To the Graduate Council :

I am submitting herewith a thesis written by Amin AL-Halabi entitled "Kinematic Design and Simulation of Long Reach Manipulators". I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

Dr. Rajiv V. Dubey, Major Professor

We have read this thesis
and recommend its acceptance :

Accepted for the Council :

Associate Vice Chancellor and
Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under the rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Amin Ash Halabi

Date 7/28/95

KINEMATIC DESIGN AND SIMULATION

OF LONG REACH MANIPULATORS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Amin AL-Halabi

August 1995

Acknowledgments

I would like to express my appreciation to all those who helped in the research and preparation of this thesis. My special thanks for Dr. Rajiv Dubey for being my major professor and advisor not only in my graduate study but also in my undergraduate. He was a major help in construction and improvement of this thesis. I would also like to thank Dr. Bill Hamel for his help and encouragement. I would like to thank Dr. Kawiecki for serving on my committee. Tan Fung Chan and Amrish Shah have probably been the main contributors to the success of this thesis. Steve Everett also joined the team at a later time and has contributed a great deal of work. I would like to thank them for their help, time and support. I would also like to thank all students in my research group for their continued support during this period.

I would like to express my appreciation for all the people in the departmental office including: Linda, Janine, Jackie, Marva, Betty and others for assisting me with all my needs and quality time we have been through.

Finally, I would like to express my thanks to all my friends and family for their support, especially to my brothers Taha and Mohammed who supported me through my undergraduate study, also to my parents whom I lost during my graduate study, which I consider the biggest loss in my life.

Abstract

The Long Reach Manipulator (LRM) acts as a platform for the various waste cleanup tools and dexterous manipulator required for the Tank Waste Retrieval system. The LRM will be used as a positioning and orienting device for these tools and the dexterous manipulator. The kinematic design of the LRM plays an important role in its performance in covering the complete workspace inside the tank, avoiding collisions and joint limits, and orienting the tool plate. Joints included beyond the number necessary for positioning, referred to as redundant joints, will be required for the LRM to have collision avoidance, joint limit avoidance, complete workspace coverage and tool plate orientation capabilities.

The emphasis of this thesis is on the investigation and choice of an optimum kinematic design of the long reach manipulator, specifically, the determination of link lengths and joint arrangements that would be easy to manufacture and reach all the waste tank volume. Comparisons of IGRIP simulations, analytical solutions and graphical solutions are used to determine the optimum long reach manipulator kinematic design.

Control schemes for mining operation are demonstrated on the old testbed. This testbed manipulator consists of Spar RMS 2500 as a Gross Positioning Manipulator (GPM) and Schilling TITAN II as a Dexterous Manipulator (DM). The purpose of the software is to demonstrate future redundancy resolution control strategies to be applied to the finalized design of LRM and DM.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robot manipulators are devices used to perform tasks that are unsuitable for a human being. Tasks can be inappropriate for human beings because they are hazardous or time consuming, or because of the limited power and accuracy of humans. As an example, robot manipulators are used in nuclear waste cleanup because of the high risk of radiation.

One of the most urgent tasks within the Department of Energy is the waste retrieval process to be performed in underground single-shell waste storage tanks. These single-shell storage tanks are full of gradually leaking radioactive waste material that is very difficult to monitor. Current U.S. Environmental Protection Agency (EPA) regulations require double-shell tanks for waste storage, but many Department of Energy sites are out of compliance with current EPA regulations[1].

The use of long reach manipulators is considered the most efficient approach for the waste retrieval process in large tanks, making the development of such a tank waste retrieval manipulator one of the most significant robotics projects. A tank waste retrieval manipulator system consist of a long reach manipulator (LRM), including a vertical deployment mast and a short reach dexterous manipulator. The long reach manipulator is considered to be the main positioning tool in the tank, while the dexterous manipulator is dedicated for final and accurate cleaning tasks.

1.1 Background

1.1.1 Waste Tank

As shown in figure 1.1 a typical tank is composed of a cylindrical volume 32.5 feet high with a 37.5 feet radius and a domelike top 12 feet high. The 12 foot dome is the available volume for the long reach manipulator to enter the tank initially. The access hole is near the center of the tank and may range from 24" to 42" inch in diameter. The waste tank is buried 8 feet below ground level. The two foot clearance shown in the figure is the distance between ground level and the bottom of the LRM support tower.

1.1.2 Long Reach Manipulator (LRM)

The long reach manipulator consists of a column that travels vertically 52 feet and carries the linkage to reach horizontally the maximum diameter of the tank. The column can be a rigid cylinder or a telescopic prismatic joint. One of the advantages of using a telescopic prismatic joint that it requires a smaller tower height. At the end of the column there are three different arrangements proposed for the joints. The first two arrangements are FET (Folded-Entry-Type) and TSET (Telescopic-Sequential-Entry-Type) manipulators, using the same configuration of REPRPE (Roll-Extend-Pitch-Roll-Pitch-Extend). The third arrangement is SET (Sequential-Entry-Type) manipulator which uses the configuration of REPRPP (Roll-Extend-Pitch-Roll-Pitch-Pitch). While the long-reach manipulator is to perform 90 to 95 % of the tank cleanup using an end effector tool, the long reach manipulator should be able to reach the entire tank volume. The final detailed cleanup is to be performed by a dexterous manipulator placed at the end of the long-reach manipulator.

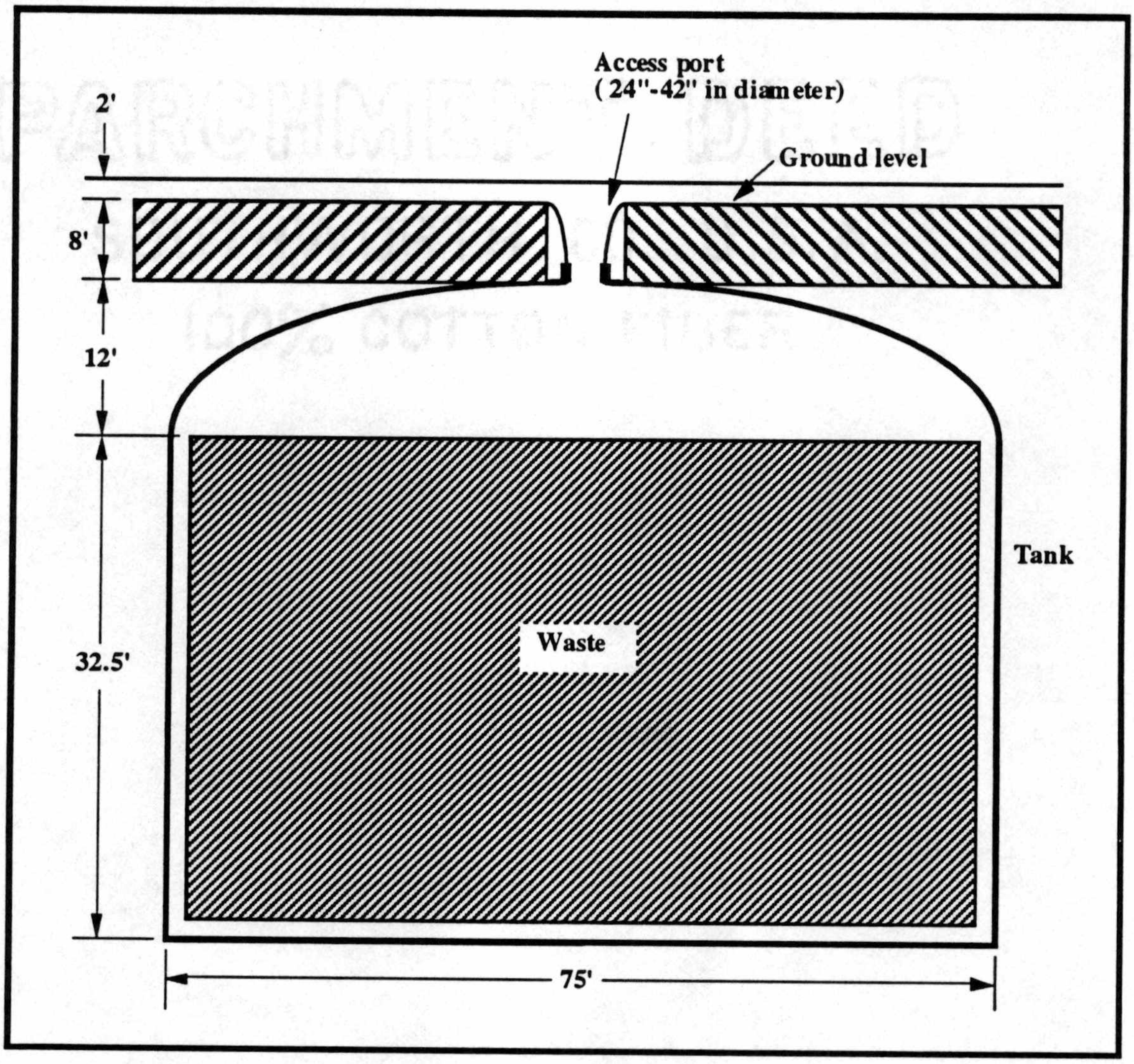


Fig. 1.1. Typical waste tank

1.1.3 Dexterous Manipulator (DM)

The proposed dexterous manipulator slave arm is a modified version of the SDI Atlas 8F manipulator slave arm[7]. The Atlas 8F was designed, built, and delivered in 1990 to the U.S. Navy . Slight modification will be required for the Atlas 8F slave to accommodate the DM requirements for the tank waste retrieval process. Modifications are to increase payload, and to add a force/torque sensor located at the end effector exchange plate. To increase payload the actuators will be redesigned to increase joint torque by 100%. In addition, a commercially available JR3 force/torque sensor will be customized for this application.

1.1.3 Waste and Obstacles Inside The Tank

The contents of the tank are assumed to be a mixture of solid and liquid waste. Initially the solid waste will be loosened and converted to a form which can be removed from the tank with cleanup tools like the sludge sucker. Obstacles inside the tank consist mainly of vertical risers extending from the bottom of the tank to the dome.

1.1.4 Design Requirements for the LRM

When the design of the LRM first began, three types of requirements were imposed: functional, kinematic, and performance. The functional requirements provided guidelines on modes of operation (robotic vs. telerobotic), dynamic and static loading on the arm and tank structure, failure recovery, and versatility. Kinematic requirements included workspace coverage, deployment, and degrees of freedom. Issues such as position accuracy, repeatability, no-load and full-load performance, and static and dynamic performance characteristics were dictated by the performance requirements.

1.2 Thesis Objective

The main objective of this thesis is to investigate and choose an optimum kinematic design of the long reach manipulator. A proposed Schilling design is simulated and investigated along with a UT modified design. The main concern is to determine the link lengths and joint arrangements that would be easy to manufacture and allow the end effector to reach all the waste tank volume. Some other requirements such as deployment without digging, avoiding obstacles and maintaining certain end effector orientations are also taken into consideration. Comparisons of IGRIP simulations of different designs are to be used to determine the optimum long reach manipulator design. Analytical and graphical solutions are also to be used mainly for the end effector orientation analysis. Control schemes for mining operation will be demonstrated on the old testbed. The old testbed consists of Spar RMS 2500 acting as a long reach manipulator and a Schilling Titan II acting as a dexterous manipulator[4]. Several approaches to resolve redundancy of the LRM and DM system are presented.

1.3 Thesis Organization

Chapter 2 of the thesis gives a background on original requirements and constraints for the long reach manipulator. Chapter 3 covers the ORNL analysis of the original proposed configurations of the long reach manipulator. Chapter 4 discusses the design of LRM proposed by Schilling and UT modified design. This chapter also includes the simulation for both designs. Comparisons based on simulations and the graphical solutions are also included. Chapter 5 outlines the final design choice based on new end effector orientation requirements that were not taken into consideration initially.

Chapter 6 discusses the implementation of the kinematic control schemes on the old testbed at PNL. This is followed by conclusions and recommendations in chapter 7.

Chapter 2

Long Reach Manipulator Requirements and Constraints

2.1 Introduction

As mentioned in the introduction, design requirements for the LRM are classified into the following three categories[1]

a) Functional requirements

b) Kinematic requirements

c) Performance requirements

The functional requirements of the LRM include the capability for telerobotic operation, that is, suitability for both telerobotic and robotic modes. Also failure recovery and avoidance of loading the tank structure are included in the functional requirements. Kinematic requirements stipulate workspace coverage, port diameter, deployment, number of degrees of freedom, tower height and transportability. Other requirements known as performance requirements specify payload, position accuracy and repeatability, no-load and maximum load velocities, and static and dynamic performance.

Although all requirements were to be taken into consideration for a final design, the main concern of the University of Tennessee was with kinematic requirements. The kinematic requirements are strongly affected by kinematic constraints. These constraints are highly important because they are the main motivation for using LRM as the waste retrieval device. The use of the long reach manipulator, deployed through the 42" diameter access port at the top center of the tank, is considered the most cost effective technique for the waste retrieval process. Also, the crystalline and insoluble contents of the waste make it impossible to use any other method, such as water treatment.

2.2 Long Reach Manipulator Functional Requirements

2.2.1 Telerobotic Operation

Telerobotic operation is the main functional requirement. Telerobotic operation implies that the LRM must be suited for both teleoperation and robotic modes. It is not anticipated that the LRM will have force reflection; but if so, it would be possible to use the LRM independently for tasks requiring force control with a heavy end effector and low positioning accuracy. In the robotic mode, the manipulator should follow a preplanned trajectory or trajectories that have been taught in the teleoperation mode. It is very crucial to recognize that such a trajectory planning approach can excite the manipulator's structural vibration modes, a situation which should be avoided. Also, trajectory planning should include obstacle avoidance capability and approaches to resolve the kinematic redundancy. Various tasks are included in the robotic mode operations such as manipulator deployment, automatic tool changing, and repetitive characterization and retrieval tasks.

2.2.2 Avoidance of Loading the Tank Structure

Loading the tank structure with the system designed for remediation is not permitted, and any contact between the manipulator and the access port should be avoided. Also, two feet clearance should be kept between the support structure for the LRM and the top ground level above the tank.

2.2.3 Failure Recovery

Relative to the high cost of providing additional equipment and an additional port to deploy the repairing equipment, it is essential for the manipulator to recover from failure without any aid. If for any reason the manipulator fails, retrieval of the manipulator should be possible without any risk of contamination or damage.

2.3. Long Reach Manipulator Kinematic Requirements

2.3.1 Workspace

Single-shell storage tanks are typically composed of a cylindrical volume 32.5 feet in height with 75 feet diameter. The bottom of the tank is flat while the roof is a 12 foot high dome. The workspace of the long reach manipulators considered must cover the entire tank volume, assuming that more than 90% of the cleaning tasks are performed without addition of a dexterous manipulator at the end.

2.3.2 Port diameter

The design of the long reach manipulator size is mainly constrained by the size of the existing port diameter; that is, no other way to deploy the manipulator inside the tank can be considered. The high cost of any alternative ways for deployment made it the only feasible one considered. LRM must be designed for deployment through existing ports ranging from 24 to 42 inches in diameter. Because of the fact that few tanks have 24 inch diameter entrance hole, and the deficiencies in performance and other requirements of designs to be deployed through the 24 inch access port, it is decided to consider only the larger 42 inch port.

2.3.3 Deployment

It is believed that the highest waste level is not higher than the cylindrical portion of the tank. For inspection purposes, a complete deployment of the manipulator inside the tank without any waste removal is required. These conditions imply that the LRM is required to be deployed within the dome clearance above the maximum waste level.

2.3.4 Number of Degrees of Freedom

As discussed in the introduction, generally a minimum of three DOF are required to provide general purpose positioning. But in this case of LRM and specific tank requirements, several factors, such as workspace, deployment and end effector requirements may dictate additional DOF. Factors . In addition to these factors, joint limits play a huge factor in the number of DOF required because of the limited range of hydraulic joints.

2.3.5 Tower Height

Tower height required for support of the manipulator is affected by the tank configuration, workspace requirements, deployment approach, and number of degrees of freedom of the manipulator. No limit for the tower height is set, but it is anticipated to minimize the height as much as possible. Unnecessarily high support towers may significantly affect costs and make installation and removal difficult.

2.3.6 Transportability

Mobility of the long reach manipulator system is necessary since a single system will be used for the remediation of multiple tanks. Provisions are necessary for initial LRM installation, and removal and transport to other tank sites or maintenance facilities. The magnitude of this requirement is measured by the degree to which the manipulator can be collapsed to a minimum length, and the maximum height required to lift the manipulator above the tank.

2.4 Performance Requirements

2.4.1 Payload

The long reach manipulator payload requirement is likely to vary between 500 and 1500 lb. Dynamic loading is not predicted at this time and will be dependent on specific end effectors.

2.4.2 Position Accuracy and Repeatability

Position accuracy and repeatability are highly dependent on tasks to be performed. Initially, the removal of 90 to 95% of the waste requires low position accuracy (on the order of several inches). On the other hand, repeatability requirements are more strict (on the order of 1 inch). Automatic end effector changing is the primary incentive for repeatability requirements. As mentioned earlier in the introduction, the long reach manipulator links experience static and dynamic deflections. As a result, to provide reasonable accuracy and repeatability, end point position measurement is required.

The removal of the 5 to 10% remaining waste is anticipated to be performed by the dexterous manipulator attached to the end of the LRM. The position and repeatability requirements become more strict relative to the closeness to the tank or other possible contact surfaces.

2.4.3 No-load and Maximum Load Velocity

No-load and maximum load velocity requirements are likely to be low (on the order of 50 in./s). Smaller industrial robots are usually designed to move at high velocities; if industrial robots are scaled up to have the same link sizes of the LRM, their tip velocities will be on the order of 500 in./s. Several tasks in the remediation process such as changing the end effector require large motions. Although the higher velocity motion would reduce the task time, the time saved comprises only a small portion of the total remediation time. The constraint on the maximum velocity must ensure collision avoidance regardless of the dynamic response of the manipulator due to its flexibility.

2.4.4 Static and Dynamic Performance

The manipulator's weight is considered the main force that causes static deflection. Static deflection and the manipulator's weight have a nonlinear inverse relationship. Since there is no apparent limit for the manipulator's weight, there is no apparent limit for static deflection, but it is necessary to limit the associated vibration amplitude that is likely to occur. Assuming a true end point position can be measured, it is acceptable to have static deflection on the order of 4 to 8 inches.

The fundamental natural frequency of the LRM is related to the static deflection, so a joint servo approach is used that would band limit the control of the manipulator. Since the LRM will not be operated as a dexterous manipulator, it is not necessary for it to have a comparable bandwidth relative to a human operator. Fundamental natural frequency of the order of 1 to 3 Hz will be acceptable. Two modes of operation will be used, teleoperation and pure robotic modes. A combination of these two modes can be used to provide constrained teleoperation or obstacle avoidance.

Damping characteristics of the manipulator are the main concerns when it comes to static and dynamic performance. Most of the research done in the control of flexible manipulators involves either active damping algorithms using the manipulator's actuators and the motion of a dexterous manipulator mounted on the end of the flexible manipulator, or passive damping of the structure. Effective damping criterion of undesired vibration induced by motion of the LRM or end effector loading is the key to the remediation time and performance of some tasks.

2.5 Kinematic Constraints

As mentioned earlier in the introduction of this chapter, the most cost effective technique for the retrieval of the waste from storage tanks is using the existing manhole at the top center of the tanks. The retrieval system will not be allowed to come in contact with the tank walls, dome or access port under any condition. The nominal storage tank dimensions were shown in figure 1.3 in the previous chapter. Vertical heights of the cylindrical volume, the dome, the burial material and the clearance add up to 54.5 feet. So, given nominal dimensions and central access port, the LRM has to have a minimum of 54.5 feet vertical and 37.5 feet horizontal reach.

The retrieval system must be deployed through the access port into the dome clearance height. Initially, it is crucial for the manipulator not to touch the waste inside the tank until certain inspection procedures are done. Thereafter, to avoid unnecessary contamination of the manipulator, only the end effector can come in contact with the waste. Therefore, at the beginning, only the 12 foot dome clearance is available for the LRM to work.

Chapter 3

Analysis of Reference Configurations of LRM

3.1 Introduction

This chapter presents three basic kinematic arrangements that provide different deployment through a constrained access port[5]. The first two arrangements are FET (Folded-Entry-Type) and TSET (Telescopic-Sequential-Entry-Type) manipulators. The former arrangements use the same configuration of REPRPE (Roll-Extend-Pitch-Roll-Pitch-Extend). The third arrangement is SET (Sequential-Entry-Type) manipulator, and it uses the configuration of REPRPP (Roll-Extend-Pitch-Roll-Pitch-Pitch). A summary of arrangements and configurations is shown in Table 3.1.

Table 3.1 Reference arrangements of LRM

Configuration	Arrangement
Roll-Extend-Pitch-Roll-Pitch-Extend	a) Folded Entry Type (FET) b) Telescopic Sequential Entry Type (TSET)
Roll-Extend-Pitch-Roll-Pitch-Pitch	a) Sequential Entry Type (SET)

The three types of manipulators mentioned above are positioning devices only and have six degrees of freedom[5]. Hence, they all possess three degrees of redundancy. This redundancy is necessary for different tasks that require obstacle avoidance, singularity elimination, coverage of the entire space, etc. Advantages and disadvantages of each arrangement and the necessity for each degree of freedom are analyzed.

Also figure 3.1 shows the initial simulation model, developed at the University of Tennessee, for the tank waste retrieval system. This and other similar models, were used to investigate the validity of each design on a Silicon Graphics Workstation.

3.2 Parametric Study of the Long Reach Manipulator Design

A parametric design study was performed to demonstrate the interaction between the manipulator and the constraint conditions[1]. From that study, structural design specifications of the LRM can be derived. A computer program was developed by ORNL to optimize the dimensions of links for various tank diameters, access port diameters, payloads and deflection design criteria.

3.2.1 Assumptions for the Manipulator Design

The LRM design for waste tank remediation is strongly affected by several assumptions. The waste level is assumed to be at the highest level, which represents 100% of the tank capacity. The manipulator has to have a working configuration in the dome clearance after it is deployed. An access port ranging between 24 and 42 inches is assumed to be at the center of the tank. Even though some underground storage tanks have an off-centered large entrance port, this study considered ports at the center of the dome.

3.2.2 Optimal Design of the Manipulator Links

A link with a circular cross section would have the highest inertia-to-area ratio for the circular entrance hole restriction[4]. However, round columns are very difficult to manufacture. Because of the relatively high moment of inertia of a hexagonal cross

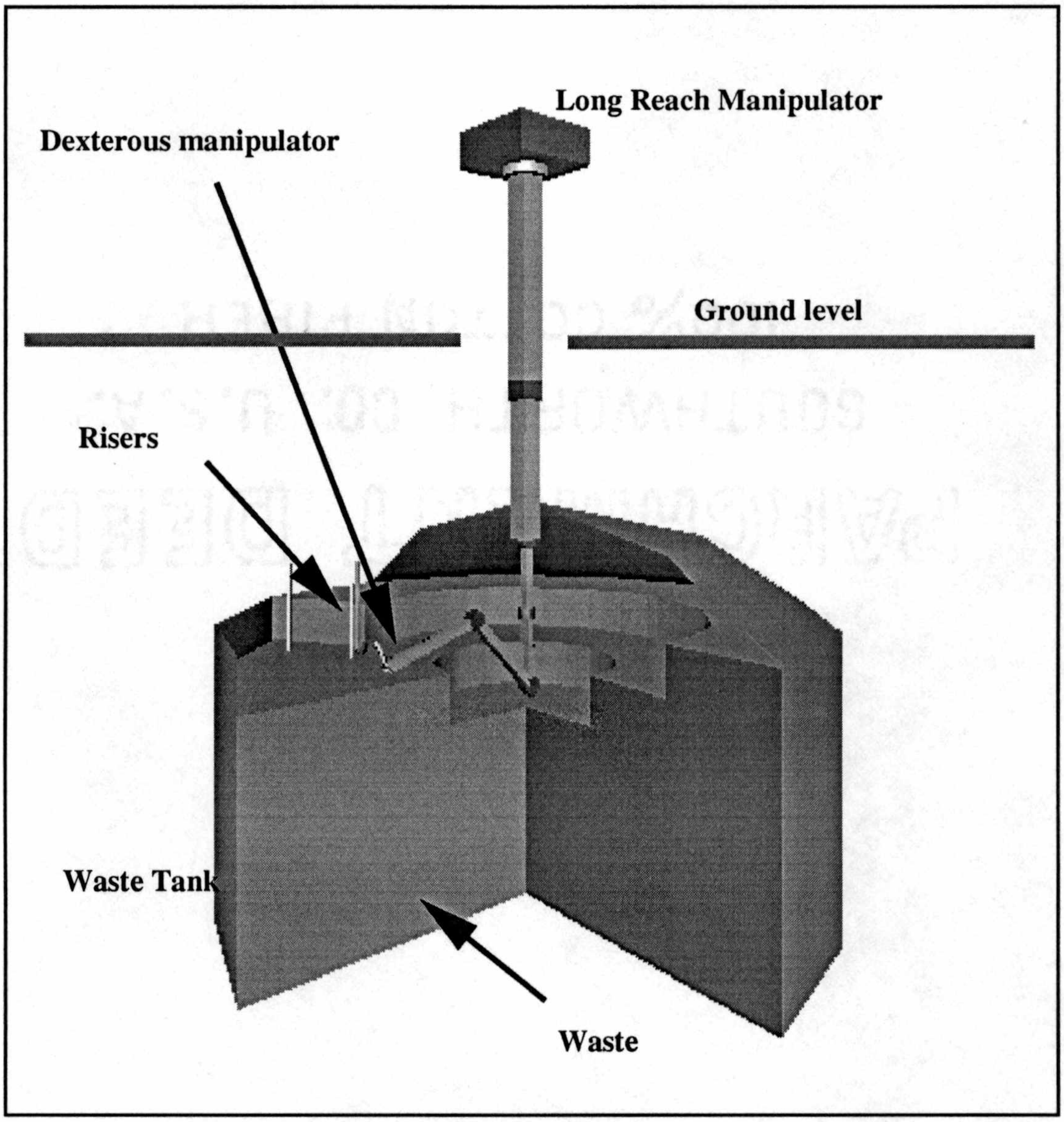


Fig. 3.1. Simulation model for waste retrieval system

section, it was chosen for the column; a square cross-section was chosen for the other links.

The analysis of link diameters led to the following results : for SET the minimum diameter of the lower links is 10 in., 11.5 in. for the upper link and 15 in. for the column. For FET manipulator, the lower and upper links diameters are the same as SET, but the column must be greater than the sum of both, and is assumed to be 120% of the sum of the two.

The size and thickness of the links are mainly determined by the static deflection which is related to natural frequency. It has been shown that a bandwidth of about one half of the fundamental natural frequency yields the maximum damping for a manipulator joint controller[13].

3.3 Comparison of Potential Configurations

First of all, the three mentioned arrangements in the beginning of this chapter have to satisfy all requirements discussed earlier. When these three arrangements were proposed and analyzed, the long reach manipulator was only considered for positioning the end effector, while the orientation would be accomplished using the dexterous manipulator mounted on the LRM.

The TSET and FET manipulators have the same configuration (R-E-P-R-P-E) shown in figure 3.2(A). The links of the TSET manipulator (shown in figure 3.3) are straight, requiring a higher tower than the FET manipulator (shown in figure 3.4) with folded links. The other configuration (R-E-P-R-P-P) shown in figure 3.2(B) is for the

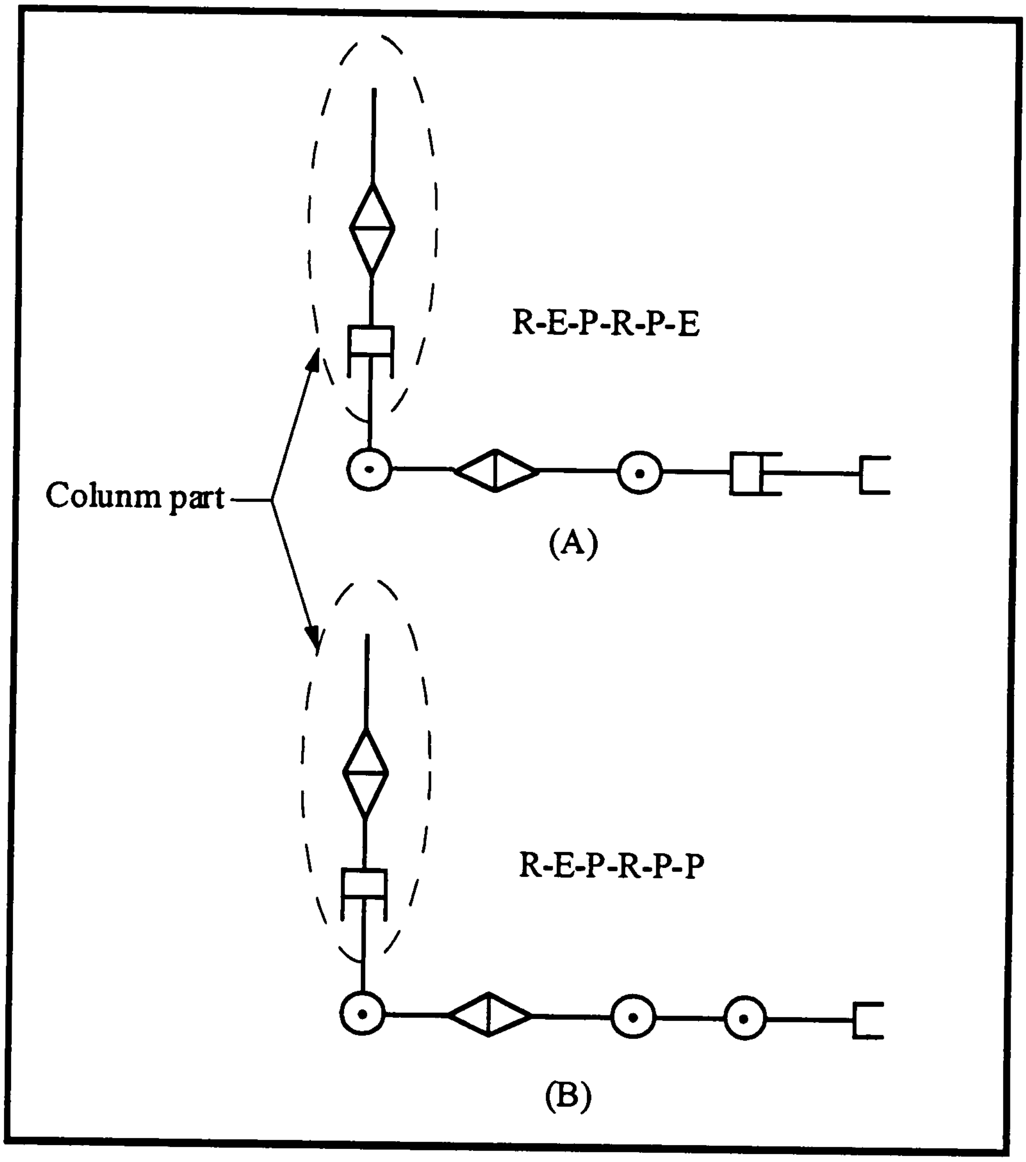


Fig. 3.2 LRM configurations (A) Roll-Extend-Pitch-Roll-Pitch-Extend
(B) Roll-Extend-Pitch-Roll-Pitch-Extend

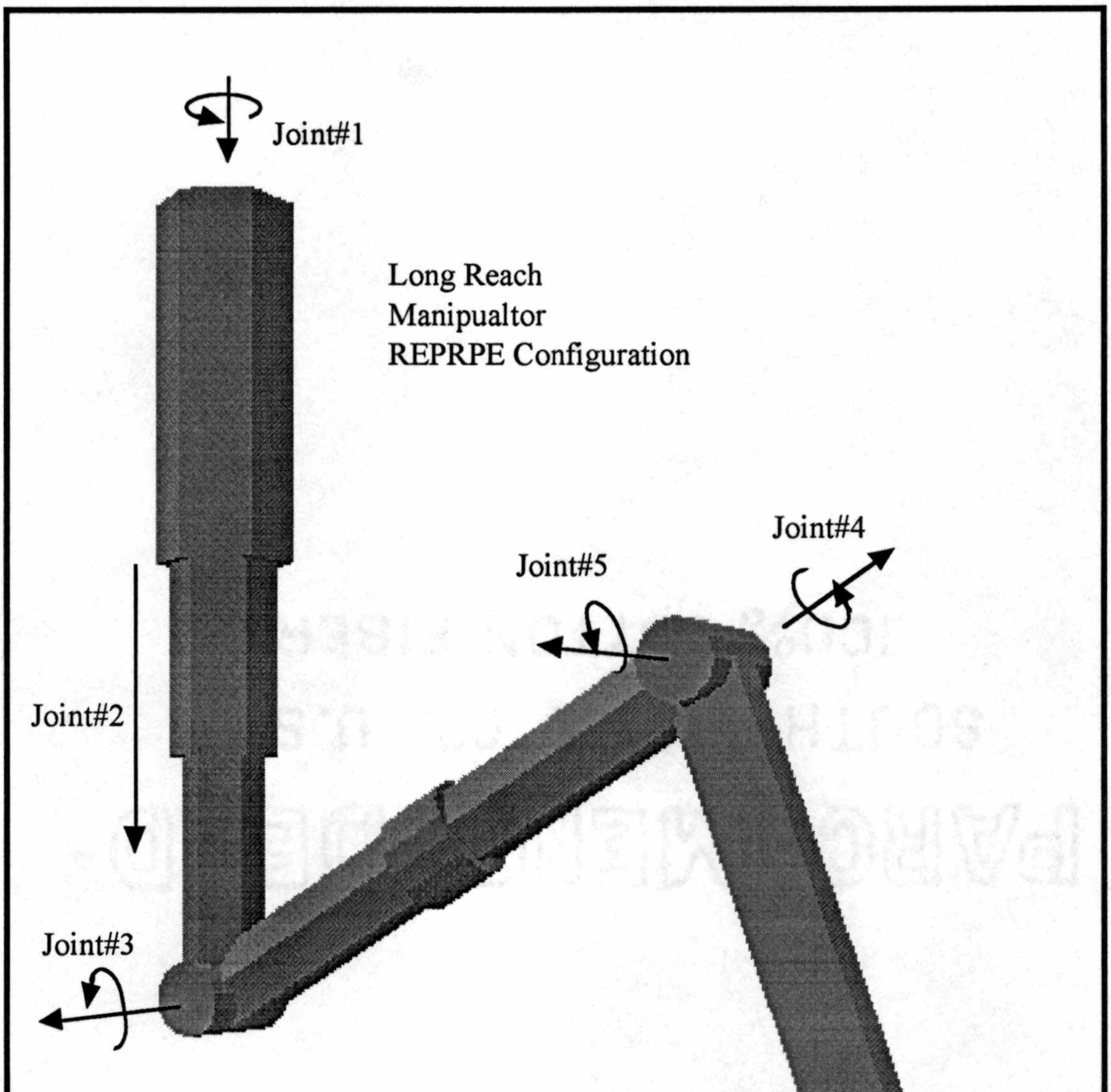


Fig. 3.3. Telescopic-sequential-entry-type (TSET) long reach manipulator

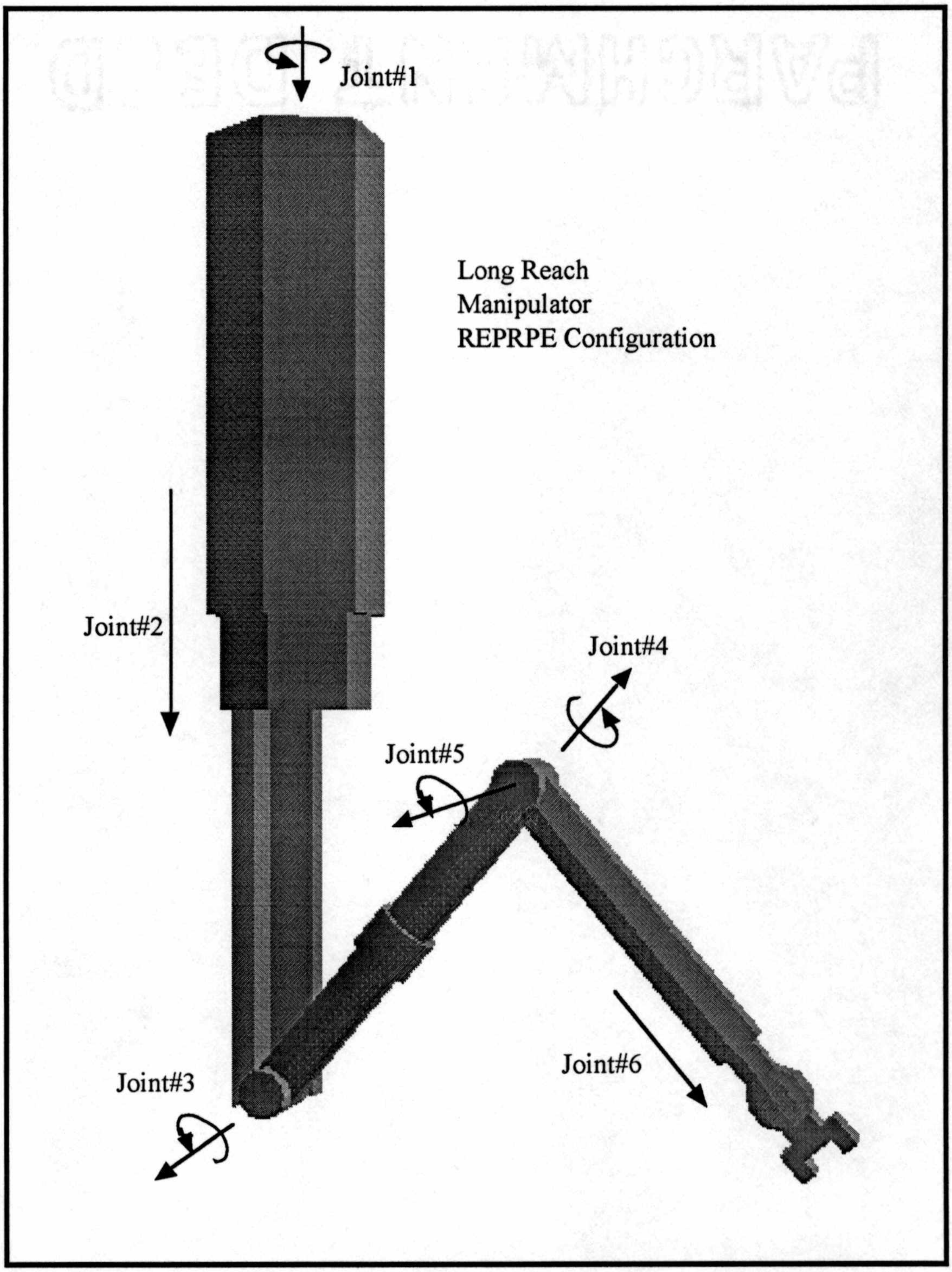


Fig. 3.4. Folded-entry-type (FET) long reach manipulator

SET manipulator (shown in figure 3.5). This manipulator requires the maximum tower height for deployment.

The two similar configurations in figure 3.2 have been identified as potential candidates for an LRM for waste tank remediation. Each is a six degree of freedom positioning device. The last joint is the only difference between the two configurations. Both configurations attempt to enlarge the workspace while allowing deployment through the tank access port into the dome clearance. The following figures show each of the three joint arrangements.

3.3.1 Comparison of Deployment Procedures

One of the most difficult operations of the LRM is its insertion through the access port in the dome clearance over the highest possible waste level. The access port is relatively very small compared with the workspace to be covered, which make deployment a challenging task to deal with.

The folded entry type (FET) LRM deployment procedure is best illustrated in figure 3.6. The FET manipulator can fold up the upper and lower arm inside the mast and unfold when lowered in the tank above the surface of the waste. The deployment of this type of manipulator is not restricted by the presence of risers or obstacles and can be deployed without any waste removal.

The telescopic sequential entry type (TSET) deployment stages are shown in figure 3.7. This type manipulator sequentially enters the tank starting with the last link. Once the TSET manipulator is deployed, the use of the extension joint is important to reach the end of the tank. Deployment of this arm requires planning because sometimes during intermediate deployment steps, it becomes necessary to avoid obstacles, the dome wall, or the waste.

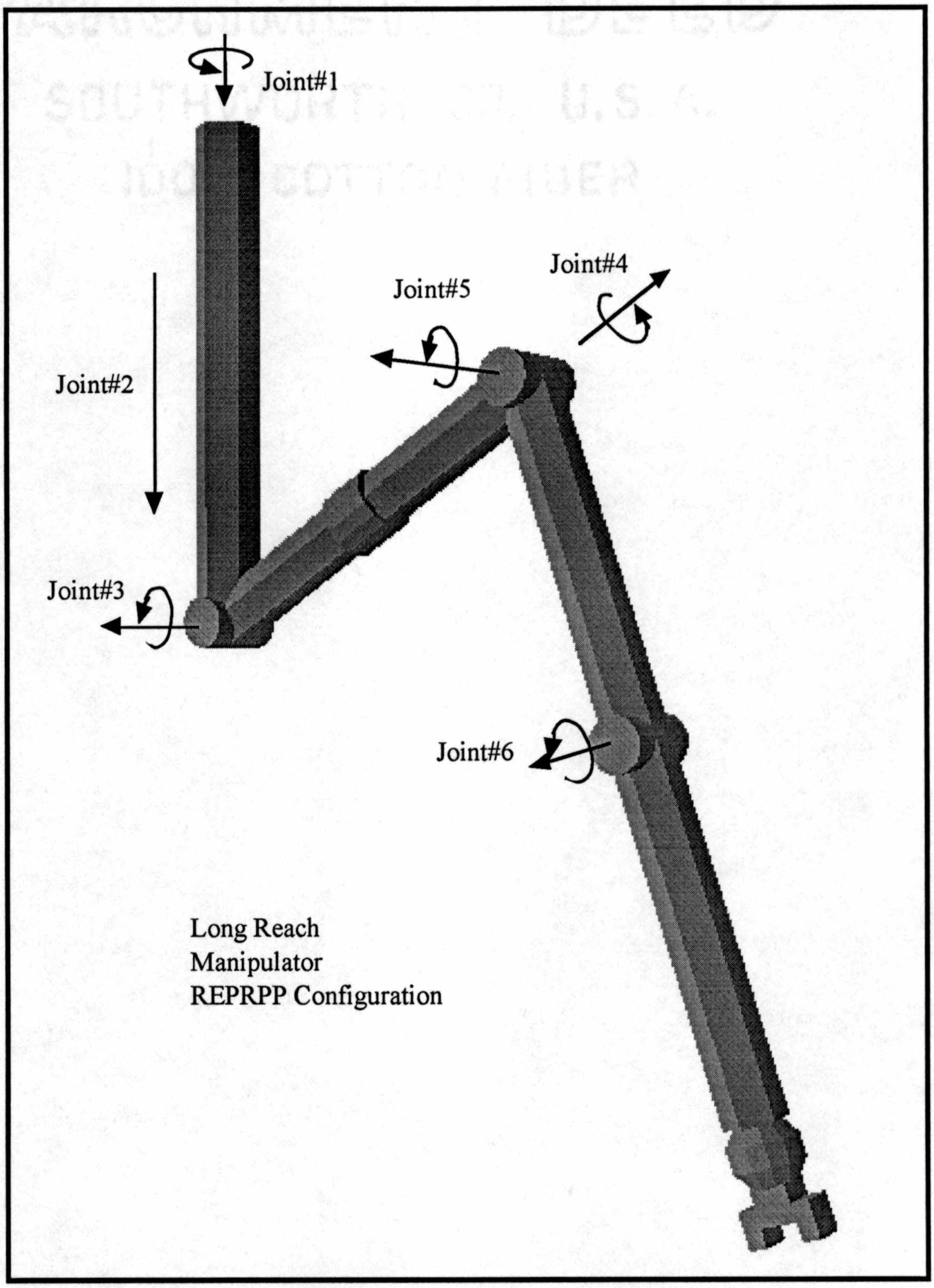


Fig. 3.5. Sequential-entry-type (SET) long reach manipulator

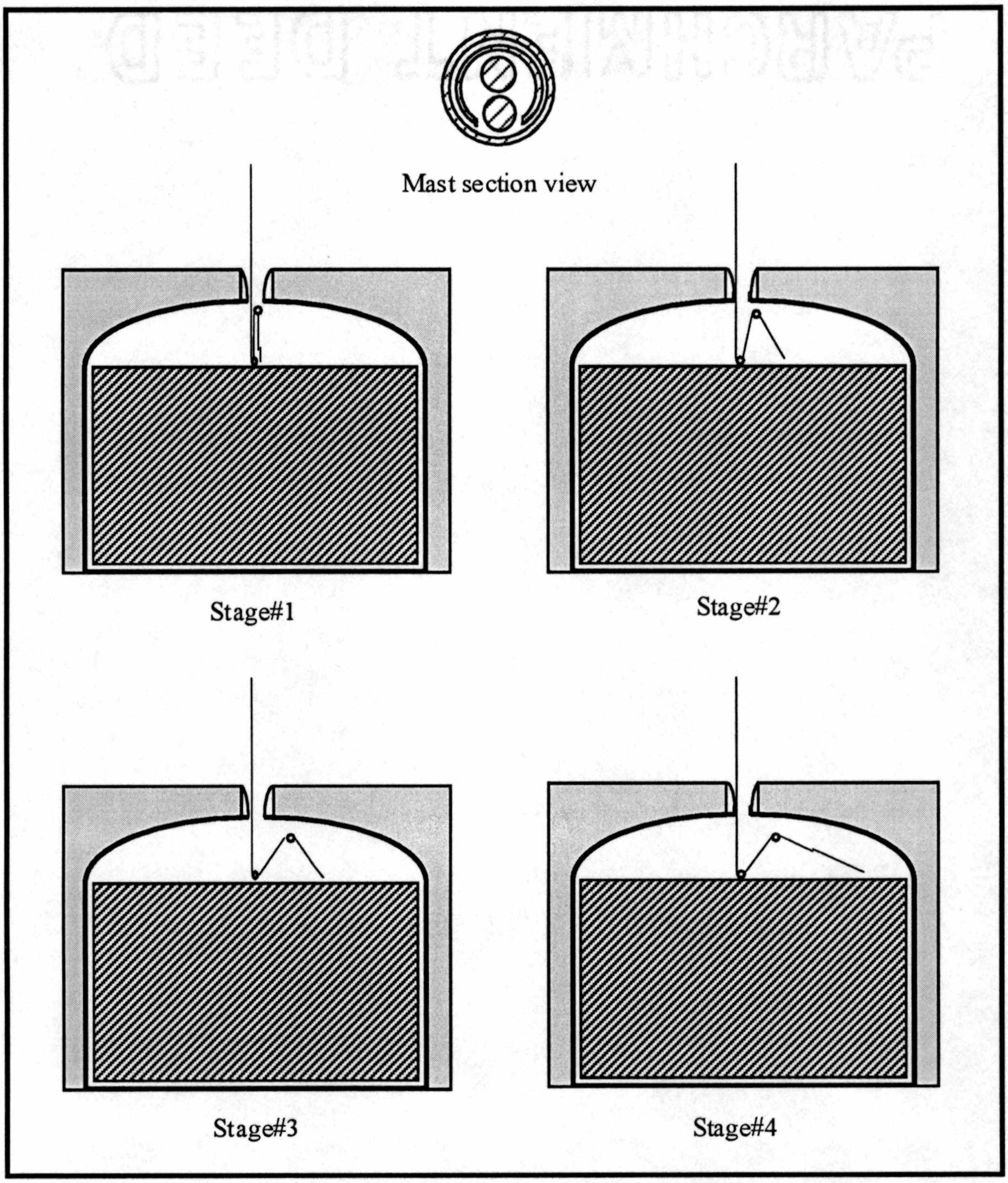


Fig. 3.6. Folded-Entry-Type (FET) long reach manipulator deployment

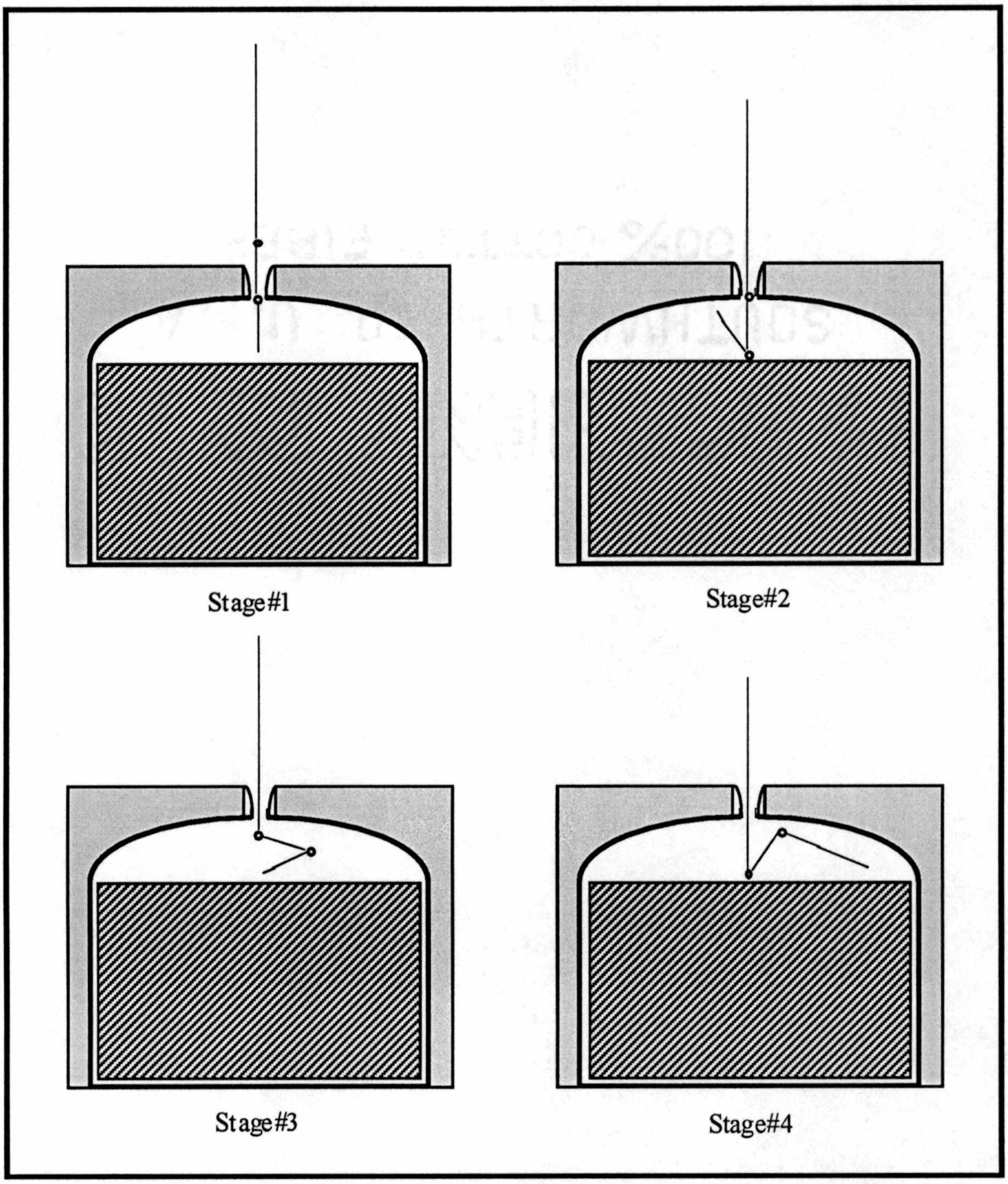


Fig. 3.7. Telescopic-Sequential-Entry-Type (TSET) long reach manipulator deployment

The sequential entry type (SET) deployment stages are shown in figure 3.8. Since this manipulator has a different configuration than first two, it will have a different strategy for deployment. This strategy will initially introduce the last link, or two last links of the arm to clear a hole to enable deployment of the entire arm. This deployment strategy will depend on the ratio of dome clearance to tank diameter.

3.3.2 Advantages and Disadvantages of Manipulator Types

Every manipulator type has its own advantages and disadvantages. Therefore, the final manipulator design choice involves trade-offs among a number of characteristics: for example, weight, desired static deflection, natural frequency, and tower height. A general comparison of the advantages and disadvantages of the three types of manipulators is discussed in the following paragraphs and summarized in Table 3.1.

The SET with a rigid column has the simplest mechanism, but on the other hand, it requires the highest support tower. The FET reduces the tower height by folding the manipulator links inside the column. The height of the tower in FET manipulator can be reduced to 28% of the SET manipulator.

Though for the same FET manipulator weight, larger deflection and lower natural frequency are encountered. So in general, for the same performance in terms of deflection and natural frequency, a larger access port is required. Two advantages of the FET are easier deployment inside the tank and easier extraction in the event of power failure.

A TSET manipulator provides one compromise between both SET and FET cases discussed above. The tower height can be reduced to 58% of that of the SET with a rigid column. Static deflection and natural frequency in this type manipulator are the same as in the SET.

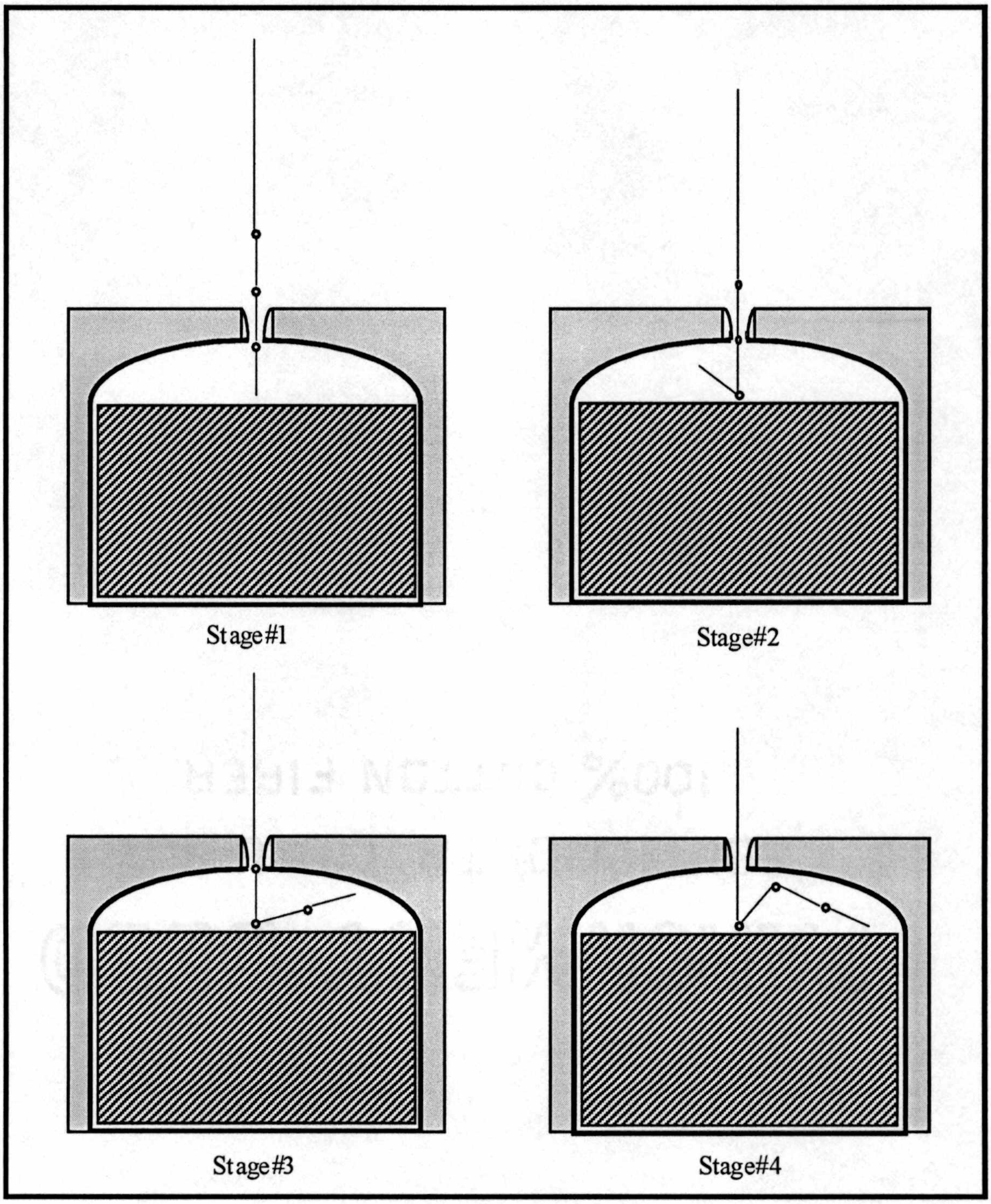


Fig. 3.8. Sequential-Entry-Type (SET) long reach manipulator deployment

Table 3.2 Advantages and disadvantages of each type of the LRM

Features	TSET	SET	FET
Tower height (rigid or telescopic column)	(Medium) 74 or 47 ft.	(Highest) 90 or 63 ft.	(Shortest) 52 or 25 ft.
Failure recovery	Medium	Most difficult	Best
Link length	Medium	Maximum	Minimum
Backlash	Same	Lowest	Same
Weight	Close to SET	Lightest	Heaviest
End-point speed	Same	Faster	Same
Natural frequency	Highest	Medium	Lowest
Static Deflection	Smallest	Medium	Largest

3.4 Necessity of Each Degree of Freedom

When used as positioning devices, all manipulators mentioned in this chapter have three redundant degrees of freedom. As a result, operation of these manipulators will require a solution of the redundant inverse kinematic problem. An appropriate solution for this case is the use of gradient projection optimization for multiple degrees of redundancy[10] which yields a closed-form solution for this 6-DOF positioning manipulator. The following explains the necessity of each DOF of the LRM.

3.4.1 First DOF

The first rotational joint at the base is necessary to maximize the workspace relative to the radial symmetry of the tank. This type rotational joint is very common to most industrial robots, and is used here in all types of LRM manipulators (FET, TSET, and SET). When this joint is included, the tank volume is reduced to a vertical plane. In other words, that if a reachable area is produced by the rest of the manipulator, the rotation of that volume about the first joint axis will represent the reachable space in the tank as shown in figure 3.9.

3.4.2 Second DOF

Comparing the spherical workspace of the robot with a rotational joint (as shown in figure 3.9) with the tank workspace to be covered it can be concluded that the second DOF which is a translation along the z axis can be integrated with the first DOF to produce a workspace similar to the tank workspace itself (as shown in figure 3.10). A telescopic prismatic joint in the column is a great advantage in minimizing the tower

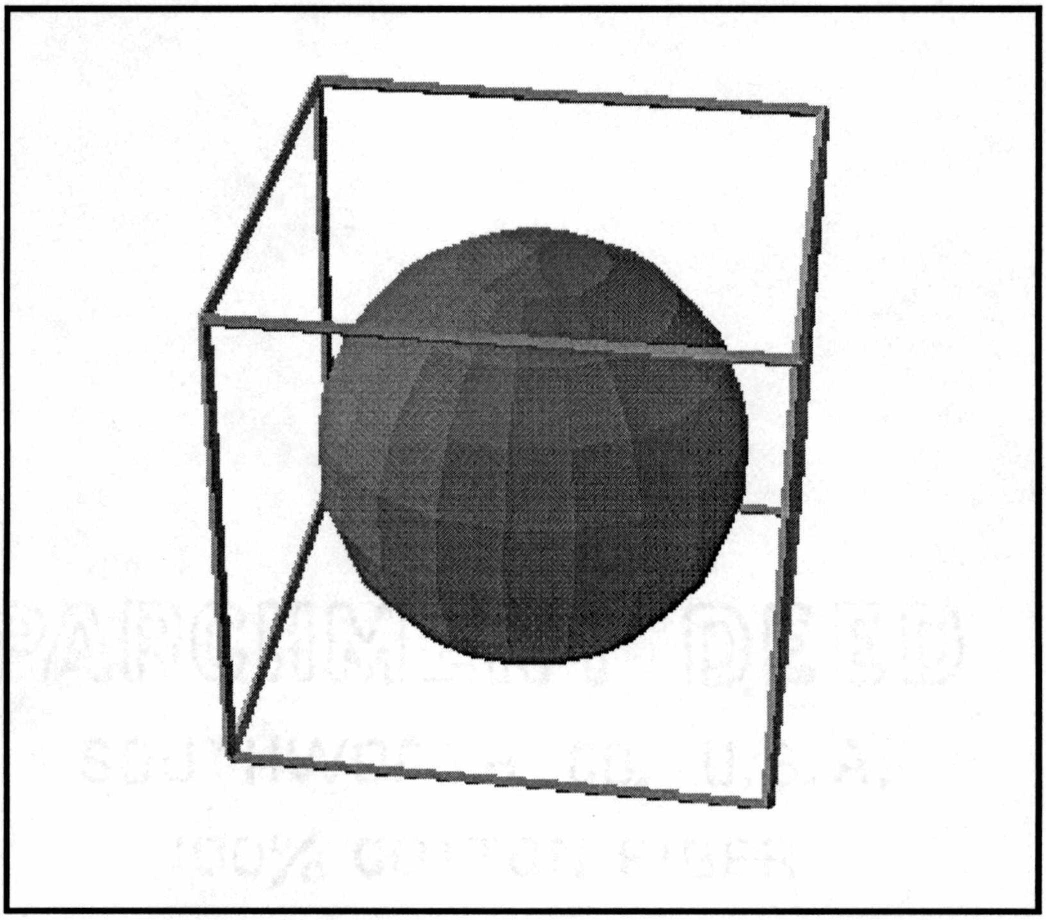


Fig. 3.9. Symmetric workspace of a robot with first rotational joint

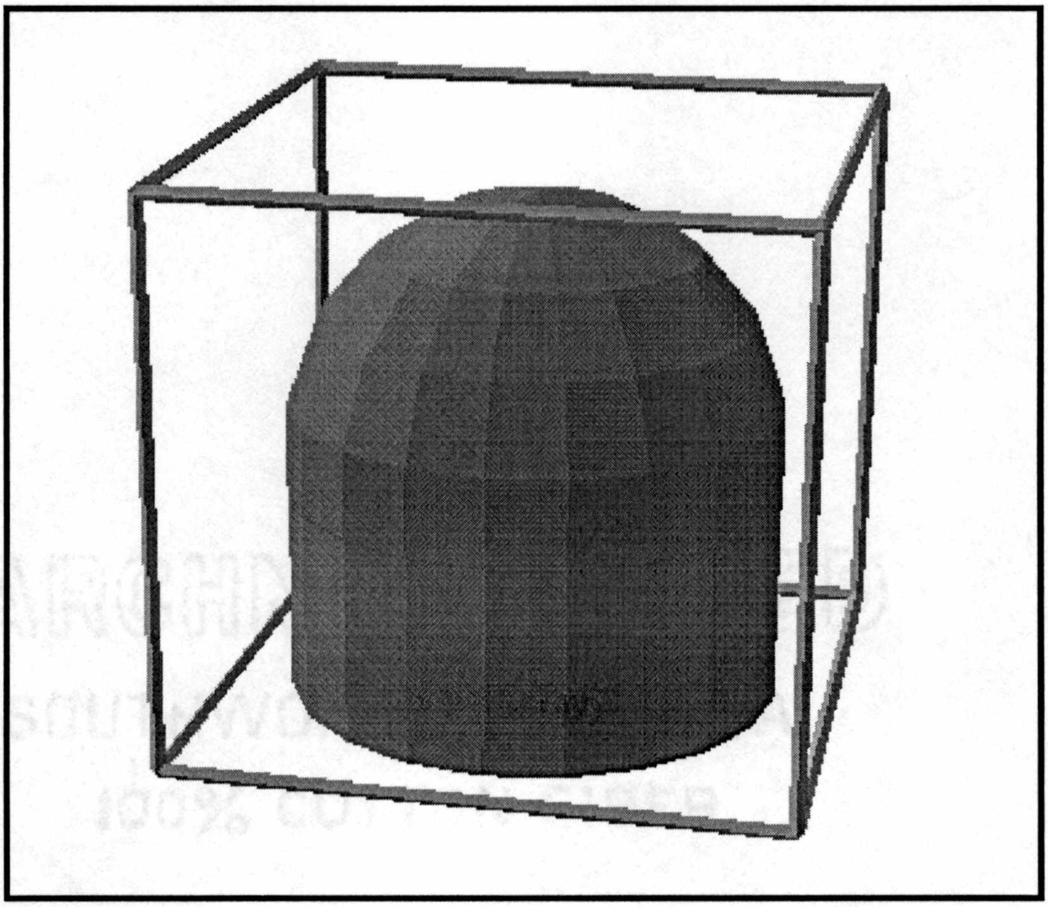


Fig. 3.10. Symmetric workspace of a robot with first rotational joint and second translational joint

height. Also, reducing the length of the column minimizes the deflection and vibration of the other links.

3.4.3 Third and Fifth DOF

The two rotational pitch joints give maximum workspace coverage for minimum overall length of the manipulator[15]. Also, rotational joints in general have less lateral backlash than prismatic joints. For both FET and TSET manipulators, equal lengths links are attached to these two rotational joints. So these arrangements provide maximum workspace for the same overall length of the manipulator while retaining the capability for deployment[4].

3.4.4 Fourth DOF

The rotational joint between the two pitch joints is a roll joint. This roll allows the LRM to work in SCARA, elbow up or down, or in intermediate configurations. Also the singularity in all three types of LRM, when the end point comes in line with the column axis is eliminated[1]. A proof of this situation is shown for the TSET manipulator. Without the fourth DOF, it was found that the rank of the positional columns of the Jacobian matrix is less than 3 when the following equation is true:

$$a_3 \cos \Theta_3 + d_5 \cos\left(\Theta_3 + \Theta_4\right) = 0$$

This equation describe the end of the manipulator when it is located along the axis of the column, and that is a singular configuration as shown in figure 3.11. The fact that it

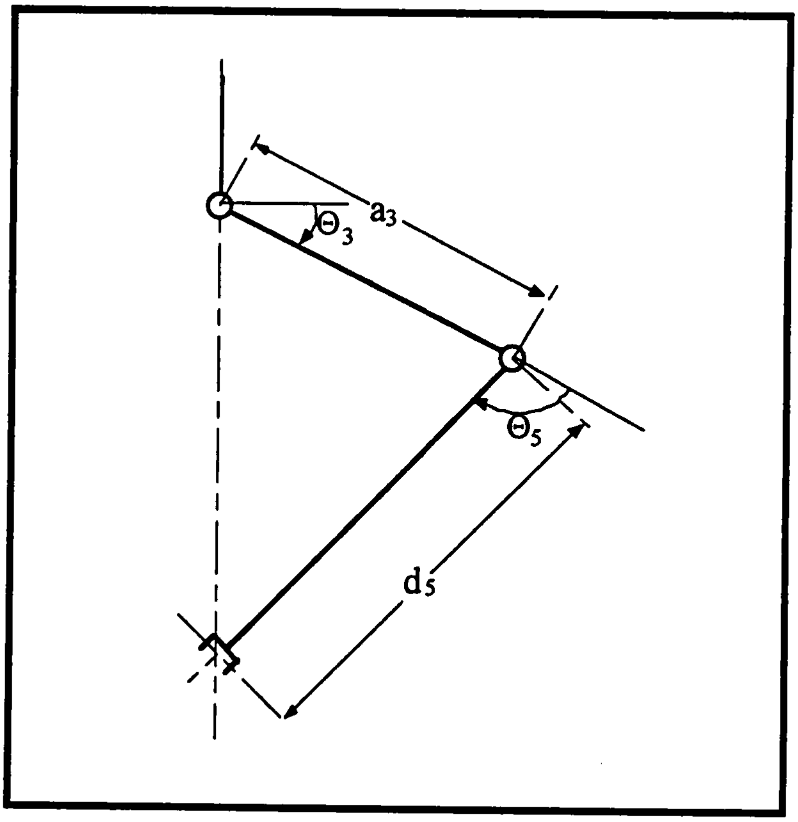


Fig. 3.11. Singular configuration without the 4th DOF

is a singularity is shown by the inability of the end point to move in the direction perpendicular to the plane, defined by the column and the two last links.

A consequence of the use of this DOF is an improvement of the connectivity of the workspace of the manipulator [14]. If this DOF is not included, then the second pitch joint (fifth DOF) at times would have to move through the plane of the two links. This lack would greatly affect the ability of the manipulator to move between two points since it will be required sometimes to switch between elbow up and down configurations.

This roll joint reduces the second pitch joint range of motion from between -140° and 140° to between 0° and 140°[5]. Then it becomes possible to use linear actuator with less backlash and friction instead of rotary actuators. Also this roll joint enables the manipulator to reach the top of the tank before any waste removal for inspection purposes.

The redundabncy provided by this additional DOF would allow orientation of the links and joints to maximize the ability to damp vibration, reduce the maximum deflection of the manipulator, optimize its mechanical advantage, or improve its manipulability[5].

3.4.5 Sixth DOF

This is the only joint that is either rotational or translational in the two different configurations used by the LRM. For the FET and TSET, this joint is prismatic and has the advantage of an increased workspace covered by the manipulator. On the other hand, the rotational joint of the SET has less lateral backlash.

The prismatic joint makes the manipulator's overall length 10 feet shorter, with the advantages of reducing the tower height, reducing the static deflection and increasing the structural natural frequency. Also, a telescopic joint is recommended to improve deployment and obstacle avoidance capabilities[5].

Chapter 4

Schilling Design and UT Modified Design

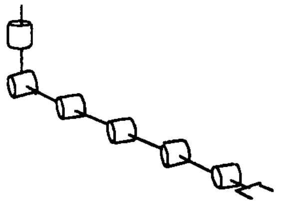
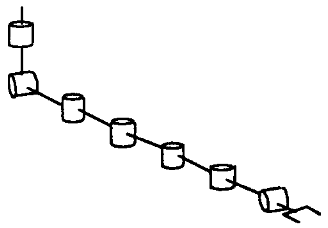
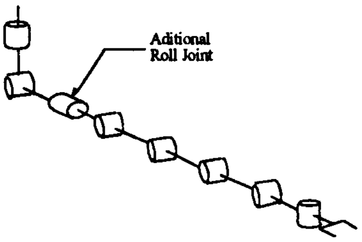
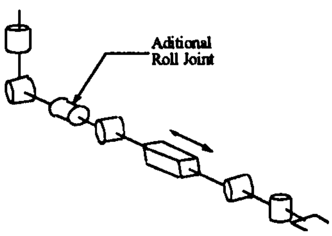
4.1 Introduction

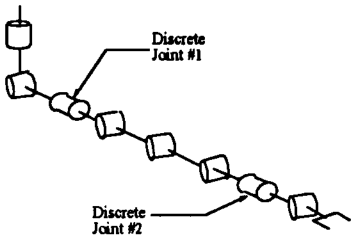
As discussed in chapter 3, several design configurations were under consideration. Although variations in design, such as folded, telescopic, and sequential entry types were considered, the sequential entry type has been selected because of insignificance of the backlash in the rotational joints relative to the backlash in prismatic joints. Shortly thereafter, the designs under consideration were further narrowed to include only the all-pitch sequential entry configuration developed by Schilling. The expensive and difficult to manufacture continuous roll joints (third DOF) were eliminated in the development process. Also, deployment was not possible with the original link lengths. As a result, four shorter links were required to reach the entire volume of the tank.

A new phase of design developed by Schilling resulted in the choice of the SDI/Spar recommended design[7]. Table 4.1 is a comparison table that summarizes the advantages and disadvantages of the considered designs for this phase.

The new design recommended by SDI/SPAR and adopted by Schilling is a seven joint manipulator with joint configurations Roll-Extend-Pitch-Pitch-Pitch-Pitch-Pitch (R-E-P-P-P-P-P) and two discrete indictable joints. The two indictable roll joints are placed between the first two and last two pitch joints. Analysis of the Schilling design revealed some limitations with regards to deployment and workspace coverage. Limitations were investigated analytically and using IGRIP simulation. A simple modification, replacement

Table 4.1 Conceptual designs (phase #2)

Concepts	Advantages	Disadvantages
• Elbow Arm Configuration 	• Good vertical motion • Simple design • Will meet reach requirements	• Can not avoid vertical obstacles or reach waste behind risers.
• Scaleup Light Duty Utility Arm (LDUA) Configuration 	• Capable of reaching around vertical obstacles • The use of existing control algorithms • Can be deployed within dome height • Will reach entire tank volume	• Require continuous mast motion. • Sizing scissor link for joints in gravity plane for high loads and moments is a problem.
• Combined Elbow/LDUA Configuration • Roll joint added is to allow elbow/SCARA configurations 	• Capable of reaching around vertical obstacles • More efficient vibration testing • Will reach entire tank volume	• Hydraulic roll joints are expensive to manufacture. • The addition of shoulder roll joint requires larger link size than allowed for deployment. • Same sizing problem in LDUA configuration.
• Translational Forearm Configuration 	• Simple deployment • Reduce the tower height • Will reach entire tank volume	• Prismatic joint requires a larger link size than allowed for deployment. • The use of the long translational segment reduces dexterity in avoiding obstacles. • Complex hydraulic services. • Roll joint problem as before.

Concepts	Advantages	Disadvantages
• SDI/SPAR RECOMMENDATION • Discrete Joints Added • Switching back and forth between elbow and SCARA configurations 	• Simplest design to reconfigure the arm • Vibration control testing can be done in both configurations • Necessity of roll joint can be evaluated before building • Will reach entire tank volume • Provides a test bed with several options • Easy hydraulic arrangement	• indexing the discrete roll joints is done manually.

of the third pitch joint by a yaw joint, was proposed by UT to improve the LRM capabilities. After the modification, the joint configurations became Roll-Extend-Pitch-Pitch-Yaw-Pitch-Pitch. This chapter gives a detailed description of each design, as well as its respective advantages and disadvantages. Also, a comparison of tank waste clean-up strategies is shown through IGRIP simulation developed at the University of Tennessee.

Simulation on IGRIP software for the Schilling LRM was done immediately after the model built at PNL was received[9]. The construction of the simulation included a tank built at the University of Tennessee with dimensions identical to existing single shell waste tanks. Also the waste in the tank is represented by rings approximately 3 by 3 feet cross section. It is assumed that the LRM is capable of removing the entire ring, by the mast rotational joint, once the end effector touches the ring.

4.2 Schilling Design

As mentioned before, the baseline design recommended by Schilling had the joint configuration Roll-Extend-Pitch-Pitch-Pitch-Pitch-Pitch. Also, two indexable roll joints were included, one after the first pitch and one after the fourth. Both discrete joints are capable of discrete positions 0° and ± 90°. Joint limits on all but the last pitch were -5° and 105°, while the last pitch had limits of 0° and 90°. This design would be used in one of two configurations, depending on the positions of the indexed roll joints: the elbow and SCARA. An illustration of the manipulator in each of these configurations is shown in Figure 4.1.

4.2.1 Elbow Configuration

The elbow configuration is the nominal configuration of the Schilling design and can be achieved when both indexed roll joints are in the zero position. A kinematic analysis of the elbow design was conducted to investigate the advantages and disadvantages of this configuration.

Obviously, the elbow configuration is the simplest arrangement of joints to design and control. Also, it satisfies the requirement that, upon failure, it can still be retrieved. In addition, there are no torsional stresses induced on the joints by the weight of the lower manipulator links, as there are in the other designs.

However, UT's kinematic analysis revealed shortcomings in the obstacle avoidance ability, frequency of retractions, and workspace volume of this design. Vertical obstacles, such as risers, cannot be avoided by the elbow configuration also. This condition implies that the sector area behind a riser would be unattainable by the LRM. Therefore, in order to clean a tank with vertical obstacles, an elbow configuration would be forced to cut them at every level of operation. Further, cleaning is impossible if retraction after complete

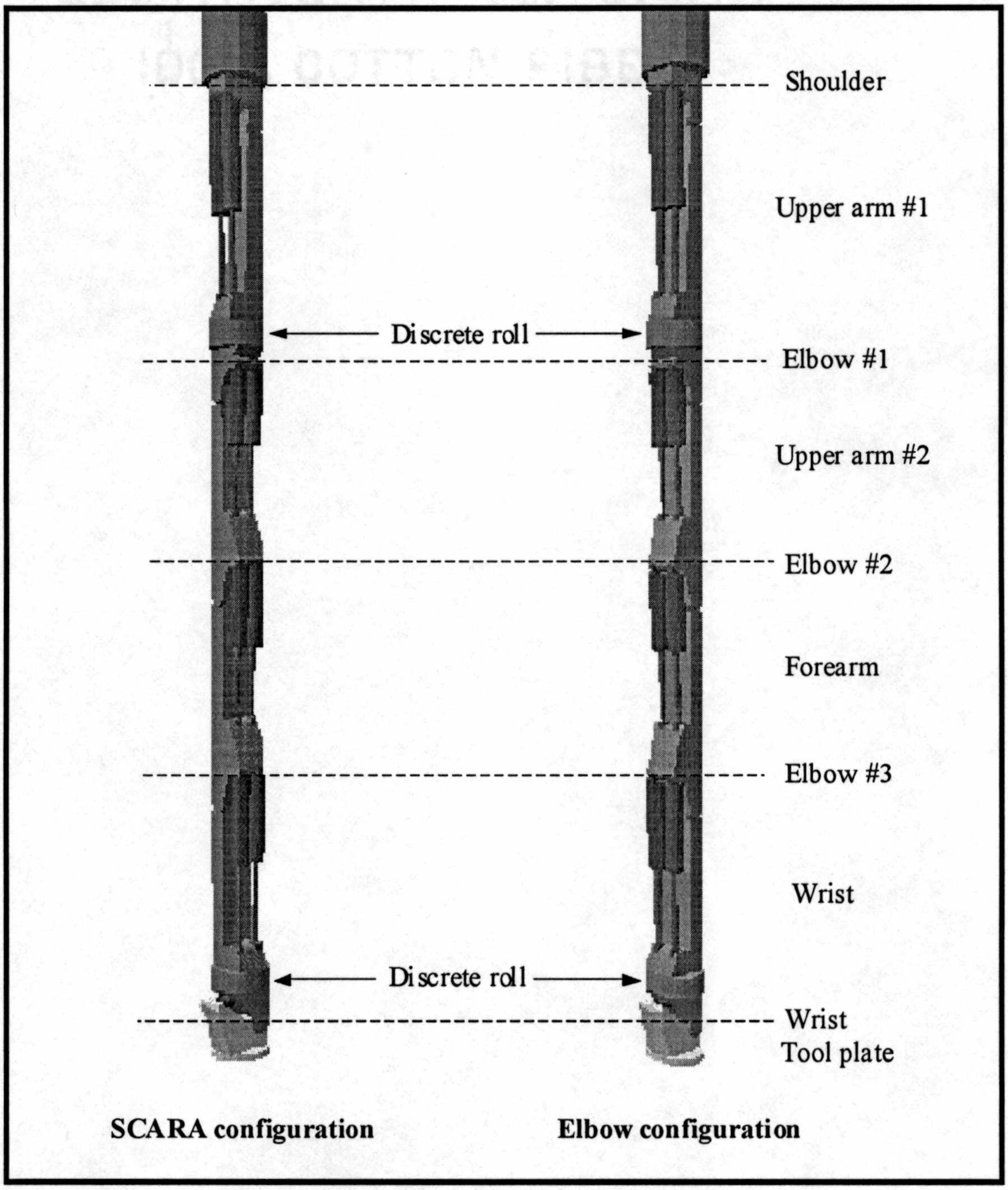


Fig. 4.1. Schilling design in elbow and SCARA configuration

deployment is disallowed. Once the manipulator is deployed, the end effector cannot clean at the top portion of the tank unless the second pitch joint is below this level. Therefore, in order to avoid interference between the arm and the waste, it must be retracted frequently when cleaning near the top. The workspace volume of the elbow configuration was also investigated. As shown by the workspace limits, represented by a solid line in Figure 4.2, the central area of the tank is not reachable after complete deployment. Note that the entire volume enclosed by this curve rotated around the central axis is inaccessible. Again, to reach this region, wider joint limits or manipulator retraction would be necessary. Although Figure 4.2 was constructed with the originally proposed link lengths, the unreachable workspace would be similar even if the lengths were changed.

4.2.2 SCARA Configuration

In order to configure the Schilling design as SCARA, the first indexed roll joint should be at 90°, and the second at -90°. The SCARA configuration had several advantages and disadvantages over the elbow, as explained below.

Because the SCARA would work in a horizontal plane, it has the ability to reach around vertical obstacles. Thus, risers would not interfere with its operation. In addition, it could completely clean a layer before advancing downward a level, so that the mast would be in use only once per level. After complete deployment, the arm could proceed to clean without subsequent retraction, as was necessary with the elbow configuration.

Unfortunately, the weight of the links in a horizontal position induce high torsional stresses on the pitch joints. In addition, the central column beneath the mast is not accessible, since the links would all be in the same plane. Interference with the mast itself prohibits cleaning of this central core. The joint limit of the mast extension also

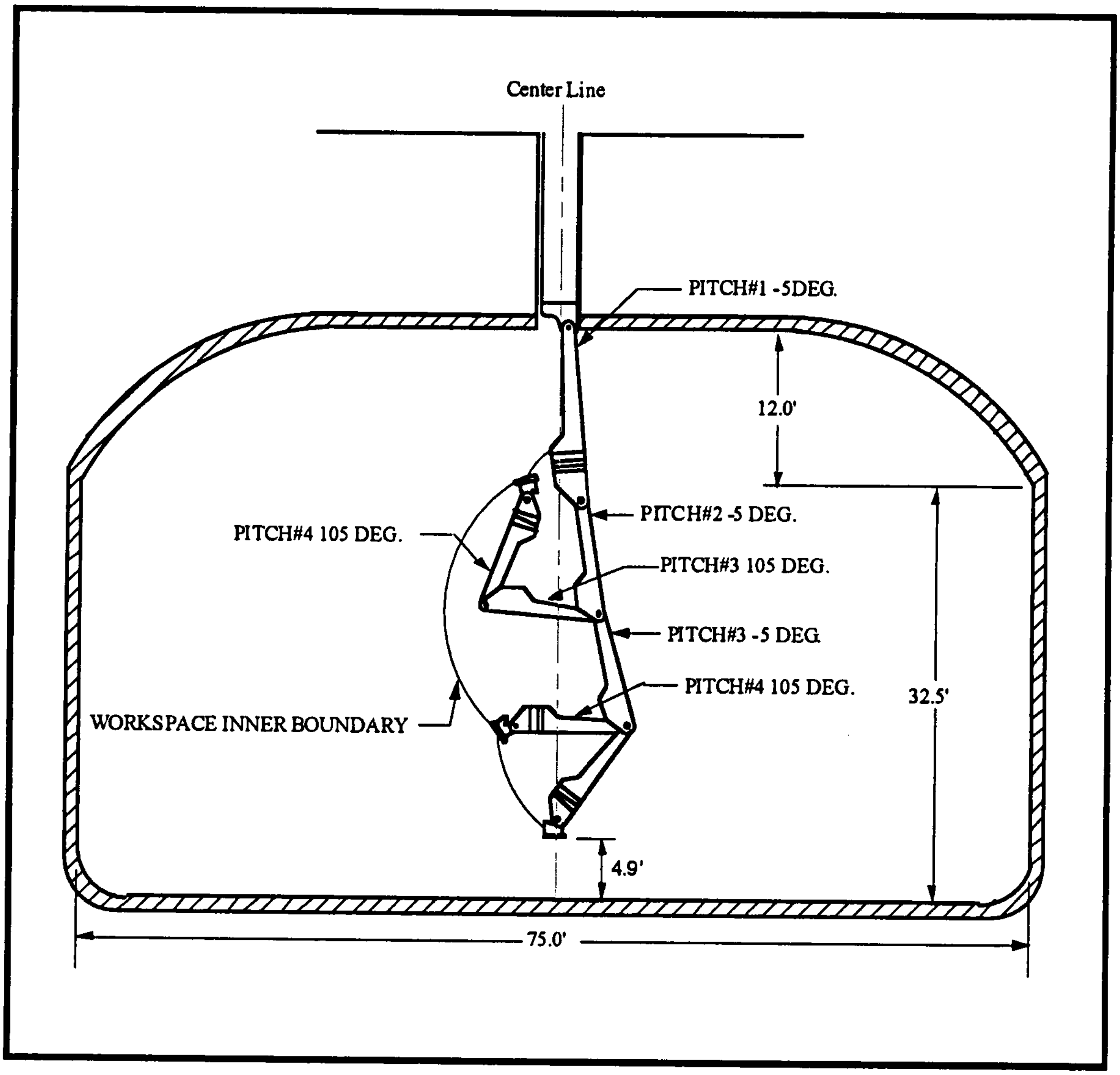


Fig. 4.2. Elbow configuration reachable volume

prohibits the end effector from reaching the bottom eight foot high cylinder. In order to reach this volume, the discrete roll joints would need to be re-indexed, so that the manipulator is returned to the elbow configuration, consuming time and energy. Figure 4.3 shows the top view of the tank with the manipulator in SCARA configuration. Also shown in this figure is the unreachable area at the central core of the tank.

4.3 Proposed Modified Design

The UT modified design is actually the Schilling design with a few modifications. In the UT design, the mid-elbow pitch joint has been changed to a yaw joint. For simplicity of manufacturing, this minor modification would be achieved by rotating the mid-elbow pitch joint by 90°. The link sizes and joint limits remain unchanged, and the new manipulator joint configuration is Roll-Extend-Pitch-Pitch-Yaw-Pitch-Pitch with two discrete roll joints at the same locations as in the Schilling design. The modified design is shown in figure 4.4 and the coordinate frames in figure 4.5. Also, the D-H table shown in Table 4.2 with sufficient link lengths for deployment without digging.

This design was proposed because of several advantages over the other designs. For example, retraction during digging was considered undesirable for several reasons, such as the possibility of contaminating the environment and wasting energy. The modified design is capable of cleaning waste material with elbow up configuration so that it does not contact the waste. Once the first index roll joint is positioned at 90°, the function of the pitch and yaw joints of the manipulator are reversed. Therefore, the second pitch joint becomes a yaw joint, permitting the manipulator to avoid obstacles. Once the modified design is deployed in the tank, continuous downward advancement of the mast is not necessary for cleaning multiple layers.

Unfortunately, the weight of the links in a horizontal position induces high

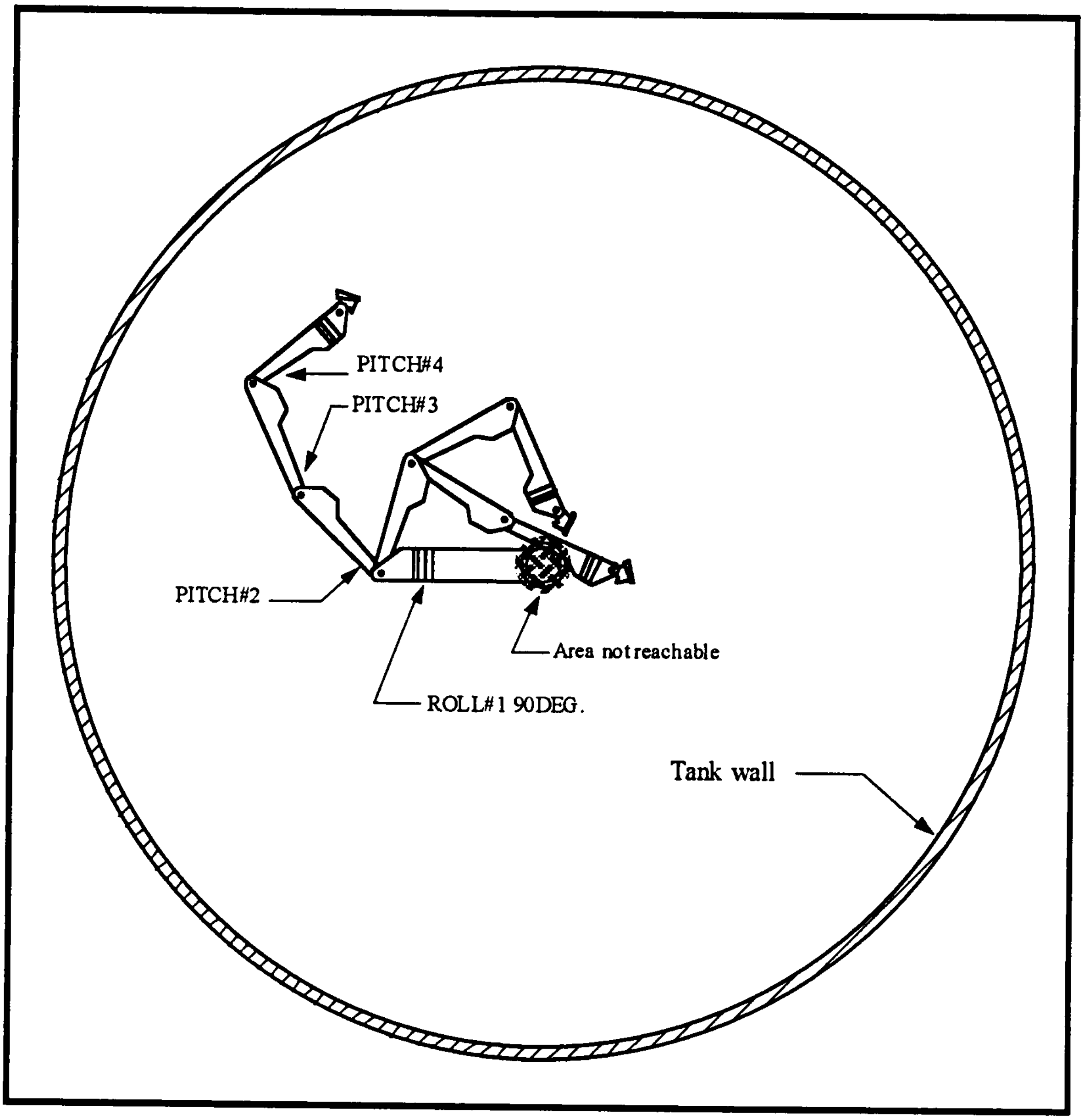


Fig. 4.3. SCARA configuration with unreachable central core

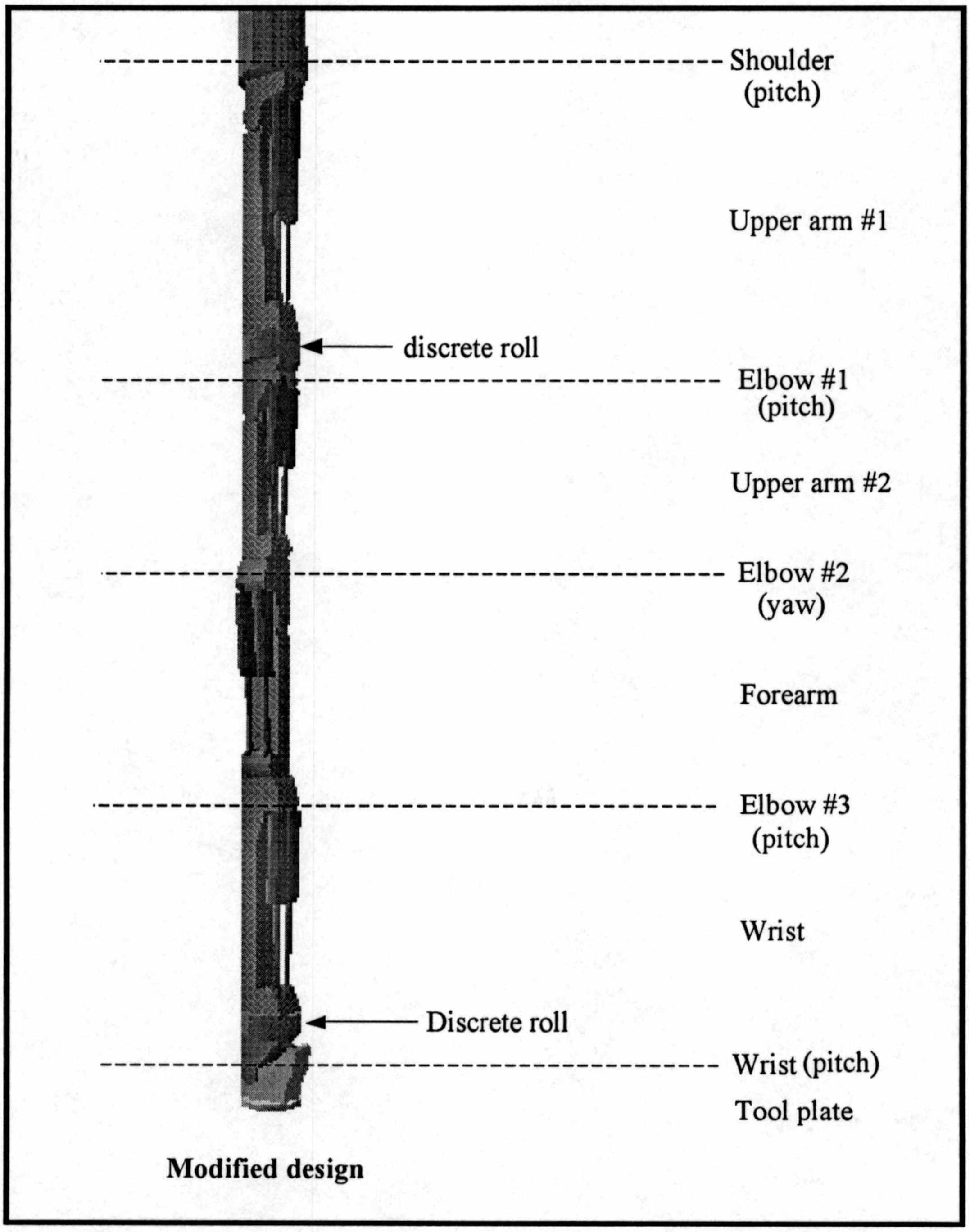


Fig. 4.4. Modified design

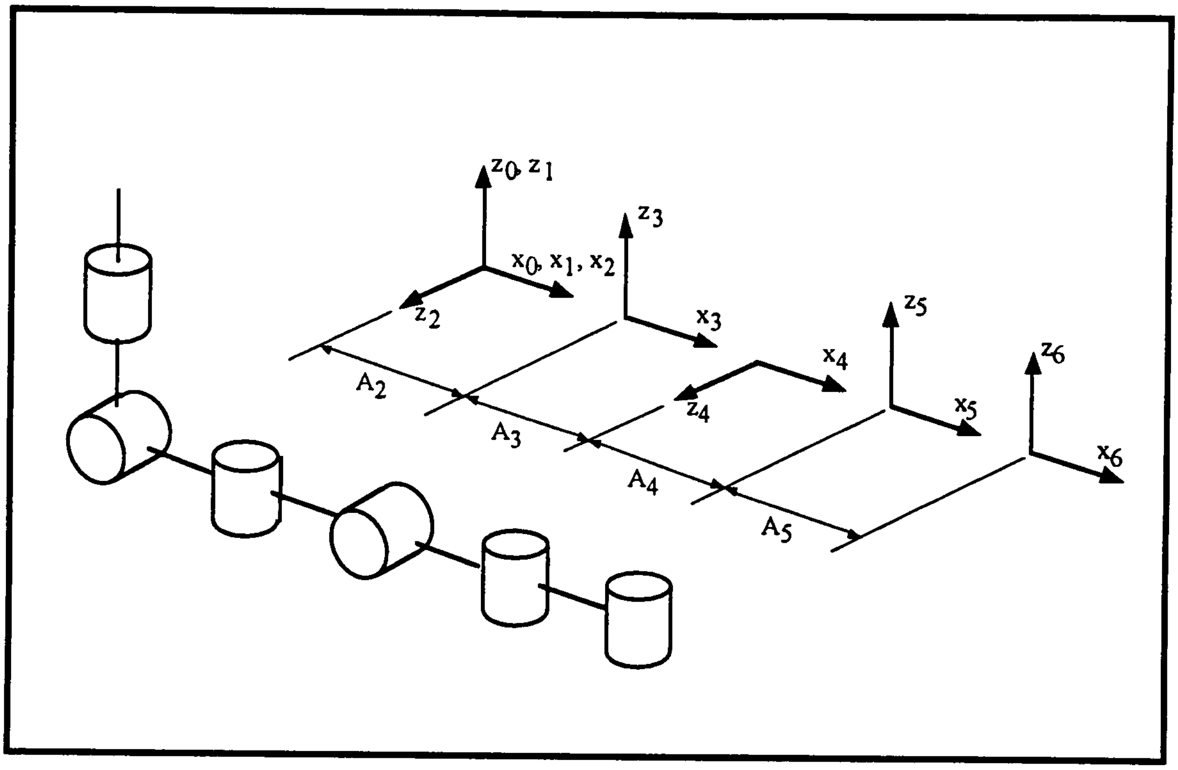


Fig. 4.5. Modified design coordinate frames

Table 4.2 D-H parameters table for the modified design

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	(θ_1)
2	0	90°	0	(θ_2)
3	A_2 (118")	-90°	0	(θ_3)
4	A_3 (118")	90°	0	(θ_4)
5	A_4 (118")	-90°	0	(θ_5)
6	A_5 (118")	0	0	(θ_6)

torsional stresses on the pitch joints. In addition, the central column of waste beneath the mast is not accessible, but it is expected to collapse once cleanup has reached the bottom part of the column.

4.4 Deployment of Shilling and Modified Designs

IGRIP simulation software was the verification tool for all the doubts generated from analytical and graphical solutions. It was shown in figure 4.2 and 4.3 that the Shilling design is not capable of covering the entire workspace in either elbow or SCARA configurations. Calculations were done and then verified by drawings to scale made in Auto CAD graphics software. Since it was necessary to determine the workspace coverage more accurately, IGRIP three-dimensional simulation software was introduced. An existing IGRIP Schilling design model built by PNL was sent to UT to be used in workspace coverage studies.

Link lengths were changed several times as recommended by various analyses. Initially, link lengths suggested by Schilling were as follows: link 1 (upper arm #1) was 162 inches, the next three were 108 inches, and the tool plate was 15 inches. The following studies regarding deployment were formulated using initial design lengths specified by PNL. The length of the upper arm #1 is greater than the allowed height for deployment. As a result, when mining strategies were formulated, it was found that some mining was necessary before complete deployment. Figures 4.6 , 4.7 and 4.8 show the elbow, SCARA and the modified designs during deployment. Notice that there has been some mining in order to deploy these arms.

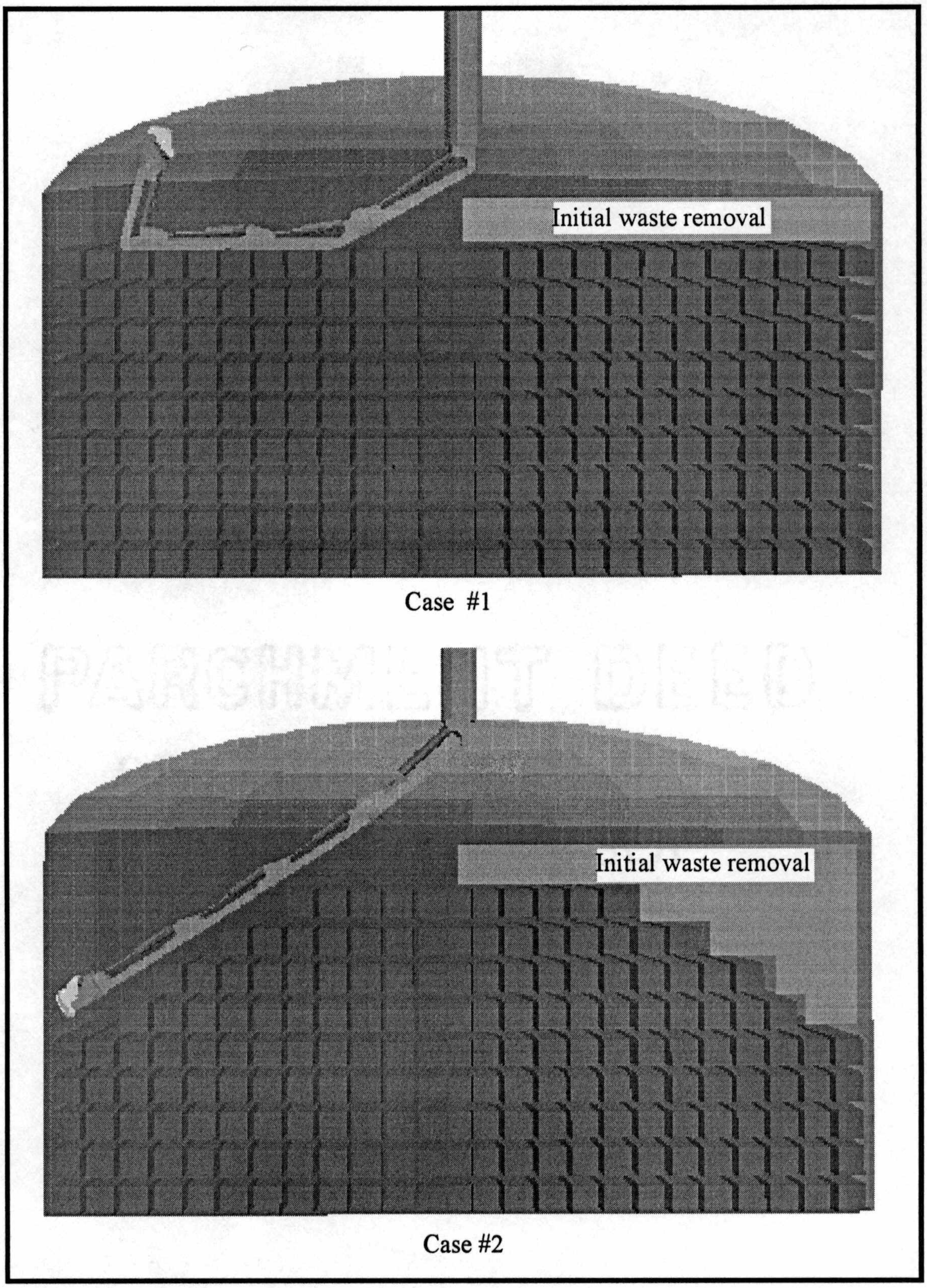


Fig. 4.6. Deployment of Shilling LRM design in elbow configuration

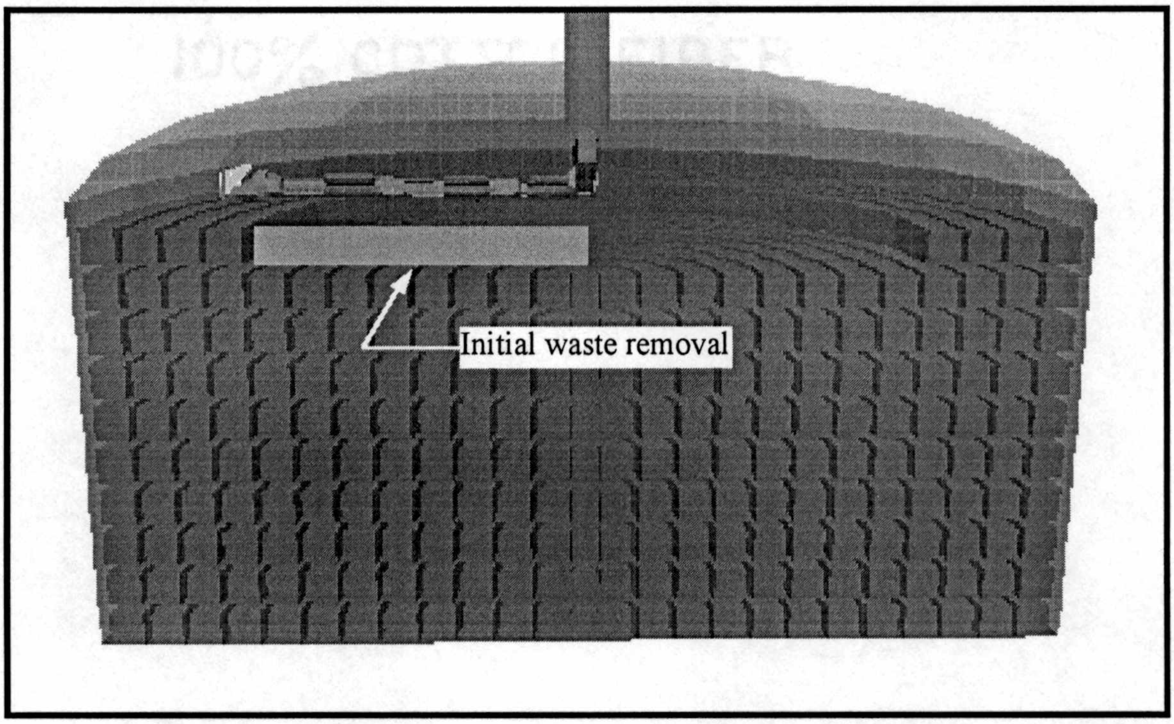


Fig. 4.7. Deployment of Schilling LRM design in SCARA configuration

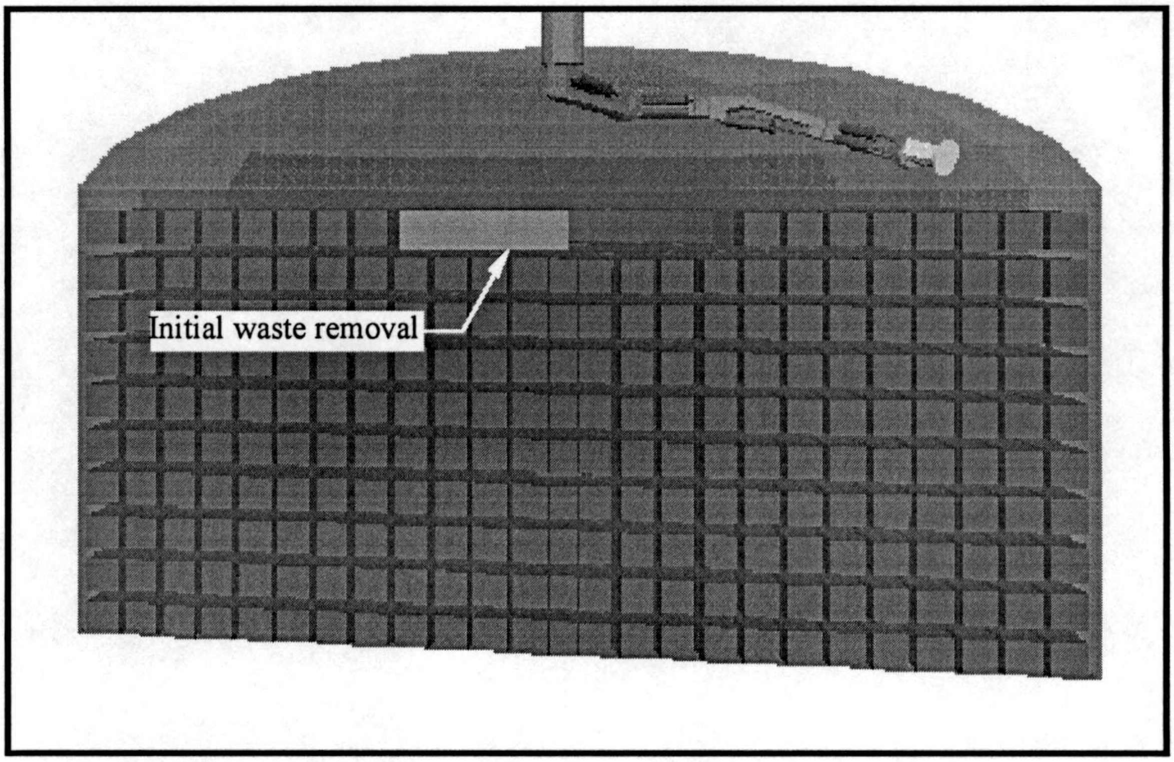


Fig. 4.8. Deployment of the LRM modified design

4.4.1 LRM Deployment of the Schilling Proposed Design

The Schilling proposed design can be deployed in two configurations, the elbow and the SCARA. More digging is necessary for elbow design deployment as shown in figure 4.6. Furthermore, for both elbow cases shown, retraction out of the tank is necessary for any further cleaning after deployment.

The elbow configuration will be introduced sequentially into the tank starting with the last link. With the first link completely in the tank, it will be necessary to clean the central top portion of the waste for further deployment. In case #1 in figure 4.6, the long reach manipulator continues cleaning while being deployed until the entire top layer disappears. Once the arm is completely deployed, no further cleaning can be done unless the arm is retracted out of the tank. In case #2, deployment is similar to case #1, but requires deeper digging near the walls. Deployment in the elbow configuration will not be considered because of its inability to proceed with the cleaning process after deployment.

In the SCARA configuration, the wrist link is introduced in the tank to perform initial waste cleanup. Once the wrist link is entirely in the tank, the central top portion of the waste is removed. The waste removed to enable the arm to be completely deployed is approximated by a cylindrical volume 4 feet high and having a 48 foot diameter. SCARA deployment along with the waste removed is shown in figure 4.7. After the initial waste removal and deployment, mining work can be continued.

4.4.2 LRM Deployment of the Modified Design

The deployment procedure of the modified design is similar to that of SCARA. The wrist link is deployed into the tank after indexing the first discrete roll joint by 90°. Some waste removal is required for the complete deployment of the arm. The waste removed to enable deployment is approximated by a cylindrical volume 4 feet high and having a 24 foot diameter. Note that this volume is only one fourth of that required for SCARA deployment. The deployment of this arm and the waste removed is shown in figure 4.8.

4.5 Workspace Coverage for the Schilling and Modified Design

This section shows the workspace covered by each arm after deployment. The column prismatic joint will be locked after deployment. Only in the elbow configuration, it will be necessary to retract the manipulator out of the tank for any further cleaning after deployment.

4.5.1 Workspace Coverage for the Schilling Design in SCARA Configuration

From the initial analysis it was shown that the Schilling proposed LRM fails to clean up a considerable central volume of the tank. This analysis was conducted for free space motion inside the tank assuming there are no obstacles. It is also assumed that once the arm is completely deployed inside the tank, the use of the column prismatic joint will be minimized during the cleanup task. Figure 4.9 shows the maximum amount of waste that can be cleaned by the Schilling arm after deployment and before the arm is retracted to continue with the mining task.

As shown by the simulation on IGRIP, once the arm is deployed in a SCARA configuration after initial waste cleanup, only two layers of waste can be cleaned up as shown in figure 4.9. The arm cannot perform any more cleanup tasks in the resulting deployed configuration. To further continue with mining, the arm will have to be

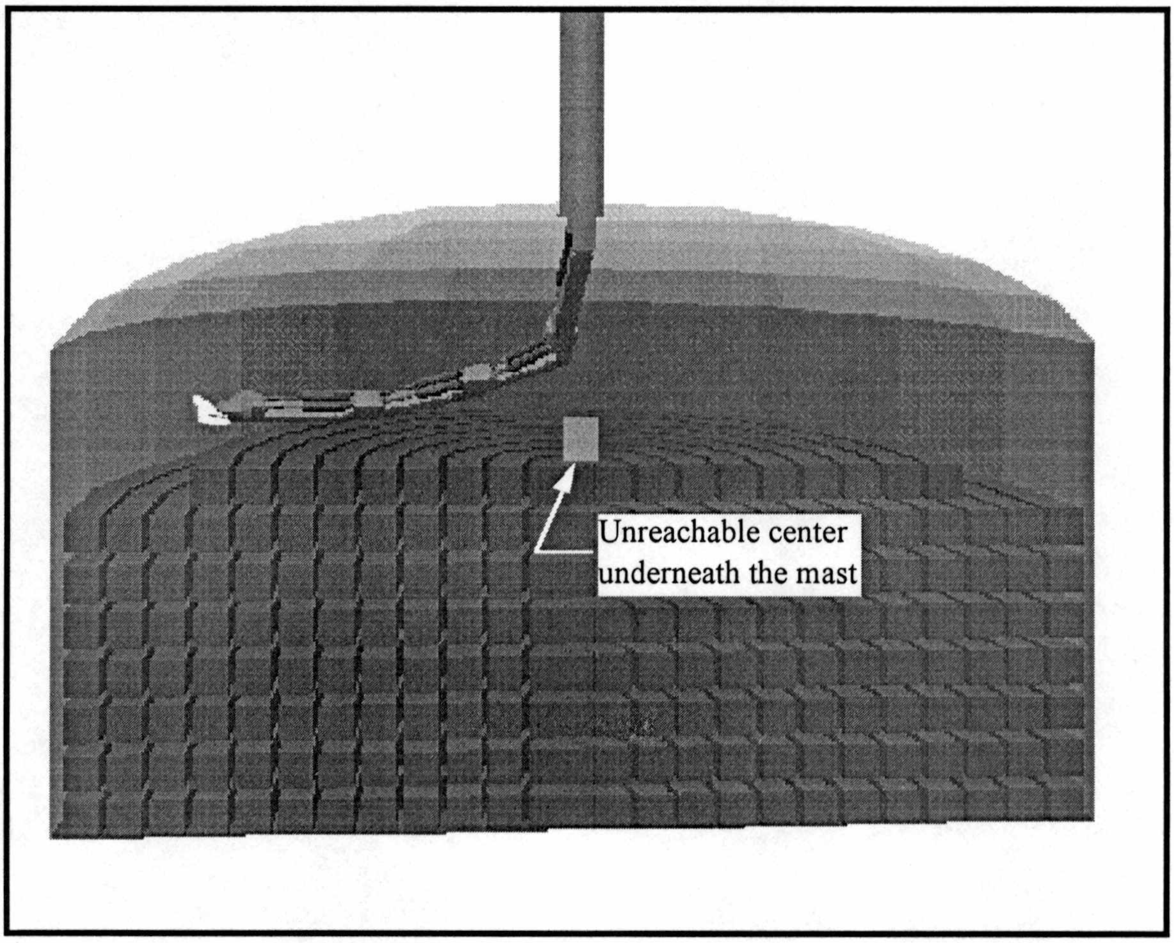


Fig. 4.9. Workspace covered by SCARA after deployment

retracted back out of the tank, and some central waste will have to be cleaned out using the column prismatic joint. The arm can then be deployed once again, and two more layers of waste can be cleaned. This sequence has to be followed for cleanup of at least 65% of the top of the tank waste volume. The arm can then be deployed in an elbow configuration to clean up the remaining waste.

Deploying the arm in the elbow configuration does not solve the workspace coverage problem. As shown in the free space motion analysis, a very large central volume of waste cannot be reached by the arm working in an elbow configuration. The arm also loses its obstacle avoidance capability. Furthermore, a lot more initial waste cleanup is needed for deployment in case of the elbow configuration.

4.5.2 Workspace Coverage for the Modified Design

After deployment of the arm, the mining of the waste is started. The column prismatic joint is locked and mining work is continued with the remaining joints. The arm starts by cleaning the first layer of waste. The arm then cleans up the second layer of waste. An approximately, 2 foot radius and 11 foot high cylindrical volume of waste cannot be cleaned at this time. But the arm can continue to clean up the remaining waste around this central waste. The arm can still access the central waste volume beneath the top 11 feet of unmined central waste. Removal of this waste will allow the top waste to collapse to the bottom, allowing it to be cleaned up. Figure 4.10 shows the central waste that cannot be cleaned immediately after initial deployment. Note that this unreached volume can be cleaned if the upper arm #2 length is reduced by approximately 2.5 feet.

All the waste clean up is performed without the use of the column prismatic joint for vertical motion and without retracting the arm back out of the tank. Figure 4.11

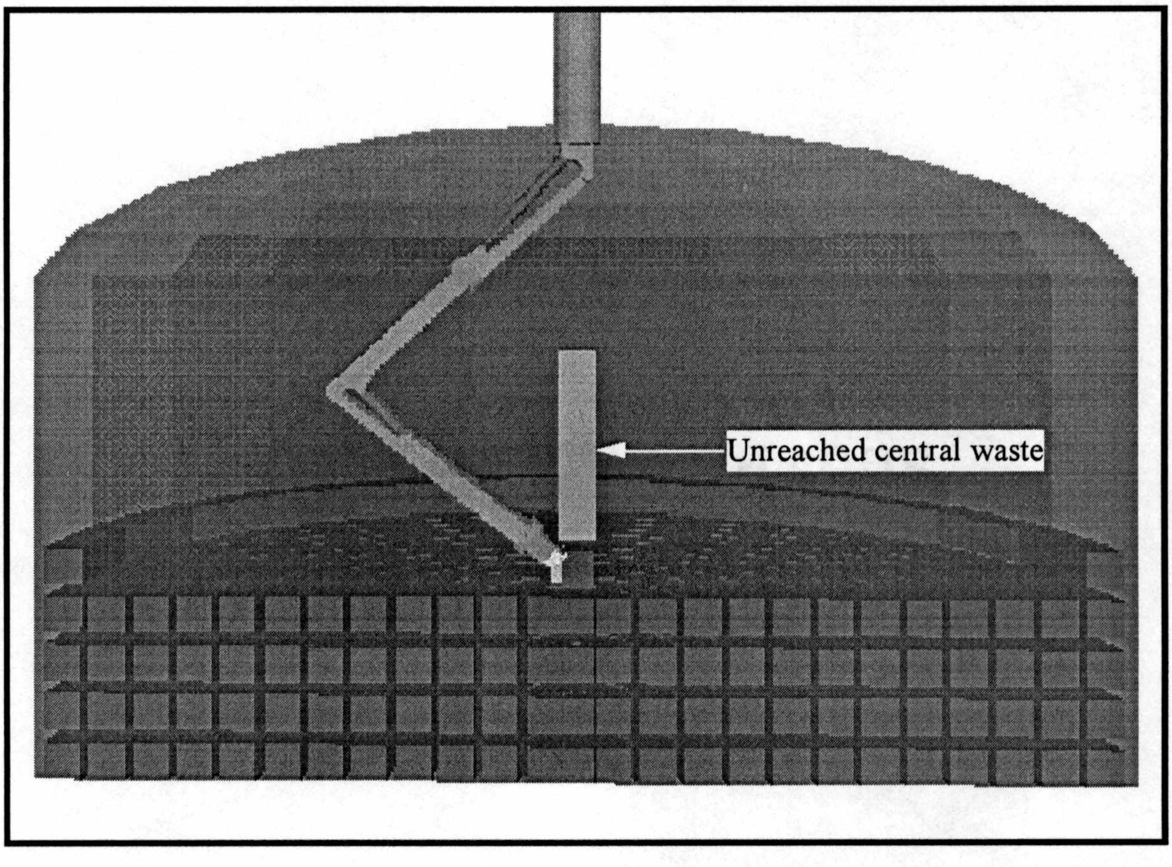


Fig. 4.10. Unreachable central waste by the modified design

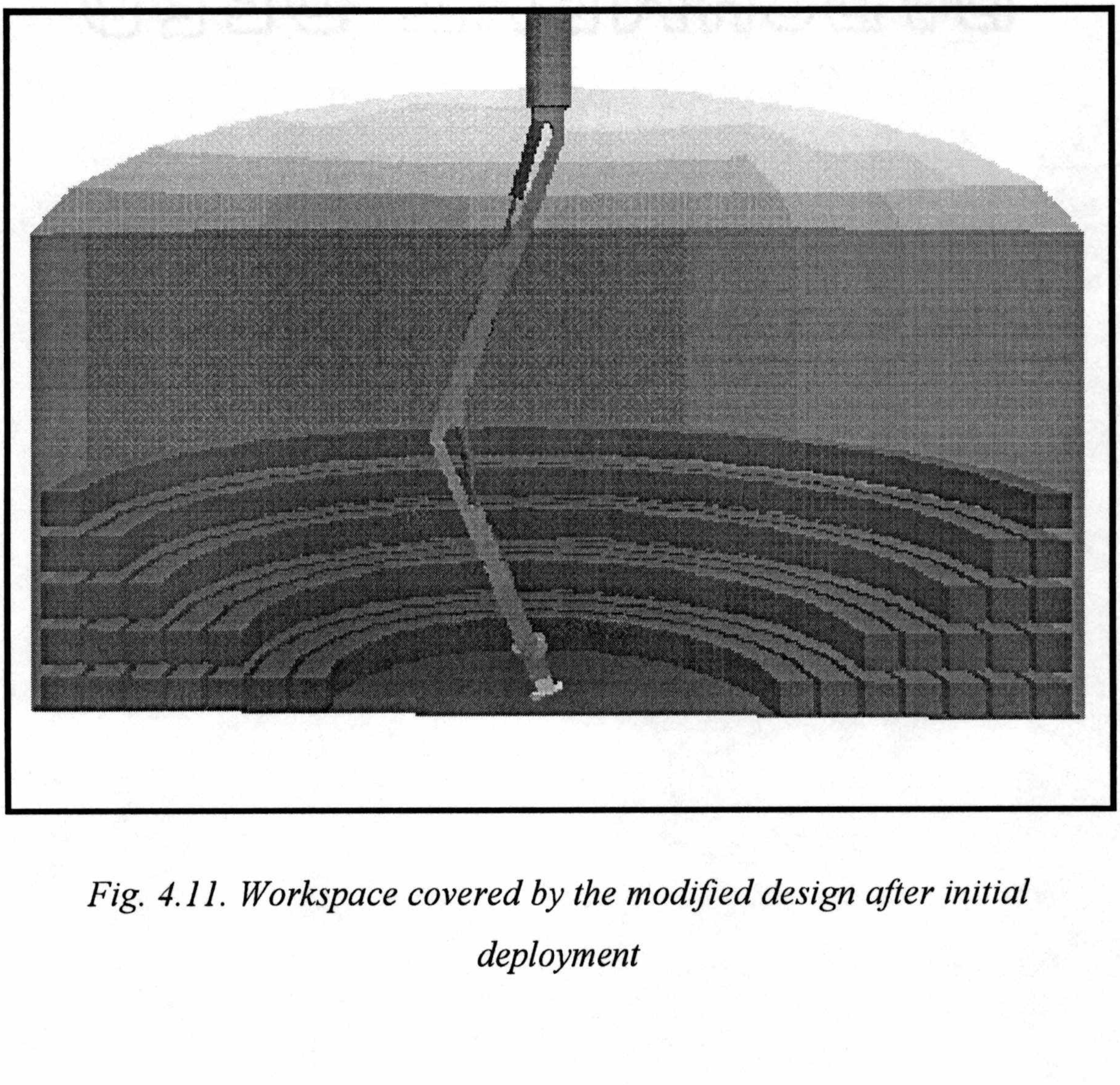


Fig. 4.11. Workspace covered by the modified design after initial deployment

shows the maximum amount of waste that can be cleaned up after the initial deployment and without the use of the column.

As shown in Figure 4.12, to continue cleaning up the remaining waste on the inner bottom perimeter of the tank, the arm is lowered a little further inside the tank using the column prismatic joint. Then the remaining waste is cleaned up. This design of the LRM can access the entire tank volume after the initial deployment without retracting the arm back out of the tank and without the use of column prismatic joint for vertical motion during mining.

4.5.3 Workspace Coverage for the Schilling Design in Elbow Configuration

Initially is was found graphically that the elbow configuration fails to cover the entire tank volume with the arm deployed inside the tank. At a later time when the elbow configuration was simulated on IGRIP, it was found that the arm should be retracted out of the tank for any further cleaning. Figure 4.13 shows the necessity of retraction of the manipulator for any further cleaning after case #1 deployment in elbow configuration.

In figure 4.14 it is shown that continuous retraction is necessary to reach the central top portion of the waste.

In figure 4.15 the arm is slightly retracted out of the tank to reach a very low level central waste. It is possible to clean the entire tank in elbow configuration but with continuous retraction and use of the mast prismatic joint.

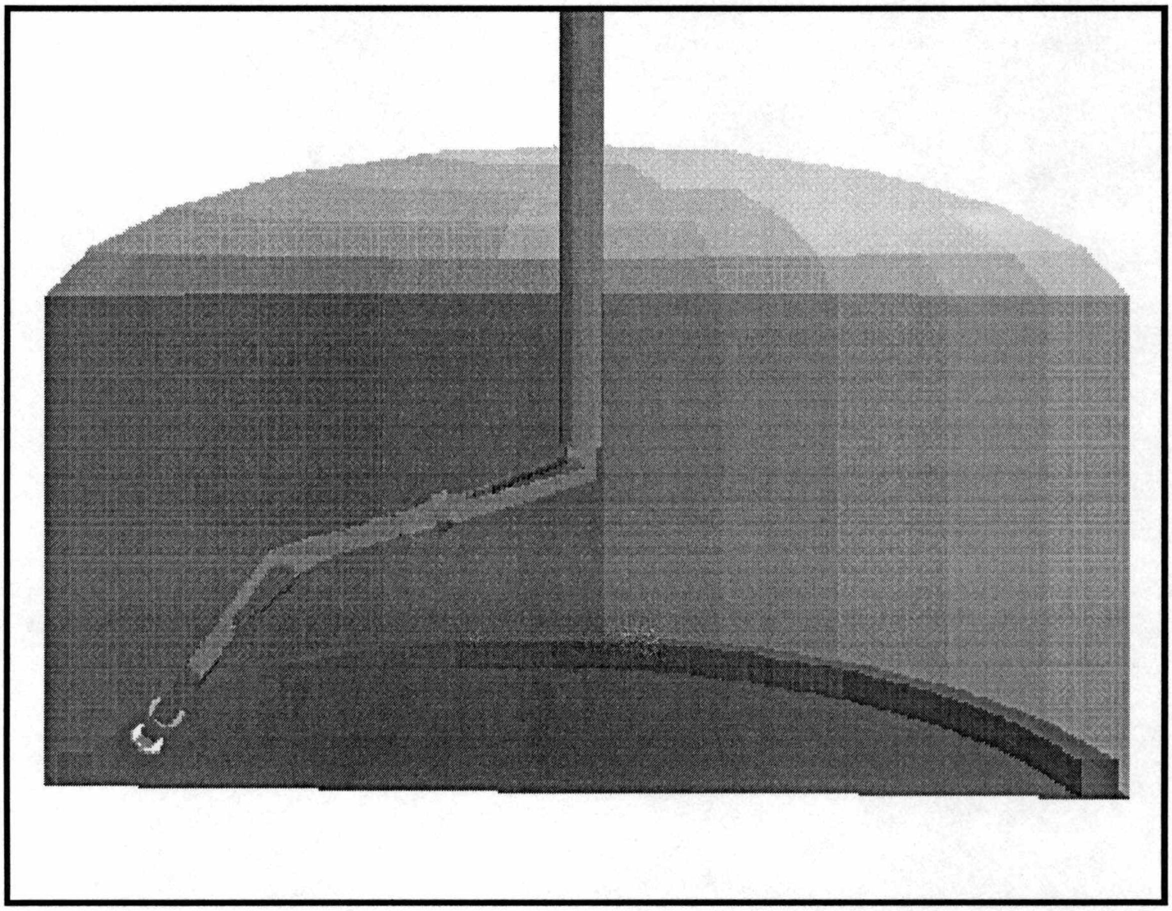


Fig 4.12. Workspace covered by the modified design after lowering the column

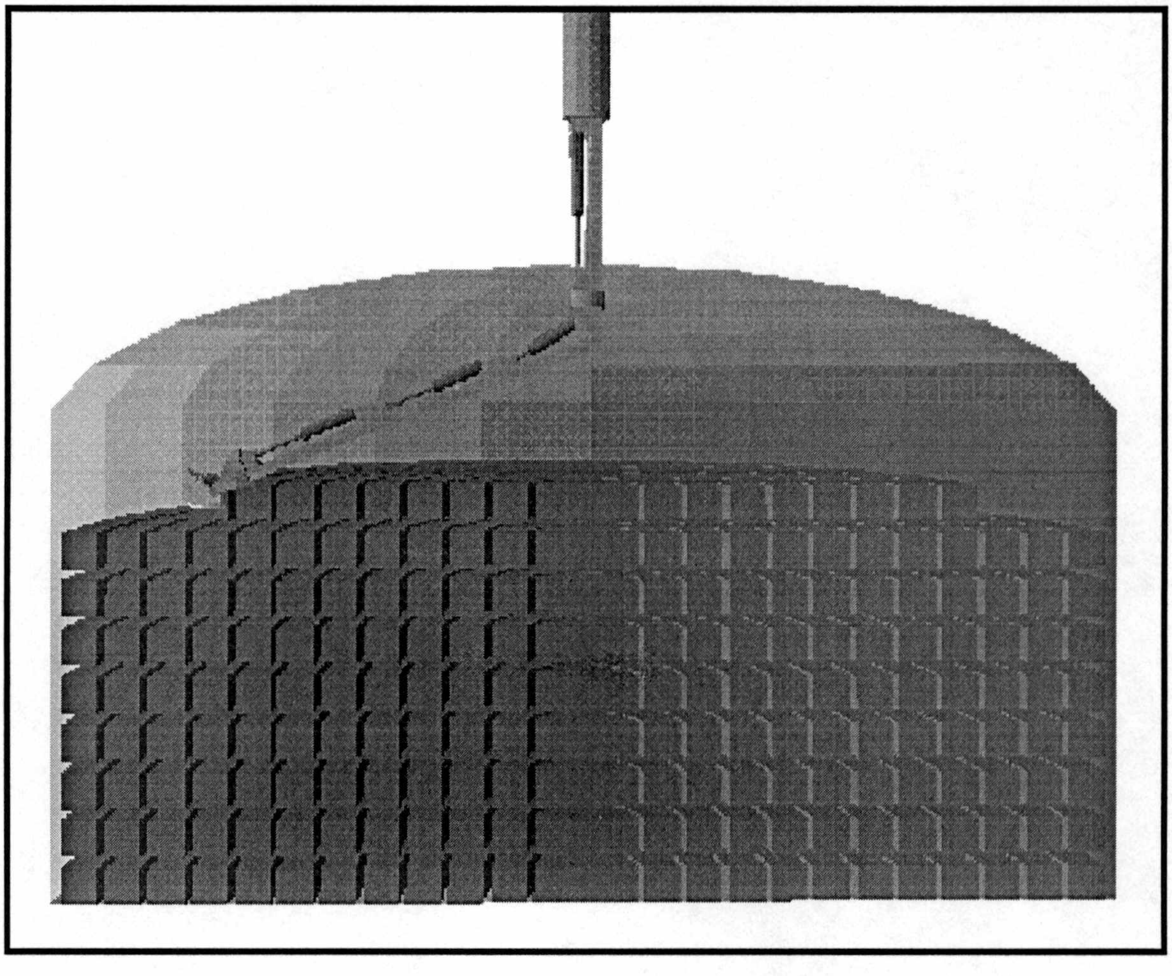


Fig 4.13. Retraction is necessary after deployment for any further cleaning

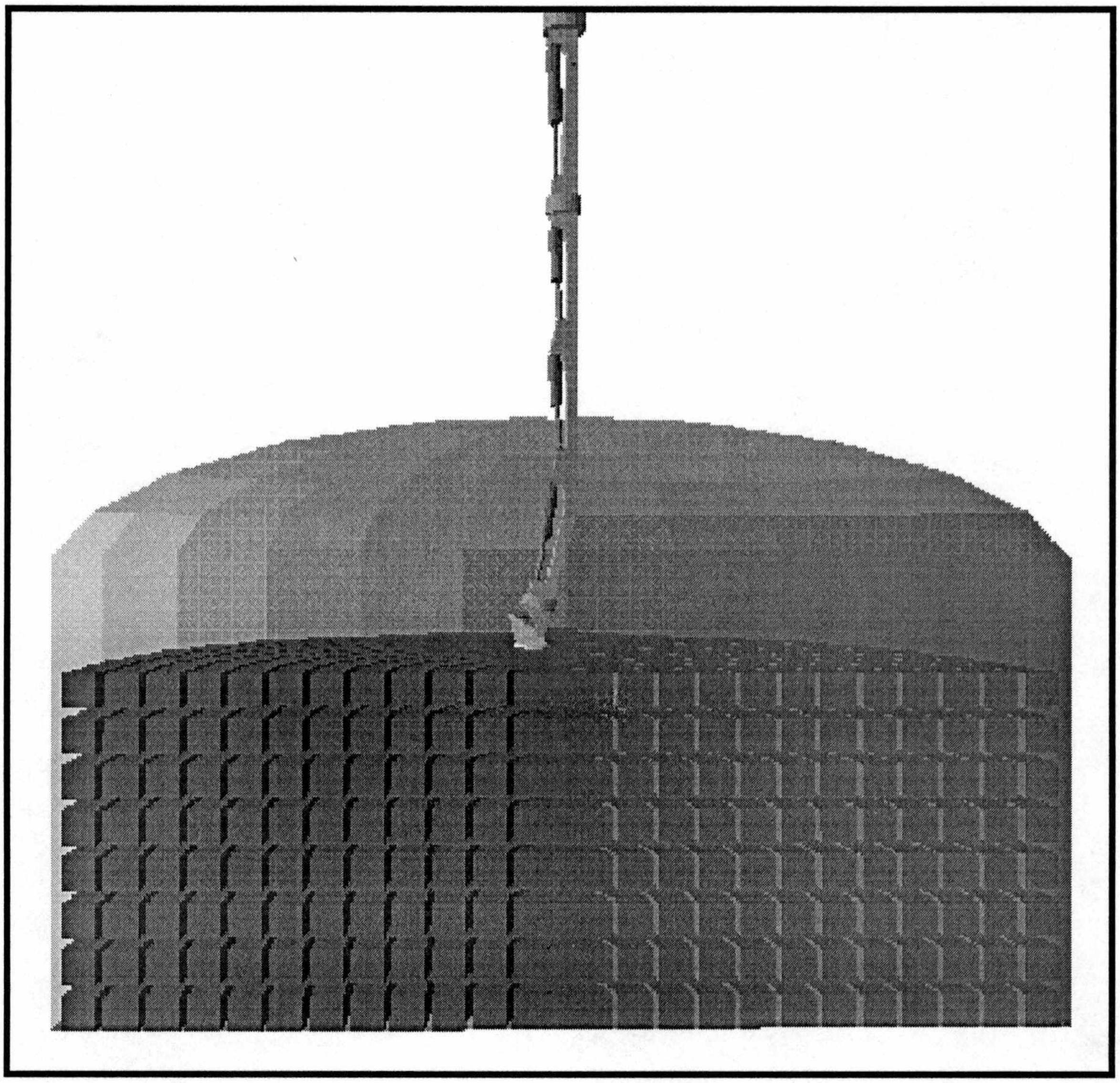


Fig. 4.14. Continuous retraction is necessary to reach the central waste

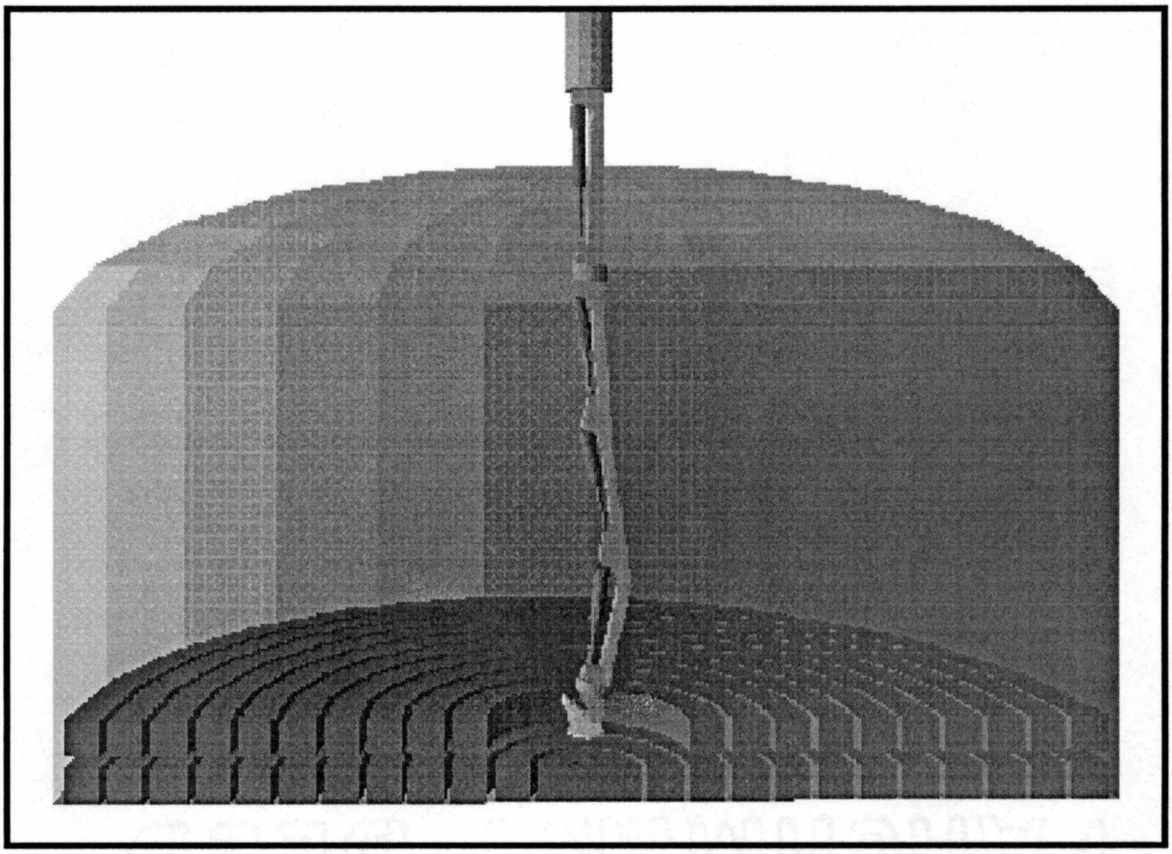


Fig. 4.15. Arm is slightly retracted to reach a very low waste level

4.6 Obstacle Avoidance

4.6.1 Schilling Design

One of the drawbacks of the Schilling design is the inability to avoid vertical risers in elbow configuration. Note, however, that in SCARA configuration, the Schilling design can avoid vertical obstacles because of the present yaw joint. Considering the cleanup to be done in the SCARA configuration around obstacles, it can be done only down to the last 8 feet from the bottom of the tank. Then the Schilling design has to be used in elbow configuration because of the mast stroke limitation. It is very clear, as shown in figure 4.16, that the elbow configuration can not avoid vertical obstacles at any joint values.

4.6.2 Modified Design

The modified design possesses a great obstacle avoidance capability in both the vertical and horizontal directions. The modified design also is capable of keeping the arm away from the waste during cleaning, which reduces the possibility of contamination by the arm. In figure 4.17 one example of vertical obstacle avoidance capability of the modified design is given.

4.7 Comparison of the Schilling and the Modified Design

As described above, the Schilling proposed design has limitations in regards to several design requirements. If the Schilling arm is entirely deployed in the tank, a small portion of the waste can be cleaned before the necessity of retraction out of the tank. This condition implies that the mast prismatic joint is used continuously in the cleaning process, a situation which should be avoided. Also, vertical obstacle avoidance capabilities of the Schilling design is limited to the SCARA configuration, which can not

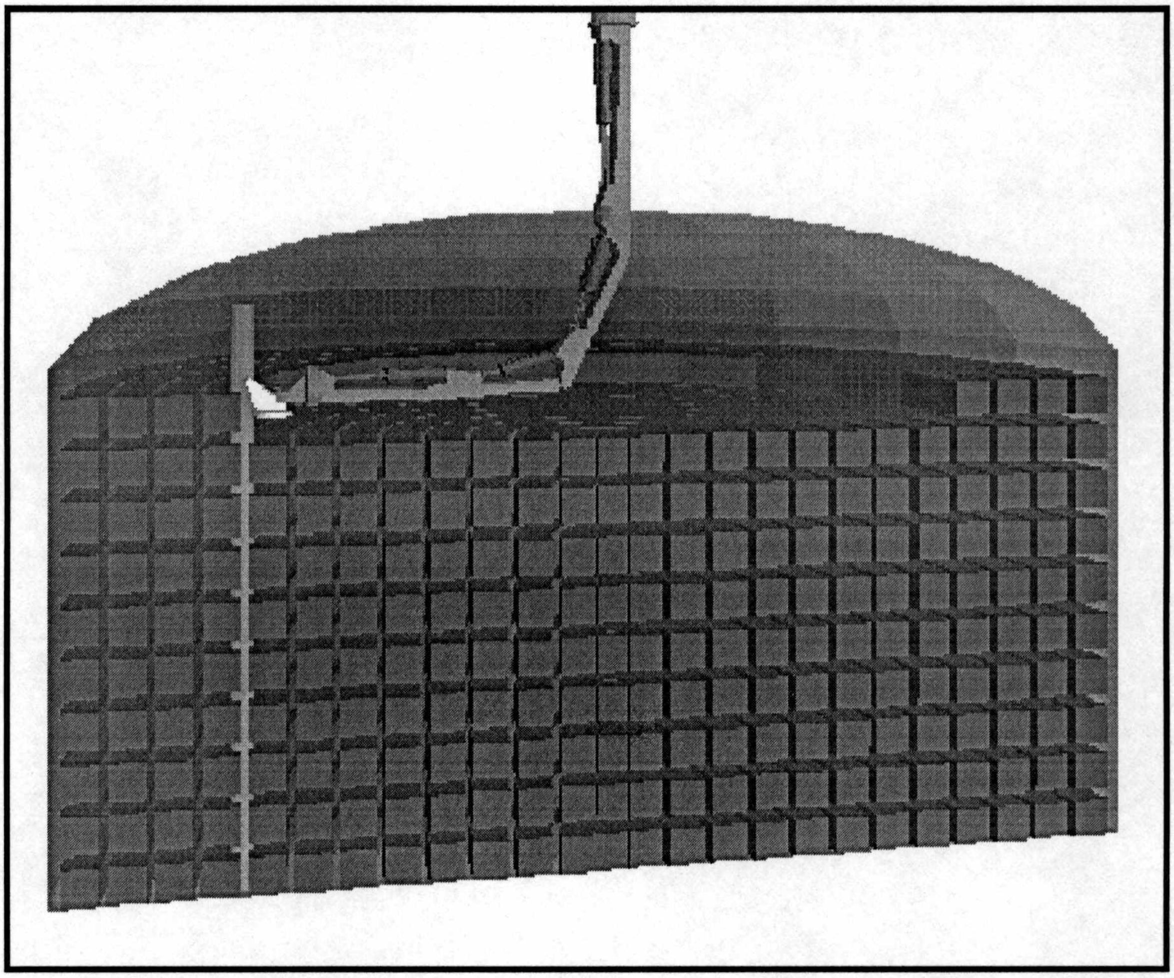


Fig. 4.16. Vertical obstacle avoidance of the Schilling design in elbow configuration

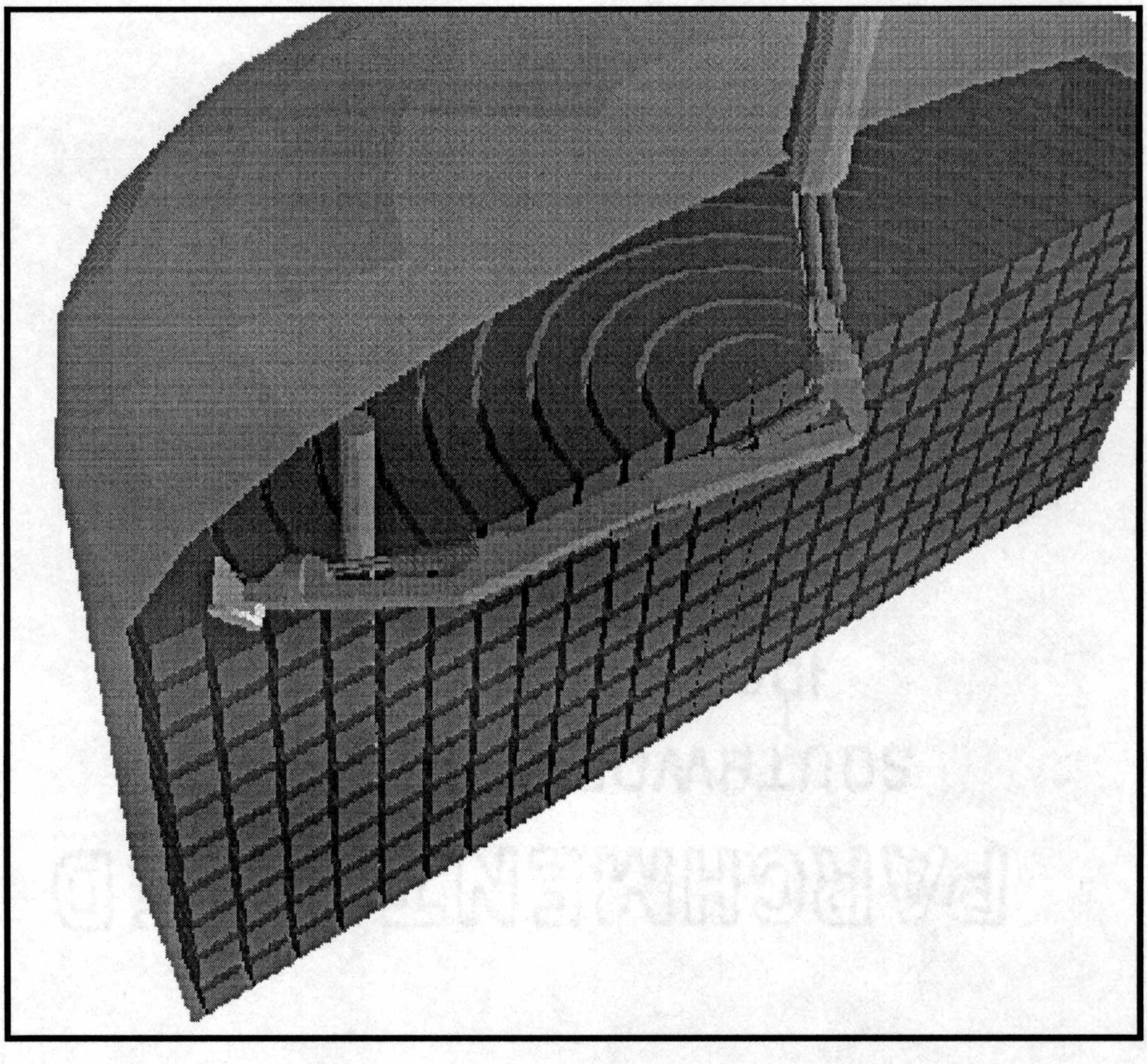


Fig. 4.17. Vertical obstacle avoidance of the modified design

be used for the cleanup process efficiently. A summary comparison of the proposed Schilling design and the modified design is shown in Table 4.3.

4.7.1 Waste Removal Necessity for Deployment

The Schilling proposed design can not be deployed in SCARA configuration unless sufficient waste is removed during deployment. The amount of waste to be removed during deployment is approximated by a cylindrical volume 4 foot high and having a 48 foot diameter as shown before in figure 4.7. Also, the Schilling design can be deployed in elbow configuration but with much more waste removal required, as shown before in figure 4.6.

On the other hand, with the same link lengths and joint motion ranges as that of the Schilling design, the modified design requires waste removal approximated by a cylindrical volume 4 foot high and having a 24 foot diameter as shown before in figure 4.8, or about one fourth the waste volume required by the Schilling design in SCARA configuration.

4.7.2 Waste Retrieval Sequence

The Schilling design in elbow configuration will not be considered for waste retrieval after deployment because of the necessity of retraction out of the tank. After deployment, the Schilling design in SCARA configuration can continue the cleaning operation, starting with the rest of the first layer, and continuing with the second layer. At the end of the second layer cleaning, the end effector will not be able to reach the central waste immediately below the mast because all the joints are in the same plane. By raising the mast it is possible to clean a small portion of the third layer waste near the wall. The clean up process ends at that point unless the arm is retracted out of the tank as shown before in figure 4.9.

Table 4.3 Comparison of the Schilling and modified LRM designs

Description	Schilling Design	Modified Design
Configuration	R-E-P-P-P-P-P	R-E-P-P-Y-P-P
Waste Removal for Deployment	Approximated as a cylinder 4' high and 48' diameter for SCARA configuration	Approximated as a cylinder 4' high and 24' diameter, one fourth the volume of Schilling.
Workspace Coverage after Deployment	Workspace coverage is less than two layers of waste without retraction	Can clean up more than 70% of tank volume after deployment and locking the mast
Use of Prismatic Joint	Have to use prismatic joint during mining and for retraction after cleaning two layers of waste.	Have to use prismatic joint only twice during the whole operation.
Obstacle Avoidance	Avoid vertical obstacles only in SCARA configuration	Obstacle avoidance in both horizontal and vertical direction.
Working Configuration	Cannot work in elbow configuration until nearly 75% of waste under mast is cleaned	Arm can work in an indexed or unindexed roll joint configuration.
Closeness to Waste	Entire arm close to waste in SCARA configuration	Arm always away from waste.
Retraction out of Tank	Retraction necessary once every two layers of waste are cleaned	No retraction necessary until the entire tank is clean.
Time for Cleanup	Time consuming because of the continuous retraction and the use of the mast prismatic joint	Saves time because it is only deployed once for entire tank cleanup.

After the modified design deployment, the rest of the first layer can be cleaned, as well as all the reached layers below. However, an approximately 4 foot diameter and 11 foot high cylindrical volume directly beneath the mast cannot be cleaned, as shown before in figure 4.10. This volume of inaccessible waste is assumed to collapse once the waste below it is cleaned up. After deployment of the modified design and locking of the mast prismatic joint, cleaning can be continued, and more than 70% of the waste can be retrieved as shown before in figure 4.11. This strategy assumes the collapse of the waste below the mast; Otherwise, the upper boom #2 should be shortened by 2.5 feet to access that waste. The remaining waste around the bottom can then be cleaned up by lowering the arm a little more inside the tank.

4.7.3 The Use of Mast Prismatic Joint

The continuous use of the mast prismatic joint should be avoided due to its very high inertia. Unfortunately, the Schilling design in SCARA configuration requires continuous movement of the mast because of layer-by-layer cleaning and because of the necessity of retraction for further cleaning after cleaning the first two layers. If the elbow configuration was used, a large central waste volume should be cleaned initially, then the mast prismatic joint could be used for further cleaning.

The mast prismatic joint of the modified design can be locked and still allow the arm to clean more than 70% of the tank waste after deployment (as shown before in figure 4.11). After this amount of waste is removed, the mast prismatic joint should be repositioned and locked to clean the amount of waste left in the tank (as shown before in figure 4.12).

4.7.4 Obstacle Avoidance

The Schilling design is unable to avoid vertical risers in elbow configuration as shown before in figure 4.13. The Schilling design in SCARA configuration can avoid vertical obstacles because of the included yaw joint. The drawback to the SCARA configuration is the closeness to the waste and the possibility of collision with the waste during the cleaning process.

On the other hand, the modified design possesses a great obstacle avoidance capability in both the vertical and horizontal directions. The modified design also is capable of keeping the arm away from the waste during cleaning, which reduces the possibility of contamination by the arm.

Chapter 5

The Final Design and Orientation Requirements

5.1 Introduction

As the Tank Waste Retrieval project progressed, design requirements were tightened to include constraints on the orientation of the end effector. These constraints required that the manipulator would have the ability to maintain both horizontal and radially pointing tool plate orientation throughout the volume of the tank. In this way, the surface of the cleaning tool could be positioned parallel with the surface to be cleaned, whether it be the wall of the tank or the top surface of the waste. The proposed Schilling and modified designs were analyzed to determine whether these constraints could be satisfied. By this time, it had been decided that discrete roll joints would not be used. Therefore, the SCARA design was not considered in these studies. The elbow configuration of the Schilling design became the dominant design with retraction allowed. A change in link lengths made it possible for the LRM in elbow configuration to be deployed without initial digging. Allowing retraction out of the tank and the new orientation requirements had a great impact on the choice of design. Taking all new requirements into consideration the elbow design became the simplest and most reliable design.

5.2 Orientation Constraints of Elbow and Modified Designs

When the orientation requirement was introduced it became necessary to investigate both the Schilling and the modified designs. Graphical and geometrical calculations were done to identify the areas where the end effector could not maintain orientation.

5.2.1 Elbow Design

Ideally, once the manipulator is lowered into the waste tank it would not be retracted. If this is to be the case with the elbow design, complete cleaning would be impossible. That is, once the elbow configuration of the Schilling design has been completely deployed in the tank, there are certain regions in which it cannot reach without colliding with the waste material.

Assuming retraction is to be allowed, the elbow configuration of the manipulator has the ability to reach all areas of the tank while maintaining a horizontal orientation of the tool plate. However, there were deficiencies in the ability of the manipulator to maintain a radially pointing orientation. As shown in Figure 5.1, there is an annular core which is unreachable by a radial pointing tool plate, even assuming retraction is allowed.

5.2.2 Modified Design

Once again, assuming no retraction, UT's modified design has the ability to reach the entire volume inside the tank, even after complete deployment of the manipulator. However, the reachable space while maintaining horizontal or radially pointing tool plate is somewhat restricted. The unreachable space assuming a horizontal tool plate includes the area near the walls of the tank as well as the central core, as shown in Figure 5.2.

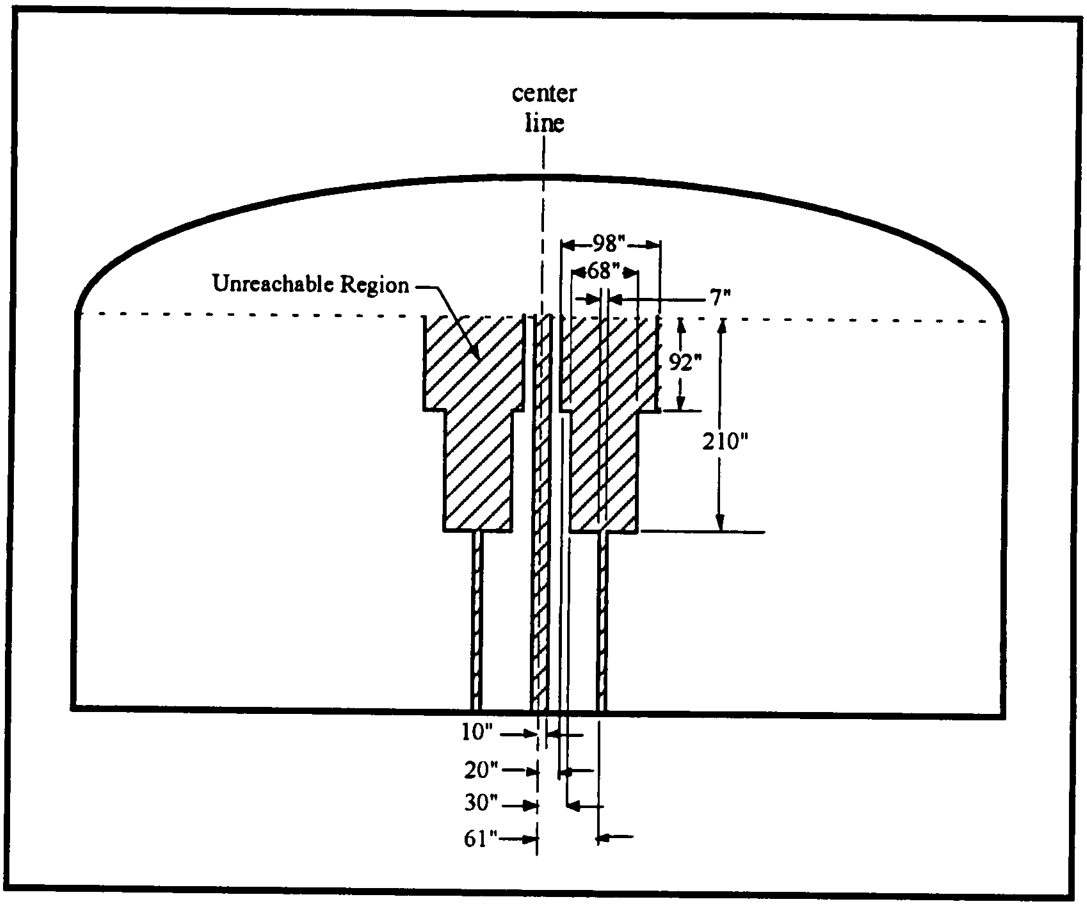


Fig. 5.1. Unreachable region for elbow design with radial tool plate

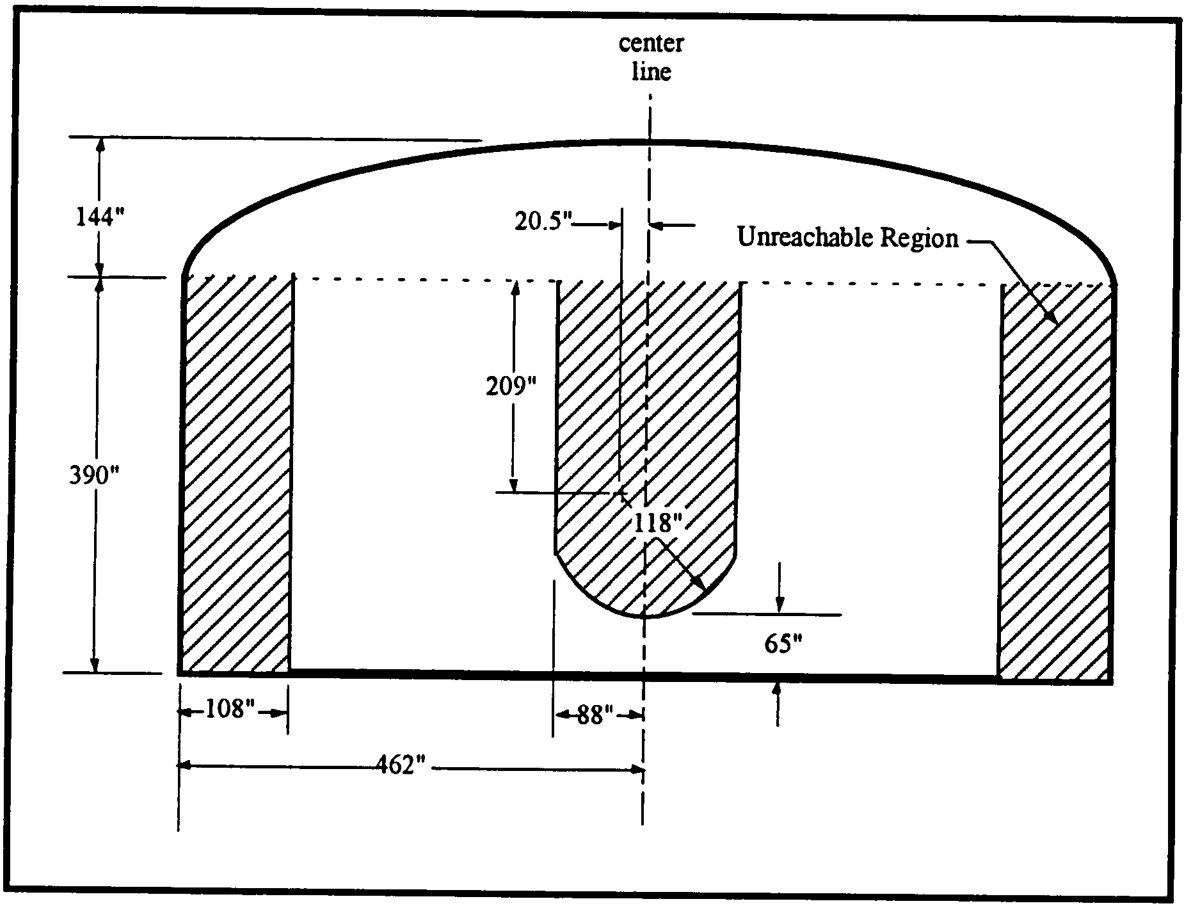


Fig. 5.2. Unreachable region for modified design for horizontal tool plate

Maintenance of a radially pointing configuration prohibits the manipulator from reaching the central area as well as the bottom corners, as shown in Figure 5.3.

If retraction is not prohibited, the workspace with orientation constraints would be much less restrictive. However, the corresponding reachable workspace of the elbow design would be comparable in size and characteristics. Therefore, considering the fact that the elbow design is simpler to construct and control, it would be a more reasonable choice of designs.

5.3 Suggestions to Improve Orientation Capabilities

There are several ways to improve the workspace coverage with constant orientations of the tool plate. The intention of the improvement is to make the modified design compete with the elbow design in orientation requirements, while maintaining the unnecessity of retraction during mining process.

5.3.1 Increasing Joint Limits

one method would be to increase the joint limits of the GPM. Increasing the maximum bound of the joint angles to 115° would reduce the diameter of the unreachable central cylinder for both tool plate orientations, as illustrated in figures 5.4 and 5.5.

5.3.2 Attaching Right Angle Extension

Attachment of a right angle extension between the tool plate and end effector would provide a horizontal surface in LRM configurations in which the tool plate is radial. In other words, regions in which the tool plate could formerly be oriented radially would be the same as those in which they could now be maintained horizontally. Therefore, assuming the right angle tool could be attached when needed, the total reachable volume with a horizontal tool plate would be improved, as shown in figure 5.6.

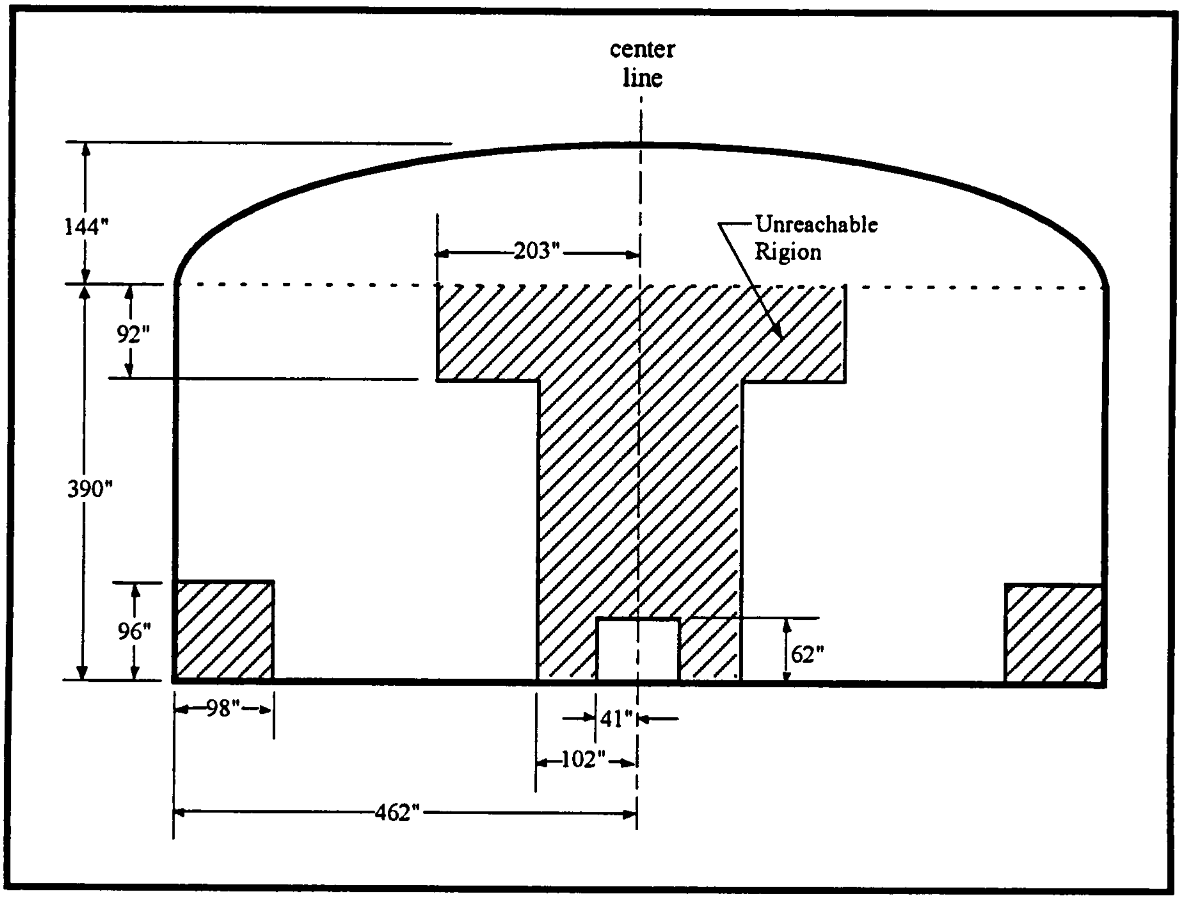


Fig. 5.3. Unreachable region for modified design for radial tool plate

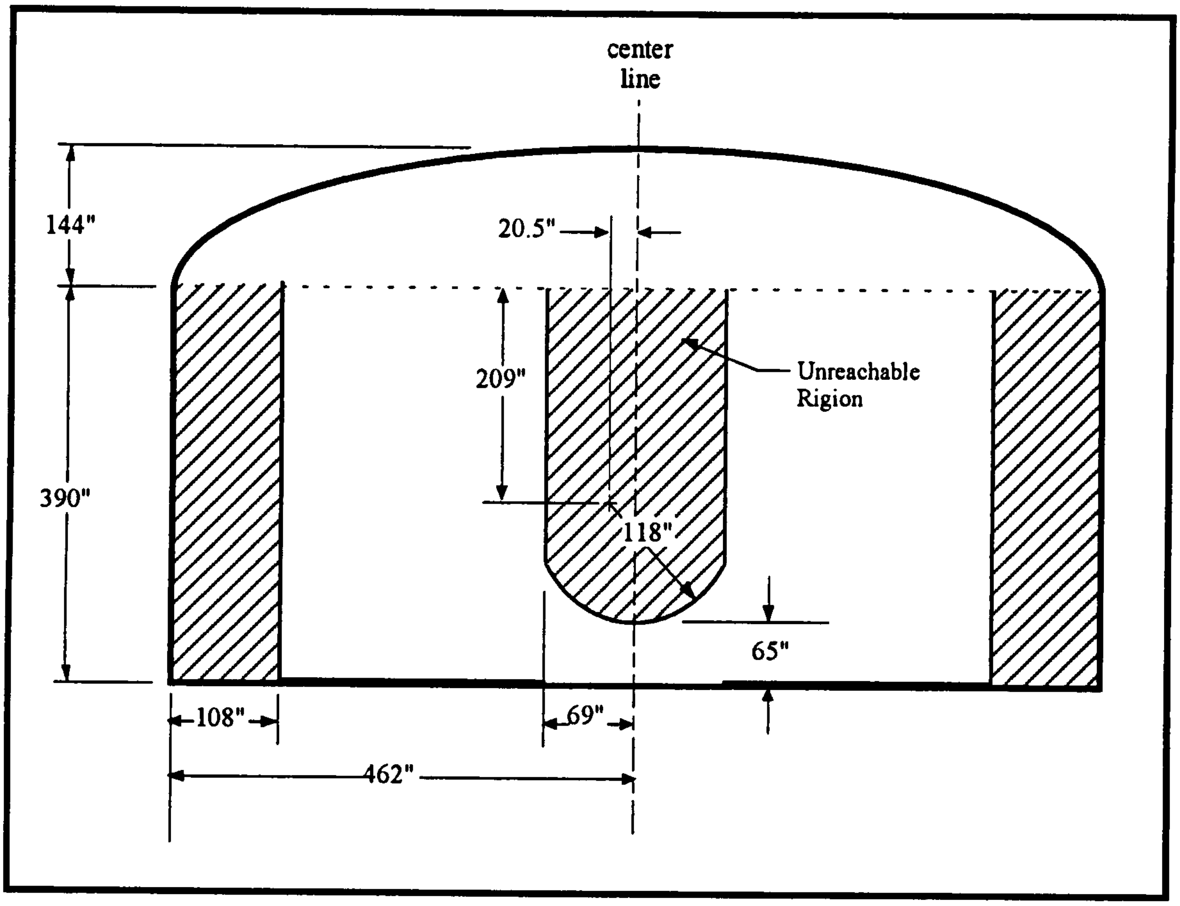


Fig. 5.4. Unreachable regions for modified design with 115° joint limits with horizontal tool plate

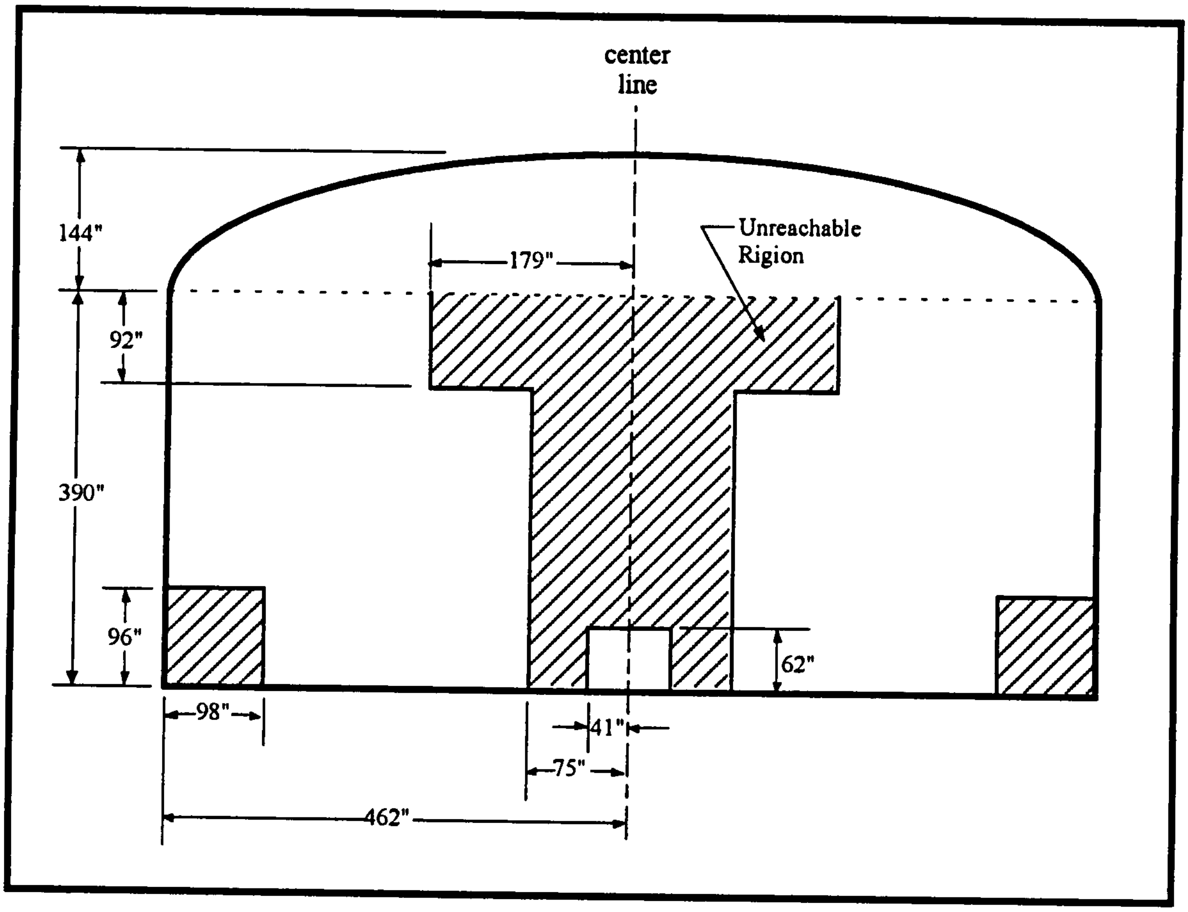


Fig. 5.5. Unreachable regions for modified design with 115° joint limits with radial tool plate

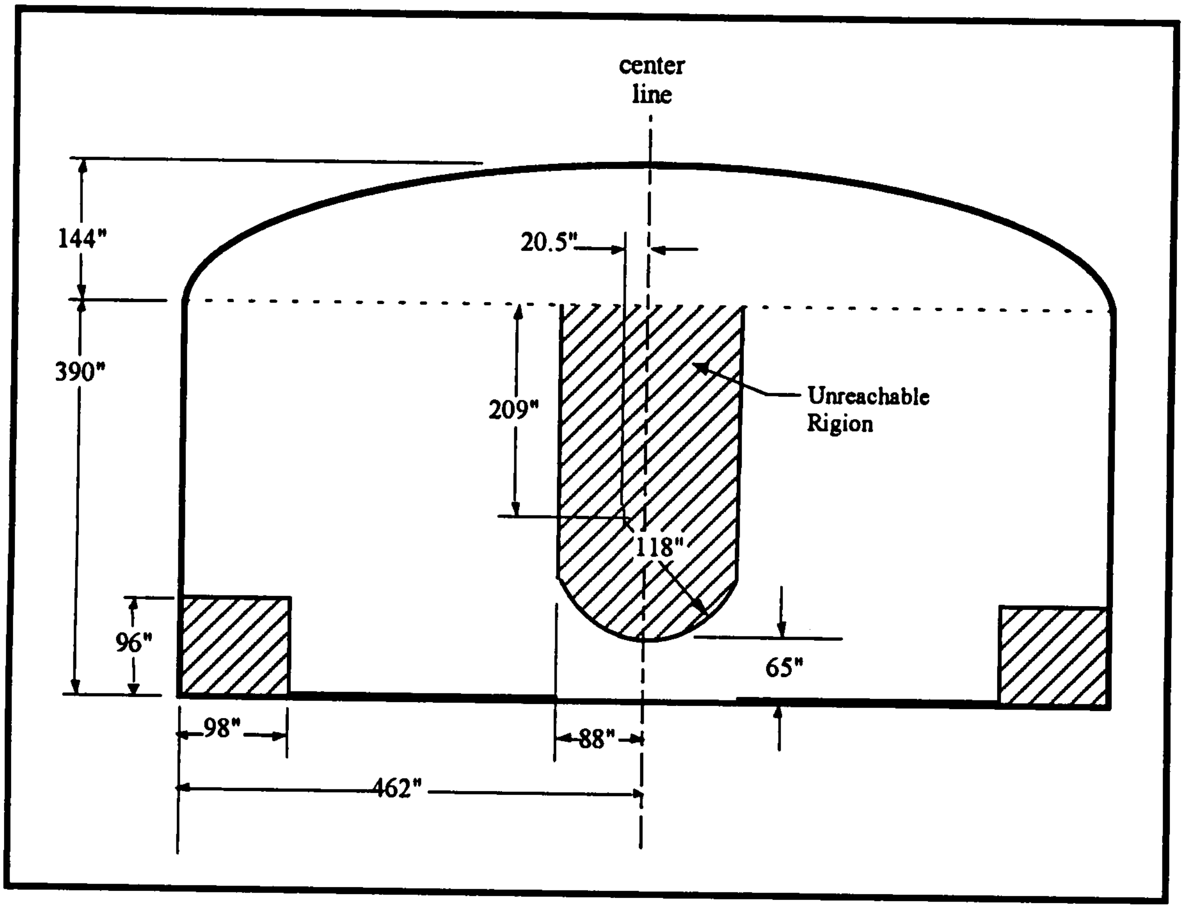


Fig. 5.6. Unreachable region for modified design with tool offset and horizontal tool plate

5.3.3 Changing the Design

If maintaining a horizontal orientation of the tool plate throughout the tank is most important, while radial pointing is only important near the walls, a new configuration proposed by UT would satisfy these constraints most effectively. The new proposal would not only have a new joint arrangement, but also new link lengths and joint limits. The new joint configuration would be roll-pitch-yaw-yaw-pitch-pitch. The requirements on link lengths would be that upper arm #1 would be longer than the forearm, and the joint limit range for elbow #1 and #2 would be changed to 0° to 120°. These new modifications are shown in figure 5.7, as well as the joint positions necessary to reach the central area beneath the mast.

5.4 Analytical End Effector Orientation

As mentioned before, geometrical solutions were taken in consideration for tool plate orientation studies. In this section an analytical solution will be introduced with an intention to be used in the original simulation program.

The modified design frame assignment and DH table used in this analysis is shown in appendix B. Several constraints should be satisfied to maintain tool plate orientation in horizontal and pointing out configurations. The coordinates frame of the base and the end effector are shown in figure 5.8.

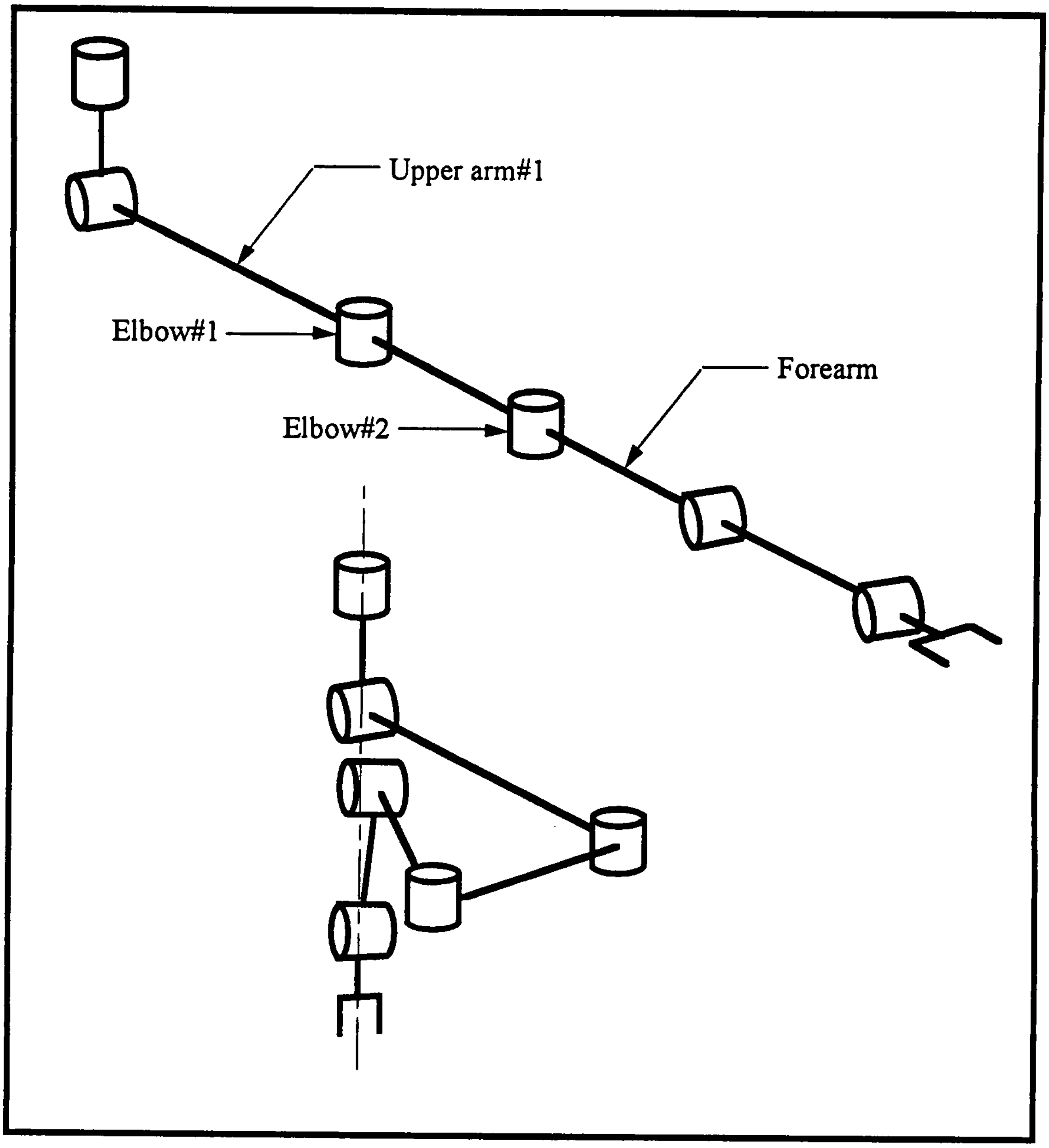


Fig. 5.7. Modified design #2 is shown to reach the central waste below the mast

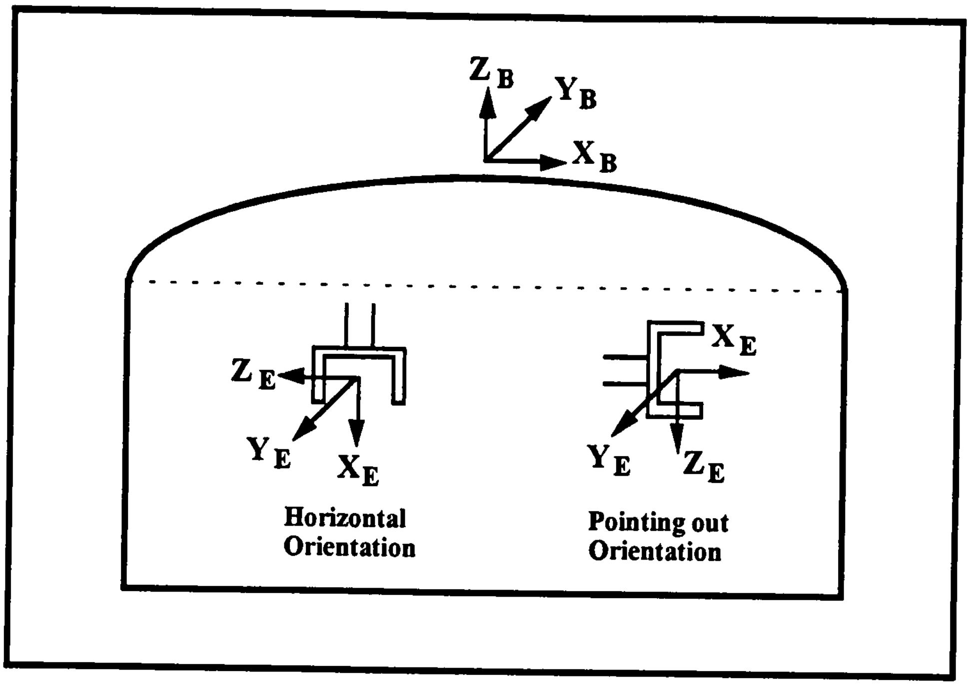


Fig. 5.8. Assignment of base and end effector coordinate frames

5.4.1 Constraints for Horizontal Orientation

For maintaining the horizontal orientation of the end effector, equation (4.1) must be satisfied. Equation (4.1) implies that the X coordinate of the end effector must be in the same direction of the Z coordinate of the base.

$$X_{E_Z} = (-c_3c_4c_5 + s_3s_5)c_{67} + c_3s_4s_{67} = -1 \qquad (4.1)$$

Relative to the complexity of solving equation (4.1) it can be replaced with two constraint equations. Equations (4.2) and (4.3) require that the projection of both $\mathbf{Z}_E$ and $\mathbf{Y}_E$ on $\mathbf{Z}_B$ to be zero.

$$Z_{E_Z} = s_3c_4c_5 + s_3s_5 = 0 \tag{4.2}$$

$$Y_{E_Z} = (c_3c_4c_5 - s_3s_5)s_{67} + c_3s_4c_{67} = 0 \tag{4.3}$$

5.4.2 Constraints for Pointing out Orientation

There is only one constraint for the tool plate orientation to be in the vertical direction. The zero projection of $\mathbf{X}_E$ on $\mathbf{Z}_B$ is that constraint given by equation (4.4). But this constraint does not guarantee the tool plate to be radially pointing out, so another constraint given by equation (4.5) must be satisfied. In this constraint the slope of the tool plate pointing vector in the horizontal plane must be equal to the slope of its postion vector.

$$X_{E_Z} = (-c_3c_4c_5 + s_3s_5)c_{67} + c_3s_4s_{67} = 0 \tag{4.4}$$

$$\frac{X_{E_Y}}{X_{E_X}} = \frac{Y}{X} \tag{4.5}$$

This analytical solution was never implemented in a real program relative to the short time given for the complete analyses.

5.5 Summarized Comparison of Proposed Designs

Although the UT design had advantages over the original design in the areas of deployment and cleaning without retraction, orientation constraints and unrestricted retraction made the original design sufficient. The final selection for the design to be developed further was made in light of these comparisons.

As discussed before, the new link lengths make complete deployment before mining begins possible for either manipulator. However, once the elbow design has been completely extended inside the tank, it must be frequently retracted in order for clean-up to proceed. The modified design proposed by UT, on the other hand, can continue and complete clean-up while remaining fully inside the tank. The SCARA configuration also has the ability to clean-up after deployment; however, it must be retracted and reconfigured to the elbow configuration in order to reach the bottom volume of the tank.

Of the designs discussed, the elbow design has the least obstacle avoidance capability. It is impossible for this design to reach areas behind vertical risers, making it necessary to cut risers at every level in order to completely clean that level. The modified design and SCARA configuration have the ability to clean around these obstacles.

None of the designs have exceptional orientation ability, but if retraction is allowed, the elbow design has the largest workspace. If however, retraction is not allowed, the modified design is the only one of the two which can complete the cleanup at all. Recall that if certain orientations are required only in certain parts of the tank, the second UT modified design is the most appropriate.

As for ease of construction and control, the elbow design is by far the best. Its all-pitch configuration implies redundancy resolution must only be accomplished in a single plane. In addition, when the redundancy provided by this configuration is not needed, one or more joints may be locked to further simplify control without affecting the reachable workspace. In addition, the links of the elbow configuration are all the same, so that only a single design for all links is needed. Furthermore, joints in this design are not subjected to yaw stresses due to the weight of the lower links. Other manipulator designs would require complicated control and redundancy resolution schemes as well as nonstandard link designs, and undesirable high yaw stresses are induced in their joints.

After consideration of each of these points, the elbow design was chosen for further development. It was decided that since retraction would be allowed during the cleanup, the workspace coverage with constant tool plate orientation of the elbow design as well as its ease of design made it the favored design.

Chapter 6

Old Testbed Analysis and Software Development

6.1 Introduction to the Old Testbed

The main reason for the analysis of the old testbed is to demonstrate the software development needed to control the final design of the long reach manipulator. It is assumed that the main part of the software is going to communicate with the simulation software IGRIP.

This testbed was originated at Oak Ridge National Laboratory (ORNL) to demonstrate the technology applicable to remediation of waste stored in large underground tanks. The telerobotic manipulator in the old testbed has 28 feet reach, a 2500 pound lift, and a 0.05 inch positioning accuracy. Although the eventual remediation system will require deployment from an overhead bridge, ORNL utilized an existing floor-mounted manipulator for the test stand. A reassembled and refurbished Spar RMS 2500 manipulator system shown in figure 6.1 was used. This manipulator was designed to operate in a reactor environment which made it suitable for waste tank remediation, in addition to long-reach, high-positioning accuracy and high lift capacity[4].

The six degrees of freedom second-generation TITAN II will act as the dexterous manipulator. The Schilling servo-hydraulic telerobotic manipulator TITAN II was developed for extreme environments. Advantages such as high-resolution bilateral force feedback, advanced telerobotic control, and end effector interchange made the TITAN II functional in poor environments. TITAN II features titanium construction, high payload, wide range of motion, intuitive control, and smooth fluid operation[8].

6.2 Software Development

The primary task will be development of the inverse kinematics for the nine degree of freedom testbed. These strategies will be used later to control the actual manipulator which will be used in the waste tanks. At this time, a library of routines which will implement the redundancy resolution is being built. In addition to a short program included in IGRIP's shared library to display the manipulator motion, various subroutines are being written to calculate forward and inverse kinematics for the SPAR and Schilling individually as well as the entire testbed as a whole. Programs written at this point are under development, and a listing of each is included in appendix D.

6.2.1 Software Description

The routines were written in C programming language. Note that a complete description is given at the beginning of each routine in appendix D. The following gives a summarized description of each routine.

(testbedCartPath):

This function calculates the necessary joint values to move the end effector of the PNL testbed manipulator along a Cartesian path between its current and a goal position and orientation.

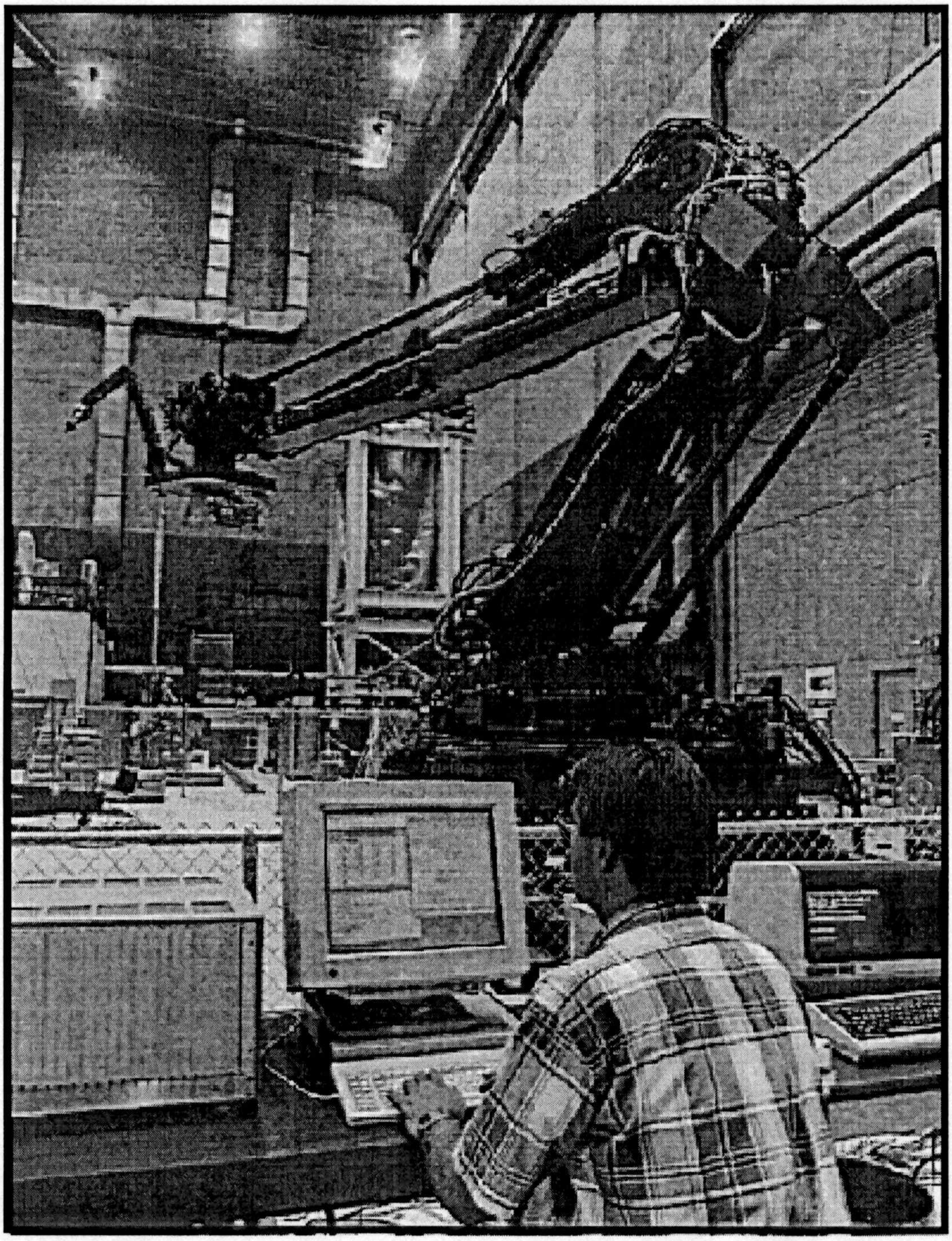


Fig. 6.1. Spar RMS 2500 manipulator

(pInvStep):

This function calculates the necessary joint value step to move the end effector of the PNL testbed manipulator one Cartesian step using the pseudo inverse method of redundancy resolution.

(testbedForKin):

This subroutine calculates the forward kinematics for the PNL testbed arm, consisting of a SPAR 2500 and Schilling Titan II manipulators.

(calcTestbedJacob):

This subroutine calculates the Jacobian for the PNL SPAR/Schillingtestbed. It takes as its inputs the nine current joint values of the entire testbed manipulator (testbedJntVals) and creates the 6x9 Jacobian matrix expressed in base coordinates (testbedJacobian).

(sparKinematics):

This subroutine calculates the kinematics, including the homogeneous transform and the Jacobian for the SPAR 2500 in the PNL testbed. It takes as its inputs the current joint values of the SPAR manipulator and creates the homogeneous transform matrix representing the position and orientation of the tool plate expressed in base coordinates. It also creates the 6x6 Jacobian relating the end effector velocity in tool plate coordinates to the joint velocities.

(schilKinematics):

This subroutine calculates the forward kinematics for the Schilling Titan II in the PNL testbed. It takes as its inputs the current joint values of the Schilling manipulator.

It creates the homogeneous transform matrix representing the position and orientation of the end effector expressed in tool plate coordinates. It also creates the 6x6 Jacobian relating the end effector velocity in tool plate coordinates to the joint velocities.

(solveLinSys):

This function solves the linear system Ax=b of (SIZE) equations. The algorithm used is LU decomposition with back substitution[12].

(invMat):

This function calculates the inverse of a (SIZE)x(SIZE)matrix. It takes as its input the matrix to be inverted and creates the inverse of this matrix. If the matrix is singular and thus has no solution, the function will return the error code SINGULARITY. Otherwise, a zero is returned. The inversion routine used is a simplified version of the Gauss-Jordan elimination algorithm[12].

6.3.2 Spar Gross Positioning Manipulator Kinematics

The transformation and Jacobian matrices of the Spar manipulator were calculated to be input in the corresponding routine (sparKinematics). The frame assignment and the DH table is shown in appendix C.

6.3.2 Schilling Dexterous Manipulator Kinematics

The transformation and Jacobian matrices of the Schilling manipulator were calculated to be input in the corresponding routine (schilKinematics). The transformation and Jacobian were interred in (schilKinematics) but never been tested. The frame assignment and the DH table is shown in appendix C.

6.3.3 Combining Serial Jacobian Technique

Calculation of the inverse kinematics for each part (SPAR and Schilling manipulators) separately as well as the entire manipulator may be required by various schemes. In order to reduce the calculation requirements, a formula was developed which would allow the construction of the entire manipulator Jacobian by using the Jacobians of each part. The following mathematical approach is used to combine the matrices to obtain the Jacobian for the entire manipulator.

Figure 6.2 shows the testbed manipulator along with the coordinate frames which will be used in the following mathematical development.

First the overall Jacobian matrix form will be the following. The joint velocity subscript G (for *G*ross positioning manipulator) will be used to represent the SPAR, and D (for *Dexterous* manipulator) to represent the Schilling.

$$\underset{6\mathrm{X}1}{\begin{bmatrix} {}^{O}\dot{r}_{EO} \\ {}^{O}\omega_{EO} \end{bmatrix}} = \underset{6\mathrm{X}9}{J} \underset{9\mathrm{X}1}{\begin{bmatrix} \dot{q}_G \\ \dot{q}_D \end{bmatrix}} \tag{6.1}$$

It will be necessary to express the Jacobian of the manipulator in terms of the Jacobians of both the SPAR and Schilling.

The Jacobian of the SPAR manipulator (the matrix that relates the tool plate velocity to the joint velocities relative the base expressed in the base frame) is calculated separately. Only the 3 currently operational joints are considered in the calculation. The leading superscript on the Jacobians express the extent of the manipulator which the

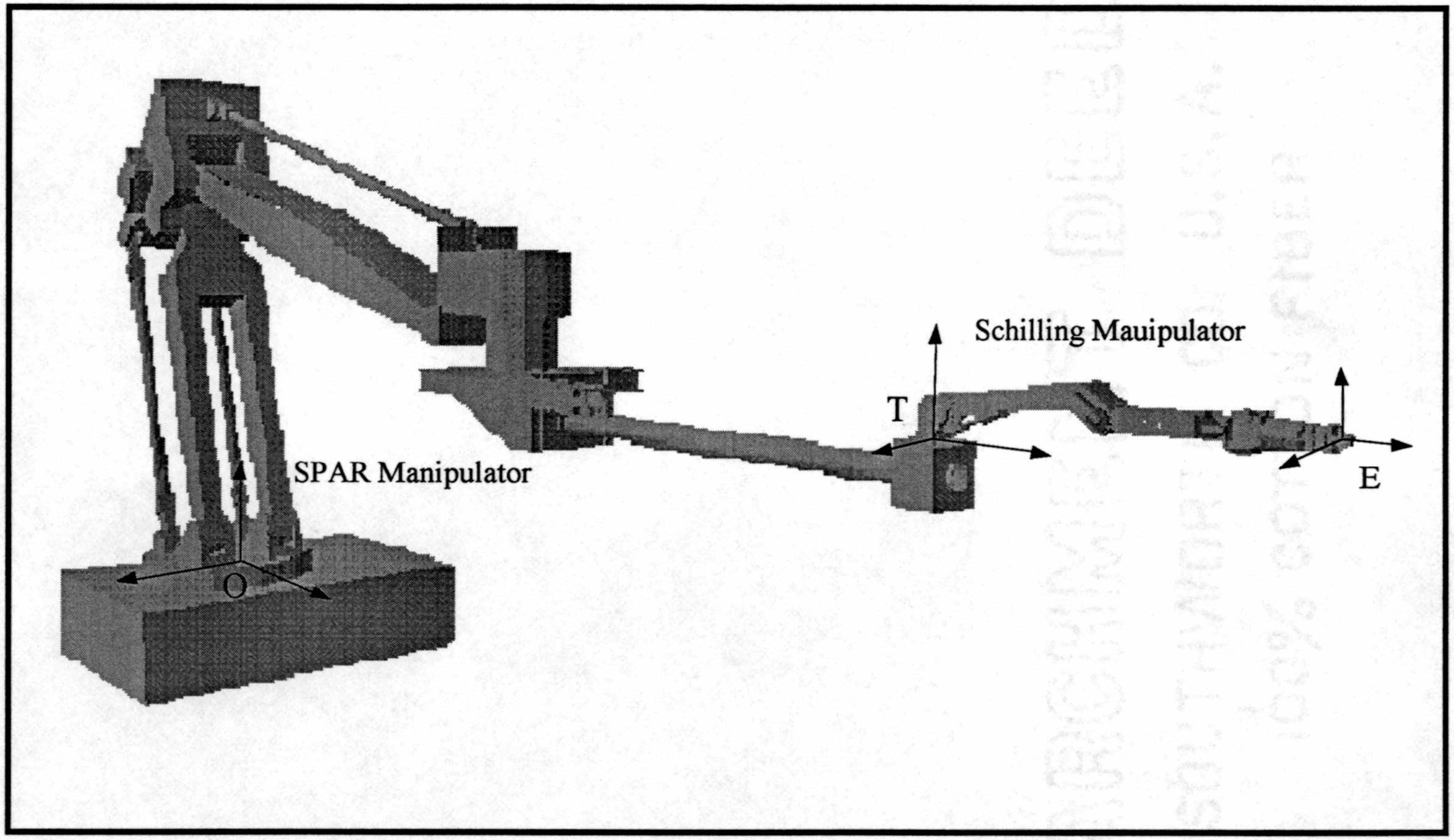


Fig. 6.2. Testbed manipulator

Jacobian describes, i.e., an OT indicates the Jacobian is for the section of the manipulator between the base (O) and tool plate (T) coordinate frames.

$$ {}^{O}\dot{P}_{G} = {}^{OT}J\,\dot{q}_{G} \tag{6.2}$$

The Jacobian equation is then partitioned into linear and angular velocity parts. The subscripts P and r represent position and rotation parts respectively

$$\begin{array}{ccc} 6X1 & = & 6X3 \quad 3X1 \end{array}$$

$$\begin{bmatrix} {}^{O}\dot{r}_{TO} \\ {}^{O}w_{TO} \end{bmatrix} = \begin{bmatrix} {}^{OT}J_{P} \\ {}^{OT}J_{r} \end{bmatrix} \begin{bmatrix} \dot{q}_{G} \end{bmatrix} \tag{6.3}$$

Thus, the linear and angular velocity equations are as follows.

$$ {}^{O}\dot{r}_{TO} = {}^{OT}J_{P}\,\dot{q}_{G} \tag{6.4}$$

$$ {}^{O}\omega_{TO} = {}^{OT}J_{r}\,\dot{q}_{G} \tag{6.5}$$

As above, the Schilling Jacobian equation is partitioned and written as separate linear and angular velocity expressions. The Schilling manipulator is a Titan II arm with six rotational joints.

$$ {}^{O}\dot{P}_{E} = {}^{TE}J\,\dot{q}_{D} \tag{6.6} $$

$$ \begin{matrix} 6X1 & = & 6X3 & 3X1 \end{matrix} $$

$$ \begin{bmatrix} {}^{O}\dot{r}_{ET} \\ {}^{O}w_{ET} \end{bmatrix} = \begin{bmatrix} {}^{TE}J_{P} \\ {}^{TE}J_{r} \end{bmatrix} \begin{bmatrix} \dot{q}_{D} \end{bmatrix} \tag{6.7} $$

$$ {}^{T}\dot{r}_{ET} = {}^{TE}J_{P}\,\dot{q}_{D} \tag{6.8} $$

$$ {}^{T}\omega_{ET} = {}^{TE}J_{r}\,\dot{q}_{D} \tag{6.9} $$

The dynamic equation for the linear velocity of the end effector relative to the base expressed in the base coordinate system is as follows:

$$ {}^{O}\dot{r}_{EO} = {}^{O}\dot{r}_{TO} + {}^{O}R_{T}\,{}^{T}\dot{r}_{ET} + {}^{O}\omega_{TO} \times {}^{O}R_{T}\,{}^{T}r_{ET} \tag{6.10} $$

Substituting the above Jacobian expressions for the SPAR and Schilling into the dynamic equation, the following expression is produced.

$$ {}^{O}\dot{r}_{EO} = {}^{OT}J_{P}\,\dot{q}_{G} + {}^{O}R_{T}\,{}^{TE}J_{P}\,\dot{q}_{D} + {}^{OT}J_{r}\,\dot{q}_{G} \times {}^{O}R_{T}\,{}^{T}r_{ET} \tag{6.11} $$

As above, the dynamic equation for the angular velocity is shown, and the appropriate Jacobian expressions are substituted.

$$^{O}\omega_{EO} = {}^{O}\omega_{TO} + {}^{O}R_{T}\,{}^{T}\omega_{ET} \tag{6.12}$$

$$^{O}\omega_{EO} = {}^{OT}J_{r}\,\dot{q}_{G} + {}^{O}R_{T}\,{}^{OT}J_{r}\,\dot{q}_{D} \tag{6.13}$$

By combining equations (6.11) and (6.13), an expression in the form of equation (6.1) may be written. The following equation shows the final matrix expression for the entire manipulator Jacobian written as a function of the individual SPAR and Schilling Jacobians.

$$\begin{bmatrix} {}^{O}\dot{r}_{EO} \\ {}^{O}\omega_{EO} \end{bmatrix} = \begin{bmatrix} {}^{OT}J_{P} - \left[{}^{O}R_{T}\,{}^{T}r_{ET}\times\right]{}^{OT}J_{r} & {}^{O}R_{T}\,{}^{TE}J_{P} \\ {}^{OT}J_{r} & {}^{O}R_{T}\,{}^{OT}J_{r} \end{bmatrix} \begin{bmatrix} \dot{q}_{G} \\ \dot{q}_{D} \end{bmatrix} \tag{6.14}$$

6.4 Planned Control Strategies

A method which will be used to determine motion strategies will be called tandem control. In these methods, the GPM and DM will be controlled with separate algorithms. Although several methods will be developed, the two which are currently planned are termed as toggled control and the workspace update method. The plan for the control system concept is shown in figure 6.3.

6.4.1 Toggled Control

In toggled control, the GPM and DM will both be maneuvered with a single joystick or Spaceball. A simple method of switching back and forth between the two will be implemented. Since neither of the two parts of the total manipulator will be redundant, i.e., have more than six joints, no specialized redundancy resolution will be needed for either one. In general, the GPM would be used to roughly position the end effector to the work area, while the DM would be used to do finer tasks.

6.4.2 Workspace Update Method

The workspace update method would control the GPM automatically, leaving the DM to be explicitly controlled by the operator. Under this method, the two manipulators would be moving simultaneously. Two types of this method will be developed, the discrete and continuous workspace position updates. In the continuous type, the GPM would be moving continuously but slowly along a set path. During this time, the operator would be able to maneuver the DM, most likely in the GPM tool plate coordinates. In the discrete update type, the GPM would only move intermittently. The operator would work inside a predefined volume, roughly the same as the workspace of the DM. When it moves out of or too close to the boundary of this workspace, the GPM would move to reposition the DM's workspace so that the area of interest is easily reachable. As this would only happen intermittently, an operator would not need to work in a moving coordinate system or constantly be required to update the base position of the DM.

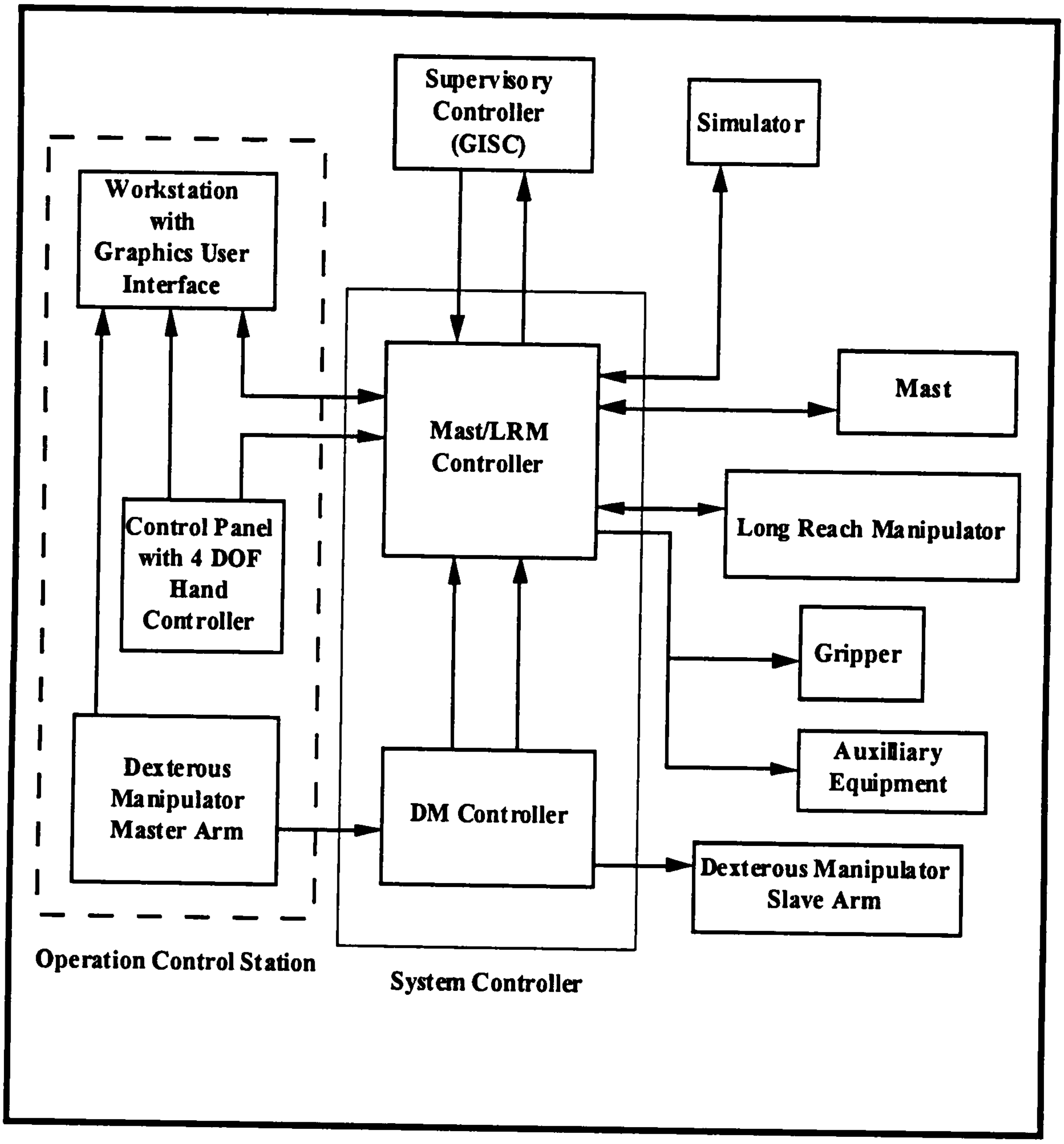


Fig. 6.3. Control system concept

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

Analysis of the proposed Schilling design for LRM revealed limitations in its deployment ability and workspace coverage. These limitations suggested the simple modification of replacing the third pitch with a yaw joint. The complete analyses of both the Schilling and the modified design was done using IGRIP simulation software. Conclusions resulting from the IGRIP simulations are discussed in the following paragraphs.

According to the project specifications, retraction of the arm after deployment is not allowed because of possible contamination of the environment. If retraction were allowed after initial waste removal and deployment, a much simpler manipulator with fewer links would be capable of satisfying all other requirements.

On the other hand, the modified design does not require retraction or the use of the mast prismatic joint throughout the mining process to completely cover the entire tank volume after deployment. In addition, the modified design has the capability of avoiding vertical and horizontal obstacles as well as maintaining the LRM in elbow up configuration to avoid contamination of the arm. An additional change suggested for the modified arm which would allow access to the center waste is the reduction of the upper arm #2 length from 9.5 feet to 7 feet.

Late in the process of analysis, additional design requirements on the orientation capabilities of the end effector were introduced. The abilities of the Schilling proposed design and the modified design to orient the end effector throughout the tank volume were evaluated using graphical methods. It was found that none of the designs have exceptional orientation ability, but if retraction is allowed, the elbow design has the largest orientable workspace. If, however, retraction is not allowed, the modified design is the only one of the two which can complete the cleanup of the tank. If certain orientations are required only in certain parts of the tank, the second UT modified design is the most appropriate.

Various strategies were proposed which would facilitate control of the long reach manipulator. For example toggled control would allow both the GPM and DM to be maneuvered with a single joystick or Spaceball; a simple method of alternating between the two would be implemented. Since neither of the two parts of the total manipulator are redundant, i.e., have more than six joints, no specialized redundancy resolution would be needed for either one. Another suggested control strategy is the workspace update method in which the GPM is controlled automatically, leaving the DM to be explicitly controlled by the operator.

7.2 Recommendations

The fact that the requirements were not clearly prioritized created the greatest difficulty in designing the long reach manipulator. For example, deployment without digging and the disallowed retraction led to the modified design. At a later time this design was disqualified because of the low priority given to the former requirements and the revised importance of the orientation requirements. For any future analysis, the priority on requirements must be well defined.

It is still necessary for the software used for simulation and real time control at PNL to be integrated with the software written at UT. Simulations showing different mining strategies should be developed.

It is recommended that all control strategies mentioned in chapter 6 be further developed. Conceptually these strategies still require many details to be filled in. For example it has not yet been determined under what circumstances the LRM should be repositioned in the workspace update method. Once these strategies are better defined, an implementation and subsequent analysis may be pursued.

References

References

[1] "Key Design Requirements for Long Reach Manipulators", by Kwon D. S., March-Leuba S., Babcock S. M. and Hamel W. R., September 1993, ORNL/TM-12251.

[2] "Implications of Access Hole Size On Tank Waste Retrieval System Design And Cost", by Babcock S. M., Kwon D. S., Burks B. L., Stoughton R. S. and Evans M. S., May 1994, ORNL/TM-12751.

[3] "Status of Long-Reach Manipulator Control Development at the Oak Ridge National Laboratory", July 20-21, 1993.

[4] "Long-Reach Manipulation for Waste Storage Tank Remediation", Jansen J. F., March-Leuba S., Kwon D. S., Babcock S. M., Burks B. L., Kress R. L. and Hamel W. R., FY 1991 Report, ORNL/TM-11999.

[5] "Parametric Design Studies of Long Reach Manipulators", by Kwon D. S., March-Leuba S., Babcock S. M., Burks B. L., and Hamel W. R., Fifth Topical Meeting on Robotics and Remote Handling, April 1993.

[6] "Technical Specifications of Tank Waste Retrieval Test Manipulator System", by Pacific Northwest Laboratory, June 1994.

[7] "Tank Waste Retrieval Test Manipulator System", by Schilling, October 1994.

[8] "TITAN II Telerobotic Manipulator System", by Schilling.

[9] IGRIP model for TWR system developed at PNL.

[10] Zghal, H., Dubey, R., and Euler, J., "Efficient Gradient Projection Optimization for Manipulators with Multiple Degrees of Redundancy," Proc. of the IEEE International Conference on Robotics and Automation, May 13-18, Cincinnati, Ohio, Vol. 2, pp. 1006-1011, 1990.

[11] Yoshikawa, T., Foundation of Robotics, Analysis and Control, The MIT Press, 1990.

[12] "Numerical Recipes" by Press, Flannery, Teukolsky,and Vettering.

[13] W. J .Book, O. Maizza-Neto, and D. E. Whitney, "Feedback Control of Two Beam, Two Joint System with Distributed Flexibility", Trans. ASME: J. Dyn. Syst., Meas. Control, pp. 442-431 (December 1975)

[14] P. Wenger and P. C hedmail, "On the Connectivity of Manipulator Free Workspace", J. Robotic Syst., 8 (6), (March/April 1988)

[15] Bard Paden and Shankar Satstry, "Optimal Kinematic Design of 6R Manipulators", Intl. J. Robotics, 8, (6), 761-99 (1990)

Appendices

Appendix A

IGRIP Simulation

Figure A.1 shows the names of the parts of the LRM Schilling design model residing in the parts directory: euler/alhalabi/lrm/newmodel/IGRIP.351A/PARTS/lrm2.

When degrees of freedom of the Schilling arm were set, the calculation of the actuators movement was necessary. The formulas for each degree of freedom reside in the shoulder DOF and can be changed in case of any change in link lengths. Figure A.2 gives an illustration of the dimensions used to construct these formulas shown after the figure.

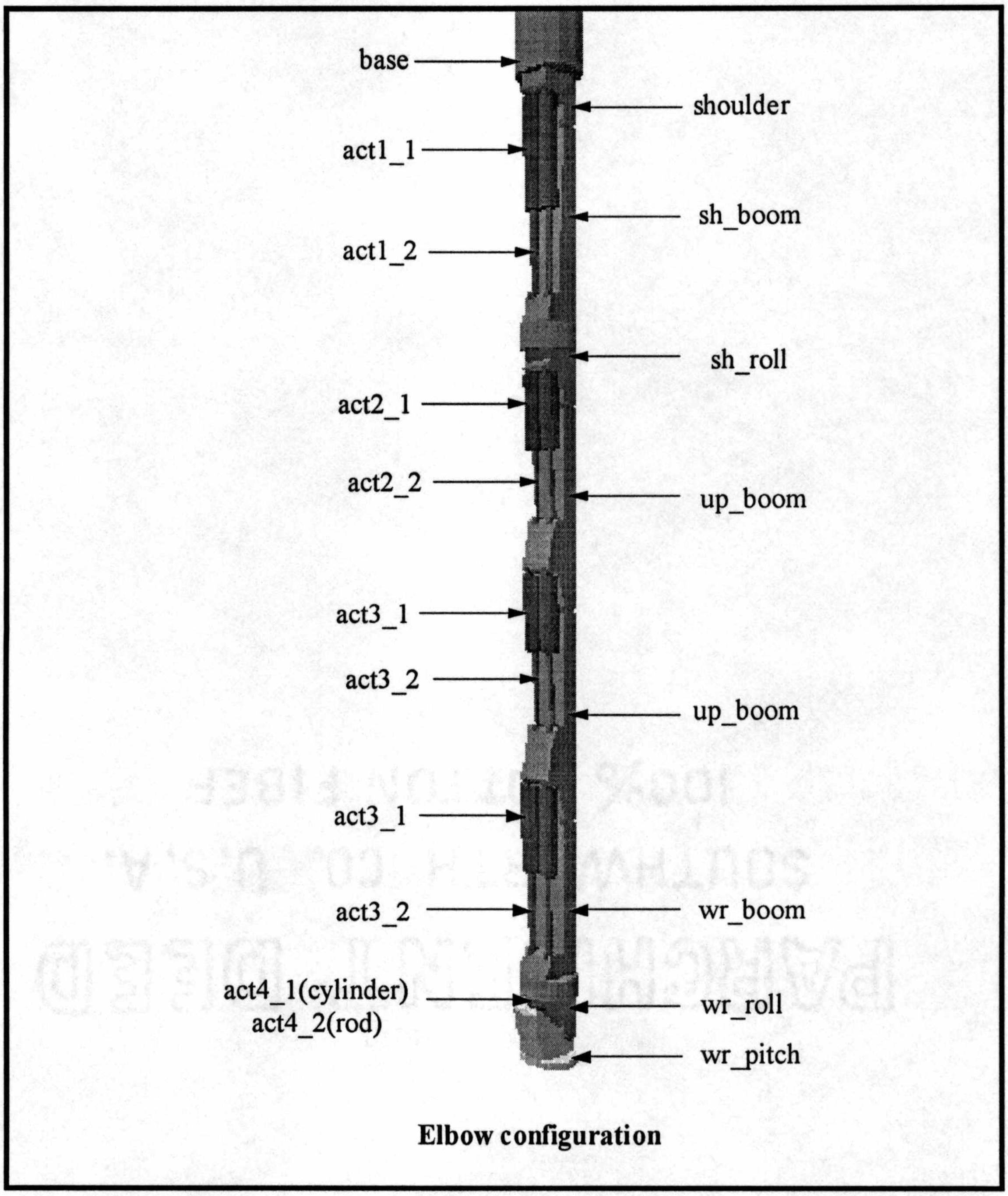


Fig. A.1 Part names of the Schilling design in IGRIP

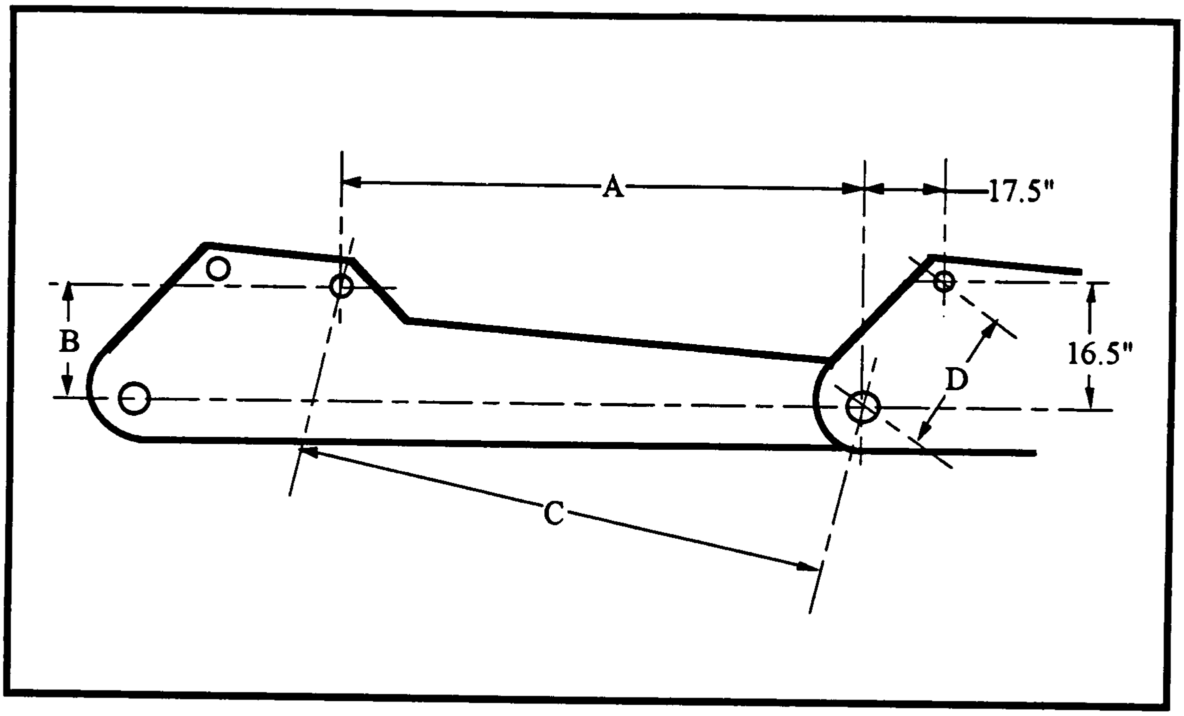


Fig. A.2. Typical link dimensions used in IGRIP simulation

$$t_i = \mathrm{atanr}(17.5/16.5) - \mathrm{atanr}(B/A) + \pi/2 - \mathrm{dof}(i) \qquad (A.1)$$

$$p_i = \mathrm{sasa}(C, t_i, D) \qquad (A.2)$$

$$a_i = \mathrm{atanr}(17.5/16.5) + p_i \qquad (A.3)$$

$$b_i = \pi - p_i - t_i - \mathrm{atanr}(B/A) \qquad (A.4)$$

If any link length is changed, it is important to make the corresponding changes in these formulas.

Other elements of the simulation reside in the same directory. For example the tank is in the parts directory and is named tank. The waste is divided into rings and each ring is a part named in the form of matrix components (w1_1, w1_2, ..., w1_13, w2_1, ..., w10_13).

Simulation done on IGRIP was done by written programs in gsl language. These programs reside in the programs sub directory in the same directory of parts. The programs written for both the Schilling arm simulation and the modified arm simulation are as follows:

Program for Deployment and Cleaning the Tank in SCARA configuration

```
/*******************************************************************************
*                                                                              *
*       description:    Schilling arm simulation in SCARA configuration        *
*       programmer:     Amin AL-Halabi                                         *
*       date:           11/14/94                                               *
*                                                                              *
*                                                                              *
*******************************************************************************/


PROGRAM oldentry
BEGIN MAIN

 $STEPSIZE = 0.1
 $MOTYPE = JOINT
 $SPEED_MODE = PERCENT
 $SPEED = 1
 UNITS = ENGLISH


 CLI("FULL SCREEN")
 DELAY 2000
 CLI("SELECT 'gpm'")
 MOVE JOINT 6 TO -90 IMMEDIATE

OPEN WINDOW 'SDI/Spar DESIGN' @ -0.8, -.4: 1 AS 1
write @1, ('INDEX THE FIRST ROLL JOINT BY 90 deg', cr)
write @1, ('THEN DEPLOY THE ARM INTO THE TANK', cr)

DELAY 500

 MOVE JOINT 8 TO 380 NOSIMUL
 CLI("DELETE PART 'W:w1-13'")
write @1, (CLS)
write @1, ('INITIAL WASTE MATERIAL REMOVAL AT THE', cr)
write @1, ('TOP TO ENABLE DEPLOYMENT', cr)

 MOVE JOINT 8 TO 375 SIMUL
 MOVE JOINT 4 TO 10 NOSIMUL
```

```
CLI("DELETE PART 'W:w1-12'")

MOVE JOINT 8 TO 360 SIMUL
MOVE JOINT 4 TO 30 NOSIMUL
CLI("DELETE PART 'W:w1-11'")

MOVE JOINT 8 TO 330 SIMUL
MOVE JOINT 4 TO 50  NOSIMUL
CLI("DELETE PART 'W:w1-10'")

MOVE JOINT 8 TO 270 SIMUL
MOVE JOINT 4 TO 85  NOSIMUL

MOVE JOINT 4 TO 70 SIMUL
MOVE JOINT 3 TO 10 NOSIMUL
CLI("DELETE PART 'W:w1-9'")

MOVE JOINT 4 TO 50 SIMUL
MOVE JOINT 3 TO 25 NOSIMUL
CLI("DELETE PART 'W:w1-8'")

MOVE JOINT 4 TO 40 SIMUL
MOVE JOINT 3 TO 45 SIMUL
MOVE JOINT 8 TO 230 NOSIMUL
CLI("DELETE PART 'W:w1-7'")

MOVE JOINT 4 TO 5 SIMUL
MOVE JOINT 3 TO 83 SIMUL
MOVE JOINT 8 TO 154 NOSIMUL

MOVE JOINT 2 TO 6 SIMUL
MOVE JOINT 3 TO 77 NOSIMUL
CLI("DELETE PART 'W:w1-6'")

MOVE JOINT 2 TO 20 SIMUL
MOVE JOINT 3 TO 67 SIMUL
MOVE JOINT 8 TO 130 NOSIMUL
CLI("DELETE PART 'W:w1-5'")
write @1, (CLS)
write @1, ('DEPLOYMENT POSSIBLE WITH PRESENT WASTE', cr)
MOVE JOINT 3 TO 75   SIMUL
MOVE JOINT 4 TO 20   NOSIMUL

MOVE JOINT 2 TO 80   SIMUL
MOVE JOINT 3 TO 10   SIMUL
MOVE JOINT 8 TO 42   NOSIMUL

MOVE JOINT 2 TO 85   SIMUL
MOVE JOINT 8 TO 22   NOSIMUL

MOVE JOINT 1 TO 25   SIMUL
MOVE JOINT 4 TO 15   SIMUL
MOVE JOINT 8 TO 14   NOSIMUL

MOVE JOINT 1 TO 40   SIMUL
MOVE JOINT 4 TO -4   NOSIMUL
```

```
  MOVE JOINT 1 TO 55   SIMUL
  MOVE JOINT 8 TO -18  NOSIMUL

  MOVE JOINT 1 TO 70   SIMUL
  MOVE JOINT 8 TO -55  NOSIMUL
write @1, (CLS)
write @1, ('ARM COMPLETELY DEPLOYED', cr)
DELAY 1000

  MOVE JOINT 1 TO 85   SIMUL
  MOVE JOINT 4 TO 36   SIMUL
  MOVE JOINT 5 TO 90   SIMUL
  MOVE JOINT 8 TO -91   NOSIMUL
write @1, (CLS)
write @1, ('CONTINUE CLEANING UP WASTE IN TOP LAYER', cr)

  MOVE JOINT 1 TO 95   SIMUL
  MOVE JOINT 2 TO 90   SIMUL
  MOVE JOINT 8 TO -111   NOSIMUL
  CLI("DELETE PART 'W:w1-4'")

  MOVE JOINT 2 TO 75   SIMUL
  MOVE JOINT 8 TO -123   NOSIMUL
  CLI("DELETE PART 'W:w1-3'")

  MOVE JOINT 1 TO 90   SIMUL
  MOVE JOINT 2 TO 60   NOSIMUL
  CLI("DELETE PART 'W:w1-2'")

  MOVE JOINT 2 TO 60   SIMUL
  MOVE JOINT 4 TO 11   SIMUL
  MOVE JOINT 5 TO 40   NOSIMUL
  CLI("DELETE PART 'W:w1-1'")

write @1, (CLS)
write @1, ('CLEAN UP WASTE IN SECOND LAYER', cr)
  MOVE JOINT 5 TO 85    SIMUL
  MOVE JOINT 8 TO -152  NOSIMUL
  CLI("DELETE PART 'W:w2-1'")

  MOVE JOINT 4 TO 30  SIMUL
  MOVE JOINT 5 TO 90  NOSIMUL
  CLI("DELETE PART 'W:w2-2'")

  MOVE JOINT 4 TO 50  NOSIMUL
  CLI("DELETE PART 'W:w2-3'")

  MOVE JOINT 4 TO 70  NOSIMUL
  CLI("DELETE PART 'W:w2-4'")

  MOVE JOINT 4 TO 85  NOSIMUL
  CLI("DELETE PART 'W:w2-5'")

  MOVE JOINT 4 TO 100  NOSIMUL
  CLI("DELETE PART 'W:w2-6'")

  MOVE JOINT 3 TO 25  NOSIMUL
```

```
CLI("DELETE PART 'W:w2-7'")

MOVE JOINT 3 TO 40  NOSIMUL
CLI("DELETE PART 'W:w2-8'")

MOVE JOINT 3 TO 50  NOSIMUL
CLI("DELETE PART 'W:w2-9'")

MOVE JOINT 3 TO 65  NOSIMUL
CLI("DELETE PART 'W:w2-10'")

MOVE JOINT 2 TO 80  NOSIMUL
CLI("DELETE PART 'W:w2-11'")

MOVE JOINT 2 TO 93  SIMUL
MOVE JOINT 4 TO 95  NOSIMUL
CLI("DELETE PART 'W:w2-12'")
write @1, (CLS)
write @1, ('CANNOT CLEAN CENTER WASTE', cr)
write @1, ('BECAUSE THE ARM WILL HIT ITSELF', cr)

-- Third layer ---------

MOVE JOINT 1 TO 80  SIMUL
MOVE JOINT 2 TO 68  SIMUL
MOVE JOINT 3 TO 0   SIMUL
MOVE JOINT 4 TO 5   SIMUL
MOVE JOINT 8 TO -143.2  NOSIMUL
CLI("DELETE PART 'W:w3-1'")

MOVE JOINT 1 TO 76  SIMUL
MOVE JOINT 4 TO 30  NOSIMUL
CLI("DELETE PART 'W:w3-2'")

MOVE JOINT 1 TO 46  SIMUL
MOVE JOINT 2 TO 58  SIMUL
MOVE JOINT 3 TO 15  SIMUL
MOVE JOINT 4 TO 20  SIMUL
MOVE JOINT 8 TO -10.2  NOSIMUL
CLI("DELETE PART 'W:w3-3'")

write @1, (CLS)
write @1, ('FOR CONTINUATION OF WASTE REMOVAL', cr)
write @1, ('THE ARM WILL HAVE TO BE RETRACTED', cr)
write @1, ('BACK OUT OF THE TANK', cr)

DELAY 50000

--------------------------------
-------- END MAIN --------
END oldentry
```

Program for Deployment and Cleaning the Tank Using the Modified Design

```
/*******************************************************************************
*                                                                              *
*        description:     modified arm simulation                              *
*        programmer:      Amin AL-Halabi                                       *
*        date:            11/16/94                                             *
*                                                                              *
*                                                                              *
*******************************************************************************/

PROGRAM newentry
BEGIN MAIN
-------------------
  $STEPSIZE = 0.1
  $MOTYPE = JOINT
  $SPEED_MODE = PERCENT
  $SPEED = 0.2
  UNITS = ENGLISH



  CLI("FULL SCREEN")
  DELAY 2000
  CLI("SELECT 'gpm'")
  MOVE JOINT 6 TO 90 IMMEDIATE
OPEN WINDOW 'Modified Design' @ -0.8, -.4: 1 AS 1

write @1, ('INDEX THE FIRST ROLL JOINT TO 90 deg', cr)
write @1, ('THEN DEPLOY THE ARM INTO THE TANK', cr)

DELAY 500

  MOVE JOINT 8 TO 380 NOSIMUL
  CLI("DELETE PART 'W:w1-13'")
write @1, (CLS)
write @1, ('INITIAL WASTE MATERIAL REMOVAL TO ENABLE DEPLOYMENT', cr)

  MOVE JOINT 8 TO 375 SIMUL
  MOVE JOINT 4 TO 10 NOSIMUL
  CLI("DELETE PART 'W:w1-12'")

  MOVE JOINT 8 TO 360 SIMUL
  MOVE JOINT 4 TO 30 NOSIMUL
  CLI("DELETE PART 'W:w1-11'")

  MOVE JOINT 8 TO 330 SIMUL
  MOVE JOINT 4 TO 55  NOSIMUL
  CLI("DELETE PART 'W:w1-10'")

  MOVE JOINT 8 TO 270 SIMUL
  MOVE JOINT 4 TO 100  NOSIMUL

  MOVE JOINT 3 TO 25 SIMUL
  MOVE JOINT 4 TO 75  NOSIMUL
  CLI("DELETE PART 'W:w1-9'")
write @1, (CLS)
write @1, ('DEPLOYMENT POSSIBLE WITH PRESENT WASTE', cr)
```

```
MOVE JOINT 3 TO 60 NOSIMUL

MOVE JOINT 3 TO 100 SIMUL
MOVE JOINT 8 TO 160  NOSIMUL


MOVE JOINT 2 TO 50 SIMUL
MOVE JOINT 3 TO 75 SIMUL
MOVE JOINT 4 TO 35 NOSIMUL

MOVE JOINT 2 TO 85 SIMUL
MOVE JOINT 4 TO 5 SIMUL
MOVE JOINT 8 TO 75 NOSIMUL

MOVE JOINT 2 TO 104 SIMUL
MOVE JOINT 3 TO 55 SIMUL
MOVE JOINT 8 TO 15 NOSIMUL

MOVE JOINT 3 TO 35 SIMUL
MOVE JOINT 4 TO -4 NOSIMUL

MOVE JOINT 1 TO 15 SIMUL
MOVE JOINT 3 TO 15 SIMUL
MOVE JOINT 8 TO 10 NOSIMUL

MOVE JOINT 1 TO 35 SIMUL
MOVE JOINT 8 TO 0 NOSIMUL

MOVE JOINT 1 TO 65 SIMUL
MOVE JOINT 8 TO -15 NOSIMUL
write @1, (CLS)
write @1, ('ARM COMPLETELY DEPLOYED', cr)
write @1, ('MOVE THE ELBOW AWAY FROM THE WASTE', cr)

MOVE JOINT 1 TO 85 SIMUL
MOVE JOINT 3 TO 20 SIMUL
MOVE JOINT 5 TO 45 SIMUL
MOVE JOINT 4 TO 96 NOSIMUL

write @1, (CLS)
write @1, ('CONTINUE CLEANING UP WASTE IN TOP LAYER', cr)
MOVE JOINT 3 TO 50 SIMUL
MOVE JOINT 5 TO 75 SIMUL
MOVE JOINT 4 TO 70 NOSIMUL
CLI("DELETE PART 'W:w1-8'")

MOVE JOINT 3 TO 40 SIMUL
MOVE JOINT 4 TO 60 NOSIMUL
CLI("DELETE PART 'W:w1-7'")

MOVE JOINT 2 TO 90 NOSIMUL
CLI("DELETE PART 'W:w1-6'")

MOVE JOINT 2 TO 80 SIMUL
MOVE JOINT 3 TO 35 NOSIMUL
CLI("DELETE PART 'W:w1-5'")
```

```
 MOVE JOINT 2 TO 68 NOSIMUL
 CLI("DELETE PART 'W:w1-4'")

 MOVE JOINT 2 TO 65 SIMUL
 MOVE JOINT 3 TO 30 SIMUL
 MOVE JOINT 4 TO 45 NOSIMUL
 CLI("DELETE PART 'W:w1-3'")

 MOVE JOINT 1 TO 81 SIMUL
 MOVE JOINT 2 TO 53 SIMUL
 MOVE JOINT 3 TO 22 NOSIMUL
 CLI("DELETE PART 'W:w1-2'")

 MOVE JOINT 4 TO 48 SIMUL
 MOVE JOINT 2 TO 32 SIMUL
 MOVE JOINT 3 TO 17 NOSIMUL
 CLI("DELETE PART 'W:w1-1'")

write @1, (CLS)
write @1, ('CLEANING THE WASTE IN THE SECOND LAYER', cr)

 MOVE JOINT 3 TO 24 SIMUL
 MOVE JOINT 4 TO 31 SIMUL
 MOVE JOINT 5 TO 20 NOSIMUL
 CLI("DELETE PART 'W:w2-1'")

 MOVE JOINT 2 TO 49 SIMUL
 MOVE JOINT 3 TO 30 NOSIMUL
 CLI("DELETE PART 'W:w2-2'")

 MOVE JOINT 2 TO 65 SIMUL
 MOVE JOINT 3 TO 33 NOSIMUL
 CLI("DELETE PART 'W:w2-3'")

 MOVE JOINT 2 TO 73 SIMUL
 MOVE JOINT 4 TO 34 SIMUL
 MOVE JOINT 5 TO 40 SIMUL
 MOVE JOINT 3 TO 37 NOSIMUL
 CLI("DELETE PART 'W:w2-4'")

 MOVE JOINT 3 TO 45 SIMUL
 MOVE JOINT 4 TO 51 NOSIMUL
 CLI("DELETE PART 'W:w2-5'")

 MOVE JOINT 3 TO 56 SIMUL
 MOVE JOINT 4 TO 65 NOSIMUL
 CLI("DELETE PART 'W:w2-6'")

 MOVE JOINT 2 TO 88 SIMUL
 MOVE JOINT 4 TO 61 NOSIMUL
 CLI("DELETE PART 'W:w2-7'")

 MOVE JOINT 2 TO 104 SIMUL
 MOVE JOINT 3 TO 58 SIMUL
 MOVE JOINT 4 TO 53 NOSIMUL
 CLI("DELETE PART 'W:w2-8'")
```

```
 MOVE JOINT 5 TO 70 SIMUL
 MOVE JOINT 3 TO 70 SIMUL
 MOVE JOINT 4 TO 65 NOSIMUL
 CLI("DELETE PART 'W:w2-9'")

 MOVE JOINT 3 TO 82 SIMUL
 MOVE JOINT 4 TO 69 NOSIMUL
 CLI("DELETE PART 'W:w2-10'")

 MOVE JOINT 3 TO 95 NOSIMUL
 CLI("DELETE PART 'W:w2-11'")

 MOVE JOINT 3 TO 104 SIMUL
 MOVE JOINT 4 TO 67 NOSIMUL
 CLI("DELETE PART 'W:w2-12'")
write @1, (CLS)
write @1, ('CANNOT REMOVE THE WASTE AT THE CENTER', cr)
write @1, ('CLEANING THE WASTE IN THE THIRD LAYER', cr)

 MOVE JOINT 4 TO 47 NOSIMUL
 CLI("DELETE PART 'W:w3-12'")

 MOVE JOINT 3 TO 94 NOSIMUL
 CLI("DELETE PART 'W:w3-11'")

 MOVE JOINT 3 TO 84 NOSIMUL
 CLI("DELETE PART 'W:w3-10'")

 MOVE JOINT 3 TO 74 NOSIMUL
 CLI("DELETE PART 'W:w3-9'")

 MOVE JOINT 3 TO 64 SIMUL
 MOVE JOINT 4 TO 32 NOSIMUL
 CLI("DELETE PART 'W:w3-8'")

 MOVE JOINT 2 TO 99 SIMUL
 MOVE JOINT 3 TO 59 SIMUL
 MOVE JOINT 4 TO 27 NOSIMUL
 CLI("DELETE PART 'W:w3-7'")

 MOVE JOINT 2 TO 79 SIMUL
 MOVE JOINT 4 TO 37 NOSIMUL
 CLI("DELETE PART 'W:w3-6'")

 MOVE JOINT 2 TO 49 SIMUL
 MOVE JOINT 4 TO 47 NOSIMUL
 CLI("DELETE PART 'W:w3-5'")

 MOVE JOINT 2 TO 29 SIMUL
 MOVE JOINT 4 TO 52 NOSIMUL
 CLI("DELETE PART 'W:w3-4'")

 MOVE JOINT 1 TO 77 SIMUL
 MOVE JOINT 2 TO -1 SIMUL
 MOVE JOINT 3 TO 49 SIMUL
 MOVE JOINT 4 TO 71 NOSIMUL
```

```
  CLI("DELETE PART 'W:w3-3'")

  MOVE JOINT 3 TO 41 SIMUL
  MOVE JOINT 4 TO 59 NOSIMUL
  CLI("DELETE PART 'W:w3-2'")

  MOVE JOINT 3 TO 30 SIMUL
  MOVE JOINT 4 TO 45 NOSIMUL
  CLI("DELETE PART 'W:w3-1'")
write @1, (CLS)
write @1, ('CLEANING THE WASTE IN THE FOURTH LAYER', cr)

  MOVE JOINT 2 TO 2   SIMUL
  MOVE JOINT 3 TO 34  SIMUL
  MOVE JOINT 4 TO 8  NOSIMUL
  CLI("DELETE PART 'W:w4-1'")

  MOVE JOINT 3 TO 43  SIMUL
  MOVE JOINT 4 TO 42  NOSIMUL
  CLI("DELETE PART 'W:w4-2'")

  MOVE JOINT 3 TO 53  SIMUL
  MOVE JOINT 4 TO 52  NOSIMUL
  CLI("DELETE PART 'W:w4-3'")

  MOVE JOINT 3 TO 63  SIMUL
  MOVE JOINT 4 TO 57  NOSIMUL
  CLI("DELETE PART 'W:w4-4'")

  MOVE JOINT 3 TO 76  SIMUL
  MOVE JOINT 4 TO 62  NOSIMUL
  CLI("DELETE PART 'W:w4-5'")

  MOVE JOINT 2 TO 37  SIMUL
  MOVE JOINT 4 TO 47  NOSIMUL
  CLI("DELETE PART 'W:w4-6'")

  MOVE JOINT 3 TO 86  NOSIMUL
  CLI("DELETE PART 'W:w4-7'")

  MOVE JOINT 3 TO 95  SIMUL
  MOVE JOINT 4 TO 43  NOSIMUL
  CLI("DELETE PART 'W:w4-8'")

  MOVE JOINT 3 TO 104  SIMUL
  MOVE JOINT 4 TO 35  NOSIMUL
  CLI("DELETE PART 'W:w4-9'")

  MOVE JOINT 1 TO 70  SIMUL
  MOVE JOINT 2 TO 52  SIMUL
  MOVE JOINT 4 TO 40  NOSIMUL
  CLI("DELETE PART 'W:w4-10'")

  MOVE JOINT 2 TO 83  SIMUL
  MOVE JOINT 4 TO 28  NOSIMUL
  CLI("DELETE PART 'W:w4-11'")
```

```
 MOVE JOINT 1 TO 67  SIMUL
 MOVE JOINT 2 TO 104  SIMUL
 MOVE JOINT 5 TO 20  SIMUL
 MOVE JOINT 4 TO 22  NOSIMUL
 CLI("DELETE PART 'W:w4-12'")
write @1, (CLS)
write @1, ('CANNOT REMOVE THE WASTE AT THE CENTER', cr)
write @1, ('CLEANING THE WASTE IN THE FIFTH LAYER', cr)
 MOVE JOINT 1 TO 58  SIMUL
 MOVE JOINT 2 TO 50  SIMUL
 MOVE JOINT 3 TO 105 SIMUL
 MOVE JOINT 4 TO 0  NOSIMUL
 CLI("DELETE PART 'W:w5-11'")

 MOVE JOINT 1 TO 48  SIMUL
 MOVE JOINT 5 TO 90  SIMUL
 MOVE JOINT 4 TO 11  NOSIMUL
 CLI("DELETE PART 'W:w5-12'")

 MOVE JOINT 1 TO 67  SIMUL
 MOVE JOINT 3 TO 102  NOSIMUL
 CLI("DELETE PART 'W:w5-10'")

 MOVE JOINT 3 TO 92  SIMUL
 MOVE JOINT 4 TO 26  NOSIMUL
 CLI("DELETE PART 'W:w5-9'")

 MOVE JOINT 3 TO 82  SIMUL
 MOVE JOINT 4 TO 33  NOSIMUL
 CLI("DELETE PART 'W:w5-8'")

 MOVE JOINT 3 TO 73  SIMUL
 MOVE JOINT 4 TO 34  NOSIMUL
 CLI("DELETE PART 'W:w5-7'")

 MOVE JOINT 3 TO 63  SIMUL
 MOVE JOINT 4 TO 33  NOSIMUL
 CLI("DELETE PART 'W:w5-6'")

 MOVE JOINT 3 TO 53  SIMUL
 MOVE JOINT 4 TO 28  NOSIMUL
 CLI("DELETE PART 'W:w5-5'")

 MOVE JOINT 3 TO 46  SIMUL
 MOVE JOINT 4 TO 13  NOSIMUL
 CLI("DELETE PART 'W:w5-4'")

 MOVE JOINT 2 TO 47  SIMUL
 MOVE JOINT 3 TO 38  SIMUL
 MOVE JOINT 5 TO 50  SIMUL
 MOVE JOINT 4 TO 1  NOSIMUL
 CLI("DELETE PART 'W:w5-3'")

 MOVE JOINT 2 TO 28  SIMUL
 MOVE JOINT 3 TO 29 NOSIMUL
 CLI("DELETE PART 'W:w5-2'")
```

```
 MOVE JOINT 1 TO 64  SIMUL
 MOVE JOINT 2 TO 7  SIMUL
 MOVE JOINT 3 TO 16  SIMUL
 MOVE JOINT 4 TO 0  NOSIMUL
 CLI("DELETE PART 'W:w5-1'")

write @1, (CLS)
write @1, ('CLEANING THE WASTE IN THE SIXTH LAYER', cr)
write @1, ('CANNOT REACH THE OUTERMOST RING', cr)
write @1, ('IT CAN BE REACHED BY LOWERING THE COLUMN', cr)

 MOVE JOINT 3 TO 28  NOSIMUL
 CLI("DELETE PART 'W:w6-2'")

 MOVE JOINT 3 TO 36  SIMUL
 MOVE JOINT 4 TO 24  NOSIMUL
 CLI("DELETE PART 'W:w6-3'")

 MOVE JOINT 3 TO 45  SIMUL
 MOVE JOINT 4 TO 36  NOSIMUL
 CLI("DELETE PART 'W:w6-4'")

 MOVE JOINT 3 TO 55  SIMUL
 MOVE JOINT 4 TO 40  NOSIMUL
 CLI("DELETE PART 'W:w6-5'")

 MOVE JOINT 3 TO 65  NOSIMUL
 CLI("DELETE PART 'W:w6-6'")

 MOVE JOINT 3 TO 76  NOSIMUL
 CLI("DELETE PART 'W:w6-7'")

 MOVE JOINT 3 TO 85  SIMUL
 MOVE JOINT 4 TO 33  NOSIMUL
 CLI("DELETE PART 'W:w6-8'")

 MOVE JOINT 3 TO 92  SIMUL
 MOVE JOINT 4 TO 20  NOSIMUL
 CLI("DELETE PART 'W:w6-9'")

 MOVE JOINT 1 TO 61  SIMUL
 MOVE JOINT 3 TO 95  SIMUL
 MOVE JOINT 4 TO 6  NOSIMUL
 CLI("DELETE PART 'W:w6-10'")

 MOVE JOINT 1 TO 56  SIMUL
 MOVE JOINT 3 TO 98 NOSIMUL
 CLI("DELETE PART 'W:w6-11'")

 MOVE JOINT 1 TO 50  SIMUL
 MOVE JOINT 3 TO 100 NOSIMUL
 CLI("DELETE PART 'W:w6-12'")

write @1, (CLS)
write @1, ('NOW CAN CLEAN THE CENTRAL WASTE ', cr)
write @1, ('WASTE ON TOP COLLAPSES DOWN ', cr)
```

```
 MOVE JOINT 1 TO 45 NOSIMUL
 CLI("DELETE PART 'W:w2-13'")
 DELAY 500
 CLI("DELETE PART 'W:w3-13'")
 DELAY 500
 CLI("DELETE PART 'W:w4-13'")
 DELAY 500
 CLI("DELETE PART 'W:w5-13'")
 DELAY 500
 CLI("DELETE PART 'W:w6-13'")

write @1, (CLS)
write @1, ('CLEANING THE WASTE IN THE SEVENTH LAYER', cr)

 MOVE JOINT 1 TO 40  SIMUL
 MOVE JOINT 2 TO 2  SIMUL
 MOVE JOINT 3 TO 89 NOSIMUL
 CLI("DELETE PART 'W:w7-13'")

 MOVE JOINT 1 TO 45  NOSIMUL
 CLI("DELETE PART 'W:w7-12'")

 MOVE JOINT 3 TO 87  SIMUL
 MOVE JOINT 4 TO 26 NOSIMUL
 CLI("DELETE PART 'W:w7-11'")

 MOVE JOINT 3 TO 82  SIMUL
 MOVE JOINT 4 TO 41 NOSIMUL
 CLI("DELETE PART 'W:w7-10'")

 MOVE JOINT 2 TO 16  SIMUL
 MOVE JOINT 3 TO 73  SIMUL
 MOVE JOINT 4 TO 40 NOSIMUL
 CLI("DELETE PART 'W:w7-9'")

 MOVE JOINT 2 TO 30  SIMUL
 MOVE JOINT 3 TO 64  SIMUL
 MOVE JOINT 4 TO 34 NOSIMUL
 CLI("DELETE PART 'W:w7-8'")

 MOVE JOINT 3 TO 53  SIMUL
 MOVE JOINT 4 TO 39 NOSIMUL
 CLI("DELETE PART 'W:w7-7'")

 MOVE JOINT 3 TO 42 NOSIMUL
 CLI("DELETE PART 'W:w7-6'")

 MOVE JOINT 3 TO 30  SIMUL
 MOVE JOINT 4 TO 31 NOSIMUL
 CLI("DELETE PART 'W:w7-5'")

 MOVE JOINT 3 TO 18  SIMUL
 MOVE JOINT 4 TO 17 NOSIMUL
 CLI("DELETE PART 'W:w7-4'")

 MOVE JOINT 2 TO 20  SIMUL
 MOVE JOINT 3 TO 5  SIMUL
```

```
 MOVE JOINT 4 TO 0 NOSIMUL
 CLI("DELETE PART 'W:w7-3'")

write @1, (CLS)
write @1, ('CLEANING THE WASTE IN THE EIGHTH LAYER', cr)
 MOVE JOINT 1 TO 38  SIMUL
 MOVE JOINT 2 TO 2  SIMUL
 MOVE JOINT 3 TO 0  SIMUL
 MOVE JOINT 4 TO 19 NOSIMUL
 CLI("DELETE PART 'W:w8-4'")

 MOVE JOINT 3 TO 7  SIMUL
 MOVE JOINT 4 TO 33 NOSIMUL
 CLI("DELETE PART 'W:w8-5'")

 MOVE JOINT 2 TO 14  SIMUL
 MOVE JOINT 3 TO 21 NOSIMUL
 CLI("DELETE PART 'W:w8-6'")

 MOVE JOINT 2 TO 37  SIMUL
 MOVE JOINT 3 TO 43  SIMUL
 MOVE JOINT 4 TO 10 NOSIMUL
 CLI("DELETE PART 'W:w8-7'")

 MOVE JOINT 3 TO 53 NOSIMUL
 CLI("DELETE PART 'W:w8-8'")

 MOVE JOINT 2 TO 32  SIMUL
 MOVE JOINT 3 TO 60 NOSIMUL
 CLI("DELETE PART 'W:w8-9'")

 MOVE JOINT 2 TO 26  SIMUL
 MOVE JOINT 3 TO 67 NOSIMUL
 CLI("DELETE PART 'W:w8-10'")

 MOVE JOINT 2 TO 21  SIMUL
 MOVE JOINT 3 TO 63  SIMUL
 MOVE JOINT 4 TO 5 NOSIMUL
 CLI("DELETE PART 'W:w8-11'")

 MOVE JOINT 2 TO 3  SIMUL
 MOVE JOINT 3 TO 75 NOSIMUL
 CLI("DELETE PART 'W:w8-12'")

 MOVE JOINT 1 TO 34 NOSIMUL
 CLI("DELETE PART 'W:w8-13'")

write @1, (CLS)
write @1, ('CLEANING THE WASTE IN THE NINTH LAYER', cr)
 MOVE JOINT 1 TO 29  SIMUL
 MOVE JOINT 2 TO 1   SIMUL
 MOVE JOINT 3 TO 62  SIMUL
 MOVE JOINT 5 TO 15 NOSIMUL
 CLI("DELETE PART 'W:w9-13'")

 MOVE JOINT 1 TO 31  SIMUL
 MOVE JOINT 3 TO 60  SIMUL
```

```
MOVE JOINT 4 TO 6 NOSIMUL
CLI("DELETE PART 'W:w9-12'")

MOVE JOINT 3 TO 56  SIMUL
MOVE JOINT 4 TO 23 NOSIMUL
CLI("DELETE PART 'W:w9-11'")

MOVE JOINT 3 TO 48  SIMUL
MOVE JOINT 4 TO 34 NOSIMUL
CLI("DELETE PART 'W:w9-10'")

MOVE JOINT 3 TO 38  SIMUL
MOVE JOINT 4 TO 40 NOSIMUL
CLI("DELETE PART 'W:w9-9'")

MOVE JOINT 3 TO 26 NOSIMUL
CLI("DELETE PART 'W:w9-8'")

MOVE JOINT 3 TO 15  SIMUL
MOVE JOINT 4 TO 34 NOSIMUL
CLI("DELETE PART 'W:w9-7'")

MOVE JOINT 3 TO 5  SIMUL
MOVE JOINT 4 TO 16 NOSIMUL
CLI("DELETE PART 'W:w9-6'")

write @1, (CLS)
write @1, ('CLEANING THE WASTE IN THE BOTTOM LAYER', cr)

MOVE JOINT 1 TO 20  SIMUL
MOVE JOINT 2 TO 3  SIMUL
MOVE JOINT 3 TO 0  SIMUL
MOVE JOINT 4 TO 0 NOSIMUL
CLI("DELETE PART 'W:w10-8'")

MOVE JOINT 3 TO 9  SIMUL
MOVE JOINT 4 TO 11 NOSIMUL
CLI("DELETE PART 'W:w10-9'")

MOVE JOINT 3 TO 20 NOSIMUL
CLI("DELETE PART 'W:w10-10'")

MOVE JOINT 3 TO 30 NOSIMUL
CLI("DELETE PART 'W:w10-11'")

MOVE JOINT 1 TO 22  SIMUL
MOVE JOINT 2 TO 0  SIMUL
MOVE JOINT 3 TO 40 NOSIMUL
CLI("DELETE PART 'W:w10-12'")

MOVE JOINT 1 TO 20  SIMUL
MOVE JOINT 3 TO 43  SIMUL
MOVE JOINT 4 TO 5 NOSIMUL
CLI("DELETE PART 'W:w10-13'")

write @1, (CLS)
write @1, ('FOR CONTINUATION OF WASTE REMOVAL', cr)
```

```
write @1, ('THE COLUMN MUST BE LOWERED', cr)

  MOVE JOINT 1 TO 95  SIMUL
  MOVE JOINT 2 TO 63 SIMUL
  MOVE JOINT 3 TO 65 SIMUL
  MOVE JOINT 4 TO 90 SIMUL
  MOVE JOINT 8 TO -245 NOSIMUL
write @1, (CLS)
write @1, ('LOWERING THE COLUMN INTO THE TANK', cr)

  MOVE JOINT 2 TO 3 SIMUL
  MOVE JOINT 3 TO 43 SIMUL
  MOVE JOINT 4 TO 52 NOSIMUL
  CLI("DELETE PART 'W:w6-1'")

write @1, (CLS)
write @1, ('CLEANING REMAINING WASTE IN THE TANK', cr)

  MOVE JOINT 3 TO 59 SIMUL
  MOVE JOINT 4 TO 51 NOSIMUL
  CLI("DELETE PART 'W:w7-2'")

  MOVE JOINT 3 TO 52 SIMUL
  MOVE JOINT 4 TO 33 NOSIMUL
  CLI("DELETE PART 'W:w7-1'")

  MOVE JOINT 3 TO 73 SIMUL
  MOVE JOINT 4 TO 39 NOSIMUL
  CLI("DELETE PART 'W:w8-3'")

  MOVE JOINT 3 TO 64 SIMUL
  MOVE JOINT 4 TO 22 NOSIMUL
  CLI("DELETE PART 'W:w8-2'")

  MOVE JOINT 1 TO 91 SIMUL
  MOVE JOINT 3 TO 50 SIMUL
  MOVE JOINT 4 TO 19 NOSIMUL
  CLI("DELETE PART 'W:w8-1'")

  MOVE JOINT 4 TO 80 NOSIMUL
  MOVE JOINT 3 TO 89 NOSIMUL
  MOVE JOINT 4 TO 35 NOSIMUL
  CLI("DELETE PART 'W:w9-5'")

  MOVE JOINT 3 TO 79 SIMUL
  MOVE JOINT 4 TO 31 NOSIMUL
  CLI("DELETE PART 'W:w9-4'")

  MOVE JOINT 1 TO 90 SIMUL
  MOVE JOINT 3 TO 69 SIMUL
  MOVE JOINT 4 TO 25 NOSIMUL
  CLI("DELETE PART 'W:w9-3'")

  MOVE JOINT 1 TO 87 SIMUL
  MOVE JOINT 3 TO 57 SIMUL
  MOVE JOINT 4 TO 22 NOSIMUL
  CLI("DELETE PART 'W:w9-2'")
```

```
    MOVE JOINT 1 TO 82 SIMUL
    MOVE JOINT 3 TO 40 SIMUL
    MOVE JOINT 4 TO 18 NOSIMUL
    CLI("DELETE PART 'W:w9-1'")

    MOVE JOINT 3 TO 95 SIMUL
    MOVE JOINT 4 TO 77 NOSIMUL


    MOVE JOINT 3 TO 98 SIMUL
    MOVE JOINT 4 TO 29 NOSIMUL
    CLI("DELETE PART 'W:w10-7'")

    MOVE JOINT 3 TO 89 SIMUL
    MOVE JOINT 4 TO 34 NOSIMUL
    CLI("DELETE PART 'W:w10-6'")

    MOVE JOINT 3 TO 79 NOSIMUL
    CLI("DELETE PART 'W:w10-5'")

    MOVE JOINT 1 TO 80 SIMUL
    MOVE JOINT 3 TO 65 NOSIMUL
    CLI("DELETE PART 'W:w10-4'")

    MOVE JOINT 1 TO 78 SIMUL
    MOVE JOINT 3 TO 54 NOSIMUL
    CLI("DELETE PART 'W:w10-3'")

    MOVE JOINT 1 TO 73 SIMUL
    MOVE JOINT 3 TO 37 NOSIMUL
    CLI("DELETE PART 'W:w10-2'")

    MOVE JOINT 1 TO 68 SIMUL
    MOVE JOINT 3 TO 19 SIMUL
    MOVE JOINT 4 TO 21 NOSIMUL
    CLI("DELETE PART 'W:w10-1'")

write @1, (CLS)
write @1, ('WASTE MATERIAL CLEANUP COMPLETE', cr)
write @1, ('COMPLETE TANK WORKSPACE COVERED', cr)
DELAY 50000
-------------------------
------ END MAIN ------
END newentry
```

Program for Cleaning the Tank in Elbow Configuration

```
*****************************************************************************
*                                                                           *
```

```
*       description:    Schilling arm simulation in elbow configuration        *
*       programmer:     Amin AL-Halabi                                         *
*       date:           2/10/95                                                *
*                                                                              *
*                                                                              *
********************************************************************************/

PROGRAM newentry
BEGIN MAIN
________________

 $STEPSIZE = 0.1
 $MOTYPE = JOINT
 $SPEED_MODE = PERCENT
 $SPEED = 0.2
 UNITS = ENGLISH



 CLI("FULL SCREEN")
 DELAY 2000
 CLI("SELECT 'gpm'")
 CLI("ROTATE 'gpm' ABOUT Z_AXIS BY 90")

 DELAY 500
 MOVE JOINT 8 TO 380 NOSIMUL
 CLI("DELETE PART 'W:w1-13'")

 MOVE JOINT 4 TO 12 SIMUL
 MOVE JOINT 5 TO 11 NOSIMUL
 CLI("DELETE PART 'W:w1-12'")


 MOVE JOINT 8 TO 375 SIMUL
 MOVE JOINT 4 TO 30 SIMUL
 MOVE JOINT 5 TO 30 NOSIMUL
 CLI("DELETE PART 'W:w1-11'")

 MOVE JOINT 8 TO 325 SIMUL
 MOVE JOINT 4 TO 60 SIMUL
 MOVE JOINT 5 TO 60 NOSIMUL
 CLI("DELETE PART 'W:w1-10'")


 MOVE JOINT 4 TO 76 NOSIMUL
 MOVE JOINT 3 TO 10 SIMUL
 MOVE JOINT 8 TO 275 SIMUL
 MOVE JOINT 5 TO 85 NOSIMUL
 CLI("DELETE PART 'W:w1-9'")

 MOVE JOINT 3 TO 31 SIMUL
 MOVE JOINT 4 TO 49 SIMUL
 MOVE JOINT 5 TO 80 NOSIMUL
 CLI("DELETE PART 'W:w1-8'")

 MOVE JOINT 3 TO 61 SIMUL
 MOVE JOINT 4 TO 3 SIMUL
```

```
    MOVE JOINT 5 TO 64 NOSIMUL
    MOVE JOINT 8 TO 260 NOSIMUL
    CLI("DELETE PART 'W:w1-7'")


    MOVE JOINT 3 TO 85 SIMUL
    MOVE JOINT 5 TO 90 SIMUL
    MOVE JOINT 8 TO 200 NOSIMUL
    MOVE JOINT 8 TO 165 NOSIMUL
    MOVE JOINT 2 TO 5 SIMUL
    MOVE JOINT 8 TO 150 NOSIMUL
    CLI("DELETE PART 'W:w1-6'")

    MOVE JOINT 2 TO 26 SIMUL
    MOVE JOINT 3 TO 61 NOSIMUL
    CLI("DELETE PART 'W:w1-5'")

    MOVE JOINT 2 TO 49 SIMUL
    MOVE JOINT 3 TO 31 SIMUL
    MOVE JOINT 5 TO 82 NOSIMUL
    CLI("DELETE PART 'W:w1-4'")

    MOVE JOINT 5 TO 90 SIMUL
    MOVE JOINT 2 TO 90 SIMUL
    MOVE JOINT 3 TO 1 SIMUL
    MOVE JOINT 8 TO 50 NOSIMUL
    MOVE JOINT 8 TO 30 NOSIMUL
    CLI("DELETE PART 'W:w1-3'")

    MOVE JOINT 2 TO 78 SIMUL
    MOVE JOINT 8 TO 5 SIMUL
    MOVE JOINT 4 TO -2 SIMUL
    MOVE JOINT 1 TO 16 NOSIMUL
    CLI("DELETE PART 'W:w1-2'")

    MOVE JOINT 1 TO 27 SIMUL
    MOVE JOINT 2 TO 66 SIMUL
    MOVE JOINT 8 TO 0 NOSIMUL
    CLI("DELETE PART 'W:w1-1'")
DELAY 1000


    MOVE JOINT 5 TO 84 SIMUL
    MOVE JOINT 2 TO 61 NOSIMUL
    CLI("DELETE PART 'W:w2-1'")

    MOVE JOINT 1 TO 17 SIMUL
    MOVE JOINT 2 TO 72 SIMUL
    MOVE JOINT 8 TO 5 NOSIMUL
    CLI("DELETE PART 'W:w2-2'")

    MOVE JOINT 1 TO 3 SIMUL
    MOVE JOINT 2 TO 85 SIMUL
    MOVE JOINT 8 TO 13 NOSIMUL
    CLI("DELETE PART 'W:w2-3'")

    MOVE JOINT 2 TO 95 SIMUL
```

```
MOVE JOINT 1 TO 0 NOSIMUL
MOVE JOINT 2 TO 67 SIMUL
MOVE JOINT 5 TO 66 SIMUL
MOVE JOINT 8 TO 130 NOSIMUL
CLI("DELETE PART 'W:w2-4'")

MOVE JOINT 3 TO 15 NOSIMUL
MOVE JOINT 2 TO 31 SIMUL
MOVE JOINT 3 TO 45 SIMUL
MOVE JOINT 5 TO 76 SIMUL
MOVE JOINT 8 TO 160 NOSIMUL
CLI("DELETE PART 'W:w2-5'")

MOVE JOINT 2 TO 11 SIMUL
MOVE JOINT 3 TO 67 SIMUL
MOVE JOINT 8 TO 166 NOSIMUL
CLI("DELETE PART 'W:w2-6'")

MOVE JOINT 2 TO 0 SIMUL
MOVE JOINT 3 TO 87 NOSIMUL
MOVE JOINT 3 TO 67 SIMUL
MOVE JOINT 5 TO 66 SIMUL
MOVE JOINT 8 TO 211 NOSIMUL
CLI("DELETE PART 'W:w2-7'")

MOVE JOINT 3 TO 51 SIMUL
MOVE JOINT 5 TO 48 SIMUL
MOVE JOINT 8 TO 262 NOSIMUL
CLI("DELETE PART 'W:w2-8'")

MOVE JOINT 3 TO 18 SIMUL
MOVE JOINT 4 TO 45 SIMUL
MOVE JOINT 5 TO 62 SIMUL
MOVE JOINT 8 TO 275 NOSIMUL
CLI("DELETE PART 'W:w2-9'")

MOVE JOINT 3 TO 0 SIMUL
MOVE JOINT 4 TO 63 SIMUL
MOVE JOINT 8 TO 283 NOSIMUL
CLI("DELETE PART 'W:w2-10'")

MOVE JOINT 4 TO 35 SIMUL
MOVE JOINT 5 TO 35 SIMUL
MOVE JOINT 8 TO 327 NOSIMUL
CLI("DELETE PART 'W:w2-11'")

MOVE JOINT 4 TO 16 SIMUL
MOVE JOINT 5 TO 16 SIMUL
MOVE JOINT 8 TO 340 NOSIMUL
CLI("DELETE PART 'W:w2-12'")

MOVE JOINT 4 TO 0 SIMUL
MOVE JOINT 5 TO 0 NOSIMUL
CLI("DELETE PART 'W:w2-13'")

MOVE JOINT 8 TO 305 NOSIMUL
CLI("DELETE PART 'W:w3-13'")
```

```
MOVE JOINT 4 TO 16 SIMUL
MOVE JOINT 5 TO 16 NOSIMUL
CLI("DELETE PART 'W:w3-12'")

MOVE JOINT 4 TO 35 SIMUL
MOVE JOINT 5 TO 34 SIMUL
MOVE JOINT 8 TO 285 NOSIMUL
CLI("DELETE PART 'W:w3-11'")

MOVE JOINT 4 TO 63 SIMUL
MOVE JOINT 5 TO 62 SIMUL
MOVE JOINT 8 TO 240 NOSIMUL
CLI("DELETE PART 'W:w3-10'")

MOVE JOINT 3 TO 19 SIMUL
MOVE JOINT 4 TO 42 NOSIMUL
CLI("DELETE PART 'W:w3-9'")

MOVE JOINT 3 TO 34 SIMUL
MOVE JOINT 5 TO 76 SIMUL
MOVE JOINT 8 TO 199 NOSIMUL
CLI("DELETE PART 'W:w3-8'")

MOVE JOINT 3 TO 58 SIMUL
MOVE JOINT 4 TO 19 SIMUL
MOVE JOINT 8 TO 161 NOSIMUL
CLI("DELETE PART 'W:w3-7'")

MOVE JOINT 2 TO 21 SIMUL
MOVE JOINT 3 TO 35 NOSIMUL
CLI("DELETE PART 'W:w3-6'")

MOVE JOINT 2 TO 34 SIMUL
MOVE JOINT 4 TO 7 SIMUL
MOVE JOINT 8 TO 124 NOSIMUL
CLI("DELETE PART 'W:w3-5'")

MOVE JOINT 2 TO 60 SIMUL
MOVE JOINT 3 TO 10 SIMUL
MOVE JOINT 8 TO 84 NOSIMUL
CLI("DELETE PART 'W:w3-4'")

MOVE JOINT 2 TO 87 SIMUL
MOVE JOINT 3 TO 0 SIMUL
MOVE JOINT 4 TO 0 SIMUL
MOVE JOINT 8 TO 0 NOSIMUL
MOVE JOINT 1 TO 4 SIMUL
MOVE JOINT 5 TO 84 SIMUL
MOVE JOINT 2 TO 80 NOSIMUL
CLI("DELETE PART 'W:w3-3'")

MOVE JOINT 1 TO 18 SIMUL
MOVE JOINT 2 TO 65 SIMUL
MOVE JOINT 8 TO -6 NOSIMUL
CLI("DELETE PART 'W:w3-2'")
```

```
MOVE JOINT 1 TO 30 SIMUL
MOVE JOINT 2 TO 56 SIMUL
MOVE JOINT 8 TO -34 NOSIMUL
CLI("DELETE PART 'W:w3-1'")

DELAY 1000

MOVE JOINT 1 TO 33 SIMUL
MOVE JOINT 2 TO 46 SIMUL
MOVE JOINT 5 TO 81 NOSIMUL
CLI("DELETE PART 'W:w4-1'")

MOVE JOINT 1 TO 18 SIMUL
MOVE JOINT 2 TO 64 NOSIMUL
CLI("DELETE PART 'W:w4-2'")

MOVE JOINT 1 TO 5 SIMUL
MOVE JOINT 2 TO 76 SIMUL
MOVE JOINT 8 TO -21 NOSIMUL
CLI("DELETE PART 'W:w4-3'")

MOVE JOINT 1 TO 0 SIMUL
MOVE JOINT 2 TO 68 SIMUL
MOVE JOINT 5 TO 68 SIMUL
MOVE JOINT 8 TO 51 NOSIMUL
CLI("DELETE PART 'W:w4-4'")

MOVE JOINT 2 TO 55 SIMUL
MOVE JOINT 5 TO 55 SIMUL
MOVE JOINT 8 TO 118 NOSIMUL
CLI("DELETE PART 'W:w4-5'")

MOVE JOINT 2 TO 30 SIMUL
MOVE JOINT 3 TO 52 SIMUL
MOVE JOINT 5 TO 66 NOSIMUL
MOVE JOINT 2 TO 17 NOSIMUL
CLI("DELETE PART 'W:w4-6'")

MOVE JOINT 2 TO 7 SIMUL
MOVE JOINT 5 TO 60 SIMUL
MOVE JOINT 8 TO 152 NOSIMUL
CLI("DELETE PART 'W:w4-7'")

MOVE JOINT 2 TO 0 SIMUL
MOVE JOINT 3 TO 68 NOSIMUL
MOVE JOINT 3 TO 49 SIMUL
MOVE JOINT 5 TO 50 SIMUL
MOVE JOINT 8 TO 184 NOSIMUL
CLI("DELETE PART 'W:w4-8'")

MOVE JOINT 3 TO 37 SIMUL
MOVE JOINT 5 TO 38 SIMUL
MOVE JOINT 8 TO 217 NOSIMUL
CLI("DELETE PART 'W:w4-9'")

MOVE JOINT 3 TO 26 SIMUL
MOVE JOINT 5 TO 26 SIMUL
```

```
MOVE JOINT 8 TO 240 NOSIMUL
CLI("DELETE PART 'W:w4-10'")

MOVE JOINT 3 TO 0 SIMUL
MOVE JOINT 4 TO 34 SIMUL
MOVE JOINT 5 TO 33 SIMUL
MOVE JOINT 8 TO 245 NOSIMUL
CLI("DELETE PART 'W:w4-11'")

MOVE JOINT 4 TO 15 SIMUL
MOVE JOINT 5 TO 15 SIMUL
MOVE JOINT 8 TO 260 NOSIMUL
CLI("DELETE PART 'W:w4-12'")

MOVE JOINT 4 TO 0 SIMUL
MOVE JOINT 5 TO 0 SIMUL
MOVE JOINT 8 TO 265 NOSIMUL
CLI("DELETE PART 'W:w4-13'")


DELAY 1000
MOVE JOINT 8 TO 223 NOSIMUL
CLI("DELETE PART 'W:w5-13'")

MOVE JOINT 3 TO 7 SIMUL
MOVE JOINT 5 TO 7 NOSIMUL
CLI("DELETE PART 'W:w5-12'")

MOVE JOINT 3 TO 16 SIMUL
MOVE JOINT 5 TO 16 SIMUL
MOVE JOINT 8 TO 217 NOSIMUL
CLI("DELETE PART 'W:w5-11'")

MOVE JOINT 3 TO 26 SIMUL
MOVE JOINT 5 TO 26 SIMUL
MOVE JOINT 8 TO 200 NOSIMUL
CLI("DELETE PART 'W:w5-10'")

MOVE JOINT 3 TO 37 SIMUL
MOVE JOINT 5 TO 37 SIMUL
MOVE JOINT 8 TO 181 NOSIMUL
CLI("DELETE PART 'W:w5-9'")

MOVE JOINT 3 TO 49 SIMUL
MOVE JOINT 5 TO 49 SIMUL
MOVE JOINT 8 TO 148 NOSIMUL
CLI("DELETE PART 'W:w5-8'")

MOVE JOINT 3 TO 66 SIMUL
MOVE JOINT 5 TO 66 SIMUL
MOVE JOINT 8 TO 90 NOSIMUL
CLI("DELETE PART 'W:w5-7'")

MOVE JOINT 2 TO 19 SIMUL
MOVE JOINT 3 TO 46 SIMUL
MOVE JOINT 5 TO 64 NOSIMUL
CLI("DELETE PART 'W:w5-6'")
```

```
MOVE JOINT 2 TO 38 SIMUL
MOVE JOINT 3 TO 27 SIMUL
MOVE JOINT 8 TO 69 NOSIMUL
CLI("DELETE PART 'W:w5-5'")

MOVE JOINT 2 TO 59 SIMUL
MOVE JOINT 3 TO 13 SIMUL
MOVE JOINT 5 TO 71 SIMUL
MOVE JOINT 8 TO 14 NOSIMUL
CLI("DELETE PART 'W:w5-4'")


MOVE JOINT 2 TO 65 SIMUL
MOVE JOINT 8 TO -2 NOSIMUL
MOVE JOINT 1 TO 12 SIMUL
MOVE JOINT 2 TO 49 NOSIMUL
CLI("DELETE PART 'W:w5-3'")

MOVE JOINT 1 TO 26 SIMUL
MOVE JOINT 2 TO 34 SIMUL
MOVE JOINT 8 TO -12 NOSIMUL
CLI("DELETE PART 'W:w5-2'")

MOVE JOINT 1 TO 38 SIMUL
MOVE JOINT 2 TO 25 SIMUL
MOVE JOINT 5 TO 75 SIMUL
MOVE JOINT 8 TO -44 NOSIMUL
CLI("DELETE PART 'W:w5-1'")

DELAY 1000
MOVE JOINT 8 TO -83 NOSIMUL
CLI("DELETE PART 'W:w6-1'")

MOVE JOINT 1 TO 30 SIMUL
MOVE JOINT 2 TO 26 SIMUL
MOVE JOINT 5 TO 67 SIMUL
MOVE JOINT 8 TO -32 NOSIMUL
CLI("DELETE PART 'W:w6-2'")

MOVE JOINT 1 TO 14 SIMUL
MOVE JOINT 2 TO 44 NOSIMUL
CLI("DELETE PART 'W:w6-3'")

MOVE JOINT 1 TO 0 SIMUL
MOVE JOINT 2 TO 60 SIMUL
MOVE JOINT 5 TO 72 NOSIMUL
CLI("DELETE PART 'W:w6-4'")

MOVE JOINT 2 TO 47 SIMUL
MOVE JOINT 5 TO 58 SIMUL
MOVE JOINT 8 TO 32 NOSIMUL
CLI("DELETE PART 'W:w6-5'")

MOVE JOINT 2 TO 38 SIMUL
MOVE JOINT 3 TO 12 SIMUL
MOVE JOINT 5 TO 49 SIMUL
```

```
MOVE JOINT 8 TO 78 NOSIMUL
CLI("DELETE PART 'W:w6-6'")

MOVE JOINT 2 TO 30 SIMUL
MOVE JOINT 5 TO 42 SIMUL
MOVE JOINT 8 TO 110 NOSIMUL
CLI("DELETE PART 'W:w6-7'")

MOVE JOINT 2 TO 23 SIMUL
MOVE JOINT 5 TO 37 SIMUL
MOVE JOINT 8 TO 134 NOSIMUL
CLI("DELETE PART 'W:w6-8'")

MOVE JOINT 2 TO 0 SIMUL
MOVE JOINT 3 TO 37 SIMUL
MOVE JOINT 8 TO 139 NOSIMUL
CLI("DELETE PART 'W:w6-9'")

MOVE JOINT 3 TO 26 SIMUL
MOVE JOINT 5 TO 26 SIMUL
MOVE JOINT 8 TO 162 NOSIMUL
CLI("DELETE PART 'W:w6-10'")

MOVE JOINT 3 TO 17 SIMUL
MOVE JOINT 5 TO 17 SIMUL
MOVE JOINT 8 TO 175 NOSIMUL
CLI("DELETE PART 'W:w6-11'")

MOVE JOINT 3 TO 7 SIMUL
MOVE JOINT 5 TO 7 SIMUL
MOVE JOINT 8 TO 183 NOSIMUL
CLI("DELETE PART 'W:w6-12'")

MOVE JOINT 3 TO 0 SIMUL
MOVE JOINT 5 TO 0 SIMUL
MOVE JOINT 8 TO 185 NOSIMUL
CLI("DELETE PART 'W:w6-13'")

DELAY 1000
MOVE JOINT 8 TO 142 NOSIMUL
CLI("DELETE PART 'W:w7-13'")

MOVE JOINT 2 TO 5 SIMUL
MOVE JOINT 5 TO 5 NOSIMUL
CLI("DELETE PART 'W:w7-12'")

MOVE JOINT 2 TO 11 SIMUL
MOVE JOINT 5 TO 11 SIMUL
MOVE JOINT 8 TO 139 NOSIMUL
CLI("DELETE PART 'W:w7-11'")

MOVE JOINT 2 TO 17 SIMUL
MOVE JOINT 5 TO 17 SIMUL
MOVE JOINT 8 TO 131 NOSIMUL
CLI("DELETE PART 'W:w7-10'")

MOVE JOINT 2 TO 24 SIMUL
```

```
MOVE JOINT 5 TO 24 SIMUL
MOVE JOINT 8 TO 117 NOSIMUL
CLI("DELETE PART 'W:w7-9'")

MOVE JOINT 2 TO 30 SIMUL
MOVE JOINT 5 TO 30 SIMUL
MOVE JOINT 8 TO 100 NOSIMUL
CLI("DELETE PART 'W:w7-8'")

MOVE JOINT 2 TO 38 SIMUL
MOVE JOINT 5 TO 38 SIMUL
MOVE JOINT 8 TO 74 NOSIMUL
CLI("DELETE PART 'W:w7-7'")

MOVE JOINT 2 TO 46 SIMUL
MOVE JOINT 5 TO 46 SIMUL
MOVE JOINT 8 TO 42 NOSIMUL
CLI("DELETE PART 'W:w7-6'")

MOVE JOINT 2 TO 55 SIMUL
MOVE JOINT 5 TO 55 SIMUL
MOVE JOINT 8 TO 0 NOSIMUL
CLI("DELETE PART 'W:w7-5'")

MOVE JOINT 2 TO 68 SIMUL
MOVE JOINT 5 TO 68 SIMUL
MOVE JOINT 8 TO -67 NOSIMUL
CLI("DELETE PART 'W:w7-4'")

MOVE JOINT 1 TO 10 SIMUL
MOVE JOINT 2 TO 61 SIMUL
MOVE JOINT 8 TO -87 NOSIMUL
CLI("DELETE PART 'W:w7-3'")

MOVE JOINT 1 TO 24 SIMUL
MOVE JOINT 2 TO 46 SIMUL
MOVE JOINT 8 TO -93 NOSIMUL
CLI("DELETE PART 'W:w7-2'")

MOVE JOINT 1 TO 36 SIMUL
MOVE JOINT 2 TO 36 SIMUL
MOVE JOINT 8 TO -122 NOSIMUL
CLI("DELETE PART 'W:w7-1'")


DELAY 1000
MOVE JOINT 8 TO -161 NOSIMUL
CLI("DELETE PART 'W:w8-1'")

MOVE JOINT 1 TO 20 SIMUL
MOVE JOINT 2 TO 56 SIMUL
MOVE JOINT 5 TO 76 NOSIMUL
CLI("DELETE PART 'W:w8-2'")

MOVE JOINT 1 TO 9 SIMUL
MOVE JOINT 2 TO 75 NOSIMUL
```

```
MOVE JOINT 1 TO 6 SIMUL
MOVE JOINT 2 TO 72 NOSIMUL
CLI("DELETE PART 'W:w8-3'")

MOVE JOINT 1 TO -5 SIMUL
MOVE JOINT 2 TO 84 SIMUL
MOVE JOINT 3 TO -4 SIMUL
MOVE JOINT 4 TO -4 SIMUL
MOVE JOINT 5 TO 69 SIMUL
MOVE JOINT 8 TO -150 NOSIMUL
CLI("DELETE PART 'W:w8-4'")

MOVE JOINT 2 TO 69 SIMUL
MOVE JOINT 5 TO 56 SIMUL
MOVE JOINT 8 TO -66 NOSIMUL
CLI("DELETE PART 'W:w8-5'")

MOVE JOINT 2 TO 58 SIMUL
MOVE JOINT 5 TO 46 SIMUL
MOVE JOINT 8 TO -14 NOSIMUL
CLI("DELETE PART 'W:w8-6'")

MOVE JOINT 3 TO 20 NOSIMUL
MOVE JOINT 2 TO 17 SIMUL
MOVE JOINT 3 TO 51 SIMUL
MOVE JOINT 5 TO 59 NOSIMUL
CLI("DELETE PART 'W:w8-7'")

MOVE JOINT 2 TO -1 SIMUL
MOVE JOINT 3 TO 69 NOSIMUL
CLI("DELETE PART 'W:w8-8'")

MOVE JOINT 1 TO 0 NOSIMUL
MOVE JOINT 3 TO 41 SIMUL
MOVE JOINT 5 TO 37 SIMUL
MOVE JOINT 8 TO 57 NOSIMUL
CLI("DELETE PART 'W:w8-9'")

MOVE JOINT 4 TO 15 NOSIMUL
MOVE JOINT 3 TO 7 SIMUL
MOVE JOINT 4 TO 48 SIMUL
MOVE JOINT 5 TO 54 NOSIMUL
CLI("DELETE PART 'W:w8-10'")

MOVE JOINT 4 TO 24 SIMUL
MOVE JOINT 5 TO 32 SIMUL
MOVE JOINT 8 TO 90 NOSIMUL
CLI("DELETE PART 'W:w8-11'")

MOVE JOINT 2 TO -5 SIMUL
MOVE JOINT 3 TO -3 SIMUL
MOVE JOINT 4 TO 38 NOSIMUL
CLI("DELETE PART 'W:w8-12'")

MOVE JOINT 4 TO 22 SIMUL
MOVE JOINT 5 TO 16 SIMUL
MOVE JOINT 8 TO 101 NOSIMUL
```

```
CLI("DELETE PART 'W:w8-13'")

DELAY 1000
MOVE JOINT 2 TO 0 SIMUL
MOVE JOINT 3 TO 0 SIMUL
MOVE JOINT 4 TO 0 SIMUL
MOVE JOINT 5 TO 0 SIMUL
MOVE JOINT 8 TO 66 NOSIMUL
CLI("DELETE PART 'W:w9-13'")

MOVE JOINT 2 TO 5 SIMUL
MOVE JOINT 5 TO 5 NOSIMUL
CLI("DELETE PART 'W:w9-12'")

MOVE JOINT 2 TO 11 SIMUL
MOVE JOINT 5 TO 11 SIMUL
MOVE JOINT 8 TO 63 NOSIMUL
CLI("DELETE PART 'W:w9-11'")

MOVE JOINT 2 TO 17 SIMUL
MOVE JOINT 5 TO 17 SIMUL
MOVE JOINT 8 TO 55 NOSIMUL
CLI("DELETE PART 'W:w9-10'")

MOVE JOINT 2 TO 24 SIMUL
MOVE JOINT 5 TO 24 SIMUL
MOVE JOINT 8 TO 41 NOSIMUL
CLI("DELETE PART 'W:w9-9'")

MOVE JOINT 2 TO 30 SIMUL
MOVE JOINT 5 TO 30 SIMUL
MOVE JOINT 8 TO 24 NOSIMUL
CLI("DELETE PART 'W:w9-8'")

MOVE JOINT 2 TO 38 SIMUL
MOVE JOINT 5 TO 38 SIMUL
MOVE JOINT 8 TO -2 NOSIMUL
CLI("DELETE PART 'W:w9-7'")

MOVE JOINT 2 TO 46 SIMUL
MOVE JOINT 5 TO 46 SIMUL
MOVE JOINT 8 TO -34 NOSIMUL
CLI("DELETE PART 'W:w9-6'")

MOVE JOINT 2 TO 55 SIMUL
MOVE JOINT 5 TO 55 SIMUL
MOVE JOINT 8 TO -76 NOSIMUL
CLI("DELETE PART 'W:w9-5'")

MOVE JOINT 2 TO 68 SIMUL
MOVE JOINT 5 TO 68 SIMUL
MOVE JOINT 8 TO -143 NOSIMUL
CLI("DELETE PART 'W:w9-4'")

MOVE JOINT 1 TO 10 SIMUL
MOVE JOINT 2 TO 61 SIMUL
```

```
MOVE JOINT 8 TO -167 NOSIMUL
CLI("DELETE PART 'W:w9-3'")

MOVE JOINT 1 TO 24 SIMUL
MOVE JOINT 2 TO 46 SIMUL
MOVE JOINT 8 TO -173 NOSIMUL
CLI("DELETE PART 'W:w9-2'")

MOVE JOINT 1 TO 36 SIMUL
MOVE JOINT 2 TO 36 SIMUL
MOVE JOINT 8 TO -202 NOSIMUL
CLI("DELETE PART 'W:w9-1'")


DELAY 1000
MOVE JOINT 1 TO 43 SIMUL
MOVE JOINT 2 TO 23 SIMUL
MOVE JOINT 5 TO 64 SIMUL
MOVE JOINT 8 TO -219 NOSIMUL
CLI("DELETE PART 'W:w10-1'")

MOVE JOINT 1 TO 24 SIMUL
MOVE JOINT 2 TO 47 NOSIMUL
CLI("DELETE PART 'W:w10-2'")

MOVE JOINT 2 TO 55 NOSIMUL
MOVE JOINT 1 TO 8 SIMUL
MOVE JOINT 2 TO 65 SIMUL
MOVE JOINT 5 TO 72 NOSIMUL
CLI("DELETE PART 'W:w10-3'")

MOVE JOINT 8 TO -179 NOSIMUL
MOVE JOINT 1 TO 1 SIMUL
MOVE JOINT 5 TO 65 NOSIMUL
CLI("DELETE PART 'W:w10-4'")

MOVE JOINT 2 TO 55 SIMUL
MOVE JOINT 1 TO 0 SIMUL
MOVE JOINT 5 TO 55 SIMUL
MOVE JOINT 8 TO -120 NOSIMUL
CLI("DELETE PART 'W:w10-5'")

MOVE JOINT 2 TO 46 SIMUL
MOVE JOINT 5 TO 46 SIMUL
MOVE JOINT 8 TO -78 NOSIMUL
CLI("DELETE PART 'W:w10-6'")

MOVE JOINT 2 TO 38 SIMUL
MOVE JOINT 5 TO 38 SIMUL
MOVE JOINT 8 TO -46 NOSIMUL
CLI("DELETE PART 'W:w10-7'")

MOVE JOINT 2 TO 30 SIMUL
MOVE JOINT 5 TO 30 SIMUL
MOVE JOINT 8 TO -20 NOSIMUL
CLI("DELETE PART 'W:w10-8'")
```

```
MOVE JOINT 2 TO 24 SIMUL
MOVE JOINT 5 TO 24 SIMUL
MOVE JOINT 8 TO -3 NOSIMUL
CLI("DELETE PART 'W:w10-9'")

MOVE JOINT 2 TO 17 SIMUL
MOVE JOINT 5 TO 17 SIMUL
MOVE JOINT 8 TO 11 NOSIMUL
CLI("DELETE PART 'W:w10-10'")

MOVE JOINT 2 TO 11 SIMUL
MOVE JOINT 5 TO 11 SIMUL
MOVE JOINT 8 TO 19 NOSIMUL
CLI("DELETE PART 'W:w10-11'")

MOVE JOINT 2 TO 5 SIMUL
MOVE JOINT 5 TO 5 NOSIMUL
MOVE JOINT 8 TO 22 NOSIMUL
CLI("DELETE PART 'W:w10-12'")

MOVE JOINT 2 TO 0 SIMUL
MOVE JOINT 3 TO 0 SIMUL
MOVE JOINT 4 TO 0 SIMUL
MOVE JOINT 5 TO 0 NOSIMUL
CLI("DELETE PART 'W:w10-13'")

MOVE JOINT 8 TO 540 NOSIMUL

------------------------------
-------- END MAIN --------
END newentry
```

Appendix B

DH Parameters for the Modified LRM Design

Figure B.1 shows the joint frame attachment for the modified long reach manipulator. It can be noted that two sets of frame assignments are presented in this appendix. The first set includes the mast links where the base frame is placed at the mast roll joint (frame 0 in figure B.1). The second set only describes the last five links with the base at the shoulder pitch joint. DH tables for the first and the second frame assignments are shown in Table B.1 and Table B.2 respectively.

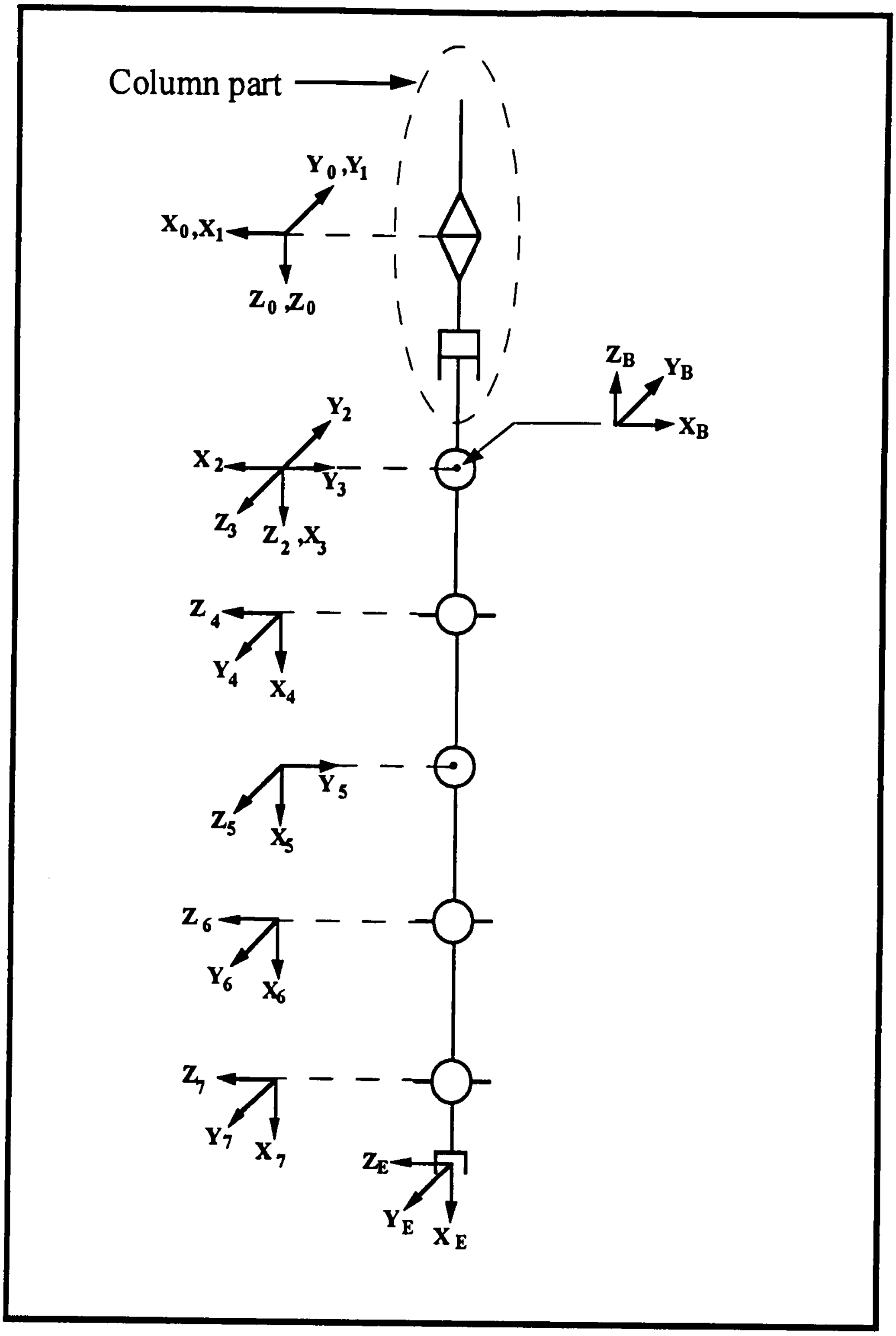


Fig. B.1. Frame Attachment for the modified LRM manipulator

Table B.1 D-H table of the modified LRM considering base at the mast roll

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0°	0	(θ_1)
2	0	0°	(L_1)	0
3	0	90°	0	$(\theta_3 + 90°)$
4	L_2	90°	0	(θ_4)
5	L_3	-90°	0	(θ_5)
6	L_4	90°	0	(θ_6)
7	L_5	0°	0	(θ_7)
E	L_E	0°	0	0

Table B.2 D-H table of the modified LRM considering base at the shoulder pitch

i	a_{i-1}	α_{i-1}	d_i	θ_i
3	0	90°	0	$(\theta_3 - 90°)$
4	L_2	90°	0	(θ_4)
5	L_3	-90°	0	(θ_5)
6	L_4	90°	0	(θ_6)
7	L_5	0°	0	(θ_7)
E	L_E	0°	0	0

Based on Table B.2, the transformation matrix from shoulder to end effector can be calculated as follows:

$$^{B}T_{E} = {}^{B}T_{3} * {}^{3}T_{4} * {}^{4}T_{5} * {}^{5}T_{6} * {}^{6}T_{7} * {}^{7}T_{E}$$

It is necessary to note the following equations for substitution :

$$\cos(\theta_3 - 90) = \sin\theta_3$$
$$\sin(\theta_3 - 90) = -\cos\theta_3$$
$$s_i = \sin(\theta_i)$$
$$c_i = \cos(\theta_i)$$
$$c_{ij} = \cos(\theta_i + \theta_j)$$
$$s_{ij} = \sin(\theta_i + \theta_j)$$

Starting at shoulder pitch the transformation matrices from each frame to the next are as follows:

$$^{B}T_{3} = \begin{bmatrix} s_3 & c_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -c_3 & s_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{3}T_{4} = \begin{bmatrix} c_4 & -s_4 & 0 & L_2 \\ 0 & 0 & -1 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ {}^{4}T_{5} = \begin{bmatrix} c_5 & -s_5 & 0 & L_3 \\ 0 & 0 & 1 & 0 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{5}T_{6} = \begin{bmatrix} c_6 & -s_6 & 0 & L_4 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{6}T_{7} = \begin{bmatrix} c_7 & -s_7 & 0 & L_5 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{7}T_{E} = \begin{bmatrix} 1 & 0 & 0 & L_E \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Using each consecutive transformation, the overall transformations can be found as follows:

$$ {}^{B}T_{5} = \begin{bmatrix} s_3c_4c_5 + c_3s_5 & -s_3c_4s_5 + c_3c_5 & -s_3s_4 & s_3c_4L_3 + s_3L_2 \\ -s_4c_5 & s_4s_5 & -c_4 & s_3s_30 \\ -c_3c_4c_5 + s_3s_5 & c_3c_4s_5 + s_3c_5 & c_3s_4 & -c_3c_4L_3 - c_3L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{B}T_{5} = \begin{bmatrix} (A11) & (A12) & (A13) & (A14) \\ (A21) & (A22) & (A23) & (A24) \\ (A31) & (A32) & (A33) & (A34) \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$
{}^{B}T_6 = \begin{bmatrix} (A11)c_6 + (A13)s_6 & -(A11)s_6 + (A13)c_6 & -(A12) & (A11)L_4 + (A14) \\ (A21)c_6 + (A23)s_6 & -(A21)s_6 + (A23)c_6 & -(A22) & (A21)L_4 + (A24) \\ (A31)c_6 + (A33)s_6 & -(A21)s_6 + (A23)c_6 & -(A32) & (A31)L_4 + (A34) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{B}T_6 = \begin{bmatrix} (B11) & (B12) & (B13) & (B14) \\ (B21) & (B22) & (B23) & (B24) \\ (B31) & (B32) & (B33) & (B34) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{B}T_7 = \begin{bmatrix} (B11)c_7 + (B12)s_7 & -(B11)s_7 + (B12)c_7 & (B13) & (B11)L_5 + (B14) \\ (B21)c_7 + (B22)s_7 & -(B21)s_7 + (B22)c_7 & (B23) & (B21)L_5 + (B24) \\ (B31)c_7 + (B32)s_7 & -(B21)s_7 + (B22)c_7 & (B33) & (B31)L_5 + (B34) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{B}T_7 = \begin{bmatrix} (D11) & (D12) & (D13) & (D14) \\ (D21) & (D22) & (D23) & (D24) \\ (D31) & (D32) & (D33) & (D34) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{B}T_E = \begin{bmatrix} (D11) & (D12) & (D13) & (D11)L_E + (D14) \\ (D21) & (D22) & (D23) & (D21)L_E + (D24) \\ (D31) & (D32) & (D33) & (D31)L_E + (D34) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

These terms of the transformation from base to the end effector can be expanded. To satisfy the orientation constraints mentioned in chapter 5 the necessary terms are expanded. As a reminder the frame assignment is shown in figure A.2.

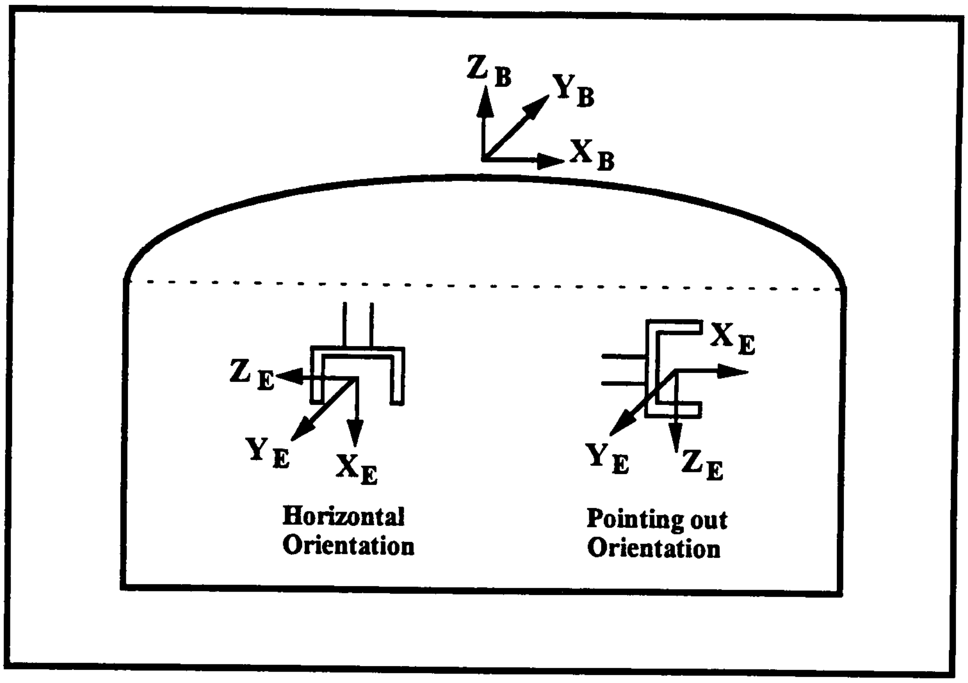


Fig. B.2. Assignment of base and end effector frames

For maintaining the horizontal orientation of the end effector, the following equation must be satisfied.

$$X_{E_Z} = -1$$

$$\begin{aligned}
X_{E_Z} &= (D31) \\
X_{E_Z} &= (B31)c_7 + (B32)s_7 \\
&= (A31)c_6c_7 + (A33)s_6c_7 - (A31)s_6s_7 + (A33)c_6s_7 \\
&= -c_3c_4c_5c_6c_7 + s_3s_5c_6c_7 + c_3s_4s_6c_7 + c_3c_4c_5s_6s_7 - s_3s_5s_6s_7 + c_3s_4c_6s_7 \\
X_{E_Z} &= (-c_3c_4c_5 + s_3s_5)(c_6c_7 - s_6s_7) + (c_3s_4)(s_6c_7 + c_6s_7)
\end{aligned}$$

The first equation can be reduced to:

$$(-c_3c_4c_5 + s_3s_5)c_{67} + c_3s_4s_{67} = -1$$

Due to the complexity of solving this equation, it can be replaced with two constraint equations.

$$
\begin{aligned}
Z_{E_Z} &= 0 \\
&= (D33) = -(A32) \\
&= s_3c_4c_5 + s_3s_5 = 0
\end{aligned}
$$

$$
\begin{aligned}
Y_{E_Z} &= (D32) \\
Y_{E_Z} &= -(B31)s_7 + (B32)c_7 \\
&= -(A31)c_6s_7 - (A33)s_6s_7 - (A31)s_6c_7 + (A33)c_6c_7 \\
&= c_3c_4c_5c_6s_7 - s_3s_5c_6s_7 - c_3s_4s_6s_7 + c_3c_4c_5s_6c_7 - s_3s_5s_6c_7 + c_3s_4c_6c_7 \\
Y_{E_Z} &= (c_3c_4c_5 - s_3s_5)(c_6s_7 + s_6c_7) + (c_3s_4)(-s_6s_7 + c_6c_7) \\
&= (c_3c_4c_5 - s_3s_5)s_{67} + c_3s_4c_{67} = 0
\end{aligned}
$$

Two constraints given by the next two equations must be satisfied for the tool plate to be radially pointing out.

$$X_{E_Z} = (-c_3c_4c_5 + s_3s_5)c_{67} + c_3s_4s_{67} = 0$$

$$\frac{X_{E_Y}}{X_{E_X}} = \frac{Y}{X}$$

Appendix C

Sample Calculation for Unreached Workspace

The following is a sample graphical solution for the modified design manipulator. This graphical solution is based on equal link lengths of 118", and no retraction after deployment of the arm is assumed. The unreached workspace with horizontal orientation of the tool plate is divided into three workspaces corresponding to three configurations of the arm. Each of these configurations is shown in the figures C.1, C.2 and C.3. In each of these figures, (A) represents the configuration with horizontal tool plate. (B) represents the top view of the arm configuration in figures C.1 and C.2, and a side view in figure C.3. In (C) the hatched area represents the unreached workspace with horizontal tool plate corresponding to the configuration shown in (A).

In figures C.1 and C.2 the horizontal tool plate orientation is maintained by the rotation of the yaw joint (Elbow #2) with a range of 0° to 105°. In figure C.3 the horizontal tool plate orientation is maintained by the rotation of elbow #3 pitch and the wrist pitch in equal and opposite direction.

The unreached workspaces are overlapped and the overall unreached workspace is determined in figure C.4.

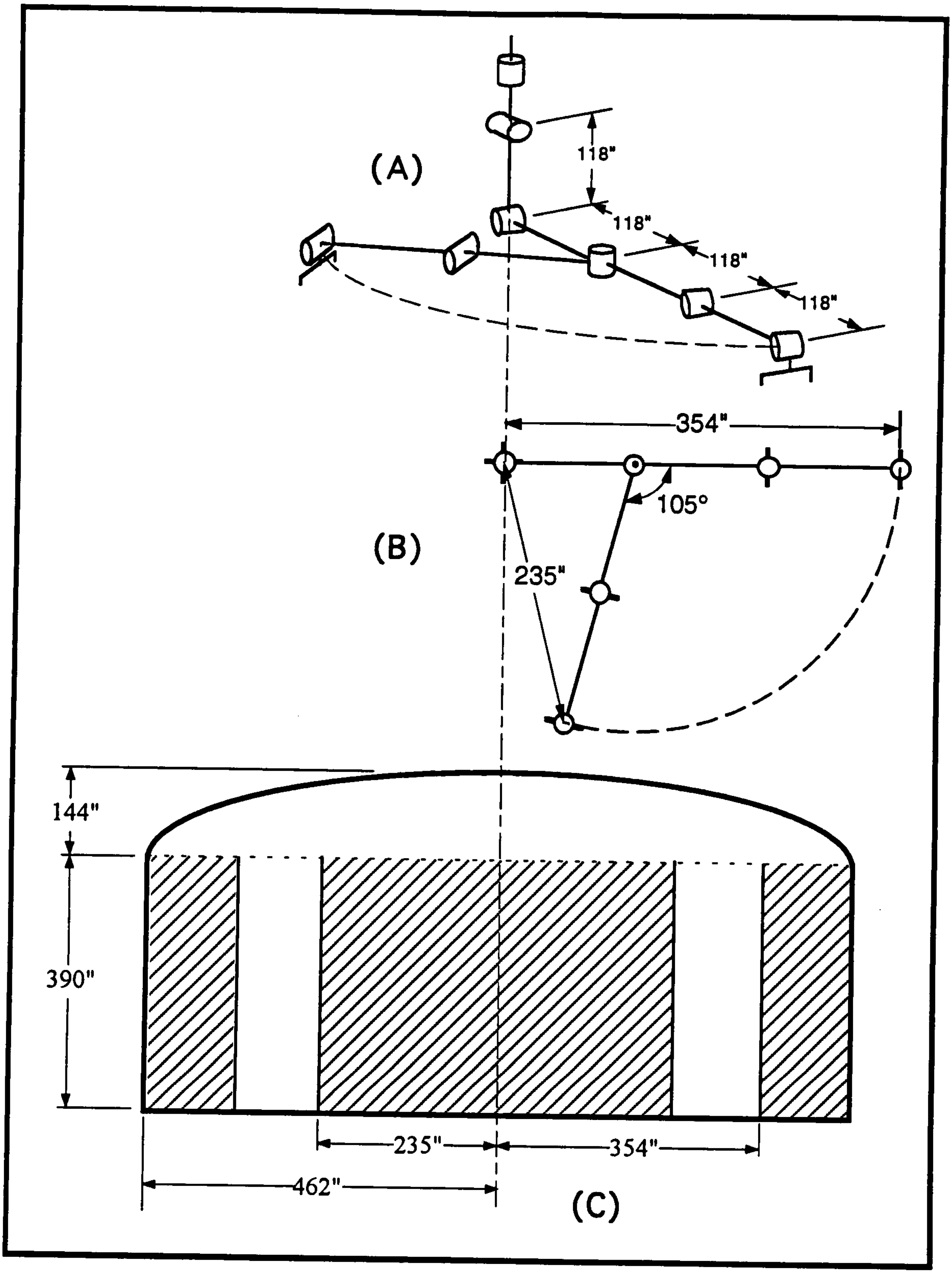


Fig. C.1. First configuration with horizontal tool plate orientation

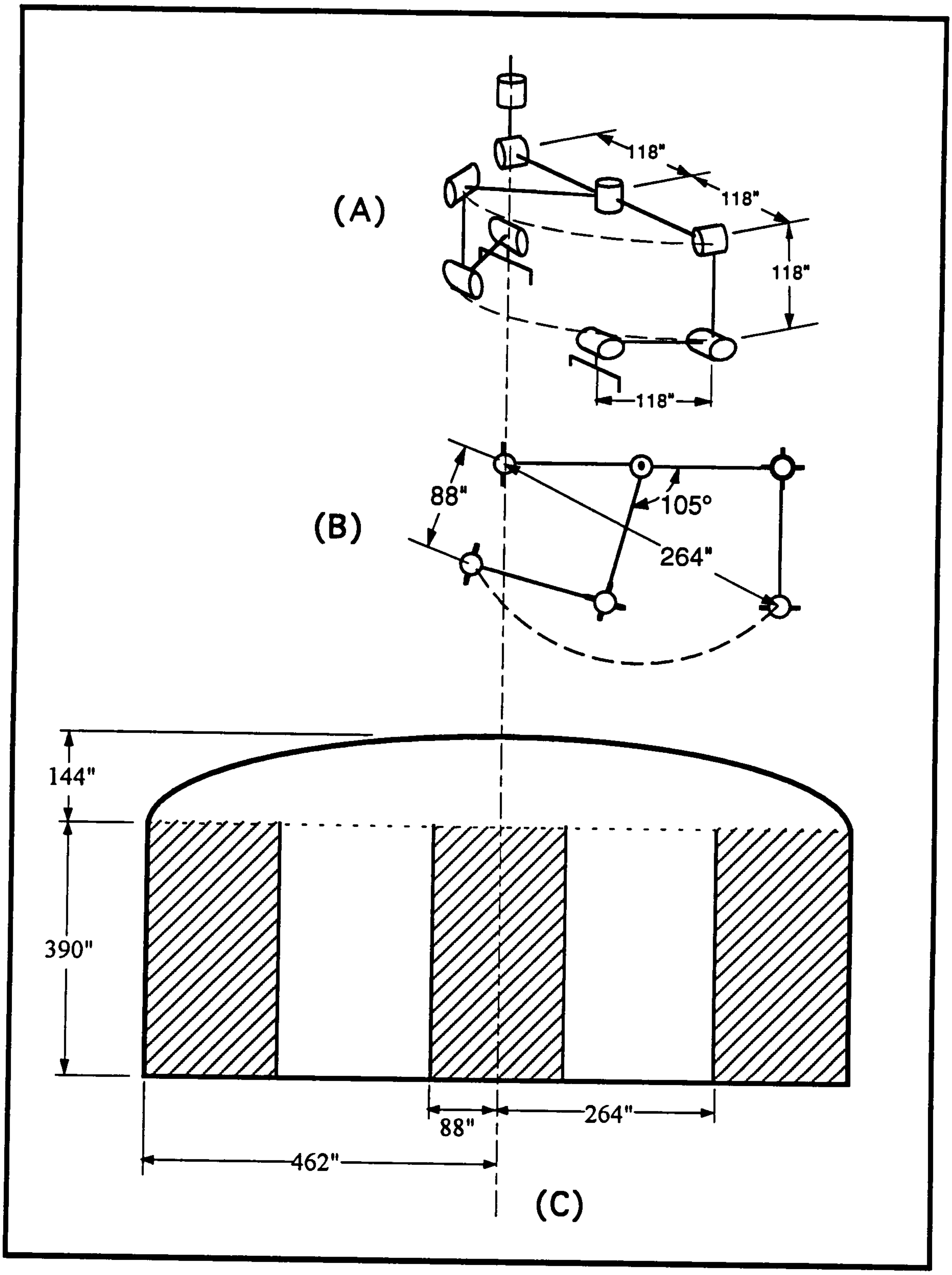


Fig. C.2. Second configuration with horizontal tool plate orientation

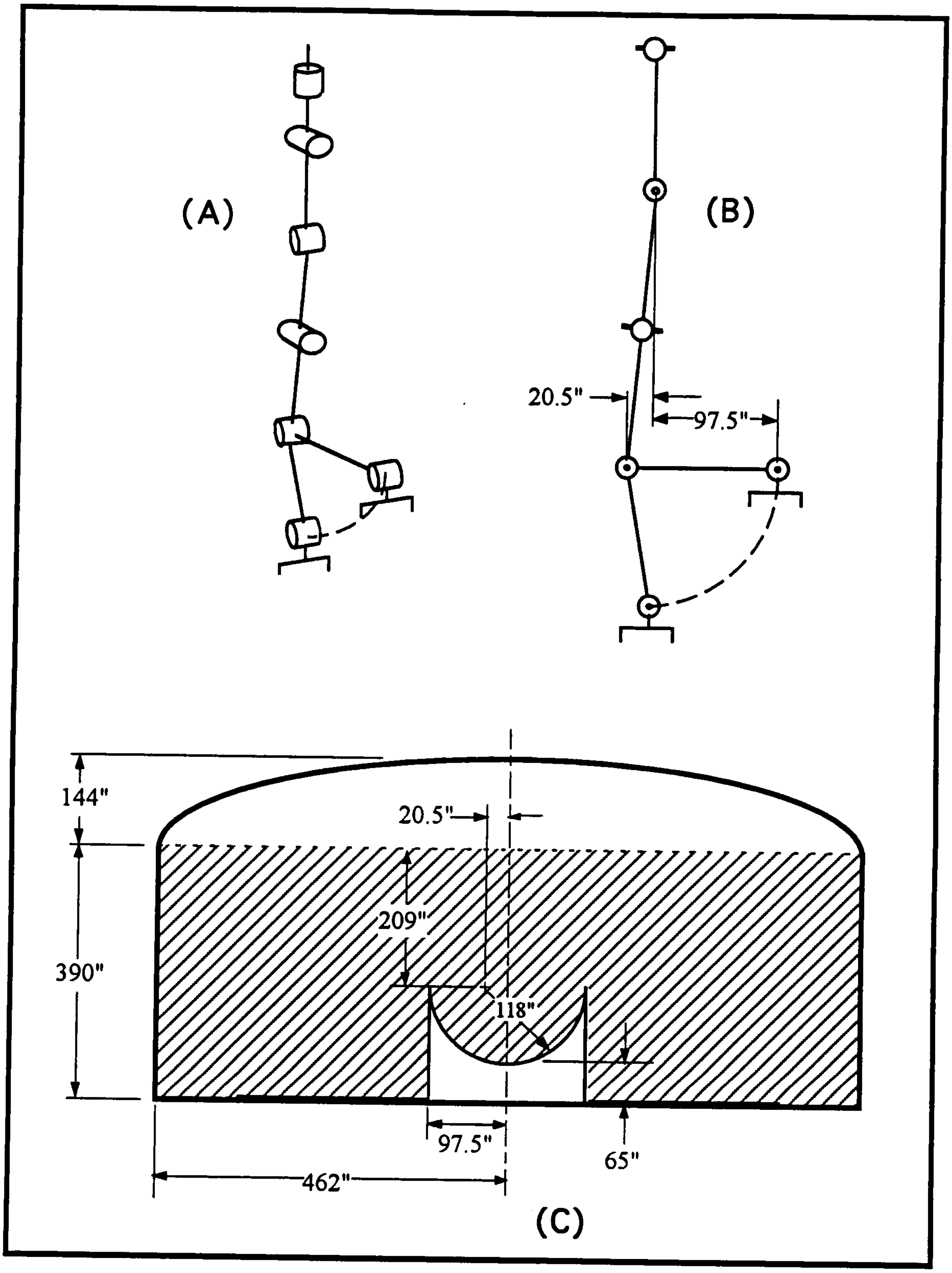


Fig. C.3. Third configuration with horizontal tool plate orientation

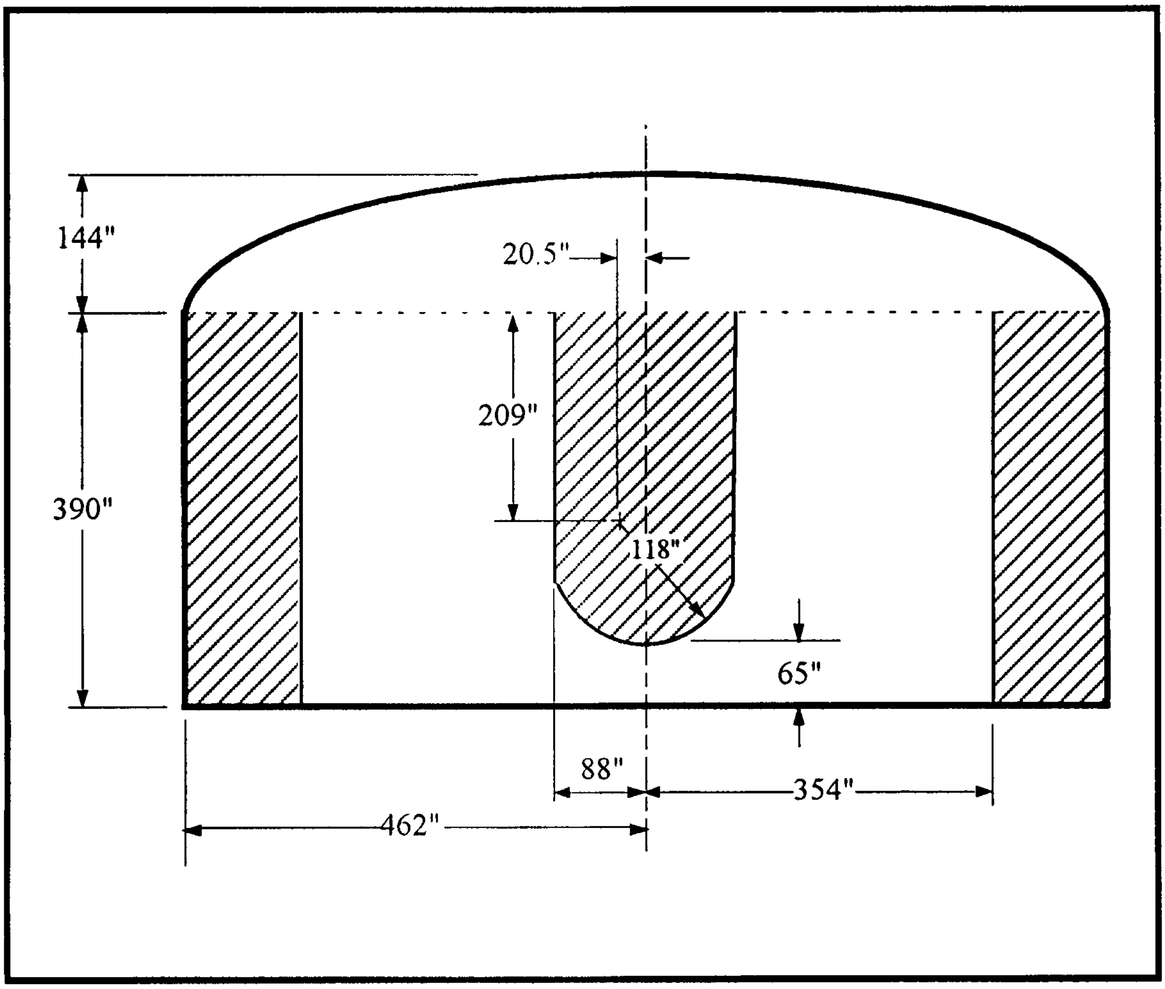


Fig. C.4. Combined unreachable region for horizontal tool plate

Appendix D

Old Testbed DH parameters

The old testbed consists of Spar RMS 2500 acting as a long reach manipulator (shown in figure D.1) and a Schilling Titan II acting as a dexterous manipulator (shown in figure D.2). For programming purposes it is necessary to calculate the transformation and Jacobian matrices. First the Spar RMS 2500 shown in figure D.1 is assigned the frames. This manipulator is unusual in that two of its links are in parallel configuration, which requires a different technique to find the DH parameters. In this technique one of the parallel links will be considered, while the other link will be replaced by a constraint equation. As an explanation, frame #3 must not change orientation with respect to the base regardless of rotation of frame #2. Therefore, if frame #2 rotates by a certain amount around the Z axis, then frame #3 must rotate the same exact amount in the opposite direction. In other words, the following equation must be satisfied during the manipulator's motion.

$$\theta_2 - \theta_3 = 0 \tag{D.1}$$

Likewise frame #5 must not change orientation with respect to base regardless of rotation of frame #4. This condition is satisfied by the following equation.

$$\theta_4 - \theta_5 = 0 \tag{D.2}$$

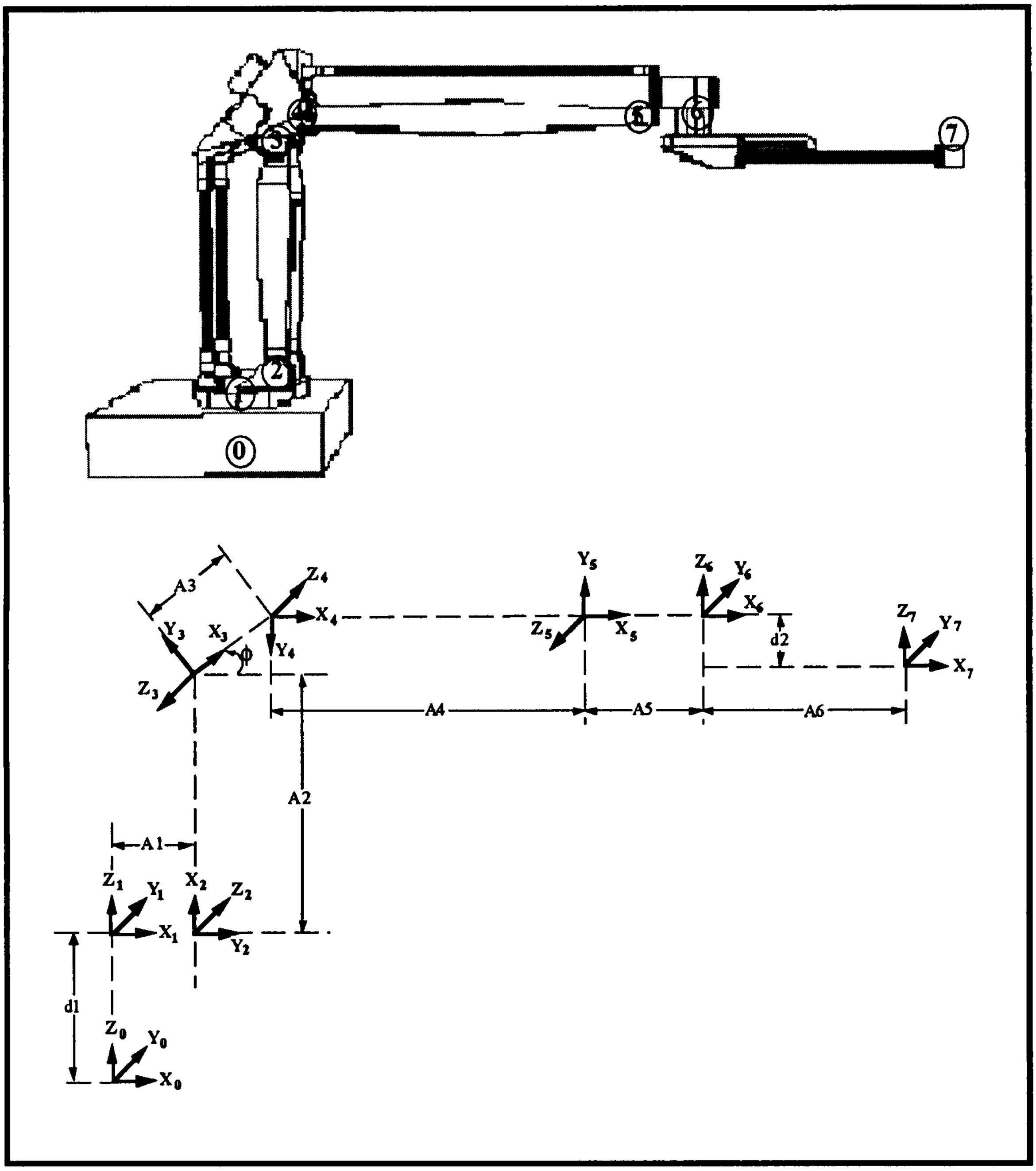


Fig. D.1. Frame assignment of the Spar manipulator in the old testbed

Table D.1 D-H table of the spar manipulator in the old testbed

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0°	d_1	(θ_1)
2	A_1	-90°	0	$(\theta_2 - 90°)$
3	A_2	180°	0	$(\theta_3 - (90° - \phi))$
4	A_3	180°	0	$(\theta_4 + \phi)$
5	A_4	180°	0	(θ_5)
6	A_5	-90°	0	(θ_6)
7	A_6	0°	$-d_2$	0

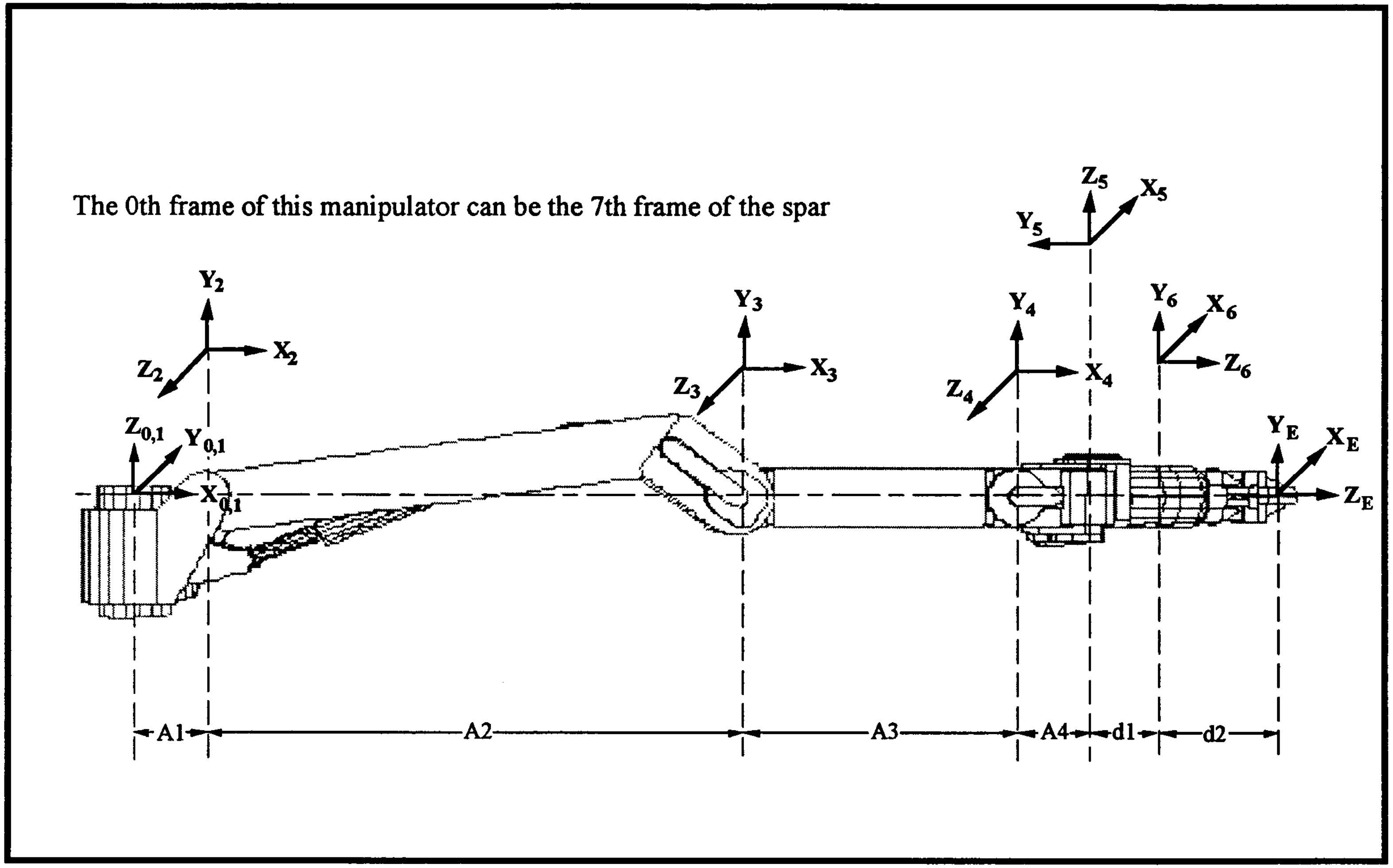


Fig. D.2. Frame assignment for the Schilling manipulator

Table D.2 D-H table of the spar manipulator in the old testbed

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	(θ_1)
2	A_1	90°	0	(θ_2)
3	A_2	0	0	(θ_3)
4	A_3	0	0	(θ_4)
5	A_4	-90°	0	$(\theta_5 + 90°)$
6	0	90°	d_1	(θ_6)
E	0	0	d_2	0

Appendix E

Old Testbed Programs

Constants Definition

```
/******************************************************************************
*                                                                             *
*       file:           testbed.h                                             *
*       programmers:    Steve Everett                                         *
*                       Amin AL-Halabi                                        *
*       date:           5/19/95                                               *
*                                                                             *
* This is the header file which defines the constants used in the set of     *
* redundancy resolution and related programs for the PNL testbed written     *
* by the above authors at the University of Tennessee.                       *
*                                                                             *
******************************************************************************/

/* Constant parameters          */

#define CMD_UPDATE_RATE 100         /* command position update rate (Hz)      */

#define UPP_JLIM_1 180          /* upper joint limits (degrees)       */
#define UPP_JLIM_2 180
#define UPP_JLIM_3 180
#define UPP_JLIM_4 180
#define UPP_JLIM_5 180
#define UPP_JLIM_6 180
#define UPP_JLIM_7 180
#define UPP_JLIM_8 180
#define UPP_JLIM_9 180

#define LOW_JLIM_1 -180                 /* lower joint limits (deg)       */
#define LOW_JLIM_2 -180
#define LOW_JLIM_3 -180
#define LOW_JLIM_4 -180
#define LOW_JLIM_5 -180
#define LOW_JLIM_6 -180
#define LOW_JLIM_7 -180
#define LOW_JLIM_8 -180
#define LOW_JLIM_9 -180


/* Error codes returned by redundancy resolution routines       */

#define J1_LIM_EXC 1            /* joint 1 limit exceeded       */
```

```
#define J2_LIM_EXC 2
#define J3_LIM_EXC 3
#define J4_LIM_EXC 4
#define J5_LIM_EXC 5
#define J6_LIM_EXC 6
#define J7_LIM_EXC 7
#define J8_LIM_EXC 8
#define J9_LIM_EXC 9

#define SINGULAR_PT 10                /* singularity encountered          */

/* Redundancy resolution schemes used in testbedCartPath.c     */

#define PSEUD_INV 1
#define NULL_EXT 2

/* Error code returned by solveLinSys       */

#define SINGULAR_EQ 1
```

Cartesian Path

```
/*******************************************************************************
*                                                                              *
*       function:       testbedCartPath                                        *
*       programmers:    Steve Everett                                          *
*                       Amin AL-Halabi                                         *
*       date:           5/25/95                                                *
*                                                                              *
*                                                                              *
* This function calculates the necessary joint values to move the end          *
* effector of the PNL testbed manipulator along a Cartesian path between its   *
* current and a goal position and orientation.  It takes as its inputs the     *
* current testbed joint values (testbedJntVals), the desired change in         *
* position and orientation (delCoordinates), and the redundancy resolution     *
* scheme (redResScheme).  Schemes available are denoted as follows:            *
*                                                                              *
*       pseudo-inverse                          PSEUD_INV                      *
*       nullspace extraction                    NULL_EXTR                      *
*                                                                              *
* It should be noted that the desired change in coordinates is expressed in    *
* the base frame and the orientation change is represented with quaternions.  *
* Also required is the transit time (time) for the maneuver.                   *
* The function creates a 9x1024 array of joint value commands (jntValCmds)     *
* where (validSteps) is the number of valid steps produced.  The function      *
* will return an error code depending on the outcome of the path plan.  A      *
* zero indicates the path plan was successful, while the following codes       *
* indicate the respective error occured:                                       *
*                                                                              *
*       joint limit 1 exceeded                  J1_LIM_EXC                     *
```

```
*               :                                           :                               *
*       joint limit 9 exceeded                          J9_LIM_EXC                          *
*       singular point encountered                      SINGULAR_PT                         *
*                                                                                           *
* If the path plan is unsuccessful, the array (jntValCmds) will contain the                 *
* path up to the point where it encountered the problem.                                    *
*                                                                                           *
*********************************************************************************************/

include <math.h>
#include "testbed.h"

int
testbedCartPath(testbedJntVals,delCoordinates,redResScheme,time,delJntVals,validSteps)

float *testbedJntVals;
float *delCoordinates;
int redResScheme;
float jntValCmds[9][1024];
int validSteps;

{

 float coordStep[7];
 int i,j;
 float testbedJacobian[7][10];
 float jntValStep[10];
 int steps2Goal;

 /* Calculate the number of steps to reach the goal in the specified time   */


 steps2Goal = time*CMD_UPDATE_RATE;          /* Note that since (steps) was      *
                                              * declared as (int), its value will *
                                              * be truncated.                   */


 /* Find the linear and angular step sizes */

 for (i=1;i<=6;i++)
  coordStep[i] = delCoordinates[i]/steps2Goal;

 /* Make the first commanded joint value equal to the current joint values  */

 for (i=1;i<=9;i++)
  jntValCmds[i][1] = testbedJntVals[i];


/*******************************************************************************************
* This is the main loop which calculates each set of joint values.  It                     *
* continues to calculate joint values as long as it has not reached the                    *
* goal, a singular point, or a joint limit.                                                *
*******************************************************************************************/


for (validSteps=1;validSteps<=steps2Goal;validSteps++)
{
```

```
/* Calculate the current testbed Jacobian                    */

calcTestbedJacob(testbedJntVals,testbedJacobian);

/*****************************************************************************
 * Calculate the joint value step according to the desired method.           *
 *****************************************************************************/

if (redResScheme = PSEUD_INV)
{

 if (pInvStep(coordStep,testbedJacobian,jntValStep)=SINGULAR_PT)
  {
   validSteps = validSteps - 1;
   return(SINGULAR_PT);
  }

}


else if (redResScheme = NULL_EXTR)
{
}


else
{
}


/* Calculate the next set of joint values and check to see if any limits  *
 * have been exceeded.                                                     */

for (k=1;k<=9;k++)
  testbedJntVals[k][validSteps+1] =
    testbedJntVals[k][validSteps] + jntValStep[k];

if ((testbedJntVals[1][validSteps] > UPP_JLIM_1) ||
    (testbedJntVals[1][validSteps] < LOW_JLIM_1)
  return(J1_LIM_EXC);

if ((testbedJntVals[2][validSteps] > UPP_JLIM_2) ||
    (testbedJntVals[2][validSteps] < LOW_JLIM_2)
  return(J2_LIM_EXC);

if ((testbedJntVals[3][validSteps] > UPP_JLIM_3) ||
    (testbedJntVals[3][validSteps] < LOW_JLIM_3)
  return(J3_LIM_EXC);

if ((testbedJntVals[4][validSteps] > UPP_JLIM_4) ||
    (testbedJntVals[4][validSteps] < LOW_JLIM_4)
  return(J4_LIM_EXC);

if ((testbedJntVals[5][validSteps] > UPP_JLIM_5) ||
    (testbedJntVals[5][validSteps] < LOW_JLIM_5)
  return(J5_LIM_EXC);
```

```
  if ((testbedJntVals[6][validSteps] > UPP_JLIM_6) ||
     (testbedJntVals[6][validSteps] < LOW_JLIM_6)
   return(J6_LIM_EXC);

  if ((testbedJntVals[7][validSteps] > UPP_JLIM_7) ||
     (testbedJntVals[7][validSteps] < LOW_JLIM_7)
   return(J7_LIM_EXC);

  if ((testbedJntVals[8][validSteps] > UPP_JLIM_8) ||
     (testbedJntVals[8][validSteps] < LOW_JLIM_8)
   return(J8_LIM_EXC);

  if ((testbedJntVals[9][validSteps] > UPP_JLIM_9) ||
     (testbedJntVals[9][validSteps] < LOW_JLIM_9)
   return(J9_LIM_EXC);

 }

}
```

Joint Step Movement

```
/*******************************************************************************
*                                                                              *
*        function:       pInvStep                                              *
*        programmers:    Steve Everett                                         *
*                        Amin AL-Halabi                                        *
*        date:           5/26/95                                               *
*                                                                              *
*                                                                              *
* This function calculates the necessary joint value step to move the end      *
* effector of the PNL testbed manipulator one Cartesian step using the         *
* pseudoinverse method of redundancy resolution. It takes as its input the     *
* desired step in position and orientation (coordStep) and creates the         *
* necessary step in joint values (jntValStep). The angular part of             *
* (coordStep) is represented with quaternions.                                 *
*                                                                              *
* Note that the pseudoinverse of the Jacobian is never actually calculated.    *
* The equation  p = J'*inv(J*J')*q is used; however, q is solved for by        *
* successive solutions of sets of linear equations. First, an intermediate     *
* vector (y) is solved in the equation (J*J')*y = p with the function          *
* (solveLinSys). Then, q may be found with q = J'*y.                           *
*                                                                              *
*******************************************************************************/

include "testbed.h"


pInvStep(coordStep,testbedJacobian,jntValStep)
```

```
float *coordStep;
float testbedJacobian[7][10];
float jntValStep[10];

{

  int i,j,k;
  float JxJtransp[7][7];
  float y[7];


  /* Calculate the product of the Jacobian and its transpose.              */

  for (i=1;i<=6;i++)
    for (j=1;j<=6;j++)
      for (k=1;k<=9;k++)
        JxJtransp[i][j] += testbedJacobian[i][k]*testbedJacobian[j][k];


  /* Solve for the intermediate vector y from [ (J*J')*y = p ]. If the set  *
   * has no solution, the function returns the error code for singularity.  */

  if (solveLinSys(JxJtransp,coordStep,y) = SINGULAR_EQ)
    return(SINGULAR_PT);


  /* Solve for the joint value step vector from [ q = J'*y ]              */

  for (i=1;i<=9;i++)
    for (j=1;j<=6;j++)
      jntValStep[i] += testbedJacobian[j][i] * y[j];

}
```

Forward Kinematics of the Testbed

```
/*****************************************************************************
*                                                                            *
*        function:        testbedForKin                                      *
*        programmers:     Steve Everett                                      *
*                         Amin AL-Halabi                                     *
*        date:            5/18/95                                            *
*                                                                            *
*                                                                            *
* This subroutine calculates the forward kinematics for the PNL testbed arm, *
* consisting of a SPAR 2500 and Schilling Titan II manipulators. It takes as *
* its inputs the nine current joint values of the entire manipulator         *
* (testbedJntVals) and creates the homogeneous transform matrix representing *
* the position and orientation of the end effector expressed in base         *
* coordinates (testbedHomoTransf).                                           *
```

```
*                                                                                    *
************************************************************************************/

void
testbedForKin(testbedJntVals,testbedHomoTransf)

float *testbedJntVals;
float testbedHomoTransf[5][5];

{

 float sparJntVals[4];
 float schilJntVals[7];
 float sparHomoTransf[5][5];
 float schilHomoTransf[5][5];
  int i,j,k,m;


 /* Partition the joint values into SPAR and Schilling joint value vectors. */

 sparJntVals[1] = testbedJntVals[1];
 sparJntVals[2] = testbedJntVals[2];
 sparJntVals[3] = testbedJntVals[3];

 for (i=1;i<=6;i++)
   schilJntVals[i] = testbedJntVals[(i+3)];


 /* Call functions to calculate homogeneous transforms for each arm.        */

 sparForKin(sparJntVals,sparHomoTransf);

 schilForKin(schilJntVals,schilHomoTransf);


 /* Multiply the transforms to get entire arm homogeneous transform.       */

 for (j=1;j<=4;j++)
  for (k=1;k<=4;k++)
    for (m=1;m<=4;m++)
    testbedHomoTransf[j][k] += sparHomoTransf[j][m]*schilHomoTransf[m][k];

}
```

Jacobian Calculation of the Testbed

```
/***********************************************************************************
*                                                                                  *
```

```
*        function:        calcTestbedJacob                                          *
*        programmers:     Steve Everett                                             *
*                         Amin AL-Halabi                                            *
*        date:            5/19/95                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
* This subroutine calculates the Jacobian for the PNL SPAR/Schilling                *
* testbed.  It takes as its inputs the nine current joint values of the             *
* entire testbed manipulator (testbedJntVals) and creates the 6x9 Jacobian          *
* matrix expressed in base coordinates (testbedJacobian).                           *
*                                                                                   *
************************************************************************************/


void
calcTestbedJacob(testbedJntVals,testbedJacobian)

float *testbedJntVals;
float testbedJacobian[7][10];

{

 float sparJntVals[4];
 float sparJacobian[7][4];
 float sparJacobPos[4][4];
 float sparJacobRot[4][4];
 float sparHomoTransf[5][5];
 float sparBase2ToolRot[4][4];

 float schilJntVals[7];
 float schilJacobian[7][7];
 float schilJacobRot[4][7];
 float schilJacobPos[4][7];
 float schilHomoTransf[5][5];
 float schilTool2EndRot[4][4];

 float tool2EndVec[4];

 float jacobUppLeft[4][4];
 float jacobUppRight[4][7];
 float jacobLowLeft[4][4];
 float jacobLowRight[4][7];

/* Declaration of auxiliary and dummy variables                    */

 int i,j,k;
float auxMat1[4][4], auxMat2[4][4];

/**********************************************************************************
* The first part of this routine caculates the elements necessary to              *
* construct the Jacobian of the entire testbed arm.                               *
**********************************************************************************/

/* Partition the joint values into SPAR and Schilling joint value vectors. */

sparJntVals[1] = testbedJntVals[1];
sparJntVals[2] = testbedJntVals[2];
```

```
sparJntVals[3] = testbedJntVals[3];

 for (i=1;i<=6;i++)
  schilJntVals[i] = testbedJntVals[(i+3)];


/* Calculate the Jacobian for SPAR and Schilling and partition each into  *
 * position and rotation parts.                                          */

calcSparJacob(sparJntVals,sparJacobian);

calcSchilJacob(schilJntVals,schilJacobian);

 for (i=1;i<=3;i++)
  for (j=1;j<=3;j++)
   sparJacobPos[i][j] = sparJacobian[i][j];

 for (i=1;i<=3;i++)
  for (j=1;j<=3;j++)
   sparJacobRot[i][j] = sparJacobian[(i+3)][j];

 for (i=1;i<=3;i++)
  for (j=1;j<=6;j++)
   schilJacobPos[i][j] = schilJacobian[i][j];

 for (i=1;i<=3;i++)
  for (j=1;j<=6;j++)
   schilJacobRot[i][j] = schilJacobian[(i+3)][j];


/* Calculate the forward kinematics of each to get the rotation matrices *
 * from base to tool plate and from tool plate to end effector           */

sparForKin(sparJntVals,sparHomoTransf);

schilForKin(schilJntVals,schilHomoTransf);

 for (i=1;i<=3;i++)
  for (j=1;j<=3;j++)
   sparBase2ToolRot[i][j] = sparHomoTransf[i][j];

 for (i=1;i<=3;i++)
  for (j=1;j<=3;j++)
   schilTool2EndRot[i][j] = schilHomoTransf[i][j];


/* Get the relative position vector from the tool plate to the end     *
 * effector expressed in tool plate coordinates from the Schilling     *
 * homogeneous transformation.                                         */

for (i=1;i<=3;i++)
 tool2EndVec[i] = schilHomoTransf[i][4];


/*************************************************************************
* The Jacobian of the entire testbed arm is constructed here.  The       *
* formula uses the elements above and may be derived using dynamic       *
```

```
 * principles. (See associated documentation for derivation.)                    *
 ******************************************************************************/


/* Construct the upper left block of the testbed Jacobian -- the part     *
 * which relates SPAR joint velocities to linear end effector velocity.  */

for (i=1;i<=3;i++)
{
 auxMat1[i][i] = 0;

 auxMat1[2][1] += sparBase2ToolRot[3][i] * tool2EndVec[i];
 auxMat1[1][2] = -(auxMat1[2][1]);

 auxMat1[1][3] += sparBase2ToolRot[2][i] * tool2EndVec[i];
 auxMat1[3][1] = -(auxMat1[1][3]);

 auxMat1[3][2] = sparBase2ToolRot[1][i] * tool2EndVec[i];
 auxMat1[2][3] = -(auxMat1[3][2]);
}

for (i=1;i<=3;i++)
 for (j=1;j<=3;j++)
  for (k=1;k<=3;k++)
   auxMat2[i][j] += auxMat1[i][k] * sparJacobRot[k][j];

for (i=1;i<=3;i++)
 for (j=1;j<=3;j++)
  jacobUppLeft[i][j] = sparJacobPos[i][j] - auxMat2[i][j];


/* Construct the upper right block of the testbed Jacobian -- the part  *
 * which relates Schilling joint velocities to linear end effector      *
 * velocity.                                                            */

for (i=1;i<=3;i++)
 for (j=1;j<=6;j++)
  for (k=1;k<=3;k++)
   jacobUppRight[i][j] += sparBase2ToolRot[i][k] * schilJacobPos[k][j];


/* Construct the lower left block of the testbed Jacobian -- the part    *
 * which relates Spar joint velocities to rotational end effector        *
 * velocity.                                                             */

for (i=1;i<=3;i++)
 for (j=1;j<=3;j++)
  jacobLowLeft[i][j] = sparJacobRot[i][j];


/* Construct the lower right block of the testbed Jacobian -- the part  *
 * which relates the Schilling joint velocities to rotational end       *
 * effector velocity.                                                   */

for (i=1;i<=3;i++)
 for (j=1;j<=6;j++)
  for (k=1;k<=3;k++)
```

```
      jacobLowRight[i][j] += sparBase2ToolRot[i][k] * sparJacobRot[k][j];

  /* Now assemble the entire testbed Jacobian.                    */

  for (i=1;i<=3;i++)
   for (j=1;j<=3;j++)
   {
    testbedJacobian[i][j] = jacobUppLeft[i][j];
    testbedJacobian[(i+3)][j] = jacobLowLeft[i][j];
   }

  for (i=1;i<=3;i++)
   for (j=1;j<=6;j++)
   {
    testbedJacobian[i][(j+3)] = jacobUppRight[i][j];
    testbedJacobian[(i+3)][(j+3)] = jacobLowRight[i][j];
   }

}
```

Transformation and Jacobian Calculation of the Spar Arm

```
/*****************************************************************************
*                                                                           *
*        function:       sparKinematics                                     *
*        programmers:    Steve Everett                                      *
*                        Amin AL-Halabi                                     *
*        date:           6/5/95                                             *
*                                                                           *
*                                                                           *
* This subroutine calculates the kinematics, including the homogeneous      *
* transform and the Jacobian for the SPAR 2500 in the PNL testbed. It takes *
* as its inputs the current joint values of the SPAR manipulator            *
* (sparJntVals) and creates the homogeneous transform matrix representing   *
* the position and orientation of the tool plate expressed in base          *
* coordinates (sparHomoTransf) and the 6x6 Jacobian relating the end        *
* effector velocity in tool plate coordinates to the joint velocities.      *
*                                                                           *
*****************************************************************************/

void
sparForKin(sparJntVals,sparHomoTransf,sparJacobian)

float *sparJntVals;
float sparHomoTransf[5][5];
float sparJacobian[7][7];

/*************** Calculate the Jacobian of spar manipulator *************/
```

```
{
    unsigned int i;
    float th, r1, r2, r3;
    float s1, s2, s3, s4, s5, s6, s7;
    float c1, c2, c3, c4, c5, c6, c7;
    float u1, u2, u3, v3;


    th=schilJntVals[1];
    s1=SIN(th);
    c1=COS(th);
    th=schilJntVals[2];
    s2=SIN(th);
    c2=COS(th);
    th=schilJntVals[3];
    s3=SIN(th);
    c3=COS(th);
    th=schilJntVals[4];
    s4=SIN(th);
    c4=COS(th);
    th=schilJntVals[5];
    s5=SIN(th);
    c5=COS(th);
    th=schilJntVals[6];
    s6=SIN(th);
    c6=COS(th);


for (i=1,i<=5,i++)
  {
   for (j=1,j<=5,j++)
    t[i][j] = sparHomoTransf[i][j];
  }

for (i=1,i<=7,i++)
  {
   for (j=1,j<=7,j++)
    jc[i][j] = sparJacobian[i][j];
  }

   jc[4][1]=jc[6][1]=0;
   jc[5][1]=1;
   jc[4][2]=c1;
   jc[5][2]=0;
   jc[6][2]= -s1;
   jc[1][3]=s1*c2*A2;
   jc[2][3]= -s2*A2;
   jc[3][3]=c1*c2*A2;
   jc[4][3]=s1*s2;
   jc[5][3]=c2;
   jc[6][3]=c1*s2;
   jc[1][4]=s1*(c2*(A3*c3+A2)+s2*D3)+c1*A3*s3;
   jc[2][4]=c2*D3-s2*(c3*A3+A2);
   jc[3][4]=c1*(c2*(A3*c3+A2)+s2*D3)-s1*A3*s3;
   jc[4][4]=c1*c3-s1*c2*s3;
   jc[5][4]=s2*s3;
```

```
jc[6][4]= -c1*c2*s3-s1*c3;
u1=s1*c2*c3+c1*s3;
u2= -s2*c3;
u3=c1*c2*c3-s1*s3;
v3= -c1*s2;
r1=u1*c4-jc[4][3]*s4;
r2=u2*c4-c2*s4;
r3=u3*c4+v3*s4;
jc[4][5]=u1*s4+jc[4][3]*c4;
jc[5][5]=u2*s4+c2*c4;
jc[6][5]=u3*s4-v3*c4;
jc[1][5]=r1*A4+jc[1][4];
jc[2][5]=r2*A4+jc[2][4];
jc[3][5]=r3*A4+jc[3][4];
u1=r1*c5+jc[4][4]*s5;
u2=r2*c5+jc[5][4]*s5;
u3=r3*c5+jc[6][4]*s5;
jc[4][6]=jc[4][4]*c5-r1*s5;
jc[5][6]=jc[5][4]*c5-r2*s5;
jc[6][6]=jc[6][4]*c5-r3*s5;
jc[1][6]=u1*A5+jc[4][5]*D5+r1*A4+jc[1][4];
jc[2][6]=u2*A5+jc[5][5]*D5+r2*A4+jc[2][4];
jc[3][6]=u3*A5+jc[6][5]*D5+r3*A4+jc[3][4];
jc[4][7]=u1*s6+jc[4][5]*c6;
jc[5][7]=u2*s6+jc[5][5]*c6;
jc[6][7]=u3*s6+jc[6][5]*c6;
r1=u1*c6-jc[4][5]*s6;
t[1][1]=r1*c7+jc[4][6]*s7;
t[1][2]=jc[4][6]*c7-r1*s7;
t[1][3]=jc[4][7];
t[1][4]=jc[4][7]*D7+A6*(u1*c6-jc[4][5]*s6)+jc[1][6];
r2=u2*c6-jc[5][5]*s6;
t[2][1]=r2*c7+jc[5][6]*s7;
t[2][2]=jc[5][6]*c7-r2*s7;
t[2][3]=jc[5][7];
t[2][4]=jc[5][7]*D7+A6*(u2*c6-jc[5][5]*s6)+jc[2][6];
r3=u3*c6-jc[6][5]*s6;
t[3][1]=r3*c7+jc[6][6]*s7;
t[3][2]=jc[6][6]*c7-r3*s7;
t[3][3]=jc[6][7];
t[3][4]=jc[6][7]*D7+A6*(u3*c6-jc[6][5]*s6)+jc[3][6];
t[4][1]=t[4][2]=t[4][3]=0;
t[4][4]=1;
jc[1][1]=t[3][4];
jc[2][1]=0;
jc[3][1]= -t[1][4];
jc[1][2]= -t[2][4]*jc[6][2];
jc[2][2]=t[1][4]*jc[6][2]-t[3][4]*jc[4][2];
jc[3][2]=t[2][4]*jc[4][2];
jc[1][7]=jc[2][7]=jc[3][7]=0;
for (i=3; i<=6; i++)
    {
    r1=t[1][4]-jc[1][i];
    r2=t[2][4]-jc[2][i];
    r3=t[3][4]-jc[3][i];
    jc[1][i]=jc[5][i]*r3-jc[6][i]*r2;
    jc[2][i]=jc[6][i]*r1-jc[4][i]*r3;
```

```
        jc[3][i]=jc[4][i]*r2-jc[5][i]*r1;
          }
    return;
}

/***** Calculate the homogeneous transformation of spar manipulator *****/

void k2107_transf(theta,t)
float *theta, t[5][5];
{
    unsigned int i;
    float th, r1, r2, r3, t1, t2, t3, w1, w2, w3, x1, x2, x3;
    float s1, s2, s3, s4, s5, s6, s7;
    float c1, c2, c3, c4, c5, c6, c7;
    float u1, u2, u3, v3, y1, y2, y3;


    th=theta[1];
    s1=SIN(th);
    c1=COS(th);
    th=theta[2];
    s2=SIN(th);
    c2=COS(th);
    th=theta[3];
    s3=SIN(th);
    c3=COS(th);
    th=theta[4];
    s4=SIN(th);
    c4=COS(th);
    th=theta[5];
    s5=SIN(th);
    c5=COS(th);
    th=theta[6];
    s6=SIN(th);
    c6=COS(th);

    th=s1*s2;
    x1=s1*(c2*(A3*c3+A2)+s2*D3)+c1*A3*s3;
    x2=c2*D3-s2*(c3*A3+A2);
    x3=c1*(c2*(A3*c3+A2)+s2*D3)-s1*A3*s3;
    t1=c1*c3-s1*c2*s3;
    t2=s2*s3;
    t3= -c1*c2*s3-s1*c3;
    u1=s1*c2*c3+c1*s3;
    u2= -s2*c3;
    u3=c1*c2*c3-s1*s3;
    v3= -c1*s2;
    r1=u1*c4-th*s4;
    r2=u2*c4-c2*s4;
    r3=u3*c4+v3*s4;
    w1=u1*s4+th*c4;
    w2=u2*s4+c2*c4;
    w3=u3*s4-v3*c4;
    u1=r1*c5+t1*s5;
    u2=r2*c5+t2*s5;
    u3=r3*c5+t3*s5;
    y1=t1*c5-r1*s5;
```

```
    y2=t2*c5-r2*s5;
    y3=t3*c5-r3*s5;
    x1=u1*A5+w1*D5+r1*A4+x1;
    x2=u2*A5+w2*D5+r2*A4+x2;
    x3=u3*A5+w3*D5+r3*A4+x3;
    r1=u1*c6-w1*s6;
    t[1][1]=r1*c7+y1*s7;
    t[1][2]=y1*c7-r1*s7;
    t[1][3]=u1*s6+w1*c6;
    t[1][4]=t[1][3]*D7+A6*(u1*c6-w1*s6)+x1;
    r2=u2*c6-w2*s6;
    t[2][1]=r2*c7+y2*s7;
    t[2][2]=y2*c7-r2*s7;
    t[2][3]=u2*s6+w2*c6;
    t[2][4]=t[2][3]*D7+A6*(u2*c6-w2*s6)+x2;
    r3=u3*c6-w3*s6;
    t[3][1]=r3*c7+y3*s7;
    t[3][2]=y3*c7-r3*s7;
    t[3][3]=u3*s6+w3*c6;
    t[3][4]=t[3][3]*D7+A6*(u3*c6-w3*s6)+x3;
    t[4][1]=t[4][2]=t[4][3]=0;
}
```

Transformation and Jacobian Calculation of the Schilling Arm

```
/*****************************************************************************
*                                                                            *
*       function:       schilKinematics                                      *
*       programmers:    Steve Everett                                        *
*                       Amin AL-Halabi                                       *
*       date:           5/18/95                                              *
*                                                                            *
*                                                                            *
* This subroutine calculates the forward kinematics for the Schilling        *
* Titan II in the PNL testbed.  It takes as its inputs the current joint     *
* values of the Schilling manipulator (schilJntVals). It creates the         *
* homogeneous transform matrix representing the position and orientation of  *
* the end effector expressed in tool plate coordinates (schilHomoTransf) and *
* the 6x6 Jacobian relating the end effector velocity in tool plate          *
* coordinates to the joint velocities.                                       *
*                                                                            *
*****************************************************************************/

void
schilForKin(schilJntVals,schilHomoTransf,schilJacobian)

float *schilJntVals;
float schilHomoTransf[5][5];
float schilJacobian[7][7];
```

Solving a Linear System

```
/************************************************************************************
*                                                                                   *
*       function:       solveLinSys                                                 *
*       programmers:    Steve Everett                                               *
*                       Amin AL-Halabi                                              *
*       date:           5/26/95                                                     *
*                                                                                   *
*                                                                                   *
* This function solves the linear system Ax=b of (SIZE) equations. It takes        *
* as its inputs the matrix A (origA) and the constant vector (b). It               *
* It creates the solution vector (x). If the system has no solution, ie,           *
* matrix A is singular, the function returns the error SINGULAR_EQ.                *
* Otherwise, a zero is returned.                                                   *
*                                                                                   *
* The algorithm used is LU decomposition with back substitution. The               *
* routine used is from "Numerical Recipes" by Press, Flannery, Teukolsky,          *
* and Vettering, p.36ff.                                                           *
*                                                                                   *
************************************************************************************/

#define SIZE 3
#define TINY 1e-20

#include "testbed.h"
#include <math.h>

solveLinSys(origA,b,x)

float origA[SIZE+1][SIZE+1];
float b[SIZE+1];
float x[SIZE+1];

{
 float matA[SIZE+1][SIZE+1];
 int d;
 int ii,i,j,k,ll;
 float aamax;
 float vv[SIZE+1];
 float sum;
 int imax;
 float dum;
 int indx[SIZE+1];

 /*******************************************************************************
  * Copy the original matrix (origA) and the original constant vector (b)       *
  * into other arrays (matA and x) since the originals will be destroyed.       *
  *******************************************************************************/

 for (i=1;i<=SIZE;i++)
```

```
{
 x[i] = b[i];
 for (j=1;j<=SIZE;j++)
 {
  matA[i][j] = origA[i][j];
 }
}

/***************************************************************************
 * Do the LU decomposition.                                                *
 ***************************************************************************/

d = 1;

for (i=1;i<=SIZE;i++)
{

 aamax = 0;

 for (j=1;j<=SIZE;j++)
  if (fabs(matA[i][j]) > aamax)
    aamax = fabs(matA[i][j]);

 if (!aamax)
  return(SINGULAR_EQ);

 vv[i] = 1/aamax;

}

for (j=1;j<=SIZE;j++)
{

 if (j>1)
   for (i=1;i<=j-1;i++)
   {

     sum = matA[i][j];
     if (i>1)
     {

       for (k=1;k<=i-1;k++)
         sum = sum - matA[i][k]*matA[k][j];

       matA[i][j] = sum;

     }

   }

 aamax = 0;
 for (i=j;i<=SIZE;i++)
 {

  sum = matA[i][j];
  if (j>1)
  {
```

```
      for (k=1;k<=j-1;k++)
       sum = sum - matA[i][k] * matA[k][j];

      matA[i][j] = sum;

    }

    dum = vv[i] * fabs(sum);
    if (dum >= aamax)
    {

      imax = i;
      aamax = dum;

    }

  }

  if (j!=imax)
  {

    for (k=1;k<=SIZE;k++)
    {

     dum = matA[imax][k];
     matA[imax][k] = matA[j][k];
     matA[j][k] = dum;

    }

   d = -d;
   vv[imax] = vv[j];

  }

  indx[j] = imax;
  if (j!=SIZE)
  {

   if (fabs(matA[j][j])<TINY)
     return(SINGULAR_EQ);

   dum = 1/matA[j][j];

   for (i=j+1;i<=SIZE;i++)
     matA[i][j] = matA[i][j] * dum;

  }

}

/*************************************************************************
 * Now do the back substitution.                                         *
 *************************************************************************/
```

```
    ii=0;
    for (i=1;i<=SIZE;i++)
    {

     ll = indx[i];
     sum = x[ll];
     x[ll] = x[i];

     if (ii)
     {

        for (j=ii;j<=i-1;j++)
        sum = sum - matA[i][j] * x[j];
     }

     else if (sum)
       ii = i;
     x[i] = sum;
    }

   for (i=SIZE;i>=1;i--)
   {

     sum = x[i];

     if (i<SIZE)
     {

       for (j=i+1;j<=SIZE;j++)
         sum = sum - matA[i][j] * x[j];
     }

    x[i] = sum / matA[i][i];
   }

return(0);

}
```

Calculating an Inverse of a Matrix

```
/*************************************************************************************
*                                                                                   *
*       function:       invMat                                                      *
*       programmers:    Steve Everett                                               *
*                       Amin AL-Halabi                                              *
*       date:           5/25/95                                                     *
```

```
 *                                                                              *
 *                                                                              *
 * This function calculates the inverse of a (SIZE)x(SIZE)matrix. It takes as   *
 * its input the matrix to be inverted (matA) and creates the inverse of this   *
 * matrix (invA).  If the matrix is singular and thus has no solution, the      *
 * function will return the error code SINGULARITY. Otherwise, a zero is        *
 * returned.                                                                    *
 *                                                                              *
 * The inversion routine used is a simplified version of the Gauss-Jordan      *
 * elimination algorithm found in "Numerical Recipes" by Press, Flannery,      *
 * Teukolsky, and Vettering, p.28.                                              *
 *                                                                              *
*********************************************************************************/

#define SIZE 3

#include "testbed.h"

invMat(matA,invA)

float matA[SIZE+1][SIZE+1];
float invA[SIZE+1][SIZE+1];

{

  int i,j,k,l,ll;
  int ipiv[SIZE+1];
  float big;
  int irow, icol;
  float dum;
  int indxr[SIZE+1], indxc[SIZE+1];
  float pivinv;


 /* Transfer the original matrix into another array, since the original     *
  * will be destroyed.                                                       */

 for (i=1;i<=SIZE;i++)
   for (j=1;j<=SIZE;j++)
     invA[i][j] = matA[i][j];


/* Begin the Gauss-Jordan routine                                  */

 for (i=1;i<=SIZE;i++)
   ipiv[i] = 0;

 for (i=1;i<=SIZE;i++)
 {
  big = 0;

  for (j=1;j<=SIZE;j++)
  {

    if (ipiv[j] != 1)
    {
```

```
      for (k=1;k<=SIZE;k++)
      {

       if (!ipiv[k])
       {

        if (fabs(invA[j][k])>=big)
        {
         big = fabs(invA[j][k]);
          irow=j;
          icol=k;
        }

       }

      else if (ipiv[k] > 1)
       return(SINGULARITY);

     }

   }

  }

  ipiv[icol] = ipiv[icol] + 1;

 if (irow != icol)
 {

   for (l=1;l<=SIZE;l++)
   {

    dum = invA[irow][l];
    invA[irow][l] = invA[icol][l];
    invA[icol][l] = dum;

   }

 }

 indxr[i] = irow;
 indxc[i] = icol;

if (!invA[icol][irow])
  return (SINGULARITY);

pivinv = 1/invA[icol][irow];
invA[icol][icol] = 1;

for (l=1;l<=SIZE;l++)
  invA[icol][l] = invA[icol][l] * pivinv;

for (ll=1;ll<=SIZE;ll++)
{

  if (ll!=icol)
  {
```

```
          dum = invA[ll][icol];
          invA[ll][icol] = 0;

          for (l=1;l<=SIZE;l++)
           invA[ll][l] = invA[ll][l] - invA[icol][l]*dum;

        }

      }

    }

    for (l=SIZE;l>=1;l--)
    {

     if (indxr[l]!=indxc[l])
     {

       for (k=1;k<=SIZE;k++)
       {

        dum = invA[k][indxr[l]];
        invA[k][indxr[l]] = invA[k][indxc[l]];
        invA[k][indxc[l]];

       }

     }

    }

}
```

Vita

Mr. Amin AL-Halabi was born on January 26, 1970 in Amman, Jordan. He completed his primary and secondary studies at AL-Ashrafi'ieh School, and high school studies at Salah EL-Deen High School. He joined the engineering program at the University of Tennessee, Knoxville in March 1988. During his studies in engineering he joined the co-op engineering program through which he worked for TRW-Koyo for a total of one year. He received his Bachelor of Engineering degree in Mechanical Engineering in May 1992. He was awarded an assistantship for Graduate studies in the Mechanical and Aerospace Engineering Department at the University of Tennessee, Knoxville. After graduation Mr. Amin is planning on starting a Business in his country Jordan.