Design and Implementation Issues for Stewart Platform Configuration Machine Tools

by

Philip J. Houdek II

B.S., Mechanical Engineering
Boston University, 1990

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Master of Science

at the

Massachusetts Institute of Technology

June 1997

Signature of Author ....
Department of Mechanical Engineering
May 8, 1997

Certified by ....
Sanjay Sarma
Assistant Professor of Mechanical Engineering
Thesis Supervisor

Certified by ....
Alexander H. Slocum
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by ....
Ain A. Sonin
Chairman, Department Committee on Graduate Students

Design and Implementation Issues for Stewart Platform Configuration Machine Tools

by

Philip J. Houdek II

Submitted to the Department of Mechanical Engineering
on May 8, 1997 in partial fulfillment of the requirements for the

Degree of Master of Science

ABSTRACT

The primary goal of this research was to explore the design and implementation of a parallel robotic mechanism used as the structure of a machine tool. The research was based on MIT's prototype commercially developed, Stewart Platform five-axis milling machine. A numerical control process planning system was developed and used to drive the machine. Methods and techniques for simulation, operation and user interaction were also investigated. Hardware, and software design, as well as use and control issues were examined. In addition, methods to evaluate new stewart platform designs based on function and task were developed.

Thesis Supervisor: Sanjay Sarma
Title: Assistant Professor of Mechanical Engineering

Thesis Supervisor: Alexander Slocum
Title: Alex & Brit d'Arbeloff Associate Professor of Mechanical Engineering

ACKNOWLEDGEMENTS

Thank you to my parents, who taught me to excel, when all I wanted to do was rebel.
They did a great job.

To my Father for his name and showing me how to walk tall and attack the day.

To my Mother, for unending love and support.

To Professor Alex Slocum, who put his head holder out and gave me a chance to shine.

To Professor Sanjay Sarma, who knows how to macro manage and still get results.

To Professors, Gutowski, Spearing, and Otto, for their support and guidance.

To Michelle, for love, help with the big 5 0, and pushing in the right direction.

1.0 Introduction

1.1 Overview

Conventional serial machine tool mechanisms use either linear or rotary movement stages. Each stage provides a degreee of freedom to the end effector of the machine. The stages are usually stacked in serial fashion, forming an open structural loop. In contrast, a Stewart Platform consists of a rigid platform supported by six variable length struts. Every set of six strut lengths defines a unique, fully constrainted position of the platform. Using the strut lengths as controlling input, the position and orientation of the end effector can be controlled as output.

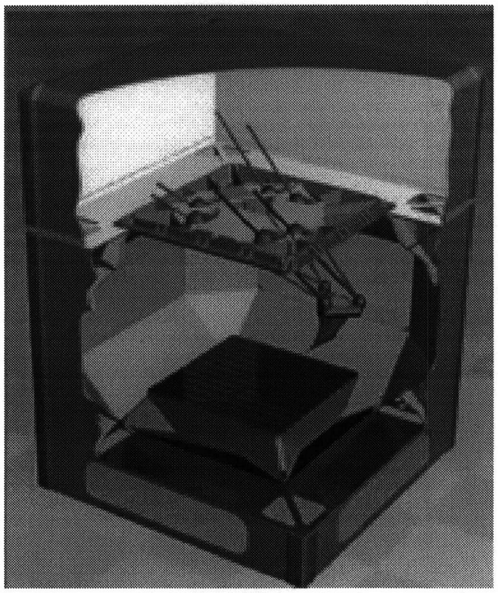

Figure 1: MIT Hexel Hexapod

The state of machine tool technology is driven by finanical and engineering concerns. This research takes initial steps towards the successful application of the Stewart Platform mechanism to the production environment as a milling machine. Our recommendations are based on several hundred man-hours spent in the installation, debugging, operation and reengineering of MIT's new Hexapod configuration machine tool. This machine tool was purchased by MIT in April of 1996 from the

Hexel Corporation, a start up company. [1] A machine tool that uses the Stewart Platform mechanism for positioning is called a Hexapod. Designing and implementing a Hexapod configuration milling machine requires considerations and solutions to issues which in some cases, are trivial in conventional serial configuration milling machines. These issues were explored in this research and several suggested solutions are purposed.

1.2 Background

The first Stewart Platform was designed and built to test tires at the Performance and Stressing Department of the New Tyre Design and Development Division, Dunlop Rubber Co., Birmingham England. [2] An article by V. Gough has photographs showing the classic Stewart Platform structure and notes the system was installed in 1954, over a decade before Stewart published his article. There is no indication the authors realized that their design might have applications beyond testing tires. Stewart proposed the use of the platform for flight simulators, machine tools and universial milling machines in 1965 [3].

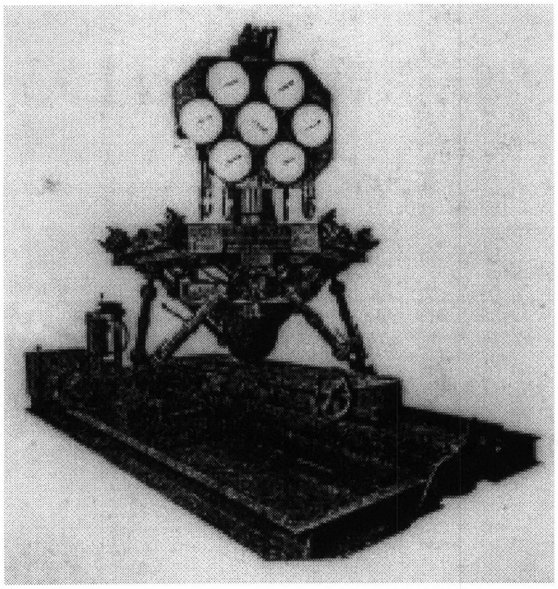

Figure 2: First Hexapod by V. Gough

It described basic operating principles of the parallel structure configuration, at which point the structure was named after Stewart. Until recently, the calculations required for real time control of the mechanism were too lengthy to be performed on anything but high powered, expensive computers. With the relatively low cost of computational power today, machine tool applications can now be controlled cost effectively. Today several companies are working towards production machine tools using the Stewart Platform structure. [4]

1.3 Organization

Production machine tools today are generally computer controlled. Computer Numerical Control (CNC) machines have strong advantages and disadvantages based on a variety of factors including structure, speed, and range of motion limits. Our research contrasts the operational and functional design requirements of conventional CNC and Hexapod CNC machines. It reviews the general process by which conventional CNC machines are used today and examines the functional requirements for a Hexapod. Observations based on extensive case studies, from design through to part fabrication, are used to suggest tools for toolpath generation, machine simulation, operator interaction, and future design evaluation.

1.4 Hexapod Configuration Milling Machine Goals

As steps are taken to bring Hexapod configuration milling machines from the drawing board to the shop floor, there are many potential benefits. Current spindle technology for material removal allows very high rotational speeds and power ratings. This enables material to be removed very quickly. Unfortunately, the performance envelope for conventional machine structures limit the application of higher performance spindles. Hexapod machine tools can be designed to move swiftly enough to utilize the full capabilities of todays high speed spindles.

Conventional machine tools which have cantilever or gantry structural configurations have inherent speed bottlenecks. They have large moving masses which must be accelerated. The need for very powerful drive systems, more precise control systems, and large structural members for stiffness, cause a spiral of design issues which are progressively more time consuming and expensive to solve. The motors and bearing produce large amounts of heat that must be shielded. Thermal shielding requires space in areas where space is at a premium. As always, the greater the level of performance required, the greater the cost in time and money. Designing a machine based on a Hexapod mechanism can minimize the standard design puzzles.

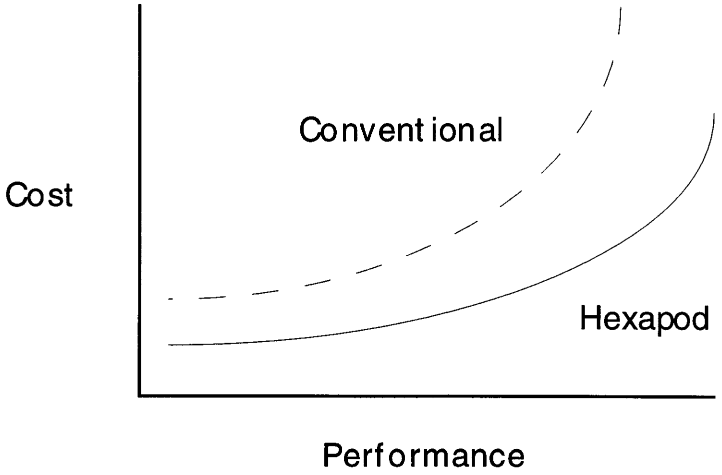

Figure 3: Relative Cost-Performance Map

First, a Hexapod structure inherently minimizes structural moving mass. Hexapod configuration milling machines move only the end effector platform. The workpiece and machine bed stay stationary. Higher accelerations and speeds are achievable. This is the case for rotation and pivot moves, as well as linear moves. Since the structure always coordinates the movement of the six legs to achieve a desired orientation or path, it has no movement bias in the directions standard to conventional machine tool stages. Applying a human term, the Hexapod can have a greater amount of dexterity than a conventional five axis machine tool.

Second, a general advantage of parallel mechanisms is that higher machine accuracy can be expected because relative errors in the actuator positions are averaged, instead of added. This minimizes the total effect of the errors on the final position of the end effector.

Third, Hexapod machine tools have a much lower cost/performance index. The structure requires fewer static and dynamic parts, which can translate into reduced construction time and more performance for the dollar. Operating costs can also be lower, due to lower energy consumption and faster material removal. Finally, the structure of Hexapods is ideally suited to modular design, which could ease service and maintanence costs.

1.5 Application

Hexapods have several potential applications. One is its use in a flexible manufacturing environment. Such an application would utilize the speed of the Hexapod mechanism, which could be combined with high speed spindles. Since Hexapods are really positioning robots, it is possible to use several interchangable head units to perform a variety of operations, such as milling, welding, cutting and assembly.

2.0 Computer Numerical Control Machine Tool Concepts

2.1 Overview

A numerically controlled machine is positioned automatically along a preprogramed path by means of coded instructions. Today, the coded instructions for all but the most simple shapes are generated using computer aided design (CAD) data and a computer aided manufacturing (CAM) software package. In addition, state of the art software packages allow changes in the design geometry to be automatically reflected in the tool path. This ability is called associativity. The tool paths are generated with respect to the design geometry, and as the geometry changes, the tool paths need only be recomputed with respect to their new set of references. In production, this allows changes rather late in the design cycle without incurring large time delays and high costs.

The CAM software is used to create an output file called the CL Data File. This cutter location file is generic to any milling machine with the same number of controllable axes. It consists of point to point moves, recorded in a standard format. Each point is given as a position with a tool axis vector. Tool selection and speed control information are also included in the file.

The CL Data File is then processed for a specific NC machine. It is processed with a conversion program which is called a Post Processor. The Post Processor takes a set of parameters about the machine as input, as well as the CL Data File to be processed. The Post Processor has the ability to output the correct syntax for the specified machine tool and to compute the transformations for the stages of the machine.

The Post Processor outputs a file in a format called G-Code. G-Code is relatively standard in the machine tool industry. The Post Processor allows for specific variations in the implementation of a machine tool controller, as well as variation in machine capabilities. The G-Code file is then downloaded to the machine tool's controller. The controller interprets the G-Code in real time and converts it into the motions to follow the required path . [5]

2.2 Broad Objectives for Numerical Control

There are many objectives which should be obtainable when implementing numerical controlled production. These objectives would be in contrast to manually controlled material removal processes.

- precise feed rate variation control
- accuracy in contouring
- following path data from CAD/CAM systems
- ability to complete work which would be impossible or impractical manually
- faster setup times
- reduction in parts handling
- flexibility that allows changes in design
- increased accuracy of duplicate parts
- better quality control
- reduced labor costs
- increased production
- more economical production
- NC machines can be programmed to replace several machines

There are several disadvantages which should also be noted. A careful analysis of the tasks to be performed and the return on investment of implementing NC manufacturing must be taken into consideration.

- high initial investment
- higher operation costs per hour than manual machine tools
- increased electrical and mechanical maintainance
- retraining of existing personnel

2.3 Numerical Control Production Advantages

Production advantages can be realized for parts with specific characteristics. The use of NC machines for production should be considered for parts which have the following specifications.

- small tolerances
- processed in small lots
- exact duplicates needed
- processed with long setup times
- straight cut milling, drilling
- non-uniform cutting speeds
- complex contours determined mathmatically

2.4 Part Geometry Design

CAD/CAM systems are used by most industries to help them survive in an atmosphere of increased global competition and ever shrinking product design cycles. These software systems can allow designers and engineers to quickly and unambiguously define their designs. This enhances communication, and allows toolpath generation which can not be performed by hand.

Systems can model the shape of the part with a high level of mathmeatical precision. The relationships between parts and other assemblies can be quickly evaluated. Specific three dimensional measurements and interference checking can also be preformed. State of the art computer aided design systems can be used as virtual prototyping tools to almost completely evaluate a design's form and function before it is manufactured for the first time. This can greatly decrease time to market and reduce development costs.

The standard output of a CAD system is generally two dimensional engineering drawings and a complete three dimensional model of a part. Companies are quickly moving to a paperless environment which eliminates the need for all but the most important drawings which describe the processes that will be used to produce the part. This means that the real output of the CAD system is a complete representation of the design part in electronic format. From this point, the CAD system may have the ability to generate the toolpath needed to drive the NC milling machine tool. If it can produce tool paths then no conversion or translation process is needed. Though in some cases, a specialized CAM software system is utilized. This requires the user to export the three dimensional geometry model from the system that created it, and import it into the CAM system so that it can be used to generate the tool paths. Note that if two different systems are used, there is no link between the geometry and the toolpath generation process. A design may change in the CAD system, but the CAM system will not know a change has occurred. In essence, if a change in the design occurs, the design must be changed in the CAD system, then transfered to the CAM system, where the toolpath generation must be performed again from scratch. This fact will generally force toolpath generation to be done at the late stages of the design process. This has a tendency to reduce manufacturing input in the design process. It also slows the process of prototyping and evaluating the production manufacturing steps.

2.5 Process Planning

This discussion of process planning assumes the part material and the quantity desired make machining practical. We further assume that the part is best suited for one or many milling operations and that it requires either three or five axes of motion to create it. The choice of axes is normally balanced with the desire to minimize the number of times the part must be fixtured, the cost of machine time, and specific traits of the design geometry.

Before toolpaths are generated, the part to be fabricated is examined to determine the manufacturing processes needed to create it. The form of the object is the first information to be evaluated. Angles, contours and shapes are considered and the number of degrees or freedom the machine must have is established.

Several parameters are used in parallel to arrive at a strategy for producing a part. Each piece of information plays a role in evaluating various options. The following set of general questions are normally taken into account as the process plan is developed. The questions are in two groups. The first group investigates the part and its geometry. The second group is more enterprise wide, covering the machine, time and resource planning of production.

Part and Machine

- size of the part
- size of the machine work volume
- angle limits of the tool
- fixturing method
- number of fixture setups
- tools used
- tool wear per operation
- operation order

Enterprise Wide

- which machine is available
- how long is the machine available
- how many parts are needed
- how fast is the machine
- how fast are tool change operations

3.0 MIT Hexapod Research Platform

3.1 Machine Description

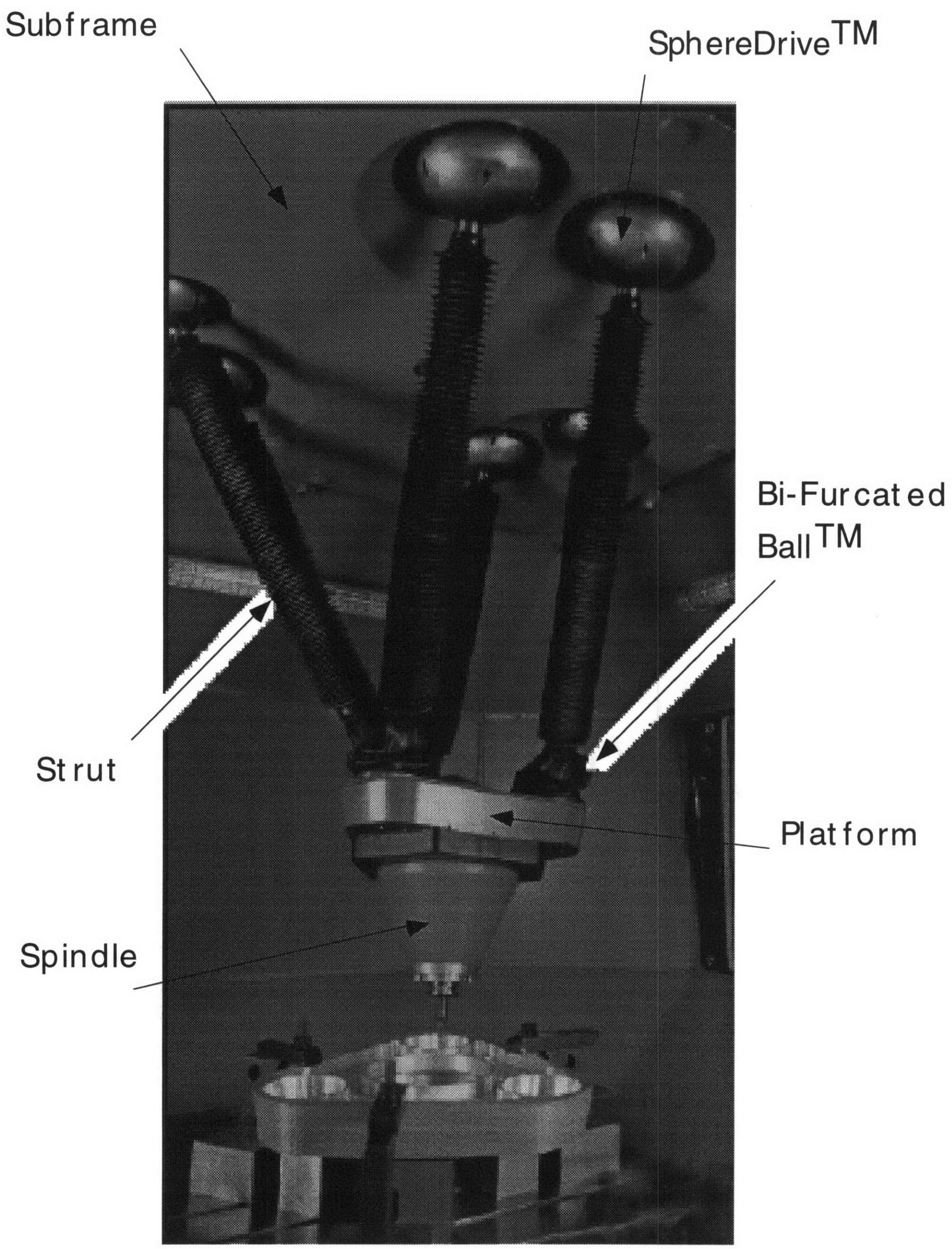

Figure 4: MIT Hexel Hexapod Five Axis Milling Machine

The MIT Hexapod was the first machine produced by the Hexel corporation of Portsmouth NH [1]. It was purchased to allow MIT to investigate a wide variety of research areas, involving not only Hexapod design and implementation, but also new methods for part fixturing, tool path generation and machine control. The mechanical, electrical and control hardware, as well as the control software were in the prototype stage when delivered. Many repairs, updates and modifications were completed in the first year of work, through a team effort between MIT and Hexel.

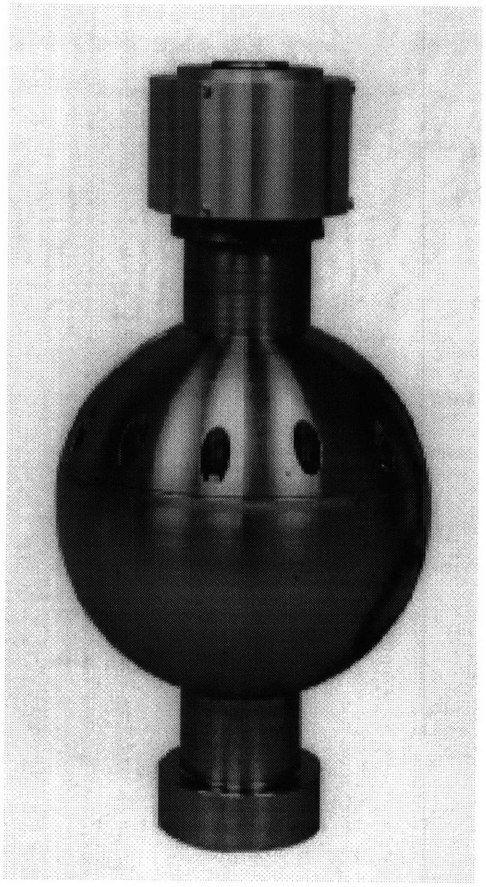

Figure 5: SphereDrive^{TM} [4]

The general machine layout is a hollow box. The top of the box is a casting called the subframe. It is the fixed ground reference. The subframe is cast with recesses around each of the openings for the six ball and socket joints. The recesses provide clearance for the struts above and below the subframe as they pivot during operation. The upper joints are called SphereDrives^{TM} [1]. They act as the ball for the joint and encase the brushless AC motor which drives the nut of the ball screw strut. The design is a very compact and elegant solution combining the mechanical joint and drive system. The six struts are hollow ball screws which actuate the platform. The struts are driven through the center of the SphereDrive^{TM}. The SphereDrives^{TM} are mounted in the subframe with a three pin kinematic coupling. Annular ring bearings fix the rotation point acting as the socket for the joint. The rings are preloaded to minimize backlash at the interface.

The six struts connect the subframe to three nodes at the movable platform. Two struts connect to each platform joint by means of a Bi-Furcated Ball^{TM} [1] joint. The joint is a ferrous sphere divided into two sections. The sections can rotate in one plane with respect to one another. This allows the platform to float freely as the strut length changes. The joint retains the ball by means of rare earth neo dymium iron boron magnets in the socket. The cup of the joint attracts the ball with approximately 250 pounds of force. This allows the cup to cover less than half the surface of the ball, creating a larger maximum achievable pivot angle. In addition, damping is provided by the shear lubricant at the ball and socket interface.

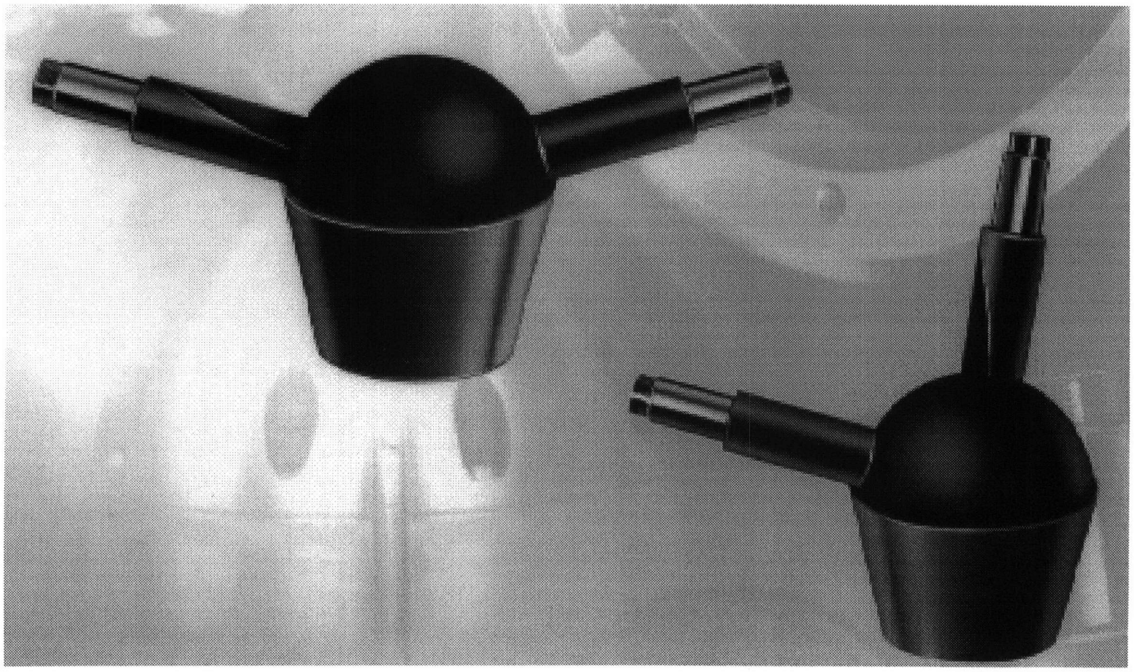

Figure 6: Bi-Furcated Ball^{TM} Joint (removed from platform) [4]

The machine is constructed of a welded plate steel frame in an open box configuration. Its cross members are hollow. In each of the cube cross members, a Shear Tube^{TM}[6] constrained layer damper is installed and potted-in-place with cement grout. The Shear Tube^{TM} is a rectangular mass wrapped in viscoelastic material. The relative motion of the Shear Tube^{TM} mass with respect to the rest of the machine frame, disipates vibrational energy in the system.

The machine is controlled with a Pentium computer and software written by Hexel. This software controller takes G-Code which has been Post Processed for the Hexapod, and drives the machine. All machine control, including manual machine motion, is performed with this software program called the AMC^{TM} [1].

3.2 Software Calibration

The machine is manufactured with tolerances which are rough in nature. The components are produced and assembled with only general care in regard to accuracy. The philosophy is to use a coordinate measuring machine on the subsystems and components of the machine to gather calibration data. The data is then input into a software database and used by the controlling software to compensate for deviations from nominal design values, a type of global error mapping. The goal of this method is to reduce cost. Speed increases in manufacturing, assembly, calibration and finishing can be achieved. Software calibration also allows interchangability of non-matching components. Calibration data on replacement components and subsystems would be used in the software database to update the system.

4.0 Hexapod Design and Implementation Issues

4.1 Industry Standard Kinematic Designations

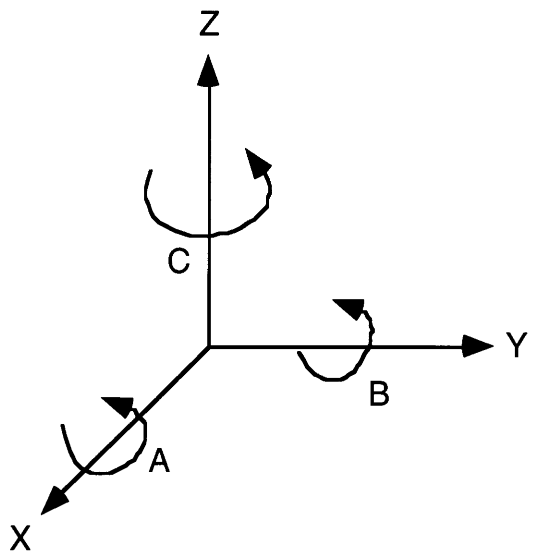

Figure 7: Direction Conventions

Industry standard machine motion designations are defined with cartesian coordinates. The Z axis extends in the positive direction normal to, and away from the tool table. The tool axis of the machine at its home position is in the negative Z direction. The rotational motion of the machine end effector is defined by the angles A, B and C. They are positive rotations about the X, Y and Z axes respectively, as defined by the right hand rule.

4.2 Hexapod Structure Definition

Stewart platform manipulators are classified by their leg configuration. A two number designation is commonly given. The first number indicates the number of nodes or joints at the machine's connection to the ground frame of reference. The second number indicates the number of nodes at the movable platform. The MIT Hexapod machine structure has a 6-3 stewart platform. It has six legs, with six nodes at the top of the machine fixed to ground, and three nodes on the moving platform. This configuration offers several advantages over a 6-6 machine configuration. First, it reduces the number of node points that must be calculated while controlling the machine. During an inverse kinematic control calculation, the coordinates of the platform nodes are found from the tool orientation and position vector. Therefore, only three nodes must be found as opposed to six. Unfortunately, the leg lengths are the output from the inverse kinematic calculation and all six lengths must still be found. This shows that the node reduction decreases the number of calculations by a minor amount.

Second, having only three nodes on the platform reduces the platform mass. This lets the platform be accelerated faster with the same power and less control system sophisication. Third, the number of joints which can experience friction, stiction, backlash and compliance is reduced.

4.3 Hexapod Research Overview

Stewart Platform mechanisms are less intuitive to evaluate than conventional serial mechanisms. Design and analysis must be done with scale models and computer simulations tools. Our research was attacked in several phases. First, a three dimensional computer solid model was build to analyze Hexapod motion. Second, a Post Processor to convert CL Data into G-Code for the MIT Hexapod was written. Third, basic and complex programs were run to test the MIT machine. Production style parts were designed and fabricated. Last, tools and methods for process planning, machine simulation, operation and redesign were explored.

4.4 Parametric Kinematic Computer Model

A functional three dimensional virtual prototype was constructed to allow rapid, straight forward analysis of Hexapod designs. The model was created using a commerical three dimensional mechanical design automation system (CAD/CAM) made by Parametric Technology Corporation called Pro/ENGINEER^{TM}. [7] The Hexapod model is controlled by a master design layout in the form of a spreadsheet. The master layout controls the model of the machine with inputs for design variables, and end effector position data. The master layout data is linked to the three dimensional model geometry. As the layout information is changed, the resulting configuration of the mechanism can be computed. The machine geometry can be viewed and evaluated from any direction. The model can be interrogated much more easily than a real structure. Three dimensional measurements, strut lengths, leg to platform angles and other information can be found with a high degree of precision. In addition, an output report lists the current configuration of the model. It uses variables established in the master layout as limits for allowable strut lengths and angles. The output report lists in table form, the strut lengths and angles, along with boolean flags which indicate if a design limit has been violated by the master layout input. The Hexapod computer model can be thought of as a three dimensional graphic calculation system.

The model was first used to gain a more intuitive understanding of how the mechanism behaved. Next, the model was configured with the machine geometry of the MIT Hexapod to test the results of calculations and simulation programs.

To construct the computer model, the subframe plane and a fixed frame of reference was built. The nodes of the subframe were then placed as points from the subframe coordinate reference. A floating coordinate system to represent the platform is created and controlled by X, Y, Z, A, C end effector data input from the master layout. The geometry is constructed to transform the A and C rotation angles into a tool axis vector.

Next, the plane normal to the tool axis vector which passes through the tool tip coordinate is used to place the three points for the platform nodes. The six nodes of the subframe and the three nodes of the platform are connected thus establishing the strut vectors and their respective lengths. Last, angle measurement variables are created to determine the strut angles with respect to the platform and pass them into the output report.

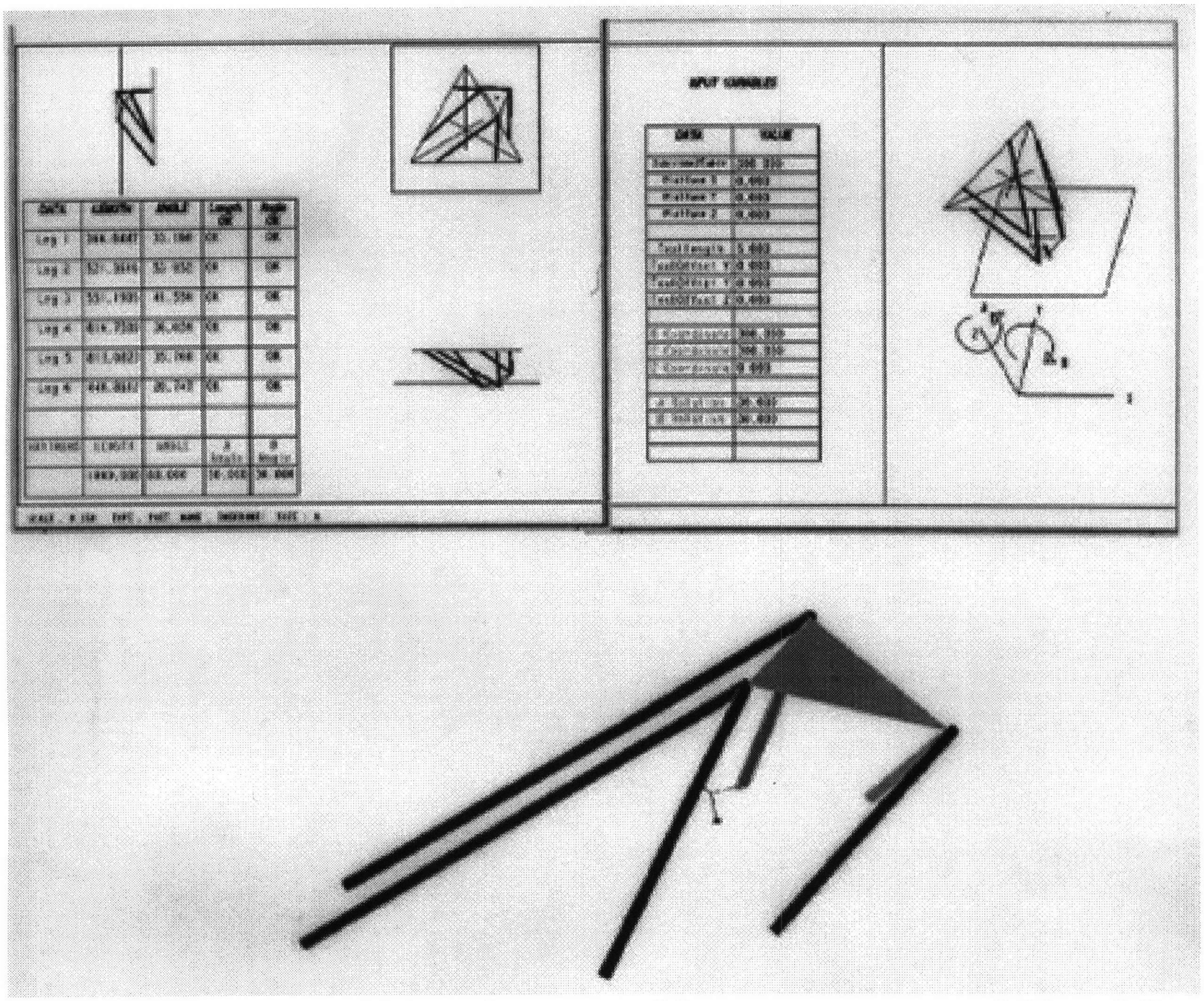

Figure 8: Virtual Model (shows input spreadsheet, output report and 3D model)

4.5 Post Processor Creation

The generic Post Processor in Pro/ENGINEER^{TM} was used to write a machine specific Post Processor for the MIT Hexapod. The generic Post Processor is a list of context sensitive questions which are used to describe the target machine and its controller. This data creates a database file for processing CL Data into G-Code. Macro programming functions are also be used to control the G-Code output.

Since the generic Post Processor is written for conventional machine tools, a modified approach was needed. A very thorough understanding of the machine mechanism, its motion and its controller, allowed it to be modeled in the Post Processor by its equivalent conventional machine geometry, and some insightful variations.

4.6 Case Study Overview

Many case studies were completed during the first year of work. The studies focused on the design through fabrication process. Projects from the MIT community were used to provide constraints, goals and deadlines. The studies included a new design of speciality soap, MassagaSoap^{TM} [6], and the molds for prototyping. Also, large scale wooden molds for the thermoformed housings of the 'Hydro-Board' were designed and fabricated. [13] In contrast, small scale, small tolerance molds were designed and machined for the 'ROBO-PIKE' Robotic Fish. [14] Each of these and other studies added insight into the design and implementation of Hexapods.

4.7 MassagaSoap^{TM} Case Study

The Massagasoap^{TM} design is soap shaped to allow the user to massage themselves while bathing. The rib geometry was optimized for stress with a closed loop, finite element optimization program linked to the design geometry. Stress was minimized by varying the rib angle. The three dimensional solid model of the design was created in Pro/ENGINEER^{TM} and toolpaths were generated. Both positive prototypes and negative molds were then machined on the MIT Hexapod. This study explored complex contouring, interlocking pieces, fixturing methods and cycle time.

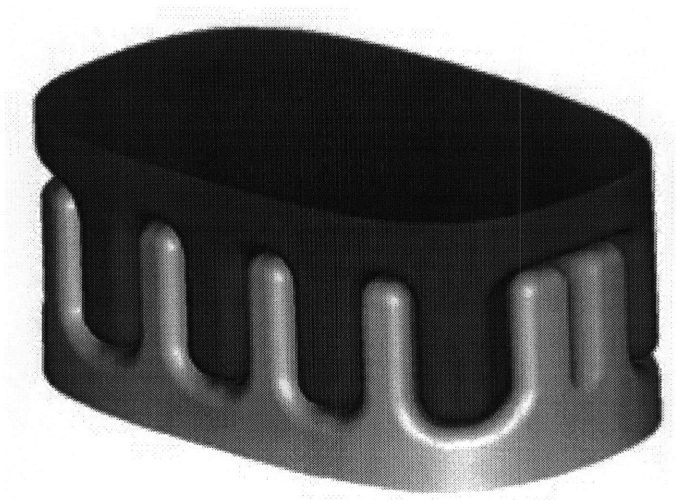

Figure 9: MassagaSoap^{TM} (interlocking)

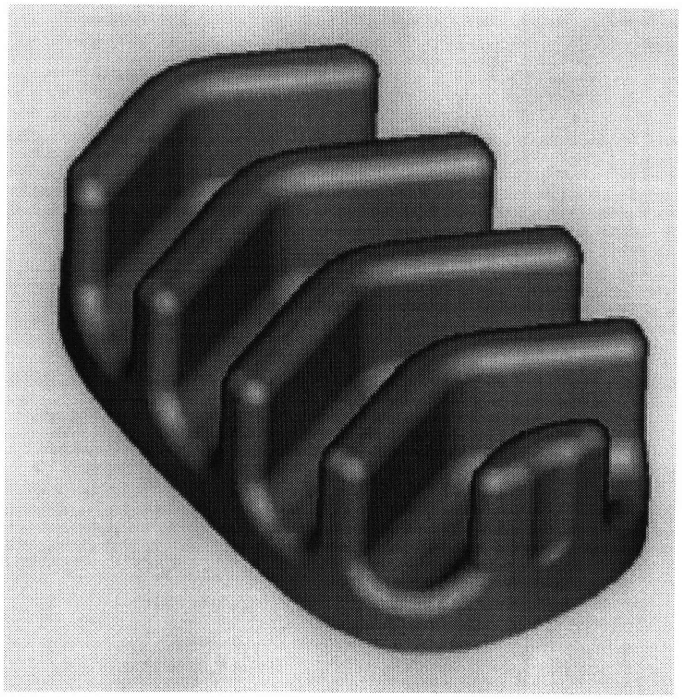

Figure 10: Top Detail Showing Rib Geometry

4.8 Hydro-Board Case Study

The hydro board was an MIT undergraduate design project. It presented a challenge for design and manufacturing, because of its large scale. The product was a battery powered, motorized kick board float for children. A full scale working prototype was required and the upper and lower plastic shells were to be thermoformed. We were asked to assist with the design of the shells and to manufacture the molds for thermoforming. We redesigned the shells based on aesthetic and ergonomic goals.

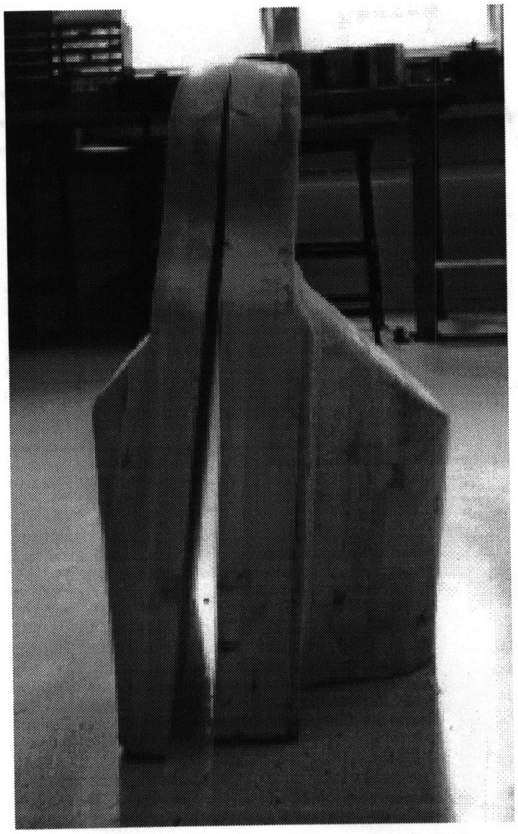

Figure 11: Hydro-Board Molds

Figure 12: Hydro-Board Completed Top Housing Mold

Figure 13: Hydro-Board Completed Bottom Housing Mold

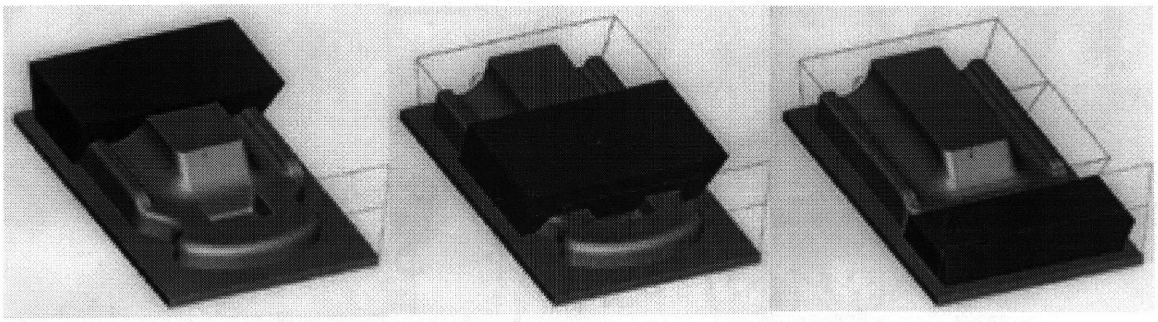

Figure 14: Hydro-Board Machining Volumes

The main reason the project was accepted was that the models would be larger than the maximum work volume of the Hexapod. The mold dimensions were 36" x 22" x 18" at the maximum. The stated work volume of the MIT Hexapod is a 600 mm cube. (23.6 inches per side) This project, more than any other, uncovered issues with the MIT Hexapod and the process of toolpath planning for a Hexapod in general.

To machine these molds the workpiece would have to be moved, and toolpaths needed to be grouped by zone to stay within the work volume of the machine. Individual volumes of reachable space were defined to solve the problem. Tool paths for each of the machining volumes were then created. The part was fixtured to slide along its longitudinal axis. It was repositioned twice for roughing and twice for finishing operations.

The bottom mold had areas were the tool would be cutting deep passes close to relatively high side walls. The clearance between the spindle and the in process geometry of the workpiece was very tight. Material removal simulations were used to insure collisions were avoided. This process proved to be very time consuming, and the results were less than satisfactory. During the machining of the bottom mold shown in Figure 15 below, the spindle body collided with the upper corner of the workpiece. The platform ball joint was released and the platform pivoted about two inches and was then held in place by the rubber boot. The problem occured because only the tool, and not the whole spindle assembly was used in the material removal simulation. It was thought that by observing the top of the tool in relation to the work in progress shape, a collision could be detected. The software, Pro/ENGINEER^{TM} only allows the tool to be used in material removal simulations, not the complete tool assembly. The complete tool assembly can be used when playing the toolpath but not when simulating material removal. Only the design part can be shown when playing toolpaths. This leaves a fundamental problem to be solved by clairvoyance. Many collisions could occur between work in progress geometry and the spindle body, which can not be adequently simulated. The best process now is to run the material removal simulation to help visualize the machining sequence and resultant geoemetry, and then modify the toolpaths on the side of caution.

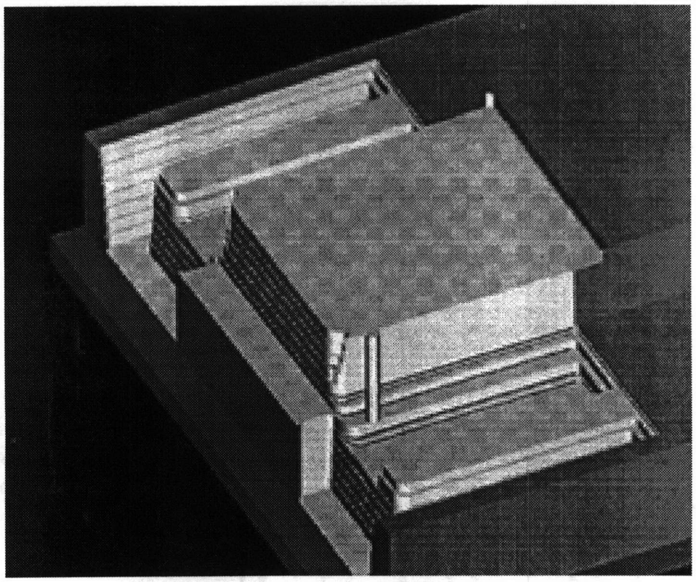

Figure 15: Material Removal Simulation

4.9 ROBO-PIKE Case Study

The Ocean Engineering Department at MIT has developed a free swimming radio controlled robotic fish. [9] The ROBO-PIKE is patterned after a Pike fish and uses precision hydrodynamic sections for its surfaces. Our group used cross section data to model the upper fin of the fish. Molds were modeled to form the fin, and toolpaths were generated. During the case study, fast machine motion along paths with sharp corners highlighted servo amplifier gain calibration problems. The machine speed was reduced to temporarily solve the overshoot problem. To date this problem which we still attribute to gain calibration error has not been resolved.

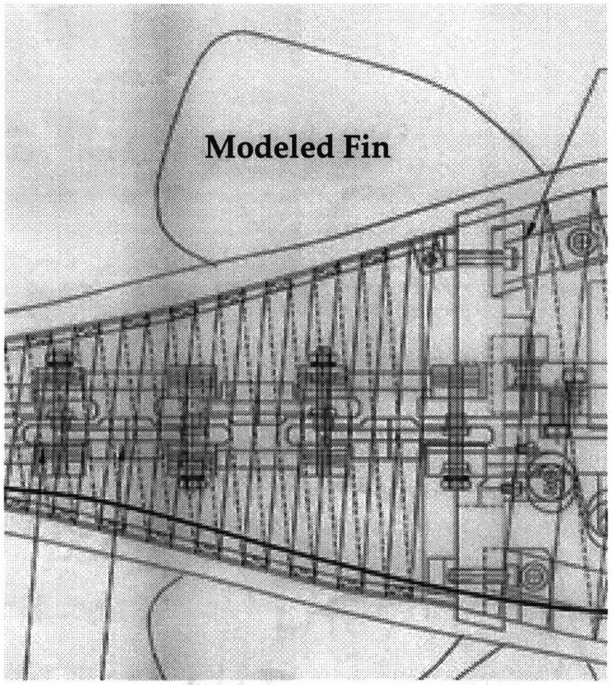

Figure 16: ROBO-PIKE Design Layout

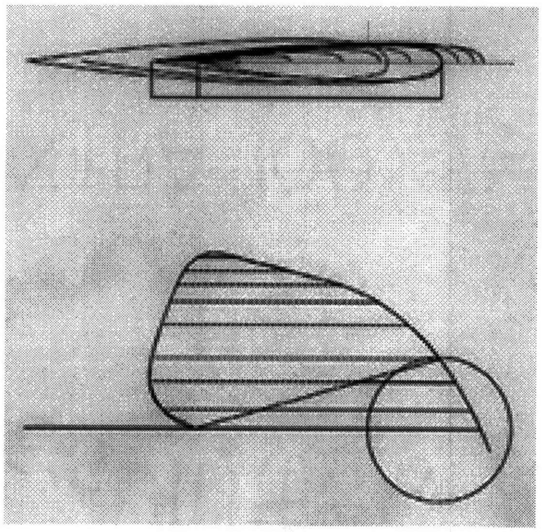

Figure 17: Fin Cross Sections

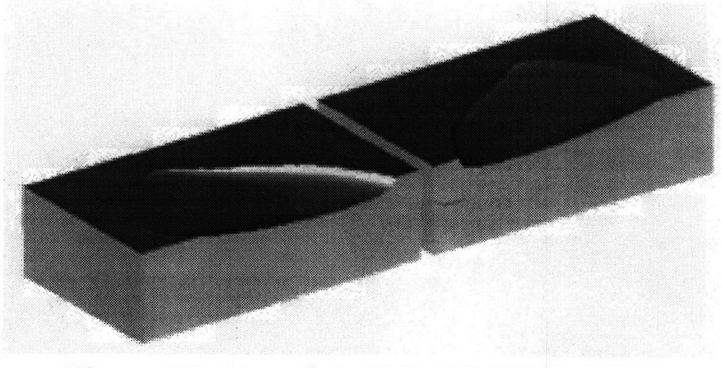

Figure 18: Completed Fin Mold Models

5.0 Hexapod Application and Design Issues

5.1 Work Volume Analysis

The basic design of the Stewart Platform is an octahedron in which one face is designated as the base, and the opposing face is designated as the platform. The platform is connected to the base by six struts which lie along the six edges of the octahedron and run between the base and the platfrom. By varying the lengths of these six struts, it is possible to cause the platform to assume a desired position with respect to the base, within its motion limits. The motion limits are the result of four mechanical limits inherent to a specific design. The work volume is defined by:
$^{1)}$ limited strut length, $^{2)}$ limited pivot angle between the platform and the strut,
$^{3)}$ limited pivot angle between the subframe and the strut, and $^{4)}$ interferance between the struts. Further limitations may be caused by the possiblity of kinematic singularities with the work volume.

The work volume of a machine is the space that is reachable by the end effector. For conventional milling machines this volume is quite regular and simple to consider when developing a manufacturing process plan. In contrast, a hexapod milling machine has a work volume which can be very generally approximated by an elliptic paraboloid. Many researchers have done work in mapping the work volume. These include Haug [10], Luh [11], Gosslein [12], and Fichter [13]. Here we examine not the calculation of the work volume, but the application of the information when developing a process plan and toolpaths. Additionally, it should be noted that standard work volume research has not taken all of the actual machine motion limits into account.

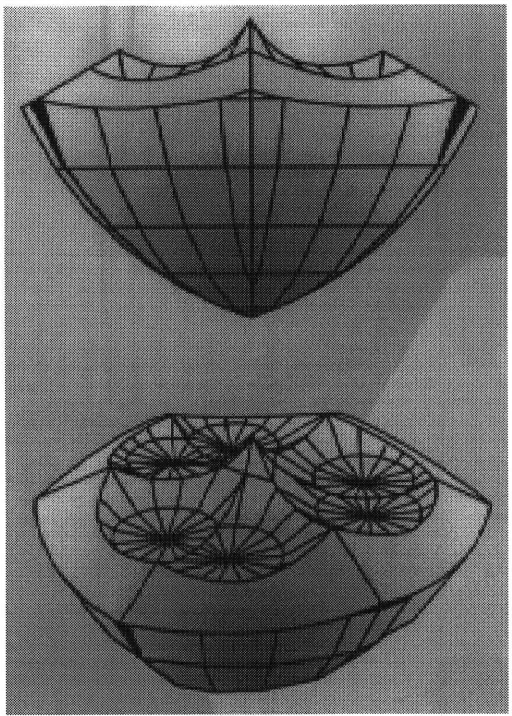

Figure 19: Work Volume [4]

Several parameters influence the work volume as it is applied to process planning and toolpath generation. First, the volume is based on the position of the end effector, and the orientation of the tool axis vector. If the tool is not tilted with respect to the table, three axis machining, the work volume takes on a different shape, in this case larger. If large tool axis vector tilt angles would be required to remove the needed material, the volume is much smaller. This is the case because positioning uses a finite resource, the available lengths of the struts.

There are four conditions which limit the work volume in a real machine. The first and most obvious is strut length, both minimum and maximum. The struts have a finite length and can only be retracted until the platform or other structure collides with the subframe. The next restriction stems from the geometry of the nodes on the platform. The joint with general ball and socket geometry can only rotate the strut a finite number of degrees before it collides with the socket or the structure of the platform. In the case of the Hexel Hexapod, the socket is magnetic and the collision would create a moment acting on the ball to remove it from the socket. It is also important to note that the distance from the plane of the platform joint centers to the tool tip directly effects the maximum rotation angle of the platform. Obviously, as the distance increases, the platform is displaced farther for a given tool axis angle.

The third condition limiting platform motion is maximum pivot angle fo the struts with respect to the subframe. On the top of the subframe, recesses must be created to allow the strut to pivot without colliding with the subframe. Services and wiring also must allow the strut to move during operation. This condition may seem to be one that is easily solved, but when striving for a design with great dexerity, it can play a role. Note that as subframe material is removed to create a larger recess and therefore strut pivot angle, the subframe will lose inherent structural stiffness.

The fourth condition limiting platform motion is the interference of the struts with one another. This occurs when the platform is rotated about the C axis. This is not a concern in the Hexel Hexapod, because the platform degree of freedom around C is removed by the controller. In other applications, including robotic painting, welding, and cutting, rotation about the C axis may be desired.

5.2 Application of Work Volume Data for Accurate Process Planning

For process planning, work volume data must be considered to utilize the full capabilities of a Hexapod milling machine. Since the volume is not a rectangle and it changes size and shape based on the tool angle, a powerful visual method for process planning is called for.

The method we developed was to use the three dimensional surface of the work volume for a given tool angle in the CAD/CAM system to choose the location and orientation of the workpiece on the table. X, Y and Z direction shifts, as well as workpiece orientation can quickly be evaluated, though only snapshots at various angles can be used. A better method would be dynamic.

5.3 Vibration Issues

The stewart platform design can be implemented to reduce dynamic and quasi-static position errors. Some dynamic errors are caused by machine vibration which leads to poor surface finish and dimensional control. Vibration can be difficult to diagnose and remedy because its input energy can come from many sources. The difficulty is further compounded by the Hexapod structure. Unwanted vibration modes can not be isolated to one control axis or structure. A detailed dynamic model must be developed and the design refined to move the structure's modes away from the frequencies of the input energies. Hexapod structure machine dynamics have been explored by Yang [14], Pang [15], and Do [16].

The structure's modes are based its stiffness. As the pose (position and orientation) of a Hexapod changes, the stiffness it the major directions changes. Therefore, the dynamic characteristics of a Hexapod change with the end effector's pose in the work volume. This creates the need for a modal map of the work volume. From this map, a 'vibrational work volume' could be found in which the dynamics of the structure are within the required limits.

5.4 Hexapod Stiffness Issues

As a baseline model we assume that the joints of the structure induce no compliance, and that struts are loaded in the axial direction only. This follows work by Yang [17]. The stiffness a Hexapod can then be found using the platform Jacobian representing its physical configuration and modeling the struts as linear springs. The stiffness is therefore dependent on the strut length for platform configuration and for the spring constant of the representative springs. Obviously, the stiffness function withing the reachable work volume is far from regular.

A stiffness map, or 'stiffness work volume' can be constructed and used in a similar manner to the 'vibrational work volume' already purposed. The stiffness work volume would show the spacial limits of the end effector based on minimum stiffness in each of the directions. A stiffness work volume could be created for each stiffness direction and a composite stiffness work volume could be developed which uses all the directional stiffness limits.

6.0 Design and Simulation Tools

6.1 Machine Inverse Kinematics

In our research, the inverse kinematics were used to test G-Code and to evaluate machine motion limits during process planning. A program was written in Mathematica^{TM} [18] which reads a G-Code file and parses the data into usable arrays. The inverse kinematic calculations were then carried out, after work by Yang [19], and Liu [20]. Strut length, and platform joint data can be calculated from the structure's calculated node positions. In addition, the program outputs data to a file in a format suitable as input for a three dimensional graphic animation of the machine's motion.

(Program Listing - Appendix B)

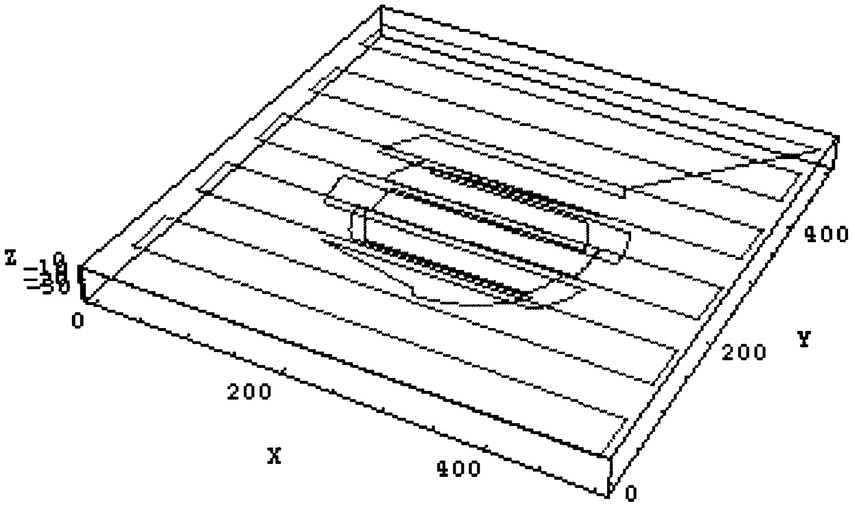

Figure 20: Inverse Kinematic Program Toolpath Display

6.2 Machine Motion Animation

The toolpath simulations in Pro/ENGINEERTM show only the tool or spindle assembly. A simulation of the structure which moves the spindle can be very useful for a Hexapod. The motion of the struts and the platform were animated with a Program written in MegahedronTM [21]. It allowed us to visualize the complete machine during the evaluation of toolpaths. The program reads the file output from the inverse kinematic calculations and animated the machine's motion.

(Program Listing - Apppendix A)

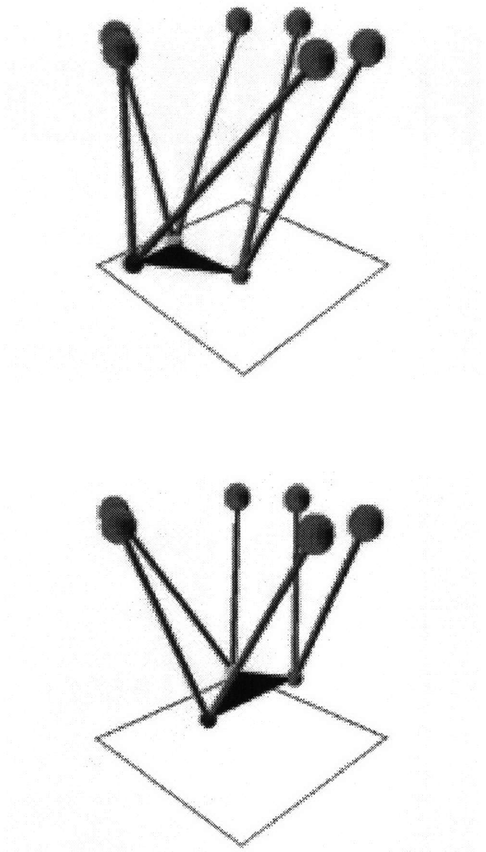

Figure 21: Animation Snapshots

7.0 Conclusions and Further Work

We have very little experience designing and implementing Hexpaod machine tools. New methods and innovative design solutions are needed to solve problems with control, accuracy, calibration, vibration, stiffness, temperature compensation, and usable work volume. The benefits of Hexapods can be achieved but large amounts of time, brain power, and money are needed before Hexapods will be production ready.

Because of the complexity of a Hexapod, we loss the intuitive ability to predict machine behavior. Once we lose forward predictability we must analyze in the reverse direction.

One path for further work is to link the computer programs written during this work and add the evaluation of vibration modes and machine stiffness. A set of design parameters could then be used to calculate a relatively complete view of a design. This view could include the physical work volume, the vibrational work volume based on input energy frequencies, and the stiffness work volumes based on required operational stiffnesses in each direction. These volumes, real three dimensional maps, could then be compared and contrasted to gain a very accurate understanding of a designs real performance envelop, not just the motion limits of its end effector.

Since we do not know how to exercise the system to provoke its hidden faults, everything is new, trial and error iteration. We need detailed models to build experience and knowledge base. Finally,we must create a forward system model where functional requirements can be used to calculate machine design parameters.

8.0 Appendix A

```
{*****************************************************}
{* Hexapod Motion Simulation Program - in Megahedron  *}
{*****************************************************}
{* Reads input file from Displacement Analysis        *}
{*****************************************************}
{*****************************************************}

do initialize,sim;

{* declare main data array *}
vector data[1..15000];


{*****************************************************}
{*Procedure to initialize the input file stream and to *}
{*read in the data points for the nodes of the machine *}
{*platform.  The data is read in ,in the format :     *}
{* MOVE #1 ball #1 data point                 *}
{* MOVE #1 ball #2 data point                 *}
{* MOVE #1 ball #3 data point                 *}
{* MOVE #2 ball #1 data point  and so on.......       *}
{*****************************************************}

procedure initialize is

{* input file name is added here *}
integer file_id = fopen 'output503' true true;

integer i,j,k;
vector a,b,c,d;

for i= 1..14556 do

freadln file_id data[i];


end; {for loop}


fclose file_id;


end; {procedure initialize}
```

```
{******************************************************}
{******************************************************}
{*Graphical Procedure of type PICTURE is one image to be*}
{*processed in the complete animation, a set of pictures*}
{*The picture sets the scene orientation, eye point of  *}
{*view as well as the objects (primitives) to be added  *}
{*to the display list.  This picture takes three        *}
{*variables as input, the variables are of the type :   *}
{* "vector" which has the form (x,y,z) where each value *}
{*is a floating point number.  The procedure uses the   *}
{*input data throughout the function to position the    *}
{*individual elements of the simulation animation.      *}
{******************************************************}
{******************************************************}

picture pict
{mandatory parameters}

vector vec1,vec2,vec3;

with

{* Position the virtual camera *}
{* set camera to TOP view *}
{eye = 0 0 1500;}

{* set camera to 3D view *}
{eye= 1250 1250 750;}
eye= 1300 1300 50;


{* Position the location of interest *}
lookat = 0 0 100;

{* set the render mode for the picture procedure *}
render_mode=shaded;
{render_mode=wireframe;}

is

{* Add a light to the scene *}
distant_light;
```

```
{* Add light from the bottom *}
distant_light with direction = 1 0 -1 ; end;


{* Add the outline of the workpiece table *}
{table outline - not shaded}

line [(-400 -400 -300) (400 -400 -300) (400 400 -300) (-400 400 -300)
    (-400 -400 -300)] ;


{* Add the Sphere Drive Nodes as spheres in the subframe *}
{subframe - sphere drive spheres - do not move}

sphere with center=408 -95 500;radius=50;color=green;end;
sphere with center=286 -306 500;radius=50;color=green;end;
sphere with center=-286 -306 500;radius=50;color=green;end;
sphere with center=-408 -95 500;radius=50;color=green;end;
sphere with center=-122 400 500;radius=50;color=green;end;
sphere with center=122 400 500;radius=50;color=green;end;


{* Add the MAIN simulation component set of objects *}
{platform - triangle and bi-balls}

triangle vec1 vec2 vec3;
sphere with center=vec1;radius=25;color=red;end;
sphere with center=vec2;radius=25;color=green;end;
sphere with center=vec3;radius=25;color=yellow;end;


{* Add the representations of the ball screw legs *}
{legs - cylinders}

cylinder with end1=408 -95 500;end2=vec1;radius=10;end;
cylinder with end1=286 -306 500;end2=vec3;radius=10;end;
cylinder with end1=-286 -306 500;end2=vec3;radius=10;end;
cylinder with end1=-408 -95 500;end2=vec2;radius=10;end;
cylinder with end1=-122 400 500;end2=vec2;radius=10;end;
cylinder with end1=122 400 500;end2=vec1;radius=10;end;
```

```
{* Add the legs using lines instead of solid cylinders -for speed *}
{ legs with lines}
{
line[(408 -95 500) (vec1)];
line[(286 -306 500) (vec3)];
line[(-286 -306 500) (vec3)];
line[(-408 -95 500) (vec2)];
line[(-122 400 500) (vec2)];
line[(122 400 500) (vec1)];
}

end;  { picture procedure }

{*******************************************************}
{*******************************************************}
{*******************************************************}
{*Graphical Procedure of type ANIMATION = thesimulation *}
{*It displays any pictures called within it in sequence *}
{*******************************************************}
{*******************************************************}
{*******************************************************}

anim sim

with

{ save_pictures = on; }
{ file_format   = raw; }

double_buffer = on;
facets = 8;
background = white;

is

integer i;

i = 1;

{ i < 14555 }
{* Begin Animation Loop *}
while i < 14555   do

pict data[i] data[i+1] data[i+2];
```

```
i = i + 3;

end;  {for loop}


end; {the anim}

{* End Program *}
{**************************************************************}
{**************************************************************}
{**************************************************************}
```

9.0 Appendix B

Stewart Platform Inverse Kinematic Analysis in Mathematica

File Input and Output

Reads a Gcode file and places it in an array

```
In[10]:=
Directory[]

Out[10]=
"Macintosh HD:Software:Mathematica 3.0 Files"

SetDirectory["\:f3b5Macintosh HD:1WORK:Thesis2"]
```

Stream Input Function Definition

```
In[11]:=
streamget:=
Module[{streamdata,maxlength,currentspot,fulldata,basearr,bl,tg,tx,ty,tz,ta,
    tc,i},

Print["Extracting the Number of Moves from G Code"];
basearr =
    ReadList["ms5a03s.tap",{Word,Word,Word,Word,Word,Word},
     WordSeparators -> {"G","X","Y","Z","A","C"," "},
     RecordSeparators-> {"\n"}];
bl=Length[basearr];
Print[" Total Number of Points =  ",bl];

fulldata = Table[{0,0,0,0,0},{bl}];
streamdata=OpenRead["ms5a03s.tap"];
```

```
Print["Checking Data Length"];
SetStreamPosition[streamdata,Infinity];
maxlength=StreamPosition[streamdata];
Print["Total File Length ",maxlength];
SetStreamPosition[streamdata,0];

i= 1;

Print["Reading Data"];

While[StreamPosition[streamdata] < maxlength,
Skip[streamdata,Character];
tg=Read[streamdata,Number];
(*Print["tg = ",tg];*)
Skip[streamdata,Character];

 Skip[streamdata,Character];
tx=Read[streamdata,Number];
(*Print["tx = ",tx];*)
fulldata[[i,1]]=tx;
Skip[streamdata,Character];

 Skip[streamdata,Character];
ty=Read[streamdata,Number];
(*Print["ty = ",ty];*)
fulldata[[i,2]]=ty;
Skip[streamdata,Character];

 Skip[streamdata,Character];
```

```
tz=Read[streamdata,Number];
(*Print["tz = ",tz];*)
fulldata[[i,3]]=tz;
Skip[streamdata,Character];

 Skip[streamdata,Character];
ta=Read[streamdata,Number];
(*Print["ta = ",ta];*)
fulldata[[i,4]]=ta;
Skip[streamdata,Character];

 Skip[streamdata,Character];
tc=Read[streamdata,Number];
(*Print["ta = ",ta];*)
fulldata[[i,5]]=tc;
Skip[streamdata,Character];
i=i+1;
];
  Print["Data Read Complete"];

Close[streamdata];
fulldata];
```

Displacement Analysis Function Definitions

```
In[12]:=
<<Calculus`VectorAnalysis`
```

Transformation Function Tp Definition

Simplified Bifurcated ball data

```
Tp[xp_,yp_,zp_,aangle_,cangle_]:=
Module[{\[Alpha],\[Gamma],iaxis,jaxis,kaxis,Rx,Rz,endx,endy,endz,
tpmatrix,tpo,i,twist},

(* tpmatrix is a 4x4 array *)
(* output array is a 4x3 array *)
(* Print["Init. tpmatrix"]; *)
tpmatrix=Table[{0,0,0,0},{4}];
output=Table[{0,0,0,0},{3}];
(* only need to do this once *)
tpmatrix[[4,1]]=0;
tpmatrix[[4,2]]=0;
tpmatrix[[4,3]]=0;
tpmatrix[[4,4]]=1;

(* Print[ xp," ",yp," ",zp," ",aangle," ",cangle]; *)

ball1={175,0,0};
ball2={-85,150,0};
ball3={-85,-150,0};
coords={ball1,ball2,ball2,ball3,ball3,ball1};

(* Print["converting angles to radians"];*)
(* convert the read in control angles to radians (alpha and gamma) *)
\[Alpha]=aangle*.01745329;
\[Gamma]=cangle*.01745329;
(* twist is the rotation of the platform wrt the base *)
(* in this case found by measuring machine *)
```

```
twist=43.0412*.01745329;
(* now the twist is added to gamma - not used now *)
(*  \[Gamma]=twist+\[Gamma]; *)
(* Print["gamma plus twist = ",\[Gamma]]; *)

(* Print["Init. unit vectors - iaixs,jaxis,kaxis"]; *)
iaxis={1,0,0};
jaxis={0,1,0};
(* k axis equals the TOOL VECTOR *)
kaxis={0,0,1};


(* Print["Define Rx, Rz, and tpo Matrices"]; *)
(* Rx and Rz are rotations only, tpo is a HTM with translations *)

Rx={{1,0,0},{0,Cos[\[Alpha]],Sin[\[Alpha]]},{0,-Sin[\[Alpha]],Cos[\[Alpha]]}};


Rz={{Cos[\[Gamma]],-Sin[\[Gamma]],0},{Sin[\[Gamma]],Cos[\[Gamma]],0},{0,0,1}};


(* not being used now *)
tpo={{Cos[twist],-Sin[twist],0,0},{Sin[twist],Cos[twist],0,0},{0,0,1,0},{0,0,
       0,1}};

 (* Print["Rotate K unit vector about X with Rx"]; *)
endz=kaxis.Rx;

(* Print["Rotate K unit vector about Z with Rz"];*)
endz=endz.Rz;
```

```
(* Use K tool vector and find the i and j vectors without rotation *)
(* Print["Use cross product to find endy and endx"]; *)

endy=CrossProduct[endz,iaxis];
endy=endy/Sqrt[endy[[1]]^2+endy[[2]]^2+endy[[3]]^2];
(* Print["endy equals ",endy]; *)

endx=CrossProduct[jaxis,endz];
endx=endx/Sqrt[endx[[1]]^2+endx[[2]]^2+endx[[3]]^2];
(* Print["endx equals ",endx]; *)

(* Print["Placing Direction Cosines into tpmatrix positions"];*)
Do[
tpmatrix[[1,i]]=endx[[i]];
    tpmatrix[[2,i]]=endy[[i]];
tpmatrix[[3,i]]=endz[[i]];
,{i,1,3}];



(* Print["Placing the translations into the tpmatrix"];*)
tpmatrix[[1,4]]=xp;
tpmatrix[[2,4]]=yp;
tpmatrix[[3,4]]=zp;
(* Print["tpmatrix"]; *)
(* Print[MatrixForm[tpmatrix]]; *)
(* TPMATRIX fill out moved to outside the loop *)

(* Print["Placing b1,b2,b3 coordinates (in the xyz coordinate system)"]; *)
```

```
b1={coords[[1,1]],coords[[1,2]],coords[[1,3]],1};
b2={coords[[2,1]],coords[[2,2]],coords[[2,3]],1};
b3={coords[[4,1]],coords[[4,2]],coords[[4,3]],1};
(* Print["b1 = ",b1,"b2 = ",b2,"b3 = ",b3]; *)
(* rotate the ball data wrt B *)
(* b1=tpo.b1;
b2=tpo.b2;
b3=tpo.b3;  *)

output[[1]]=tpmatrix.b1;
output[[2]]=tpmatrix.b2;
output[[3]]=tpmatrix.b3;
(* Print["output"]; *)
(* Print[MatrixForm[output]]; *)
output];
```

XYZ Extraction Function 1

Take the array name and the row to extract data from

```
In[14]:=
xyzs3[listdb_,num_]:=
Module[{xyzdb,i},
i=num;
xyzdb={0,0,0};
xyzdb[[1]]=listdb[[i,1]];
xyzdb[[2]]=listdb[[i,2]];
xyzdb[[3]]=listdb[[i,3]];
xyzdb];
```

XYZ Extraction Function 2

Extracts the first three members from the x,y,z,a,c array

```
In[15]:=
xyzget[listdb_]:=
Module[{i,xyzdb,maxlength},

Print["Extracting the XYZ data from the array"];
maxlength=Length[listdb];
xyzdb=Table[{0,0,0},{maxlength}];
Do[
xyzdb[[i,1]]=listdb[[i,1]];
xyzdb[[i,2]]=listdb[[i,2]];
xyzdb[[i,3]]=listdb[[i,3]];
,{i,1,maxlength}
];

xyzdb];
```

Output Data Function

Takes an input array which is n x 3 and outputs to a file

```
streamput[array_]:=
Module[{i},
streamdata=OpenWrite["output01"];
Print["Beginning Streamput "];
Do[
WriteString[streamdata,array[[i,1]]," " ,array[[i,2]]," " ,
```

```
array[[i,3]],"\n"];


i=i+1;

,{i,1,Length[array]}
];
Print["Exiting Streamput"];
Close[streamdata];
];
```

Data Processing Function Definition

NOTE: process function now writes data to file - output01

```
In[17]:=
process[in_]:=
Module[{i,j,outpb1,outpb2,outpb3,pics},

Print["Entering PROCESS function"];
(* outp=Table[{0,0,0},{Length[in]*3}]; *)
(* outpb1={0,0,0};
outpb2={0,0,0};
outpb3={0,0,0}; *)


streamdata=OpenWrite["output01"];
Print["Opening File"];
```

```
Do[
plat=Tp[in[[i,1]],in[[i,2]],in[[i,3]],in[[i,4]],in[[i,5]]];

(*pics= Graphics3D[{RGBColor[1,0,0],
        Line[{xyzs3[plat,1],xyzs3[plat,2],xyzs3[plat,3],
          xyzs3[plat,1]}]},Axes-> True,BoxStyle -> RGBColor[0,0,0],
      AxesLabel->{"X","Y","Z"}]; *)
(* Print["coords",xyzs3[plat,1],xyzs3[plat,2],xyzs3[plat,3]]; *)
 (*  Show[pics];*)

(* outpb1[1]=plat[[1,1]];
outpb1[2]=plat[[1,2]];
outpb1[3]=plat[[1,3]];

outpb2[1]=plat[[2,1]];
outpb2[2]=plat[[2,2]];
outpb2[3]=plat[[2,3]];

outpb3[1]=plat[[3,1]];
outpb3[2]=plat[[3,2]];
outpb3[3]=plat[[3,3]];   *)

WriteString[streamdata,plat[[1,1]]," " ,plat[[1,2]]," " ,plat[[1,3]],"\n"];
WriteString[streamdata,plat[[2,1]]," " ,plat[[2,2]]," " ,plat[[2,3]],"\n"];
WriteString[streamdata,plat[[3,1]]," " ,plat[[3,2]]," " ,plat[[3,3]],"\n"];



(* Length[in] *)
```

```
,{i,1,Length[in]}];

Print["Leaving PROCESS function"];
outp];
```

Test Evaluations

```
In[18]:=
process[streamget];

"Extracting the Number of Moves from G Code"
" Total Number of Points =  "\[InvisibleSpace]54
"Checking Data Length"
"Total File Length "\[InvisibleSpace]2126
"Reading Data"
"Data Read Complete"
"Entering PROCESS function"
"Opening File"
"Leaving PROCESS function"
```

END Inverse Kinematic Analysis Program

10.0 Bibliography

[1] Hexel Corporation, manufactures the Hexel Hexapod, Hexel licenses SphereDrive^{TM} and Bi-Furcated Ball^{TM} from Geodetics Ltd [4], AMC^{TM} is a Trademark of the Hexel Corporation, Portsmouth, NH

[2] Universal Tyre Test Machine, Bough, V.E. and Whitehall, S.G., *Proc. 9th Int. Tech Congress F.I.S.I.T.A.*, May 1962, page 117

[3] Stewart, D., "A Platfrom with Six Degrees of Freedom", *The Institution of Mechanical Engineers, Proceedings 1965-66*, Volume 180, Part 1, No.15

[4] Geodetic Technology International Limited, manufactures the Geodetic Hexpod and holds patents to the SphereDrive^{TM} and Bi-Furcated Ball^{TM} technology, West Sussex, England

[5] Seames, Warren S., *Computer Numerical Control - Concepts and Programming* 3rd Edition, Prentice Hall, Englewood Cliffs, NJ. 1995

[6] Aesop Inc., Shear Tube^{TM} is a trademark of Aesop Inc., MassagaSoap^{TM} is a Trademark of Aesop Inc., NH

[7] Parametric Technology Corp., Pro/ENGINEER^{TM} is their Trademark, Waltham, MA

[8] Thanks to the Hydro-Board 2.73 Team

[9] Special thanks to John Muir Kumpf, Principle ROBO-PIKE Designer

[10] Haug, E.J., Luh, C.M., Adkins, F.A., Wang, J.Y., "Numerical Algorithms for Mapping Boundaries of Manipulator Workspaces", *Journal of Mechanical Design*, Transactions of the ASME, Vol. 118, June 1996

[11] Luh, C.M., Adkins, F.A., Haug, E.J., Qiu, C.C., "Working Capability Analysis of Stewart Platforms", *Journal of Mechanical Design*, Transactions of the ASME, Vol. 118, June 1996

[12] Gosselin, C., "Determination of the Workspace of 6-DOF Parallel Manipulators", *Journal of Mechanical Design*, Transactions of the ASME, Vol. 112, Sept. 1996

[13] Fichter, E.F., "A Stewart Platform Based Manipulator: General Theory and Practical Construction", *International Journal of Robotics Research*, Vol. 5, No. 2, Summer 1986

[14] Yang, H., and Masory, O., "Vibration Analysis of Stewart-Platform", Dept. of Mechanical Engineering, Florida Atlantic University, Pub. Unknown

[15] Pang, H., and Shahinpoor, M., "Inverse Dynamics of a Parallel Manipulator", *Journal of Robotic Systems* ,11(8), pp.693-702, 1994

[16] Do, W.Q.D., and Yang, D.C.H., "Inverse Dynamic Analysis and Simulation of a Platfrom Type of Robot", *Journal of Robotic Systems*, 5(3), 209-227, 1988

[17] Yang, H., and Masory, O., "Vibration Analysis of Stewart-Platform", Dept. of Mechanical Engineering, Florida Atlantic University, Pub. Unknown

[18] Wolfram, Stephen, *The Mathematica Book*, 3rd Edition, Cambridge University Press, Cambridge, England, 1996

[19] Yang, D.C.H., and Lee, T.W., "Feasibility Study of a Platform Type Robotic Manipulator from a Kinematic Viewpoint", *Journal of Mechansims, Transmissions and Automations in Design*, Vol.106, June 1984

[20] Liu, K., Fitzgerald, J.M., and Lewis, F.L., "Kinematic Analysis of a Stewart Platform Manipulator", *IEEE Transactions on Industrial Electronics*, Vol.40, No.2, April 1993

[21] MegahedronTM is a Trademark of the Syndesis Corp., Jefferson WI