix} LA0XR \\ LA0YR \\ LA0ZR \end{pmatrix} = R(-H0)\ R(-P0) \begin{pmatrix} LA0 \\ 0 \\ 0 \end{pmatrix}$$