THE UNIVERSITY
of LIVERPOOL

The Design and Control of A Remotely Operated Underwater Vehicle

UNIVERSITY
LIBRARY

by

John Richard Swainston

Thesis submitted in accordance with the requirements of The University of Liverpool for the Degree of Doctor in Philosophy

August 1996

Department of Electrical Engineering and Electronics

For Liz et al

Acknowledgments

I would first of all like to express my thanks to Dr. David Shimmin who has shown great patience, tolerance, encouragement and friendship throughout the duration of this research. Without his help much of this work may not have been undertaken, and this thesis may not have been completed.

I must also thank my supervisor Professor J. Lucas for giving me the chance to complete this research. I would also like to thank all the members of the mechanical workshop John, Tommy, Bob, Keith and Ian for turning fantasy into reality by constructing the Aquasphere from some very iffy engineering drawings. Likewise I would like to thank the members of the electronic workshop Ron, George, Mike, Alan and Ian for assistance in creating the electronic systems designed for the project. I would also like to thank other members of the department including the Head of Department Professor G. R. Jones for the use of the laboratory facilities and for support in some non-academic matters encountered during my research. I would also like to thank the other members of staff, both technical and non-technical, who are all involved in the running of the department

I would also like to thank the rest of my colleagues in the research group, who throughout the years have provided many friendships and even more distractions, making working in the department a more pleasant place.

In addition I would like to thank Comex and the Science and Engineering Research Council for financial support towards this project.

Lastly, but not leastly, I would also like my friends outside of the department for attempting to keep me sane, happy and healthy throughout the years, and particularly during the writing of this thesis. I would also like to thank my family, for their patience in waiting for me to "get a proper job", and for everything else they have done over the years.

Finally I would like to thank Liz for always being there for me…..

John R. Swainston

September 1996.

Abstract

The exploitation of natural resources has lead to the increased demand for Remotely Operated Vehicles for use in the inherently dangerous field of underwater operation. Conventional underwater vehicles require one or more highly skilled pilots to control and maintain the position of the vehicle underwater. This task is difficult and fatiguing requiring a high level of concentration for the pilot. Conventional underwater vehicles are highly specialised for specific tasks and require reconfiguration or replacement to carry out non-standard tasks.

By designing a novel six degrees of freedom underwater vehicle called the Aquasphere a fully controllable underwater vehicle has been created. The unique design of this vehicle allows many of the deficiencies found in conventional designs to be overcome. The configuration of the vehicle allows for compensation to be made for horizontal drift and other disturbance factors.

As the thrusters mounted on an underwater vehicle have a profound effect on the performance of the vehicle as a whole, a detailed study of the thrusters has been completed. Using the information on the thrusters a computer modeling method for a thruster system has been implemented. To drive the thrusters a pulse width modulated technique has been used to overcome the stiction commonly found in thruster systems.

One of the most important components of a remotely operated vehicle is the sensor array attached to it. The performance of the sensors can be the limiting factor for the performance of the vehicle as a whole. To achieve the necessary control of the vehicle a study of current sensor systems has been undertaken.

To overcome the highly non-linear control of the Aquasphere a fuzzy proportional integral derivative based control system has been used. This method gives both a highly stable control system, with the benefit of providing non-linear control for the vehicle. Automatic depth holding, automatic heading and automatic leveling control has been added to the vehicle to reduce the complexity of the control task for the pilot.

Contents

1. INTRODUCTION 1

1.1 The M.A.S.T. Project 3

1.2 Thesis Outline 5

2. REMOTELY OPERATED VEHICLES 7

2.1 ROV Design 7

2.2 Vehicle Control Problems 11

2.2.1 Vehicle Modeling 12

2.2.2 Cable Drag 12

2.2.3 Manipulator Arm Motion 12

2.2.4 Thruster Systems 13

2.2.5 Sensor Systems 13

2.3 Vehicle Control Methods 15

2.3.1 PID Control 15

2.3.2 Fuzzy Logic 15

2.3.3 Adaptive Control Techniques 16

3. VEHICLE DESIGN 18

3.1 The Aquacube 18

3.1.1 Design of the Aquacube 18

3.2 Original Vehicle Concept and Design for the Aquasphere 20

3.2.1 Basic Sub-Member Design 23

3.2.2 Thruster Mounting Design 24

3.2.3 Spider Joint Design 26

3.2.4 The Completed Frame 26

3.3 The Aquasphere Prototype Construction 28

3.3.1 The Internal Frame 29

3.3.2 Design of the Electronics Boxes 30

3.3.2.1 The Networking Box 30

3.3.2.2 The Sensor Box 31
3.3.2.3 Thruster Power and Control Box 33
3.3.2.4 The Thruster Cowling 33
3.3.3 Mounting The Vision System 34
3.3.3.1 The Pan and Tilt Camera Unit 35
3.3.3.2 The Laser Range-finder Mounts 37
3.3.3.3 The Mounted Vision System 39
3.3.4 Mounting the Manipulator Arm 39
3.3.5 The Aquasphere Prototype 40

3.4 The Final Design of the Aquasphere 41
3.4.1 The Internal Frame 42
3.4.2 The Box Support Frames 43
3.4.3 Final Component Locations 44

3.5 Vehicle Buoyancy 46

3.6 The Completed Aquasphere 49

4. THRUSTER CONTROL AND DRIVE 50

4.1 Thruster Specification 50

4.2 Thruster Testing 51
4.2.1 Thruster Dry Tests 52
4.2.1.1 Rotational Speed 52
4.2.1.2 Motor Torque 53
4.2.1.3 Thruster Back E.M.F. 56
4.2.1.4 Thruster Step Response 56
4.2.2 Thruster Wet Tests 59
4.2.2.1 Thrust Tests 60
4.2.2.2 Thruster Step Tests 61
4.2.3 Thruster Model 63
4.2.3.1 Model Top Level 63
4.2.3.2 Thruster Demand Model 64
4.2.3.3 Power Amplifier Model 65
4.2.3.4 Thruster Model 65
4.2.3.4.1 Stiction Model 65
4.2.3.4.2 Motor Model 66
4.2.3.4.3 Lumped Parameter Propeller Model 68

4.2.3.4.4 The Overall Thruster Model 73
4.2.4 Model Verification 74

4.3 Motor Drive 77
4.3.1 P.W.M. Motor Drive 77
4.3.1.1 Waveform Generation Circuit 78
4.3.1.1.1 Triangular Waveform Generation 78
4.3.1.1.2 P.W.M. Generation Circuit 80
4.3.1.1.3 Gate Drive Circuit 81
4.3.1.1.4 Design of the H-Bridge Power Drive 82
4.3.1.1.5 Constructing The H-Bridge Power Drive 85

4.4 Advanced Motor Drive 87
4.4.1 Microcontroller Systems 87
4.4.1.1 The INTEL 87C51FA Microcontroller 87
4.4.1.2 Microcontroller System Design 89
4.4.1.2.1 Serial Interfacing 90
4.4.1.2.2 P.W.M. Generation 92
4.4.1.2.3 Counter/Timer Input 94
4.4.2 Thruster Gate Drive 94
4.4.3 Power Drive Stage 95
4.4.4 The New Power Module 96

5. SENSOR SYSTEMS 97

5.1 Sensor Specifications 97
5.1.1 Pressure Sensors 98
5.1.2 Flow Sensors 98
5.1.3 Gyroscopic Sensors 99
5.1.4 Flux-Gate Sensors 99
5.1.5 Accelerometers 100
5.1.6 Attitude Sensors 101
5.1.7 GPS Systems 101
5.1.8 Acoustic Sensors. 103
5.1.9 Visual Systems 104
5.1.10 Navigationally Related Sensors 105
5.1.10.1 Temperature Sensors 105
5.1.10.2 Salinity Sensor 105

5.2 The Aquasphere's Sensor Systems 105

5.3 Sensor Performance **107**
5.3.1 Vision System Performance 107
5.3.2 Acoustic System Performance 107
5.3.3 NMEA Interfaced Devices 108
5.3.3.1 Flux Gate Compass Performance 108
5.3.3.2 Echosounder Performance 108
5.3.3.3 Flow Sensor Performance 108
5.3.4 Pressure Sensor Performance 109
5.3.5 Gyroscope Performance 109
5.3.6 Accelerometer Performance 110
5.3.7 Inclinometer Performance 110

5.4 Sensor Summary **110**

6. NETWORKING SOFTWARE **111**

6.1 The Original Interface System **111**
6.1.1 PC61A Interface Card 112
6.1.2 PC 66 Interface Card 113
6.1.3 NMEA Sensor Interfacing 114
6.1.3.1 NMEA Hardware Interface 114
6.1.3.2 NMEA 0183 Sentence Format 116
6.1.3.2.1 Sensor Address Information 117
6.1.3.2.2 Sample NMEA Data 118
6.1.3.3 NMEA Interpretation Software 118

6.2 Transputer Networking System **119**
6.2.1 P.C. Interfacing 120
6.2.1.1 PC Hardware Interface Software 120
6.2.2 Network Data Protocol 121
6.2.2.1 Thruster Protocol 123
6.2.2.2 Sensor Protocol 124
6.2.2.3 Vehicle Control System Protocol 125
6.2.3 PC Communication Software 126
6.2.3.1 Data Output Software 127
6.2.3.2 Data Input Software 128
6.2.4 Hardware Interfacing 130
6.2.4.1 NMEA Sensor Interfacing 131
6.2.4.2 Pressure Sensor Interfacing 131
6.2.4.3 Inclinometer Interfacing 131

6.2.4.4 Gyroscope Interfacing 131
6.2.4.5 Accelerometer Interfacing 131
6.2.4.6 Thruster Interfacing 132

7. VEHICLE CONTROL 133

7.1 Equations of Motion 134
7.1.1 Euler's Theorem 135
7.1.2 ROV Equations of Motion 136
7.1.2.1 Translational equations of motion 137
7.1.2.2 Angular equations of motion 137
7.1.2.3 Attitude equations: 137
7.1.2.4 Position equations 138
7.1.2.5 Vector Notation 138
7.1.3 Derived Equations 139

7.2 Vehicle Control 140
7.2.1 Standard PID Controller 141
7.2.2 Fuzzy PID Controller 143
7.2.2.1 Fuzzy PID design 145

7.3 Aquasphere Control Results 148
7.3.1 Control Variable Input/Output 148
7.3.2 Aquasphere Depth Control 150
7.3.3 Aquasphere Heading Control 152
7.3.4 Aquasphere Pitch & Roll Control 153

8. CONCLUSIONS AND FUTURE WORK 154

8.1 Thesis Summary 154
8.1.1 Vehicle Design 154
8.1.2 Thruster Systems 154
8.1.3 Sensor Systems 154
8.1.4 Networking System 155
8.1.5 Control System 155

8.2 Summary of Results 155

8.3 Suggestions for the Future 156

9. REFERENCES AND BIBLIOGRAPHY **158**

9.1 Chapter 2 **158**

9.2 Chapter 3 **161**

9.3 Chapter 4 **161**

9.4 Chapter 5 **162**

9.5 Chapter 6 **163**

9.6 Chapter 7 **163**

9.7 Bibliography **165**

10. APPENDICES **166**

10.1 Solid Modeling Information **166**

10.1.1 Thruster Mounts Solid Modeling Information 166

10.1.2 Spider-Joint Solid Modeling Information 167

10.1.3 Internal Frame Supports Solid Modeling Information 168

10.1.4 Vertical Frame Supports Solid model Information 169

10.1.5 Network Box Solid Model Information 170

10.1.6 Sensor Box Solid Modeling Information 170

10.1.7 Thruster Box Solid Modeling Information 171

10.1.8 Vision Box Solid Modeling Information 172

10.1.9 Thruster Cowling Solid Modeling Information 172

10.1.10 Camera System Solid Modeling Information 173

10.1.11 Camera Platform Solid Modeling Information 174

10.1.12 Laser Canister Solid Modeling Information 174

10.1.13 Laser Mounts Solid Modeling Information 176

10.1.14 Manipulator Mount Solid Modeling Information 176

10.1.15 Internal Frame Mounts Solid Modeling Information 177

10.1.16 Internal Frame Solid Modeling Information 178

10.2 Aquasphere Assembly Details **182**

10.3 Simulink Thruster Model **191**

10.4 Power Drive Layout **202**

10.5 Microcontroller Data **203**

10.5.1 SCON: Serial Port Control Register 203
10.5.2 CCAPMn: PCA Modules Compare/Capture Registers 204
10.5.3 Programmable Counter Array (PCA) Overview 204

10.6 Network Function Calls 207

10.7 Transputer Network Interface Cards 208

Index of Figures.

Figure 1, Vehicle Referance Frame 2
Figure 2, A Search, Survey and Light Work ROV 8
Figure 3, The Jason Junior ROV 9
Figure 4, The Hyball ROV 10
Figure 5, The Aquacube Prototype Vehicle 19
Figure 6, The Aquacube 20
Figure 7, The Vehicle Sub-Frame Section 23
Figure 8, The Basic Sub-Frame Assembly 24
Figure 9, Thruster Mount 25
Figure 10, A diagram of the finished Thruster Mount 25
Figure 11, The Spider-Joint 26
Figure 12, The Aquasphere's Basic Frame 27
Figure 13, The Original Aquasphere Design Concept 27
Figure 14, Internal Frame within Aquasphere 29
Figure 15, Proposed Internal Configuration 30
Figure 16, The Networking Box Design 31
Figure 17, The Sensor Box 32
Figure 18, Thruster Control Box 33
Figure 19, The Thruster Cowling 34
Figure 20, The Pan-ant-tilt Mechanism 35
Figure 21, The Camera Platform 36
Figure 22, The Laser Canister Mounts 37
Figure 23, The Laser Frame Mounts 38
Figure 24, The Vision System 39
Figure 25, Manipulator Arm Platform 40
Figure 26, The Aquasphere Prototype Vehicle 41
Figure 27, The Aquasphere's Final External Frame 43
Figure 28, The Final Vehicle Frame 44
Figure 29, The Aquasphere Solid Model 45
Figure 30, The Assembled Vehicle 45
Figure 31, Vehicle Buoyancy 47
Figure 32, The Completed Aquasphere 49
Figure 33, The Mounted Thruster 51
Figure 34, Rotational Speed Vs Input Voltage 53

Figure 35, Thruster Torque Characteristics 54
Figure 36, Adjusted Load Characteristics 55
Figure 37, Motor Generated Back E.M.F. 56
Figure 38, 10 Volt Input Step Response 57
Figure 39, Thruster Current for 10volts input step 58
Figure 40, Thruster Step Response 59
Figure 41, The Open Channel Flume 60
Figure 42, Thruster Flow Tests 61
Figure 43, A Typical Step Response 62
Figure 44, Thruster Model Top Level 64
Figure 45, Thruster Demand Sub-system 64
Figure 46, The Power Drive 65
Figure 47, The Stiction Model 66
Figure 48, Motor Representation 66
Figure 49, Thrust vs Propellor Velocity^2 72
Figure 50, The Thruster Model 73
Figure 51, System Step Response for 15V Step Input 75
Figure 52, Model Step Response for 15V Step Input 75
Figure 53, Systems Current Step Response to 15V input 76
Figure 54, Models Current Step Response to 15V input 76
Figure 55, A Block Diagram of the P.W.M. Drive. 78
Figure 56, Triangular Waveform Generator 79
Figure 57, The P.W.M. Generation Circuit. 80
Figure 58, The P.W.M. Generation Board 81
Figure 59, The Gate Drive Circuit 82
Figure 60, Schematic Diagram of an H-bridge Drive 83
Figure 61, The H-Bridge Power Circuit 84
Figure 62, The P.W.M. Power Drive 85
Figure 63, The Thruster Power Box 86
Figure 64, Intel 87C5FX Block Diagram 89
Figure 65, The Microcontroller Test Board 90
Figure 66, PCA 8-bit PWM Mode 93
Figure 67, Duty Cycle with respect to CCAPnH 93
Figure 68, The EL7661 Bridge Driver 95
Figure 69, Power Drive Block Diagram 96

Figure 70, Navigation Using GPS 102
Figure 71, Autohelm Sensor Connection 115
Figure 72, NMEA Interface 116
Figure 73, The PC -Transputer Interface Card 120
Figure 74, The Data Display System 130
Figure 75, Rotation in order $\theta_1 + \theta_2$ 135
Figure 76, Rotation $\theta_2 + \theta_1$ 136
Figure 77, Conventional PID Control 141
Figure 78, Aquacube test result 142
Figure 79, Aquasphere Control Strategy 144
Figure 80, Membership functions for fuzzification 145
Figure 81, Weights for controller output defuzzification 146
Figure 82, Aquasphere's Depth Response 151
Figure 83, Thruster Demand 151
Figure 84, Aquasphere Auto-Heading Response 153
Figure 85, Possible Future Vehicle Design 157
Figure 86, Internal Frame Supports 168
Figure 87, Vertical Frame Supports 169
Figure 88, Additional Frame Supports 177
Figure 89, Internal Frame Support 1 179
Figure 90, Internal Frame Support 2 180
Figure 91, Internal Frame Support 3 181
Figure 92, The Basic Frame 184
Figure 93, Vertical Frame 1 Inserted 184
Figure 94, Vertical Frame 2 Inserted 185
Figure 95, Vertical Frame 3 Inserted 185
Figure 96, Laser Brackets Mounted 186
Figure 97, Networking Box Installed 186
Figure 98, Sensor Box Installed 187
Figure 99, Vision Box Installed 187
Figure 100, Thruster Box Installed 188
Figure 101, Manipulator Box Installed 188
Figure 102, Camera System Installed 189
Figure 103, Laser Rangefinders Installed 189
Figure 104, Thrusters Installed 190

Figure 105, The Manipulator Arm Installed 190
Figure 106, Motor Drive Track Layout 202
Figure 107, Power Board Component Layout 203
Figure 108, The Transputer Processor Card 209
Figure 109, The Analogue to Digital Input Card 209
Figure 110, The Digital to Analogue Output Card 210
Figure 111, The Link Adapter Card 210
Figure 112, The RS232/NMEA Interface Card 211

1. Introduction

Man's desire to explore the unknown, combined with the possibilities for financial reward, has lead to the development of tools which aid man's ability to study and interact with the underwater world. Technology has been used throughout the century to provide solutions to man's inability to explore these regions unaided. Diving operations have progressed from early pump driven breathing suits, to chemical rebreather units, to SCUBA equipment, and to current state of the art atmospheric diving equipment, with countless stages in between. Similarly the early screw driven bathyspheres have now been replaced with modern submersibles and submarine systems. Technology has managed to allow greater safety for subsea intervention, however, this is partially offset by the more complex and potentially dangerous tasks now carried out underwater.

Recently technology has lead to the development of Remotely Operated Vehicles (ROVs). These vehicles have been developed to provide inspection, monitoring and intervention capabilities for such tasks as the offshore oil industry, salvage, minehunting and research. ROVs can in part replace the requirement for manned intervention underwater, which gives obvious safety benefits in addition to increased endurance and economy.

There are three main methods by which underwater tasks are performed. Ambient pressure diving (in which the diver's body is subjected to the external water pressure) which is suitable for comparatively shallow depths (up to 300m), atmospheric diving (in which the diver is protected from external pressure by a reinforced suit) for greater depths, and remotely operated vehicles. Human divers can only work for short periods at significant depths using ambient pressure techniques because of problems due to oxygen toxicity and nitrogen narcosis. Use of helium as the main component of the breathing gases causes severe speech distortion which gives problems of communication between the diver and the diving control station. The health risks and costs associated with long decompression cycles, with the divers having to make prolonged stops during the diving and resurfacing operations, are also an important disadvantage for this approach. The equipment and support costs of atmospheric diving are very high, and the type of operation which can be performed is limited. Whilst this mode of diving removes the physiological problems associated with diving the diver is no longer able to make direct contact with the environment but instead has to use servo manipulators in order to affect repairs, recover equipment or make adjustments.

The Remotely Operated Vehicle (ROV) is deployed in situations where the costs or safety risks associated with human divers are too high. In practice, a small ROV is often deployed alongside ambient pressure divers for safety monitoring, and only performs an observational role. Other common tasks for small ROVs include object retrieval, cable recovery, inspection and surveying. In all these applications a prime requirement is for the ROV to be readily driven to the operating site and then held steady or moved along a simple trajectory. Position control and stabilisation of the ROV are crucial for its successful deployment in these tasks.

A free-swimming object has six-degrees of freedom : three of translation and three of rotation. Many vehicle designers effectively reduce the complexity of the control requirements by ballasting the vehicle so that the roll and pitch are self-correcting. Figure 1 shows the notation of ROV state vectors according to the Society of Naval Architects and Marine Engineers (SNAME)

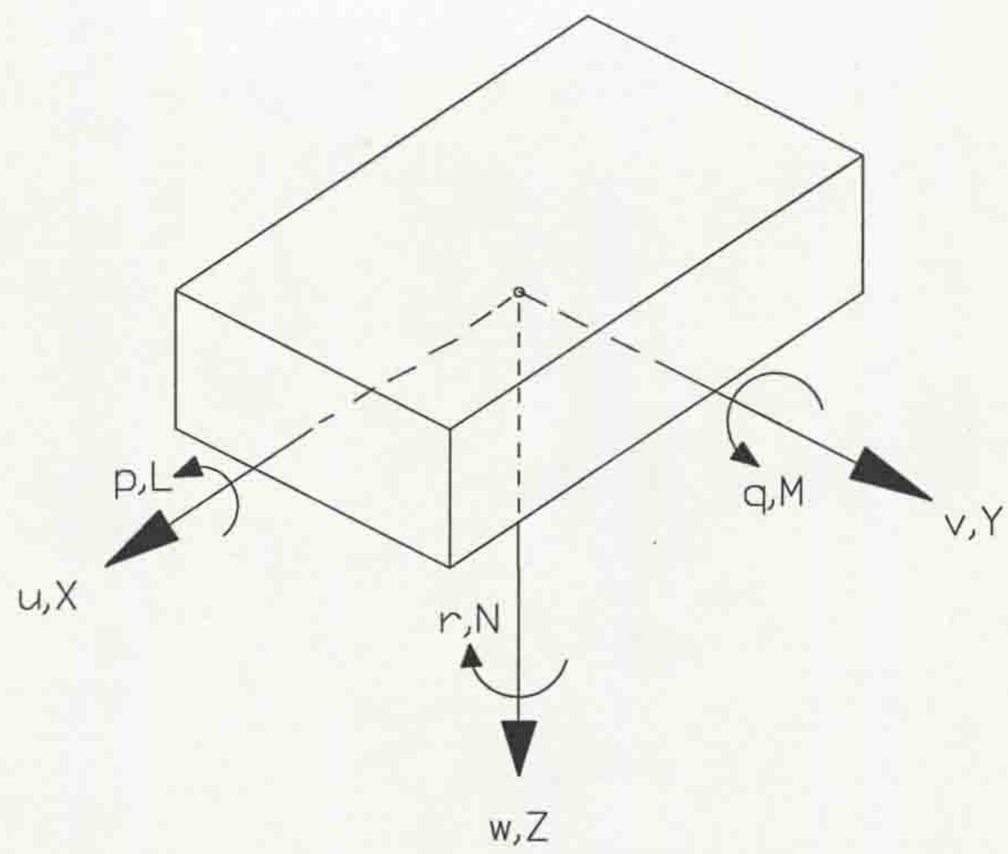

Figure 1, Vehicle Referance Frame

Where

X-axis translation	=	Surge.
Y-axis translation	=	Sway.
Z-axis translation	=	Heave.
X-axis rotation	=	Roll.
Y-axis rotation	=	Pitch.
Z-axis rotation	=	Yaw.

The main control problems are then those of depth holding (hovering) and bearing holding. It would be useful if an ROV could maintain a lateral position in spite of drift currents and umbilical drag. However, such a closed loop control scheme would require lateral absolute position sensing, which is difficult to achieve with present sensing technology.

The instrumentation required in order to enable feedback control is a minimum of one sensor per controlled degree of freedom. In many cases the accuracy and noise problems with measurement increases this requirement. Sensor fusion may be used to gain information from a multiplicity of sensors, helping to eliminate noise and other problems found in the sensing methods. For example the hovering task would use depth measurement near the surface, and draught measurement near the sea bed. All the sensors to be used must be robust, capable of marinisation and interfacing with the vehicle's networking system. Marinisation is the conversion of a system to withstand water ingress, salt corrosion and water pressure encountered in subsea systems.

1.1 The M.A.S.T. Project

The project was funded by the EEC under the MAST-0030-C project, M.A.S.T. standing for **M**arine **A**dvancement by **S**cience and **T**echnology. The full title for this project was "Combined Sensor and Information Technology for Subsea Positioning, Imaging and Control For Task Implementation." The principle objective of the project was to develop a ROV with enhanced capabilities, by the application of new technologies. Four main areas were identified as requiring technological enhancement, after the study of currently used vehicles. The project was split into four separate tasks to deal with each of the proposed areas of enhancement.

Task 1 : To design a vision system which would provide information to the rest of the vehicle's components from a series of high level commands. To achieve this subsystems for image capture, range-finding and low-level image processing would have to be built, and a high-level software shell created to process requests from the other vehicle subsystems.

Task 2 : To design and build a networking and data logging system involving hardware data capture, data transmission and data output. The networking system

should be able to interface to all the other subsystems and allow for free data communication between all subsystems

Task 3 : To design and build an electrically driven manipulator arm to undertake simple subsea operations.

Task 4 : To design and build an underwater vehicle capable of holding the other vehicle subsystems with a thruster and control system capable of maintaining the position of the vehicle in a stable manner. The system should be able to receive high-level commands such as "stay level", "maintain depth", "maintain heading" and "move 1m forwards" performing these requests automatically

The basis of the work carried out for this thesis was to achieve the objectives specified by Task 4 of the project. To achieve this a six degrees of freedom vehicle, the Aquasphere, was developed. To drive the vehicle, electronics were designed and constructed to power the thrusters mounted on the frame. Sensor systems were added and interfaced to be able to determine the position and attitude of the vehicle. A control system was created to provide positional control of the vehicle, reading in the sensor information and outputting demands to the thruster systems.

To demonstrate these technologies the final goal of the project was to allow an ROV pilot to give the vehicle a series of high-level commands allowing the vehicle to swim towards a target, maintain a steady hover position by the object, and to attach a cable to the object using the manipulator arm.

The design of the sub-systems was done in a modular style, with distributed processing methods. This was done so that the sub-systems could be tested individually, and the requirements on the networking system would be reduced. Also the intention was to provide systems which would allow the development of an A.U.V. (Autonomous Underwater Vehicle [Described in Chapter 2]) in the future.

The MAST-0030-C project was a collaborative project between The University of Liverpool (England), The University of Thrace (Greece), Stolt Comex Seaway (Marseilles, France), MOPU (Spain) and K.F.A. (Germany). The majority of the work was carried out at Liverpool, with low-level image processing being provided by Germany, ultrasonic sensors

provided by Thrace, top-level graphical programming by Spain, and testing/support facilities provided by Stolt Comex.

1.2 Thesis Outline

In addition to this section the thesis is divided into seven other principle sections, plus the references and appendices.

Chapter 2 Remotely Operated Vehicles describes the history of ROVs, their development from manned vehicles, and their place in industry today. It describes the limitations of conventional diving techniques and discusses the advantages of ROV use in certain situations. The chapter outlines typical applications for ROVs, and the difficulties present in carrying out these tasks. The chapter poses solutions to some of the difficulties encountered in ROV use by the use of emerging technologies. The control methods for ROVs are discussed, and the use of non-conventional control methods is discussed.

Chapter 3 Vehicle Design details the design of the underwater vehicle *Aquasphere* from the initial concept to the final vehicle. The intermediate design stages are followed, including the design of a one-axis prototype vehicle (the Aquacube), the initial prototype vehicle, and a final prototype construction. Detail is given behind the design of the vehicle, and why the changes had to be made from the initial concept vehicle. The requirements for a test vehicle are given, and the problems of trimming an underwater vehicle are discussed. Difficulties in the operation of the vehicle are highlighted, and design solutions are give to solve the problems experienced during the testing phase.

Chapter 4 Thruster Drive and Control describes the characteristics of the thrusters used on the Aquasphere. It describes how the thruster characteristics were determined from a combination of wet and dry testing methods, and describes the difficulties of controlling such characteristics. The stages for the power drive circuitry are presented, including the P.W.M. generation circuitry, and the power stage design. The marinisation strategy for the thrusters and the motor drive circuitry, and methods of improving upon this are given. The use of microcontrollers to minimize the design, and to give distributed processing is discussed, and the use of new power drive components in the design is given.

Chapter 5 Sensor Systems details the necessity for sensor systems for the effective control of an underwater vehicle. The type and specification of sensors ideally required for vehicle control is shown, and the specifications and limitations of the sensors used on the vehicle are given. The interfacing methods needed for the sensors is given, and the signal processing methods used for the sensors are also discussed.

Chapter 6 Networking Software gives the requirements for a networking system for the Aquasphere. The topology of the network is shown, and the development of the network protocols is shown. The networking of the sensor systems, and the thruster drives is detailed, and the linking of the network to the control systems P.C. is shown. The operating modes of the control system, particularly the master-slave control of the control system is detailed. The multitasking software for the control system is shown, and the necessity for this approach is given.

Chapter 7 Vehicle Control gives a description of the conventional control techniques used on the Aquasphere, and describes the fuzzy control techniques used on the vehicle. The method of implementation for the control methods used on the Aquasphere is shown, and a summary of the control methods used during is given. Problems experienced in the testing and control of the vehicle are shown, and solutions to the problems are described.

Chapter 8 Conclusions and Future Work gives a summary of the work carried out for this project. The most important results and achievements are highlighted, and the difficulties experienced during the project are given. A discussion of further work and the improvements currently in progress for the Aquasphere are given.

2. Remotely Operated Vehicles

With the current popularity in the field of underwater exploration and intervention there has been a corresponding increase in the research being carried out in this area. The task of constructing and testing an ROV is expensive, difficult and time consuming. These factors have caused the creation of relatively few research ROVs, with the majority of vehicles being utilized by commercial interests. The published work to date is principally produced from a select number of research institutions, or are theoretical works published on data from these institutions.

2.1 ROV Design

There is very little published information on the specific designs of ROVs. As the majority of ROVs are commercial vehicles this information is not generally available. Most vehicles used by research establishments are purchased from commercial companies, and so little design information is available from these sources.

The design of commercially available ROVs varies hugely. The expensive nature of underwater exploration and operation has led to a great deal of specialization in the design of commercial systems. Often several vehicles of varying complexity are used to carry out a complex task such as the maintenance and repair of an underwater pipeline. For example an inspection ROV may be deployed first to locate a fault, and from the information received from this vehicle a larger, more complex vehicle, may be deployed. There is no "standard design" amongst ROVs, although they can basically be fitted into two categories, the first being a torpedo shaped streamlined body, and the second being a rectangular shaped block, with buoyancy at the top and the heavier ROVs systems at the bottom. This second shape of ROV is the type most frequently found. The classification of ROVs as defined by the ROV Review [2.1] can be further subdivided as:-

1. Large and Medium Work and Support ROVs.
2. Inspection, Observation, Search, Survey and Light Work ROVs
3. Inspection, Observation, Search and Survey ROVs
4. Low-cost ROVs (Under £25,000)
5. Very Low-cost ROVs (Under £5,000)
6. Military ROV Systems
7. Custom and Special purpose ROV Systems

8. Autonomous (untethered) Underwater Vehicles (AUVs)

Fitting a ROV into any one of the above categories can prove difficult, as the boundaries of definition carry much overlap. However, category 1 ROVs are principally the very expensive and specialised vehicles used for pipe and cable laying. These ROVs would not be used for repair tasks until another cheaper ROV system had inspected the area.

Type 2 ROVs are small to medium sized vehicles with cameras and sensing systems such as sonar, pressure sensors, temperature sensors, flux-gate compasses and gyroscopes. In addition these vehicles have a rudimentary manipulator arrangement, normally only having two degrees of freedom. The most common configuration for the manipulator is a gripper containing a blade arrangement and a wrist rotation. This can then be used to cut cables that may be causing an obstruction to itself or to other vehicles.

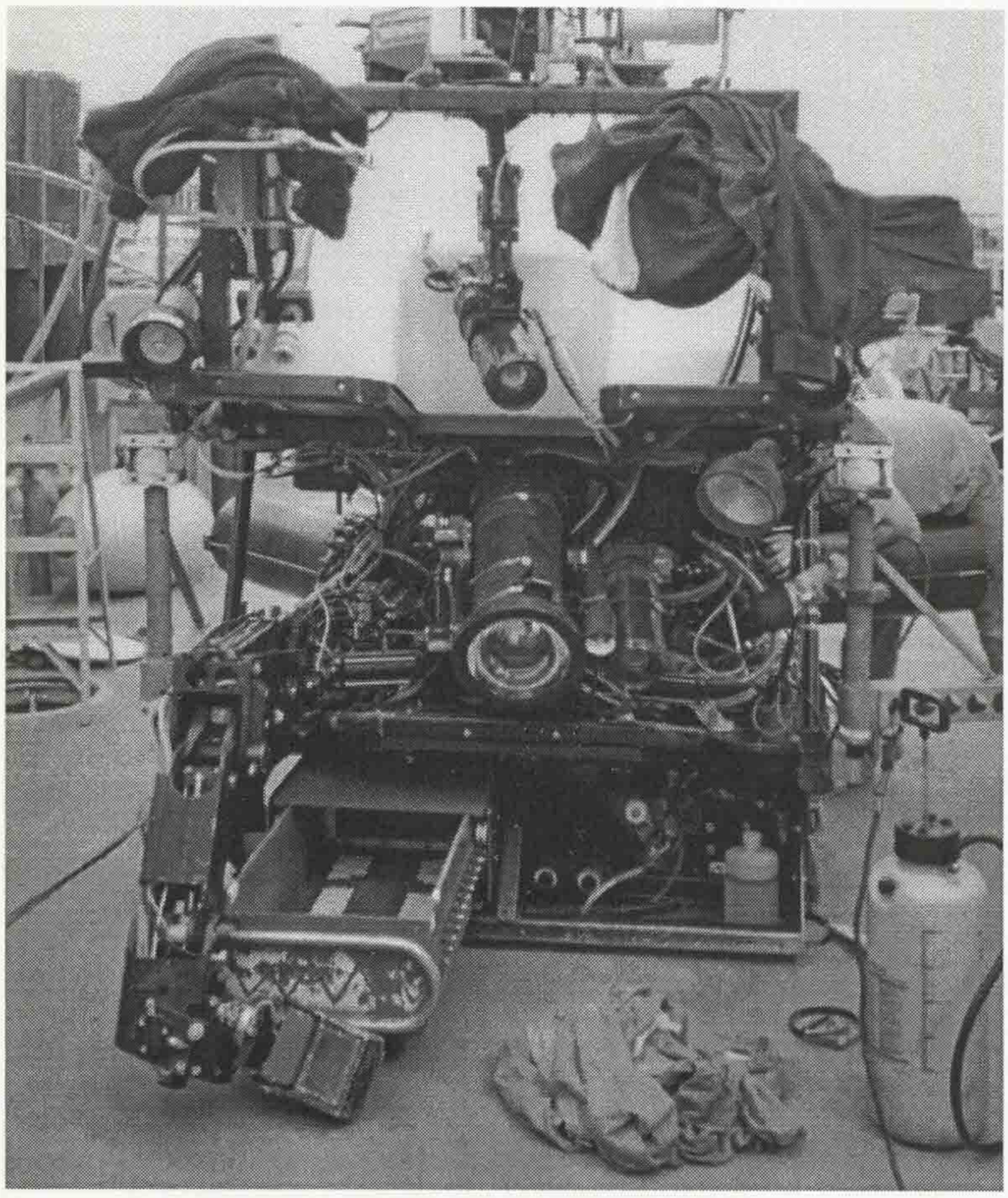

Figure 2, A Search, Survey and Light Work ROV

Figure 2, shows a search, survey and light work ROV, as used by the Monterey Bay Aquarium Foundation[2. 2] for show demonstrations, and research use.

Type 3 ROVs are basically flying camera and sensor platforms, with a sensor array such as that for type 2 ROVs but with no capacity for practical work. These vehicles may be used in conjunction with type 1 ROVs to provide the operator with another view of the work to be carried out. Figure 3 shows the Jason Junior (Newman [2.3]) inspecting the Titanic. The discovery of the Titanic in 1985, and the subsequent use of ROVs for its exploration, led to renewed interest and finance into the field of ROV research. One of the offshoots of this discovery was the creation of the much publicized JASON project, which undertakes research into ROVs, and provides teaching projects and liaison with schools around America.

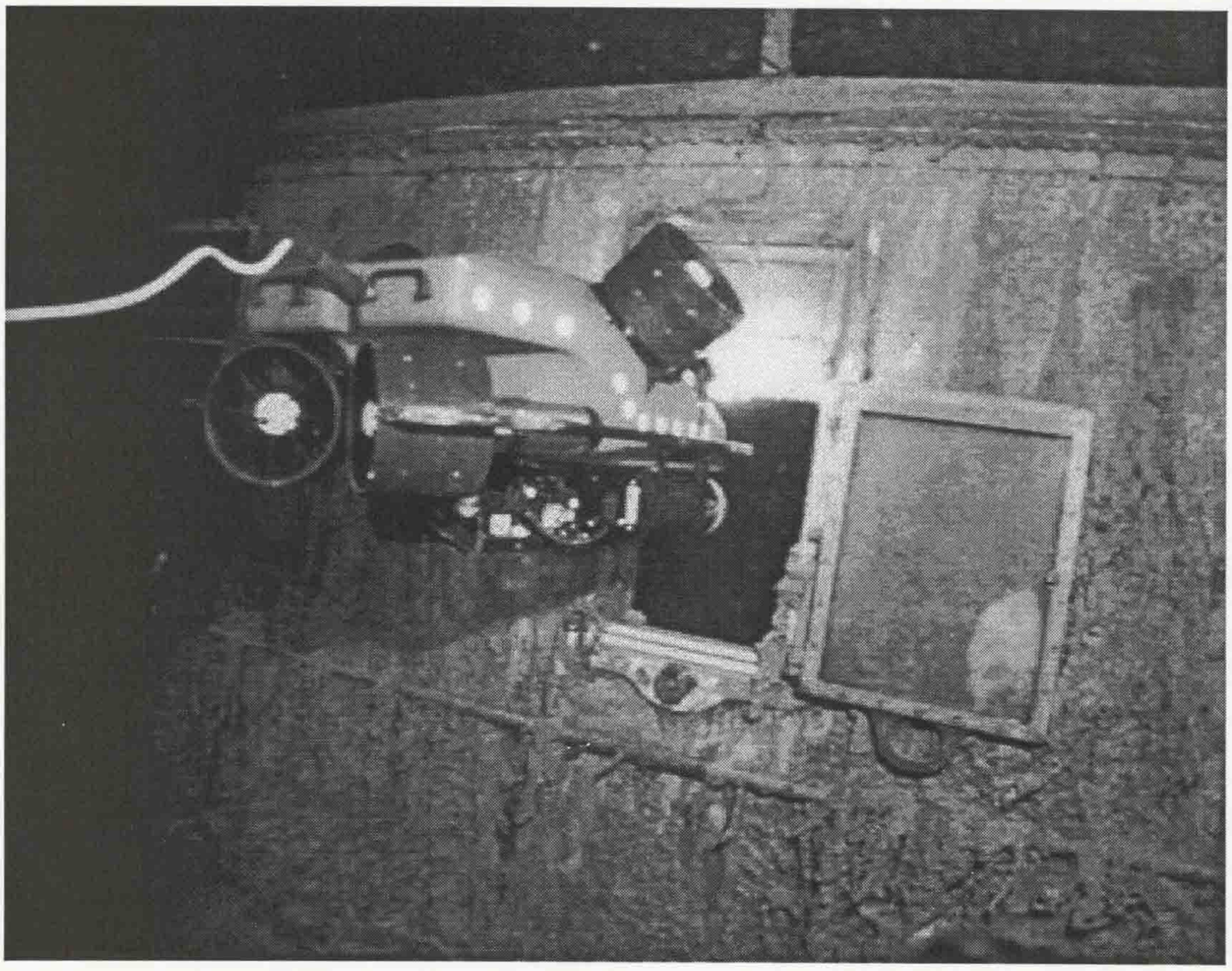

Figure 3, The Jason Junior ROV

Type 4 ROVs are principally for visual observation only, and lack the sophisticated sensor systems such as sonar that more expensive vehicles may have. These vehicles normally have a pressure sensor to indicate depth, and a compass or gyroscope to indicate heading, in addition to one or two cameras. Different cameras may be used for low-light conditions, and often a still's camera may be mounted to provide a permanent visual record.

Figure 4 shows the *Hyball* ROV manufactured by Hydrovision[2.4]. The vehicle has a camera mounted centrally on a rotating turntable which gives the vehicle 360° of vision. Lights mounted on the turntable, and on the front of the vehicle provide illumination. The vehicle's sensor systems include a gyroscope, a flux-gate compass and a 30bar pressure sensor. The vehicle is controlled from a top-station which gives sensor readings as a graphical overlay to the camera picture. Control is carried out using a three-axis joystick (a standard joystick configuration with an extra top rotation) connected to the top-station. The vehicle is powered by four thrusters and weighs approximately 50kg.

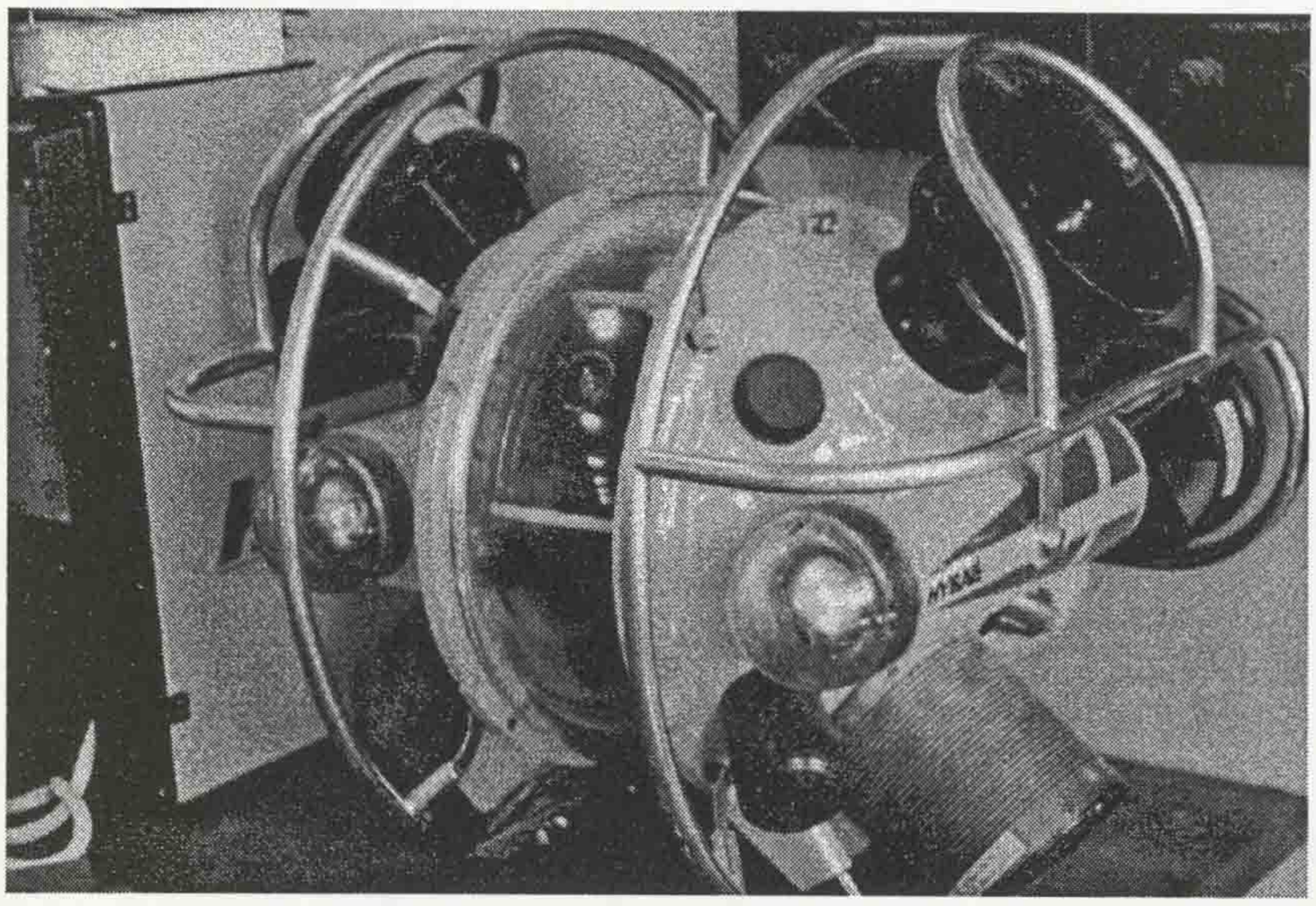

Figure 4, The Hyball ROV

Type 5 ROVs are a very small section of the market. These vehicles are not of the very high robustness and reliability that industrial use of an ROV demands, and are not used for any off-shore or time-critical work. The main market for these vehicles has been for diving clubs, small scale marine-life monitoring and for pool and tank inspection. It is only recently that the price of such vehicles has dropped to allow them to come into the "amateur" market.

Type 6 ROVs are a large range of very highly specialised vehicles to carry out military tasks. These vehicles are typically used to work in hazardous environments such as minefields (Hyland [2.5]). The vehicles designed for such a task have a very low magnetic signature, and a very low noise emission level. Many other types of military ROVs exist for a large range of diverse tasks; however the specifications and specialty of these vehicles is often classified information.

Type 7 ROVs are possibly the hardest to define by type, but can be described as special or custom vehicles which are used to carry out an unusual or non-standard tasks, and which does not fall into the other categories.

Type 8 ROVs are presently only research vessels, and are not really suitable for constructive tasks, although Ageev[2.6] has proposed that they may be used for search and survey tasks in the future. Tonshoff[2.7] also proposes that with advances in sensor technology "the automatic execution of underwater tasks like cleaning, inspecting, cutting and welding at production platforms or gas and oil pipelines" may be achieved. Zorpette[2.8] states that "the few built so far have achieved what might be the undersea equivalent of strolls in the park". Within five or ten years developers expect that future generations of these machines will be "earning their keep in such fields as marine science, ecology, the offshore-oil industry, and naval operations"(Zorpette[2.8]). At present these tasks are beyond the AUV, and the principle role of the AUV is to test automatic navigation techniques. Currently the vehicles are typically lightweight, low powered, torpedo shaped vessels with a single propeller propulsion system, being steered by movable fins. An additional problem for AUVs is that of powering them. As the vehicle has no umbilical there is no provision of power from a remote source as in standard ROV. The development of lighter, more reliable, safer and higher powered electrical sources for use in innumerable applications is constantly been undertaken. The constraints placed upon the power supply for an AUV are given by Rosenfeld *et al* [2.9]. The principle constraint is that of operating in a constrained environment, where the production of explosive or corrosive gases by conventional power sources is undesirable. Rosenfeld *et al* [2.9] propose the use of fuel cell technology to overcome this problem.

2.2 Vehicle Control Problems

The behavior of an underwater vehicle is a complex non-linear time-varying multivariable system. The ROV is subjected to external forces such as sea current, turbulence when near subsea structures, varying buoyancy and cable drag effects. In addition the non-linear characteristics of thrusters, motion of a manipulator arm, and the change in vehicle buoyancy by sample acquisition all go to make up this highly non-linear system.

2.2.1 Vehicle Modeling

The modeling of such a system is complex, expensive and time consuming. The standard method of testing requires the use of a Planar Motion Mechanism (PMM) (Dandi and Every[2.10], Yoerger *et al* [2.11], Goheen and Jefferys[2.12], Qiao[2.13]) which involves forcing the vehicle through the water in a series of tests to determine the equations of motion for the vehicle. Following these tests considerable effort is required to extract the equations from the test information. The addition, removal or reconfiguration of any of the ROVs systems requires the results to be retaken, as the model becomes invalid.

Goheen and Jefferys [2.12] propose a method of extrapolating the characteristics of a vehicle from data gathered from the operation of the vehicle in normal use. The models created by such a system would be considerably cheaper, and the data more readily available than the conventional techniques. However, the sensor data needed to complete this model needs to be accurate and comprehensive. Not all ROVs, particularly the smaller less expensive ones, have the required sensor data to formulate such a model.

2.2.2 Cable Drag

The problem of modeling the cable forces is a complex task in itself. The drag imparted on the vehicle by pulling a long cable through the water, which is itself being subjected to forces of turbulence and water flow can use a significant portion of the vehicles overall power. Most texts modeling the dynamics of the vehicle[2.10][2.11][2.12][2.13] conveniently ignore the effects the cable has on the vehicle, and consider is as an "external force". Trahan *et al* [2.14] and Dand *et al* [2.10] have proposed computational techniques for estimating the drag on a cable system. The drag effects are difficult to model, and are effected by external considerations such as the water flow profile of the area, and the effects of abrasion on the cable causing additional drag. Some commercial systems now use force measurement sensors to measure the actual force of the cable/vehicle interaction directly, without having to rely on computer estimates.

2.2.3 Manipulator Arm Motion

The motion of a manipulator arm mounted to an ROV can also cause disturbance to the system. The drag caused by the motion of the arm through the water can impart turning moments on the vehicle, in addition to the effects of the change in position of mass of the manipulator. Dunnigan and Lane[2.15] have built on work carried out by Castro[2.16] to

propose a method for modeling the effects of motion on a rudimentary robotic arm. The addition of such a model to the control system allows for compensation to be made by the control system for the arm's motion. The principle used to model the arm [2.15][2.16] is the added mass principle, in which the drag coefficient of the arm is modeled as the force needed to cause motion to a calculated mass. This allows standard equations of motion to be used to estimate the drag, without having to perform complicated hydrodynamic calculations.

2.2.4 Thruster Systems

The behavior of the thruster systems has a direct effect on the performance of the vehicle as a whole. Yoerger *et al* [2.11] states that "the dynamics of the thruster dominate behavior of the vehicle". Therefore, it is important to have an model of the thrusters to incorporate into the overall system controller. The principle work in this field has been carried out by Yoerger *et al* [2.11], whose methods have been implemented by Yuh and Gonugunta[2.17], Fossen and Sagatun[2.18] amongst others. Yoerger *et al's* [2.11] paper provides a method for calculating the characteristics of a thruster from a series of experimental test results. The many variables needed to calculate the thrust coefficient empirically are reduced to lumped parameters which can be obtained from the tests made on the thruster. Fossen and Sagatun [2.18] propose an additional method for derivation of a thruster model using slightly different techniques to those given by Yoerger *et al* [2.11]

The effects of marinisation of the thrusters causes a high degree of stiction (coulombic friction) on the thruster systems. Yang and Tomizuka[2.19] have devised a method for overcoming this stiction through the use of an adaptive pulse width modulation technique. Using this technique it is possible to compensate for the non-linearities in the thruster system.

2.2.5 Sensor Systems

For the possibility of closed loop control on an underwater vehicle the use of accurate sensor systems is essential. The determination for the vehicle's absolute position is difficult, but is also the most desired parameter. Langrock[2.20] details the methods by which exact position can be calculated for a vehicle within the vicinity of a rectangular array of high frequency transponders. Techniques for eliminating errors from reflections by combining the transponder information with a dead reckoning model, and a world model are used to improve system accuracy. This method has the drawback that the transponders must

be precisely located previous to the navigation of the vehicle. To overcome this problem Goodchild[2.21] uses a system which has the transponders fixed to buoys on the surface. A GPS system in the buoys determine their location which is passed to the vehicle via modulation of the sonar pulse from the buoy. With information from four buoys the vehicle is capable of determining its exact location. The advantage of this system [2.21] over conventional transponder techniques [2.20] is that the transponder buoys are easily installed and retrieved, without the need for positional accuracy, compared to underwater transponder types. Kwak *et al* [2.22] suggests that the use of buoys could be avoided by fixing the GPS system to the vehicle itself. The GPS system would be used to update the inertial guidance system on a vehicle until the vehicle submerged over the area of interest. An internal dead reckoning method would be used to guide the vehicle until the vehicle returned to the surface, and recalibrating the estimated position using the GPS. This method would be suitable for short missions, however Goodchild's[2.23] or Langrock's [2.22] system would be more suitable for prolonged use.

Flux gate compasses are frequently used to provide heading information on underwater vehicles. However, pitch, yaw and acceleration forces can all introduce inaccuracies into the sensor. Hellard[2.23] provides solutions to these inaccuracies by combining pitch and yaw compensation with a three axis solid state sensor. Hellard[2.23] also details the difficulties of mounting flux gate devices, and proposes methods for reducing the magnetic disturbances effecting the sensor. Crocker[2.24] uses a similar technique in his design of a motion tolerant compass system. Weiss[2.25] suggests the use of a gyroscope system to supplement the behavior of a flux-gate compass, although does not recommend the use of the sensor on its own for orientation measurement, due to drifts in the system.

To help increase accuracy and reduce sensor uncertainty and noise sensor fusion techniques given by Rigaud *et al* [2.26] , and Tonshoff[2 7] can be used. These methods process data from a multitude of sensors and produce a best fit estimate of results from the available data. This data is then given to the control system which can act upon this estimate instead of the actual sensor results. In this way inaccuracies and non-linearities of sensors can be overcome, giving sufficient sensorial input to the sensor fusion system.

In the future three-dimensional acoustically based sensor systems proposed by Hansen [2.27] and Wylie[2.28] may be used to aid the automatic and manual control of an ROV. These systems use multiple sensor arrays to detect an acoustic return pulse and

formulate an acoustic image from this return. These systems are in their infancy, and are not currently available for purchase.

2.3 Vehicle Control Methods

The problems in controlling an underwater vehicle due to their non-linear behavior has led to a large amount of research into solving this problem. With the development of new control methods such as fuzzy logic, neural networks, expert systems and variable structure system sliding mode control, the methods now used to control ROVs are wide and varied. At present there has been no clear winner in the intellectual battle to find the most suitable control technique.

2.3.1 PID Control

Currently a simple PID (Proportional Integral Derivative) controller is most commonly used to control commercial vehicle systems The properties of this controller are widely used, giving robust control when accurately tuned. The integral term of the controller reduces the steady state error of the system, whilst the derivative term helps to regulate the system's overshoot. The proportional term provides an overall controller gain, if required. However, a standard PID controller has problems when coping with an non-linear system. Astrom *et al* [2.29] highlights the difficulties of using PID controllers, and proposes features which could be incorporated into future generations of PID controller. The features he proposes are based on analyzing the response of the system, and applying tuning criteria to the control. Conventional PID controllers become less accurate if the dynamics of the plant changes. To counter this Yanagawa and Miki[2.30] have used a PID controller in conjunction with a neural network to produce a self tuning PID controller for simple motor control. Also, Miki *et al* [2.31] have applied the use of fuzzy logic to improve the performance of a PI motor controller.

2.3.2 Fuzzy Logic

The ability of fuzzy logic to deal with non-linear problems in the real world has led to the use of its techniques on underwater vehicles. Fuzzy logic was formulated by Zadeh[2.32] in 1965. The benefit of fuzzy logic control for the control of a non-linear system

is its ability to deal with linguistic variables. These variables are normally formed from "common sense" or "rule of thumb" knowledge of the system. For example to park a car in a tight parking space the commands "back a bit", "left a bit", "go forwards", "go slowly", "stop" may be given to the driver to aid their control of the car. These type of non-empirical commands are known as the linguistic variables, and the interpretation of these variables is the basis of fuzzy logic. Fuzzy logic's ability to use linguistic variables for the control of a poorly defined system make it suitable for the control of a complex system such as an underwater vehicle, where the model may not be clearly defined. Several papers and articles introducing the concept of fuzzy logic control have been written (Dettmer [2.33], MacLeod [2.34], Myers[2 35], Cox [2.36], Kohn[2.37], Klir[2.38]) in addition to the early papers by Zadeh [2.32][2.39]

The use of fuzzy logic has been applied to underwater vehicle use by Farbrother *et al*[2.40], Stephens[2.41], Stephens and Shimmin[2.42], Shimmin and Lucas [2.43], Shimmin *et al* [2.44] and Smith[2.45]. The use of these controllers has given experimenters control results comparable to those found for a tuned system with known plant equations. The addition of adaptive mechanisms to the fuzzy system maintains the control accuracy over a range of operating conditions.

2.3.3 Adaptive Control Techniques

The operating conditions in which underwater vehicles are used causes disturbances to the vehicle which alter the dynamics of the vehicle's control. For example the force of the umbilical can increase with the vehicles depth, and the cable length. These disturbances can lead to the degradation of performance of a fixed control strategy. To over come this the use of adaptive systems has been proposed.

To adapt a system there must be a method for comparing the output of the system against a known variable. The comparison model is normally a derived model of the system from experimental test results, or parameter estimation. This system model is then used to adapt the control of the plant to mimic the behavior of the model. This method has been used by Goheen and Jefferys[2.12] on ROV systems. Astrom and Wittenmark [2.46] propose an adaptive system to reduce the system errors. However, without a fixed reference model the system can become unstable if the adaptive system changes undesirably. or components change.

The underlying control system for the adaptive methods varies. Yuh and Lakshmi[2.47], Yuh and Gonnugunta[2.17], Stover and Gibson[2.48] have used an adaptive neural network control system for controlling the operation of an ROV. Although results of the research have shown promise, work carried out thus far (by [2.47] and [2.17]) has been on model simulations, and not on a physical system. Neural networks function by mimicking the neural pathways in the brain. The networks are trained by repetitive exposure to the test data which modifies the gains of the pathways within the network. A disadvantage of this system is that it is difficult to incorporate knowledge into the network from the start. When testing a physical system it is possible that damage could be caused to the vehicle whilst training is being undertaken. A solution to this is to use a supervisory expert system to prevent the training system from carrying out any potentially dangerous maneuvers. The use of an expert system supervisor has successfully been used by Stephens[2.41], and Stephens and Shimmin[2.42] for control system training on the Aquacube.

Schultz *et al* [2.49] have implemented a genetic algorithm to search for the optimum control parameter. The algorithm works by making changes in the control coding, and preserving changes which led to a benefit in the overall vehicle control. This technique relies on a model for the vehicle to achieve training, and is currently not-suitable for online use.

Variable Structure Sliding (VSS) control techniques were originally devised by Utkin[2.50] and have since been refined by other experts in this field such as Itkis[2.51] and Zinober[2.52]. This technique has been applied to use on underwater vehicles by Cunha *et al* [2.53], Olgac et al [2.54] and Fossen and Sagatun[2.18]. These controllers force the behavior of the system to follow that of the model by high-speed non-linear switching of the controller. For successful use in a ROV the system requires thrusters capable of high speed thrust variation. Simulation results on ROVs using this technique have been carried out by [2.52][2.53][2.54], giving successful results. However, results by Stephens[2.41] has shown limitations with this technique due to oscillation caused by the high frequency switching methods.

3. Vehicle Design

As detailed in chapter 2 the design of commercially available vehicles varies greatly. The vehicles are usually application specific, and their design reflects the specific task for which the ROV is used. Many conventional designs also have limitations in operation due to their specialization, and are unable to carry out a operations for which they are not specifically designed. A desire to create a vehicle which was capable of a wide range of tasks, by utilising new technologies, lead the design of the *Aquasphere*.

3.1 The Aquacube

Before the design and construction of a major test vehicle was undertaken a simple test-bed vehicle was designed. The purpose of this vehicle was to provide a test unit for the thrusters, their drive systems, the sensors and the control system. Although this was a much simpler vehicle, having only one degree of freedom, a great deal of work was carried out by Swainston[3.1], Stephens[3.2] and Kodogiannis[3.3] to try out control techniques and modeling methods on the vehicle. The experience gained in determining the hydrodynamic equations for the vehicle could then be applied to the major vehicle. The advantage of the Aquacube for this work was that its small size meant that it could be easily transported and quickly made operational. In addition, the vehicle could be used in a much smaller tank or pool to be tested, such as the University's swimming pool. This meant that a large number of tests could be carried out on the vehicle, more than would be able to be carried out on a full size ROV.

3.1.1 Design of the Aquacube

The full information for the design of the Aquacube is detailed by Swainston[3.1], and is included here to show the natural progression of the vehicle's design. The design of the vehicle is shown in Figure 5.

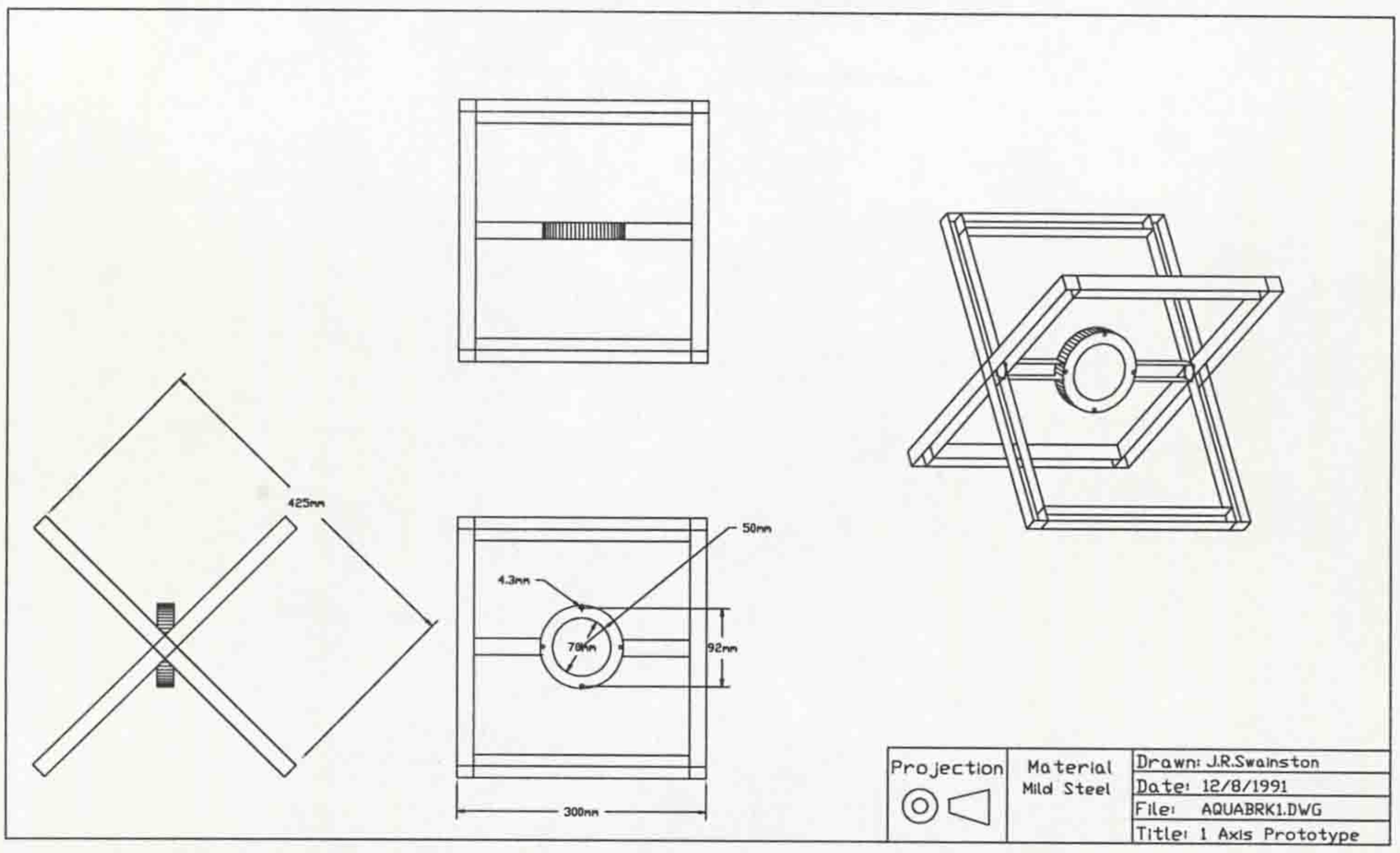

Figure 5, The Aquacube Prototype Vehicle

The Aquacube (Figure 5) is basically a frame within which a thruster is mounted vertically. The frame itself is constructed of mild steel 25mm square section tubing, with a central circular mount for the thruster. Buoyancy for the Aquacube consisted of plastic buoyancy material, and sponge floatation material held on by plastic ties. The vehicle was trimmed by the addition of lead weights to the base of the vehicle's frame. A pressure sensor was mounted on the base of the vehicle to provide information on the vertical position of the vehicle. Two cables, one to power the thruster and one connected to the pressure sensor were attached to the vehicle, returning to the surface. To reduce the effects of the cable weight plastic "eggs" of buoyancy material were fixed to the cable at intervals to provide a neutrally buoyant cable.

A picture of the assembled one-axis prototype, the Aquacube, is shown in Figure 6.

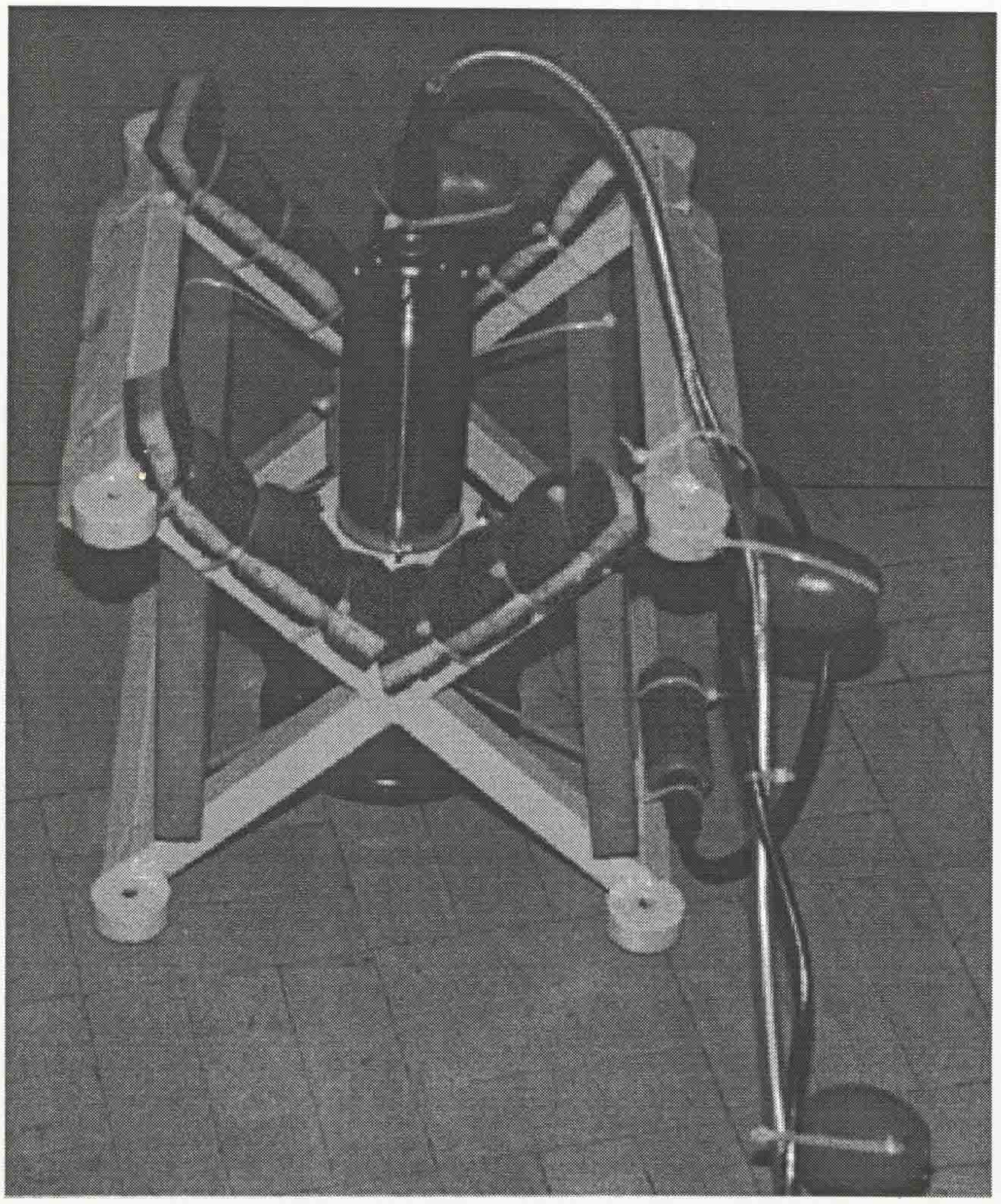

Figure 6, The Aquacube

3.2 Original Vehicle Concept and Design for the Aquasphere

The basic definitions and specifications for existing vehicles were examined before starting the design of a new ROV, and several commercial and research vehicles were studied. It became apparent that the level of technology used for these vehicles was low, and that great benefits could be had by using modern control and processing techniques on these vehicles. The design of the vehicle was carried out so as to create a test bed for these novel technologies. Originally the concept of the vehicle was dominated by the desire to test out new vehicle control methods, and the design of the vehicle was carried out with this purpose specifically in mind. However, later in the project the limitations imposed on the vehicle to provide a test bed for other technologies such as the robotic arm, the vision system and the networking system radically altered the design of the vehicle from that originally envisaged. This chapter details the evolution of the design from the original concept to the finally realized vehicle.

The original idea behind the design of the *Aquasphere* was to create a vehicle which had the capability of full six degrees of freedom movement. Having the full six-degrees of freedom would allow the control system to be able to direct and stabilize the vehicle in any plane and orientation.

It was envisaged that a vehicle could be created which had no "front or back" in the conventional sense. Instead the vehicle would be able to rotate and translate freely about it's axes, enabling the vehicle to carry out tasks upon a surface at any angle. To do this the center of mass would need to be as close as possible to the center of the vehicle. In addition the center of effect for the buoyancy would have to act at the same point of the center of mass, leading to an inherently unstable vehicle. A conventional ROV has a manipulator arm with a restricted field of movement. This means that often it is impossible to use a manipulator arm to carry out tasks because of the limited working envelope[1] within a plane. The vehicle was designed with the ability to fully rotate, bringing the working area within the working envelope of a manipulator. This would then enable a vehicle configured in the same manner to work on, for example, both the underside of a ships' hull and on the top and sides of a pipeline assembly. Currently this type of versatility is unique, and commercial vehicles have either to be reconfigured for such different tasks, or different types of ROV have to be used for each varied task.

To minimize the effects of cross-correlation between movement along different axes it was decided to make the vehicle symmetrical about each axis, thus creating a sphere. Also, by doing this the control strategy would be greatly simplified, as one set of the equations of motion for the vehicle could be derived and applied to each translation. Likewise only one set of equations need be derived which could then be applied to each of the three rotations. The spherical shape would help minimize translational and rotational drag coefficients, allowing rotation and translation of the vehicle to be carried out with the minimum of power expenditure.

The original concept of the spherical vehicle was thus conceived. Without the facilities to create a totally spherical vehicle at the University, and due to the prohibitive cost of construction elsewhere it was decided to build an external exoskeletal frame to house a smaller internal sphere. The frame would provide mounting for the vehicles thrusters, and

[1] A robotic arm's working envelope is the volume within which the arm's end-effector can move.

support the internal spherical shell containing the rest of the vehicles electronics. To give the six degrees of freedom it was necessary to have six thrusters mounted in three mutually perpendicular pairs. Each pair when operated in conjunction would give a translation along an axis, and when operated in opposition would give a rotation about the perpendicular axis.

With these criteria in mind the original prototype of the *Aquasphere* was designed. The dimensions of the outside of the vehicle were set to 1.3m, giving sufficient internal space for the electronic subsystems, and being of a manageable size to be able to be moved to the testing facilities. The components of the vehicle, whilst creating a complex symmetrical shape, are all simple and identical. This enabled the vehicle to be constructed relatively easily, and using the Departments in-house facilities.

The vehicle frame is constructed from six identical sub-frames connected together by eight "spider-joints". Attached to each of these sub-frames is a mounting bracket, giving the six mountings needed for the thrusters, in addition to maintaining symmetry. The unusual design gives an almost spherical shape, and due to its triangular sub divisions gives a design with great structural strength. Ideally the vehicle's frame should have been constructed of a corrosion resistant material such as duralium alloy, stainless steel or high titanium content steel. However, due to financial restraints and the inherent difficulties in welding such alloys, the vehicle frame was constructed from "speed frame". "Speed frame" is mild steel square section tubing with a tough plastic coating. The coating gives a measure of corrosion resistance, and being mild steel the tubing is easy to cut and weld. After construction all welds were treated with rust resistant paint, before the whole vehicle was painted with "Hammerite" paint.

3.2.1 Basic Sub-Member Design

The design of the Aquasphere was based around a simple modular component. Figure 7 shows a drawing for the two basic members from which the vehicle is constructed.

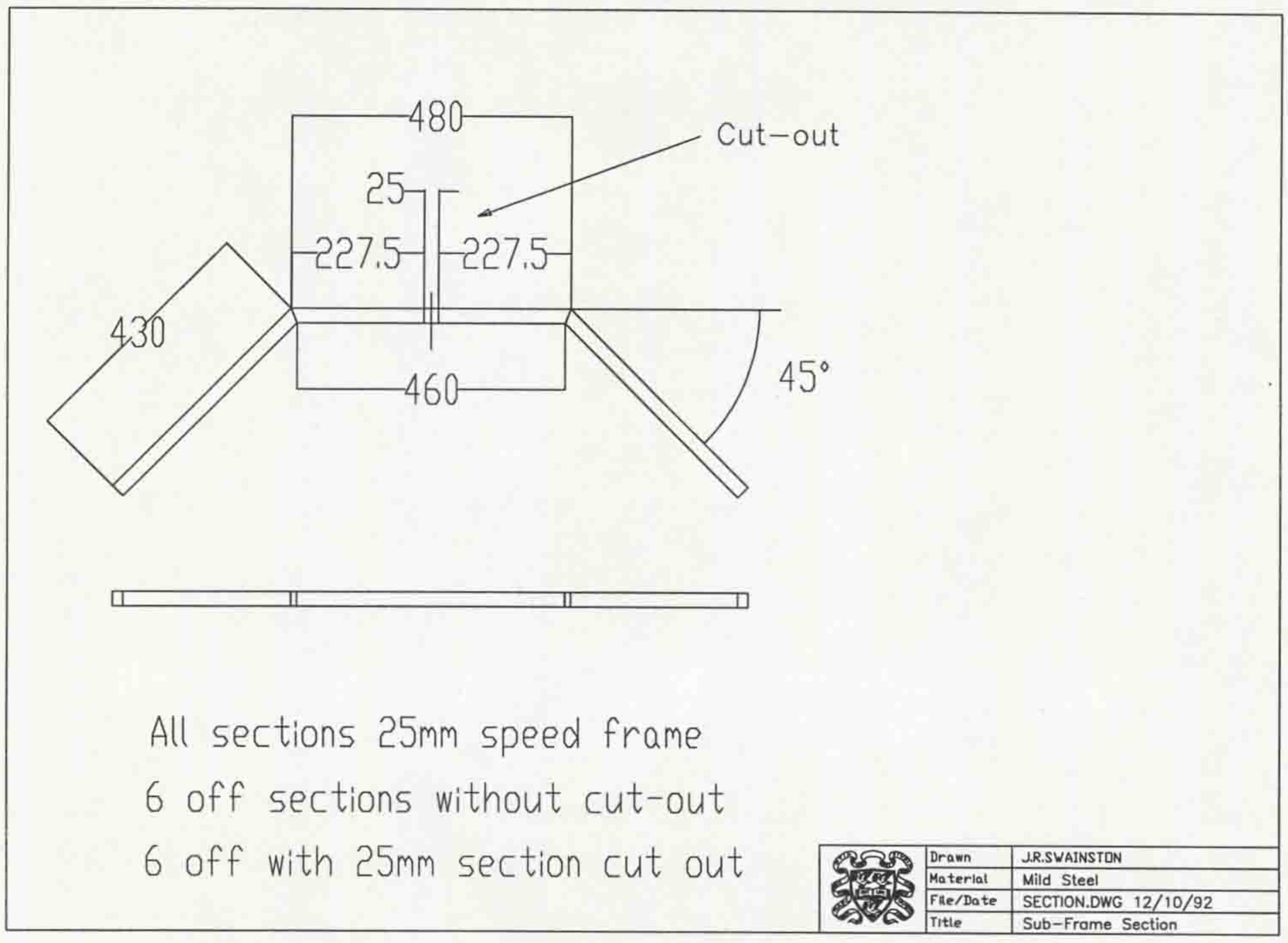

Figure 7, The Vehicle Sub-Frame Section

These two types of basic member detailed in Figure 7 (those with a section cut-out, and those without a section cut-out) are then welded together at right angles to form a sub-frame unit. Figure 8 shows a diagram of the two basic members welded together, forming one of the six sub-frames.

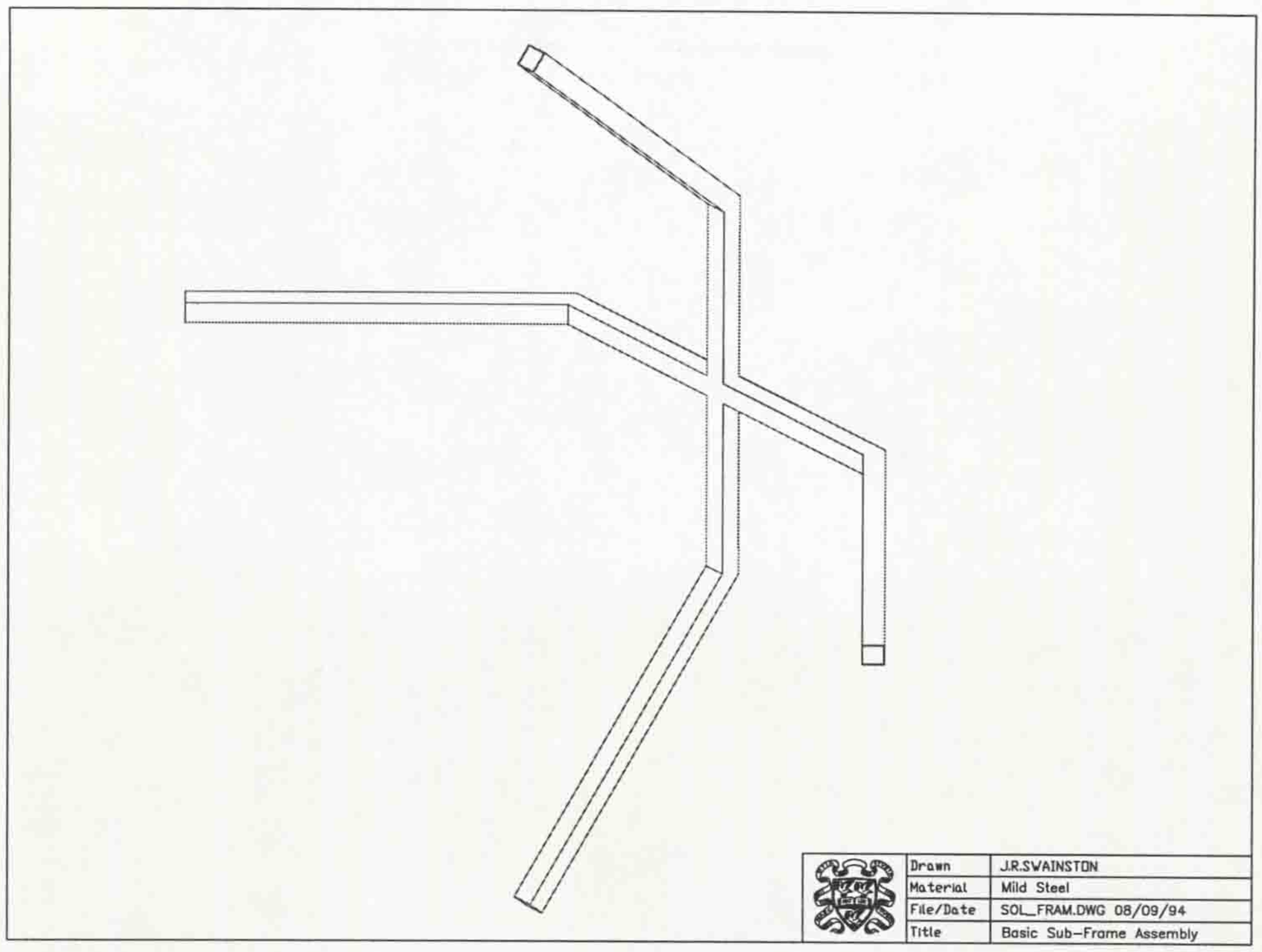

Figure 8, The Basic Sub-Frame Assembly

3.2.2 Thruster Mounting Design

Once the sub-frames had been assembled the mounts for the thrusters could be constructed and fixed to each of the six sub-frame assemblies. Figure 9 shows the technical drawing for the construction of the thruster mount. The original design included o-ring seals for the thruster mounts. At a later stage this was modified so that the o-ring seals were made redundant. Once the thruster mounts were manufactured they were welded to a 100mm x 100mm x 10mm mild steel plate to give a larger area of contact between the thruster mounts and the sub-frames. Figure 10 shows a diagram of the finished thruster mounting, after the back-plate has been welded on to it. These thruster mounts were then welded onto each of the "cutout" member of the sub-frames.

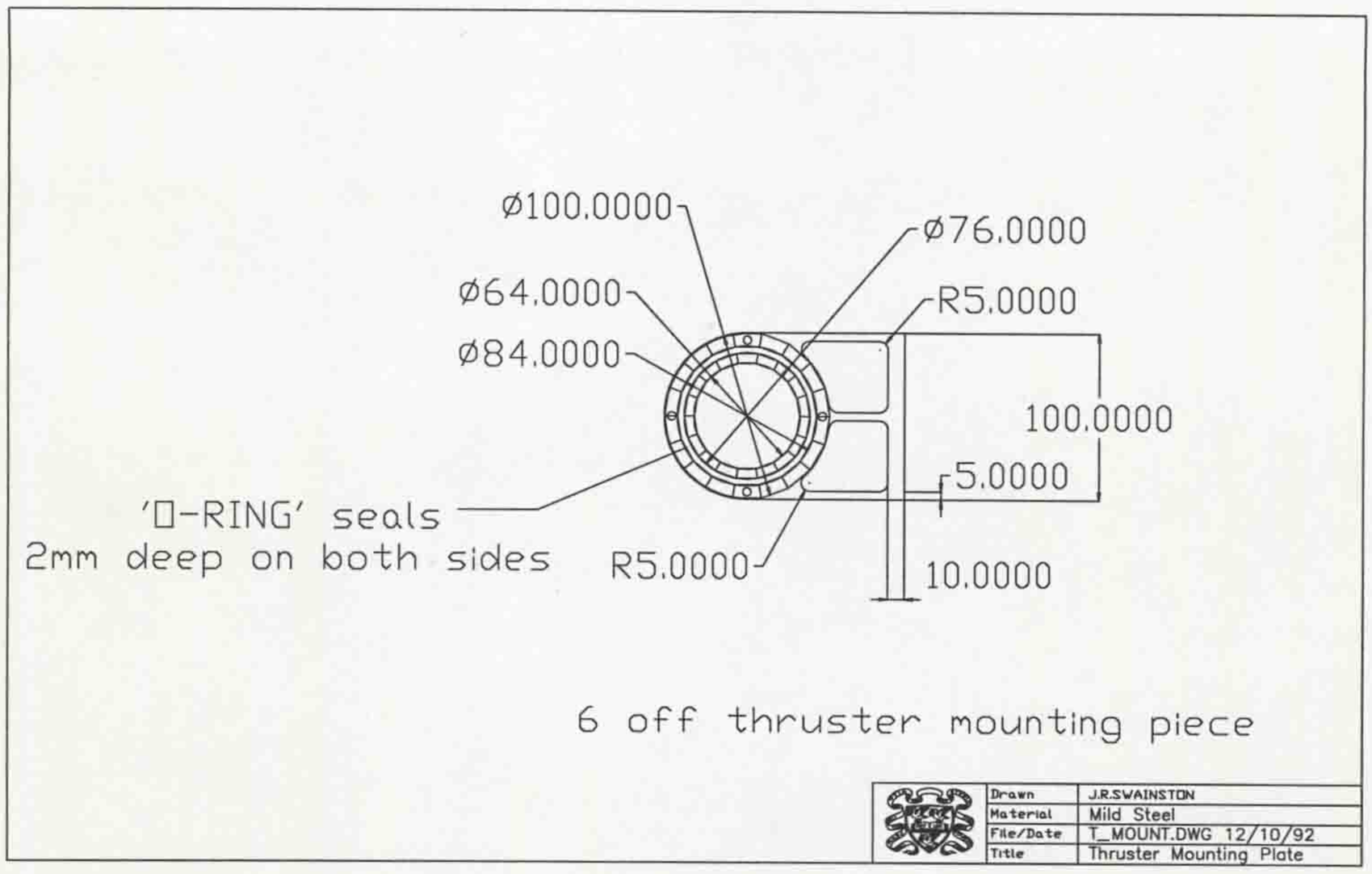

Figure 9, Thruster Mount

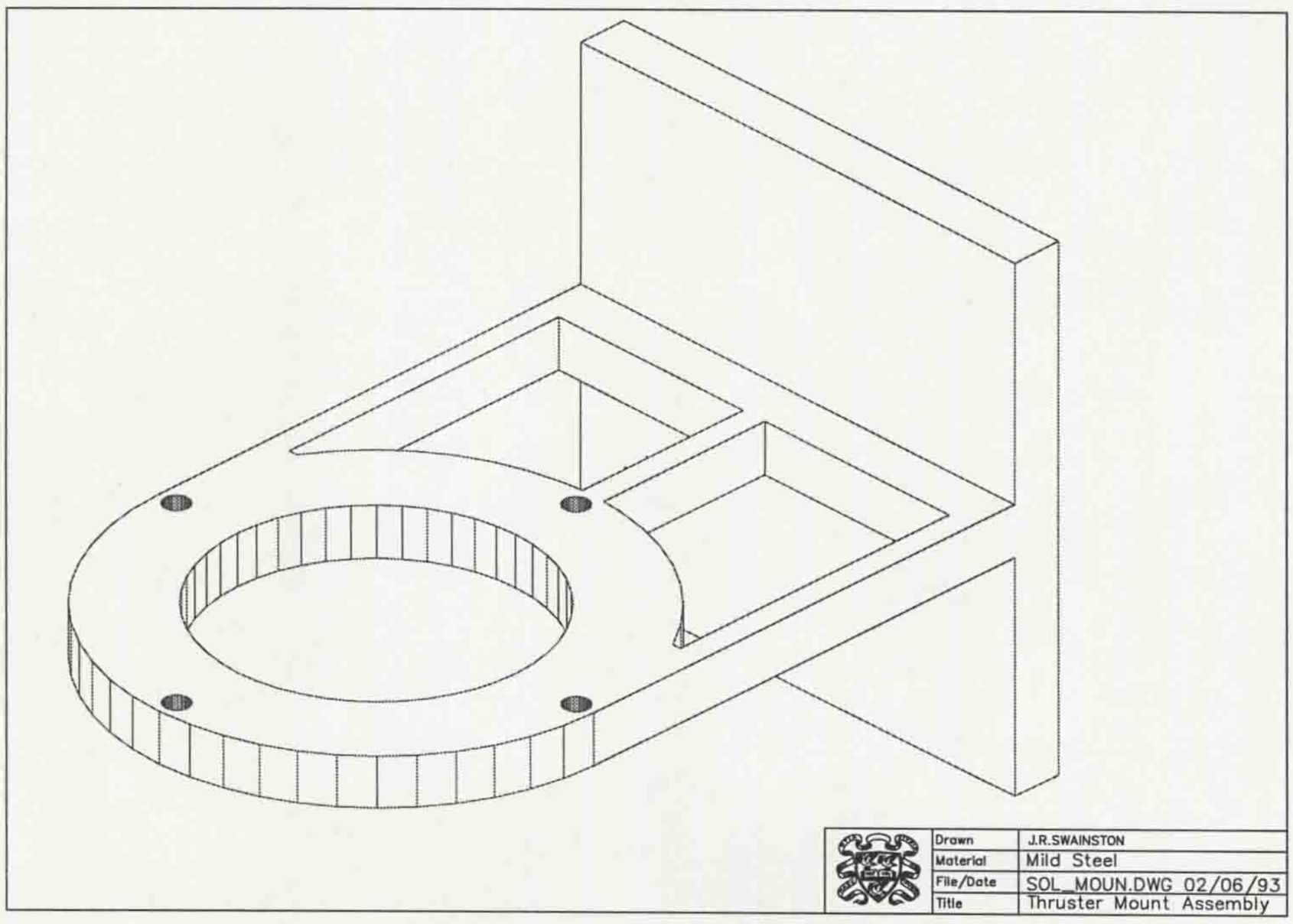

Figure 10, A diagram of the finished Thruster Mount

3.2.3 Spider Joint Design

To connect the six sub-frames together eight "spider-joints" were designed. Originally it was thought that the vehicle could be dismantled by having a non-permanent fixing between the spider-joints and the sub-frames . However, to give greater rigidity, strength and water resistance these joints were welded. By sealing and creating a hollow volume within the frame a small amount of buoyancy provided to compensate for the weight of the frame. Figure 11, shows the technical drawing for the construction of the spider-joints.

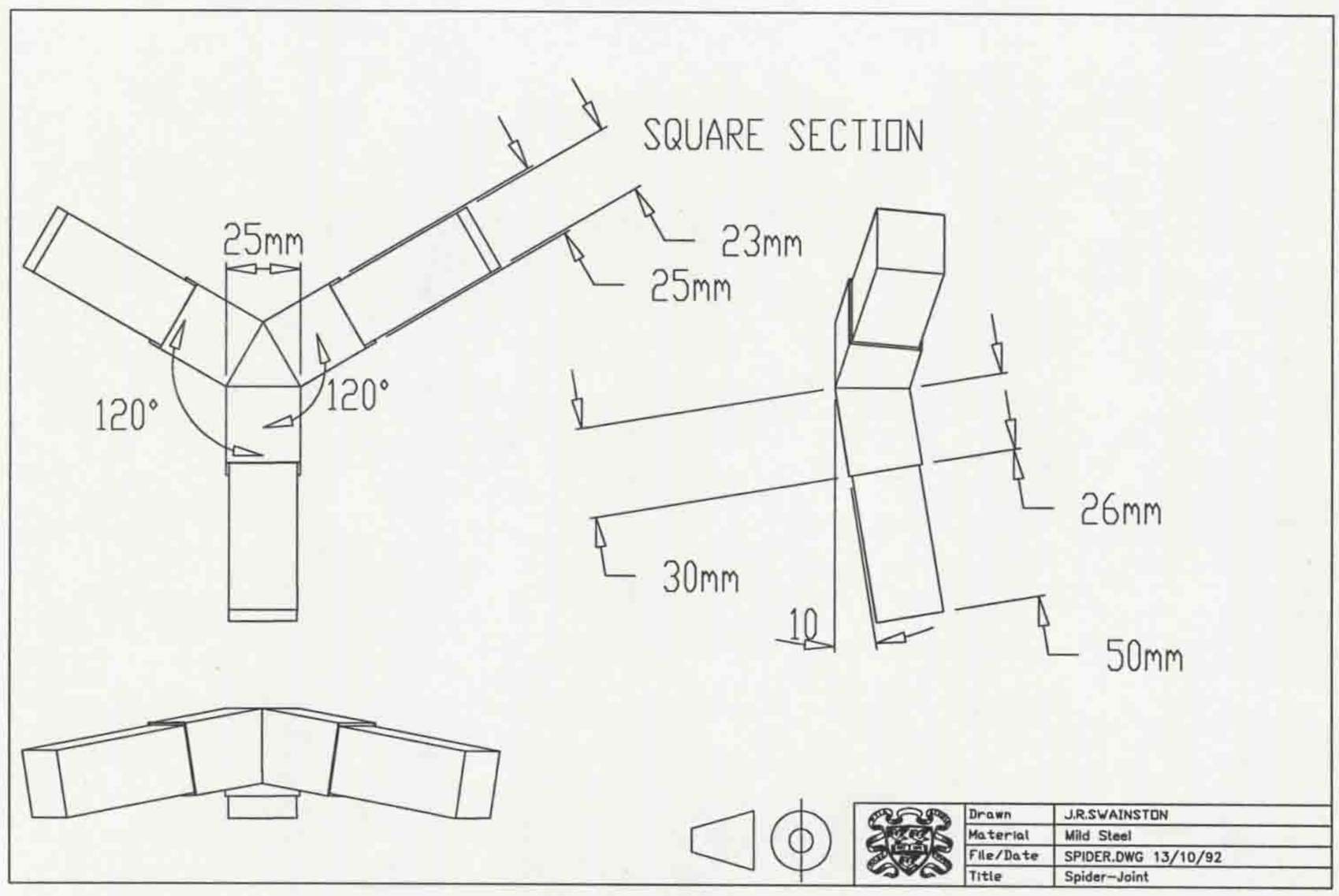

Figure 11, The Spider-Joint

3.2.4 The Completed Frame

From these technical drawings it is difficult to imagine the shape of the completed frame. However, from these basic building blocks the vehicles' frame can be assembled, creating the basic frame of the *Aquasphere* as shown in Figure 12. From these drawings a prototype drawing of the envisaged vehicle was created, including the thrusters, the frame and the central sphere containing the electronic systems. Thus, the original vehicle design, created from the components detailed above is shown in Figure 13.

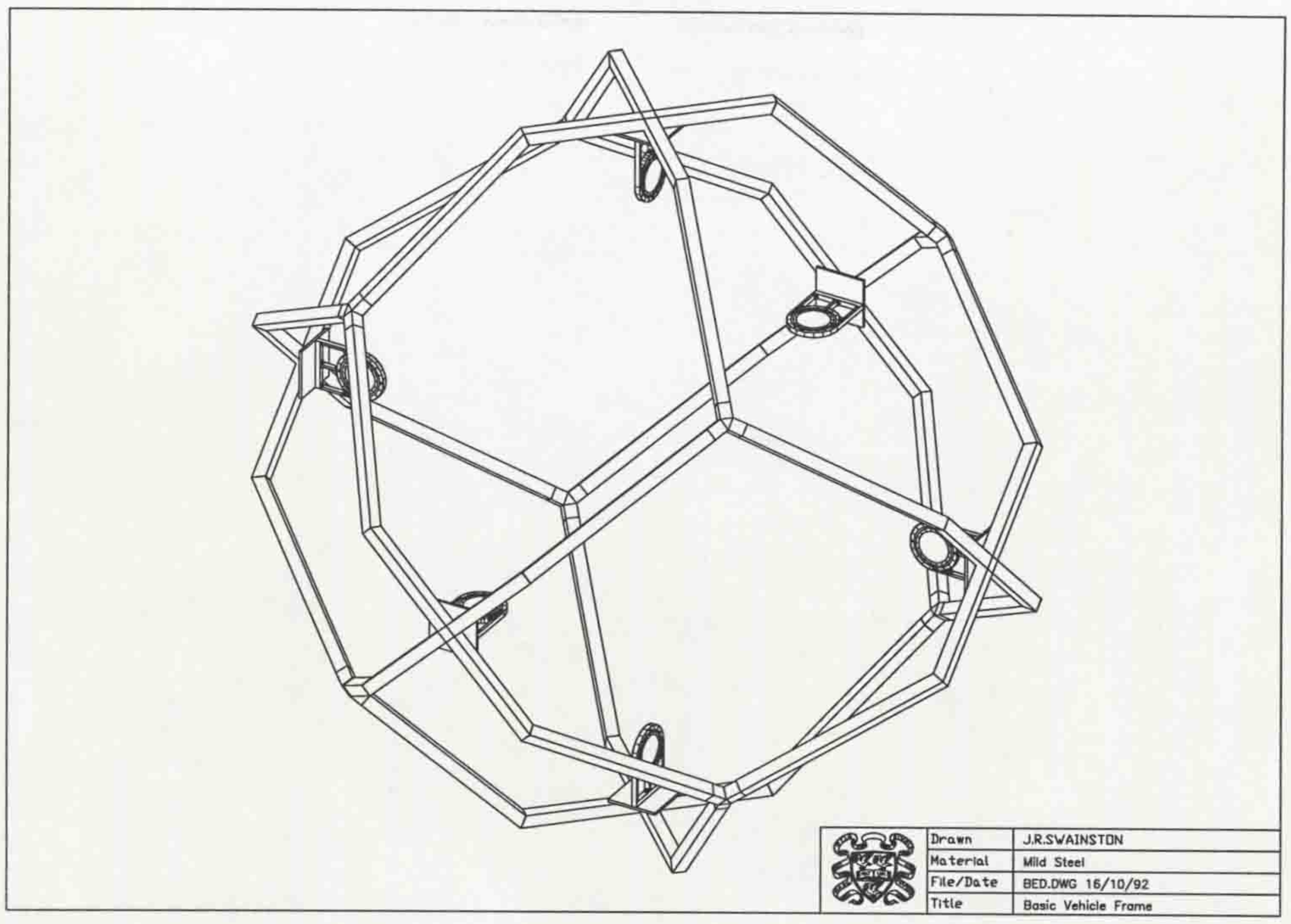

Figure 12, The Aquasphere's Basic Frame

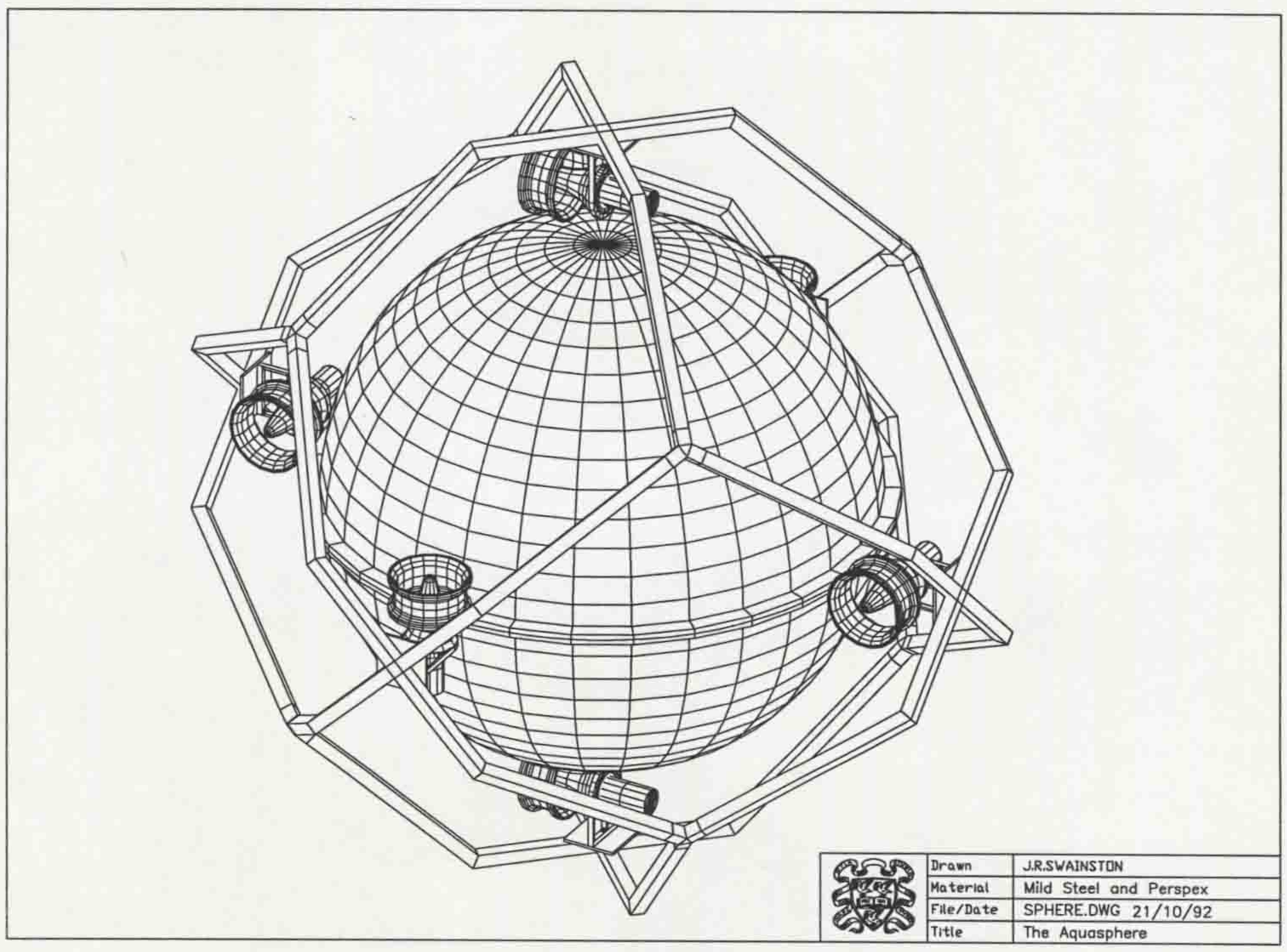

Figure 13, The Original Aquasphere Design Concept

3.3 The Aquasphere Prototype Construction

After the initial design was completed construction of the frame was undertaken. At that stage it was decided that although the frame and its design was satisfactory further work would have to be done to give an internal frame to hold the vehicles' electronic systems. A sphere of sufficient size to hold the vehicle electronics would have given 750kg of buoyancy, and would have cost in excess of £20,000. Therefore, it was decided that a framework would be built inside the vehicle to support boxes holding the electronic systems. A standard 19" rack-mounted arrangement was chosen, so that all the electronic systems could be constructed, and fitted in standard mounting boxes.

The electronic systems to be mounted in the *Aquasphere* were:-

- A thruster drive and control system, requiring one 19" box for extended length eurocards.
- A sensor system, requiring a magnetically inert box for the flux gate compass, and sufficient room for other sensors.
- A networking system, requiring one 19" box for double height standard length eurocards.

Later additions to the vehicle would be:-

- A 3D vision system, requiring one 19" box for standard length eurocards, plus two laser mounts and a camera on a pan and tilt mechanism.
- A robotic manipulator system, requiring the mounting of a robotic manipulator, and one 19" box for extended length eurocards.

To accommodate these systems a simple frame was constructed within the *Aquasphere*, and boxes were designed with a simple clamping arrangement so that they could be slid within the frame, and the vehicle could be balanced. Later, when the optimum position of the boxes was determined the clamping arrangement for the boxes could be made more permanent. To aid in the development of the *Aquasphere* the computer package Autocad[3.4] was used to help the visualization and design of the vehicle. With the release of Autocad version 11 it was possible to create a computer model of the vehicle and its sub-systems to help estimate the mass and rotational characteristics of the vehicle. All drawings

created for the vehicle were then regenerated to take advantage of the information created by this new package.

3.3.1 The Internal Frame

As mentioned above a simple internal frame was created so that the boxes carrying the electronics could be mounted. Four 25mm x 6.25mm section bars were welded across the frame at two of the sides, at the top and bottom of the frame. Previous to this there had been no distinction between front, back, top and bottom, but with the introduction of the frame within the vehicle boundary resulted in a definite orientation. The bars were given mounting holes spaced at 10cm intervals to allow the positioning of the boxes backwards and forwards.

Figure 14 shows the positioning of the cross bar and the vertical struts on the vehicle.

In conjunction with the design of the frame the design of the boxes was considered. A flexible layout of the boxes within the frame was proposed. This layout was later modified, initial tests on the vehicle were carried out using the initially proposed arrangement. This design is shown in Figure 15.

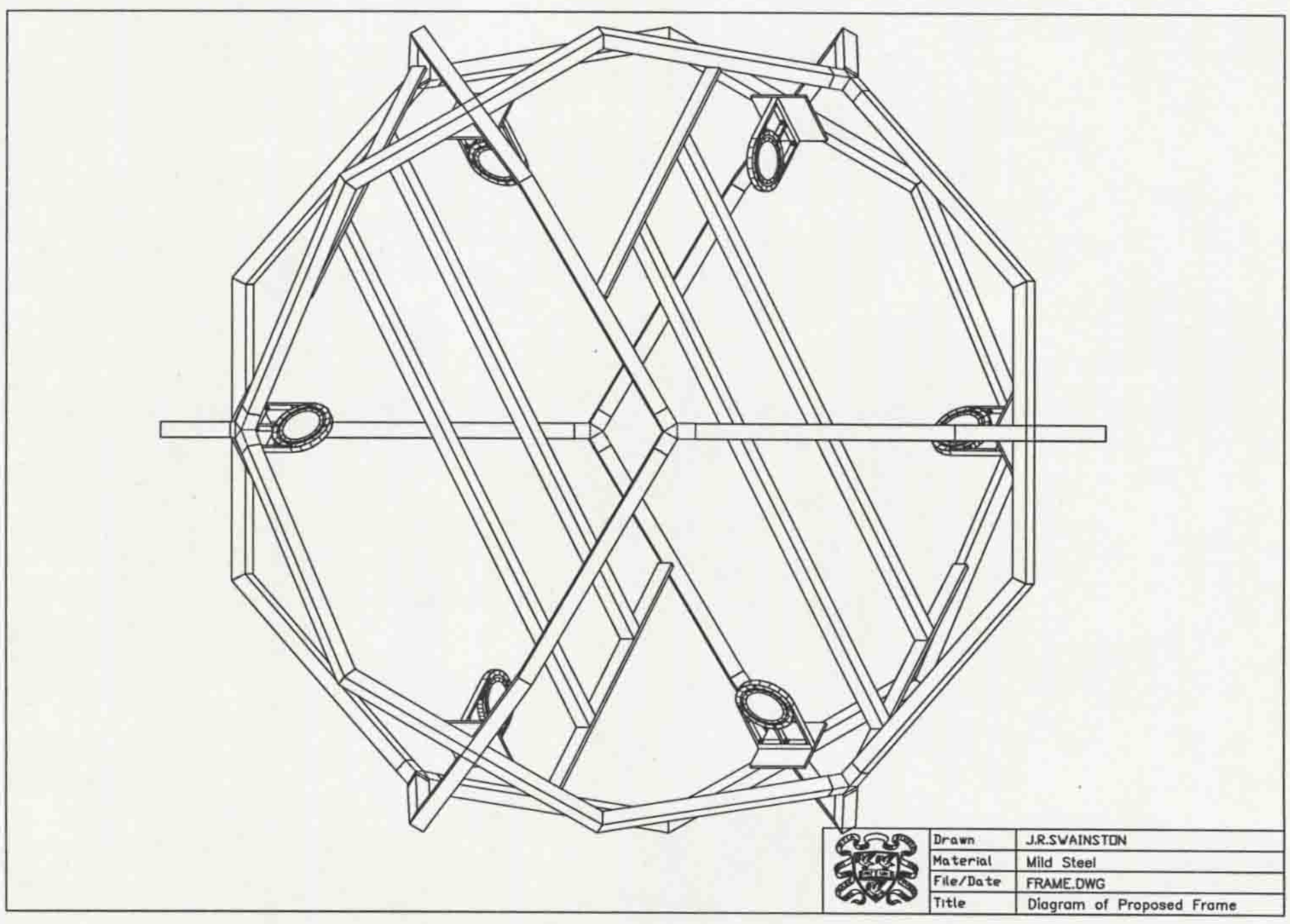

Figure 14, Internal Frame within Aquasphere

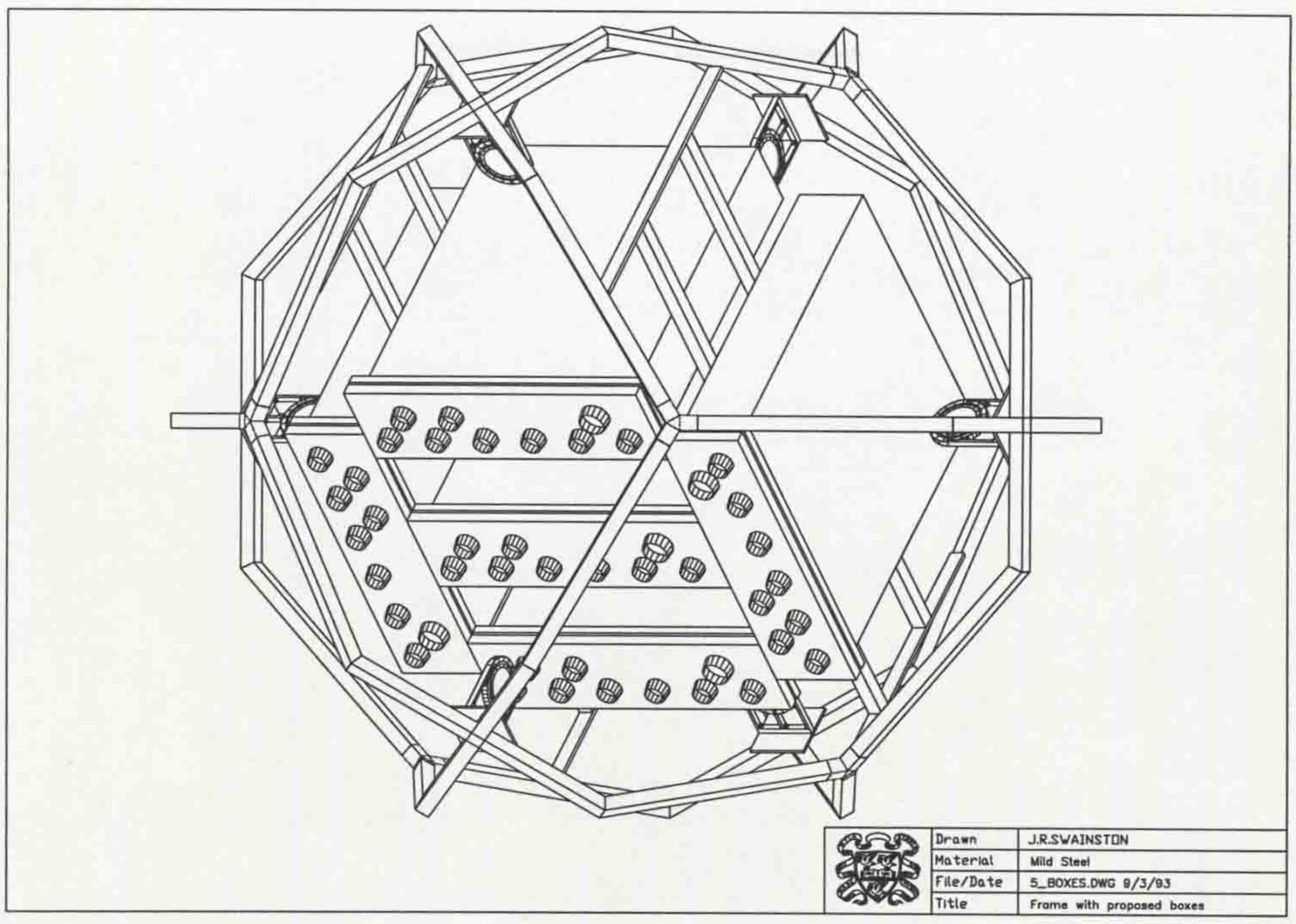

Figure 15, Proposed Internal Configuration

3.3.2 Design of the Electronics Boxes

3.3.2.1 The Networking Box

The first box constructed was to house the network interfaces. A general specification for the box was given to one of the partners in the project, Stolt Comex Seaway.

> To build a box to withstand at least 30m pressure, large enough to hold a standard length double height eurocard rack, with mounting holes to fix a rack mounted system within it. The box should have a detachable front cover upon which the network's connectors can be fixed.

From this specification the first prototype box was created. The drawing for the networking box is shown in Figure 16.

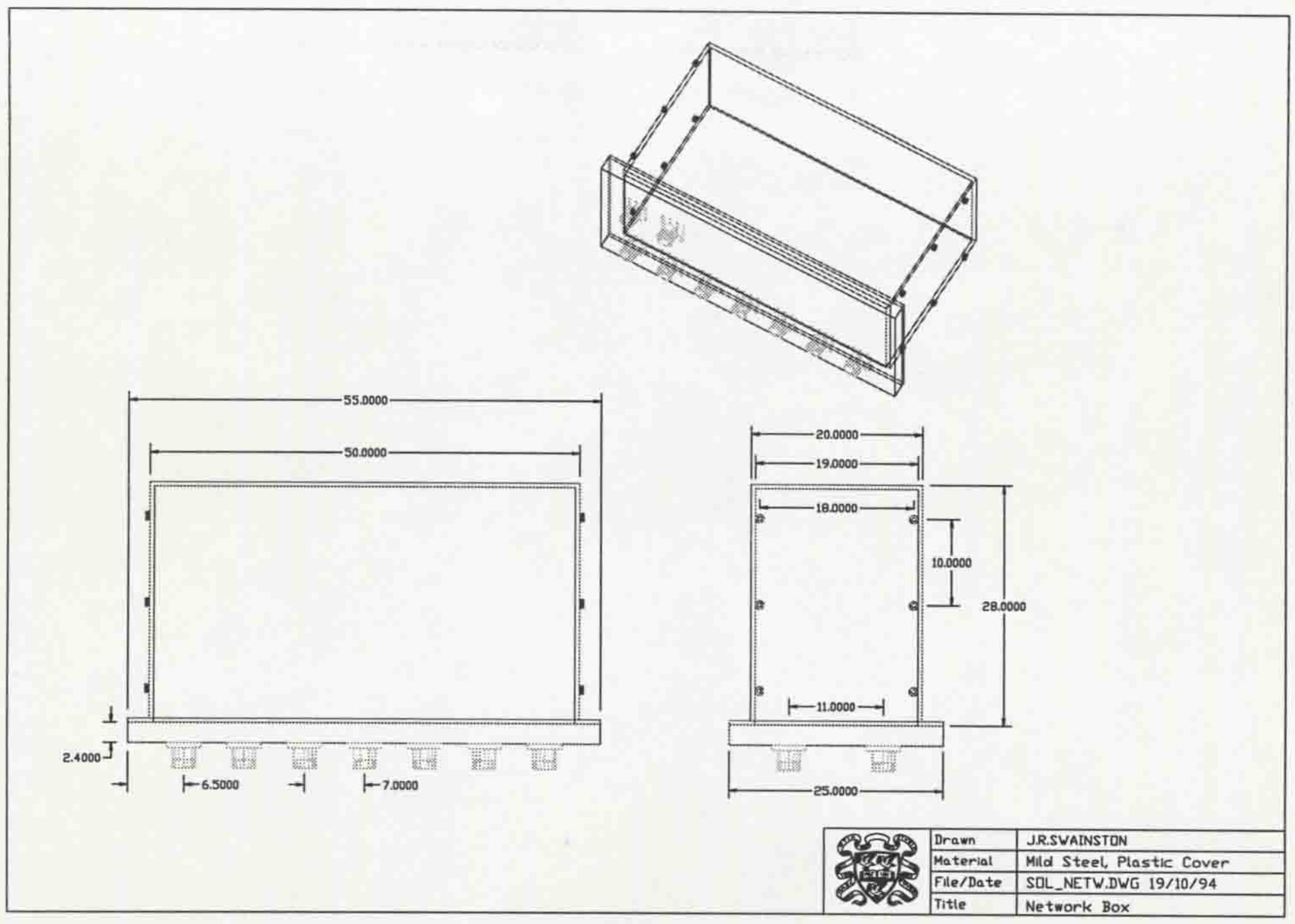

Figure 16, The Networking Box Design

Detailed information on the mass and inertia of the components is given in appendix 10.1, derived from the Autocad solid modeling package.

An o-ring seal was added to the front cover, and stainless steel bolts were used to fix the front cover to the box. Underwater electrical "Jupiter" connectors were added to the face of the box to provide the necessary interconnections.

3.3.2.2 The Sensor Box

The design of the sensor box was determined by the special mounting needs of the sensors held within the box. Due to the box containing a flux-gate compass, the structure of the box had to be of a magnetically inert material. Aluminium was chosen for the bulk of the structure, with plastic front panels secured by stainless steel bolts. In addition to the flux-gate compass there were pressure sensors, a combined paddle wheel water speed and temperature sensor, and an ultrasonic depth sounder. The pressure sensor could be easily accommodated by a tapped hole on one surface, but the ultrasonic sensor needed a large hole on the base of the box to allow a clear line of sight to the bottom of the vessel. The

positioning of the paddle wheel sensor is arbitrary, but was chosen to go on the base of the box also. Figure 17 shows the drawings for the construction of the Sensor box.

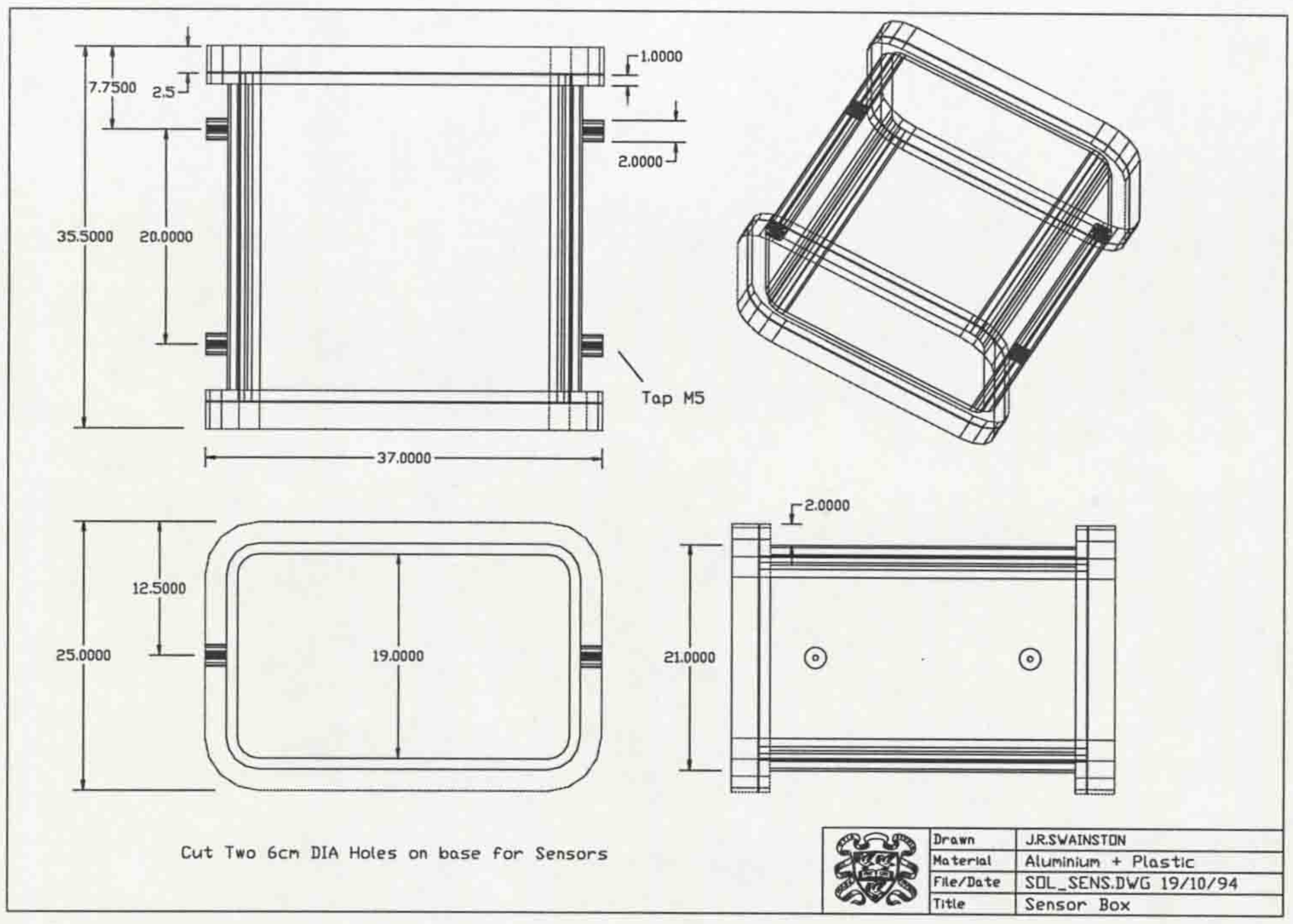

Figure 17, The Sensor Box

Originally, the aim was to mount the flux gate compass in the aluminum and plastic sensor box. However, later in the project additional equipment was introduced into the sensor case which caused interference with the compass. The compass therefore needed relocating to another position. To reduce the effects of magnetic interference from the steel frame and the electronic systems a plastic casing was made, within which the compass element was mounted. This casing was mounted at a remote location on the vehicle, as far away as possible from the frame and other sources of magnetic interference. The relocation of the compass in this casing solved the problems of interference. More information on this is given in Chapter 5.

3.3.2.3 Thruster Power and Control Box

When designing these boxes it was important to bear in mind that a flat-sided square box provides very little protection against compression at large depth. The network prototype box was constructed from a thick section of steel plate, and was therefore very heavy. The flat surface area of the thruster control box was greater again than the networking box, so to give greater strength to the design ribs were added. With the added ribbing a slightly thinner wall thickness was chosen to try to reduce the mass of the boxes. Figure 7, shows the design of the thruster control box.

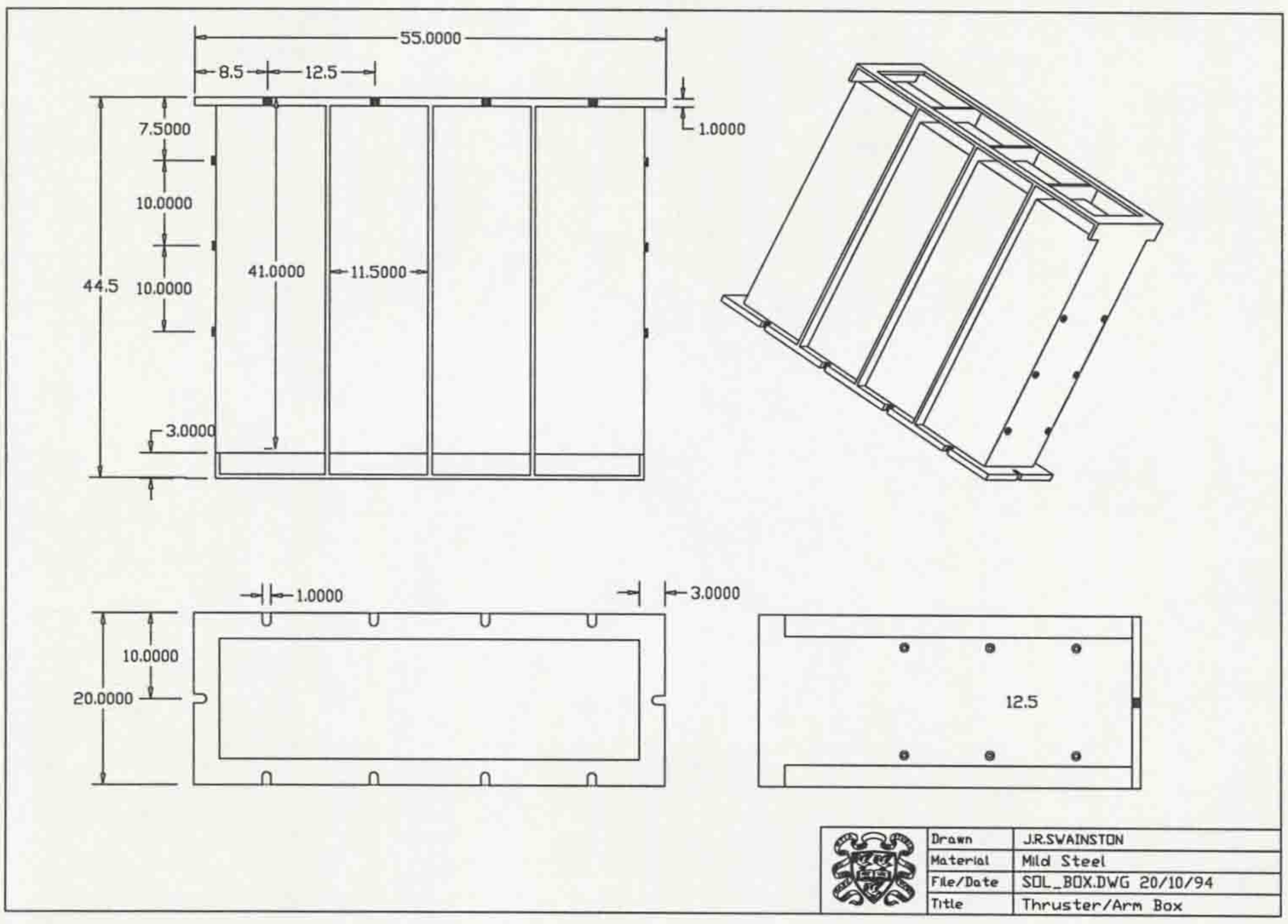

Figure 18, Thruster Control Box

3.3.2.4 The Thruster Cowling

For the thrusters to be mounted on the vehicle an enclosure had to be designed to seal the thrusters and their connections. A simple cylindrical design with an underwater connector mounted at one end was used. The cowling can be fixed to the thrusters semi-permanently to form a sealed unit. This unit of the thruster and cowling can then be mounted and unmounted from the vehicle, fixing to the thruster mounts, without damaging the

integrity of the waterproof seal. The seal, an o-ring, is built into the body of the thruster itself. For strength, and corrosion resistance, the cowling was fabricated from stainless-steel. Figure 19, shows the design for the thruster cowling.

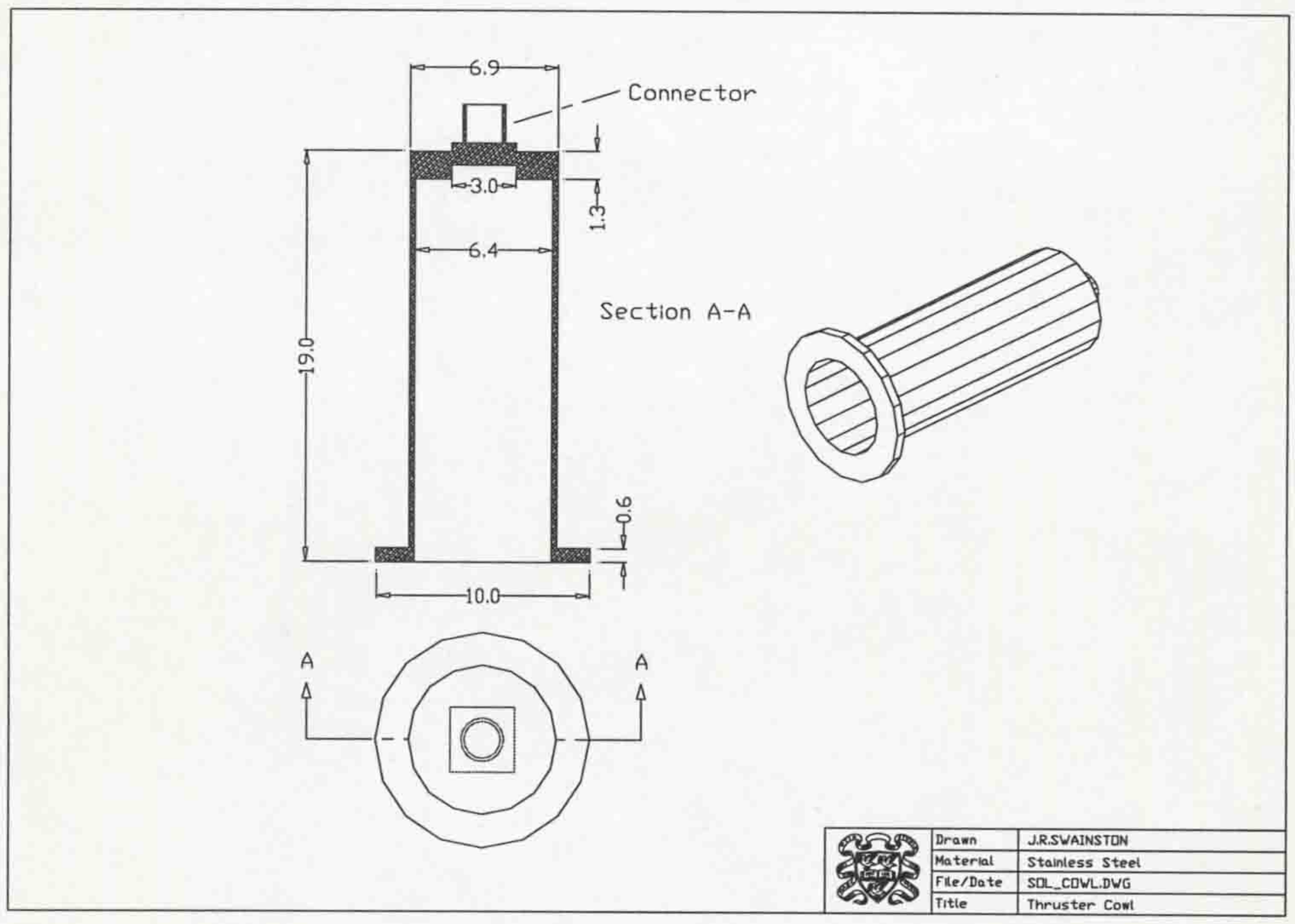

Figure 19, The Thruster Cowling

3.3.3 Mounting The Vision System

The original design of the vehicle was to provide a testbed for novel control methods. However, later in the project it was decided to use the Aquasphere to test a three-dimensional vision system and underwater robotic manipulator. The addition of these systems required extra boxes to be mounted on the vehicle. The addition of extra boxes and the new vehicle components required modifications to be made to the vehicle frame.

The dimensions of the boxes used for the thruster control and for the control of the robotic arm are identical. The design of the box for the vision system is identical to the design of the arm and power boxes, except that the box is shallower in depth

In addition to the main electronics box for the vision system other pieces of equipment needed to mount to the vehicle for the vision system to work. Most important for

the vehicle is the addition of a camera system, and to utilize the three-dimensional capabilities of the system two laser systems are also required. The system designed by Yu[3.5] required a camera to be mounted between two laser systems which were separated by as wide a baseline as possible. The advantage of this vision system was that the three dimensional ranging of the system was independent of the camera optics. This allowed the use of a camera system which was mounted on a pan-and-tilt mechanism, with a zoom facility. In conjunction with Stolt Comex Seaway a marinised housing for the laser systems and the pan-and-tilt camera mechanism was designed.

3.3.3.1 The Pan and Tilt Camera Unit

The solid model of the pan-and-tilt camera unit is shown in Figure 20. Additional information for the design of this unit is included in Yu[3.5].

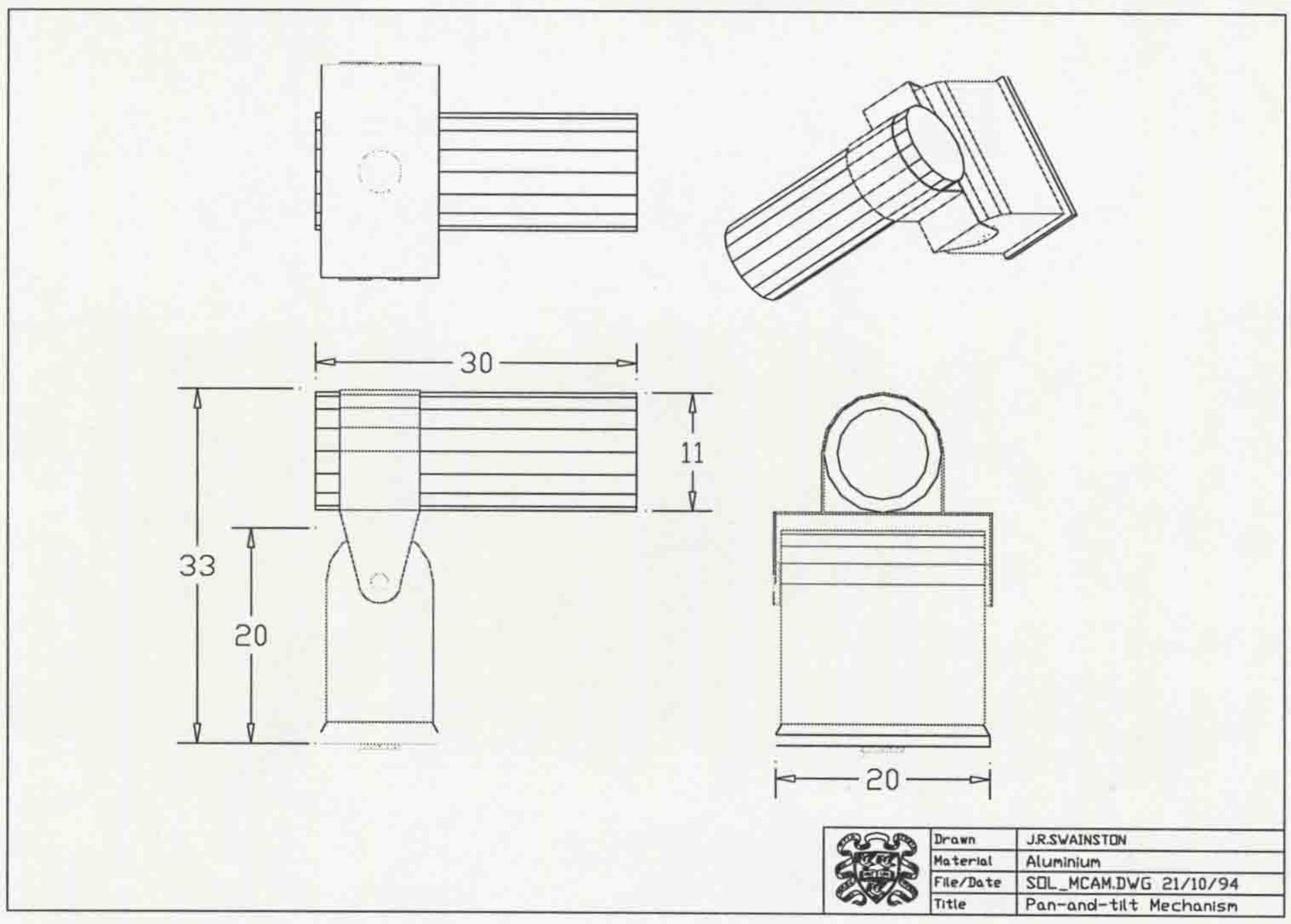

Figure 20, The Pan-ant-tilt Mechanism

To mount the camera system on the vehicle, the design of the external frame had to be altered. To give the maximum amount of space for the vision system and the robot arm, whilst providing a large base line between the two laser systems the location of the vision system was chosen to be at a point on the outer frame which was offset with respect to the

internal frame. A platform was designed to replace one of the spider-joints, and provide a mount for the camera system. Although the weight of the camera system is low, the positioning of the camera at such a critical point of stress meant that the structural strength of the frame was reduced. However, the introduction of the internal frame compensated for this reduction in strength. The design of the camera platform was such as to allow the camera full movement for its pan-and-tilt mechanism, whilst being light weight and strong. The design for the camera platform is shown in Figure 21.

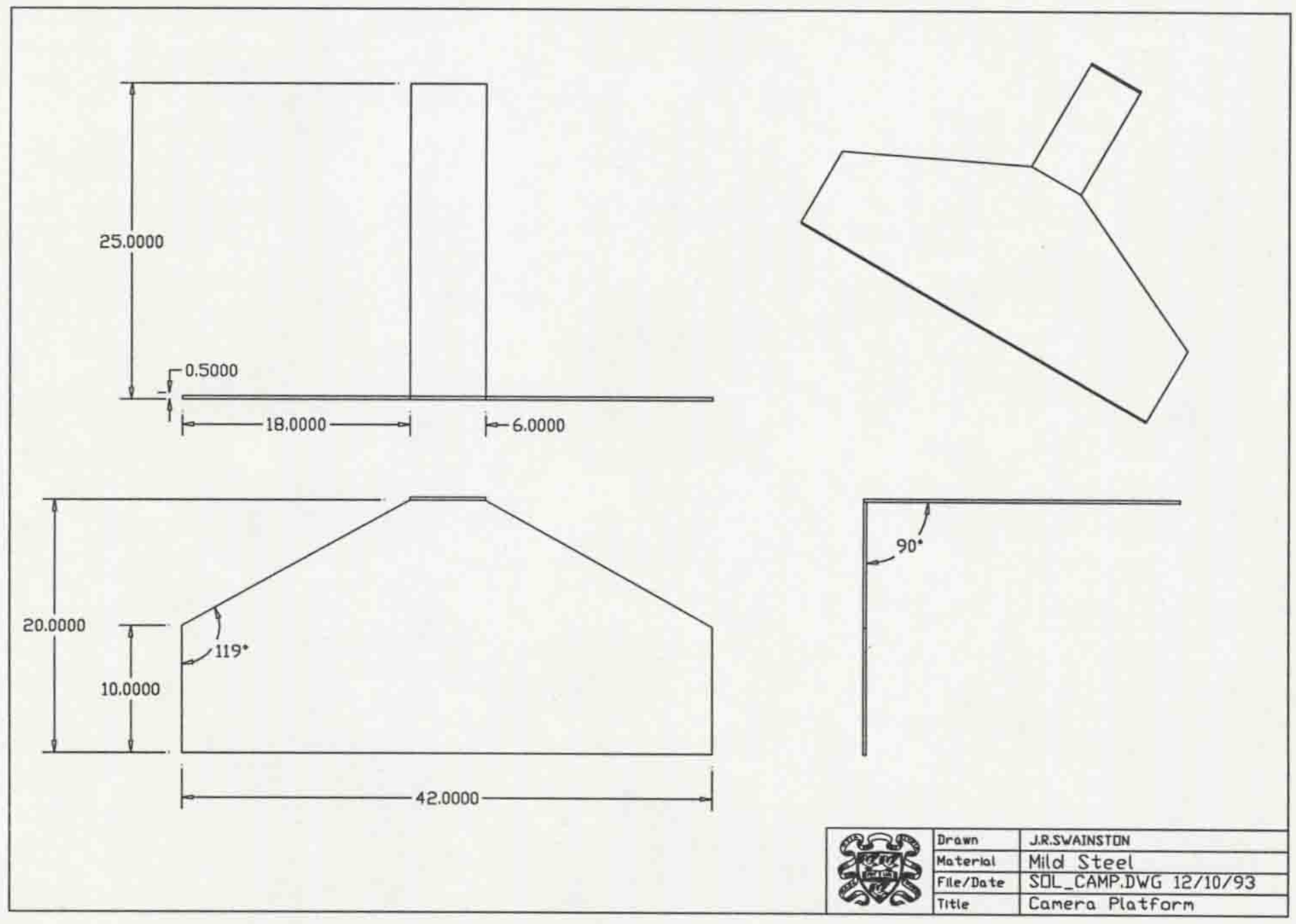

Figure 21, The Camera Platform

To fix the camera platform to the vehicle one of the spider joints had to be cut off the frame. The platform was welded into place, sealing the cut ends of the frame. A hole drilled in the center of the platform provided a mount for the camera.

3.3.3.2 The Laser Range-finder Mounts

The laser range finding system designed by Yu[3.5] was fitted into a marinised cylindrical container. The cylinder had a circular glass end from which the laser light could be emitted, and a connector for the electronics at the other end The container itself was constructed from nylon, with a wall thickness of 1cm. A fixing bracket was designed to give an attachment to the laser cylinder. A drawing of the container, designed by Yu, attached to the mounting bracket is shown in Figure 22. A metal band holds the cylinder to a base plate which has two threaded bolts to allow the pitch of the cylinder to be adjusted with respect to the vehicle.

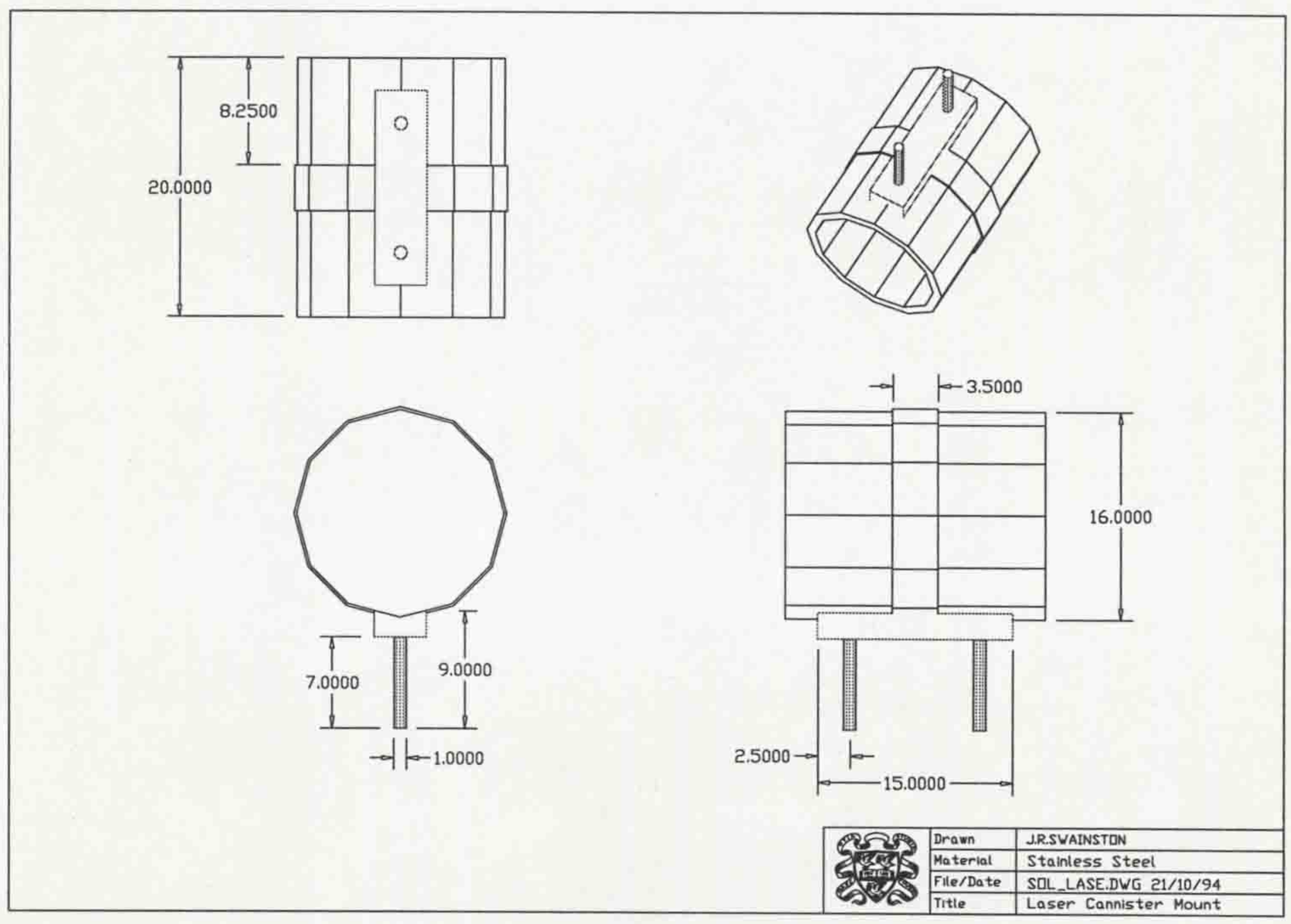

Figure 22, The Laser Canister Mounts

To fix the laser mounts to the frame of the Aquasphere a simple bracket was constructed. The bracket clamps onto the frame, providing an adjustable mount for the setting up and calibration of the vision system. The brackets are made from 1mm stainless steel plate which was cut and then welded together. Figure 23 shows the drawing for the laser frame mounts.

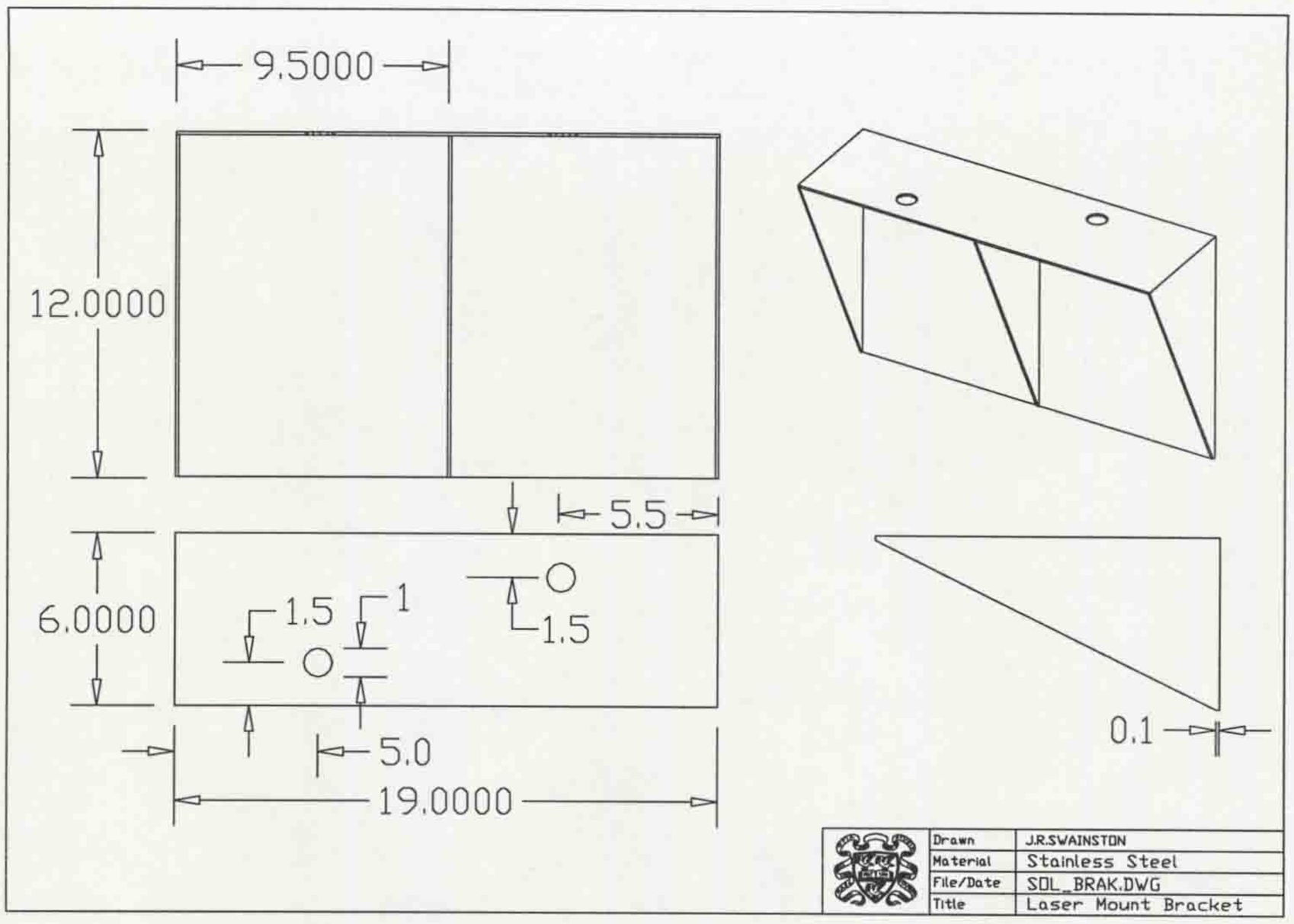

Figure 23, The Laser Frame Mounts

3.3.3.3 The Mounted Vision System

Figure 24 shows a closeup of the vision system. The top of the robotic arm can be seen angled to one side at the base of the picture. The other principle components of the vision system are labeled.

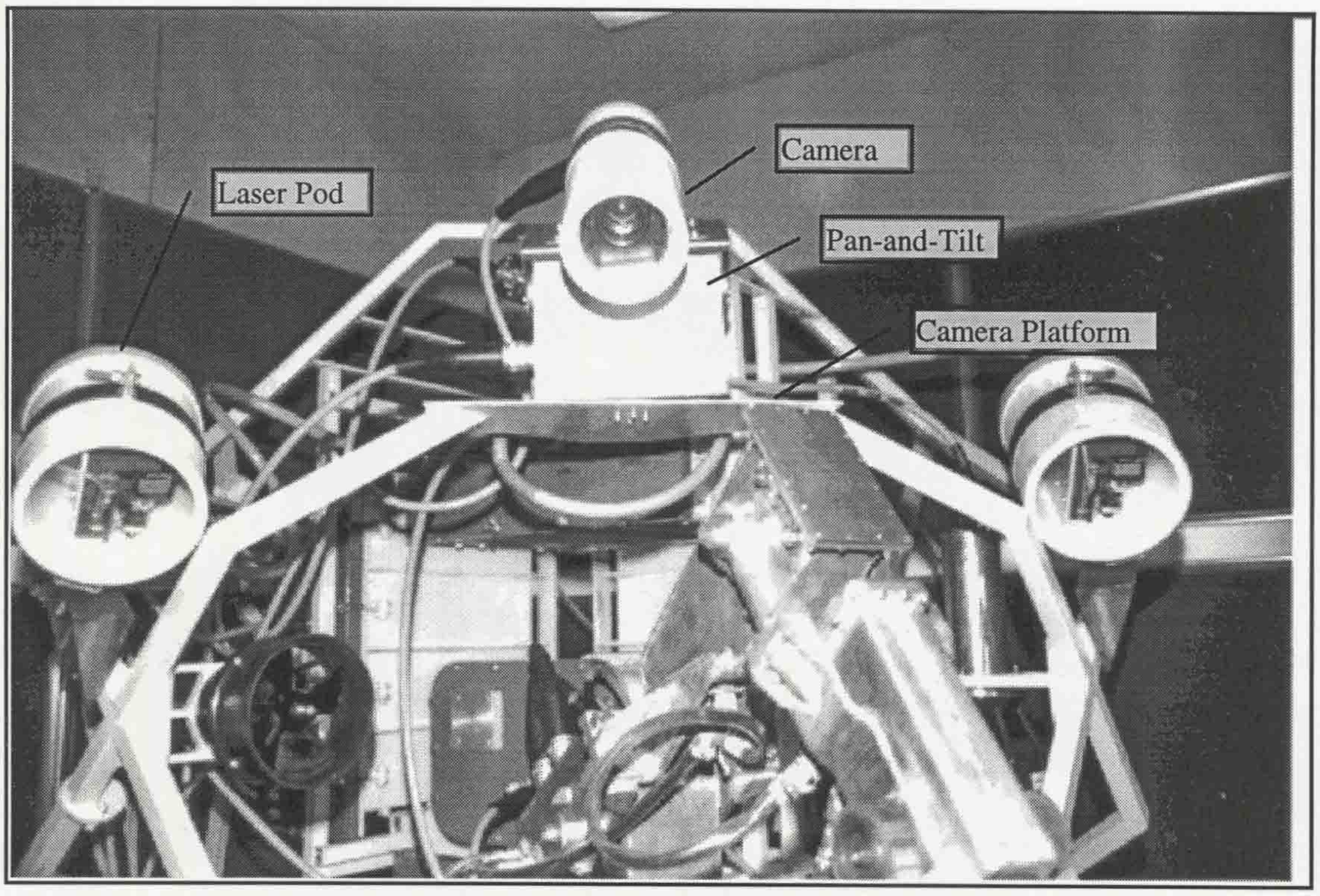

Figure 24, The Vision System

3.3.4 Mounting the Manipulator Arm

The working space of the manipulator arm, designed by Sarafis[3.6] was to be as large as possible to permit maximum flexibility in arm movement. In addition, the mount for the arm would have to be strong to cater for the forces exerted upon it by the arm, and the mount should ideally be located near to the center of mass for the vehicle so as to minimize instability caused by the manipulator arm.

The location of the manipulator arm also had to comply with the constraints of mounting the vision system (Section 3.3.3), as the manipulator arm needed to be mounted in line with the vision system. To mount the arm a bracket was constructed which would mount over the spider-joint below the camera platform. This location gave the maximum position

for strength and lateral movement of the manipulator arm. The mount was further from the centre of the vehicle than the ideal for the positional control of the vehicle. However, this compromise had to be made for the useful operation of the manipulator arm. The mounting plate itself was designed so that the base of the manipulator arm would fit directly onto it, being held down by four stainless steel bolts. The design of the manipulator arm mounting plate is shown in Figure 25.

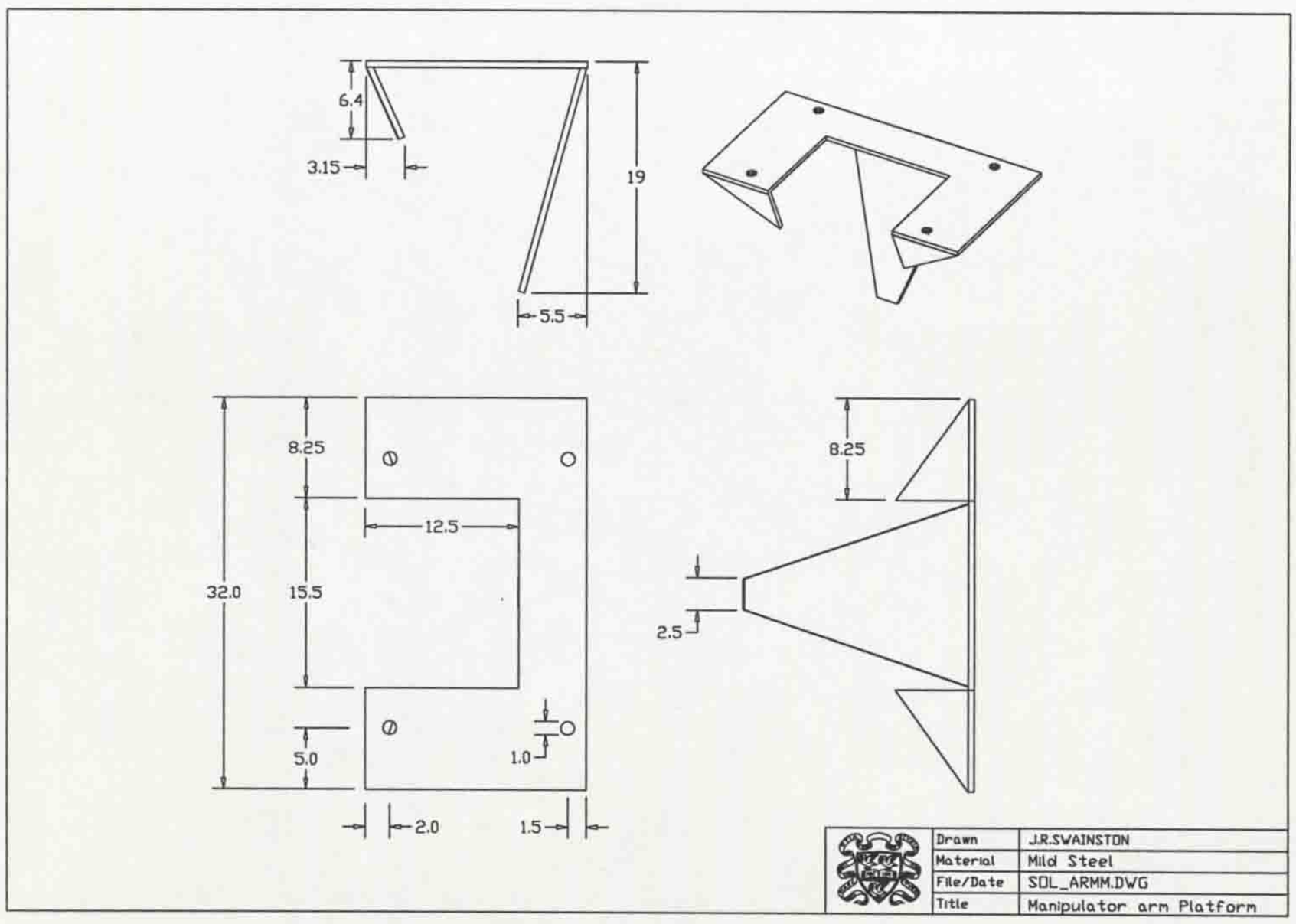

Figure 25, Manipulator Arm Platform

3.3.5 The Aquasphere Prototype

Before the design of the vehicle could be finalized testing of the vehicle had to be carried out. A picture of the prototype vehicle being lowered into the Coburg dock at Liverpool marina is shown in Figure 26. At this stage the manipulator arm platform had not been designed and fitted, and the foam buoyancy was attached by plastic packaging ties. The laser systems were also not included at this stage, as they could have potentially been damaged by a collision with the dock walls, due to their being mounted externally to the vehicle's frame. The system of packaging ties allowed the placement of the buoyancy to be altered to help trim the vehicle. In addition the boxes were adjusted vertically within the

frame to find their optimal positions. The frame could also be moved laterally in 10cm steps to trim the vehicle. This became particularly useful when a lead weight to simulate the manipulator arm was later attached. As can be seen in the picture (Figure 26) the design of the prototype vehicle had radically changed from the initially proposed shape (Figure 13).

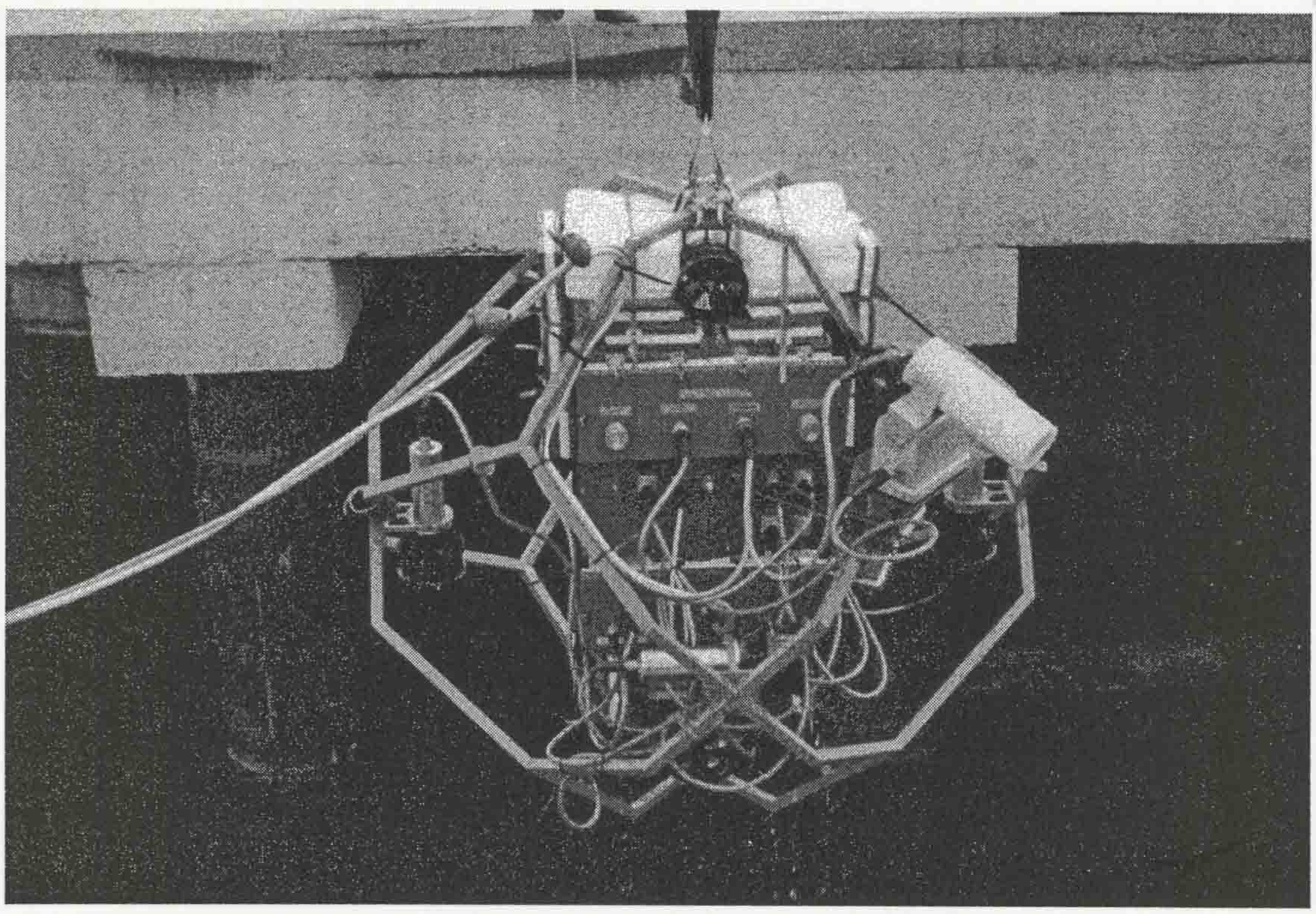

Figure 26, The Aquasphere Prototype Vehicle.

3.4 The Final Design of the Aquasphere

From the initial testing of the vehicle at the Coburg Dock, Liverpool it was found that further design changes would have to be made to the Aquasphere. One of the most important design changes to be made was to accommodate the weight of the manipulator arm. The design of the manipulator arm by Sarafis[3.6] was initially such as to produce a neutrally buoyant manipulator arm. However, by the final stages of its design the mass of the manipulator arm rose to 23kg with a displacement of 7 litres. To help the marinisation of the manipulator arm, the hollow cells of the arm could be filled with dielectric oil to eliminate a pressure differential at depth. Filled with the dielectric oil the manipulator arm's mass increased to 30kg. To compensate for this mass at one side of the vehicle the frame had to be redesigned to allow the boxes to be mounted at the opposite side of the vehicle.

From the initial tests it was found that the sensor box was approximately 9kg positively buoyant, whilst the networking box was 14kg negatively buoyant. Placing the Sensor box at the base of the vehicle, with the networking box mounted above it made the vehicle unstable. Therefore it was decided that the networking box should be mounted at the base of the vehicle, with the sensor box mounted above it on the frame. This created the most stable configuration, but occluded the path of the ultrasonic sensor, mounted on the base of the sensor box. As further ultrasonic transducers were to be mounted remotely around the vehicle the loss of this sensor was not expected to cause any difficulty for the control system.

3.4.1 The Internal Frame

From the trials carried out in the Coburg dock, positions were determined for the location of the component boxes. The ability to relocate the boxes was now not needed, and a frame with fixed box locations was designed.

To mount the frame within the vehicle new supports had to be constructed to allow mounting of the boxes further away from the manipulator arm.

The new frame mounts were fitted on the top and bottom of the vehicle on the two frame sub-sections furthest from the manipulator arm and camera platform. A spacing of 101cm was left between the frame mounts vertically. It was decided that a further two of these mounts should be attached to the base of the vehicle, on the sections nearest the arm to give the vehicle additional strength, by the formation of a square of supports. A diagram of the Aquasphere's new frame is shown in Figure 27 .

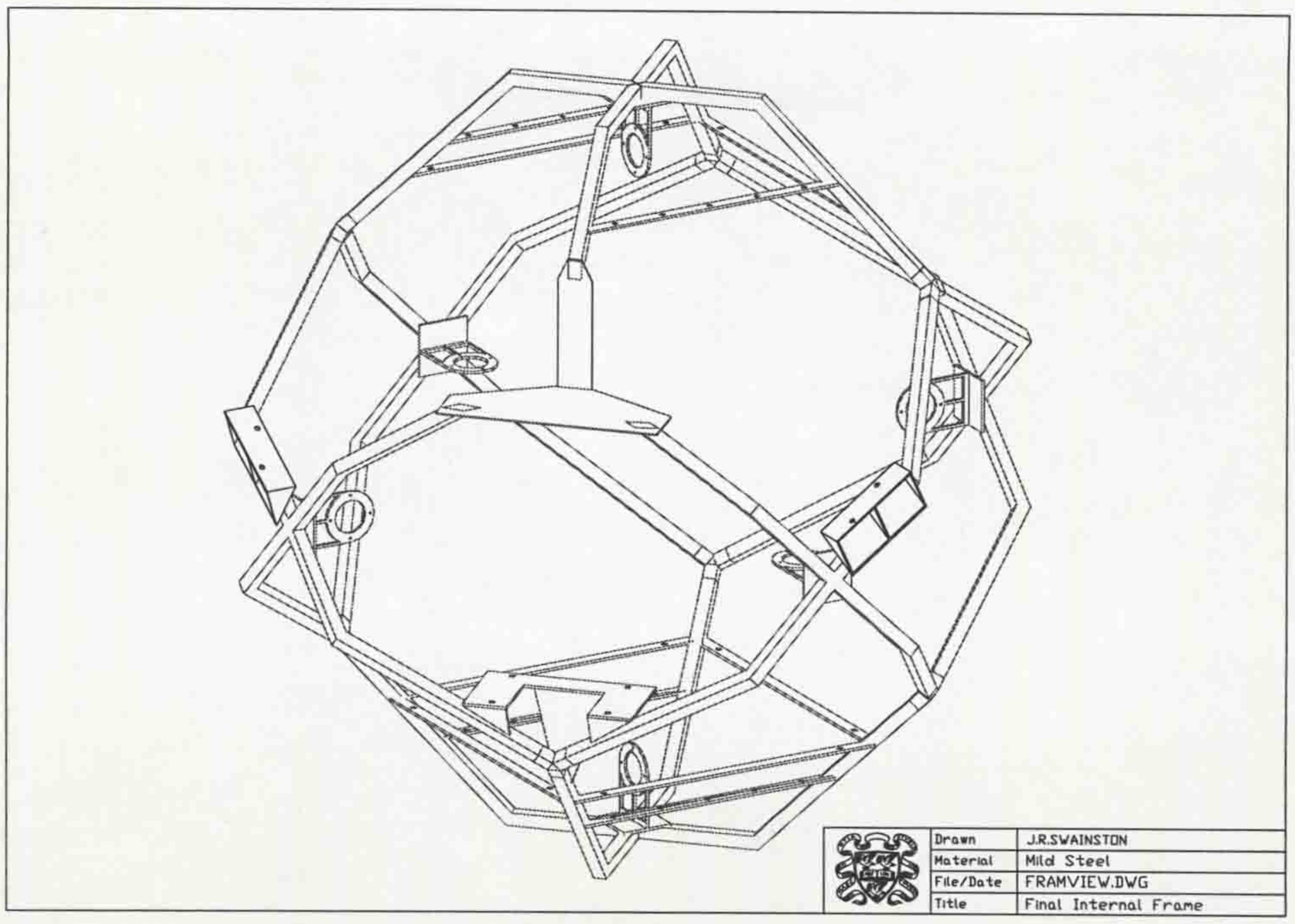

Figure 27, The Aquasphere's Final External Frame

3.4.2 The Box Support Frames

To support the boxes, within which the electronic subsystems were mounted, three new vertical support members had to be created

With the insertion of the three vertical frame pieces Aquasphere's frame was completed. A diagram of the vehicle's completed frame is shown Figure 28.

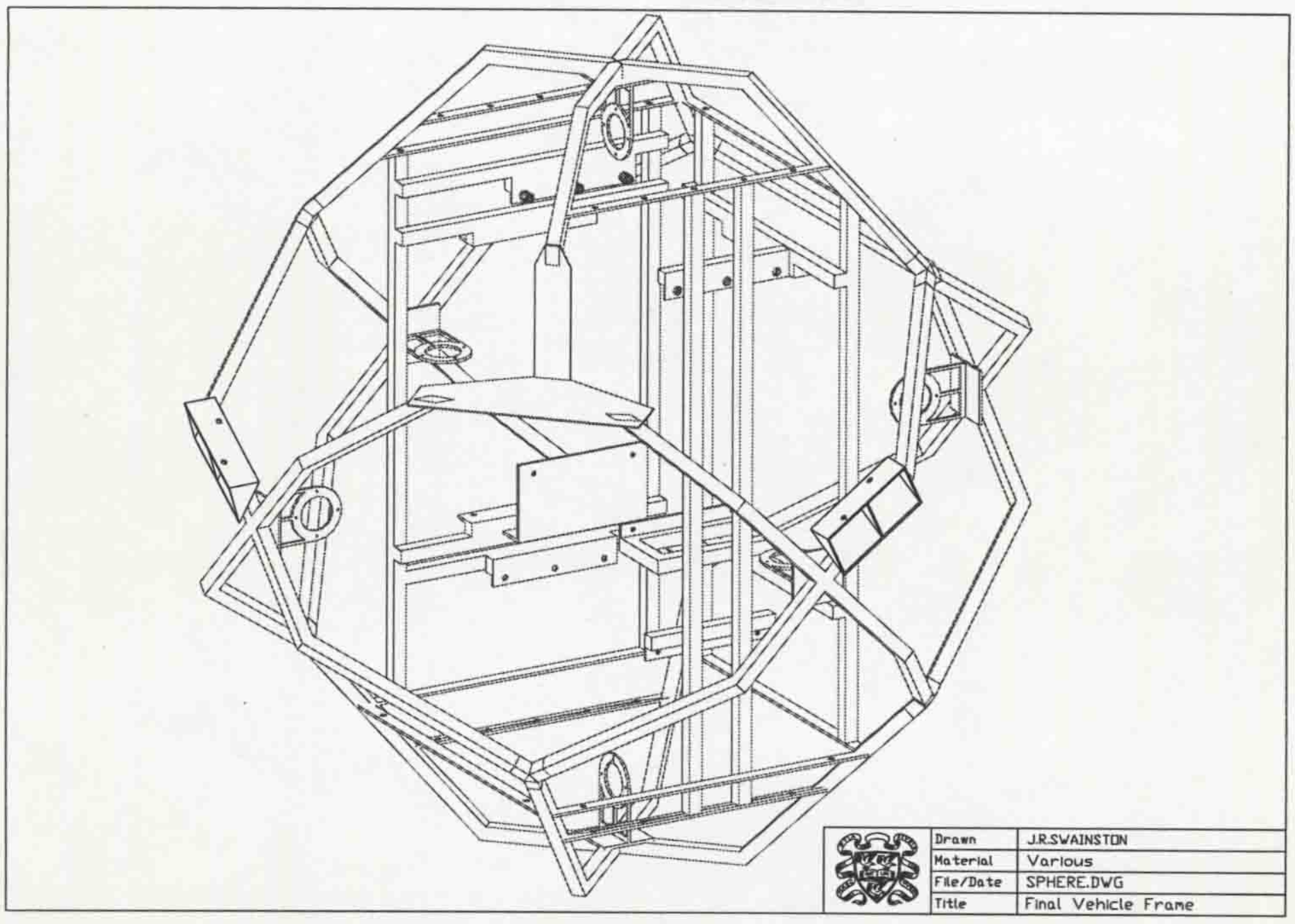

Figure 28, The Final Vehicle Frame

3.4.3 Final Component Locations

The final positioning of the components attached to the Aquasphere determined by the solid modeling of the Aquasphere can be seen in Figure 29. The solid model can be compared with the photograph of the assembled vehicle shown in Figure 30. The method of assembling the Aquasphere and the details of the Autocad model are given in appendix 10.1and10.2.

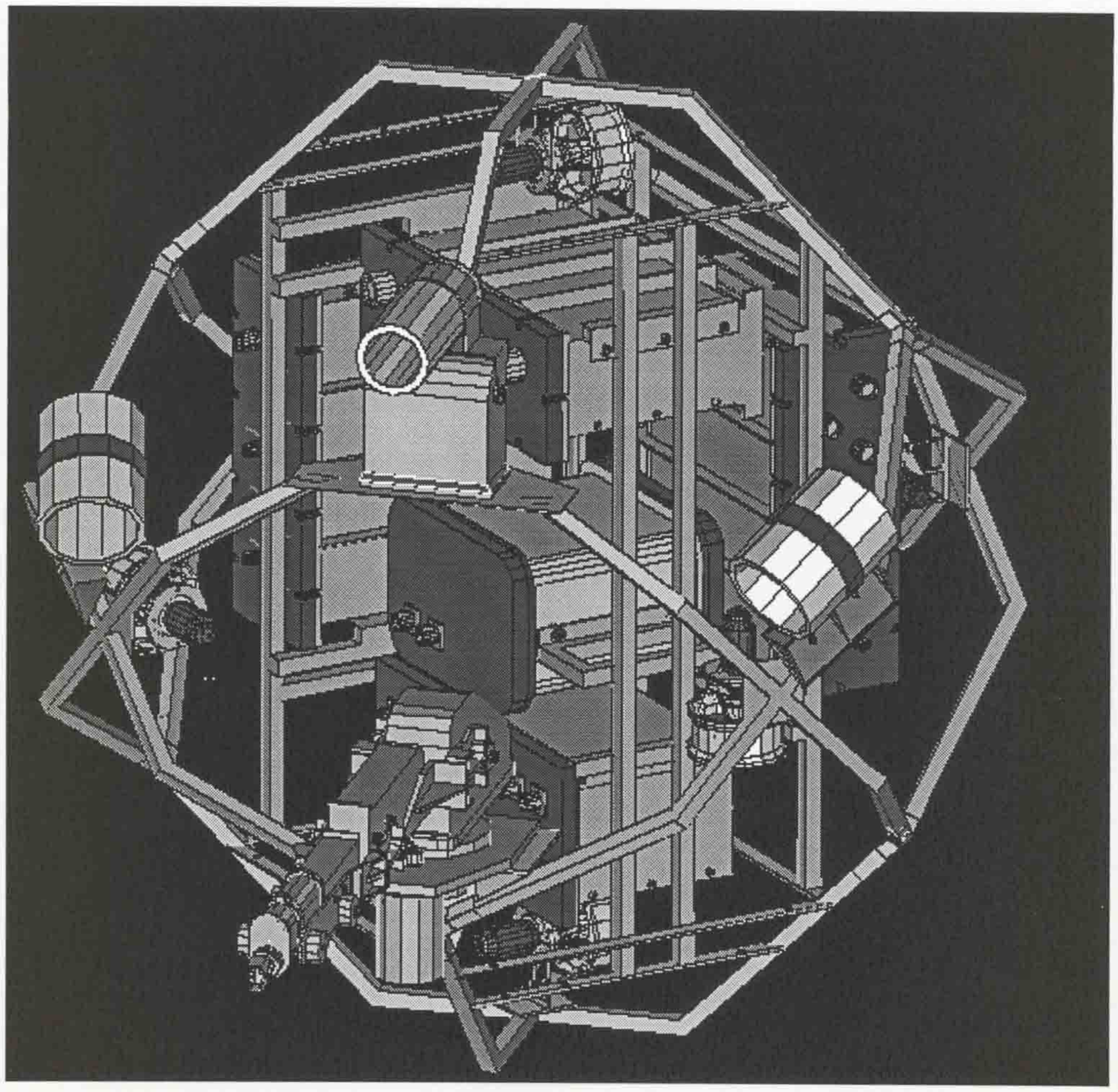

Figure 29, The Aquasphere Solid Model

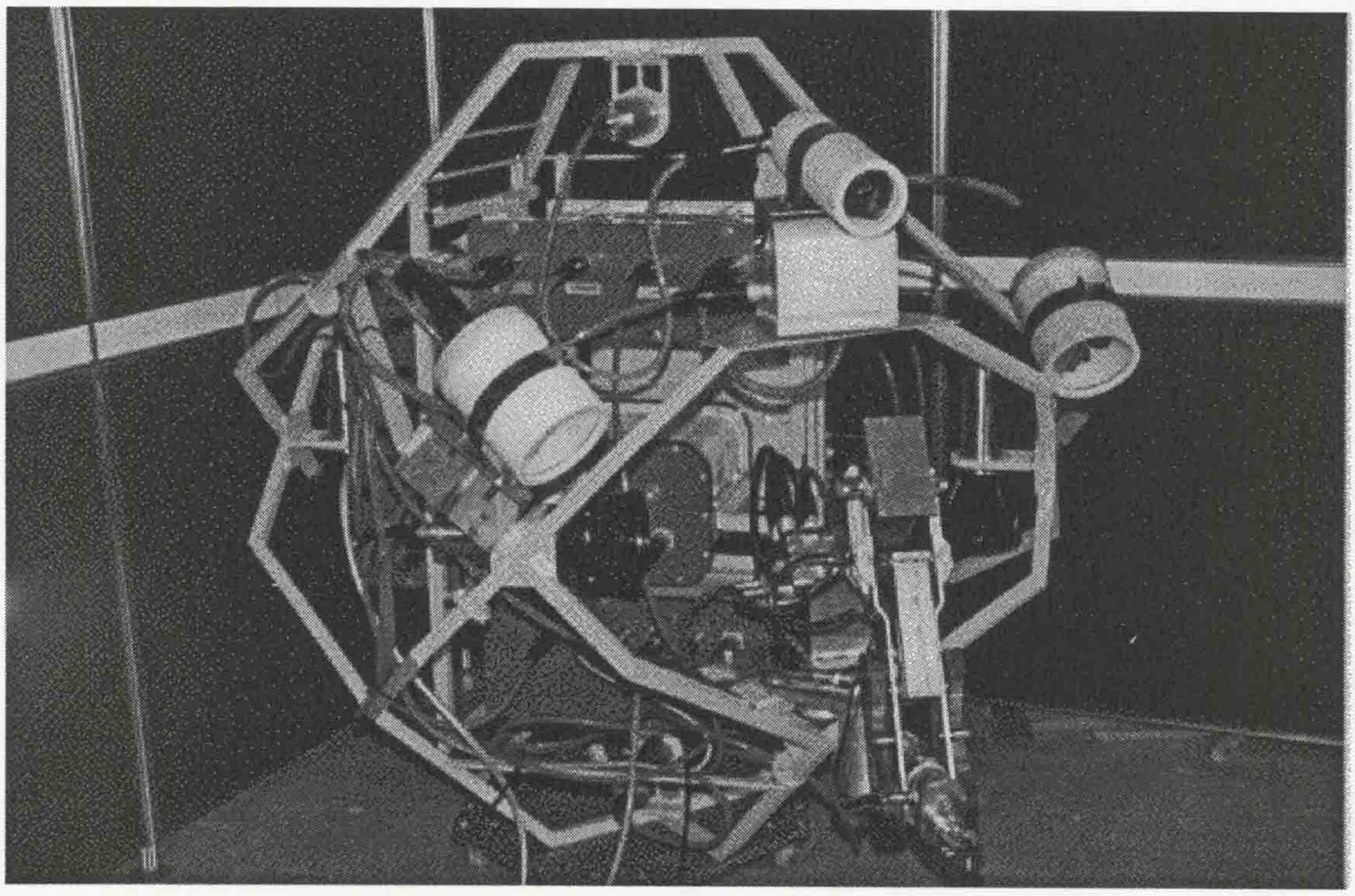

Figure 30, The Assembled Vehicle

3.5 Vehicle Buoyancy

A great deal of detail has been given regarding the design of the vehicle, but little mention has been made of the need for buoyancy for the vehicle. For the correct operation of the vehicle it must be trimmed in such a way that the vehicle is level when floating in the water, and is slightly positively buoyant. Underwater vehicles are normally made slightly positively buoyant so that they will return to the surface in the event of a systems failure, or a cut or damaged umbilical. During the testing stages of the prototype configurations foam floatation material was used to provide buoyancy for the vehicle. This had the advantage of being able to be easily cut and shaped to fit to the vehicle. The buoyancy could also be easily added and taken away from the vehicle, being attached by a system of straps. To trim the vehicle further small lead weights could be added to locations around the vehicle to give a level trim. For the final vehicle a more professional and permanent method of mounting the buoyancy to the vehicle was needed. To determine the amount of buoyancy required, an estimate of the vehicles mass and displacement was made, from both the solid models created by Autocad and by physically measuring the mass of objects, including the box contents.

Component	Mass (kg)	Displacement(cm^3)	Buoyancy(kg)
Frame	**25.34**	**10020**	**-15.32**
Thruster Box	**41**	**35600**	**-5.4**
Manipulator Box	**43**	**35600**	**-7.4**
Vision Box	**36**	**27600**	**-8.4**
Sensor Box	**24**	**31900**	**+7.9**
Network Box	**41.5**	**28000**	**-13.5**
Internal Frame 1	**8.1**	**1032**	**-7.068**
Internal Frame 2	**9.0**	**1146**	**-7.854**
Internal Frame 3	**3.04**	**3870**	**-0.803**
Camera System	**3.85**	**3500**	**-0.35**
Laser System (2)	**7**	**8042**	**+1.042**
Manipulator Arm	**30**	**7000**	**-23**
Thrusters (6)	**10.4**	**10800**	**+0.4**
Total	**282.23**	**204110**	**-78.12**

Table 1, The Aquasphere's Buoyancy

As can be seen from Table 1, a total of 78.12kg of buoyancy had to be found for the Aquasphere. In addition, to maintain the trim of the vehicle the buoyancy boxes would have to be arranged so as to give maximum buoyancy around heavy items such as the robotic arm.

Due to the unusual shape of the vehicle, and the mounting of the existing components, there were no large areas left unfilled within the Aquasphere to give buoyancy. It was decided that the buoyancy for the vehicle was to be mounted within the exoskeletal frame, despite the greater difficulty this posed, as greater protection would be provided by the frame. Aluminium boxes were designed to fit into what spaces were left around the vehicle. The aluminium boxes were made from sheet metal, folded into a box shape, with riveted sides. A detachable top panel was made to allow the boxes to be filled with buoyancy material. A simple "U" clamp was bolted to the sides of the boxes to allow the boxes to be fixed to the vehicle. Eight 25cm x 15cm x 12cm boxes were made, which were affixed in pairs by the thrusters at the top, bottom and two nearest thrusters to the manipulator arm. One large box 55cm x 55cm x 8cm was fixed to the frame directly above the vision box, and another box 35cm x 55cm x 20cm was mounted behind the networking box. The locations of the buoyancy boxes can be seen in Figure 31.

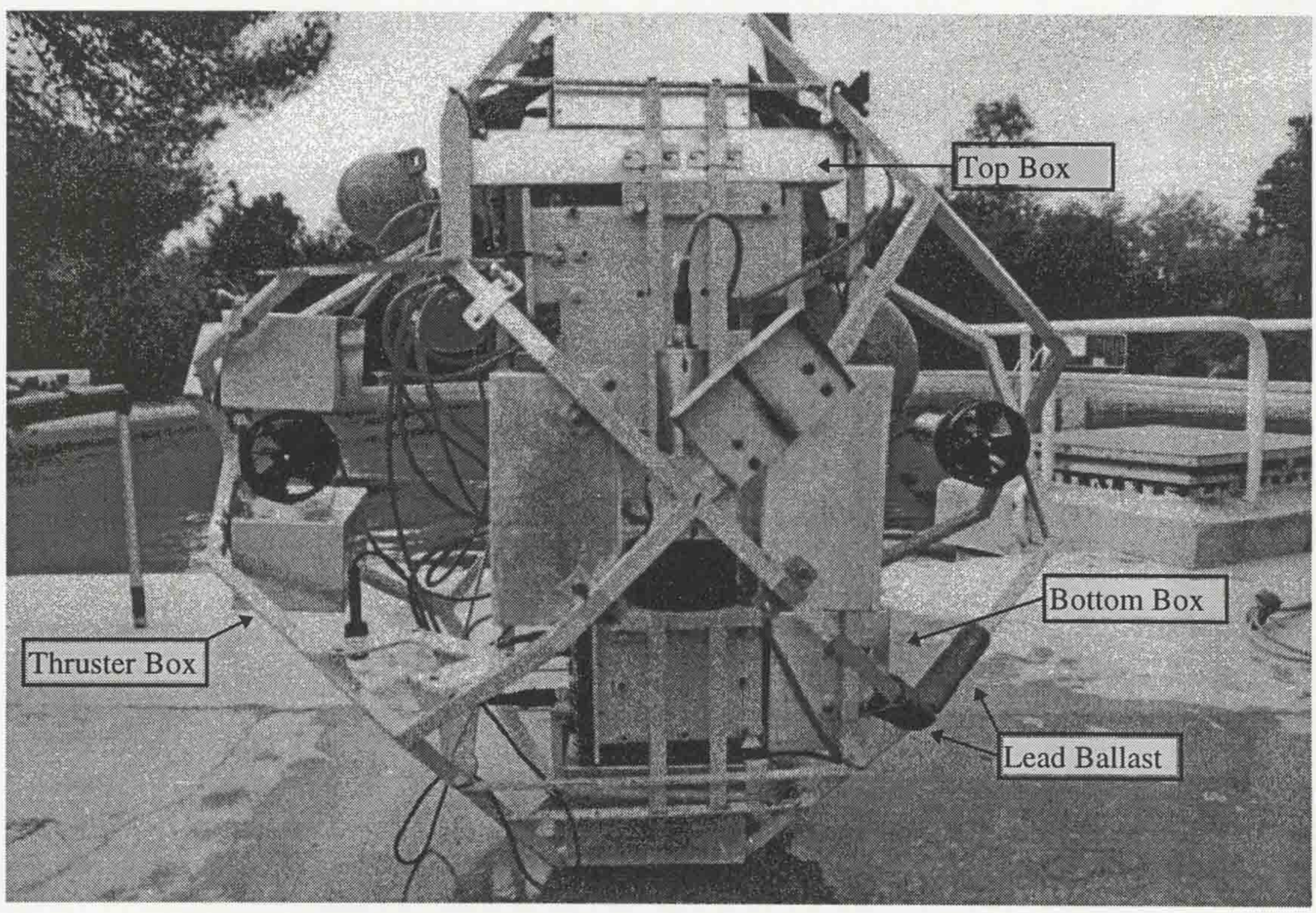

Figure 31, Vehicle Buoyancy

Component	Mass(kg)	Displacement(cm^3)	Buoyancy(kg)
Vehicle	**282.23**	**204110**	**-78.12**
8 of Thruster Boxes (25*15*12cm)	**7.2**	**36000**	**28.8**
1 of Top Box (55*55*9cm)	**5.5**	**27225**	**21.72**
Bottom Box (55*35*20)	**7.7**	**38500**	**30.8**
Total	**325.4**	**305836**	**3.2**

Table 2, Required Buoyancy Solution

During final testing it was found that the buoyancy material contained within the buoyancy boxes was not suitable. The foam material within the boxes was found to compress at depth, reducing the buoyancy of the vehicle. A positive feedback effect with increasing depth reducing buoyancy was experienced. Therefore, the foam material within the boxes was replaced with a less-compressible plastic material. The greater density of this material meant that additional volume was needed to achieve the buoyancy required, so two large blocks of this more dense buoyancy material were bolted to the top of the vehicle. The additional buoyancy did not need to be mounted within an enclosure, as it had sufficient strength to be mounted alone. In addition a sphere of buoyancy material 20cm in diameter was mounted beneath the manipulator arm.

Component	Mass(kg)	Displacement(cm^3)	Buoyancy(kg)
Vehicle	**282.23**	**204110**	**-78.12**
8 of Thruster Boxes (25*15*12cm)	**14.4**	**36000**	**21.6**
1 of Top Box (55*55*9cm)	**10.9**	**27225**	**16.33**
Bottom Box (55*35*20)	**15.4**	**38500**	**23.1**
Additionally Mounted Buoyancy	**44.9**	**73679**	**28.7**
Total	**367.83**	**379515**	**11.61**

Table 3, The Final Buoyancy Solution

As can be seen from Table 3 with the addition of the extra buoyancy the trimming of the vehicle is now possible..

3.6 The Completed Aquasphere

After its many prototype constructions and versions the final vehicle, seen being tested in Marseilles, France, is shown in Figure 32. The additional buoyancy detailed in the section *Vehicle Buoyancy* can clearly be seen affixed to the top of the vehicle. All the components of the vehicle can be seen mounted to the vehicle, including the manipulator arm holding a grappling hook.

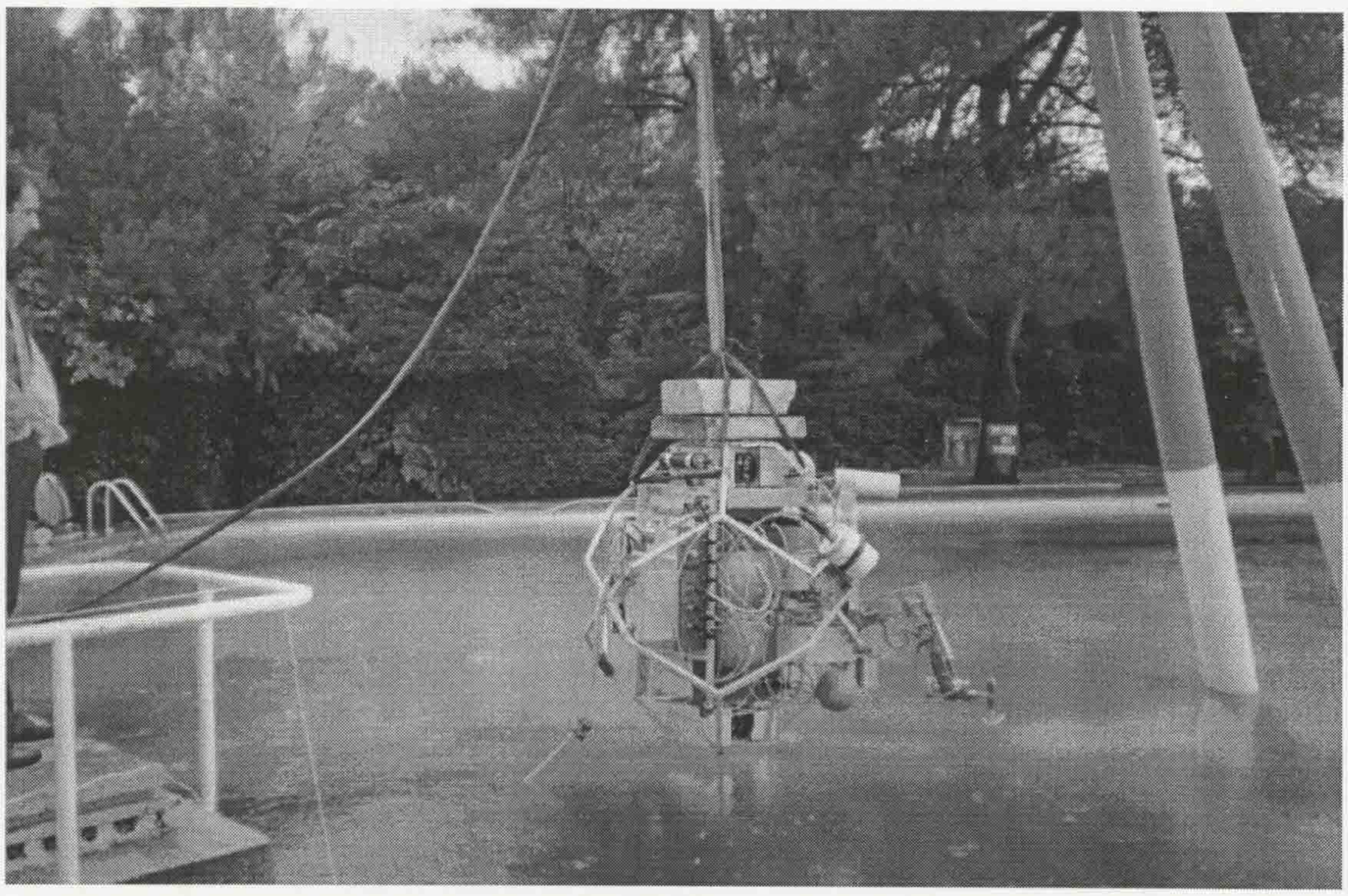

Figure 32, The Completed Aquasphere

LIVERPOOL UNIVERSITY LIBRARY

4. Thruster Control and Drive

As described in chapter 3, six thrusters must be mounted on the vehicle to be able to fully control the vehicle in six-degrees of freedom. For accurate stabilisation and navigation of the vehicle it was necessary to be able to control the thrusters with a variable speed bi-directional drive unit. This drive unit must also be able to be interfaced easily into the networking system, and require the minimum of processing power from the network. The motor drives should also be efficient since all power is supplied down the umbilical cable, and they should be as small and light as possible.

4.1 Thruster Specification

The thrusters used on this project were supplied by Hydrovision[4.12], as used on the Hyball ROV. The thrusters are driven by a two-pole permanent magnet brushed dc motor without skew manufactured by Marx.

The motors are rated at 25V 20Amps, and are very compact for a motor of this power, being only 9.5cm long with a diameter of 5cm. The motor's drive shaft is connected to an 8:1 reduction gearbox which then transfers the power to an output shaft. The output shaft is sealed by an o-ring to give waterproofing for the thruster, and the gearbox is also packed with grease to help eliminate water ingress. The output shaft then leaves the centre of the thruster, and a propeller is fixed to the shaft. The combination of motor cogging, gearbox friction and friction from the O-ring waterproof seal results in a thruster having very high stiction[2]. This means that the torque needed to start the motor turning is very high. The propeller is a four-bladed plastic propeller 12.5 cm in diameter, with a 30 degrees pitch angle and approximately 60 percent blade area coverage. The blades are flat surfaced fixed to a central boss 4cm in diameter and are designed to snap if an object severely fouls the thruster, reducing the overall damage, particularly to the expensive parts of the system. To seal the thruster at the rear a stainless steel cowling was built, as described in Chapter 3. Power for

[2] Stiction = Static Friction. There are two coefficients of friction for an object, the friction required to get an object moving from rest (Static Friction or Coulomb Friction) and the friction encountered whilst the object is moving (Dynamic Friction). Normally these values are similar, so only one definition "Friction" is used. However, when these variables are significantly different it is important to treat them as two separate values.

the thruster was carried via a cable from the thruster control box, connecting to the thruster with a "Jupiter" underwater connector. The propeller is mounted within a duct which increases the dynamic and static efficiency of the thruster, and also protects the fan blades.

A picture of the thruster mounted on the Aquasphere is shown in Figure 33.

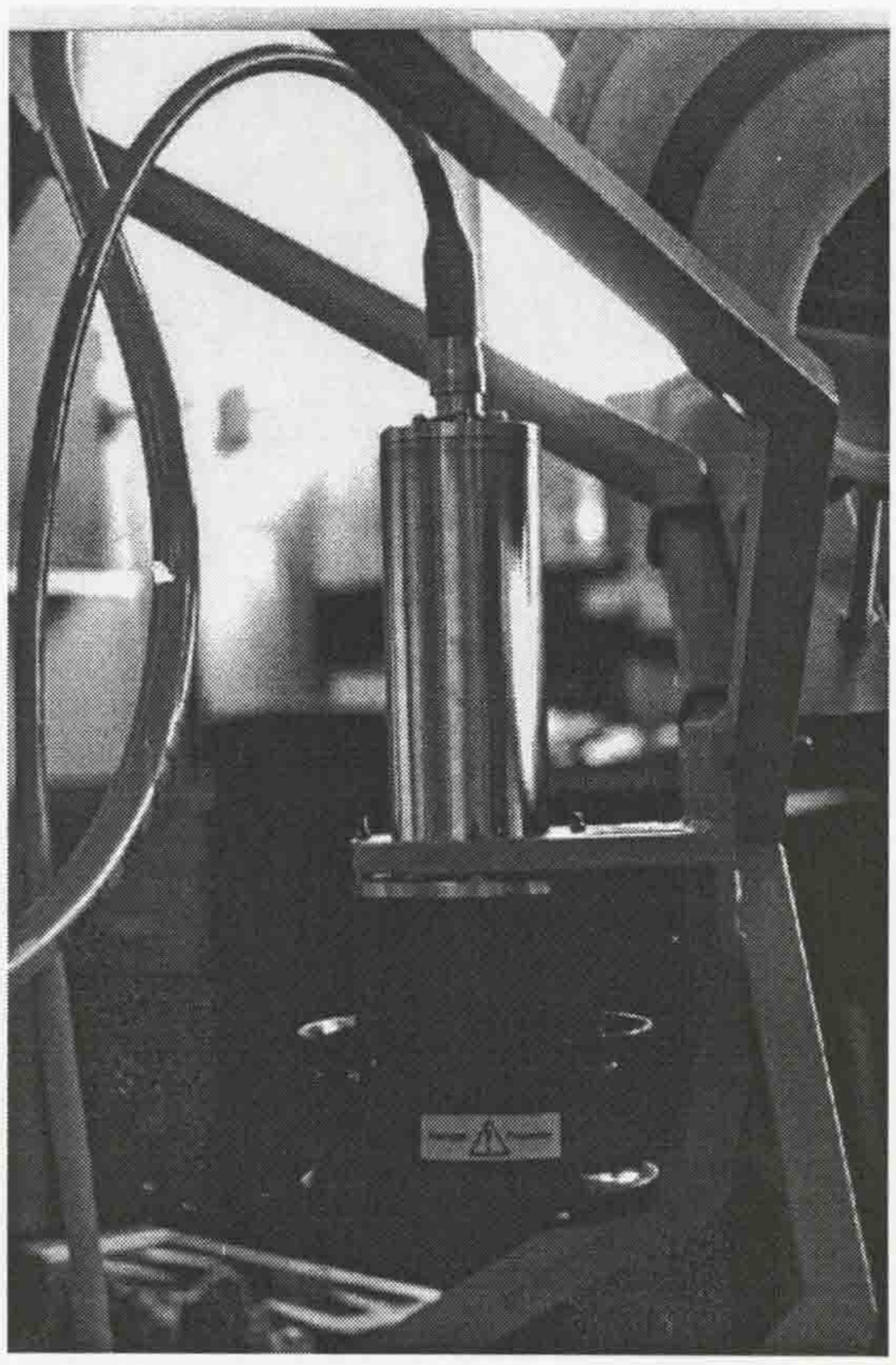

Figure 33, The Mounted Thruster

4.2 Thruster Testing

The characteristics of these thrusters used presented potential problems for the thruster control .

Yoerger *et al*[4.2] state "The automatic control of underwater vehicles represents a difficult design problem due to the nature of the dynamics of the system to be controlled. Controllers based on simple models of vehicle mass and drag usually yield disappointing performance." Yoerger[4.2] states "that for a wide class of vehicles, the dynamics of the thrusters dominate the control problem and must be properly considered to obtain good

results". Therefore, for the overall control of the vehicle to succeed it is important to have an accurate model for the thrusters.

To determine how to control and drive the thrusters a series of tests were carried out on a bench dynamometer and in the high speed water flume in the Department of Mechanical Engineering at the University of Liverpool designed by Millward *et al* [4.1]. By combining the results from these tests the characteristics of the thrusters can be determined.

4.2.1 Thruster Dry Tests

To be able to characterize the basic behavior of the thrusters a series of tests were carried out upon them. The first test carried out was to determine the rotational speed of the motor under no load. The second set of tests involved coupling the thruster to a dynamometer test rig to measure the torque of the motor, and the final set of dry tests were to investigate the dynamic response of the thruster . The tests were initially performed on three of the thrusters, however, as the initial results showed no significant differences between the performance of the different thrusters the data presented was obtained from tests performed on one thruster (Thruster 1).

4.2.1.1 Rotational Speed

The experiment was performed by varying the input voltage from a stabilized power supply, and measuring the shaft's rotational speed using a mechanical meter. No constraints were placed upon the current received by the motor, except for a limit of 20A to prevent damage to the motor. This current limit was not approached, as a maximum current of 4.4 amps was used at full power. The results of this test are shown in Figure 34.

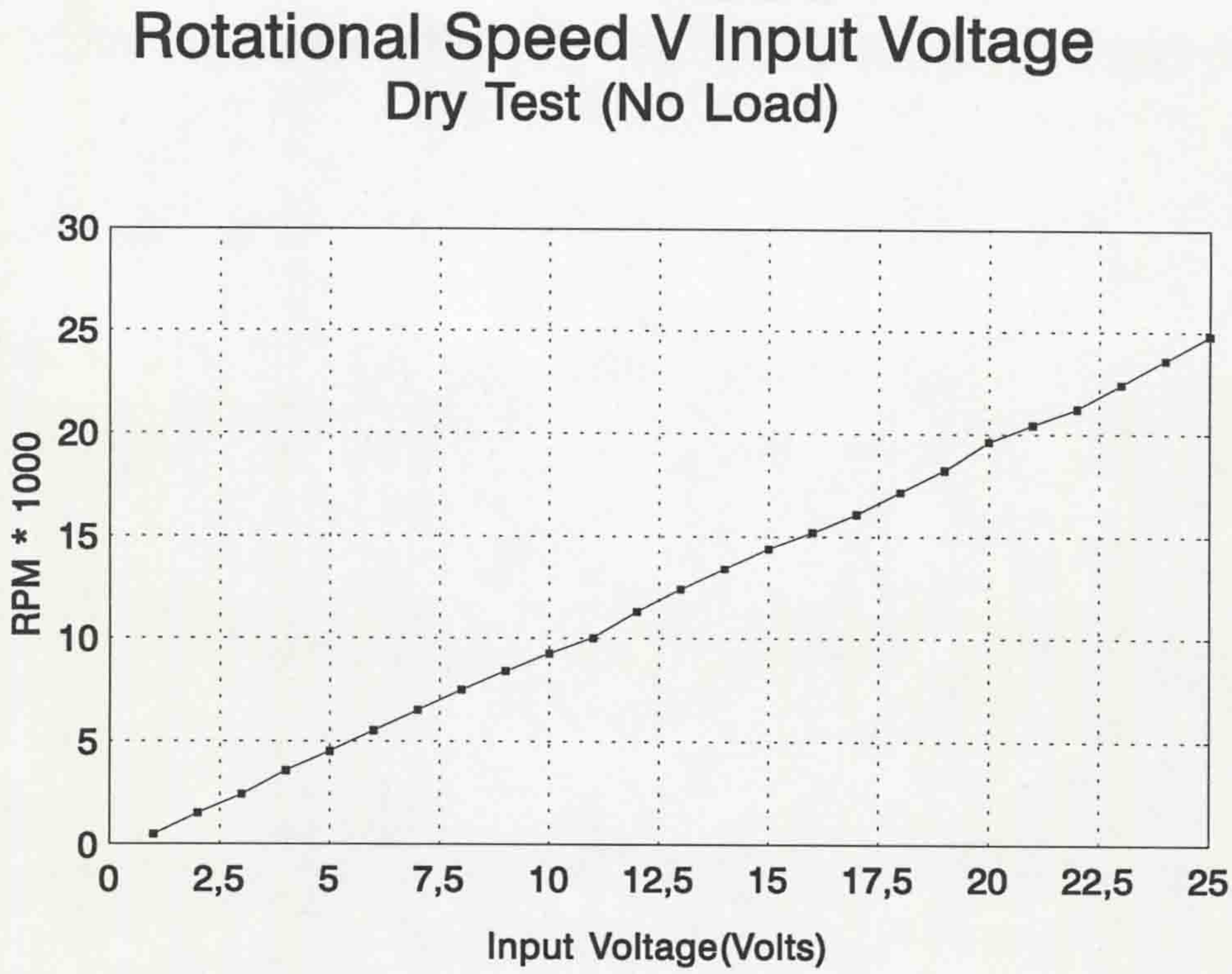

Figure 34, Rotational Speed Vs Input Voltage

It can be seen from this result that the output speed of the thruster is approximately linearly proportional to the input voltage, as expected for a brushed dc permanent magnet excited motor. In this test the rotational speed of the motor itself was measured, therefore the output speed of the thruster is reduced by the gearbox by a factor of eight. The rotational speed of the propeller can be estimated using a best linear fit from the above graph by the equation:-

$$RotationalSpeed(rpm) = \frac{(InputVoltage(v) * 987) - 524}{8} \qquad \textbf{Equation 1}$$

4.2.1.2 Motor Torque

The second dry test carried out on the thrusters was the characterisation of the motor's torque coefficient. To carry out these tests the propeller and cowling for the thruster was removed, and the output of the gearbox shaft was coupled to a dynamometer rig by a self-aligning shaft coupler. The motor was then given a limited supply of current regulated by a power supply, with the supply voltage set to 25Volts. The loads applied to the thruster were adjusted by the dynamometer apparatus to ensure that the motor did not reach stall

conditions. From the set of results taken for the thruster the graph of results shown in Figure 35 was taken.

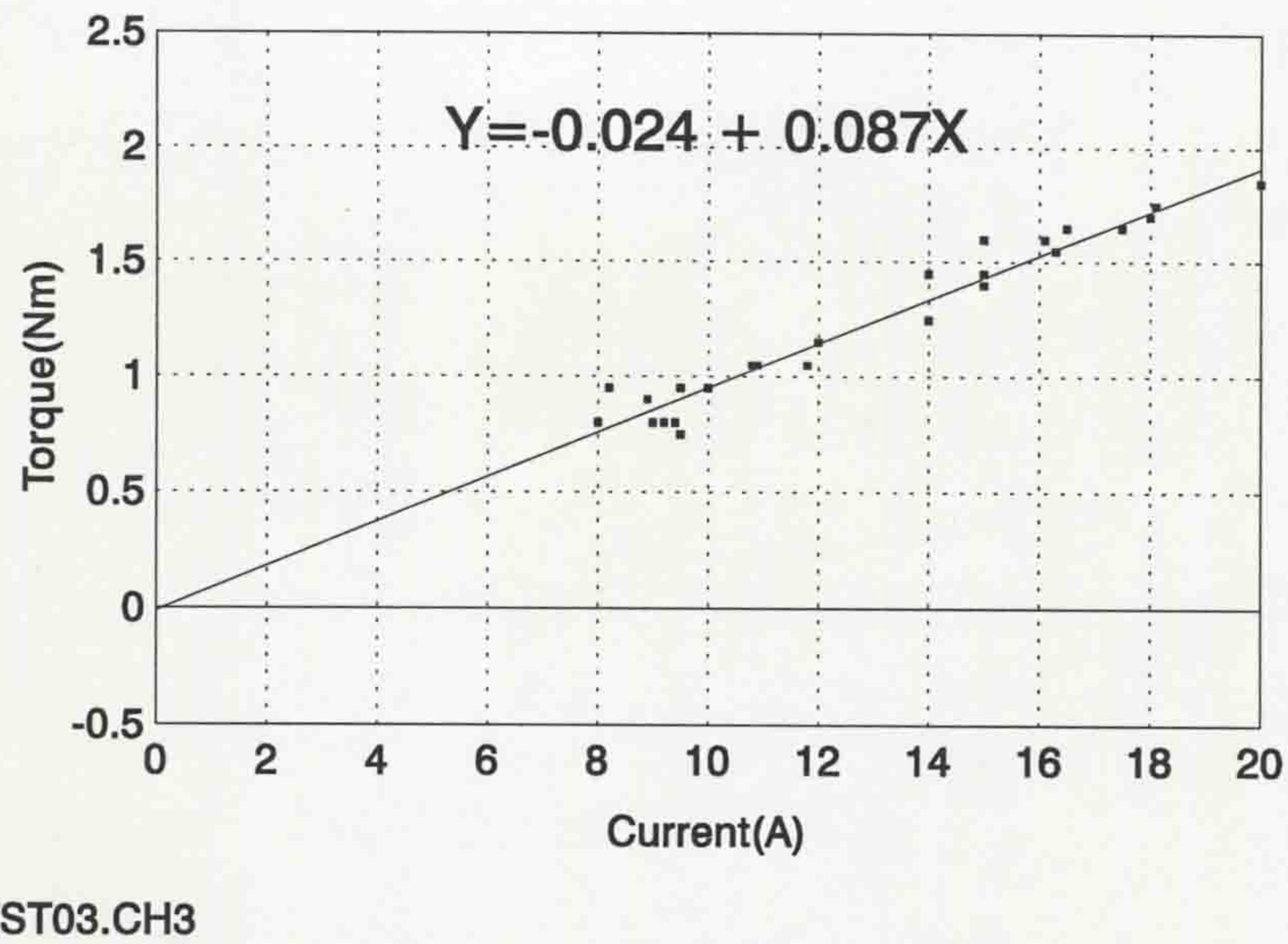

Figure 35, Thruster Torque Characteristics

From Figure 35 the torque characteristics for the thrusters can be calculated as being:-

$$OutputShaftTorque(Nm) = 0.087 * InputCurrent(A) - 0.024$$

Equation 2

As can be seen the torque of the motor is directly proportional to the input current applied to the motor. The torque readings were taken from the output shaft of the gearbox. (The torque of the motor without the mechanical advantage of the gearbox is therefore the above value divided by eight.)

Although the above results (Figure 35) shows a linear response between current and torque it fails to show the operational region of the thruster. In reality the thruster shows a large hysteresis effect at low current values, caused by the coulombic friction of the thruster

system. Also there is a minimum current that needs to be applied for the motor to turn at all, so the graph will in reality not linearly approach zero. The startup current needed for the thruster has been measured as 4.8Amps, whilst the stall current for the motor is 1.8Amps. Applying this information to the previous data set gives the modified load characteristics as shown in Figure 36.

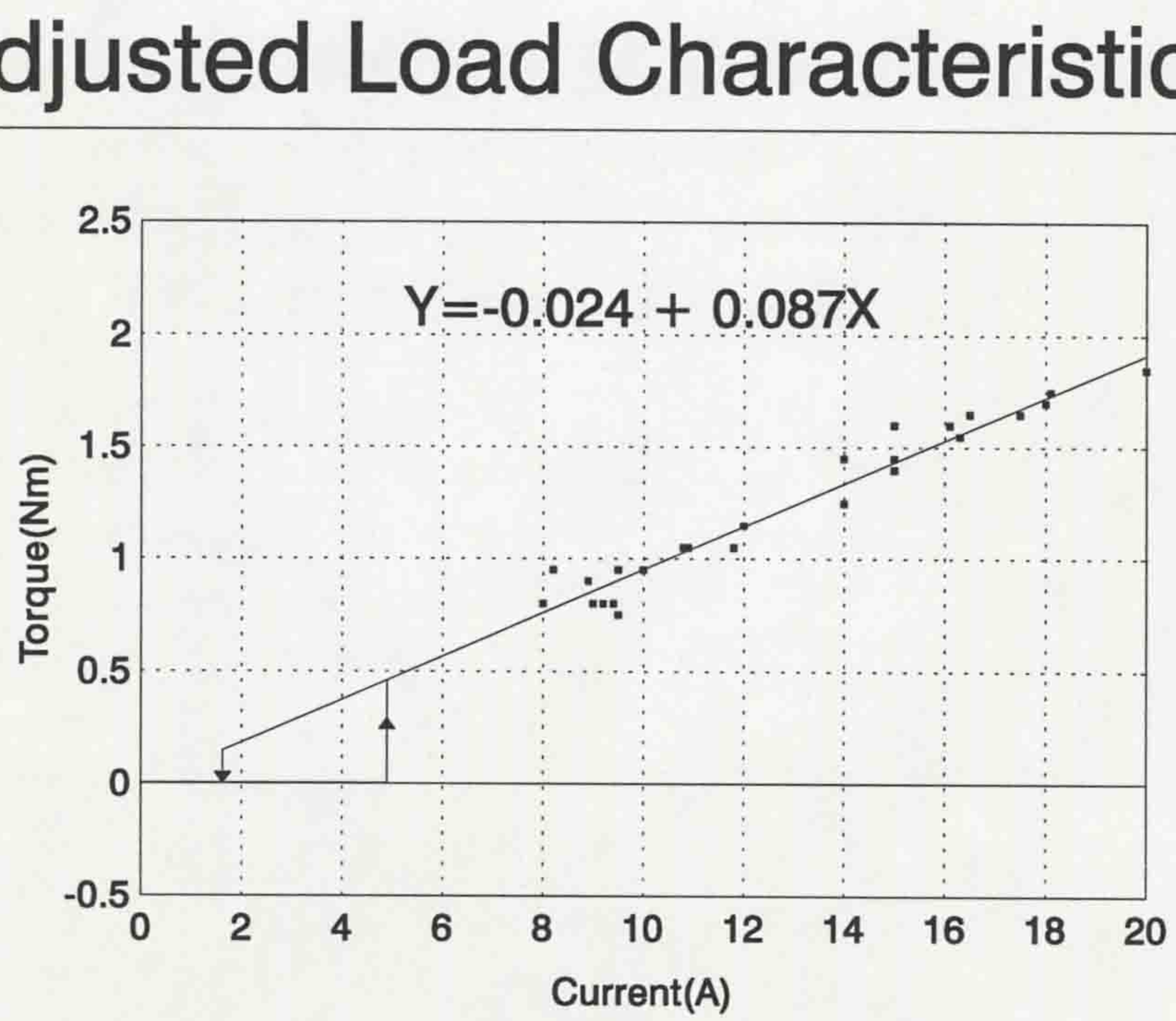

Figure 36, Adjusted Load Characteristics

The hysterisis shown is particularly important when trying to control the thruster at low speeds. Low speed control is particularly important as to maintain the hover of a vehicle constant small adjustments need to be made to the thrust from the thrusters. To overcome this the drive for the motors, or the motor control system, must be capable of taking this problem into consideration.

4.2.1.3 Thruster Back E.M.F.

To find the back E.M.F. for the motor the motor was driven as a generator, and the voltage generated was measured. This was performed for several different input speeds to give the following results.

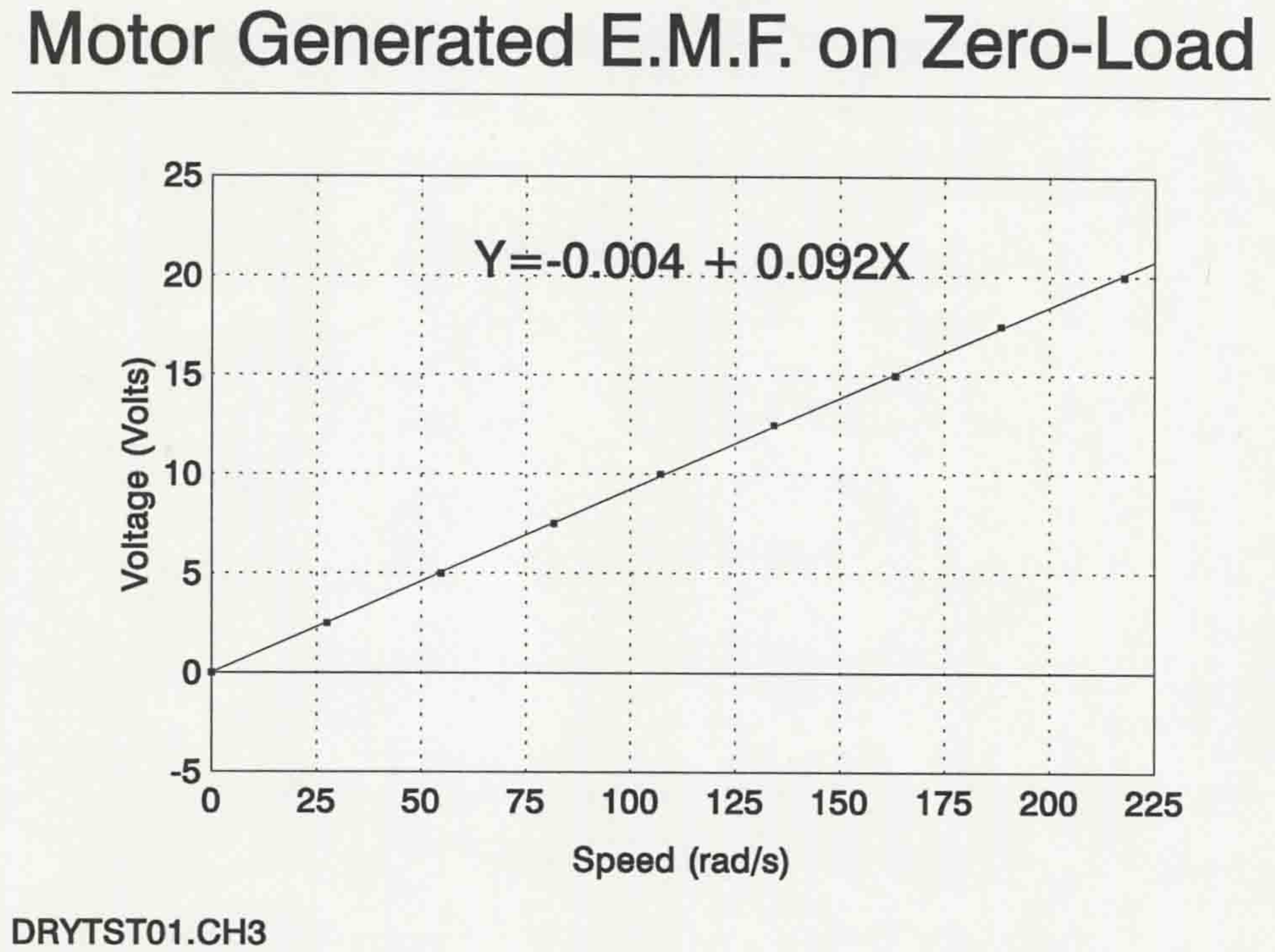

Figure 37, Motor Generated Back E.M.F.

As can be seen from the graph the motor generated E.M.F. is given by:-

$$E.M.F.(v) = 0.092 * RotationalSpeed(rads/s)$$

Equation 3

4.2.1.4 Thruster Step Response

To investigate the dynamic characteristics of the thrusters a series of step response tests were performed on the thruster. The data was recorded using a digital storage scope to measure the voltage across the motor, and the current through the motor. The current was measured using the voltage across a low-ohm shunt resistor (0.0025 ohm). From these results, and the results given by Equation **1** and Equation 2 the speed of the motor, and the

torque can be calculated. Also the time constant for the motor, the rise time and fall time for the thruster can be found. The rise and fall times were calculated from 10% to 90% of the maximum step. The results for the thruster were taken for various heights of step response, with a current limit set at 20Amps to prevent damage to the motors.

A typical plot for the step response of the thruster from rest to 9250r.p.m. (10 Volts input step) is shown in Figure 38. The Y axis is the generated back e.m.f. of the motor, while the X axis indicates the time. The scale of these axes can be taken from the data given by the scope plot.

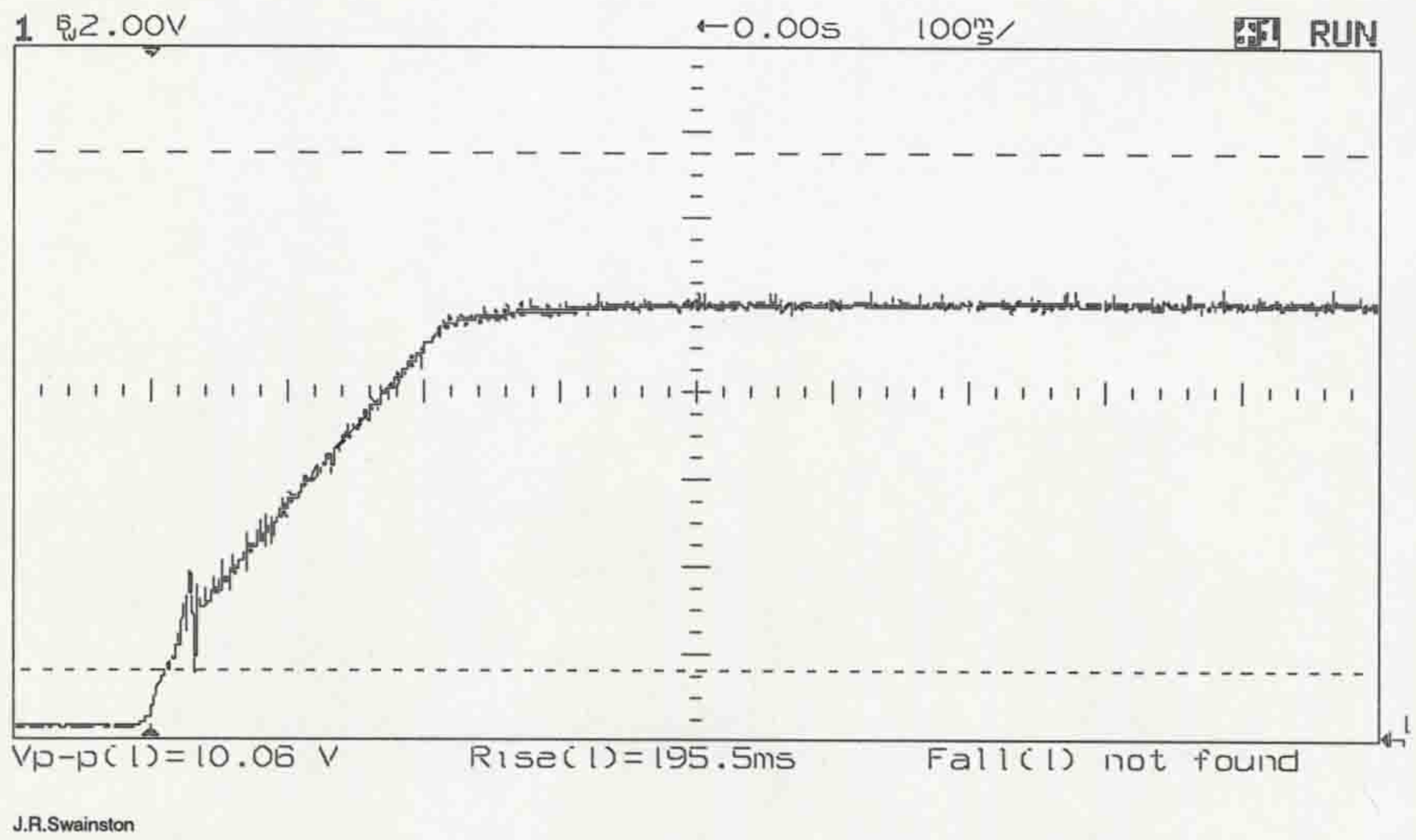

Figure 38, 10 Volt Input Step Response

As can be seen from Figure 38 the motor has a 10%-90% rise-time of 195.5ms with an approximate exponential time constant of 135ms. The motor accelerates linearly to the maximum speed, and then maintains this speed.

The corresponding plot for the current used by the motor is shown in Figure 39. This graph shows a very high current peak as the motor starts. This peak is caused by the motor stalling due to the coulomb friction of the thruster. The peak series resistor voltage of 509.4mV corresponds to 20.4Amps, indicating that the current limiting of the power supply was reached. The duration of the peak is 41ms, after which the motor starts to turn.

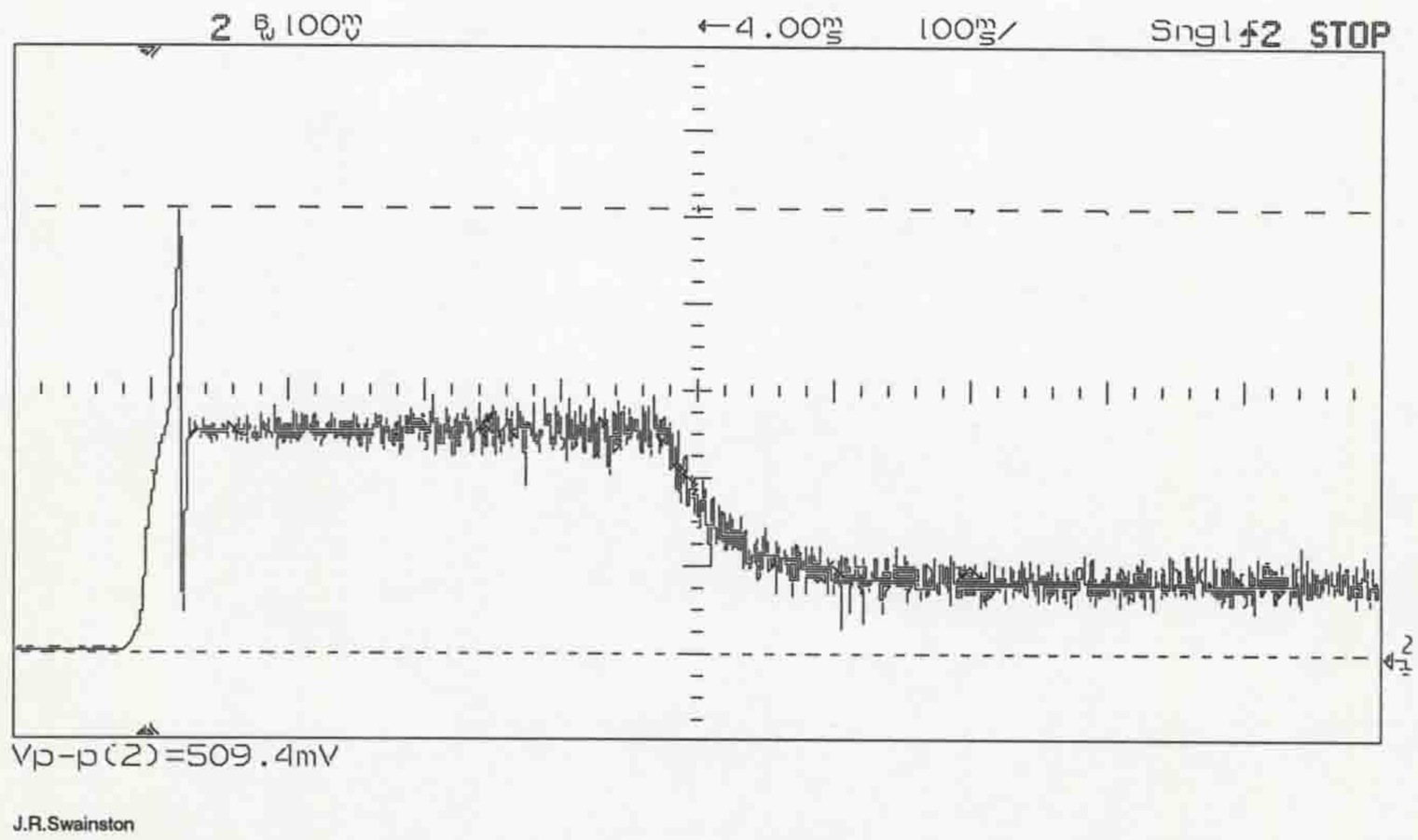

Figure 39, Thruster Current for 10volts input step

It is this current peak that can cause difficulty in the design of a power drive circuit, as the circuit must be able to cope with almost short-circuit conditions for a small period of time (typically 41ms), as shown in Figure 39.

Figure 40, shows the response to a voltage step. The e.m.f. measured is shown in the top trace, whilst the voltage across the series resistor is shown in the bottom trace. The graph shows three different regions:-

1. The Power is off, and the motor is at rest.
2. The power is turned on, the motor sticks at first, taking a large current pulse, then it accelerates as fast as it can, taking the maximum current the armature resistance will allow until it reaches maximum speed. The current then decreases to a level to sustain the speed of the motor.
3. The power is switched off, and the motor slows down. As can be seen from the graph the fall time for the motor is almost 1 second.

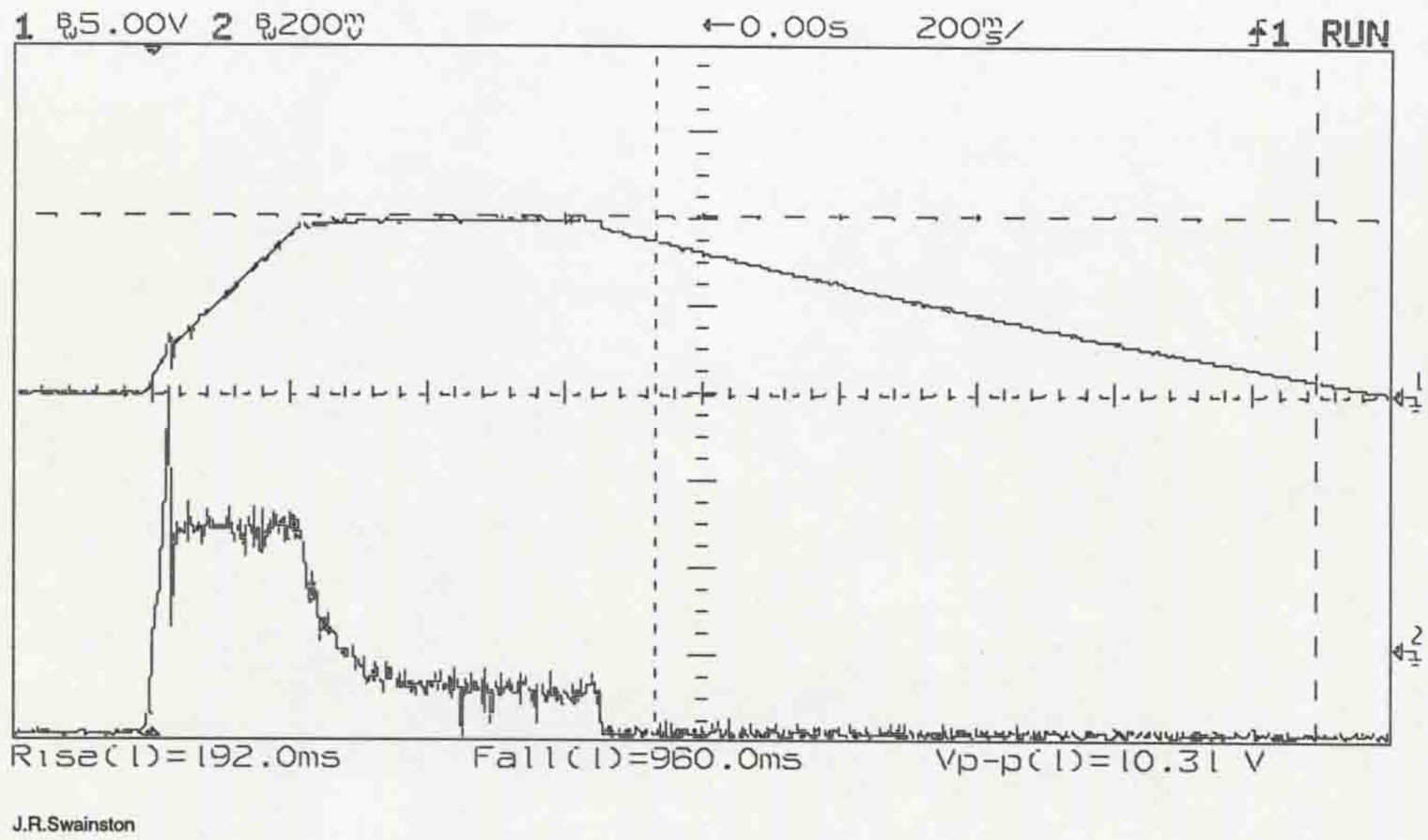

Figure 40, Thruster Step Response

The following table shows a summary of the data obtained from the step response applied to the motor in the dry, under no load.

Step Size (V)	Max Speed (r.p.m.)	Current (A)	10%-90% Risetime(s)	10%-90% Falltime(s)	T_c
10	**9250**	**3.2**	**0.192**	**0.96**	**0.16**
15	**14400**	**4.0**	**0.310**	**1.345**	**0.22**
20	**19560**	**4.2**	**0.402**	**1.748**	**0.28**

4.2.2 Thruster Wet Tests

To be able to characterise the behavior of the thrusters a set of test results were carried out on the thrusters in a re-circulating flume. The flume is located in the Mechanical Engineering Department at The University of Liverpool and is detailed by Millward *et al* [4.1]. The flume has a wide speed range, and can be used as either an open channel, a cavitation channel, a closed water channel or a cavitation tunnel. The flume is typically used for the research and development of high-speed surface vessels, sailing boats, hovercraft,

tankers and submersibles. For the purpose of testing the thruster the flume was used in the open channel configuration. The thruster was held underwater in the centre of the flume's channel by a clamp to which an hydrofoil section was attached to minimize the effects of the clamp arrangement. The thruster was coupled to a force block designed by Millward *et al*[4.4] to allow the thrust measurements for the thruster to be taken. Figure 41 shows the hull of a ship being tested in the open channel of the flume at The University of Liverpool (Photograph provided courtesy of R. Cotes, Mechanical Engineering, The University of Liverpool). A similar arrangement for the testing of the thruster was used, with the thruster being held beneath the surface.

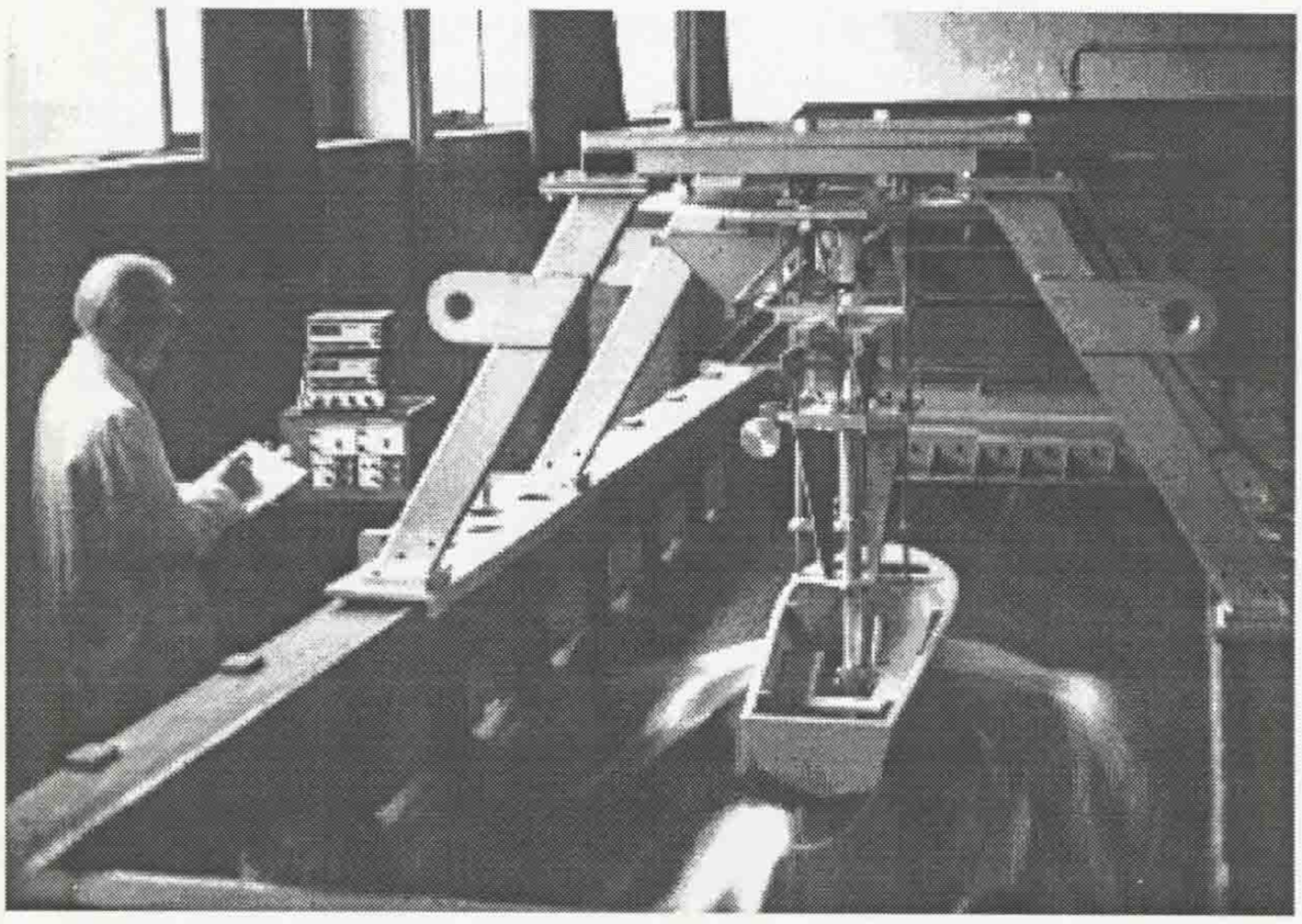

Figure 41, The Open Channel Flume

4.2.2.1 Thrust Tests

The thruster was mounted in the flume as described above. The torque of the motor was adjusted by the current of the motor, with a voltage limit set to 25 volt. The force exerted by the thruster on the force block assembly was noted, which is equal to the thrust that the thruster was exerting on the water. This test was repeated for various water flow rates through the flume to study the effects of propeller slip[3]. The results were repeated with the thruster rotated 180° to see if the results for the thruster depended upon its orientation.

[3] Propellor Slip is encountered when a propellor turns within a relatively moving medium. It is caused by the reduction of the differential of speed between the propellor face and the medium, causing the propellor to race as the medium slips past the propellor. With a moving medium the effective thrust is reduced due to this effect.

The repeated tests showed that there was no difference in backwards and forwards thrust generated by the thruster, and that the drag in both flow directions was in practice the same.

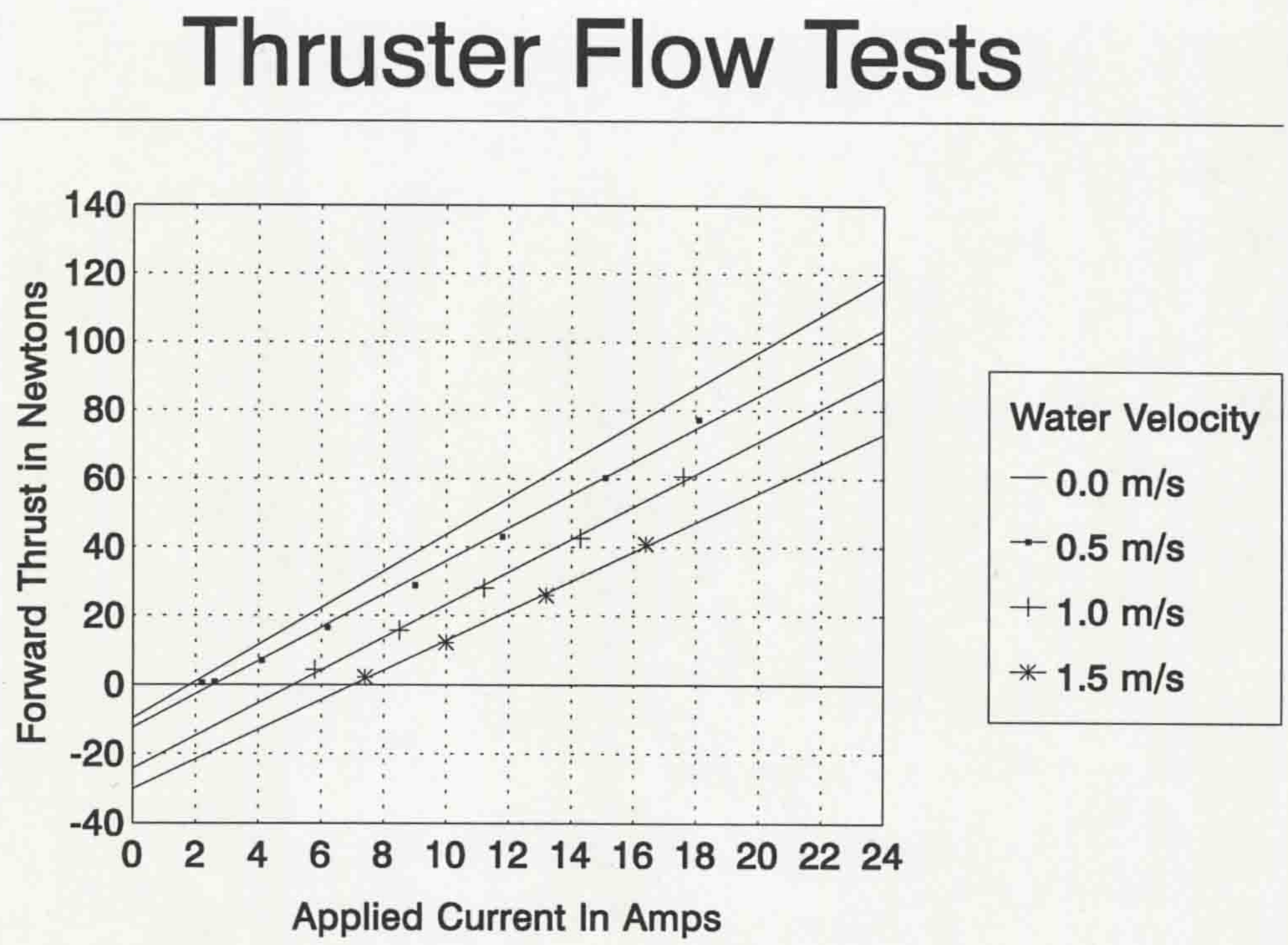

Figure 42, Thruster Flow Tests

Figure 42 shows the relationship between the input current (and therefore the torque) to the thruster and the thrust generated by it. The negative values of thrust are obtained when the thrust of the thruster is less than the drag force of the thruster through the moving water.

From Figure 42 the relationship between thruster current and thruster thrust can be approximated as:-

$$Thrust(N) = (MotorCurrent(A) * 4.67) - (VehicleSpeed(m/s) * 20)$$

Equation 4

4.2.2.2 Thruster Step Tests

To test the performance of the thrusters in a realistic manner the motor response results were repeated in water. The thrusters were mounted in their marinised cowlings and held underwater whilst a step input voltage was applied. The results showed significant

differences between these tests and the corresponding dry-test results. This is due to the damping effect of the water, and the extra work required due to energy being imparted on the water. A typical response for the thruster is seen below in Figure 43, where Y is the back e.m.f. and X is time.

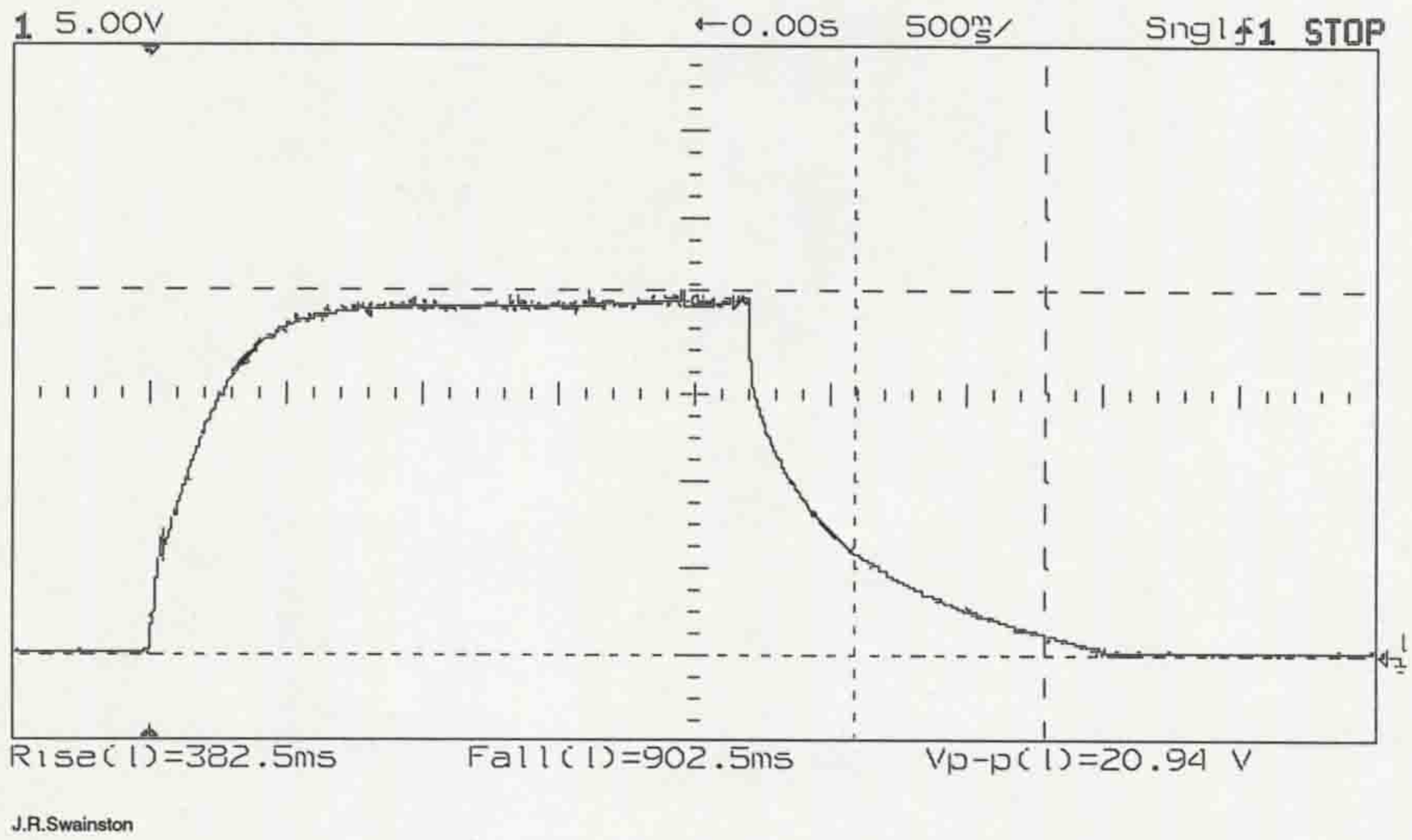

Figure 43, A Typical Step Response

Step Size(V)	Steady State Current(A)	Risetime(ms)	Falltime(ms)
5.0	**3.2**	**123**	**655**
7.5	**4.85**	**142**	**835**
10.0	**7.0**	**130**	**885**
12.5	**9.6**	**162**	**922**
15.0	**14**	**200**	**980**
17.5	**18**	**270**	**972**
19.0	**19.5**	**345**	**940**
20	**20**	**392**	**935**

4.2.3 Thruster Model

From the data presented for the thruster it was possible to build up a computer simulation of the thruster, so that testing of control systems could be tried out relatively quickly and easily without causing any damage or wear to the physical system.

To create the computer model of the thruster the computer package Matlab[4.6] was used. Matlab on its own is a powerful mathematical matrix handling tool (Matlab's name stems from "**Mat**rix **Lab**oratory"), and a wide range of "Toolboxes" have been produced to provide a graphical front-end for the underlying matrix function mathematics provided by the Matlab package.

A toolbox for Matlab called "Simulink" was used to aid in the creation of the thruster model. Simulink allows basic control system blocks to be added together to create a more complex overall model. Its modular design means that complex simulations can be constructed from a graphical user interface, without the need for the underlying code to be written. Simulink also allows sub-systems to be created and embedded within the overall design. This means that many levels of design can be incorporated into a single "block" which can be used as a "black box", or the user can enter into the underlying levels and explore the system detail. The simulation was constructed with sub-systems which correspond directly with the real hardware devices.

The listing for the thruster model in contained in Appendix 10.3 **Simulink Thruster Model**.

4.2.3.1 Model Top Level

At the top level for the simulation a simple structure was created to link together the component parts of the thruster system. The simulated thruster demand signal is linked into the model for the power drive system. The output voltage from the power drive is connected into the model for the thruster, in addition to the velocity of the vehicle. This is because the output thrust of the thrusters is dependent upon the vehicle's velocity, as well as the motor's parameters as detailed in Section 4.2.2.1. The value for the vehicle's speed could be taken from a model of the entire vehicle, of which this thruster model would be a sub-system. To display the response of the system graphical displays for the current taken by the thruster, the thruster's rotational speed, the thrust generated, and the demand signal are connected. Although the variables for motor velocity and current are not required by an overall control

system they serve as useful indicators to verify the model against the actual test data. The schematic diagram for the top level is shown in Figure 44.

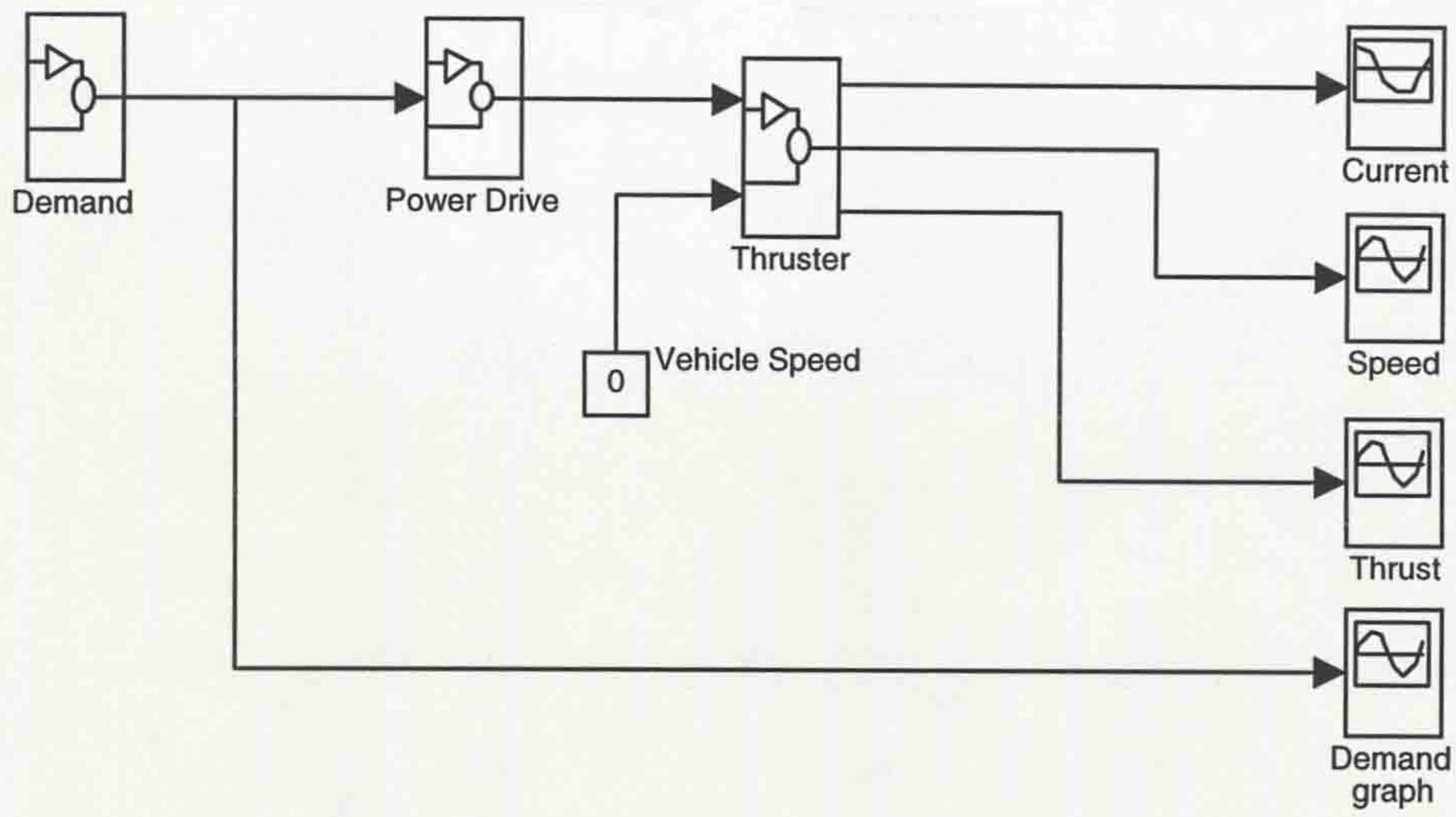

Figure 44, Thruster Model Top Level

4.2.3.2 Thruster Demand Model

The demand signal received via the Network on the real vehicle is a ±10 volt signal. To recreate this signal a simple sub-system was devised within Simulink. The bipolar output from a signal generation block was added with a dc. bias to give a demand signal. This signal is then limited to ensure that the signal falls within the ±10volt limit that the hardware supports. This signal then forms the "Demand" signal output. The schematic representation of the demand signal is shown in Figure 45. The form of signal from the Signal Generator unit can be set to give a square wave, ramp or sinusoidal demand,

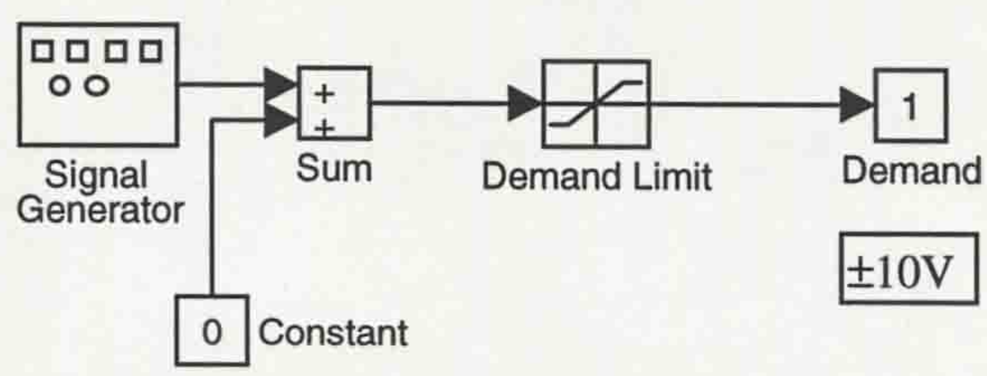

Figure 45, Thruster Demand Sub-system

4.2.3.3 Power Amplifier Model

The signal from the thruster demand is connected into the power drive, as it is in the real vehicle. A simple linear amplifier simulates the average voltage output of the thruster drives. The PWM is not modeled in detail as the computation involved in modeling a 20kHz signal would cause the simulation time to increase to an unacceptable level. Figure 46 shows the schematic diagram of the linear amplifier used to drive the thruster for the thruster modeling.

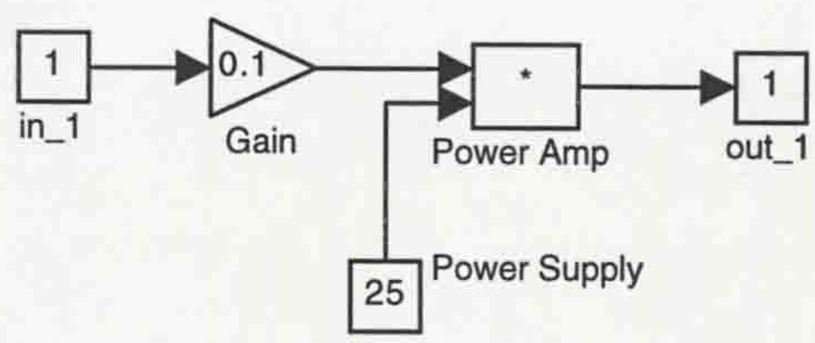

Figure 46, The Power Drive

The input is reduced to ±1 and multiplied by the supply voltage to give the output voltage from the power drive.

4.2.3.4 Thruster Model

The largest and most complex part of the model is for the thruster itself. The model for the thruster is based on the standard second order model for a motor, a model for the coulomb friction of the motor, combined with the Lumped Parameter Model for the propeller system designed by Yoerger and Slotine[4.2]. By combining these three models the overall model for the thruster could be found.

4.2.3.4.1 Stiction Model

In order to model the effects of stiction (coulomb friction) on the thruster performance a separate model for the stiction was created. Due to the hysterisis of the motor two separate models for the stiction were used, with the motor's speed determining which model is applicable.

Two stiction models were created. The first is for where the motor was at rest and the current is increased, and the second for when the motor changes direction, so the motor passes through the zero speed point. For each case the input current is compared to the value at which the motor would stall, 1.8amp for when there was motion and 4.8amps when the motor is started from rest. This is detailed in section 4.2.1.2. If the current signal is in the region in which the motor would stick an output value of zero is sent, otherwise the motor current is outputted.

A selection switch with |Speed| as its input selects which of the two stiction models is used. If the speed is non-zero the moving model is chosen, else the static model is selected. A block diagram of the Stiction Model can be seen in Figure 47.

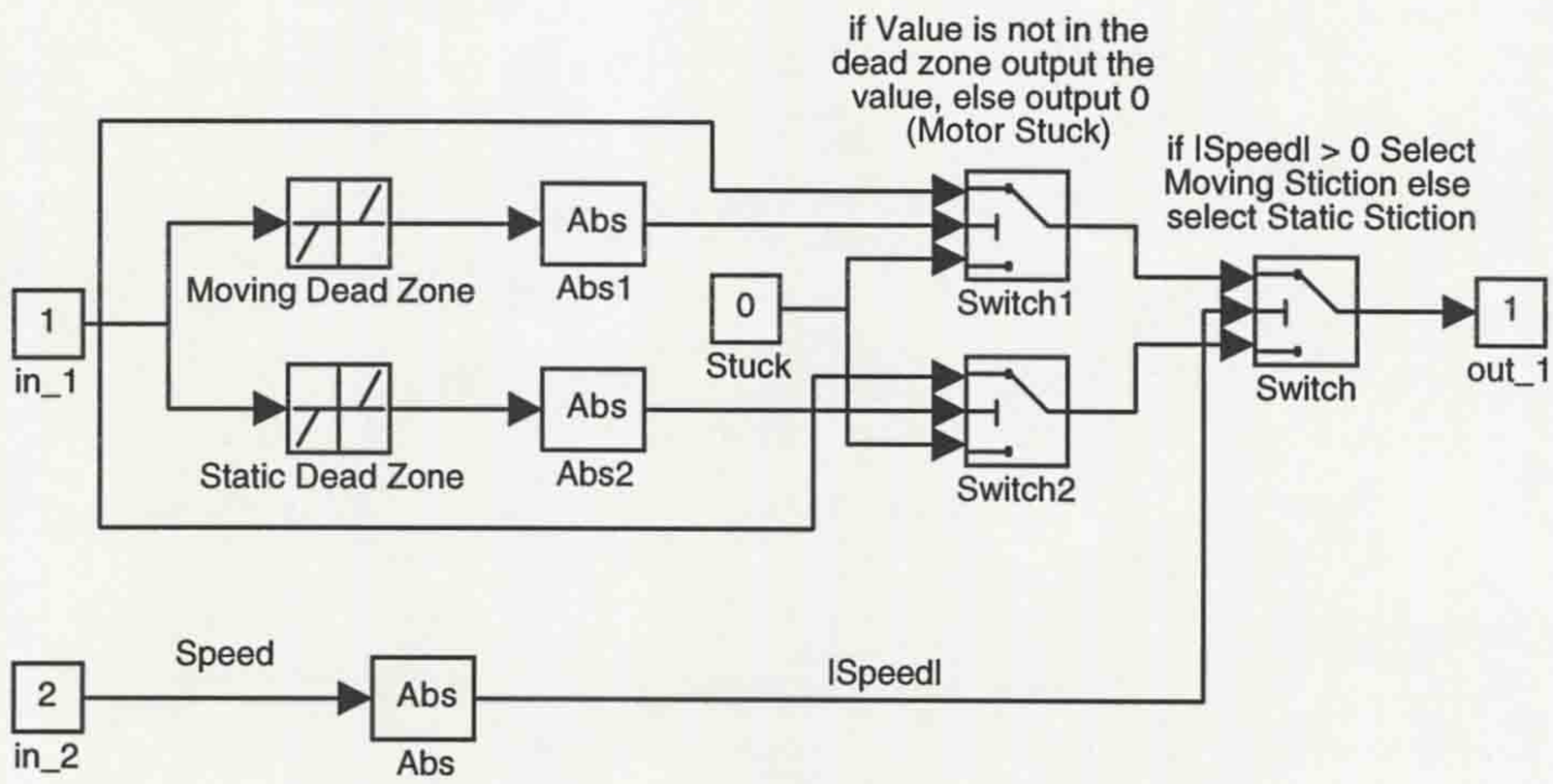

Figure 47, The Stiction Model

4.2.3.4.2 Motor Model

The standard representation for modeling a D.C. motor as given in most standard control books (e.g. Dorf[4.5]) and is shown in Figure 48.

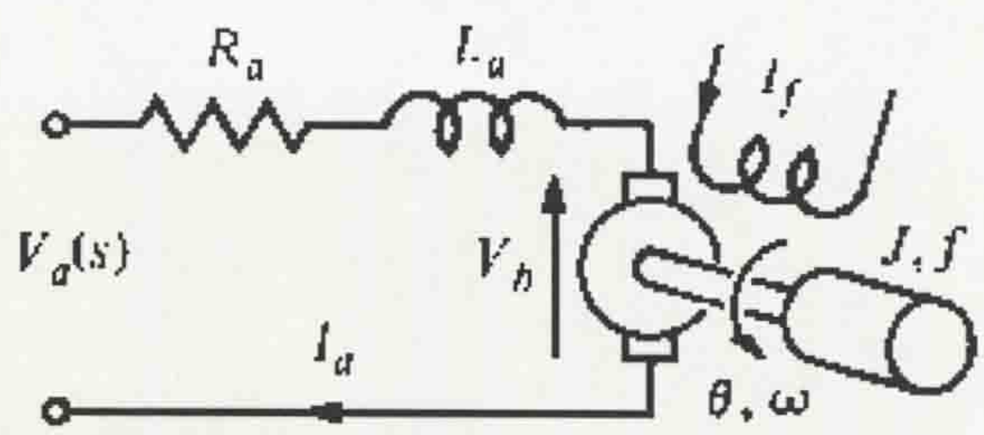

Figure 48, Motor Representation

Where:-

R_a	= Armature Resistance
L_a	= Armature Inductance
I_a	= Armature Current
V_b	= Back emf voltage
I_f	= Field Current
J	= Moment of Inertia
f	= Viscous Friction Coefficient
ω	= Speed
θ	= Position

For an armature controlled D.C. motor the torque of the motor can be expressed as:-

$$T_m(s) = (K_1 K_f I_f) I_a(s)$$

Equation 5

For a permanent magnet motor the field current I_f is constant, so torque can be expressed in terms of one variable and one constant K_m (the motor constant).

$$T_m = K_m I_a(s)$$

Equation 6

The current through the armature (I_a) is related to the input voltage (V_a) by the expression:-

$$V_a(s) = (R_a + L_a s) I_a(s) + V_b(s)$$

Equation 7

The electromotive-force back-voltage V_b is proportional to the motor speed giving:-

$$V_b(s) = K_b \omega(s)$$

Equation 8

Equating Equation 7 and Equation 8 gives the armature current (I_a) as :-

$$I_a(s) = \frac{V_a(s) - K_b\omega(s)}{R_a + L_a s}$$

Equation 9

The torque of the motor (T_m)is equal to the load torque (T_L) plus the disturbance torque (T_d). However, assuming the disturbance torque is much less then the load torque the torque of the motor is equal to the load applied to it. This load is principally supplied by the resistance of the physical system and its inertia (T_{Lmot}), the stictional load (T_{Lstick}) and the load caused by the propellor (T_{Lprop}).

$$T_m(s) = K_m\left[\frac{V_a(s) - K_b(s)}{R_a + L_a s}\right] = T_{Lmot}(s) + T_{Lprop}(s) + T_{Lstick}(s)$$

Equation 10

The load of the system caused by the friction of the motor and its inertia can be defined as:-

$$T_{Lmot}(s) = Js^2\theta(s) + fs\theta(s)$$

Equation 11

The inertia of the system can be worked out from the time constant of the step response from the equation:-

$$T_{water} = \frac{JR_a}{K_b^2}$$

Equation 12

From Equation 3 $K_b = 0.092$, and from the step tests $T_{water} = 0.13$ seconds. Measuring R_a at 0.24 ohms gives a value for J of:-

$J = 4.584e\text{-}3kgm^2$

4.2.3.4.3 Lumped Parameter Propeller Model

The model presented here, devised by Yoerger and Slotine[4.2], is based upon a propeller driven by a torque source(τ) at angular velocity (Ω). The thruster shroud has a cross-sectional area (A) and encloses a volume (V). The density of the ambient fluid is designated (ρ) and the volumetric flowrate of the thruster is (Q).

Yoerger makes several assumptions so as to simplify the equations for the overall model:

- The energy stored is due solely to the kinetic energy of the fluid in the duct.
- The kinetic energy of the external ambient fluid is negligible (i.e. the fluid is at rest)
- Friction Losses are negligible.
- Ambient fluid is incompressible.
- Fluid flow at the thruster intake and exhaust is parallel, one-dimensional, and at ambient pressure. Rotational Flow effects are ignored.
- Gravity effects are negligible.
- The thruster is completely symmetric with respect to the flow direction.

The standard equation for the kinetic energy of a body is:-

$$k.e. = \frac{1}{2}mv^2$$

Substituting the previous variables the kinetic energy of the fluid within the thruster cowl can be expressed in terms of Q as :-

$$E^*(Q) = \frac{1}{2}\rho V\left[\frac{Q}{A}\right]^2$$

The general equation for momentum is mass * velocity. This can be expressed in terms of Q by:-

$$\Gamma = \frac{dE^*}{dQ} = \rho V\frac{Q}{A^2}$$

Equation 13

As this function has units of momentum/area it is commonly referred to as the pressure momentum.

Due to the conservation of energy, the energy of these two are equal. Hence the kinetic energy (Q) can be expressed as:-

$$E(\Gamma) = \frac{A^2}{2\rho V}\Gamma^2$$

Equating these in the terms of pressure momentum gives:-

$$\frac{dE}{dt} = \frac{A^2}{\rho V}\Gamma\dot{\Gamma} = \Omega\tau - KQ$$

Equation 14

Where:-

$\Omega\tau$ = The power input from the thruster propeller

K = The exciting kinetic energy per volume.

$\dot{\Gamma}$ = the rate of change of the pressure momentum with time.

The exciting Kinetic energy per volume (K) can be expressed as:-

$$K = \frac{A^2\Gamma^2}{2\rho V^2} = \frac{\gamma^2}{2\rho}$$

Where $\gamma \equiv A\Gamma / V$ is the fluid momentum per volume within the thruster.

The thruster and ambient fluid are coupled by linear momentum, which is equal to the thrust developed, giving:-

$$Thrust = \gamma Q$$

Equation 15

Assuming that there is no cavitation the flowrate of the thruster can be expressed as a function of the angular velocity Ω and the thruster/propeller characteristics. The difference between the expected advance and the actual advance is the slip and is typically expressed as a ratio σ.

$$\sigma = \frac{\Omega pA - Q}{\Omega pA}$$

where p is the pitch of the propeller (the axial distance traveled by the propellor blade per radian).

From the above equation $Q(\Omega)$ can be expressed as:-

$$Q(\Omega) = \eta pA\Omega$$

Equation 16

where $\eta \equiv 1-\sigma$ and is referred to as the propeller efficiency.

From Equation 13, Equation 14 and Equation 16

$$\dot{\Gamma} = \frac{\tau}{\eta p A} - K$$

For a fixed propeller system pitch (p) is constant, as is the area of the shroud (A). If we assume that the propeller efficiency (η) is constant throughout its range then the thruster equations can be expressed in terms of Ω as:-

$$\dot{\Omega} = \frac{\tau}{\eta^2 p^2 \rho V} - \frac{\eta p A}{2V}\Omega|\Omega|$$

Equation 17

$$Thrust = A\rho\eta^2 p^2 \Omega|\Omega|$$

Equation 18

Equation 17 can be further simplified by reducing the constants to give the lumped parameter model of the thruster:-

$$\dot{\Omega} = \beta\tau - \alpha\Omega|\Omega|$$

Equation 19

where τ is the input torque. From the steady state results of the wet tests the relationship between α and β can be determined.

$$\dot{\Omega} = 0 = \beta\tau - \alpha\Omega|\Omega|$$
$$\therefore \beta\tau = \alpha\Omega|\Omega|$$
$$\beta/\alpha = \frac{\Omega|\Omega|}{\tau}$$

Equation 20

Using the steady state values from the full power state:-

Ω = 201 radians/sec

τ = 1.9 Nm^{-1}

Giving β/α = **21263**.

Equation 18 can be simplified to give the lumped parameter model.

$$Thrust = C_t \Omega|\Omega|$$

Equation 21

To calculate the value for C_t a graph of the thrust obtained verses the speed of the propellor squared was drawn. The values of thrust were extrapolated from the motor current using Equation 4. The linear relationship shown by this graph shows agreement with the relationship presented by the model.

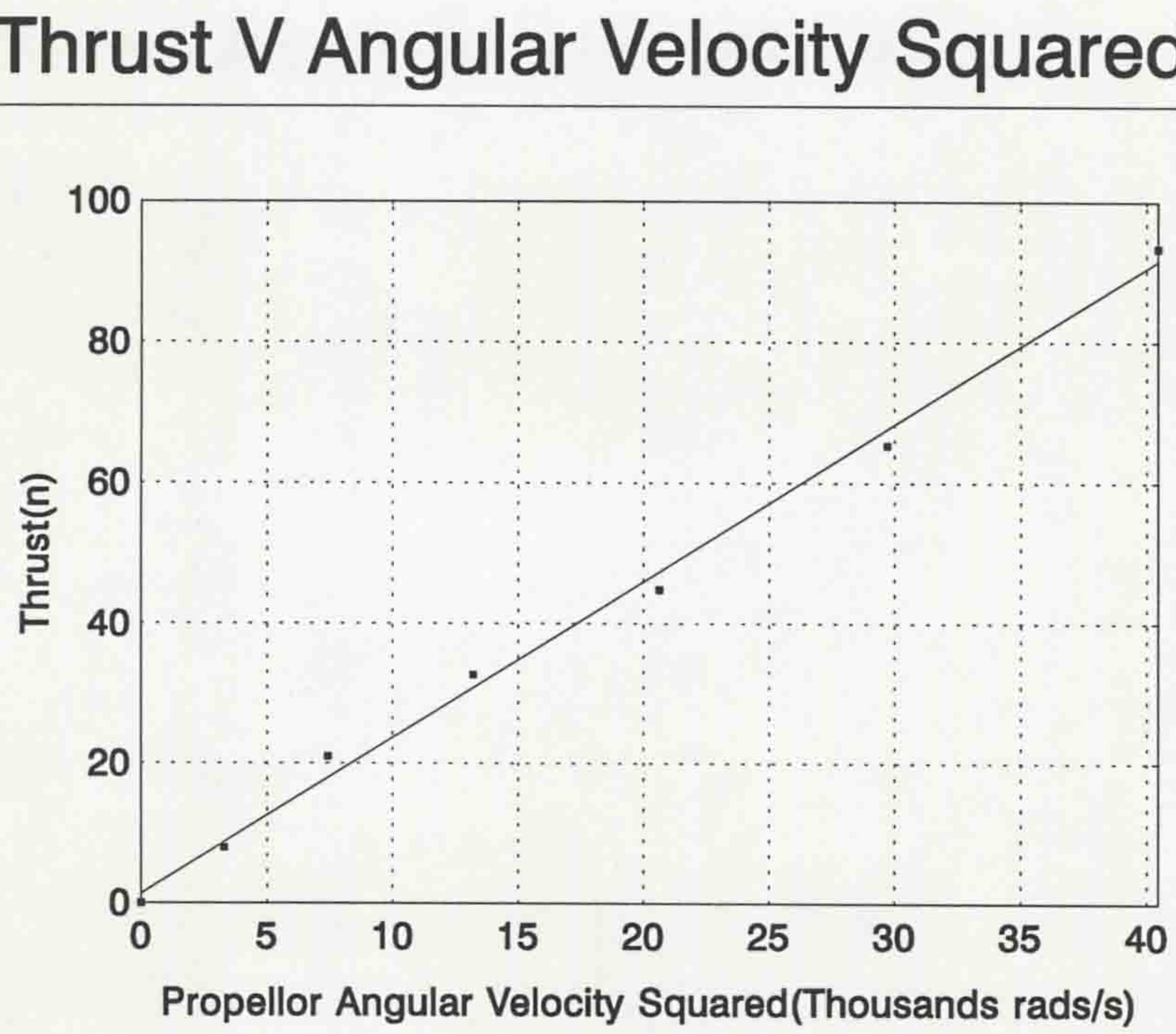

Figure 49, Thrust vs Propellor Velocity^2

From Figure 49 the value for C_t can be calculated as:-

$$C_t = 2.234e\text{-}3\ Ns^2$$

4.2.3.4.4 The Overall Thruster Model

By combining the models for stiction, the motor, and the propellor with the information taken during the wet and dry tests the overall model for the thruster can be formed.

From Equation 2;	K_m	=	0.087Nm/A
From Equation 3;	K_b	=	0.092V/rads^{-1}
From Equation 4;	Slip	=	20.0N/ms^{-1}
From Section 4.2.1.4;	R_a	=	0.240 Ω
From Figure 49;	C_t	=	2.234e-3Ns^2
From Equation 20;	β/α	=	21263
From Equation 12;	J	=	4.584e-3kgm^2

Figure 50 shows the finished model for the thruster incorporating the motor model, the stiction model and the propellor model. The parameters of Voltage and vehicle speed are input to the model, which gives the current used, the propeller's rotational speed, and the overall thrust achieved.

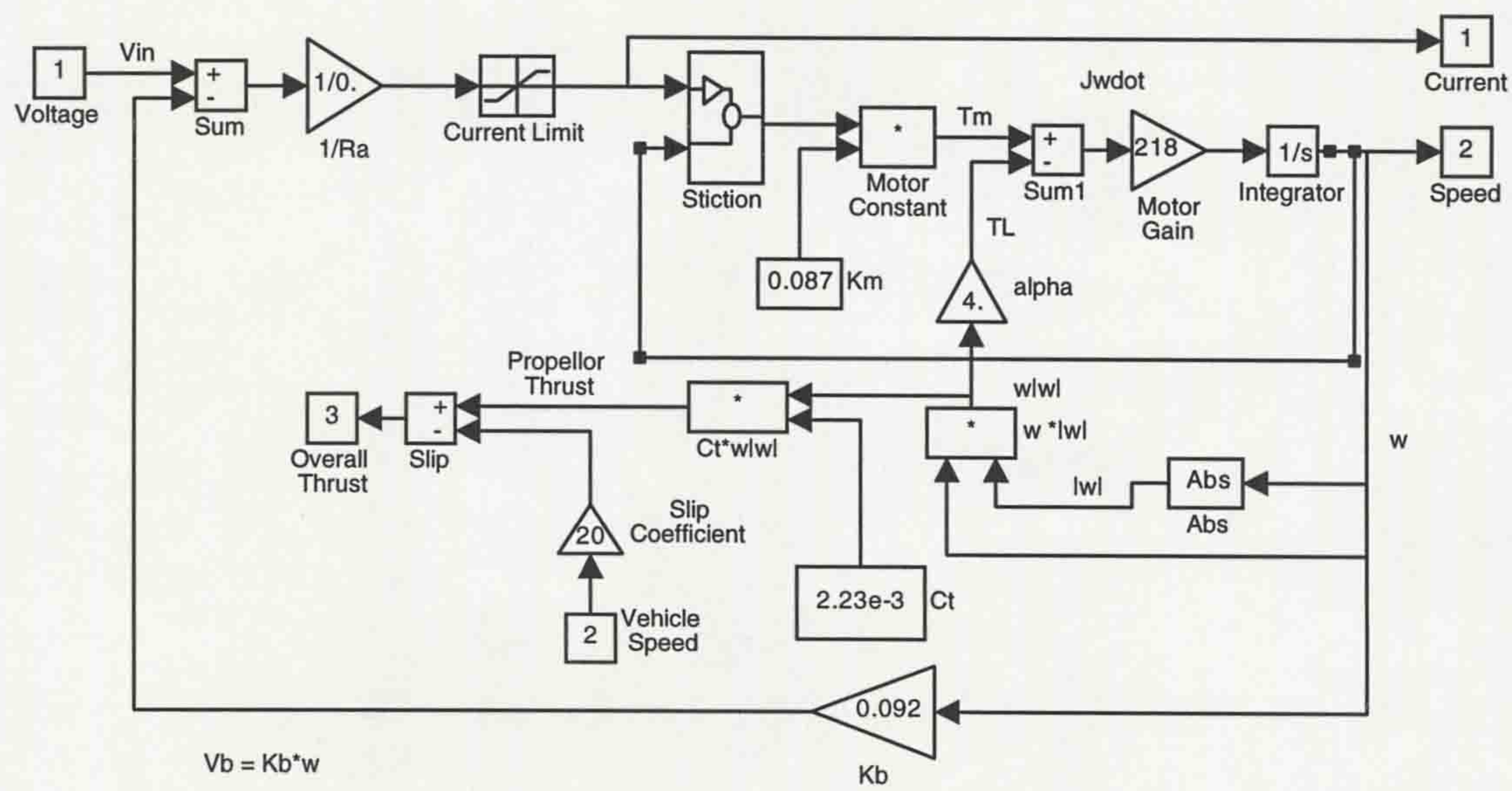

Figure 50, The Thruster Model

4.2.4 Model Verification

To verify the model the results of the model should be compared with the actual values measured from the physical system. As can be seen in Table 4 the steady state response of the model corresponds very well to the actual results taken for the thruster.

Input Voltage(v)	Actual Steady State Current(A)	Model Steady State Current(A)
5.0	**1.9**	**1.8**
7.5	**4.5**	**4.25**
10.0	**7.0**	**6.8**
12.5	**9.6**	**9.5**
15.0	**14.0**	**14.0**
17.5	**19.0**	**19.1**
20	**20.0**	**20**

Table 4

In addition to the steady state response of the system the dynamic response of the system is also important. The following four graphs show the dynamic response of the model compared to the same input to the test system.

Figure 51 shows the speed response of the actual system, whilst Figure 52 shows the speed step response of the model to the same input. As can be seen the model follows the actual system very closely, with the largest discrepancy being the difference in the decay times. This is probably due to that the tests for the physical system did not drive the motor to 0V, but turned off the power, allowing the motor to freewheel. Had the motor been hard driven to 0V a result more like that of the model would be expected. The X axis indicates time, whilst the Y axis indicates the motors speed, via the back emf voltage.

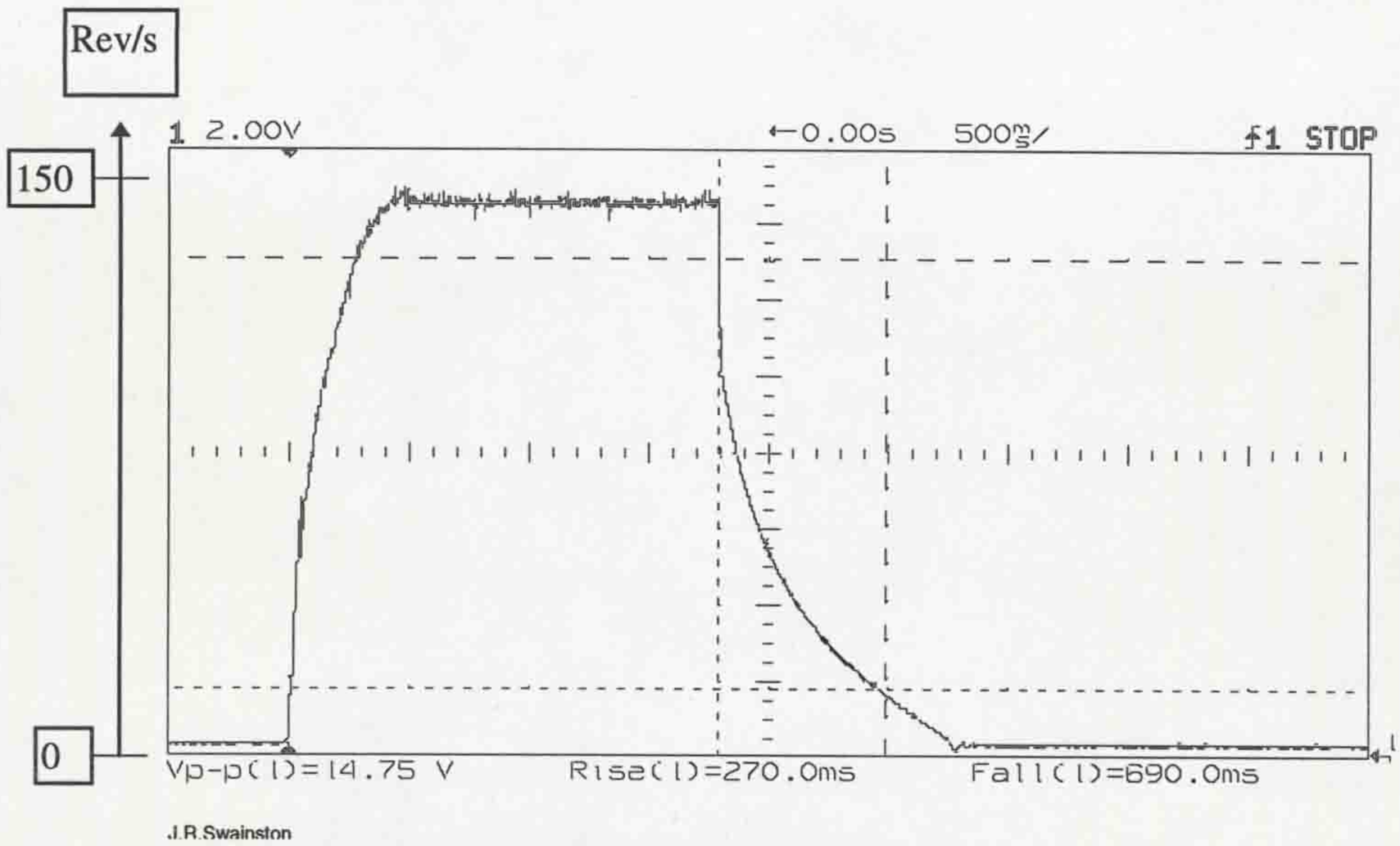

Figure 51, System Step Response for 15V Step Input

Figure 52, Model Step Response for 15V Step Input

Figure 53 and Figure 54 show the actual and simulated current response for the system. As can be seen the model generally follows the form of the actual data, but is more idealized in its form. The rise times and fall times for the model are much faster due to the physical constraints of the power supply in the real system.

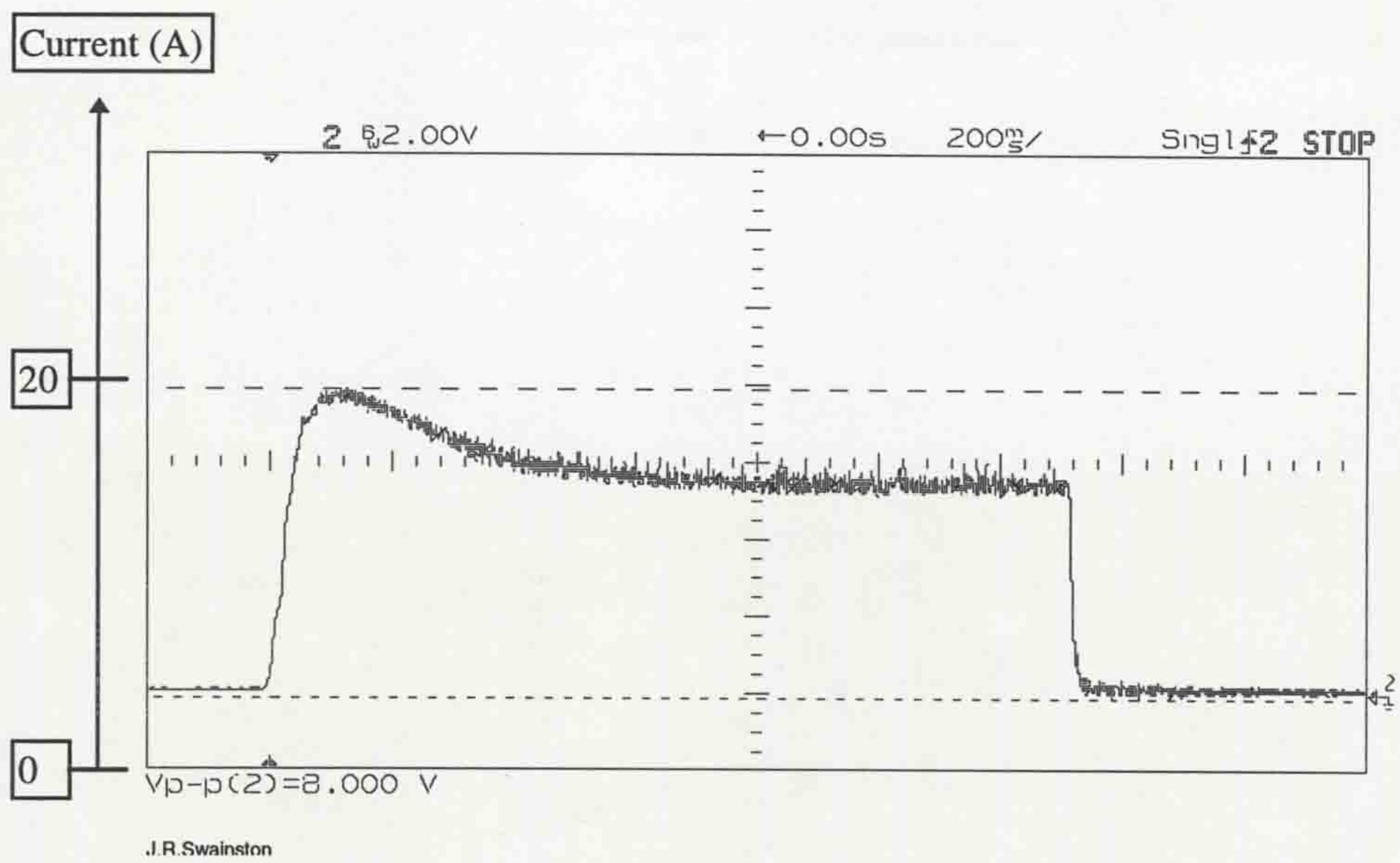

Figure 53, Systems Current Step Response to 15V input

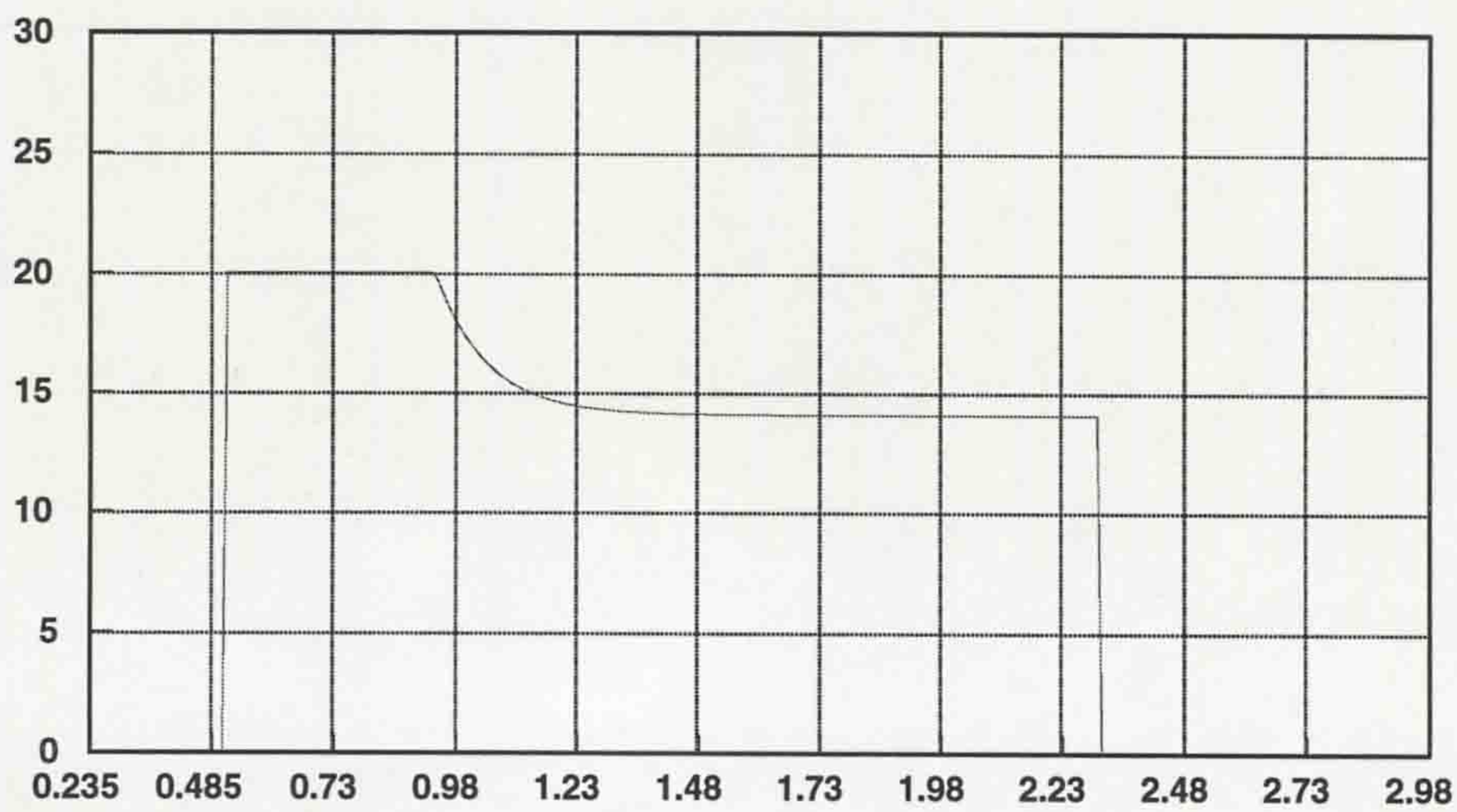

Figure 54, Models Current Step Response to 15V input

4.3 Motor Drive

To control the motor a bi-directional variable speed D.C. motor drive was required. The specification for the thruster's motor with a maximum input of 25volt 20amp is unusual, since most motors with a 500W rating opt for a higher input voltage and consequently lower current rating. Finding a commercial power drive system proved difficult because of the motor's unusual ratings; however one commercial drive was purchased and evaluated. The commercial power drive manufactured by Soprel Servodrives (Model VMD60-10/20) proved inadequate for the purpose. This was due to the large current spike on startup which caused damage to the drive, and due to the inadequate control system built into the drive controller. This control system was unable to cope with the high speed, low inertia motor being used, being designed for motors with a longer time constant. This resulted in the system being closed loop unstable, causing oscillations which damaged the power stages of the drive. Attempts to modify this control loop proved unsuccessful, and an alternative had to be found. It was decided that a dedicated motor drive should be built to power the thrusters.

From the tests carried out on the thruster it was seen that the motors needed a high initial startup current. To help overcome this problem a P.W.M. (Pulse Width Modulation) motor drive was specified, as the high current peaks of the pulses are capable of overcoming some of the stictional problems. The P.W.M drive can also be made more compact than other linear amplifier systems.

4.3.1 P.W.M. Motor Drive

The P.W.M. motor drive designed was based upon an original design by Bennett[4.7]. The drive can be separated into several basic areas:-

- The P.W.M. waveform generation circuitry
- The P.W.M. signal amplifier
- The gate drive circuitry
- The power drive circuit
- The active snubber circuit
- Signal smoothing, filtering circuit

A block diagram of the motor drives showing the major parts of the design is shown in Figure 55.

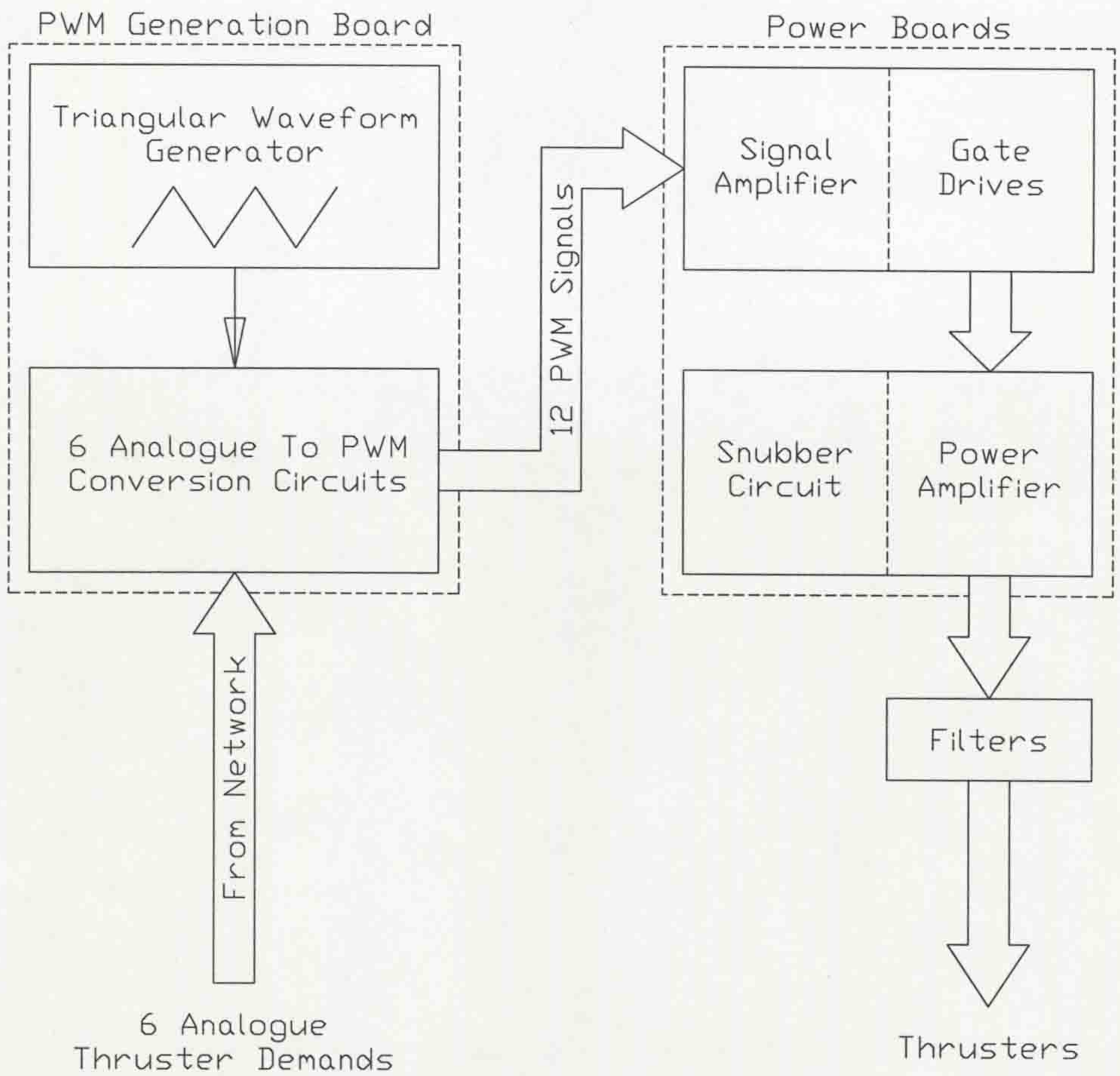

Figure 55, A Block Diagram of the P.W.M. Drive.

4.3.1.1 Waveform Generation Circuit

The basis behind the generation of a pulse width modulation signal is the generation of a reference waveform. The reference waveform can be compared with the input signal to create the basic P.W.M. signal. By altering the frequency and amplitude of the reference signal the corresponding changes can be made to the operation of the motor drive.

4.3.1.1.1 Triangular Waveform Generation

The waveform generation circuit used for the power drives in the Aquasphere was based upon that built for the Aquacube. There are many types of waveform generation circuits detailed for such a common waveform, for example by Horowitz and Hill[4.8] but this version was chosen due to its simplicity and low component count. Figure 56, shows the schematic for the triangular waveform generator used.

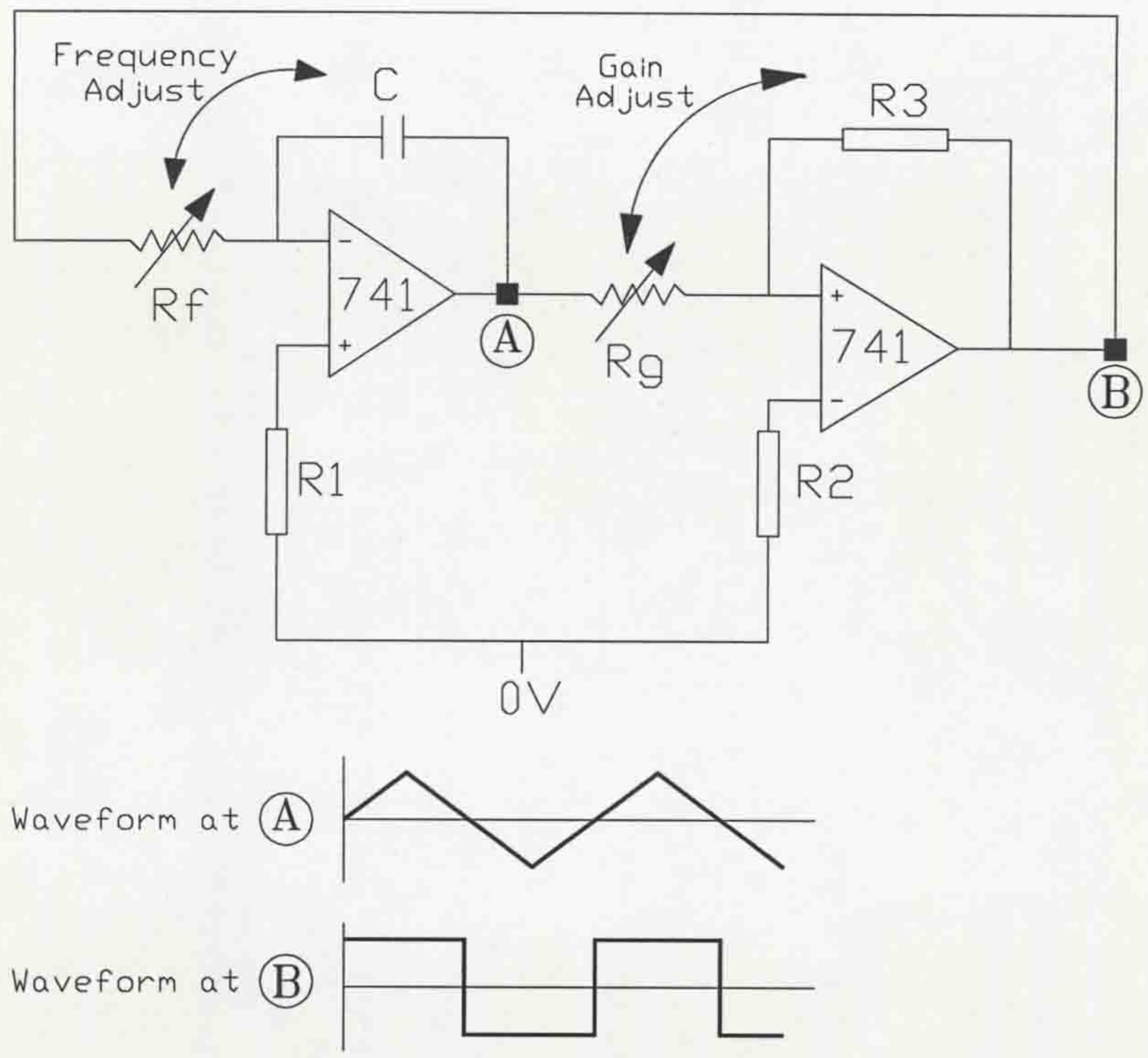

Figure 56, Triangular Waveform Generator

The waveform generator comprised an integrator and a comparator stage. Upon switch on the comparator goes high or low, which is integrated generating a slope. When the comparator threshold is crossed the comparator toggles, and the slope of the integrator reverses. By altering the parameters of the integrator the frequency of the generator can be adjusted, and by altering the threshold of the comparator the amplitude of the triangular waveform can be set. The triangular waveform was initially set to an amplitude of 10V with a frequency of 20kHz.

By creating a single P.W.M. signal generation board only one such oscillator was needed, with its output sent to the input stages of all six P.W.M. generation circuits.

4.3.1.1.2 P.W.M. Generation Circuit

To generate the pulse width modulated signal from an analogue input and a triangular waveform is relatively simple. The analogue input needs only to be compared with the reference waveform and the output of the comparator is a pulse width modulated signal. The width of the pulse is directly proportional to the magnitude of the analogue input. Figure 57, shows the circuit necessary to produce the pulse width modulated signal.

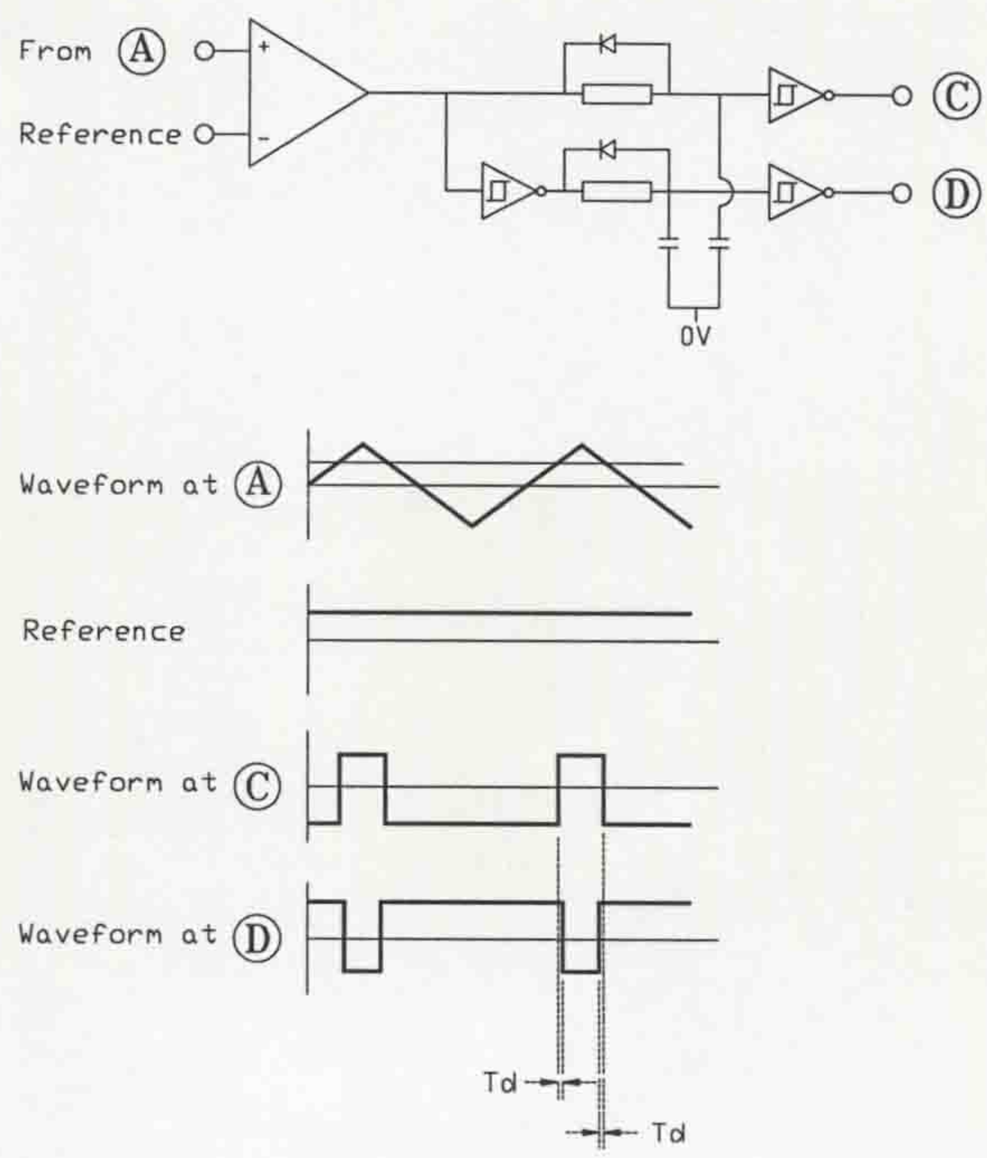

Figure 57, The P.W.M. Generation Circuit.

As can be seen from Figure 57, the comparison of the two signals gives the pulse width modulation signal required for a single gate drive. However, the configuration of the power system used was a full H-bridge drive which requires two opposing signals to switch on and off the opposing sides of the bridge. To simply invert the signal which turns on and off the power devices can cause problems as race conditions exist which would allow devices on the same side of the bridge to be turned on simultaneously, thus shorting the power supply. To avoid this a delay circuit was added to the output of the P.W.M. generator, to ensure that the devices switched at separate times, allowing one set of the devices to be fully switched off before the other devices were activated. The delay circuit was created by forming an RC time constant which is inputted into a Schmitt trigger to give a delay. The

switching speed of the devices used was 2μs, so the time delay T_d was set to 4μs to give the power devices time to switch and settle.

Figure 58 shows the P.W.M. generation circuit constructed on veroboard. The six analogue demand signals, and the board's power are inputted from the 15-way D-type connector on the right of the picture, are low-pass filtered to remove signal noise, and inputted into the six LM311 comparators in conjunction with the triangular waveform. After the delaying and inverting stage the signals are outputted in six pairs, one to each of the power drive modules.

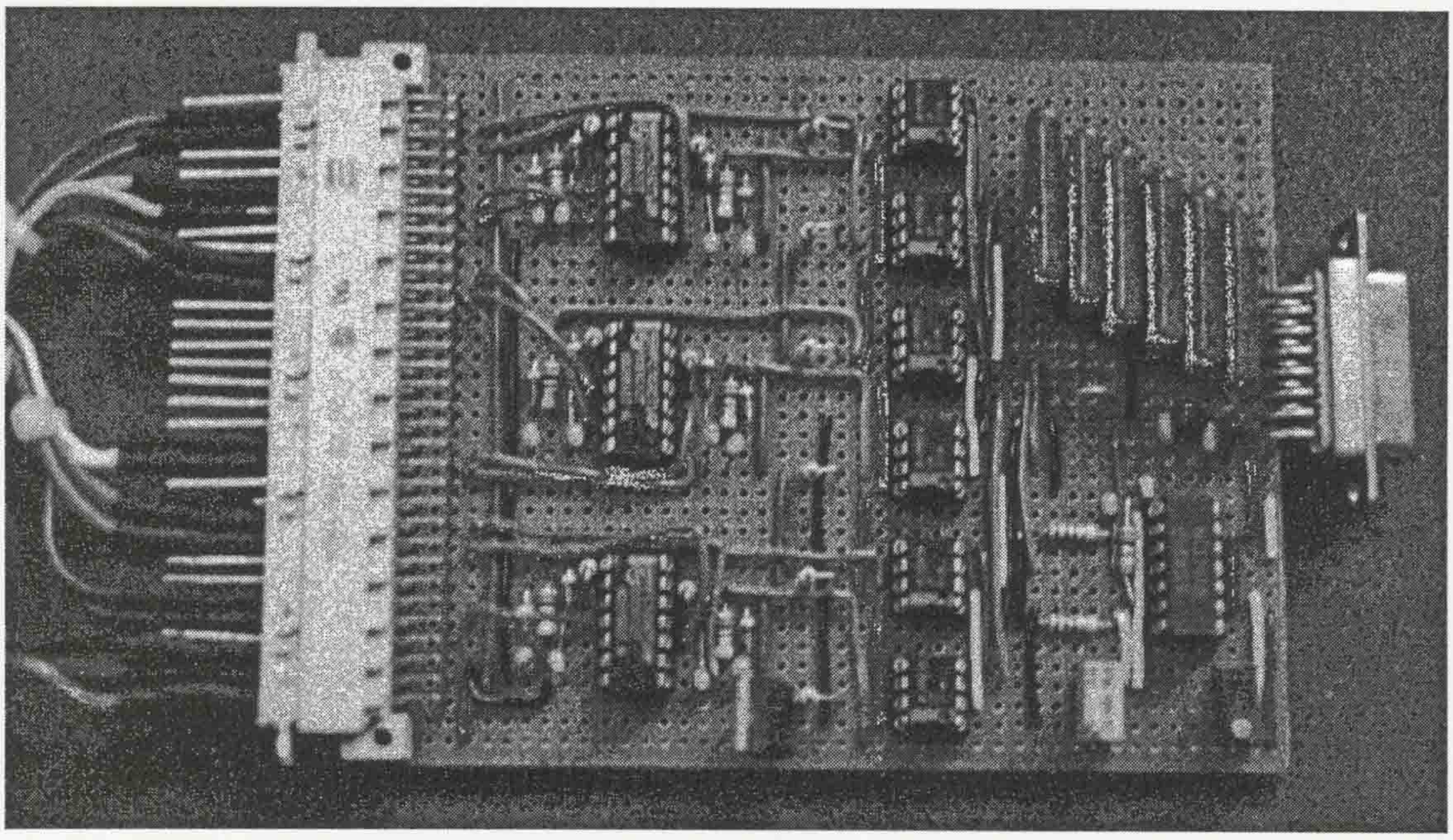

Figure 58, The P.W.M. Generation Board

4.3.1.1.3 Gate Drive Circuit

As the source of the mosfet devices is at a floating voltage then a floating drive circuit for the gate/source of the power drives was needed. To drive the gate the circuit shown in Figure 59 was used. This Gate Drive system was originally used in the prototype drive for the Aquacube.

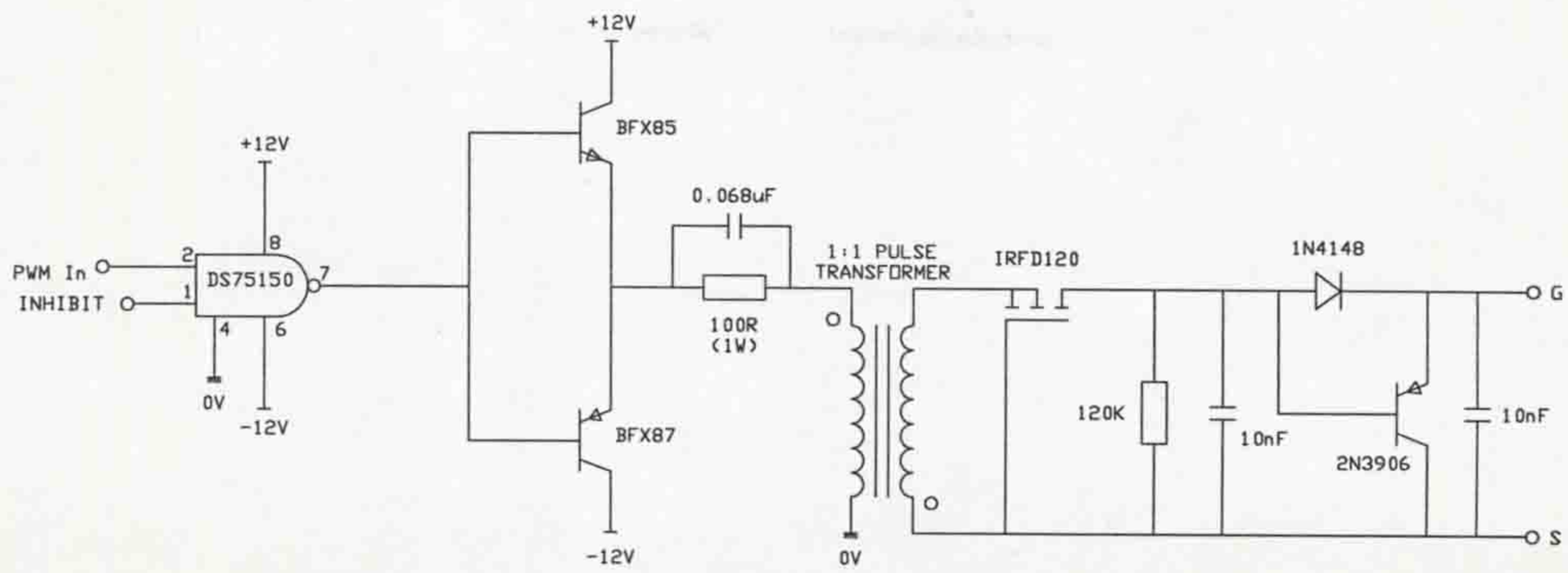

Figure 59, The Gate Drive Circuit

The gate drive circuit is split into two main parts. On the left hand side of the circuit is an amplifier stage which increases the power of the input signal. The signal is connected to a DS75150 line driver which drives the gates of two bipolar transistors. This causes a current pulse as the input voltage changes state, sending a signal through the pulse transformer. The pulse transformer gives the isolation necessary between the input and the floating voltage on the gate/source junction. The pulse on the secondary side of the transformer causes the gate of the mosfet to be switched on, or turned off. The charge to switch the gate on is held capacitively after the turn on pulse. This charge is held until the opposite pulse which causes the charge on the gate to be dissipated, turning off the device.

Four of these gate drive modules are needed per drive unit to activate each of the four quadrants of the H-bridge.

4.3.1.1.4 Design of the H-Bridge Power Drive

The H-bridge provides the final stage of amplification for the power drive circuit. The H-bridge comprises of four identical quadrants which operate in diagonally linked pairs. The H-bridge allows the thruster do be driven bi-directionally from a unipolar power supply.

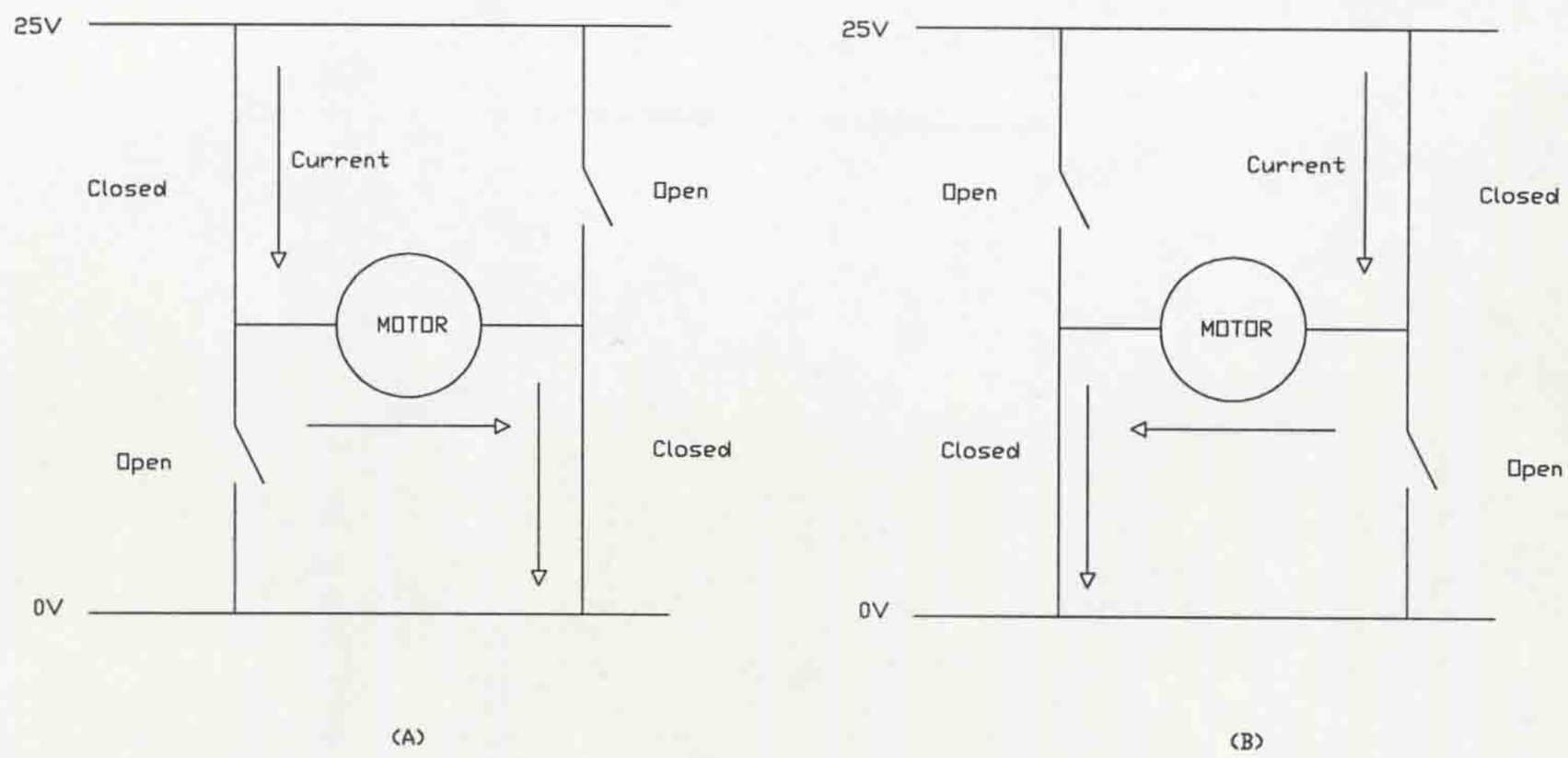

Figure 60, Schematic Diagram of an H-bridge Drive

Figure 60 shows the basis behind the operation of the H-bridge drive. The drive is split into four quadrants with the quadrants linked in pairs diagonally. When one pair is activated (as in (A)) the current flows one direction through the motor, and when the other pair of devices are activated (as in (B)) the current flows the opposite direction through the motor. In this way bi-directional motion of the motor can be achieved from a unipolar supply. The Duty Cycle, determines the speed of the motor. It is essential that the devices are not all switched on at once, which would cause the power supply to short.

The power devices used in the H-bridge were mosfet devices, type BUK456-200B. Mosfet devices were chosen in preference to bipolar transistors as they are more suitable for operation at a high frequency. Al-Taee [4.9] gives the following advantages for using a mosfet device instead of a bipolar-junction transistor.

1. It has considerably faster switching speeds and therefore has lower switching losses.
2. It is a voltage driven device with extremely high input impedance, hence it requires simpler gate drive circuitry and much lower driving power.
3. It has an integral body-drain diode, which can be used in voltage source inverters as a flywheel diode instead of using an external device.
4. It is more suitable for parallel operation because of the positive temperature coefficient which forces the current to be uniformly distributed between the paralleled devices.

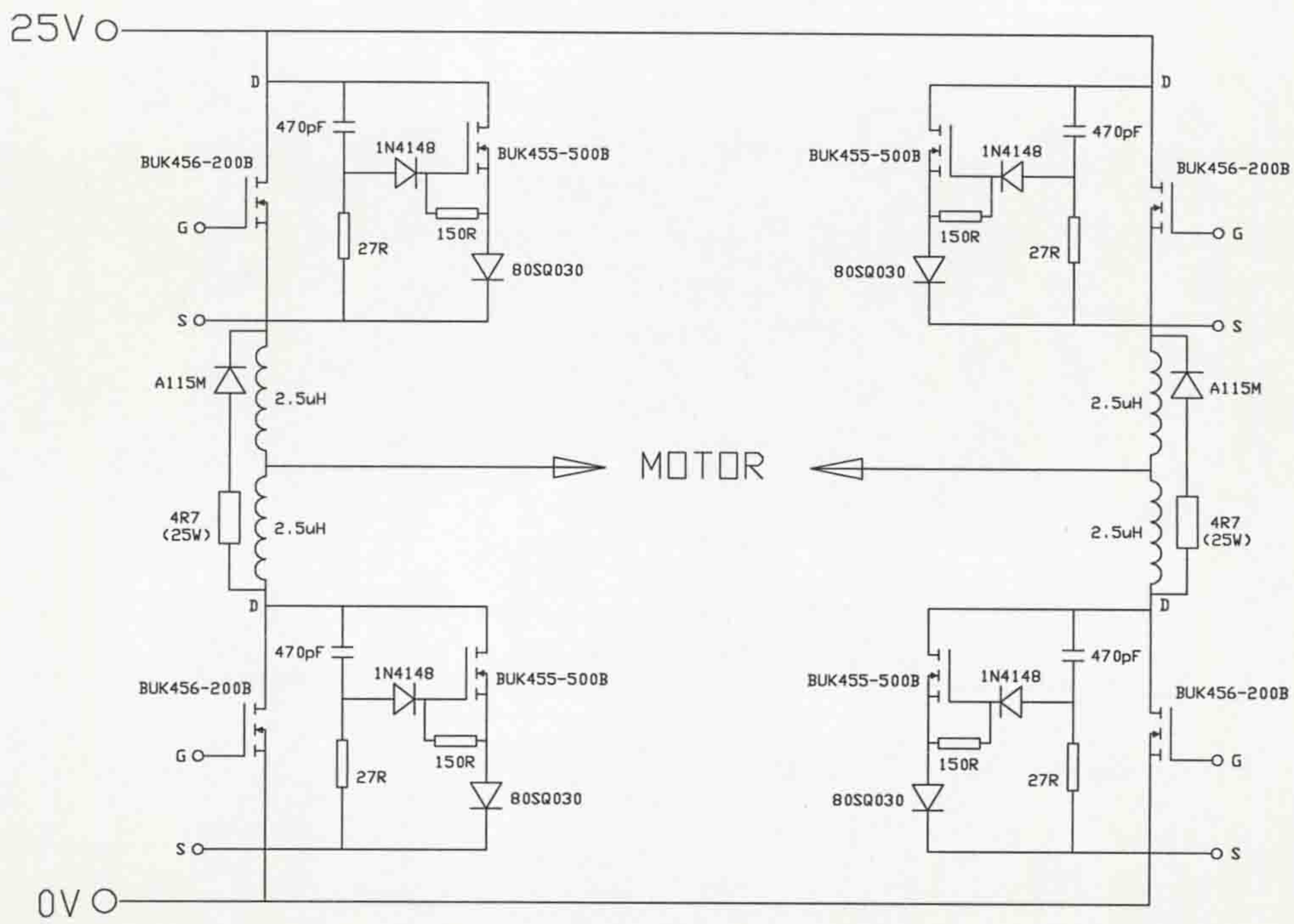

Figure 61, The H-Bridge Power Circuit

Figure 61 shows the circuit used for the thruster's power drive. The schematic above shows the four power units required, and the gate and source connections to which the four gate drive circuits are connected. Between the stages a small inductance was added (5μH) to limit the rate of rise of current at turn-on, limiting the reverse recovery current (I_{rr}) which could potentially damage the power device. To dissipate the stored energy a power resistor was added to this central portion of the drive. A turn-off snubber circuit was added to each of the power quadrants to limit the over voltage peak across the mosfet.

4.3.1.1.5 Constructing The H-Bridge Power Drive

From the design detailed in section 4.3.1.1.4 a power amplifier had to be constructed. In designing the power board other design criteria had to be met, as well as the schematic of the board. Firstly the power devices would have to mounted in such a way that they could be attached to a heat sink, and secondly the track capacitance of the gate drive circuitry would have to be kept as low as possible for h.f. performance.

To achieve this the power devices were mounted in a row near the edge of the board, allowing the gate drive devices to come close to the power devices, minimizing track length, and therefore gate drive track capacitance. An aluminium bus-bar was attached to the power devices to remove the heat from the devices. The power rails could then link the devices from behind the MOSFETS, under the bus-bar. The board layout designed using the computer package Cadstar can be seen in appendix 10.4.

Figure 62 shows a picture of the fully constructed power board. The power MOSFETS can be seen mounted against the aluminium bus bar, on which the power resistors are also mounted.

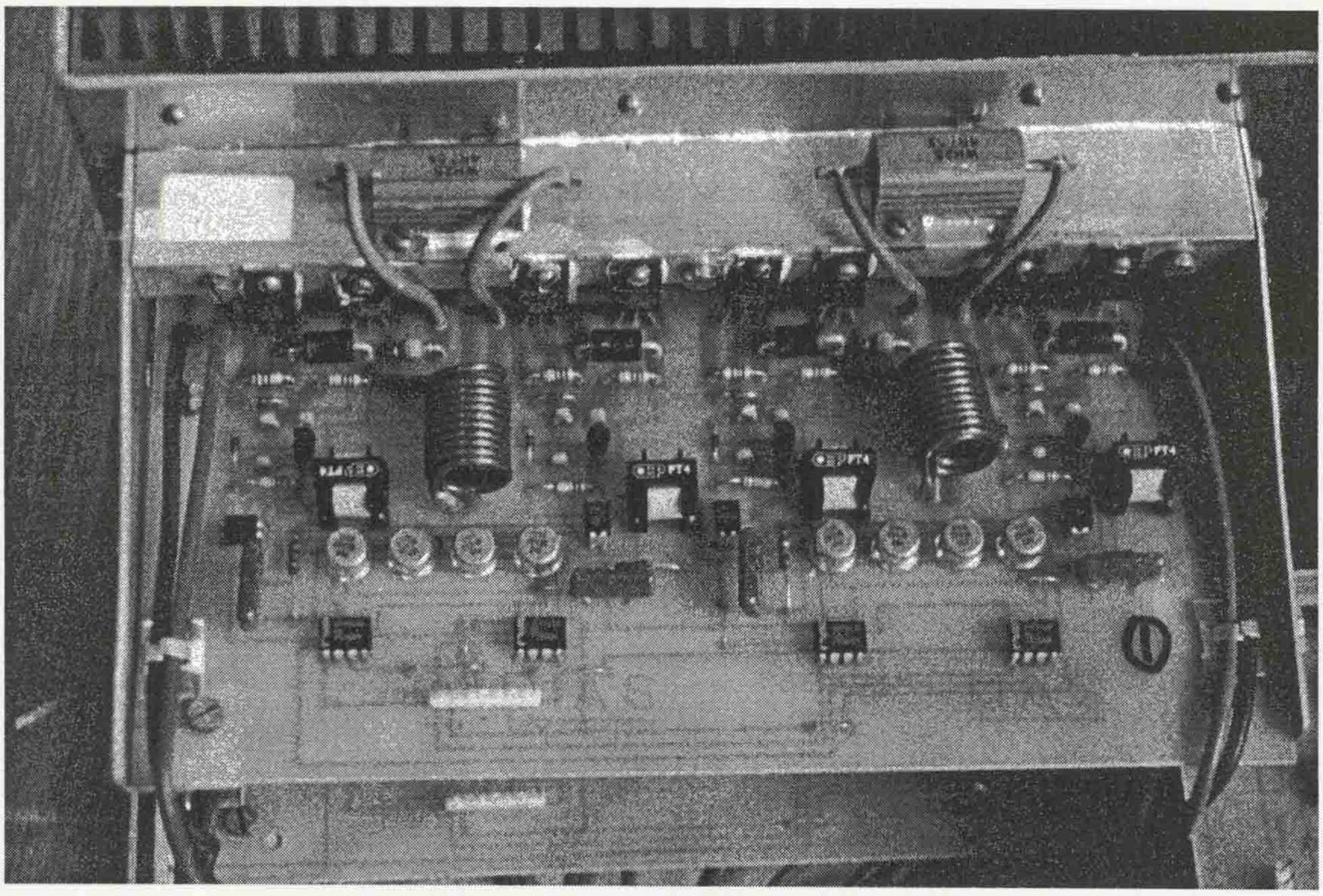

Figure 62, The P.W.M. Power Drive

The six motor drive units needed to be fitted into a box which would then be mounted into the marinised thruster box fixed to the vehicle. An aluminium box was constructed which would fit into a standard 19" rack mount. The outsides of this box had cooling fins attached to drain heat from the aluminium bus bar. The power drive units were mounted in pairs around three of the sides of the box, whilst the P.W.M. generation card was mounted vertically in the space in the middle of the box.

In order to reduce noise from the power drive unit to other systems on the vehicle such as the vision system, large capacitors were added across the D.C. power supply at the vehicle end of the system, to reduce ripple on the power supply caused by the loading of the motors. One effect of having one single clock for the generation of the P.W.M. signals is that all the power drives activate simultaneously causing a large instantaneous load on the power supply. The capacitors help reduce the ripple in the power supply, which can potentially cause noise in other systems.

In order to further reduce noise low pass filters were added to the output supplies to the thrusters to reduce harmonics generated on the motor drive lines. Ferrite cores were also added to the control input signals, and the signal power supply lines to reduce the possibility of interference through these lines. In tests on the vehicle the power drives were found to cause no measurable interference on the rest of the vehicle.

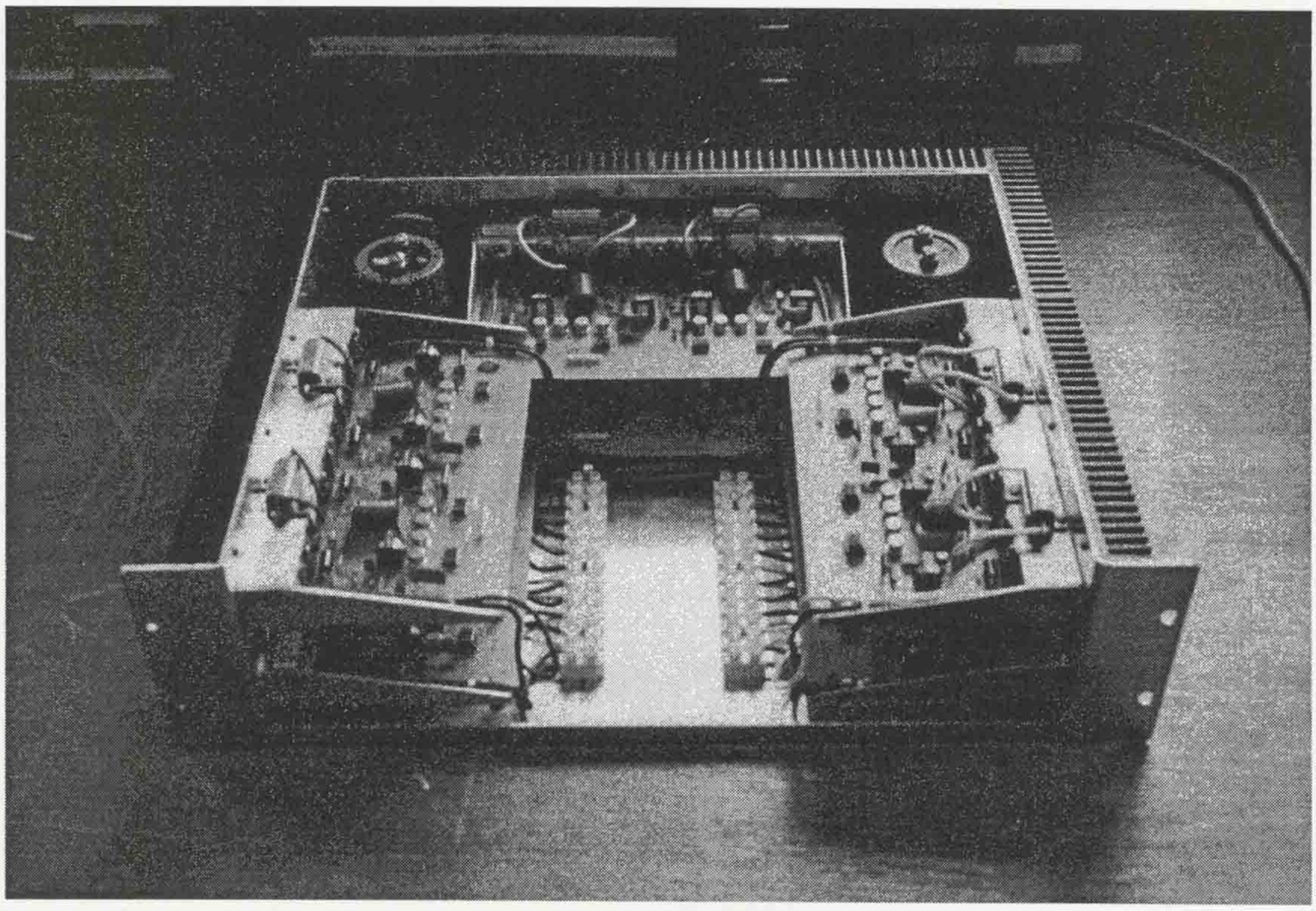

Figure 63, The Thruster Power Box

4.4 Advanced Motor Drive

After the final trials of the vehicle a necessity was seen for the reduction in size and weight of the vehicle's systems. Another look was taken at the design of the power drive systems with the aim to reduce the size and complexity of the motor drives, whilst increasing efficiency. With the advances made in semiconductors during the time of the project, and the greater diversity of devices available it was seen that another smaller more simple motor drive could be constructed. Also a closed loop controller for the thruster was required, so an optical disc was fixed to the motor's shaft, with two optical transmitter/receiver units to give the motor's speed and direction.

4.4.1 Microcontroller Systems

To reduce the load on the networking system, and to aid the transference of the technology on the Aquasphere onto an autonomous underwater vehicle it was decided that the thrusters should contain their own processing and drive units. The investigation of current devices highlighted the new microcontroller devices presently available.

Microcontrollers are true computers on a chip. Their design incorporates all the features found in a microprocessor CPU, in addition to the other features needed to make a computer such as ROM, RAM, parallel I/O, serial I/O, counters and a clock circuit. The microcontroller is capable of fetching data, performing limited calculations on the data, and controlling the outputs based on this data. The small amount of ROM and RAM on microcontrollers means that the use of microcontrollers is limited to control the operation of a machine using a fixed program that is stored in ROM, and does not alter over the lifetime of the system. The instruction set for most microcontrollers is less than that for most microprocessors, but is highly optimised for input/output to the ports on the microcontroller. Although not as powerful as a microprocessor system microcontrollers are very compact, need little or no external hardware for their operation, and have a low power consumption. These features make them ideal for use in relatively simple repetitive systems.

4.4.1.1 The INTEL 87C51FA Microcontroller

From the several hundred microcontrollers currently available a suitable version had to be chosen for the task of controlling the thruster system. From the technical data available a choice had to made as to which microcontroller should be used. The specification of the

microcontroller used, the INTEL 87C51FA, taken from the microcontroller's manual[4.10] is shown below in Table 4.

The 8XC51FX is a highly integrated 8-bit microcontroller based on the MCS-51 architecture. As a member of the MCS-51 family, the 8XC51FX is optimized for control applications. Its key feature is the programmable counter array (PCA) which is capable of measuring and generating pulse information on five I/O pins. Also included are an enhanced serial port for multi-processor communications, an up/down timer/counter, and a program lock scheme for the on-chip program memory. Since the 8XC51FX products are CHMOS, they have two software selectable reduced power modes: Idle Mode and Power Down Mode.

All MCS-51 devices have a separate address space for Program and Data Memory. Up to 64 Kbytes each of external Program and Data Memory can be addressed.

The 8XC51FX uses the standard 8051 instruction set and is pin-for-pin compatible with the existing MCS-51 family of products.

- Full-Duplex Programmable Serial Port with
  - Framing Error Detection
  - Automatic Address Recognition
- Three 16-Bit Timer/Counters with
  - One Up/Down Timer/Counter
  - Clock Out
- Power-Saving Modes
  - Idle Mode
  - Power Down Mode
- Interrupt Structure with
  - Seven Interrupt Sources
  - Four Priority Levels
- Programmable Counter Array with
  - Compare/Capture
  - Software Timer
  - High Speed Output
  - Pulse Width Modulator
  - Watchdog Timer
- Four 8-Bit Bi-directional Parallel Ports

Table 5

The main features of this device which make it suitable are the serial I/O, the built in counter timers, and the hardware P.W.M. signal generation. This enables the microcontroller to control the power to the thruster by adjusting the duty-cycle of the inbuilt P.W.M signal, whilst measuring the speed of the thruster using the counter-timers, all being controlled remotely via a serial port interface.

The internal block diagram of the 87C51FA microcontroller is shown in Figure 64, showing the PCA modules which form the serial interface, and the hardware counter timers.

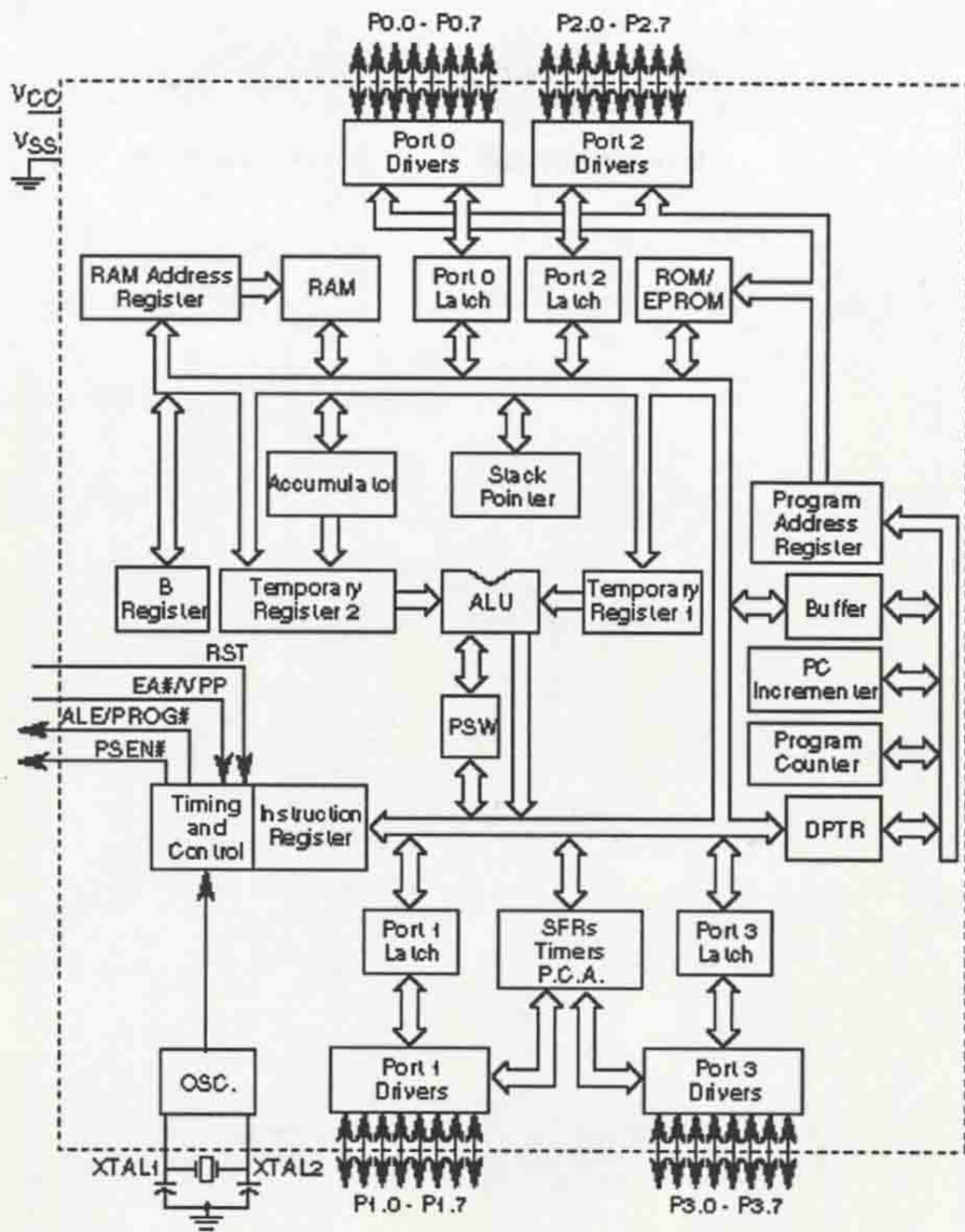

Figure 64, Intel 87C5FX Block Diagram

4.4.1.2 Microcontroller System Design

To build the microcontroller system a development kit supplied by Phillips[4.11] was purchased to design and test the software needed for the microcontroller system. This system provided the programming tools and cross assemblers required to program the MCS-51 family of microcontrollers, including the 87C51FA variant.

Using the development system routines were tested to generate P.W.M. signals, and to test the digital input and counter timer modules for the microcontroller. However, as the development system uses the microcontroller's serial I/O link a separate module had to be designed to test out the serial communication of the device. This separate module consisted of the microcontroller device, a 11.059MHz oscillator, and a MAX233 serial converter chip to give RS232-TTL logic conversion. Port outputs were linked to mosfet line driver chips, and a digital channel input was available via a Molex connection. A photograph of the test board is shown in Figure 65.

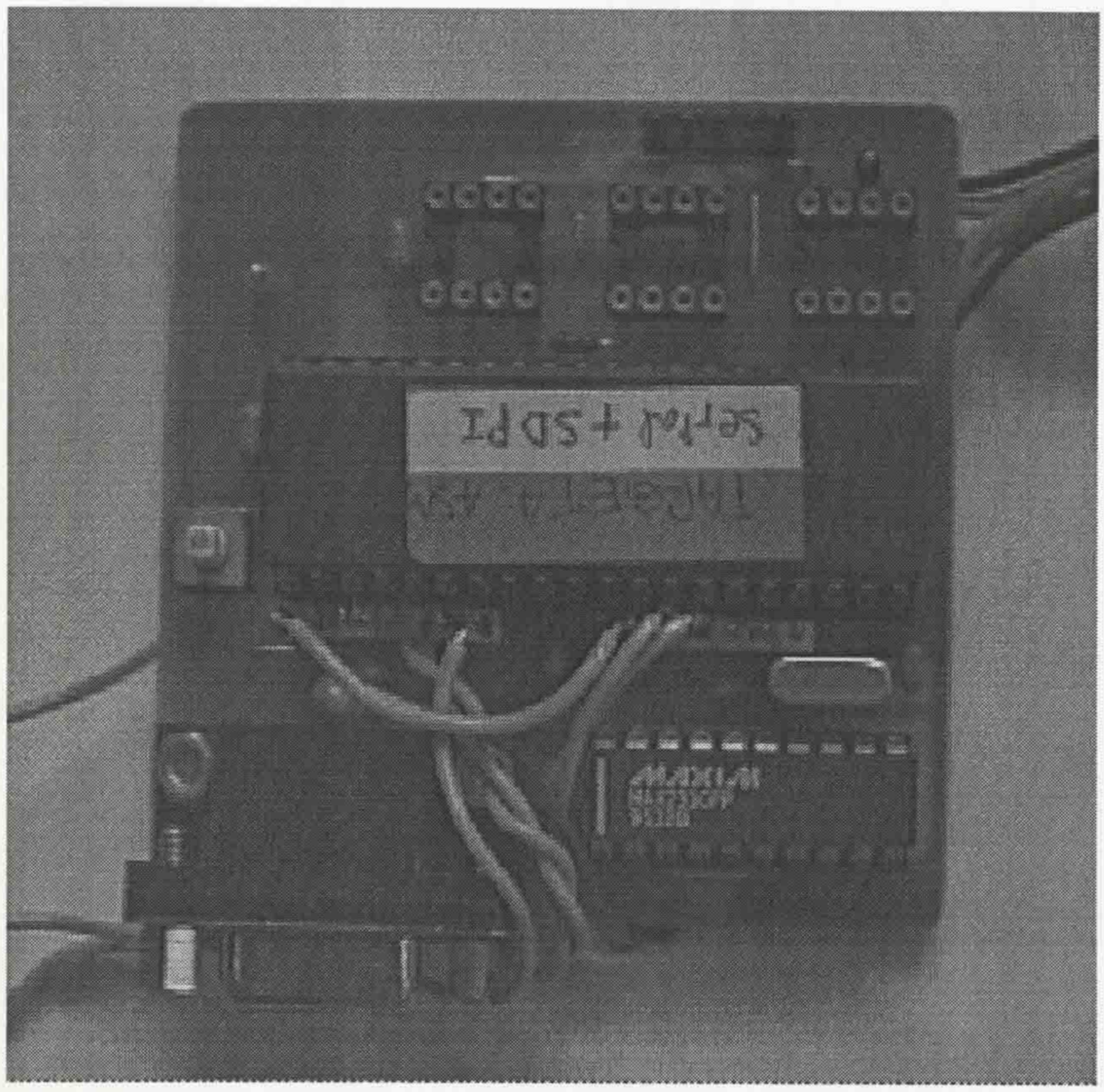

Figure 65, The Microcontroller Test Board

Due to the flexibility of the microcontroller the ports on the microcontroller shown in Figure 64 have several functions each. To be able to use the microcontroller for any purpose requires the program status registers to be set to put the microcontroller into the desired state. To perform the task required it was necessary to program several of the ports. The ports required are a serial port, a PWM port and a digital input port.

4.4.1.2.1 Serial Interfacing

To set the microprocessor for serial port operation the SCON register (Serial CONtrol) register must be set to give the required mode of operation. The microprocessor chosen can operate in four different modes of operation

Mode 0: Serial data enters and exits through RXD. TXD outputs the shift clock. 8 bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 the oscillator frequency.

Mode 1: 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1).

Mode 3: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit and a stop bit (1). Mode 3 is the same as Mode 2 in all respects except the baud rate. The baud rate in Mode 3 is variable.

For this project mode 1 was selected, as this method conforms to the data standard transmitted by RS232 interfaces common on personal computers. The serial port information for the microprocessor can be seen in Appendix 10.5.1.

When data is received by the UART the Receive Interrupt Flag is set. For handling serial data a procedure must be tied to this interrupt to deal with the incoming data. Once the data has been read in the flag is reset in software to allow more data to be received. For test purposes a simple routine was written to receive two bytes, and to output this information to the P.W.M generation routine.

To transmit data the Transmit interrupt flag must be checked, and if it is set the value to be sent can be entered into the output register. Once the data is sent the flag must be reset by software.

To set the baud rate for the device one of the microcontroller's counter/timer modules must be used. The baud rate for the microcontroller can be set using the formula below when the timer is in auto reload mode:-

$$BaudRate = \frac{2^{SMOD1} * OscillatorFrequency}{32 * 12 * [256 - (TH1)]}$$

Equation 22

In summary to program the serial port on the microcontroller the user must:-

1. Program the SCON register to set the appropriate mode
2. Set the mode of the TIMER 1 to auto reload mode
3. Set the value in Timer 1 to give the desired baud rate.
4. Tie a serial data handling routine to the Receive interrupt flag.

4.4.1.2.2 P.W.M. Generation

To generate a pulse width modulated signal one of the microcontroller's PCA modules must be set to operate as a P.W.M. output. To do this the CCAPMn registers must be set as shown in Appendix 10.5.2

Any or all of the five PCA modules can be programmed to be a Pulse Width Modulator. The frequency of the PWM depends on the clock sources for the PCA timer. The PCA generates 8-bit PWMs by comparing the low byte of the PCA timer (CL) with the low byte of the module's compare registers (CCAPnL). When CL < CCAPnL the output is low. When CL > CCAPnL the output is high. The value in CCAPnL therefore controls the duty cycle of the waveform. To change the value in CCAPnL without output glitches, the user must write to the high byte register (CCAPnH). This value is then shifted by hardware into CCAPnL when CL rolls over from 0FFH to 00H which corresponds to the next period of the output. Figure 66 shows the operation of the PCA module in PWM generation mode.

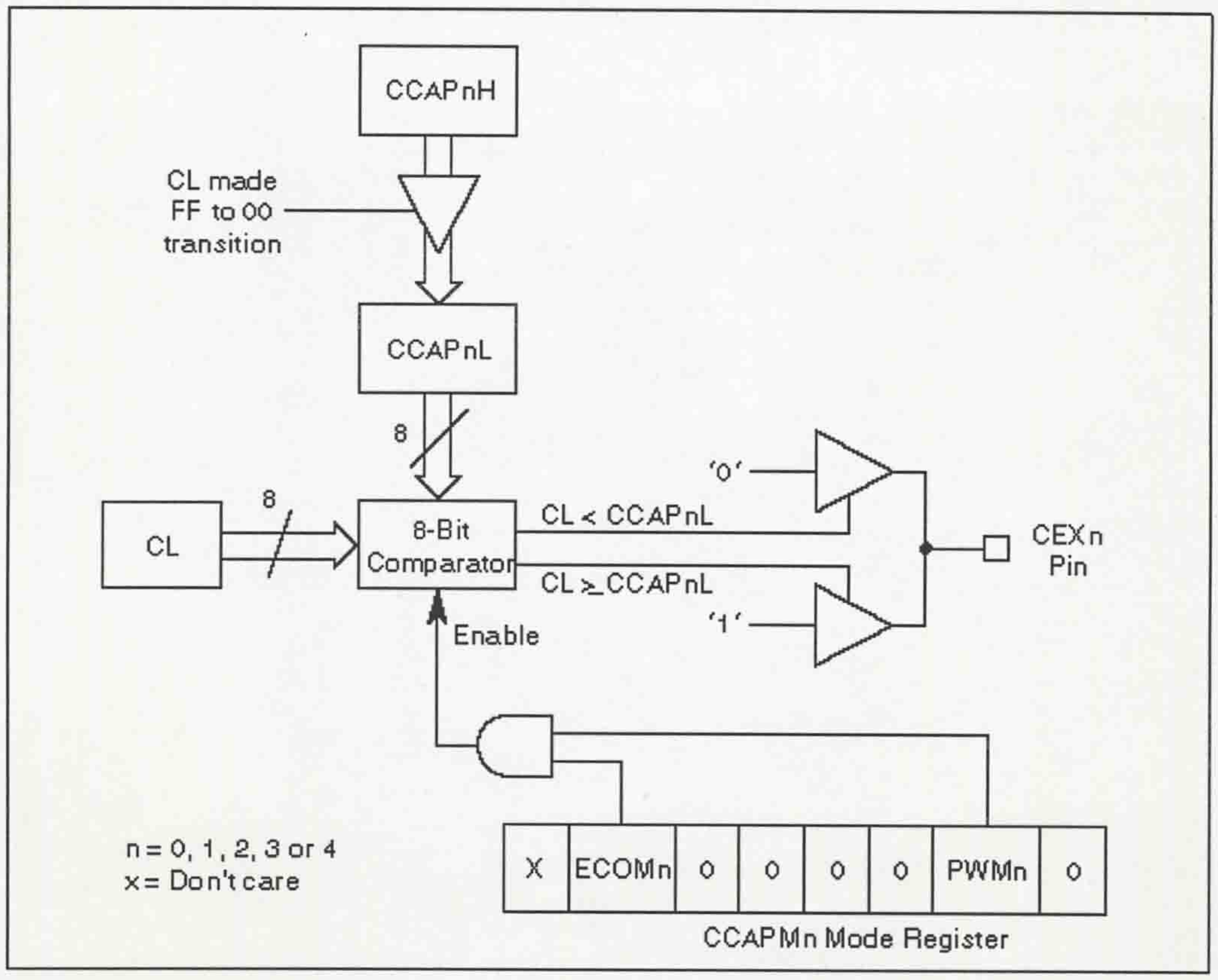

Figure 66, PCA 8-bit PWM Mode

CCAPnH can contain any integer from 0 to 255 to vary the duty cycle from a 100% to 0.4%, as seen in Figure 67.

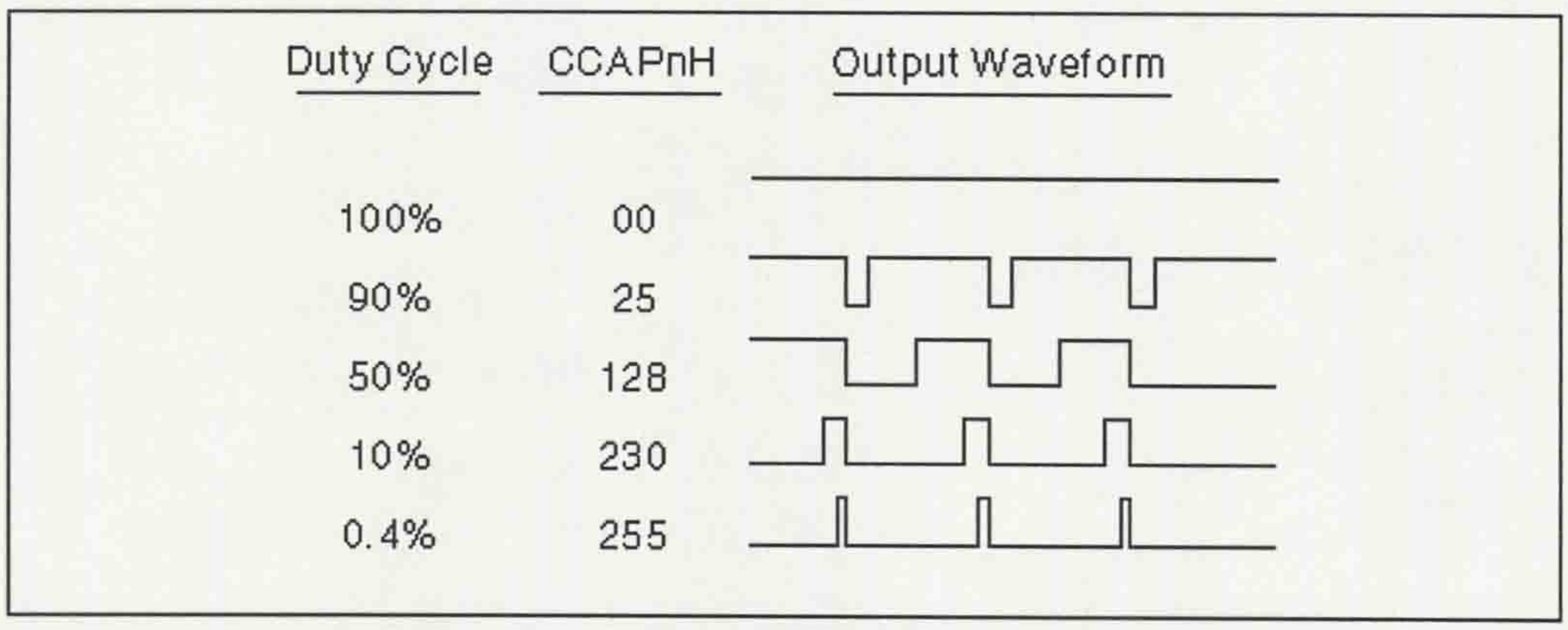

Figure 67, Duty Cycle with respect to CCAPnH

Therefore to generate a PWM signal from the microcontroller the user must:-

1. Set the PCA control register CCAPMn to x1000010b to activate the PWM mode.
2. Set the frequency of the PWM signal by setting the speed CL of the timer register.
3. Move the duty cycle value into CCAPnH.

4.4.1.2.3 Counter/Timer Input

To read in the speed from the optical encoder attached to the motor the digital input from the shaft encoder had to be read in. To do this one of the input ports needs to be configured as a counter/timer module. To do this the CMOD register needs to be configured to set the PCA port to be a counter timer. The information on programming the CMOD register is included in Appendix 10.5.3.

To program the counter timer the following steps must be carried out:-

1. Set the PCA to counter mode by setting CMOD.
2. Set the frequency of the oscillator input such that at the maximum rotational speed of the motor the counter module gives full scale deflection.

4.4.2 Thruster Gate Drive

Since the design of the original power module many integrated circuit semiconductors have been produced to aid the construction of P.W.M. drives. Several full-bridge P.W.M. on a chip modules have been produced, but as yet the output current for these devices is limited to about 2amp. For the required current of 20amp it is still necessary to use a power device in conjunction with a gate drive circuit.

In addition to the complete units incorporating P.W.M. generation several devices have been released which include MOSFET/IGBT base drive, floating bootstrap power supply, and control logic to drive full bridge circuits.

One such device presently available is the EL7661CN Bridge Driver shown in Figure 68. This device is just one of several devices available at the moment, and shows the typical form of the devices.

The device is a full bridge driver rated to up to 100V which requires almost no external components. The dead time of the device is configured by adding a single resistor to the D_{SET} pins, to avoid shoot through conditions. The input of the device operates at TTL levels, and is suitable to operate at up to 1MHz.

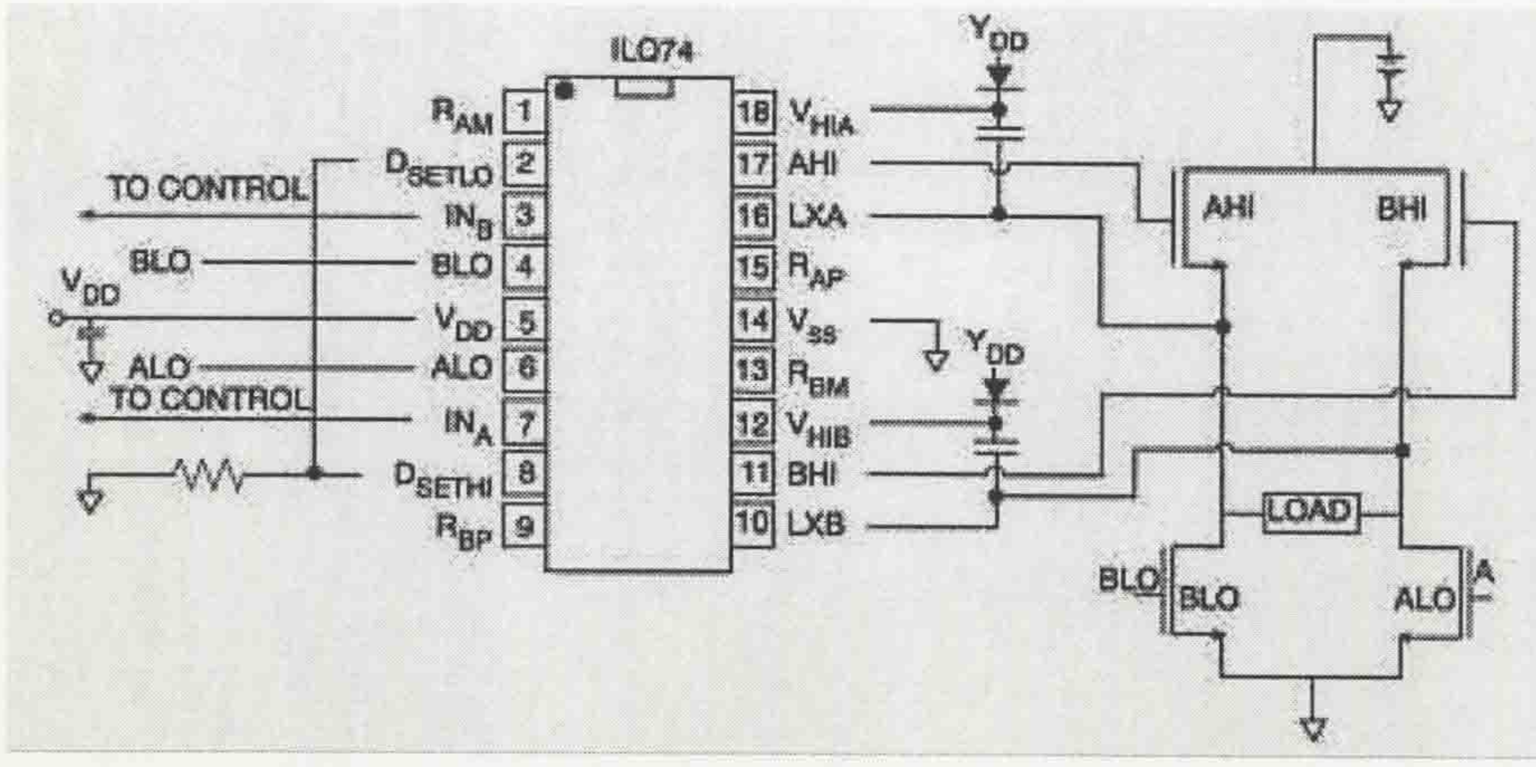

Figure 68, The EL7661 Bridge Driver

With the use of this device the component count and therefore the size of the power drives can be drastically reduced. In addition, the 10mA supply current this device needs greatly improves efficiency over the analogue devices used in the previous driver configuration.

4.4.3 Power Drive Stage

To power the thrusters a power device had to be chosen. A large range of MOSFET devices and IGBT devices are available which meet the 25V 20Amps criteria. A device with high switching speed and low drain source resistance is desired. The price of the devices does not vary greatly across the range of devices, so the choice of device was not restricted by this consideration.

The device finally chosen was the BUZ100L Mosfet manufactured by Siemens, rated at 60Amps continuous current at up to 50Volts. The drain source resistance R_{DSon} is 0.018Ω and the device comes in a compact TO-220 package. The device is capable of withstanding instantaneous currents of 240amp and is therefore capable of withstanding the surge currents present during device switchon.

The overspecification of the power device means that the snubber network is not needed to protect the device. This simplifies the design of the power stage further to contain only the power devices themselves. The mosfet devices have built in diode protection, so the addition of these is not necessary in the design of the power bridge.

4.4.4 The New Power Module

The block diagram for the new thruster power drive is shown in Figure 69.

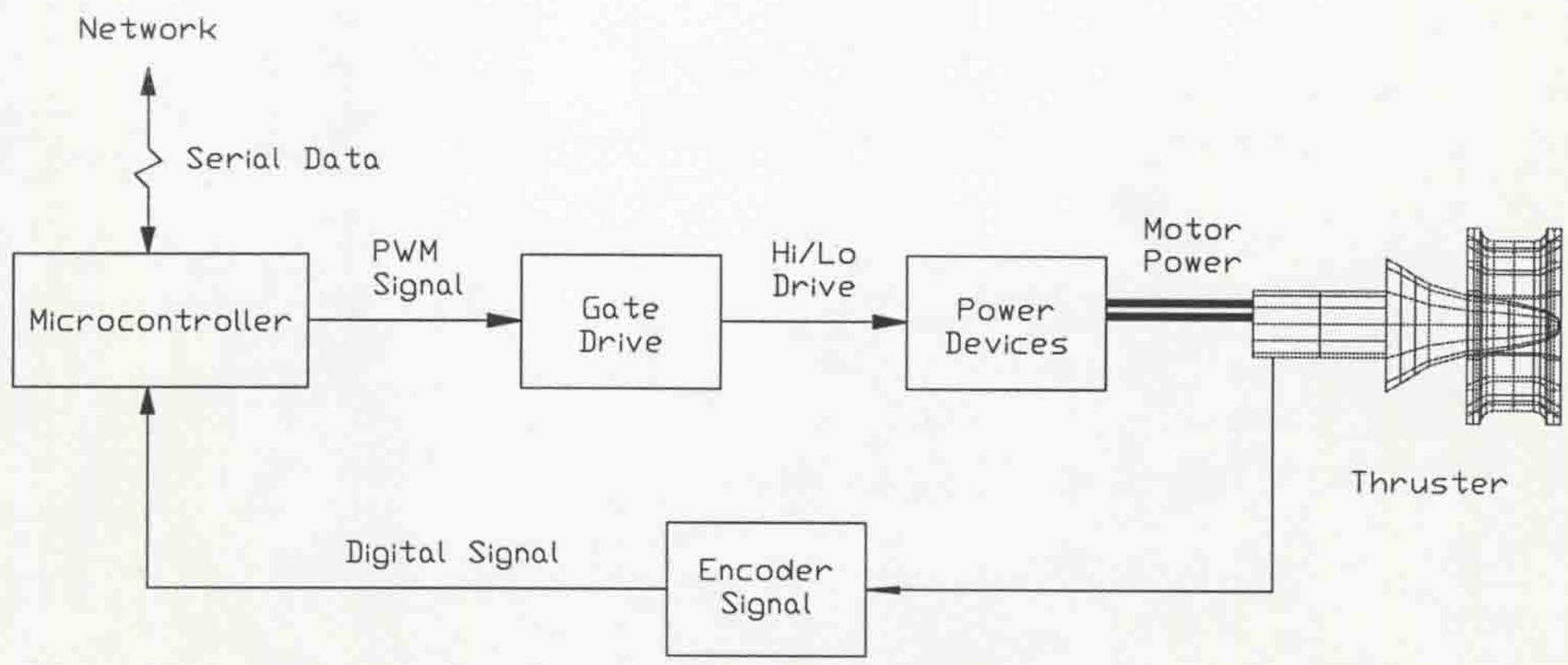

Figure 69, Power Drive Block Diagram

Using the arrangement shown in Figure 69 allows for a closed loop control system to be developed to control the thruster. The use of a current loop serial protocol such as RS483 allows the thrusters to be commoned together onto one set of serial lines. Thruster addressing techniques must then be introduced to allow for individual control. This addressing can be done either in software, or by using the hardware automatic address recognition facility built into the 87C51FA microcontroller.

The design of the new controllers also means that built in diagnostic routines can be created, so that the motors detect faults such as material snagged in the propeller, and can notify the control system whilst acting accordingly. The distributed processing provided by the system will help to lower network traffic, and lower the output requirements on the network, as to drive the thruster system the network now only requires one serial card.

A new design for the thruster drive is under-development, but has not yet been implemented on a working vehicle.

5. Sensor Systems

In order to be able to provide closed loop control of the vehicle it is essential that reliable sensor devices are connected. The minimum number of sensors required for positional control is one sensor per degree of freedom. As the Aquasphere has six degrees of freedom, a minimum of six sensors are required for full control. Having more than one sensor per axis, (of the same type, or of different types) can increase the accuracy of the sensor information through sensor fusion. This combination of multiple sensor inputs for navigation purposes is undertaken by human beings automatically for everyday tasks. Rigaud et al[5.1] state that for underwater systems "Generally the pilots build their own internal model of the submarine surroundings from the integration of various and highly noisy sensorial data". However Rigaud continues by saying that "the inherent inaccuracy in sensing, and the ambiguities in sensor interpretation lead to a heavy intellectual load on the pilot." As it is the aim of the project to reduce the complexity of the information from the vehicle, this data analysis should be carried out as much as possible by the vehicle, presenting the rationalised data in an acceptable form for the pilot.

Some degrees of freedom on the vehicle are more easily measured than others. The greatest problem for measurement is movement in the horizontal Z-plane, whilst measuring distance along the Z-axis vertically is relatively easy. All the sensors on the vehicle must be capable of marinisation, and robust enough to withstand the handling, transportation and operation of the vehicle. The sensors must also be small enough for mounting within the restricted space on the vehicle, and have a low power consumption. Financial constraints were also considered when making decisions upon which sensor could be used.

5.1 Sensor Specifications

At least one sensor is required for each degree of freedom of control. A large range of sensor systems are currently available; however due the limited market of ROV use, and the financially costly tasks for which ROVs are employed means that sensor systems can be prohibitively expensive. Therefore, a compromise had to be reached between sensor accuracy and cost.

For each type of sensor there are many types and variants with different operating conditions and specifications. It is beyond the scope of this thesis to provide an analysis of

all the sensor types and variants, but it is possible to give a general overview of some of the types of sensor used on ROV systems for navigational purposes.

5.1.1 Pressure Sensors

There are many type of pressure sensor available, but the type most frequently used on underwater vehicles are piezo-electric pressure transducers, or diaphragm gauge pressure transducers. These transducers measure the difference between the external pressure and a reference pressure. The reference pressure can either be in a sealed chamber, or a venting pipe sets the reference pressure to be at ambient air pressure. Normally the advantage of the vented tube pressure sensor is that a true pressure reading above local atmospheric pressure is possible with remote transducers. However, for a remotely operated vehicle it is not practical to have a tube returning to the surface. In this case the enclosure in which the pressure transducer is mounted contains the reference pressure, often a vacuum. Any deformation of the vessel containing the sensor body can alter this reference pressure, causing inaccuracy and non-linearity in the sensors readings. For underwater vehicles sealed chamber pressure transducers are preferred.

Since,

$$P = h\rho g \qquad \text{(Equation 23)}$$

the depth of the vehicle is therefore directly related to the pressure measured. Thus by using a pressure sensor the control of movement along the Z axis can be obtained.

5.1.2 Flow Sensors

Flow sensors are occasionally used on ROV systems to give an indication of relative water velocity. Typical types of sensor are helical screw flow sensors, ultrasonic flow sensors and magnetic paddle wheel flow sensors. With the exception of the ultrasonic flow sensor, these devices operate by measuring the interaction of the water with a mechanical device. For accurate flow measurements the medium whose flow rate is required should be enclosed by a tube, limiting the field of operation of the device. Standard flow sensors are difficult to mount on complex vehicles, as the turbulence caused by the vehicle through the water can alter the results vastly. Also, as a standard ROV does not normally fly in one direction for a great deal of time, these sensors have limited use for standard ROV tasks. However, on AUVs which travel principally along one axis these sensors can be very useful to calculate estimated position from the data log as described by Rigaud[5.1].

Complex and expensive electrostatic 3-axis flow sensors can be used, but for effective use these have to be mounted away from the vehicle. These types of sensor tend to be very expensive, and susceptible to damage. However, such devices can prove much more useful than the mechanical types for accurate measurement.

The use of these sensors can provide an estimate of distance travelled from the initial datum point. This is only possible when using expensive sensor arrays mounted correctly on a vehicle, and the estimate deteriorates over time. Commercial ROVs use the flow sensors as an indicator only when there is no other information available from other sensor. They are typically used for answering simple queries such as "am I still going forwards?", "is the vehicle moving left?" when visual information is not available.

5.1.3 Gyroscopic Sensors

Gyroscopic sensors are one of the most widely used sensing systems on ROVs and AUVs. Simple one axis gyroscopes to three axis solid state devices are available, with widely varying specifications for their performance. Weiß[5.2] analyses a gyroscopic device for its suitability as an orientation estimation device. He concludes by stating that gyroscopic devices can be used for accurate orientation estimation, but they may need periodic recalibration due to drift.

Normally only a single axis gyroscope is used to determine the vehicles heading. The information from the gyroscope is used to maintain automatic heading control from a datum point that is set by the operator. In this way a relative positional sensor is created, which recalibrates every time the new datum is selected. By using the gyroscope in this way the problems due to drift are reduced.

The gyroscope is most useful in conjunction with absolute positional sensors, which can provide recalibration information to the gyroscope system.

5.1.4 Flux-Gate Sensors

Flux gate magnetic sensors are used widely for underwater vehicle use. The magnetic heading reference data is usually required for the provision of automatic heading control. The standard configuration for this control uses the magnetic sensor as an accurate

stable directional reference(compass). This reference can then be used to update the attitude and navigational sensors, compensating for sensor drifts. Hellard[5.3] states that "accurate magnetic sensors can facilitate the use of lower precision inertial sensors whilst maintaining overall system performance, thus reducing navigation system costs."

Flux gate compasses are relatively cheap devices, but are susceptible to errors due to the pitch and roll of the vehicle. To compensate for this compass sensors are normally mounted on a gimbal arrangement to automatically provide attitude correction. However, this gimbaled system introduces errors when subjected to acceleration. Crocker[5.4] states that "Linear accelerations of the package will produce a tilt error in the gimbal or vertical reference". Crocker continues "The tilt will cause the instantaneous measured azimuth to be in error, and at high latitudes the error is quite high, approximately 2.3 degrees in azimuth for 1 degree east-west tilt." To compensate for these errors several solutions are possible. One widely used solution is to mount the compass on a gyroscopically balanced platform, however, Crocker[5.4] says this solution is "costly, and has other disadvantages such as higher power consumption fragility and complexity." Due to the low acceleration forces found on the Aquasphere a simple gimbaled solution was used, and the error accepted.

More importantly for this application was the magnetic influence from the vehicle itself. Distortions in the magnetic field due to the vehicle's ferromagnetic structure, power supplies, power cables and motors all cause errors to be introduced into the interpretation of the magnetic field. To cater for non-dynamic distortions in the field the compass can be calibrated by an operation known as a "compass swing". This operation involves slowly rotating the vehicle three-hundred and sixty degrees, allowing the flux to be measured, and the compass calibrated. To further reduce magnetic distortions the compass system was mounted in a remote location at the top of the vehicle, as far away from distortions as possible, in a magnetically inert casing.

The magnetic compass provides absolute heading information, and allows the yaw of the vehicle to be measured and controlled.

5.1.5 Accelerometers

The use of accelerometers for the calculation of the vehicle's position was investigated. Linear accelerometers are prone to drift and error, particularly in the measurement of low accelerations. The low intensity non-periodic accelerations provided by the movement of the Aquasphere are unsuitable for measurement by accelerometer packages,

and the integration of the accelerometer's signal to provide position would give large errors, increasing with time. However, in spite of these factors an accelerometer was included on the final vehicle for test purposes. Only one sensor was included on the vehicle, to give an estimation of acceleration and distance traveled along the X axis.

5.1.6 Attitude Sensors

To be able to control the pitch and roll of the vehicle attitude sensors were required. Various type of inclinometers are available, but most are based on either a mercury resistive method, or a pendulum arm method. From the large number of similar products available two devices were purchased. These were mounted on the final vehicle vertically, at ninety degrees to one another to provide information on the vehicles pitch and roll.

5.1.7 GPS Systems

The GPS, or Global Positioning System, was developed to provide precise position, velocity, and time information. The system was designed to provide this navigation capability anywhere in the world at any time regardless of weather conditions. GPS systems comprise three separate sub-systems, normally called segments. The three main segments are:-

GPS Space Segment — This consists of 24 operational satellites in six orbital planes. The satellites operate in circular orbits with a 12-hour period. The precise spacing of satellites in orbit is arranged so that a minimum of four will be in view of any user, thereby ensuring worldwide coverage. The satellites broadcast navigation and system data, atmospheric propagation correction data, and satellite clock bias information.

GPS Control Segment — This includes a number of monitor stations and ground antennas located through out the world. The monitor stations use passive GPS receivers to track all the satellites in view in order to accumulate ranging data from satellite signals. This information is processed at the Master Control Station to determine satellite orbits and to update the navigation message of each individual satellite by transmitting the data to the satellite from the ground antennas.

•GPS User Segment — This consists of User Equipment (UE) sets and associated support equipment. UE sets, utilizing data transmitted from the satellites,

derive navigation and time information for local use. It is this equipment that can be mounted on underwater vehicles.

The operation of the user equipment is simple. Each satellite periodically transmits a signal at a specified time. The user, assuming time of transmission and knowing time of receipt of each signal, computes a range to each satellite called a pseudorange. The user then constructs four equations using four calculated ranges and solves for position in three dimensions and a clock bias. A diagram of the workings of GPS, as described by Navstar[5.5], is shown in Figure 70.

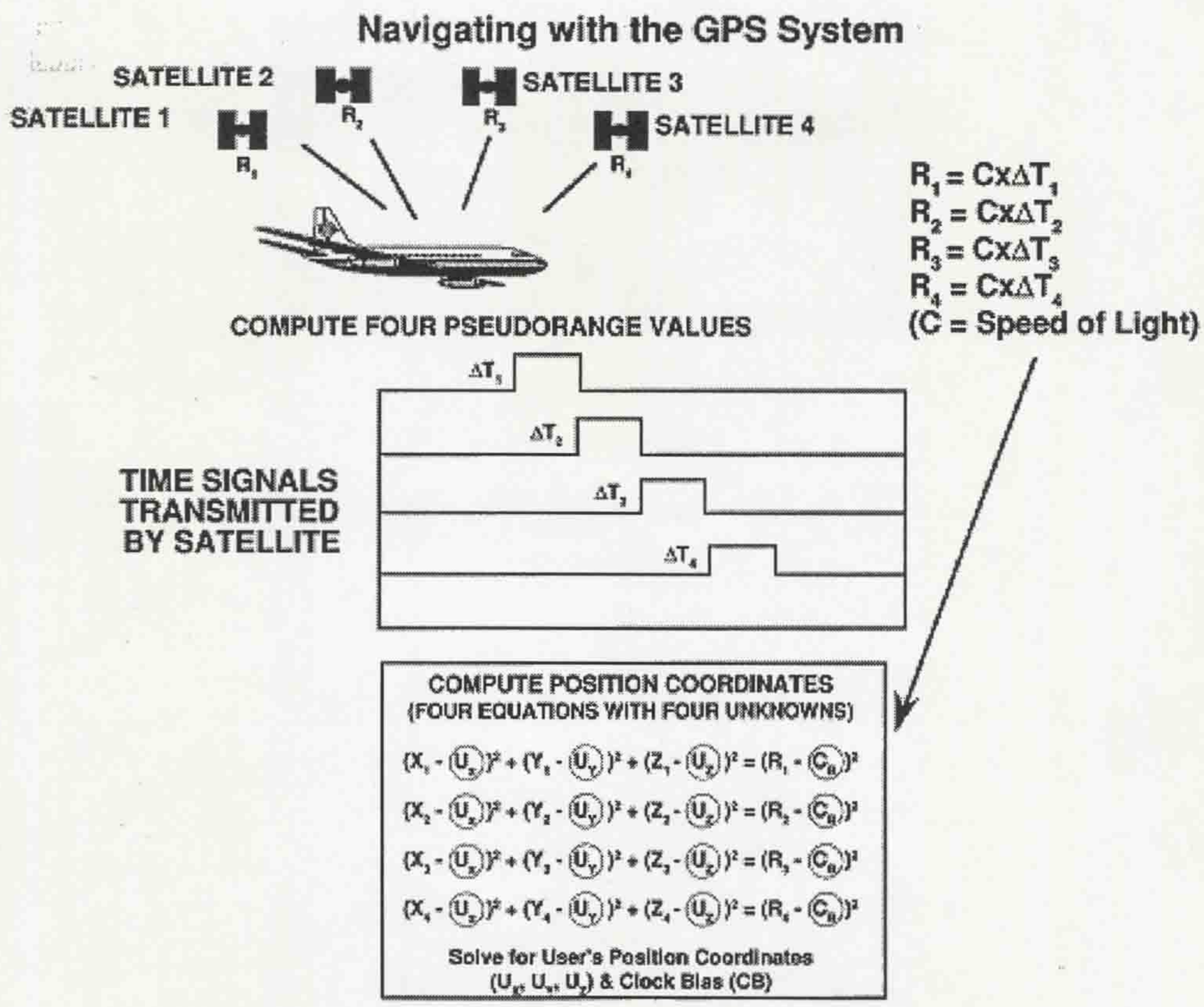

Figure 70, Navigation Using GPS

GPS provides two different levels of accuracy. The Standard Positioning Service (SPS), used primarily by the civil community, provides a navigational accuracy of 100 meters. The Precise Positioning Service (PPS) provides full system accuracy of 16 meters in position, 0.1 meter/second in velocity, and time to 100 nanoseconds. This service is primarily used by military establishments.

GPS data can only be derived from a surface antenna due to the operating frequency of transmission, so this system can only be used when the vehicle is at the surface. However, the system provides accurate data to give the vehicle's starting location, and subsequent motion can be described from this reference point. Goodchild[5.6] states that "by adding the sea surface facility, enhanced accuracy navigation can be achieved for underwater operations because it provides a mechanisms for the remote fine tuning the sea water data instrumentation and the inertial navigation sensors." The GPS system on an underwater vehicle detailed by Kwak *et al* [5.7]can be used to reset the calibrations of other sensors, when the confidence threshold of the sensors is passed. For example, a vehicles dead reckoning system becomes inaccurate over time. If the data from the guidance system is felt to be inaccurate then the vehicle can surface, recalibrating the dead reckoning system using GPS.

The use of a GPS system can provide accurate positional information for the vehicle in three dimensions, the drawback being that this data is only present when carrying out surface operations.

5.1.8 Acoustic Sensors.

There are many types of acoustic sensors available for use on ROVs. They range from simple echosounders to complex three dimensional sonar systems.

Echosounders have a narrow divergence beam transducer, which transmits a pulse and then works out the distance by time of flight methods of the return pulse. This method gives the distance of a point from the vehicle, and these transducers are normally used to measure draught. By sweeping the transducer head a scan of points can be made, creating a two dimensional slice of range information. By correlating the data from these scans relative movement of the vehicle with respect to the surface underneath the vehicle may be possible. The update rate of these systems is related to the size of the scanning grid. Increasing resolution by increasing the number of sample points has to be carefully considered, as a slower frame rate may prove inaccurate on a dynamic system. Often "T" shaped scans are taken, to give relative velocity for the vehicle in X and Y axes, whilst keeping the sampled data points to a minimum.

Echolocators can be used within a fixed environment. Transponders transmit pulses periodically at a set frequency. By using time of flight techniques the distance from a beacon

can be determined. By triangulation the absolute position of the vehicle within the environment can be known. The different beacons can be differentiated between by the frequency signature of the transmitted signals. Echolocation sensors can be accurate at determining the absolute position of a vehicle, however, these sensors are only of any use within a predetermined area. The beacons must be placed accurately prior to the use of a vehicle, which can be an expensive and time consuming task. The beacons must also be recovered after an operation, incurring further costs. These sensors are only used for vehicles carrying out tasks in a heavily used area of water, for example at the legs of an oil-rig. Using echolocation devices in such an environment can cause additional problems such as path distortion and multipath reflections.

Recent research has lead to the development of acoustic three dimensional cameras for underwater vehicles. Hansen[5.8] gives details for a three dimensional camera system currently being developed. This device is intended for ROV applications, since "the camera will be able to provide acoustically based visual information to the user of an ROV - even if turbidity restricts visual sight." Hansen also states that "unlike conventional sonars, the camera provides three dimensional information and the image is based on one acoustic pulse - making the camera virtually insensitive to movement". This camera, as well as other such devices being developed will be able to give the vehicle significantly more navigational information. Although not an absolute position sensor the device should allow navigation in poor visibility conditions, allowing for full three dimensional control within the scanning range of the device.

5.1.9 Visual Systems

The visual information provided by a camera mounted on the vehicle is the most important sensing system that the operator has. It enables the operator to navigate the vehicle through the water when visibility is good, and allows close station keeping relative to fixed objects even in low light and poor visibility conditions. The vision system designed by Yu[5.9] incorporates 3D vision capabilities with image recognition techniques. The aim of the vision system was to provide positional information for objects in the water, relative to the base frame of the vehicle. The system utilized a two spot laser triangulation system to calculate the distance of objects and their relative X, Y and Z positions from the vehicle. This positional information could then be read by the rest of the vehicles subsystems to provide relative positional control in the X, Y and Z axes.

Other commercial systems are available for 3D vision underwater, normally using a single spot fixed camera system. However, due to the limited visibility that maybe encountered underwater these techniques are not widely used.

5.1.10 Navigationally Related Sensors

Although these sensors do not provide any navigational information themselves they may be used in conjunction with other sensors to provide calibration and status information. For vehicles which operate in widely varying locations the use of these sensors is recommended.

5.1.10.1 Temperature Sensors

Several types of temperature sensors can be used to measure water temperature. Platinum resistive temperature sensors are widely used, due to their linearity and low-cost. Temperature measurement is important for the calibration of ultrasonic devices, as according to Wood[5.10] "the speed of sound in water increases with temperature by approximately 0.2% per degree centigrade".

5.1.10.2 Salinity Sensor

Salinity is one of the parameters which affects the transmission speed of sound in water. For accurate calibration of acoustic devices salinity sensors are required According to Wood[5.10] the speed of sound in water increases by 0.1% per 0.1% increase in salinity.

5.2 The Aquasphere's Sensor Systems

For the control of the Aquasphere, sensors had to be chosen from ranges of sensor outlined in the previous section. As mentioned previously these sensors are primarily bought by military establishments, offshore support companies and petrochemical companies in small quantities. The price of these devices is therefore relatively high, so compromises had to be made between performance and cost when purchasing. It was not possible to purchase or develop examples of all the sensor types for inclusion on the final vehicle so the most important sensors needed to complete our tasks were chosen. The following sensor systems were used on the Aquasphere.

Sensor	Model	Information Provided	Specification
Vision System	Created by The University of Liverpool	Relative position of target object in X, Y and Z axes	Resolution ±2mm, Working area approximately 1m by 1m by 5m
Acoustic System	Created by The University of Thrace	N/A	N/A
Flux-Gate Compass	Autohelm ST50 Steering Compass	Absolute Heading (Yaw)	Heading Error 0.1° Maximum. F.S.D. 0-360°
Ultrasonic Echo Sounder	Autohelm Z091	Relative distance from bottom	F_{NOM} = 200kHz, Beamwidth @-3db Minimum range 0.7m
Flow Sensor (with integral temperature sensor)	Autohelm ST500	Linear velocity perpendicular to angle of mounting	Resolution 6.67Hz/knot Thermister(10k ohm at 25 degrees Celsius)
Depth Sensor	Druck PTX-530-00	Absolute Z position	4-20mA dc, 1 Bar (10m) Maximum.
Rate Gyroscope	Futaba FPG153BB	Gives rotational movement about principle axis	Output proportional to angular acceleration. Output range 4.8-6V
Accelerometer	Entran EGC-500DS	Linear Acceleration along principle axis	Range ±500g, Sensitivity ±0.4mV/g Resonant frequency 3.5kHz
Inclinometer	Accustar DC-OUT30	Absolute measurement for Pitch and Roll.	Full Scale Motion ±30°, Resolution ±0.1° Output at F.S.±12V

5.3 Sensor Performance

The sensor array on the Aquasphere presented several problems for the control of the vehicle. The limitations and inaccuracies of some of the sensor systems meant that full positional control of the vehicle was not possible. It was found that some of the sensor systems were unsuitable or inaccurate. As these problems were found out only in the final testing stage it was impossible to replace them with more suitable products. The performance of the individual systems, and the effects of their performance is detailed in the following sections.

5.3.1 Vision System Performance

The vision system used on the vehicle was in the process of development and testing at the time of the trials. Although the accuracy of the system is sufficient the update between readings is in the order of several seconds. When this system was mounted on the Aquasphere the movement of the vehicle was such that the vision system was unable to "lock on" to the object. Thus when the vehicle was free-swimming the control system was unable to use the information from the vision system. Yu[5.9] proposes techniques for the improvement of the vision system, using laser switching techniques, and improved processing methods.

This deficiency in the vision system meant that automatic guidance towards the task object was not possible. When the vehicle was parked stationary next to target the object it was possible for this information to be used successfully, as was demonstrated in the trials.

5.3.2 Acoustic System Performance

As part of the collaborative project an acoustic sensing system capable of absolute position measurement within a known environment was to be designed and created by the University of Thrace. However, at the time of the final trials a working prototype had not been constructed, so this system was unavailable for use.

As there was no position information for the X and Y planes this meant that absolute positional control along these axes was not possible.

5.3.3 NMEA Interfaced Devices

The equipment that used the NMEA (National Marine Electronics Association) data transmission protocol had problems introduced due to this interfacing method. The sensors using this standard are the flux gate compass, the echo sounder and the flow sensor. The problem with the NMEA protocol is that the data is only output at approximately 1Hz, making control of the vehicle difficult due to the slow sampling rate. Apart from this low sample speed the NMEA data proved reliable and robust This protocol is more fully detailed in the following chapter.

5.3.3.1 Flux Gate Compass Performance

As mentioned in section 5.1.4 problems can be associated with the use of flux gate systems. Precautions were taken to minimize errors from the vehicle's electronics by mounting the gimbaled sensor in a remote location on the vehicle in a magnetically inert casing. The sensor was also calibrated by swinging the compass once assembled on the full vehicle. This sensor performed well, with low noise. Except for the low data rate due to the NMEA interface the sensor performed well throughout the trials.

5.3.3.2 Echosounder Performance

An echosounder was purchased early on in the research to measure the vehicle's draft. This sensor was mounted in the sensor box which was fitted at the base of the original prototype vehicle. In initial tests this sensor performed well, with the data being correlated with the data from the pressure sensor. However, on the final vehicle the location of the sensor box was changed to maintain the stability of the vehicle which occluded the line of sight of the sensor. As an additional sensor provided by the University of Thrace was to be provided the omission of this sensor was not thought to be important. Therefore during the final tests the use of this sensor was not available. By mounting the sensor remotely, or by relocating the sensor box this sensor could be used in the future.

5.3.3.3 Flow Sensor Performance

As the flow sensor is also mounted on the base of the sensor box similar problems arose in the use of this sensor. The close proximity of the sensor to the networking box reduced flow over the sensor, limiting its use. Again, as the vehicle was supposed to have an

acoustic positioning system, from which the velocity could be derived, the removal of this sensor from active use was not thought critical.

In initial tests the performance of this sensor was poor. The paddle wheel used is capable of measuring speeds of up to 58 knots, and at the low velocity of the Aquasphere the measurements proved inaccurate. Also due to the shape of the Aquasphere there are no areas within the shell subject to only linear flow. The turbidicity of the flow around the paddle wheel can further complicate matters.

5.3.4 Pressure Sensor Performance

The pressure sensor performed excellently throughout the project. The sensor proved to be reliable, with low noise levels and high linearity. No problems were experienced with this transducer, from its initial installation to the final trials. The pressure sensor provides the absolute Z position of the vehicle, allowing control along the vertical axis.

5.3.5 Gyroscope Performance

Due to the slow rate of update from the flux gate compass a gyroscope was included in the sensor array. The aim was to use the gyroscope's data in between absolute position samples from the compass. Every sample from the flux gate would then be used to recalibrate the gyroscope. In this way gyroscopic drift would be reduced. However, the results from the gyroscope mounted on the vehicle were unsatisfactory. It is unclear whether the fault lay in the interfacing hardware needed for the sensor, or in the sampling and integration routines. The results from the gyroscope gave fluctuating values, even when the vehicle was stationary. In the final trials it was felt that the data received from the gyroscope was incorrect, so all gyroscopic sensor data was ignored. The electrical noise caused by switched mode power supplies could have possibly caused signal errors which would have been compounded by integration.

5.3.6 Accelerometer Performance

The accelerometer mounted on the vehicle had a maximum tolerance of 500g, and gave 0.4mV/g. For the very low accelerations found on the vehicle the sensitivity of the device was insufficient. The electrical noise and quantization error of the digitizing circuitry was in excess of the measured signal, rendering the data from the sensor useless. It was hoped that such a system could have been used in conjunction with the Acoustic system, providing two systems which could be compared and correlated. The use of a more expensive sensor unit may have provided us with more suitable data. A more suitable device would have a full-scale output at 0.5g or less, with very low bandwidth (0-1Hz). Such devices are not readily available.

5.3.7 Inclinometer Performance

The inclinometers mounted on the vehicle performed well throughout the trials. Initial bench tests showed that the quoted accuracy and linearity figures were correct, and that the interfacing hardware and software was working. The two inclinometers mounted perpendicularly to one another allowed the control system to control the roll and pitch of the vehicle.

5.4 Sensor Summary

Due to the disappointing results from some of the sensors, and the lack of availability of other sensors, full control of the vehicle was no longer possible. In addition sensor fusion techniques could not be used as there was no duplication of results from the sensor array. The sensors available provided information for the vehicles roll, pitch, yaw and depth, allowing control of these variables.

6. Networking Software

In order for the control system to be able to work, the software needed to have access to the sensor information. Additionally the control system had to operate in two modes. Firstly as a stand alone control system, and secondly to be controlled as a slave system to the overall vehicle user interface. In order to do this, networking software had to be developed which would provide the necessary sensor information and system demands transparently to the simultaneously operating control system.

As the IBM compatible computer is not a true multitasking system a method had to be developed to allow for the simultaneous operation of these two processes. To do this an interrupt driven system was created to time-slice multiplex the processes, allowing the two processes to run simultaneously.

In addition, a data transmission protocol for the networked data transmission and reception had to be created in conjunction with Houghton[6.1], the designer of the transputer based networking system.

To be able to transfer information between the sensors, control system and thrusters each of these systems had to be interfaced into the transputer network. Due to the diverse nature of these units different interfacing techniques had to be developed for each system. Due to the modular design of the networking system, it was possible to add the capability for interfacing with the sensors by adding the appropriate interface card and writing the appropriate signal conditioning routines. Prior to the final trials when the networking system had been designed and implemented, an alternative method had to be used to control the vehicle. This method was implemented on an IBM compatible computer, using PC based interface cards.

6.1 The Original Interface System

The interfacing, control and signal processing techniques were originally carried out on a PC based system. For the initial trials on the vehicle the sensors present were the depth sensor, and the Autohelm sensors connected to the NMEA data connection. In addition the thruster power box was connected requiring six analogue thruster demands. To interface to these devices interface cards were purchased for the personal computer to allow the control system to transfer data between itself and the vehicle components. Without the transputer

interface to multiplex data to the vehicle separate umbilicals were necessary for sensor data acquisition, thruster power and thruster control.

To interface to the vehicles systems the following interface devices were used:-

6.1.1 PC61A Interface Card

This card manufactured by Amplicon[6.2] is a full length sixteen channel differential analogue to digital input card for the IBM personal computer (ISA bus). The board is configured as standard to receive 4-20mA input signals, but can be hardware reconfigured to receive other input signals. The card is capable of a resolution of 5µA from the 12bit ADC with a sampling rate of up to 30Hz. This slow update rate is due to the built in dual slope integration and filtering techniques used by the card's hardware.

This card was connected to the pressure sensor mounted on the vehicle, and was used extensively for the tests on the Aquacube, and for the preliminary tests on the Aquasphere. The interfacing software written in C, provided calibration routines for the sensor, and a routine to request the current vehicles depth. The structure of the depth measurement software is shown below:-

```
Main
    Calibrate Sensor
        1.Get Depth
            Measure sensor current from PC61A card
            Calculate depth from sensor current and calibration data
            Return Depth
        2. Control Routine
        3. Return to 1.
End
```

6.1.2 PC 66 Interface Card

The PC 66 interface card is a twelve channel digital to analogue output card. The card fills an 8-bit full length slot in a PC, and provides the computer with 12 channels of bipolar analogue output up to ±10volts. The card has built in remote sensing to provide a stable output signal, and reducing voltage drops in long cables. The card has an output resolution of 12 bits, giving a 1 in 4095 resolution, relative the output range selected. The setting time of the card is less than 1μS allowing for almost instantaneous demand updates. The structure of the interfacing software, written in C, is shown below.

```
Main
    1. Control System Calculates Thruster Demands
    2. Start at Thruster 1
    3. Check if Thruster Demand has Changed
    4.If value has changed then set the new value
        Set_Thruster (Thruster_Number, Demand_Voltage)
            Convert voltage to 12 bit data for card
            See which i/o port to output data to for this thruster
            Output data to relevant thruster's port
    5. If not the last thruster go to 2. To check the next thruster.
    6. Goto 1.
End
```

This card was connected via a signal umbilical to the thruster PWM generation card mounted on the vehicle. The use of this card allowed the control of the thrusters quickly and accurately, without the need for the networking system to be installed and operational. This method was used extensively, until the integration of all the systems in the final trails.

6.1.3 NMEA Sensor Interfacing

To read in data from the Autohelm sensors the data from the NMEA compatible interface had to be interpreted. According to NMEA News, "The National Marine Electronics Association (NMEA) is dedicated to the education and advancement of the marine electronics industry and the market which it serves. It is a non-profit association composed of manufacturers, distributors, dealers, educational institutions, and others interested in peripheral marine electronics occupations". Although not an ISO recognized standard this protocol has the backing of several of the major nautical electronics companies, and is widely used for nautical navigation equipment. The NMEA 0183 standard defines an electrical interface and data protocol for communications between marine instrumentation.

6.1.3.1 NMEA Hardware Interface

The NMEA standard allows a single "talker", and several "listeners" on one circuit. The recommended interconnect wiring is a shielded twisted pair, with the shield grounded only at the talker. NMEA-0183 recommends that the talker output comply with EIA-422. This is a differential system, having two signal lines, A and B. The voltages on the "A" line correspond to those on the older TTL single wire specified in the earlier NMEA 0182 standard, while the "B" voltages are reversed (while "A" is at +5, "B" is at ground, and vice versa).

Autohelm "Seatalk" is a proprietary bus for communication between various Autohelm instruments. Some of the instruments can act as NMEA-0183 talkers or listeners. Data received from an external NMEA-0183 device will, if Seatalk understands the sentence, be re-transmitted, but not necessarily in the same sentence type. The specific sentences sent will depend on the data available on the Seatalk bus (i.e. sentences containing wind speed and direction will only be sent if the Autohelm system includes a wind instrument). The ST50 Multi-Function adapter acts as an NMEA talker device adapter on the Seatalk bus, reading in the sensor data present on the Seatalk bus, and outputting the data in an NMEA format.

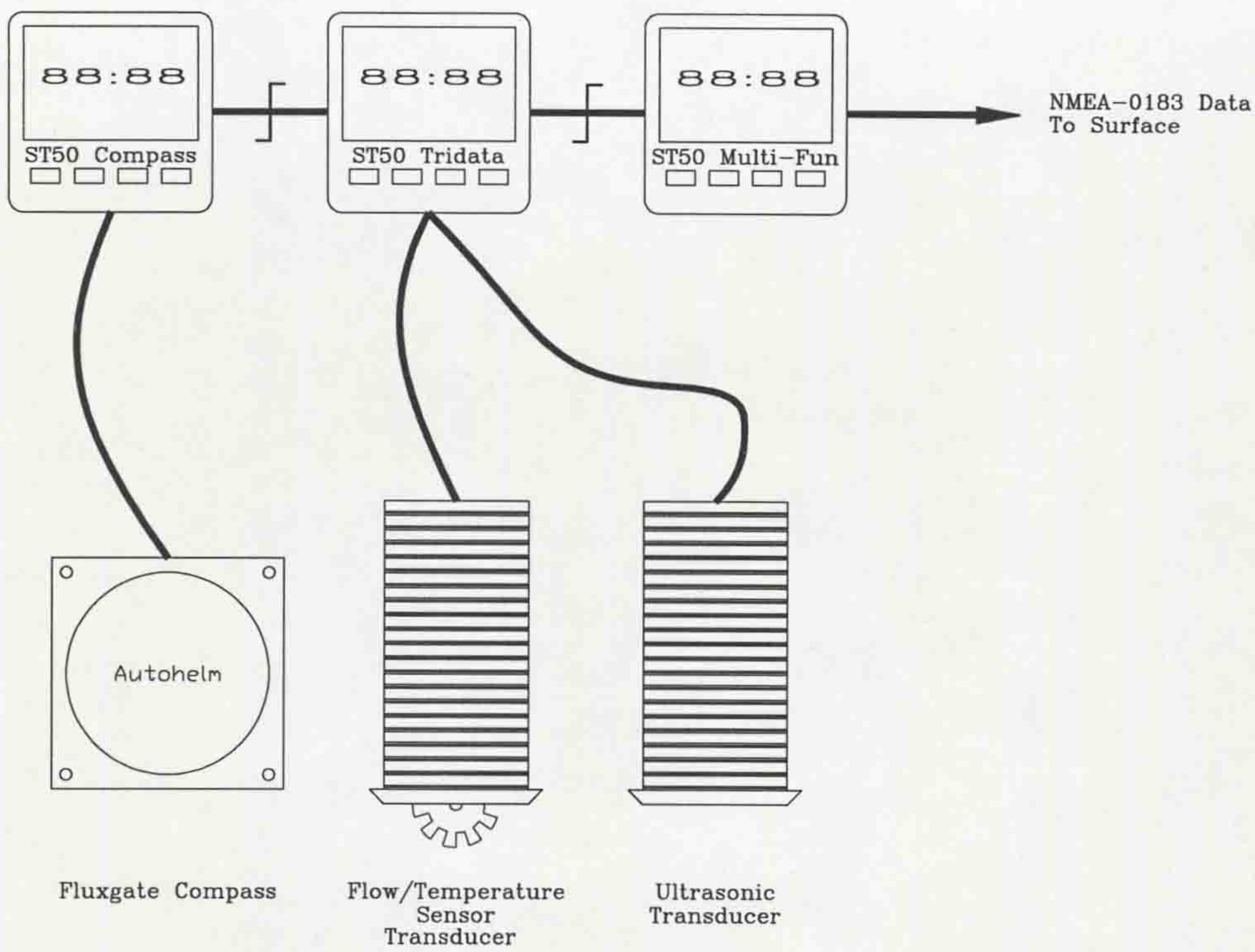

Figure 71, Autohelm Sensor Connection

Figure 71 shows the interconnection of the Autohelm devices. Inter device connection is carried out using Autohelm's proprietary Seatalk data bus, whilst the multi-function adapter provides the NMEA data.

The data structure provided by the Autohelm sensor complied with NMEA 0183 standards, but the data signal itself complied with the earlier TTL level specification. To convert the data from NMEA to RS232 compatible the data had to be first inverted, and then level shifted to RS232 standards.

To do this a simple interface circuit was built containing a 7404 inverter and a MAX232 device. The MAX232 manufactured by Maxim is a device which converts a TTL signal to RS232 and vice versa. It needs only a +5V input and ground, as internal charge pump converters give the bipolar output necessary for RS232 levels. A block diagram of this is shown in Figure 72.

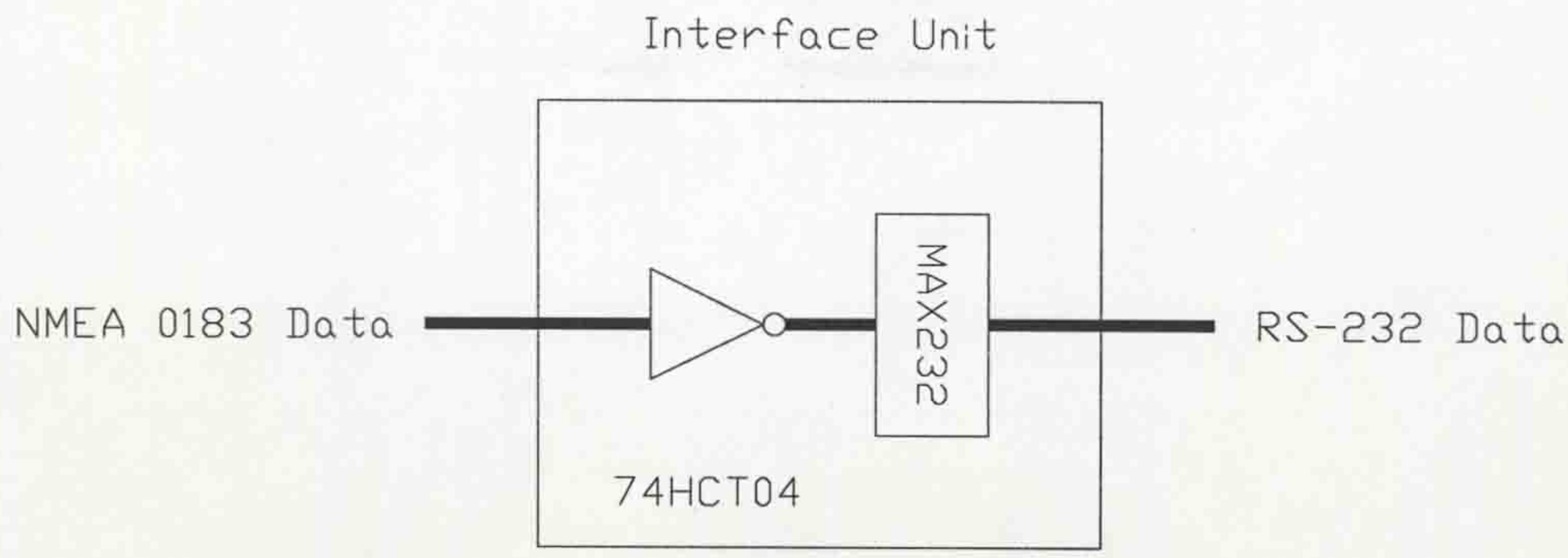

Figure 72, NMEA Interface

6.1.3.2 NMEA 0183 Sentence Format

Under the NMEA-0183 standard, all characters used are printable ASCII text (plus carriage return and line feed). NMEA-0183 data is sent using :-

- Baud Rate = 4800
- Data Bits = 8 (d7 = 0)
- Parity = None
- Stop Bits = One

The data is transmitted in the form of "sentences". Each sentence starts with a "$", a two letter "talker ID", a three letter "sentence ID", followed by a number of data fields separated by commas, and terminated by an optional checksum, and a carriage return/line feed. A sentence may contain up to 82 characters including the "$" and CR/LF. If data for a field is not available, the field is simply omitted, but the commas that would delimit it are still sent, with no space between them. Since some fields are variable width, or may be omitted, the receiver software should locate desired data fields by parsing commas, rather than by character position within the sentence. The optional checksum field consists of a "*" and two hex digits representing the exclusive OR of all characters between, but not including, the "$" and "*".

The standard allows individual manufacturers to define proprietary sentence formats. These sentences start with "$P", then a 3 letter manufacturer ID, followed by whatever data the manufacturer wishes, following the general format of the standard sentences.

Some common talker IDs are:

GP	Global Positioning System receiver
LC	Loran-C receiver
OM	Omega Navigation receiver
II	Integrated Instrumentation (e.g. Autohelm Seatalk system)

The talker unit typically sends a group of sentences at intervals determined by the unit's update rate. This update rate for the Autohelm devices is approximately 1Hz. There are many different devices which can be connected to the NMEA data connection, all with different talker IDs and sentence IDs. The full list of devices and their data protocols are included in the NMEA [6.3] specification. However, the address fields transmitted by the Autohelm sensors used is included in section 6.1.3.2.1.

6.1.3.2.1 Sensor Address Information

The sensor addresses transmitted by the Autohelm sensors mounted on the vehicle are:-

DBT - Depth below transducer

e.g. DBT,0017.6,f,0005.4,M

0017.6,f	=	17.6 feet
0005.4,M	=	5.4 Metres

Either the distance in feet can be given, or in meters, or both fields may be transmitted.

HDM - Heading, Magnetic

e.g. HDM,235.,M

235.,M	=	Heading 235°. Magnetic

MTW - Water temperature, Celsius

e.g. MTW,11.,C

11.,C	=	11 degrees Celsius.

VHW - Velocity, Heading through Water

e.g. VHW,333.,T,333.,M,05.00,N,09.26,K

333.,T	=	True Heading 333°
333.,M	=	Magnetic Heading 333°
05.00,N	=	Velocity through water 5 knots.
09.26,K	=	Velocity through water 9.26 kilometers/hour

The velocity and heading information may be duplicated to give both types of reading, or only one of each type may be displayed.

6.1.3.2.2 Sample NMEA Data

A sample of the NMEA data logged from the vehicle is shown below.

```
$IIDBT,0002.7,f,,,,
$IIVHW,333.,T,333.,M,00.00,N,,
$IIHDM,333.,M
$IIMTW,20.,C
$IIDBT,0002.7,f,,,,
$IIVHW,333.,T,333.,M,00.00,N,,
$IIHDM,333.,M
$IIMTW,20.,C
$IIDBT,0002.7,f,,,,
$IIVHW,333.,T,333.,M,00.00,N,,
$IIHDM,333.,M
```

Table 6, Sample NMEA Data

From the sample data given in Table 6 it can be seen that the information was transmitted from an integrated instrument (talker id "II"), and that the time of data sampling the vehicle was stationary, with 2.7 feet below the vehicle, heading towards 333° in water at 20° Celsius.

6.1.3.3 NMEA Interpretation Software

For the control system to be able to accept the NMEA, data software had to be written to interpret the data, and store it as variables for the control system to use. The interpretation software had to work in conjunction with the control system, so an interrupt driven routine was written to read in the serial port data and store this information in a rolling buffer. Every time the control loop had finished a calculating the output values this buffer was checked to see if new data had been received. The structure of this procedure is shown below

```
START
    Check to see if end of sentence character has been received
    If not then Goto FINISH
        If an end of sentence has been received then check for a previous start of sentence
        character
        If no start of sentence found then discard incomplete information and Goto FINISH
        If start of sentence is found then
            Parse the Address field to see which data it is
            Assign the data to a variable according to the address field
            Delete processed data from buffer.
FINISH
```

This software was implemented into the control system and was used extensively until the final trials, when this interpretation algorithm was programmed into the networking software itself. The hardware and software performed perfectly throughout the testing phase of the Aquasphere.

6.2 Transputer Networking System

For the final trials on the Aquasphere use was made of the transputer based networking system developed by Houghton[6.1]. The use of this system allowed all sensor and vehicle data to be available to all the vehicle's subsystems, and allowed the transmission of control and demand signals from the master control computer to the sub-systems. Also by using the transputer network the number of umbilicals needed were reduced, as data could be digitally multiplexed along the data lines, removing the need for separate sensor and control cables. The network could also be used to carry out the acquisition and processing of information, freeing the control system from this task.

To use the network with the control system both the previously developed hardware and software systems had to be changed. Methods of P.C. interfacing, data transmission, sensor interfacing and thruster interfacing had to be implemented, in addition to control system modifications.

6.2.1 P.C. Interfacing

For the IBM compatible personal computer to be able to talk to the networking system, and vice versa, an interfacing technique had to be developed. The basis for the hardware communication between the P.C. and the transputer network is the transputer interface card designed by Chambers[6.4], shown in Figure 73.

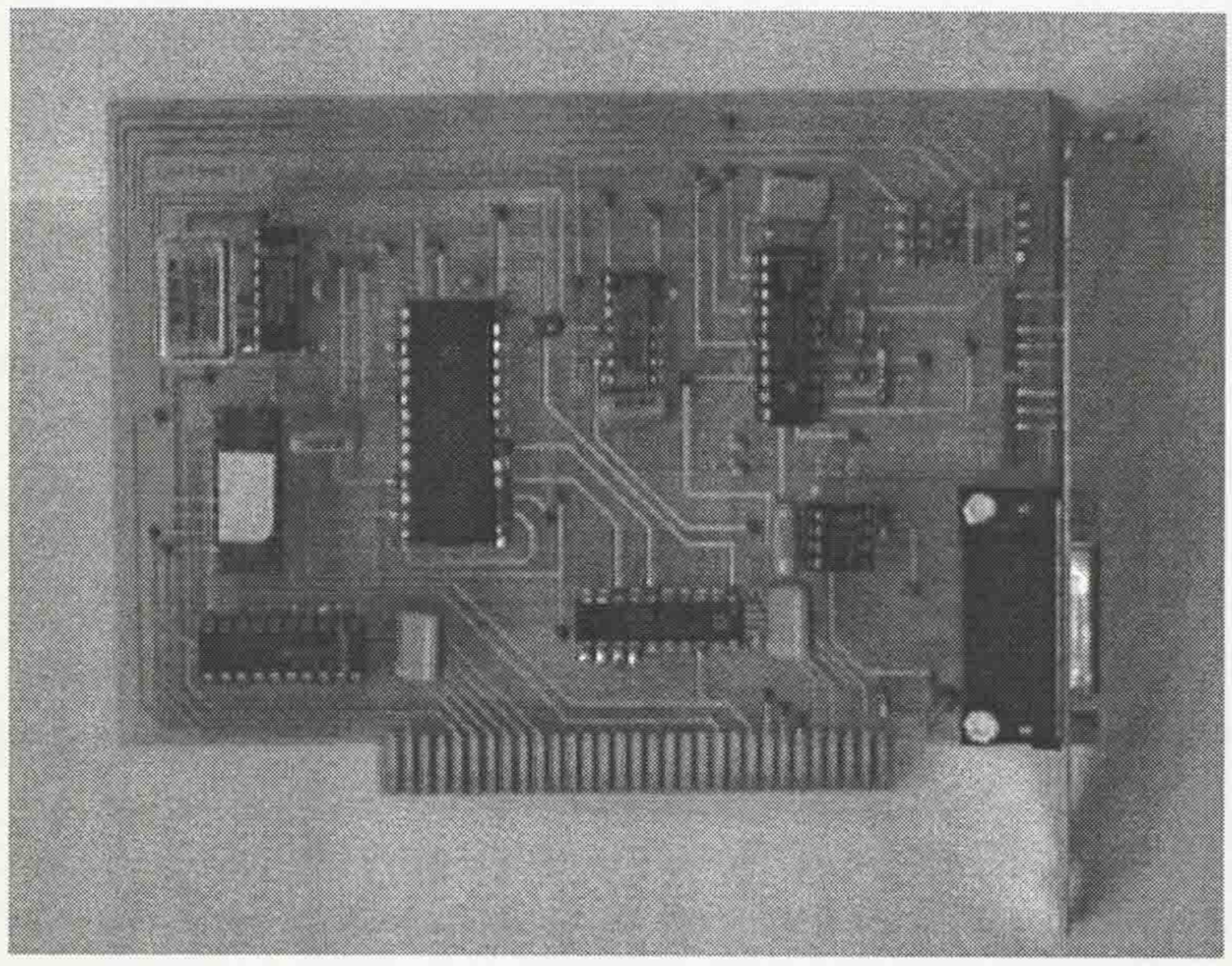

Figure 73, The PC -Transputer Interface Card

The Transputer interface card contains a IMS CO12 Link adapter which converts the microprocessor bus information to Inmos link compatible data. The data is buffered to provide RS422 output to allow a more noise immune system capable of connection over long cable lengths.

6.2.1.1 PC Hardware Interface Software

To communicate with the hardware interface card software had to be written in "C" which would allow background communication to the network, as well as running the control software. To accomplish this software was written to communicate with the card, and interpret the incoming data.

Communication with the card is relatively simple. The card contains two status registers, and two data registers. The status registers are:-

- Previously transmitted data accepted
- New data received

whilst the data registers are:-

- Data output for link
- Data received from link

The data is transmitted one byte at time. To transmit to the network the output status is checked to see if a byte can be sent, if it can be sent the value is transmitted until the data transmission is acknowledged. If the data cannot be sent within a time out period the process returns an error signal, indicating transmission failure.

To receive data the data received flag has to be polled. When the flag indicates data has arrived the byte can be read from the data received register. The drawback of this system is that processor time is wasted polling the port. A more efficient solution would to have a hardware interrupt triggered when data is received. This would allow the PC processor to be freed to accomplish other tasks.

6.2.2 Network Data Protocol

Before the rest of this system could be developed the basic network data protocol had to be decided upon. In addition to the list of supported command types a basic sentence structure had to be agreed. The network is divided up into several nodes each of which has an identifying address. The data packets must be able to transmit data efficiently between each of these nodes.

The protocol shown in Table 7, taken from the MAST report[6.5] shows the finally agreed protocol. The two header bytes provide synchronization for the packet, followed by an identifiers for the destination address and the source address for the packet. These identifiers provide additional safety checks, as only certain systems were authorized to talk to other systems. For example only the control system could make thruster demand requests, whilst it was not allowed to send motion requests to the manipulator arm. Additional checks are made on the data by checking the length of the data with the length value specified, and a two byte two's complement checksum is also calculated to check for any data corruption during transmission.

Byte	Function	Value	Length
1	Header 1st Byte	0	1
2	Header 2nd Byte	1	1
3	Destination Node	Node	1
4	Source Node	Node	1
5	Length of Data Field	(len)	1
(6 to(+len-1))	Data field of message	(data)	(len)
+ len	Checksum of data field low byte	Csumlow	1
+ len +1	Checksum of data field high byte	Csumhigh	1

Table 7, Network Packet Protocol

This transmission protocol was implemented into the vehicle control software, as it was into all the subsystems connected to the network. The error rate of the system quoted by Houghton[6.1] was 1 error in 10^5 transmissions. Using this as the basis for data transmission the protocols between each of the sub-systems could be created.

The network had to link together several separate concurrently running systems. So each system was allocated a node address, to allow the transfer of data directly from one system to another. The nodes connected to the vehicle were given identification numbers related to their physical location. The systems mounted on the vehicle had addresses starting at 0, and the systems mounted on the surface were designated identification numbers starting at 128. The addresses of the systems connected to the vehicle are shown in Table 8.

Vehicle Systems		Surface Systems	
Node	System	Node	System
0		128	Top Station
1		129	ROV Position Display
2	Sonar System	130	Sonar Display
3	Camera Pan and Tilt	131	Pan & Tilt Display
4	Thruster System	132	Vehicle Control
5	Laser and Camera	133	Vision System
6	Sensor Box	134	Joystick System
7	Manipulator Arm	135	Arm Display

Table 8, Network Nodes

Using the above information it was possible for sub-systems to be able to communicate to one another over the network. For example when the control system wishes to alter the thruster values it sends a message from node 132 to node 4. The networking system then interprets these addresses and routes the information appropriately. A set of commands for each of the nodes was developed jointly between the network designer and the people responsible for each node, however only the ones pertinent to this project are shown. The full set of networking commands used on the vehicle is detailed in the MAST report [6.5]

6.2.2.1 Thruster Protocol

A set of commands was developed for communication with the thrusters. The commands developed for the thruster communication are shown in Table 9. This table is reproduced from the MAST report [6.5].

Command	Function	Data	Data Length	Reply	Reply Length
0	Request Status	-	1	0=ok	2
1	Set thruster 1 (0 to 4095)	0..4095	3	-	-
2	Set thruster 2 (0 to 4095)	0..4095	3	-	-
3	Set thruster 3 (0 to 4095)	0..4095	3	-	-
4	Set thruster 4 (0 to 4095)	0..4095	3	-	-
5	Set thruster 5 (0 to 4095)	0..4095	3	-	-
6	Set thruster 6 (0 to 4095)	0..4095	3	-	-
7	Read thruster 1 speed	-	1	Thruster Speed	3
8	Read thruster 2 speed	-	1	Thruster Speed	3
9	Read thruster 3 speed	-	1	Thruster Speed	3
10	Read thruster 4 speed	-	1	Thruster Speed	3
11	Read thruster 5 speed	-	1	Thruster Speed	3
12	Read thruster 6 speed	-	1	Thruster Speed	3
253	Read Inhibit Status	-	1	0=no inhibit	2
254	Set Inhibit Status	0 = OK, 1=inhibit	2	-	-
255	Emergency Stop	-	1	-	-

Table 9, Thruster Node Protocol

The protocol allows the thruster's speed to be set individually, and the set speeds to be read back from the system. In addition to these fundamental control commands, additional commands are available to provide extra protection for the thruster systems.

Request Status	checks that the networking system is functioning correctly, this value returns 0 for no error.
Set Inhibit Status	sets the inhibit on the vehicle. With the inhibit set the thrusters will not operate, despite the thruster demand value set. This is an extra measure of security to avoid accidental operation of the propellers, which could potentially cause damage or injury.
Read Inhibit Status	reads the current status of the inhibit flag.
Emergency Stop	resets all thruster demands to 2048 (0V) and activates the inhibit.

6.2.2.2 Sensor Protocol

A set of commands were devised to allow the reading of the sensor values from the sensor box. The reply to the information request is the value of the sensor, plus a time stamp indicating when the value was taken. This is because the request may take time to process, and for the control it is important that the sample time is known. For example the NMEA sensor readings may be as much as a second out of date, so the control system had to take this into account.

The commands sent to the sensor box are shown in Table 10. This table is reproduced from the MAST report [6.5].

Command	Function	Data	Data Length	Reply	Reply Length
1	Read Pressure	-	1	INT16+time	7
2	Read Compass	-	1	16bit+time	7
3	Read Speed	-	1	Real32+time	9
4	Read Tilt 1	-	1	16bit+time	7
5	Read Tilt 2	-	1	16bit+time	7
6	Read Rate Gyro	-	1	8bit+time	6
7	Read Acceleration	Channel	2	Chan, Real32 + time	10
8	Read Velocity	Channel	2	Chan, Real32 + time	10
9	Read Position	Channel	2	Chan, Real32 + time	10
10	Reset Accelerometer Integration	-	1	-	-
11	Read Temperature	-	1	byte + time	6
12	Read Ultrasonic Range	-	1	Real32+time	9
254	Read Status	-	1	0 = OK	2

Table 10, Sensor Protocol

6.2.2.3 Vehicle Control System Protocol

To be able to operate the vehicle, the pilot had to be able to control all the vehicle's technologies from a central control system. The commands issued by the pilot on this central control system were transmitted to the sub-systems to allow them to carry out the requests. The system was designed in this way to allow each system to be able to perform either individually, or as a slave under the control of the pilot. This allowed each subsystem to be developed using different technologies, and distributed processing capability for the system as a whole. Towards this end a set of commands that the vehicle control system could accept were developed. These commands are shown in Table 11, taken from the MAST report[6.5].

The protocol allows the supervisory computer to give high level commands such as *Set Depth* and *Set Heading* which when activated causes the control system to perform automatically. Activating these automatic control systems removes the necessity of the pilot to manually control the variables, thus simplifying the task for the pilot.

Command	Function	Data	Data Length	Reply	Reply Length
1	Move Absolute (XYZ)	3*Real 32	13	(see 129)	-
2	Move Relative (XYZ)	3*Real 32	13	(see 130)	-
6	Set depth	Real 32	5	-	-
7	Set heading	16 bit	3	-	-
8	Set tilt 1	Real 32 angle	5	-	-
9	Set tilt 2	Real 32 angle	5	-	-
10	Set XYZ velocities	3*Real 32	13	-	-
11	Set XYZ rotation speed	3*Real 32	13	-	-
12	Set auto depth mode	0=off, 1=on	2	-	-
13	Set auto heading mode	0=off, 1=on	2	-	-
14	Set Auto level mode	0=off, 1=on	2	-	-
129	Arrived at absolute XYZ	-	1	0 = arrived	2
130	Arrived at relative XYZ	-	1	0 = arrived	2
254	Read status	-	1	0 = OK	2
255	Emergency Stop	-	1	-	-

Table 11, Vehicle Control System Protocol

6.2.3 PC Communication Software

The communication software on the control PC had to be rewritten to implement the protocols previously detailed. The software created a shell within which the control algorithms could run, providing all sensor information and data output. The control system could then be altered without the need to rewrite any of the interfacing modules. A list of the procedures written for the networking system is given in Appendix 10.6

The communications software was separated into two main areas. These areas can be broadly divided into data output routines, and data input routines. These routines were written as a separate module, which could then be linked together with the control software.

The actual software written is lengthy and involved. However, a highly simplified outline of the software can be given.

6.2.3.1 Data Output Software

As the transputer network is a multi-tasking system the time delay between making a request for information and that request being fulfilled is variable, depending on the load on the system. To cater for this, networking routines had to be written to allow for a non-periodic sample time.

The basis of the output routines is a *Data Request* routine. Similar request routines were written for each set of data required. The structure of the *Data Request* routine is shown below.

```
Data Request (Sensor Required)
Is a request for this data outstanding
    If not then note the current time and send request to network.
    If Yes then check the time elapsed since the request
        If time is within limits then exit procedure
        If time is outside limits send failure report, and reset request
End
```

The software checks to see if an outstanding request is present to avoid multiple requests being made on the network without the data being received. This was necessary to be able to synchronize the data transmission rate with the computer's reception rate. This was done as problems were initially found with the difference between the two rates causing unresolved packets on the network. A time out was introduced to prevent stuck requests in the network. The request routines implemented the data protocol described in section 6.2.2. All output data from the machine had the source node defined as 132, and the designated node dependent on the request. Separate routines were written to construct the sentence

structure, including the header character, addresses, data and CRC information for each request type.

An interrupt driven routine was written to request updated sensor values for all the thruster and sensor variables. This routine was tied into the clock tick interrupt, which activates at 18.2Hz. This value allowed the sensors to be updated at a rate suitable for the control system, and at a rate that the computer could cope with sufficiently. The request rate in Hz for each sensor could be set, to cause automatic requests for updated sensor readings at the optimum rate. Making faster requests could saturate the network, and provide insufficient time for the control computer to process the incoming data and act upon the results. To determine the optimum transmission rates monitoring software was written to monitor the transmission and receive rate. The request rate and polling rate of the system were adjusted to give the optimum performance of the networking shell, without inducing networking errors.

In addition to the output routines necessary for the request of system data routines were also written to acknowledge the reception of incoming data from the supervisory control computer. These routines were automatically activated upon retrieval of the control information.

6.2.3.2 Data Input Software

As the reception of data does not trigger an interrupt it was necessary to poll the incoming data flag. The polling of this register must be done so as to provide a sufficient data rate for reception, whilst allowing other processes to continue. To do this the polling routine was tied onto the timer interrupt to provide periodic data reception, as well as being called after each cycle of the control software.

The basic structure of the receive routine is shown below.

```
Receive Routine
    Check if any data has come in
    If no data has arrived then goto EXIT
        If data has arrived read data until there is no more, or until time-out
        Store this incoming data into ring buffer.
        Parse data buffer to extract received data, making checks for CRC, destination
            address and Header data
        Reset Data Requested flags for received data
        Delete interpreted data from buffer.
EXIT.
```

The above pseudocode shows the basic operation of the reception software. As can be seen from the above the system attempts to read all incoming data, until no more is available. It therefor important that the traffic on the network is not too high, as all processor time would be spent in communication tasks alone. Limits were placed on the data transmission rates of the software, and on the data reception, using the transmission, reception and error rate routines added to the software. Limits on transmission rates were also introduced into the networking software, to avoid network saturation and hogging by one system.

The received data is CRC error checked and parsed. The data is then interpreted and stored in a temporary buffer where the updated data can be read at the end of a control cycle. This is to prevent the interrupt routine changing variables mid-way through the control loop. The commands from the supervisory computer require an acknowledgment signal. When these signals were received in addition to the interpretation of the incoming data flags were set to cause the transmission of an acknowledge signal.

A data display screen was created to allow the user to view the sensor information. A screenshot of simulated data input is shown in Figure 74. The display screen is separated into several different display areas. A main window shows a diagram of the X,Y, and Z rotations of the Aquasphere. The other smaller windows display data for the vehicle's depth, speed, temperature, heading and each thruster's power setting.

The graphs for depth, depth below vehicle, and ground profile show a scrolling graph of the data, providing a visual profile of the bottom of the water. This profile is calculated from the vehicle's current depth and the distance currently below the vehicle.

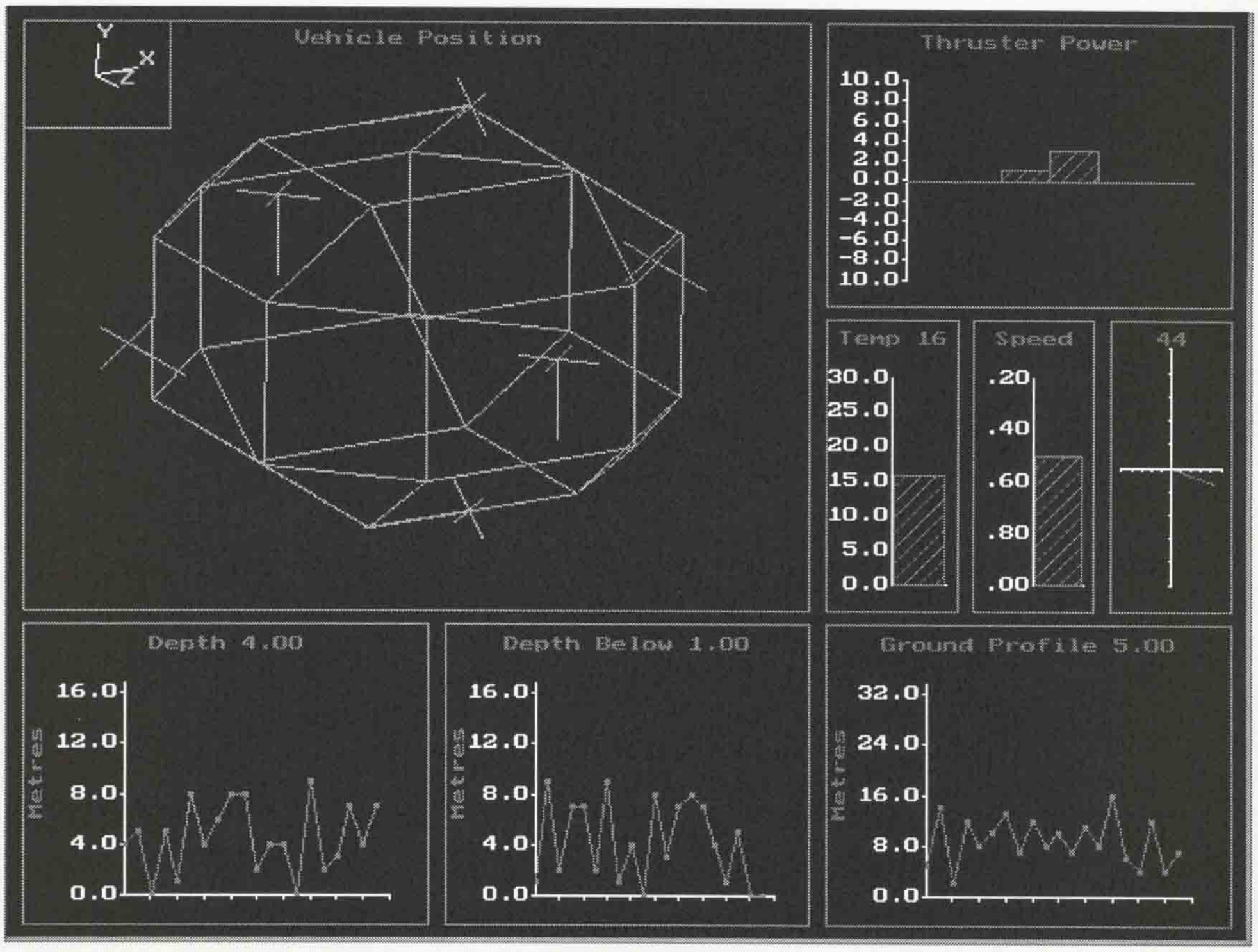

Figure 74, The Data Display System

This display system was removed from the final system, to reduce the load on the computer, and to simplify the overall software. In its place a simple text based system was implemented.

6.2.4 Hardware Interfacing

For the sensors and thrusters to be connected to the vehicle interface circuits had to be used. The network system created by Houghton[6.1] was based on a modular system. The modules consisted of processor cards, analogue input cards, analogue output cards, digital input/output cards, serial input/output cards and current input cards. By the connection of the sensors and thrusters to these devices the networked system could be created.

6.2.4.1 NMEA Sensor Interfacing

To interface the NMEA sensors to the network a standard RS232 serial data card was modified to receive the NMEA data as described in section 6.1.3.1 and the data reception software outlined in section 6.1.3.3 was rewritten to run on the transputer system. This software converted the NMEA data, and stored the data within the transputer network. Every time the sensor updated the information the new values were calculated and stored. Requests for the NMEA data were given information from this data cache, along with the time at which the results were taken.

6.2.4.2 Pressure Sensor Interfacing

The pressure sensor was connected to a current input card mounted on the networked system. The current through the sensor was returned in response to command 1, Read Pressure, sent to the sensor box. The range of the data transmitted from the pressure sensor is 4-20mA, being read at 12 bit resolution. The sample rate of a single channel is 50kHz, allowing an almost instantaneous reading to be taken.

6.2.4.3 Inclinometer Interfacing

The signals from the two inclinometers were connected to the input channels of an analogue to digital module. The interface has the same 12 bit resolution and 50kHz sample rate as the current input card. This allows the pan and tilt for the vehicle to be read almost instantly. The information from the inclinometers can be requested separately, using commands 4 and 5 sent to the sensor box.

6.2.4.4 Gyroscope Interfacing

The result from the gyroscope was used for testing purposes only. The device was connected to the ADC card, and returned a 8-bit value upon request. This information is obtained by sending command 6 to the sensor array.

6.2.4.5 Accelerometer Interfacing

The signal from the accelerometer was preamplified to increase the amplitude of the signal. This value was then digitized using the ADC card. The value from this sensor was constantly sampled by the networking system, and integrated to provide velocity and

distance. The integration could be reset using command 10 to the sensor system, whilst the current values could be read using commands 7, 8 and 9.

6.2.4.6 Thruster Interfacing

The six inputs to the PWM generation circuits mounted in the thruster drive box were connected to the outputs of six DAC channels. The channels have a 12bit resolution and a bipolar output of ±10volts. This corresponds directly to the ±10volts input that the PWM card requires. The 12bit card allows for a resolution of 1-in-2048 per direction. These levels are set when the thruster values are set using commands 1-6 sent to the thruster box, provided the inhibit is not enabled. The PWM cards have a hardware inhibit signal to turn off the drives. It was envisaged that an additional analogue or digital output could have been used to activate this inhibit. However, due to the lack of connections on the cabling, and a shortage of free output channels this connection was not made. However, the inhibit logic was implemented into the networking software.

7. Vehicle Control

The aim of the control system was to be able to control the Aquasphere for full six degrees of freedom. With a six degrees of freedom vehicle it is possible to attain full control of the vehicle's motion. The vehicle can be stabilised against external forces which cause translation and/or rotation. These external forces may be caused by water current flow or by wave action against the vehicle. Thus this configuration provides the possibility of having a completely stable and manoeuvrable platform.

A study by the Department of Energy [7.1] tested several commercial vehicle systems in a set of trials in Loch Linnhe, Scotland. The trials highlighted the problems encountered when carrying out typical subsea tasks. The task of station holding was given particular note, as this task was found to take up to 35% of the total time. The limitations of the vehicles to maintain station holding from the report [7.1] were given as:-

(i) "The inability of the vehicle to counter the side acting currents at the site."
(ii) "The pilot workload needed to maintain the vehicle on station, in addition to the demands of the task"
(iii) "The inability of the pilot to hold a position in the horizontal plane within the required limits.

The report concluded that "the ultimate improvement would be an automatic positioning system that would leave him (the pilot) free to concentrate on the actual task. It was noted that when auto-heading and auto-depth systems were used the overall performance was improved as a result. An auto-position system requires a more advanced form of control but if this can be developed, the facility would be a major step forward in ROV technology"

The development of the control system was done so that full positional control could be carried out given the appropriate sensor information. As the available sensor information was limited, the control system was restricted to providing auto-depth, auto-heading and auto-level control.

For control of any system it is desirable to know the equations for the performance of the plant. For a six degrees of freedom vehicle the equations governing the motion of the vehicle are complex and difficult to derive empirically. The complexities of modelling such

a structure are so great, and the typical results from the simulation of such a system so diverse that in practice the only way to discover the dynamics of the system is through physical testing of the vehicle, or testing of a scale model of the vehicle. Goheen and Jefferys[7.2] emphasise this in saying that "dynamic models for remotely operated underwater vehicles (ROVs) are essential for the creation of autopilots and pilot training simulators. However, the derivation of these models is difficult because ROV dynamics are strongly coupled, highly nonlinear and ill-defined. In addition , expensive, specialised testing equipment is required for conventional modelling". The conventional method for the modelling of vehicles is to use a Planar Motion Mechanism (PMM) attached to a linear tank. The vehicle is oscillated in one degree of freedom and the resultant equations for that particular motion can be derived. This work is very time consuming and expensive, using equipment that is not available here at the University of Liverpool. Alternative methods for testing the vehicle and its control systems had to be used.

The original design of the Aquasphere was done to minimize the cross axis coupling found in conventional designs, and to simplify the equations of motion by presenting symmetry to each axis. For a commercial ROV, the SeaPup, the equations of motion derived from testing showed that a total of ninety-two derivatives were necessary for a model of the vehicle in six degrees of freedom. By introducing symmetry the number of derivatives could be greatly reduced. In practice this was not possible as the vehicle evolved from the original spherical shaped design into a non-symmetrical one in its final version.

7.1 Equations of Motion

The equations of motion for submarines and aircraft are widely available, but most texts assume that the vehicle has the classical design, typically a long cylindrical body. This classical design gives equations that are biased greatly towards the systems needs, and does not give rise to a fully controllable system such as the Aquasphere. For example, in a submarine it is not necessary to rotate the vehicle about the longitudinal axis, or to translate the vehicle sideways. Therefore a more generic type of equation had to be used to suit our vehicle.

There is a great deal of published literature regarding the dynamics of *flying vehicles.* A flying vehicle is defined as one which is not mechanically linked to the media in which it operates. However, the bulk of this literature concentrates on the aerodynamic configurations, as more effort has been devoted to aircraft design than to maritime problems. Hydrodynamic and hydrostatic forces, in addition to rotational moments, set submarine and remotely operated

vehicle operation apart from aircraft. However, it is possible to apply fundamental laws of mechanics in order to determine the equations of a body in three dimensional space.

The basic laws applied are:-

- Newton's 2nd law of motion.(*Applied force is proportional to the rate of change of momentum;* **F =ma**)
- Euler's equations for angular movement.

7.1.1 Euler's Theorem

This theorem states that: "Two component rotations about different axes passing through a point are equivalent to a single rotation about an axis passing through the point. If more than two rotations are applied, they can be combined into pairs, and each pair can be further reduced to combine into one rotation" [7.3].

When a body is subject to rotations it is important to take into account the order in which they are applied. This is because "finite rotations do not obey the law of vector addition (i.e. $\theta_1 + \theta_2 \neq \theta_2 + \theta_1$), and hence they cannot be classified as vector quantities" [7.3]. For example, consider two finite rotations $\theta_1 + \theta_2$ applied to an object. Each rotation has a magnitude of 90° and a direction defined by the right-hand rule, as indicated by the arrow. The resultant of the operation is shown on the right.

Figure 75 shows the positions of the block when rotations are applied in the order $\theta_1 + \theta_2$

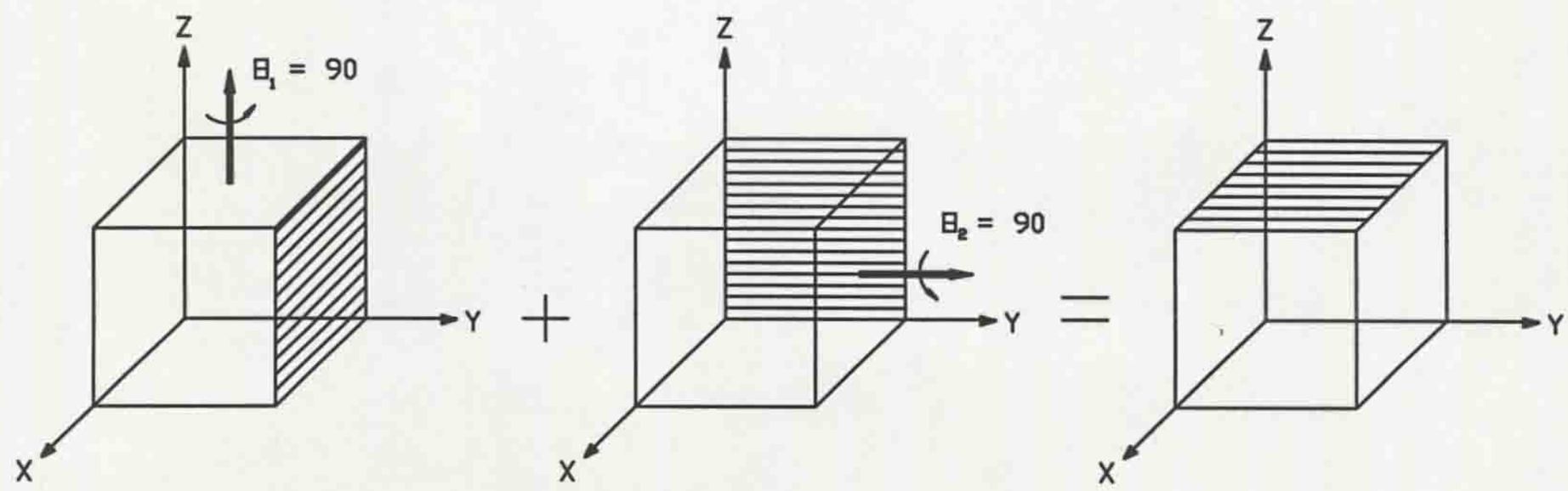

Figure 75, Rotation in order $\theta_1 + \theta_2$

When these two rotations are applied in the order $\theta_2 + \theta_1$ the resultant orientation is not the same as it was before.

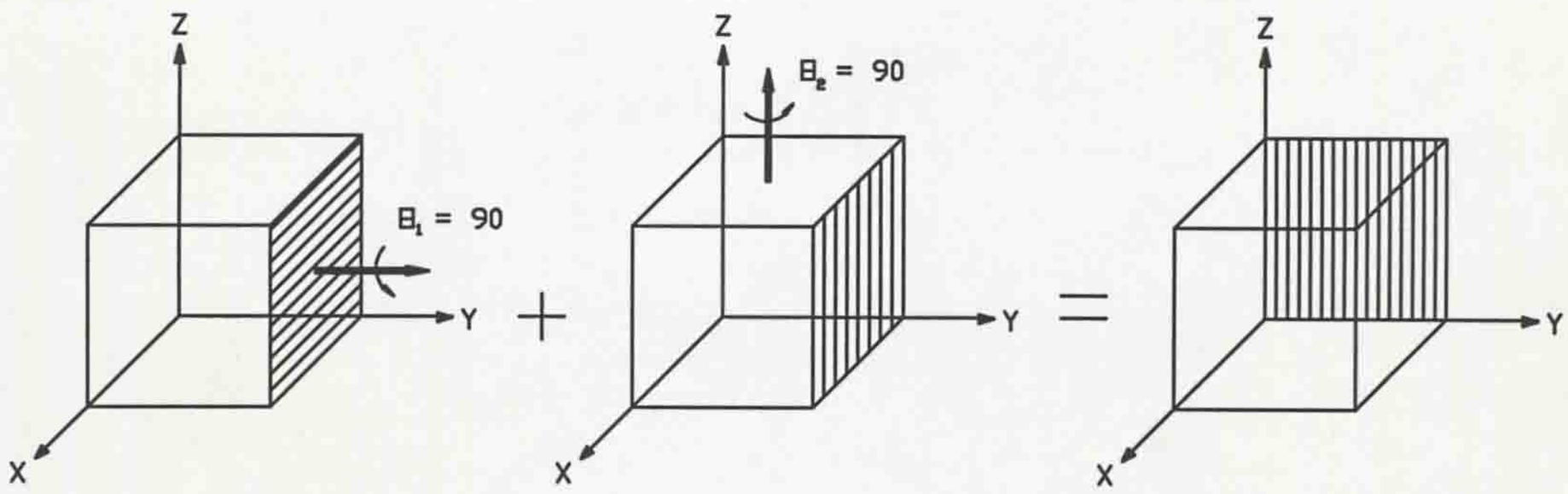

Figure 76, Rotation $\theta_2 + \theta_1$

Due to the result of the rotations being non-commutative a standard Cartesian reference frame and vector notation cannot be used for a freely rotating body such as the Aquasphere. Instead, Euler notation and equations must be used for defining the rotations of a body.

The assignment of the axes and their rotations can be seen in **Figure 1**. The six degrees of freedom are three translations X, Y, Z and the three rotations Yaw = (Ψ), Pitch = (θ) and Roll = (ϕ). These Cartesian variables (X, Y, Z, ϕ, θ, Ψ) can be mapped into the corresponding Euler angles (u, v, w, p, q, r) by the relation:-

$$\dot{\Psi} = \left(\frac{q \sin \phi + r \cos \phi}{\cos\theta} \right)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \qquad \textbf{Equation 24}$$

$$\dot{\phi} = P + \dot{\Psi} \sin\theta = p + (q \sin \phi + r \cos \phi)\tan\theta$$

7.1.2 ROV Equations of Motion

From the available information it was decided that the SNAME equations of motion suggested for ROV use by Etkin [7.4] , Yoerger and Slotine [7.5] *et al* should be used to attempt to steer the vehicle in the three directions *X*, *Y* and *Z*, and to rotate the vehicle about each axis. The equations are suitable for a rigid body (i.e. one which does not flex or deform) with six degrees of freedom

The equations of motion are divided into four categories:-

7.1.2.1 Translational equations of motion

The linear forces and accelerations for the vehicles can be derived for each axis from the equation F = ma as:-

$$F_X = m(\dot{u} + qw - rv)$$ **Equation 25**

$$F_Y = m(\dot{v} + ru - pw)$$ **Equation 26**

$$F_Z = m(\dot{w} + pv - qu)$$ **Equation 27**

7.1.2.2 Angular equations of motion

The angular equations of motion for each axis can be shown as:-

$$T_x = I_x \dot{p} - I_{xz}(\dot{r} + pq) - (I_x - I_z)qr$$ **Equation 28**

$$T_y = I_y \dot{q} - I_{zx}(r^2 + p^2) - (I_z - I_x)rq$$ **Equation 29**

$$T_z = I_z \dot{r} - I_{zx}(\dot{p} - qr) - (I_x - I_z)pq$$ **Equation 30**

7.1.2.3 Attitude equations:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(q) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(q) & \cos(\phi)\sec(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$ **Equation 31**

7.1.2.4 Position equations

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} A\,B\,C \\ D\,E\,F \\ G\,H\,I \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Equation 32

where

$$A = \cos(\theta)\cos(\psi)$$

$$B = \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi)$$

$$C = \cos(\phi)\sin(\theta)\cos(\psi)$$

$$D = \cos(\theta)\sin(\psi)$$

$$E = \sin(\phi)\sin(\theta)\cos(\psi) + \cos(\phi)\sin(\psi)$$

$$F = \cos(\phi)\sin(\theta)\sin(\psi)$$

$$G = \sin(\theta)$$

$$H = \sin(\phi)\cos(\theta)$$

$$I = \cos(\phi)\cos(\theta)$$

7.1.2.5 Vector Notation

Using the equations shown in section 7.1.2.1 to section 7.1.2.4, the complete motion of the vehicle can be calculated. An alternative approach to the equations of motion for the vehicle is by taking the sum of all the acting vectors, as shown in Equation 33. This approach is used as the factors in the equation are more readily derived by experimentation and mathematical modelling than the standard SNAME equations. Of these values hydrodynamic and hydrostatic forces for the vehicle can be measured experimentally given the appropriate test facilities, while the mass matrix and umbilical forces can be calculated mathematically. All other values can be determined from sensor readings..

$$[\hat{M}][\dot{Q}] = F_d\,(u,v,w,p,q,r) + \hat{H}\,(u_r,v_r,w_r,p_r,q_r,r_r) + G\,(\phi,\theta,\psi) + T + U$$

Equation 33

where

$\hat{M}$	=	Mass matrix including hydrodynamic added mass.
$\dot{Q}$	=	$(u, v, w, p, q, r)^T$, the velocity vector.
F_d	=	Vector of rigid body kinematic forces.
$\hat{H}$	=	Vector of hydrodynamic forces.
G	=	Vector of hydrostatic forces.
T	=	Vector of forces due to thruster.
U	=	Vector of forces due to umbilical cable.
r	=	Relative motion between vehicle and water.

7.1.3 Derived Equations

As has been mentioned it is possible to derive the equations of motion for a vehicle from a combination of mathematical analysis and experimental data.. The experimental data requires that the absolute position and orientation of the vehicle are known throughout the tests. These tests are normally carried out by moving the vehicle through the fluid attached to force measurement equipment. Such equipment was not available, and the timescale of the project was such that the final vehicle configuration was only determined near the end of the project. Goheen and Jefferys[7.6] state "Following trials considerable effort is needed to derive a model from experimental results, leading to a prohibitively expensive model which will be inaccurate as soon as different auxiliary equipment is added to the ROV". In this case the configuration of the vehicle changed throughout the development of the project, and during testing the configuration of the vehicle changed frequently, as sub-systems were installed and removed.

It is theoretically possible to derive a model of the vehicle from the logging of sensor information and demand signals. Using System Identification (SI) techniques and Parameter Estimation (PE) techniques combined with data taken from the free running trails of an ROV a model of the vehicle can be formulated. The creation of this model requires accurate sensor information, preferably over extended trials of the vehicle.

For the Aquasphere the quality and availability of sensor information meant that this approach could not be used. There was insufficient data available to be able to model the vehicle, as two of the six parameters needed for the model were absent (X and Y position). Also the changing form of the vehicle meant that very little data could be logged with the vehicle in its final configuration. The size of the vehicle meant that testing was not possible in

the recirculating flume located at the university, and external testing facilities would have been prohibitively expensive. Due to these considerations it was not possible to derive a model from experimental or test information.

7.2 Vehicle Control

Without the aid of a model of the vehicle off-line testing for control systems could not be carried out, allowing the optimum control solution derived. To attempt to overcome the lack of experimental test data, and of a plant model, a robust control system was necessary. Farbrother[7.7] *et al* says for ROV control that "Classical PID control has been used in the past, however such design relies upon an accurate linear model of the plant. If the plant is highly non-linear then a series of such controllers must be designed to cover the operating envelope." Where a model of the plant is not known a robust or adaptive control technique is much more desirable. Non conventional control techniques such as fuzzy logic can also deal with non-linearities directly.

With the recent popularity of novel unconventional control techniques such as fuzzy logic, neural networks, genetic algorithms and expert systems much work has been done into the development and implementation of these control techniques. Often these control methods have been developed without a real world application for them. The control of an underwater vehicle is a highly non-linear task, with changing parameters and a high degree of cross coupling. Therefore this application has provided an ideal experimental test bed for these novel technologies, and has been the subject of a great deal of experimentation for them.

The previous development of the Aquacube by Swainston[7.8] provided a test bed for control system verification. The compact size and the simplicity of the vehicle enabled it to be easily transported and quickly made operational. This allowed a significant amount testing time for the control of the vehicle to be carried out, providing the ability to test several different control strategies. An in depth investigation of the controllers tested on the Aquacube is given by Stephens[7.9], Stephens and Shimmin[7.10], Shimmin, Stephens and Swainston[7.11], and Kodogiannis[7. 12]. Shimmin and Lucas[7. 13] summarised the suitability of the control systems used on the Aquacube by saying that the vehicle has "been tested with conventional, SEMRAS, VSS, fuzzy and neural networks schemes. To date the fuzzy and neural net approaches have been significantly more accurate and robust than the other techniques." The studies carried out on the Aquacube lead to the development of a fuzzy PID controller for the Aquasphere.

7.2.1 Standard PID Controller

The block diagram of a conventional P.I.D. controller as part of a control system is shown in Figure 77

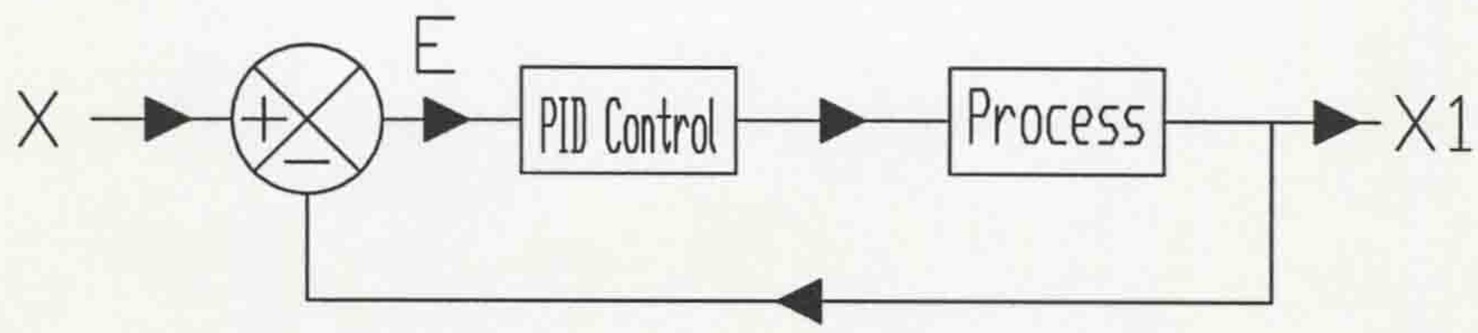

Figure 77, Conventional PID Control

The variable **X** is the demand value, **X1** is the measured output value, and **E** is the error between these two. It is the job of the process controller to control the process so that the error between the required and actual values is kept to a minimum. In our case the process is the response of the Aquasphere to external stimulation.

The equation for a PID controller is shown below.

$$Error(E) = X - X1$$

$$PID\ Controller = T_p\left(1 + sT_d + \frac{1}{sT_i}\right)$$

Where **Equation 34**

Tp = Proportional Gain
Ti = Integral Gain
Td = Derivative Gain

The values for the gains can be calculated by a combination of experimental test results and tuning methods such as Ziegler-Nichols tuning, or can be derived by experimentation. The Ziegler-Nichols method is based upon introducing oscillation into a closed loop proportional controller and extracting the gains from this data. In tests on the vehicles it was difficult to determine the point of oscillation of the system, causing the Ziegler-Nichols values to be incorrect. To determine the optimum control values a large amount of testing needed to be carried out. Although this was possible for the Aquacube such testing time was not available

for the Aquasphere, creating one of the primary reasons for the development of an alternative control strategy.

Despite the requirements for tuning of the PID controller manually the results from the system compared favourably with the test results taken obtained by Stephens[7.9] and Kodogiannis[7.12].

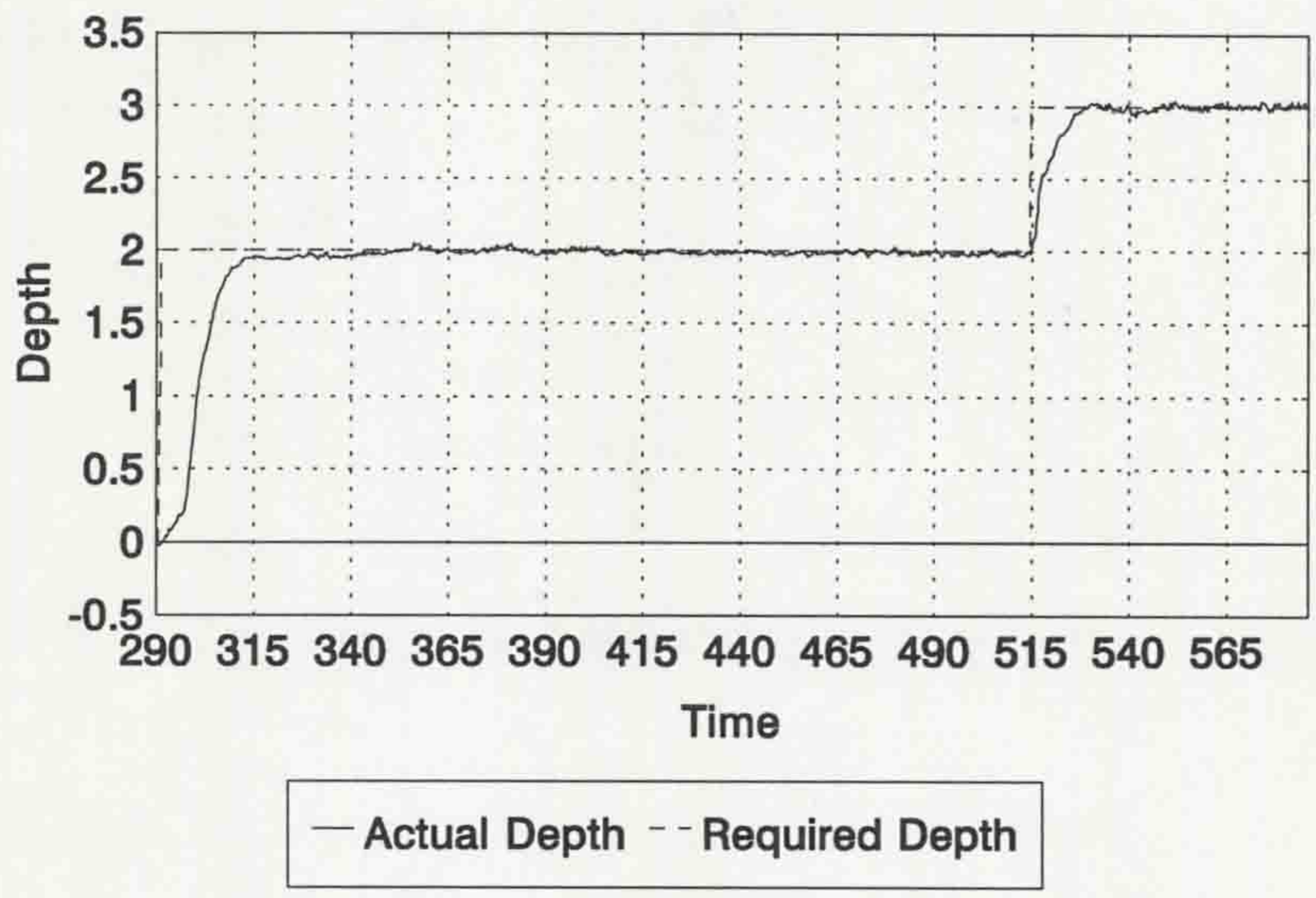

Figure 78, Aquacube test result

Figure 78 shows a sample test result for the Aquacube. The graph shows the vehicles response to a step input in depth demand. For this result the gains for the PID controller were Ti= 0.16, Td = 0.2, Tp = 1.75. As can be seen the tuned PID controller is capable of accurate control of the system. The steady-state accuracy of such a system is high compared to other methods due to its integral term. High steady-state accuracy was necessary as the principle task of the vehicle was to "hover" at specified depths. This PID system has shown the feasibility of control, and has provided a reference result for other control methods to be compared to. The need for tuning has lead to modification of the PID controller. Astrom's *et al* paper[7.14] highlights the necessity of modifications to the PID controller, and proposes methods for doing this. For our application the modifications applied to the controller were to add a fuzzy logic rulebased controller.

7.2.2 Fuzzy PID Controller

For the Aquasphere an alternative approach was used to that initially taken on the Aquacube. The success of the unconventional control techniques used on the Aquacube allowed such a control system to be fitted to the Aquasphere. From the control techniques previously applied it was decided that a fuzzy PID controller should be used for the control of the Aquasphere.

The fuzzy PID controller was chosen as it combines the proven PID control strategy with the non-linear control of fuzzy logic. The control of an ROV is inherently a non-linear problem. The principle reasons behind the non-linearity of the plant are:-

- The drag of the vehicle is a non-linear function of its velocity due to the turbulent flow around a bluff body.
- The thrust of the thruster is non-linear compared to the applied voltage.
- The buoyancy of the vehicle is non-linear. The buoyancy can become compressed at depth, giving lower buoyancy at greater depth.

The additional complexity of the control of the Aquasphere meant that one single controller was not used, but instead four separate control routines were used. The separate controllers used in the control system were:-

- Depth Controller
- Heading Controller
- Pitch Controller
- Roll Controller

Originally a scheme using six controllers was envisaged, but due to the lack of data for the position of the vehicle in the Z plane the controllers to control these variables was omitted from the final system. The modular design of the system would allow the introduction of these controllers, should the sensor data be improved.

Each of the above controllers can be switched on and off during the control systems operation. The sum of the control information from each of the systems is taken and output to the plant. A diagram of the implemented control system is shown in Figure 79

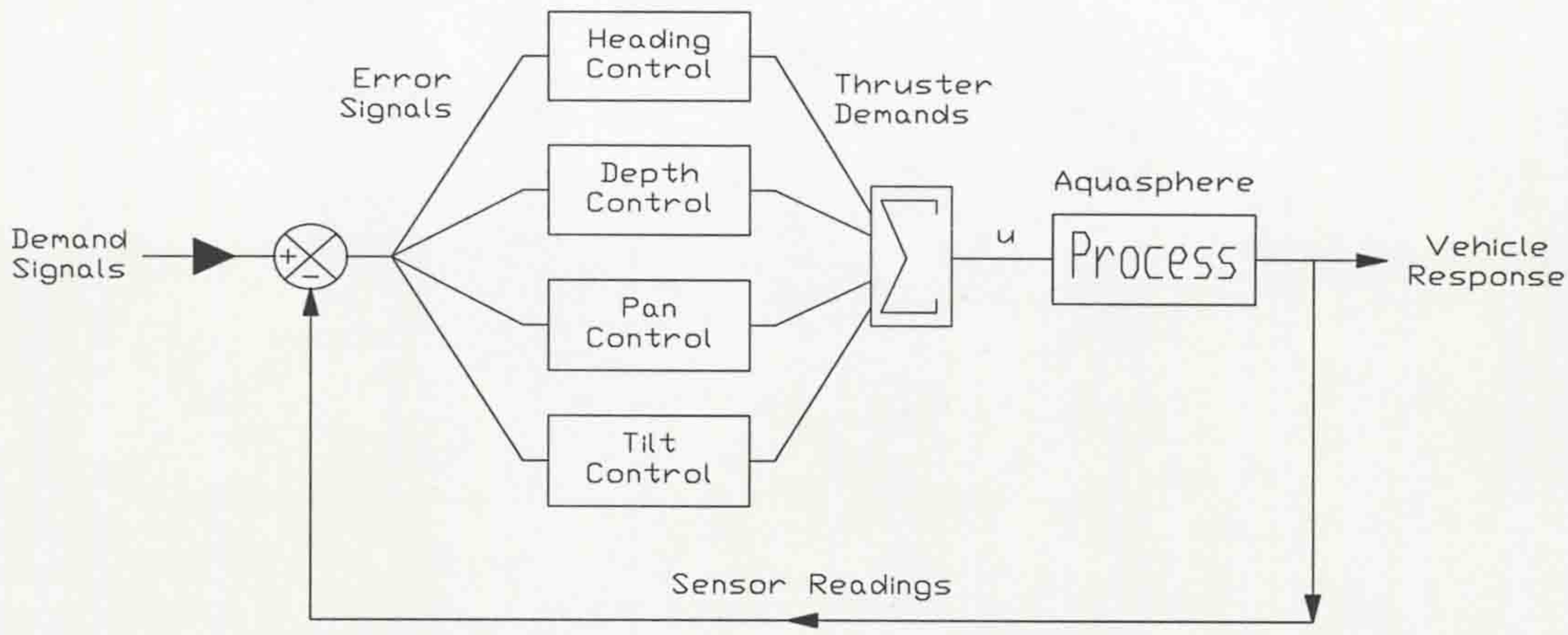

Figure 79, Aquasphere Control Strategy

Each automatic process can be activated and deactivated whilst the program is running. Each process receives as its input the sensor information received from the network, and the demand information either entered at the control console, or received remotely from the master computer. Each process makes adjustments to the control signals of all six thrusters. These demands are then added to give the final control signal output, which is converted from the Euler coordinates to the Cartesian coordinate system. This is done as the thrusters are mounted in a Cartesian axis frame. When the vehicle is rotated a different combination of the thrusters is used from the standard configuration. In the extreme case of 90° rotation an entirely different set of thrusters is used. For example, with the vehicle in the standard upright position thrusters 1&2 are used. With the vehicle tilted activating these thrusters will cause a horizontal component to be introduced. A compensating vector must be added to prevent the unwanted translation. By converting the data from Euler to Cartesian this thrust vectoring is achieved.

The automatic processes can be used in conjunction with manual control. The operators can select which automatic processes they require, and this control function is taken away from the pilots, easing the load upon them. For example the depth of the vehicle can be manually controlled by altering the vertical power vector of the vehicle. When the desired depth is achieved the pilot can request auto-depth mode, and the vehicle will attempt to maintain this parameter. Subsequent requests for vertical motion are from then on interpreted as required depth settings, allowing the operator to request other demand depths.

Similar strategies are used for automatic pitch and roll control, and automatic heading. The pilot can at all times override the automatic settings, providing an extra measure of safety.

7.2.2.1 Fuzzy PID design

The fuzzy controller used on the Aquasphere fuzzified the input variables e, de/dt and $\int e$, where e is the parameter error, into five fuzzy sets. The linguistic labels for these sets and the abbreviation to them is shown below.

LN	= Large Negative
SN	= Small Negative
Z	= Zero
SP	= Small Positive
LP	= Large Positive

The membership functions for fuzzification are shown below in Figure 80. A simple linear function was used for the boundary conditions.

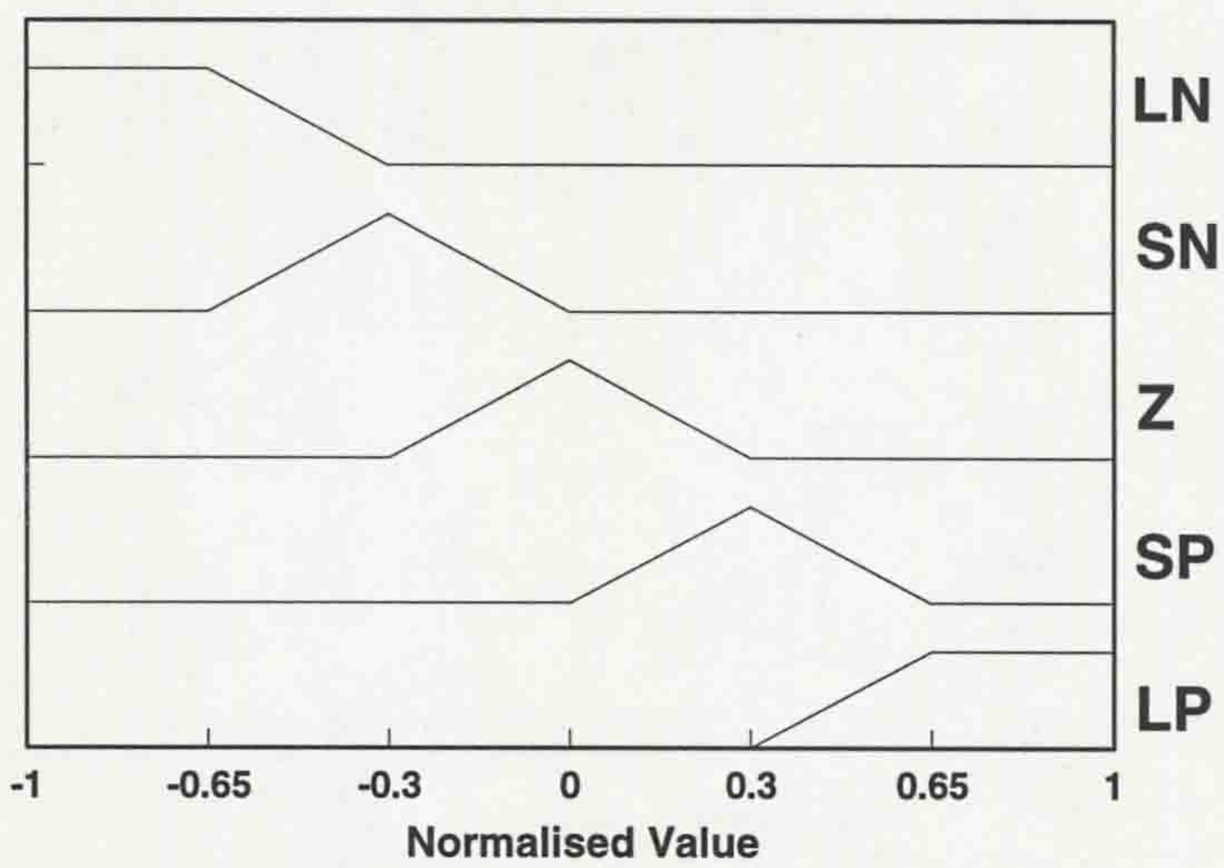

Figure 80, Membership functions for fuzzification

The control algorithms used were based on a central computational procedure. To fuzzify and defuzzify the information from each of the different control routines all input data was scaled to unity, plus or minus one. Upon defuzzification the returned controller output was rescaled to give the correct output. In this way only one procedure had to be written to

perform the fuzzy computation on all the various input data. The weights for the defuzzification of the output are shown in Figure 81.

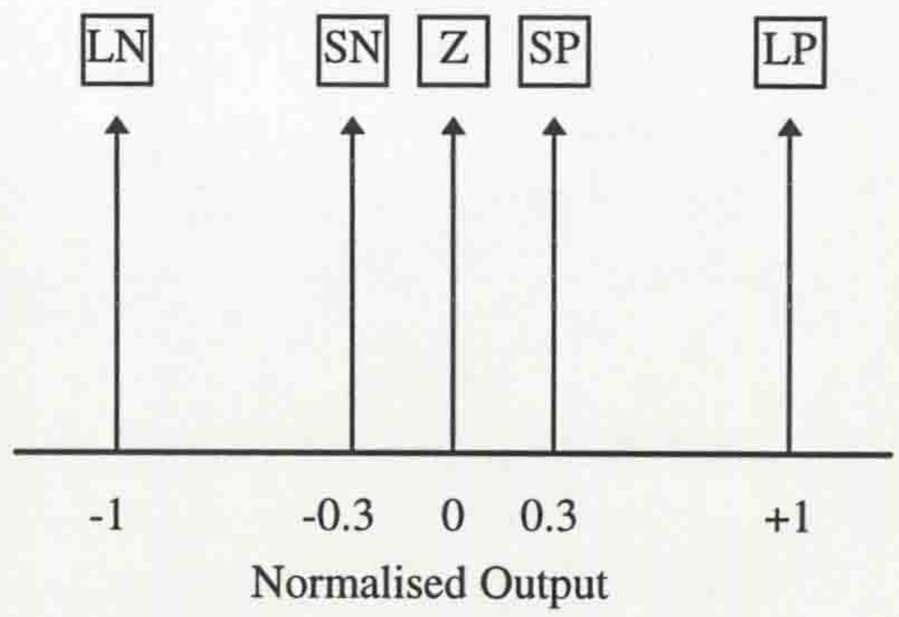

Figure 81, Weights for controller output defuzzification

With the three input variables a 5 by 5 by 5 fuzzy rulebase was used. Below in Table 12 to Table 16 are the fuzzy rulebases which form the 5x5x5 rulebase cube.

Er dEr/dt	LN	SN	Z	SP	LP
LN	LP	LP	LP	SP	Z
SN	LP	LP	LP	Z	Z
Z	SP	SP	Z	SN	SN
SP	Z	Z	SN	SN	SN
LP	Z	SN	SN	LN	LN

Table 12, 5 by 5 fuzzy rulebase ∫e =LN

Er dEr/dt	LN	SN	Z	SP	LP
LN	LP	LP	LP	SP	Z
SN	LP	LP	LP	Z	Z
Z	SP	SP	Z	SN	SN
SP	Z	Z	SN	SN	SN
LP	Z	SN	SN	LN	LN

Table 13, 5 by 5 fuzzy rulebase ∫e =SN

Er dEr/dt	LN	SN	Z	SP	LP
LN	LP	LP	SP	SP	Z
SN	SP	SP	SP	Z	Z
Z	SP	SP	Z	SN	SN
SP	Z	Z	SN	SN	SN
LP	Z	SN	SN	LN	LN

Table 14, 5 by 5 fuzzy rulebase ∫e =Z

Er dEr/dt	LN	SN	Z	SP	LP
LN	LP	LP	SP	SP	Z
SN	SP	SP	SP	Z	Z
Z	SP	SP	Z	SN	SN
SP	Z	Z	LN	LN	LN
LP	Z	SN	LN	LN	LN

Table 15, 5 by 5 fuzzy rulebase ∫e =SP

Er dEr/dt	LN	SN	Z	SP	LP
LN	LP	LP	SP	SP	Z
SN	SP	SP	SP	Z	Z
Z	SP	SP	Z	SN	SN
SP	Z	Z	LN	LN	LN
LP	Z	SN	LN	LN	LN

Table 16, 5 by 5 fuzzy rulebase ∫e =LP

Event	Rule Modification Value
LN	**-1.0**
SN	**-0.3**
Z	**0.0**
SP	**0.3**
LP	**1.0**

Table 17, Rule modification descision table

Table 17, shows the rule modification table applied to the fuzzy rulebase.

The fuzzy control scheme used on the Aquasphere combined Zadeh[7.15] and Yager [7.16] logic operators. The Zadeh and Yager logic operators used are shown below.

Zadeh:

$$\mu_A - (X) = 1 - \mu(X)$$
$$\mu_{A \cup B}(X) = \max\left[\mu_A(X), \mu_B(X)\right]$$
$$\mu_{A \cap B}(X) = \min\left[\mu_A(X), \mu_B(X)\right]$$

Equation 35

Soft Yager:

$$\mu_{A \cup B}(X) = \min\left[1, \mu_A(X) + \mu_B(X)\right]$$
$$\mu_{A \cap B}(X) = 1 - \min\left[1, 2 - (\mu_A(X) + \mu_B(X))\right]$$

Equation 36

Hard Yager:

$$\mu_{A \cap B} = 1 - \min\left[1, 2 - (\mu_A(X) + \mu_B(X))\right]$$

Equation 37

For the control system used on the Aquasphere the Zadeh *min* operator was used for fuzzy intersection (AND), while a soft Yager operation was used for fuzzy union. The combination of these operators was found by Shimmin *et al* [7.11] to provide a smoother and more accurate control than by using the Zadeh *max* or the hard Yager intersection shown above. A discussion of the merits of using the various operators available is given by Klir and Folger[7.16].

The controller output for the fuzzy control can be expressed as:-

$$\left(\overbrace{\frac{u}{K_u}}^{Fuzzified}\right) = \cup\left[\left(\overbrace{K_p.e}^{Fuzzified}\right)\cap\left(\overbrace{K_i.\int e.dt}^{Fuzzified}\right)\cap\left(\overbrace{K_d.\frac{de}{dt}}^{Fuzzified}\right)\right] \qquad \text{Equation 38}$$

Where

K_u	=	Output Gain
K_p	=	Proportional Gain
K_i	=	Integral Gain
K_d	=	Derivative Gain
'$\cap$'	=	Zadeh AND
'$\cup$'	=	Yager soft OR
e	=	Error variable

7.3 Aquasphere Control Results

Using the fuzzy control system described above testing was carried out on the Aquasphere. The control system required the tuning of four control loops to achieve the closed loop control desired. The testing of the Aquasphere was a complicated, expensive and time consuming operation, so a great deal of time was not available for tuning. In addition, in the final trials components of the vehicle were mounted and dismounted during the trials for their own testing, which alters the dynamics of the plant. However, in the limited time available the control variables were adjusted to give the results presented here.

7.3.1 Control Variable Input/Output

To adjust the many control variables a configuration file was written which contained the control parameters for the vehicle. Due to the very different dynamics for each of the controlled variables the values for the gains of each system were kept separate. To avoid re-entering all the input data for each test run of the control of the Aquasphere, the control program first read in the system gains and the rulebase values from the configuration file.

All the data from the vehicle's sensor systems and the control systems demand were logged. This provided the opportunity to analyze the data off-line, and for future analytical use. The variables for the vehicle's control systems were also logged to this file. A sample

of the output data file is shown below. The file contains header information about the test, and contains a copy of the control data given to the fuzzy control system.

```
File Created At Fri Oct 29 11:06:53
Data Generated By Program SPHERE2.EXE
Depth Using FUZZY Controller.
Heading Using FUZZY Controller.
Pitch Using FUZZY Controller.
Roll Using FUZZY Controller.

AutoDepth Settings
KP =    +1.000000   KD =    +3.000000   KI =    +0.100000   KU =   +20.000000

AutoHeading Settings
KP =    +0.200000   KD =    +3.000000   KI =    +0.003000   KU =    +4.000000

AutoPitch Settings
KP =    +0.150000   KD =    +1.000000   KI =    +0.010000   KU =   +20.000000

AutoRoll Settings
KP =    +0.150000   KD =    +1.000000   KI =    +0.010000   KU =   +20.000000

Rules
-2 -2 -2 -1 +0
-2 -2 -2 +0 +0
-1 -1 +0 +1 +1
+0 +0 +1 +1 +1
+0 +1 +1 +2 +2

-2 -2 -2 -1 +0
-2 -2 -2 +0 +0
-1 -1 +0 +1 +1
+0 +0 +1 +1 +1
+0 +1 +1 +2 +2

-2 -2 -1 -1 +0
-1 -1 -1 +0 +0
-1 -1 +0 +1 +1
+0 +0 +1 +1 +1
+0 +1 +1 +2 +2

-2 -2 -1 -1 +0
-1 -1 -1 +0 +0
-1 -1 +0 +1 +1
+0 +0 +2 +2 +2
+0 +1 +2 +2 +2

-2 -2 -1 -1 +0
-1 -1 -1 +0 +0
-1 -1 +0 +1 +1
+0 +0 +2 +2 +2
+0 +1 +2 +2 +2
```

The data file also included a log of the measured variables `Time`, `Depth`, `Speed`, `Heading`, `Range`, `Tilt1`, `Tilt2`, `Temp`, `Demand_Depth`, `Demand_Heading`, `Demand_Pitch`, `Thruster_1`, `Thruster_2`, `Thruster_3`, `Thruster_4`, `Thruster_5` and `Thruster_6` power.

From analyzing the data produced from this output file the performance of the system can be studied. The graphical data for the control of the vehicle is taken from these system log files.

7.3.2 Aquasphere Depth Control

A sample control result for the Aquasphere to a step input in the demand depth is shown in Figure 82. The corresponding output of the control system to the thrusters can be seen in Figure 83.

The rulebase given in Table 12 to Table 16 was used for this test, in addition to gains of :-

$K_p = 1.0$

$K_d = 3.0$

$K_i = 0.1$

$K_u = 20.0$

The response of the vehicle is sluggish, and there is a small amount of overshoot and oscillation. Further tuning of the system could remove these problems. However, the steady state response of the system is excellent, maintaining the depth of the vehicle with almost no drift.

The corresponding demand value to the thruster shows the non-linearity of the control, with an offset necessary to counter the effects of positive buoyancy.

Due to the large mass of this vehicle, the power to mass ratio of this vehicle is poor, with only approximately 0.5N thrust per kilogram. Commercial vehicles normally have a figure in excess of 3N per kilogram. This lack of power explains the sluggish response of the vehicle to system demands, and indicates that accurate tuning of the vehicle is essential for any form of control.

Aquasphere Depth Control

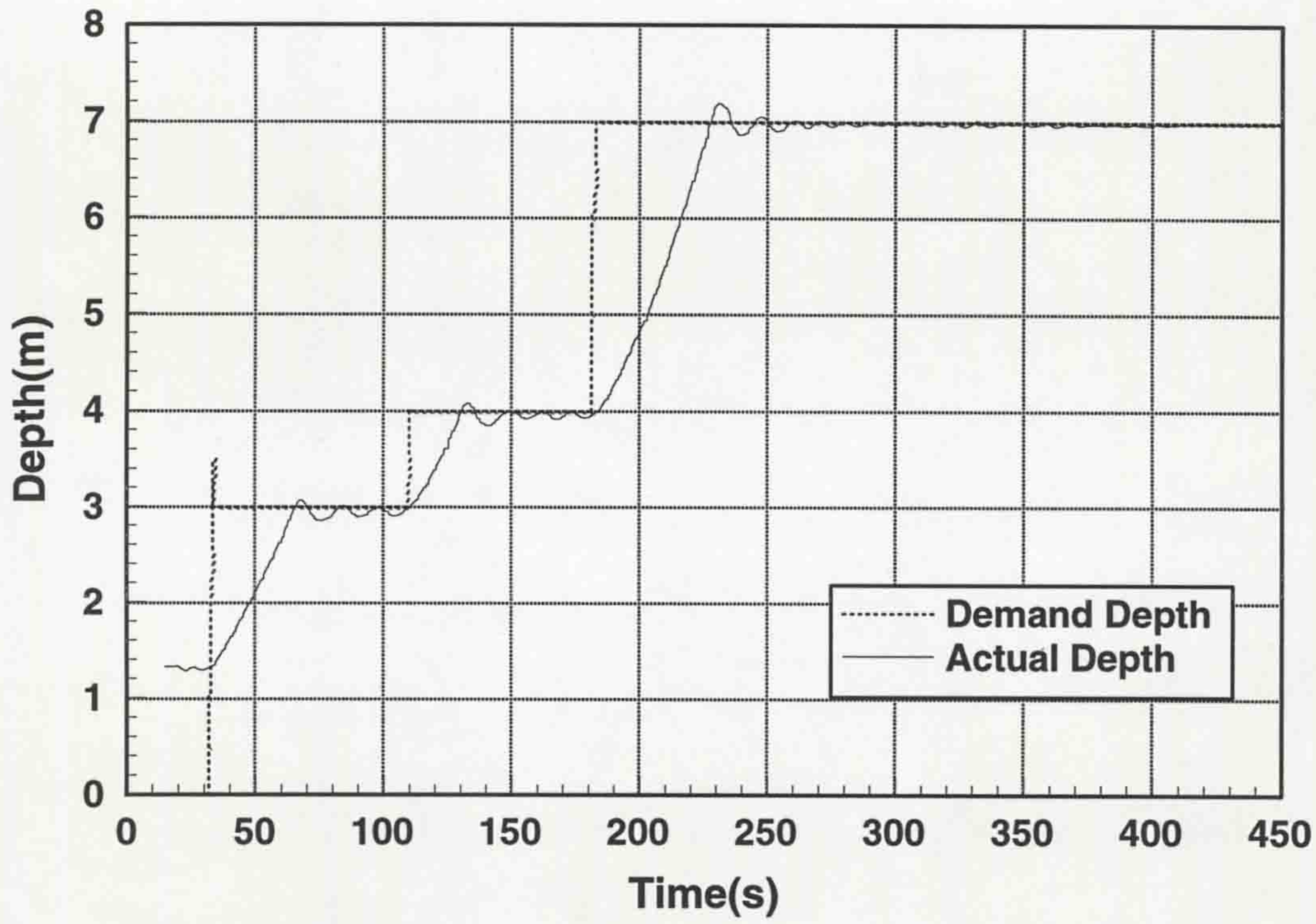

Figure 82, Aquasphere's Depth Response

Corresponding Thruster Demand

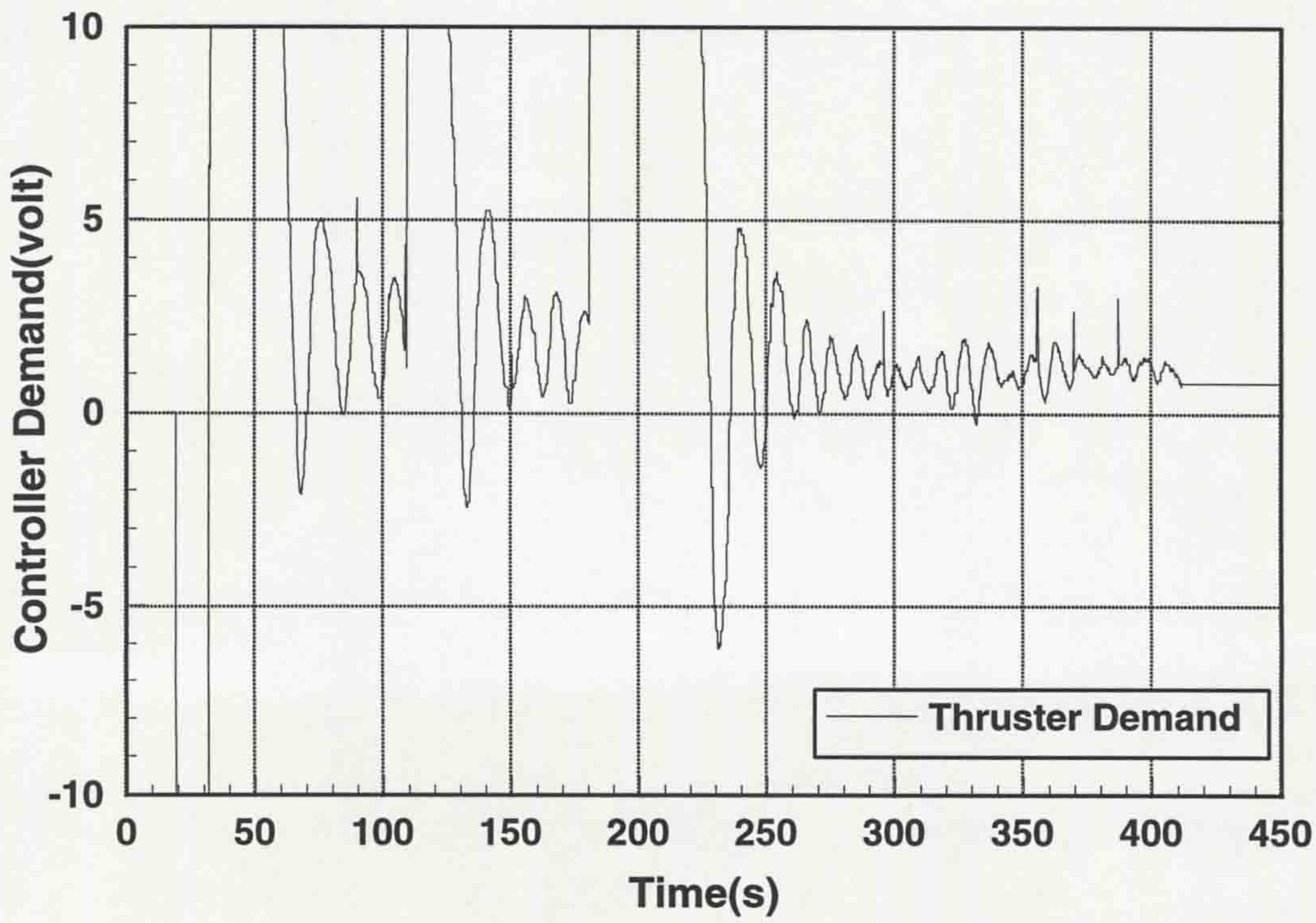

Figure 83, Thruster Demand

7.3.3 Aquasphere Heading Control

One of the other principle control systems on the vehicle was the automatic heading control. In tuning the heading control the same difficulties were experienced in availability of testing as previously mentioned in section 7.3.2. An additional problem experienced with the control of the heading was the slow update rate of the heading sensor. The update rate of approximately 1Hz lead to a lag in the control loop which allowed instability in the control. Combined with the large mass of the vehicle this posed problems for the heading control.

Without a model for the vehicle it was not possible to interpolate between samples accurately, and the information from the rate gyroscope was not reliable. However, a linear interpolation was made between samples to maintain the control output. From the test data a more accurate method for interpolation may be devised.

To control such a system the response of the vehicle was slowed down to avoid oscillation and overshoot. This is reflected in the low gains used for the auto-heading control system.

The gains used for the automatic heading control were:-

$K_p = 0.2$

$K_d = 3.0$

$K_i = 0.1$

$K_u = 20.0$

A graph of the automatic heading response for the Aquasphere is shown in Figure 84.. The slow update rate of the sensor causes overshoot, and hunting in the response of the vehicle. Further tuning of the vehicle may improve the response of the vehicle from that shown in the graph. A sensor with a faster update rate is recommended if the vehicle is to be developed in the future.

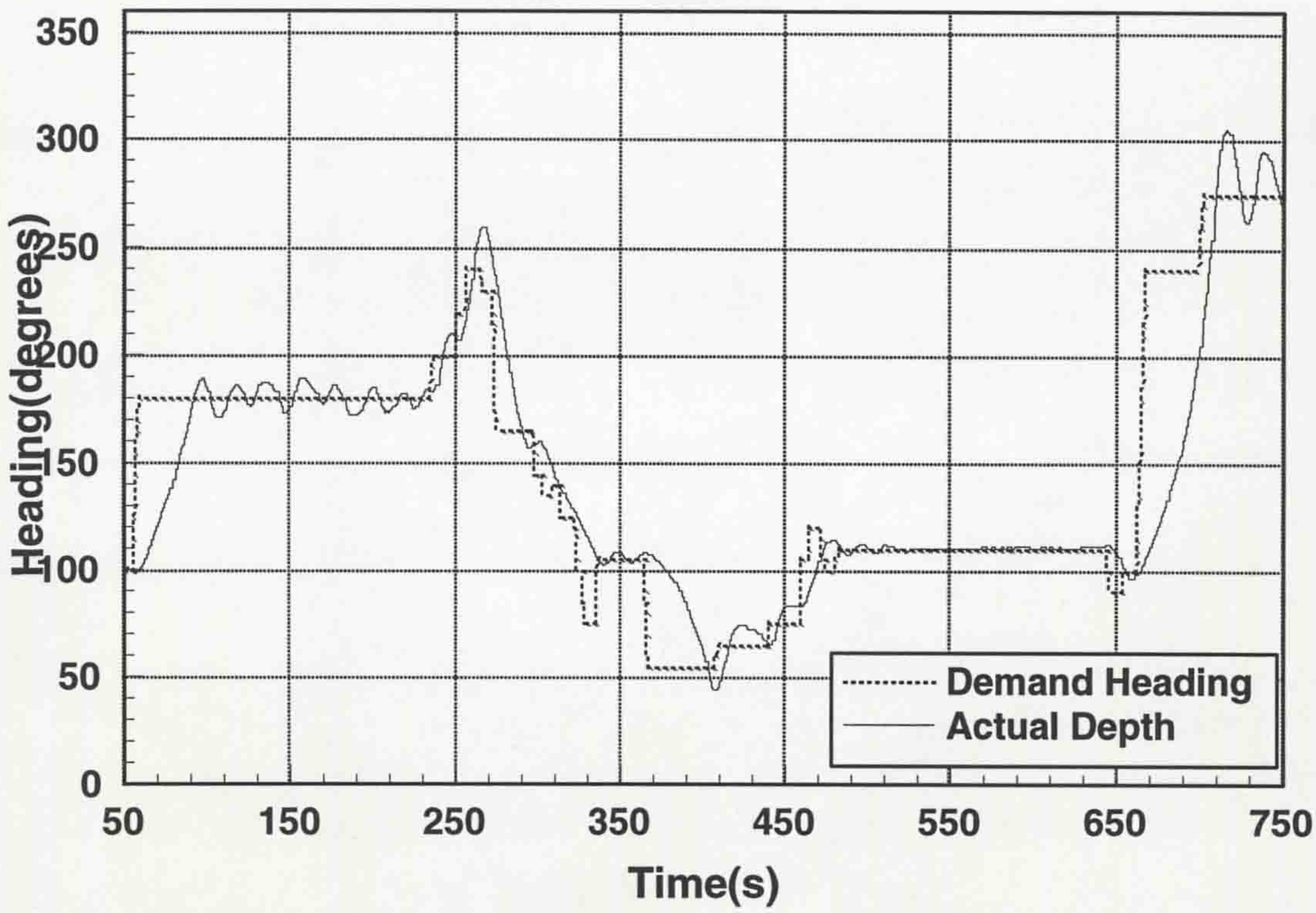

Figure 84, Aquasphere Auto-Heading Response

7.3.4 Aquasphere Pitch & Roll Control

Due to the changes in the configuration of the Aquasphere the vehicle no longer had a centre of mass next the centre of buoyancy. This meant that the vehicle was no longer rotationally unstable for the pitch and roll axes. The new configuration of the vehicle, whilst providing stability, made the force necessary to alter the vehicle's pitch and roll too great for the thrusters to have any significant effect. In attempting to correct the vehicles pitch and roll the full power of the thrusters was used. This only altered the angle of the vehicle approximately two degrees. When this function was operated there was no residual power with which to navigate, or to perform other control functions. Also, as the desired values were never reached due to the lack of thruster power the control system could not be tuned

This meant that although the control system could be seen to be trying to reduce the error it was unable to achieve its aim. The automatic level control system was therefore not used for the final trials, as it lead to the other control systems being impaired.

8. Conclusions and Future Work

8.1 Thesis Summary

The aim of this thesis was to design a remotely operated vehicle system with a closed loop control system capable of adjusting and stabilising the motion of the vehicle. To achieve this aim a variety of separate tasks needed to be completed

8.1.1 Vehicle Design

An underwater vehicle, the Aquasphere, was designed to provide a suitable test bed for the control, sensor, thruster and networking aspects of this project. In addition the vehicle provided a suitable platform for other technologies, such as the manipulator arm and vision system, to be used. The unique design of the Aquasphere allows the capability of full positional control, and the capacity to stabilise the vehicle from external forces at any vector. The design of the vehicle altered from the originally conceived spherical control test platform to a fully populated vehicle. Marinisation of the electronics was incorporated into the design, and methods for mounting the technologies added towards the end of the project were provided.

8.1.2 Thruster Systems

The problems introduced into a control system caused by the non-linear behavior of thruster systems led to extensive testing being carried out on the particular thruster used. From the experimental data a model of the dynamics of this non-linear system was created for incorporation into a control system. To power the thruster a pulse width modulation drive system was created, suitable for connection with the networking system. With the aim of improving the efficiency, weight and size of the thruster drives, and through the availability of new components, preliminary work has been undertaken into improved thruster drives.

8.1.3 Sensor Systems

A variety of sensors were connected to the vehicle to provide the information required for closed loop control. To choose sensors from the large range available a brief study was made into the sensor requirement for the vehicle. On the final vehicle two tilt

sensors, a flux-gate compass, a temperature sensor, a flow sensor, an ultrasonic range sensor, a gyroscope, an accelerometer and a pressure sensor were used. An ultrasonic absolute position sensor was to be attached, but this was not available in time for the final trials. The variables successfully measured by these sensors were roll, pitch, heading (yaw) and depth, allowing closed loop control of these parameters.

8.1.4 Networking System

To be able to transfer data between the vehicle and the control system an interrupt based communication package was written. The software converted the outgoing information into packets suitable for transmission on the networking hardware, whilst being able to accept, decode and interpret incoming data from the system. The software allowed the control system to be commanded from a remote location, retaining the full capability of the system.

8.1.5 Control System

To control the highly non-linear motion of the vehicle an investigation was made into the equations of motion for an ROV. Without the necessary sensor or test data it was not possible to formulate a model of the vehicle's characteristics. For the control of the vehicle a fuzzy PID controller was used. This combined the robustness of a conventional controller, with the non-linear control capabilities of a fuzzy based system. Separate control routines were written to control each measured degree of freedom of the vehicle. Tuning of the control system allowed control of the vehicles 'hover' and heading to be made. Auto-leveling systems were implemented, but tuning of this these was not possible due to the dynamics of the system.

8.2 Summary of Results

The design concept of the Aquasphere proved to be successful. Although the vehicle underwent several stages of modification the overall design concept was kept throughout the project. The evolution of the vehicle from its concept of a freely rotating vehicle with no defined forwards direction and zero metacentric height, to a more standard vehicle with defined features and stability was completed successfully. The vehicle itself was reliable throughout the testing stages, and needed minimal maintenance (except for occasional painting). The thruster systems and their drives performed well throughout the tests.

However, as the size and the mass of the vehicle and its systems increased the thrust to mass ratio of the vehicle became insufficient The performance of some of the sensors was inadequate for the project, and this was reflected in the overall results for the control. The lack of the absolute positioning module meant that these testing sensors had to be used for the final trials. The networking system performed well throughout the final testing stages, after initial transmission and reception problems were resolved. The difficulties experienced in being able to test the fully assembled vehicle meant that there was very little time in which the control system could be tested. Although the control system performed well, further testing would be required to fully tune the control system variables. Also the limitations of the sensor system, and the available thrust ratio, imposed limitations on the quality and availability of the control systems used on the final vehicle.

8.3 Suggestions for the Future

The suggestions for future work on the project depends upon the role for which the vehicle is to be used in the future. Before further development it is necessary to know if the vehicle is intended as a control test platform, as the Aquasphere was originally, or as a test bed for other technologies. Suggestions for improving the performance of the existing Aquasphere system are:-

1. Improve the quality of the sensing system by choosing alternative sensor modules, or one of the now widely available ROV sensor modules which incorporates all the standard measurement sensors in one complete unit.
2. Modify the thruster drives to make use of new technology, giving smaller, more efficient, self contained, closed-loop controlled thrusters.
3. Either the mass of the Aquasphere must be reduced by the removal or reengineering of the current systems, or new thrusters must be developed with increased thrust capabilities.
4. A self-tuning control system could be used on the vehicle, eliminating the need for manual tuning of the control system. Ideally a model of the vehicle should be formulated to allow for adaptive model reference techniques to be used.
5. The Networking cards should be modified to provide hardware interrupts, eliminating the need for software polling of the incoming data.
6. The mass of the vehicle and the buoyancy position should be altered to give a lower metacentric height, to allow the vehicle to freely rotate. This must be done in conjunction with 3.

7. Alternative methods for buoyancy should be investigated, including the use of specially shaped buoyancy, or adaptive buoyancy methods.

For the design of a new vehicle most of the points raised above should be undertaken. In the design of a more conventional vehicle the stability should be incorporated into the design, with the classical "buoyancy at the top and weight at the bottom" shape. The vehicle should be modeled prior to design using a package such as AutoCad, requiring an exact inflexible specification of the vehicle to be decided upon in the initial project stages. A concept drawing of a classically shaped vehicle using the existing vehicle components is shown in Figure 85.

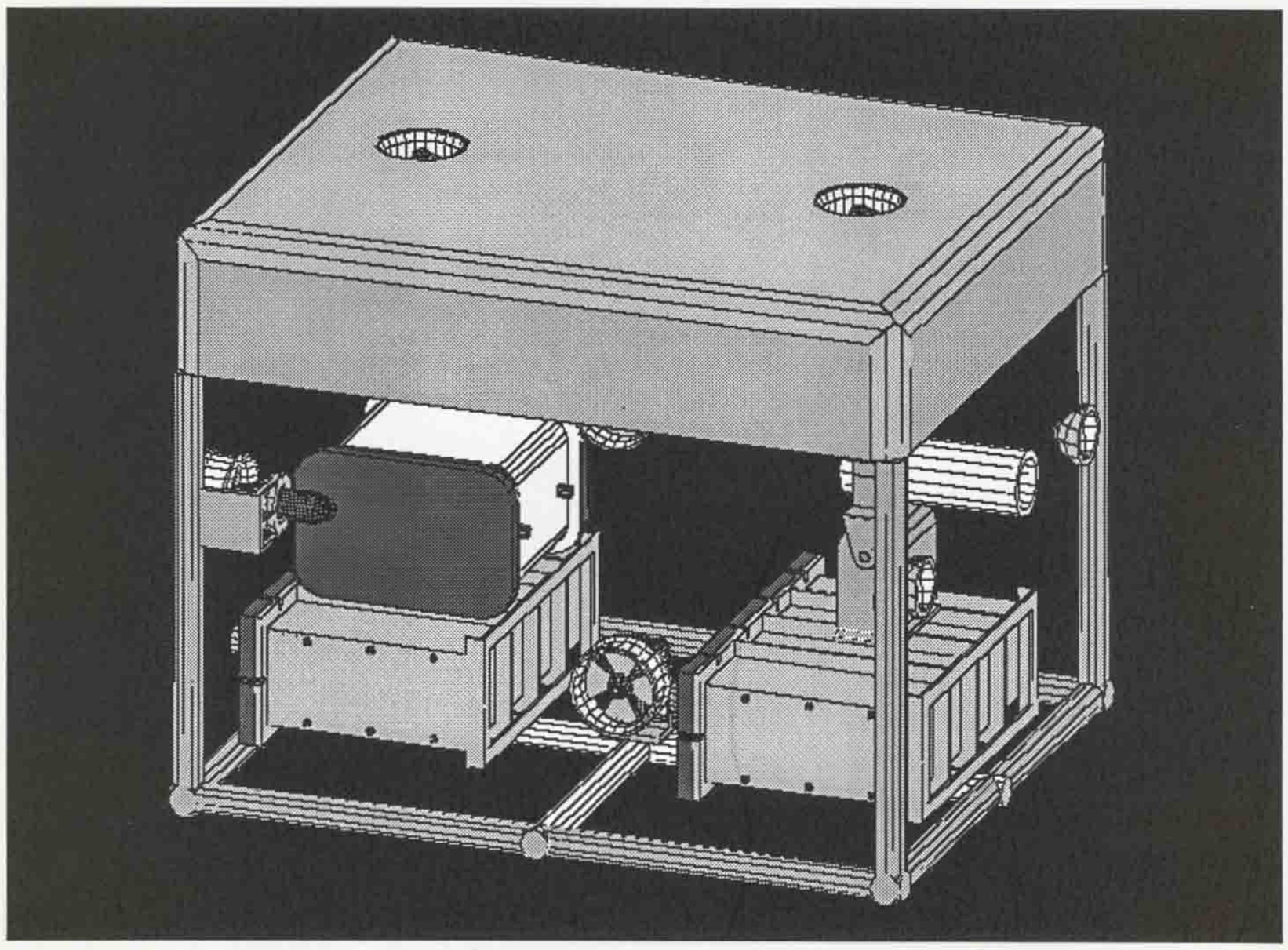

Figure 85, Possible Future Vehicle Design

9. References and Bibliography

9.1 Chapter 2

[2.1] ROV Review, 4th ed. 1991-92, 1993. Windate Enterprises Inc., ISBN 0-9623145-2-8

[2.2] Hover F. S., Newman J. B., "Rov deployment design for performance and survivability", Monterey Bay Aquarium, Pacific Grove, Ca, 93950., Oceans 94.

[2.3] Newman J. B., "Fiber-Optic Data Network for the ARGO/JASON Vehicle System", IEEE Journal of Oceanic Engineering, April 1990, Vol. 15, No. 2, pp66-71.

[2.4] Hydrovision Ltd., Seaway Retail Park Aberdeen.

[2.5] John C.Hyland, Fred J.Taylor, "Mine Avoidance Techniques for Underwater Vehicles", IEEE Journal of Oceanic engineering, Vol. 18, No. 3, July 1993.

[2.6] M.D.Ageev, L.V. Kiselev, A. Ph. Shcherbatyuk, "Tasks for Autonomous Underwater Robot", IEEE Journal of Oceanic engineering, May 1991.

[2.7] Hans Kurt Tonshoff, Ioannis Papadimitriou, Ralf Zeyn, "Multisensoriel Control and Supervision of Autonomous Mobile Underwater Robots", IEEE Journal of Oceanic engineering, April 1991.

[2.8] Glenn Zorpette, "Autopilots of the deep", IEEE Spectrum August 1994

[2.9] Robert L. Rosenfeld, Paul R. Prokopius, Alfred P. Meyer, "Fuel Cell Power System Development for Submersibles", IEEE 0-7803-0704-6/92.

[2.10] Dand I. W.,Every M. J.,"An overview of the hydrodynamics of umbilical cables and vehicles, parts I and II", Proc. SUBTECH 83, London 1983.

[2.11] Dana R. Yoerger, John G. Cooke, Jean-Jacques E. Slotine, "The Influence of Thruster Dynamics on Underwater Vehicle Behavior and Their Incorporation into Control System Design", IEEE Journal of Oceanic Engineering, Vol. 15, No. 3, July 1990.

[2.12] Kevin R. Goheen, E. Richard Jefferys, "Multivariable Self-Tuning Autopilots For Autonomous and Remotely Operated Underwater Vehicles", IEEE 0364-9059/90/0700-0144, 1990.

[2.13] Qaio Hong, "On the simulation of an ROV system", University of Strathclyde, May 1991.

[2.14] Russell E. Trahan, Steven Yeadon, Maurice G. Theile Jr., "Optimal Estimation of Layback Distance For Marine Towed Vehicles", IEEE journal of Oceanic Engineering, Vol.16, No.2, April 1991.

[2.15] M. W. Dunnigan, D. M. Lane, "Evaluation and Reduction of the Dynamic Coupling Between an ROV and Manipulator", IEE CD-ROM 1993.

[2.16] Castro E. L., "A marine Robot Model", University of Strathclyde, May 1991.

[2.17] Yuh J., Kishan V. Gonugunta, "Learning Control of Underwater Robotic Vehicles", IEEE 1050-4279/93, 1993.

[2.18] Thor Fossen, Svien Sagatun, "Adaptive control of Nonlinear Underwater Robotic Systems", IEEE International Conference on Robotics and Automation, April 1991"

[2.19] Sansik Yang, Masayoshi Tomizuka, "Adaptive Pulse Width Control for Precise Positioning Under the Influence of Stiction and Coulomb Friction", Journal of Dynamic Systems Measurement and Control, September 1988, Vol. 110.

[2.20] Donald G. Langrock, "Application of Dynamic Positioning Techniques to Working Class Remotely Operated Vehicles (ROVs)", Mobil North Sea Limited, Aberdeen AB1 1SA, IEE CD-ROM 1993.

[2.21] C. Goodchild, "Precision Navigation and Guidance of Underwater Vehicles", Dept. of Aerospace Engineering, University of Glasgow, IEE CD-ROM 1993.

[2.22] S.H. Kwak , J.B. McKeon, J.R. Clynch , R.B. McGhee, "Incorporation of Global Positioning System into Autonomous Underwater Vehicle Navigation", IEEE 0-7803-0704-6/92.

[2.23] M. J. Hellard, "Magnetic Sensors for Underwater Navigation", Ultra Electronics Limited - Magnetics Division, IEE CD-ROM 1993.

[2.24] T.R. Crocker, "Motion Tolerant Compass and Attitude Sensor", 3D Instruments Ltd., Unit B, 1 Exe Street, Exeter EX4 3HD, Tel. 01392 496540.

[2.25] Gerhard Weiß, "Usage of a Vibrating Gyroscope for Orientation Estimation", University of Kaiserslautern, P.O. Box 3049, D-67653 Kaiserslautern.

[2.26] V. Rigaud, Irisa Ifremer, L. Marcé, "Sensor Fusion for AUV localization", International Journal of Robotic Research, Vol. 5, 1990.

[2.27] R. K. Hansen, "An Acoustic Camera for 3D Underwater Imaging", IEE conference on Acoustic Sensing and Imaging, 29-30th March 1993, Conference publication No. 369.

[2.28] Wylie S. R., "An Underwater Ultrasonic Imaging System", Ph.D. Thesis, The University of Liverpool, 1996.

[2.29] K. J. Åström, C.C. Hang, P. Persson, W. K. Ho, "Towards Intelligent P.I.D. Control", Automatica, Vol. 28, No.1, pp1-9, 1992.

[2.30] Satoshi Yanagawa, Ichiro Miki, "PID Auto-Tuning controller using a single neuron for D.C. Servomotor", IEE CD-ROM 1991.

[2.31] I. Miki, N. Nagai, S Nishiyama, T Yamada, "Vector Control of Induction Motor with fuzzy PI controller", IEEE, IAS, Annual Meeting, vol.1, pp.341-346, 1991.

[2.32] Zadeh L. A., "Fuzzy sets", IEE Information and Control, 1965, No. 8, pp. 338-353

[2.33] Dettmer R., "Fuzzy Control, engineering for empiricists", IEE Review, January 1994, pp. 17-20.

[2.34] MacLeod A. "Fuzzy logic cleans up for Hoover's motor control", Article, New Electronics on Campus, Spring 1994, pp. 22-26.

[2.35] Myers F. S., "Future clear for fuzzy flying", Article, Physics World, March 1994, pp.13

[2.36] Cox E., "Adaptive fuzzy systems", IEEE Spectrum February 1993, pp. 27-31

[2.37] Kohn M., "The logic that washes whiter", Article, The Independent, Thursday 5th May 1994, pp.23.

[2.38] Klir G. J., Folger T. A., "Fuzzy Sets, Uncertainty, and Information", Prentice-Hall, ISBN 0-13-345984-5, 1988.

[2.39] Zadeh L. A., "Fuzzy Logic", Computer Magazine, April 1988.

[2.40] N. Farbrother, B. A. Stacey, R. Sutton, "Fuzzy Self-organising control of a Remotely Operated Submersible", Royal Naval Engineering College, IEE CD-ROM 1993.

[2.41] Stephens M. , "Intelligent Adaptive Control of Remotely Operated Vehicles", Thesis Ph.D. Liverpool University, September 1993 .

[2.42] Stephens M., Shimmin D.W., "Robust Hierarchical Expert System Based Fuzzy Logic Controller with Application to Subsea Vehicles", IEE Control '94, Conference Publication No. 389

[2.43] D. W. Shimmin, J. Lucas, "Control Strategies for Underwater Vehicle Autopilots", IEE C9 Colloquium on Control and Guidance of Underwater Vehicles, IEE Savoy Place, 3rd December 1993.

[2.44] Shimmin D.W., Stephens M., Swainston J.R., "Adaptive control of a submerged vehicle with sliding fuzzy relations", Fuzzy Sets and Systems 79 (1996), pp. 15-24.

[2.45] S. M. Smith, G. J. S. Rae, D. T. Anderson, "Applications of Fuzzy Logic to the Control of an Autonomous Underwater Vehicle", Second IEEE International Conference on Fuzzy Systems, Vol. 1 and Vol. 2, Ch. 243, pp. 1099-1106, 1993

[2.46] Åström K. J., Wittenmark B, "Adaptive Control", Addison-Wesley, ISBN 0-201-09720-6, 1989.

[2.47] J. Yuh, Ranganath Lakshmi, "An Intelligent Control System for Remotely Operated Vehicles", IEEE Journal of Oceanic Engineering, Vol. 18, No. 1, January 1993.

[2.48] Stover J. A., Gibson R.E., "Continuous Inference Networks for Autonomous Systems", Applied Research Laboratory, Pennsylvania State University.

[2.49] Alan C. Schultz, John J. Grefenstette, Kenneth A. De Jong, "Adaptive Testing of Controllers for Autonomous Vehicles", IEEE 0-7803-0704-6/92.

[2.50] Utkin. V. I, "Variable Structure Systems with Sliding Modes", IEEE Trans. On Automatic Control, April 1977,, Vol AC22, No 2.

[2.51] Itkis U., "Control Systems of Variable Structure", Wiley NY, 1976.

[2.52] Zinober A. S. I., El-Ghezawi O. M. E., "Multivariable Variable-Structure Adaptive Model Following Control Systems", IEE proceedings, January 1982, Vol. 129, No.1

[2.53] José P. V. S. da Cunha, Ramón R. Costa, Liu Hsu, "Input/output variable structure position control of a remotely operated underwater vehicle", IEEE 7803-0078/91/0600-1305.

[2.54] Olgac N., B. E. Platin, J. -M. Chang, "Sliding mode control of remotely operated vehicles for horizontal plane motions", IEE proceedings-D, Vol. 138, No.5, September 1991.

9.2 Chapter 3

[3.1] J.R.Swainston, "Positional Control of Underwater Vehicles", M.Sc Thesis. Liverpool University, December 1992.

[3.2] M. Stephens, "Intelligent Adaptive Control of Remotely Operated Vehicles", Thesis Ph.D. Liverpool University, September 1993 .

[3.3] Vasilios S Kodogiannis, "Neural Network Techniques for modelling and learning control of an underwater robotic vehicle", Thesis Ph.D., Liverpool University, 1994.

[3.4] Autocad produced by Autodesk Ltd., Cross Lanes, Guildford, Surry, GU1 1UJ

[3.5] Rui Yu, "A three dimensional vision system for remotely operated vehicles", Ph.D. Thesis, The University of Liverpool, August 1994.

[3.6] Sarafis, Ilias Thoma, "Electrically driven underwater manipulator for remote operated vehicles", Thesis Ph.D., Liverpool University, 1994

9.3 Chapter 4

[4.1] Millward A., K Nicholson, J.H. Preston, "The Use of Jet Injection to Produce Uniform Velocity in A High Speed Water Channel", Journal of Ship Research, Vol. 24, No. 2, June 1980, pp128-134.

[4.2] Dana R. Yoerger, John G. Cooke, Jean-Jacques E. Slotine, "The Influence of Thruster Dynamics on Underwater Vehicle Behavior and Their Incorporation into

Control System Design", IEEE Journal of Oceanic Engineering, Vol. 15, No. 3, July 1990.

[4.3] J.R.Swainston, "Positional Control of Underwater Vehicles", Thesis M.Sc. Liverpool University, December 1992.

[4.4] Millward A., Rossiter J., "The Design of a Multi-Purpose Multi-Component Strain Gauge Dynamometer", Strain, February 1983, pp27-30.

[4.5] Richard C. Dorf, "Modern Control Systems", ISBN 0-201-05319-5.

[4.6] Matlab, Created by The MathWorks Inc., Cochituate Place, 24 Prime Park Way, Natik, Mass. , 01760.

[4.7] Bennett, Nicholas, "A vector controlled ac drive incorporating an ultrasonic voltage source inverter and a permanent magnet synchronous machine", Ph.D. Thesis Liverpool University 1993.

[4.8] Paul Horowitz, Winfield Hill, "The Art of Electronics", Second Edition, Cambridge University Press, ISBN 0-521-37095-7

[4.9] Majid A. Al Taee, "A Synchronous Ultrasonically Modulated Drive System Incorporating A Permanent Magnet Machine", Ph.D. Thesis Liverpool University 1990.

[4.10] Intel, "MCS®51 Microcontroller Family User's Manual", February 1995, order number 272383-002.

[4.11] Ciebo, DB-51 Development Board, Supplied by Philips Semiconductors.

[4.12] Hydrovision Ltd., Seaway Retail Park Aberdeen.

[4.13] Sansik Yang, Masayoshi Tomizuka, "Adaptive Pulse Width Control for Precise Positioning Under the Influence of Stiction and Coulomb Friction", Journal of Dynamic Systems Measurement and Control, September 1988, Vol. 110.

9.4 Chapter 5

[5.1] V. Rigaud, Irisa Ifremer, L. Marcé, "Sensor Fusion for AUV localization", International Journal of Robotic Research, Vol. 5, 1990.

[5.2] Gerhard Weiß, "Usage of a Vibrating Gyroscope for Orientation Estimation", University of Kaiserslautern, P.O. Box 3049, D-67653 Kaiserslautern.

[5.3] M. J. Hellard, "Magnetic Sensors for Underwater Navigation", Ultra Electronics Limited - Magnetics Division, IEE CD-ROM 1993.

[5.4] T.R. Crocker, "Motion Tolerant Compass and Attitude Sensor", 3D Instruments Ltd., Unit B, 1 Exe Street, Exeter EX4 3HD, Tel. 01392 496540.

[5.5] Navstar, Federal Radionavigation Plan, DOT-VNTSC-RSPA-92-2/DOD-4850.5 National Technical Information Service, Springfield, Virginia 22161, 1992.

[5.6] C. Goodchild, "Precision Navigation and Guidance of Underwater Vehicles", Dept. of Aerospace Engineering, University of Glasgow, IEE CD-ROM 1993.

[5.7] S.H. Kwak , J.B. McKeon, J.R. Clynch , R.B. McGhee, "Incorporation of Global Positioning System into Autonomous Underwater Vehicle Navigation", IEEE 0-7803-0704-6/92.

[5.8] R. K. Hansen, "An Acoustic Camera for 3D Underwater Imaging", IEE conference on Acoustic Sensing and Imaging, 29-30th March 1993, Conference publication No. 369.

[5.9] Rui Yu, "A three dimensional vision system for remotely operated vehicles", Ph.D. Thesis, The University of Liverpool, August 1994.

[5.10] Alexander Wood, "Acoustics", Published by Blackie & Son Limited, 1950.

9.5 Chapter 6

[6.1] M. A. Houghton, "A Transputer Network for Intelligent Control Systems", Ph.D. Thesis, The University of Liverpool, December 1995.

[6.2] Amplicon Liveline Limited, Centenary Industrial Estate, Brighton.

[6.3] "NMEA Data Protocol", NMEA, P.O. Box 3435, New Bern NC, 28564-3435, USA.

[6.4] Chambers S. P, "TIPS: a transputer based real time vision system", Thesis Ph. D., The University of Liverpool, 1990.

[6.5] MAST-0030-C Report, "Combined Sensor and Information Technology for Subsea Positioning, Imaging and Control For Task Implementation", Project Coordinators The University of Liverpool.

9.6 Chapter 7

[7.1] "Underwater vehicle trials, Loch Linnhe 1981,82", Dept. of Energy Offshore Tech. Board, London, UK OTP 14, CIRA, 1983.

[7.2] Kevin R. Goheen, E. Richard Jefferys, "Multivariable Self-Tuning Autopilots For Autonomous and Remotely Operated Underwater Vehicles", IEEE 0364-9059/90/0700-0144, 1990.

[7.3] R.C.Hibbler, "Engineering Mathematics Dynamics", 4th Edition, ISBN 0-02-946250-9.

[7.4] Etkin B., "Dynamics of Atmospheric Flight", John Wiley and Sons, 1972.

[7.6] K.R. Goheen, E. R. Jefferys, "The Application of Alternative Modelling Techniques to ROV Dynamics", IEEE CH2876-1/90/0000/1302, 1990.

[7.7] N. Farbrother, B. A. Stacey, R. Sutton, "Fuzzy Self-organising control of a Remotely Operated Submersible", Royal Naval Engineering College, IEE CD-ROM 1993.

[7.8] J.R.Swainston, "Positional Control of Underwater Vehicles", Thesis M.Sc. Liverpool University, December 1992.

[7.9] Stephens M. , "Intelligent Adaptive Control of Remotely Operated Vehicles", Thesis Ph.D. Liverpool University, September 1993 .

[7.10] Stephens M., Shimmin D.W., "Robust Hierarchical Expert System Based Fuzzy Logic Controller with Application to Subsea Vehicles", IEE Control '94, Conference Publication No. 389.

[7.11] Shimmin D.W., Stephens M., Swainston J.R., "Adaptive control of a submerged vehicle with sliding fuzzy relations", Fuzzy Sets and Systems 79 (1996), pp. 15-24.

[7.12] Vasilios S Kodogiannis, "Neural Network Techniques for modelling and learning control of an underwater robotic vehicle", Thesis Ph.D., Liverpool University, 1994.

[7.13] D. W. Shimmin, J. Lucas, "Control Strategies for Underwater Vehicle Autopilots", IEE C9 Colloquium on Control and Guidance of Underwater Vehicles, IEE Savoy Place, 3rd December 1993.

[7.14] K. J. Åström, C.C. Hang, P. Persson, W. K. Ho, "Towards Intelligent P.I.D. Control", Automatica, Vol. 28, No.1, pp1-9, 1992.

[7.15] Zadeh L. A., "Fuzzy Logic", Computer Magazine, April 1988.

[7.16] Klir G. J., Folger T. A., "Fuzzy Sets, Uncertainty, and Information", Prentice-Hall, ISBN 0-13-345984-5, 1988.

9.7 Bibliography

L. E. Virr, "Increased electric shock risk underwater due to electrode configuration and insulating boundaries", IEE proceedings, Vol. 137, No.5, September 1990.

L. E. Virr, "Role of electricity in subsea intervention", IEE proceedings, Vol. 134, No.6, June 1987.

"Code of Practice for the Safe Use of Electricity Under Water.", ISBN 086017-191-4.

Emmanuel I. Agba, Ming Z. Huang, Tin-Lup Wong, "A Hybrid Simulator For Undersea Robotic Telemanipulation", IEEE proceedings 1990 Southeastcon, pp636-641.

Russell E. Trahan Jr., D. Steven Yeadon, Maurice G. Thiele Jr., "Optimal Estimation of Layback Distance For Marine Towed Cables", IEEE Journal of Oceanic Engineering, Vol. 16, No. 2, April 1991.

Butler P. Hine III, Carol Stoker, Michael Sims, Daryl Rassmussen, Phil Hontalas, "The Application of Telepresence and Virtual Reality to Subsea Exploration", NASA Ames Research Centre publication 1994.

Rodney Coates, John Miller, "Cutting the ROV Umbilical", IEE CD-ROM 1993.

Ikuo Yamamoto, Yuuzi Terada, Kozo Yokomura, "An Application of a Position Keeping Control System to Floating Offshore Platform", IECON 91 Proceedings, IEEE CH2976-9/91/0000/1867.

Bozzo G.M., Maddalena D., Terribile A, "Laboratory and Sea Tests of the Supervisory Controlled Underwater Telemanipulation System", IEEE 7803-0078/91/0600-1336.

Gordon A., "Use of Laser Scanning System on mobile underwater platforms", IEEE 0-7803-0704-6/92

Yu Chih-Ho, Negahdaripour S., "Underwater experiments for orientation and motion recovery from video images", IEEE 1050-4729/93.

10. Appendices

10.1 Solid Modeling Information

The design of the ROVs components using the Autocad package allowed data on the designed items to be extracted. The most useful values extracted are the mass properties, giving mass and moments of inertia for the designed object.

10.1.1 Thruster Mounts Solid Modeling Information

The solid model created of the thruster mount provides the following information about the properties of the thruster mounts:-

```
Ray projection along X axis, level of subdivision: 7.
Mass:              1292.845 gm
Volume:            164.4841 cu cm  (Err: 1.582272)

Bounding box:            X: -5  --  10.7 cm
                         Y: -5  --  5 cm
                         Z: -4.5  --  5.5 cm

Centroid:                X: 7.053572  cm    (Err: 0.02672679)
                         Y: 0.008586003  cm    (Err: 0.04110152)
                         Z: 0.501698  cm    (Err: 0.01491085)

Moments of inertia:      X: 18543.74  gm sq cm (Err: 197.9783)
                         Y: 100900.7  gm sq cm (Err: 303.672)
                         Z: 105610.7  gm sq cm (Err: 468.948)
Products of inertia:    XY: 43.55946  gm sq cm (Err: 253.5305)
                        YZ: 8.451202  gm sq cm (Err: 59.48168)
                        ZX: 4566.176  gm sq cm (Err: 182.1684)

Radii of gyration:       X: 3.787262  cm
                         Y: 8.834336  cm
                         Z: 9.038176  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                         I: 18218.16 along [0.9999981 -0.001926254 -0.0003858898]
                         J: 41287.88 along [0.000384793 -0.0005697364 0.9999998]
                         K: 36252.64 along [-0.001926474 -0.999998 -0.0005689941]
```

10.1.2 Spider-Joint Solid Modeling Information

The solid modelling data for the spider-joint gives:-

```
Ray projection along X axis, level of subdivision: 7.
Mass:           1087.154 gm
Volume:         138.3148 cu cm  (Err: 0.8686934)

Bounding box:         X: -7.985242  --  8.031028 cm
                      Y: -13.75272  --  -9.996685 cm
                      Z: 87.6945  --  101.5669 cm

Centroid:             X: 0.02182493  cm    (Err: 0.000176345)
                      Y: -11.87272  cm    (Err: 0.07504267)
                      Z: 96.2776  cm    (Err: 0.6296974)

Moments of inertia:   X: 10244042  gm sq cm (Err: 69552.85)
                      Y: 10102949  gm sq cm (Err: 68737.62)
                      Z: 166803.8  gm sq cm (Err: 1150.602)
Products of inertia: XY: -281.7145  gm sq cm (Err: 2.295534)
                     YZ: -1242703  gm sq cm (Err: 8185.267)
                     ZX: 2284.45  gm sq cm (Err: 19.05652)

Radii of gyration:    X: 97.07114  cm
                      Y: 96.40034  cm
                      Z: 12.38675  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                      I: 13554.6 along [0.9994256 -8.929629e-06 0.03388804]
                      J: 25708.22 along [8.061586e-07 1 0.0002397286]
                      K: 13556.54 along [-0.03388805 -0.0002395635 0.9994256]
```

10.1.3 Internal Frame Supports Solid Modeling Information

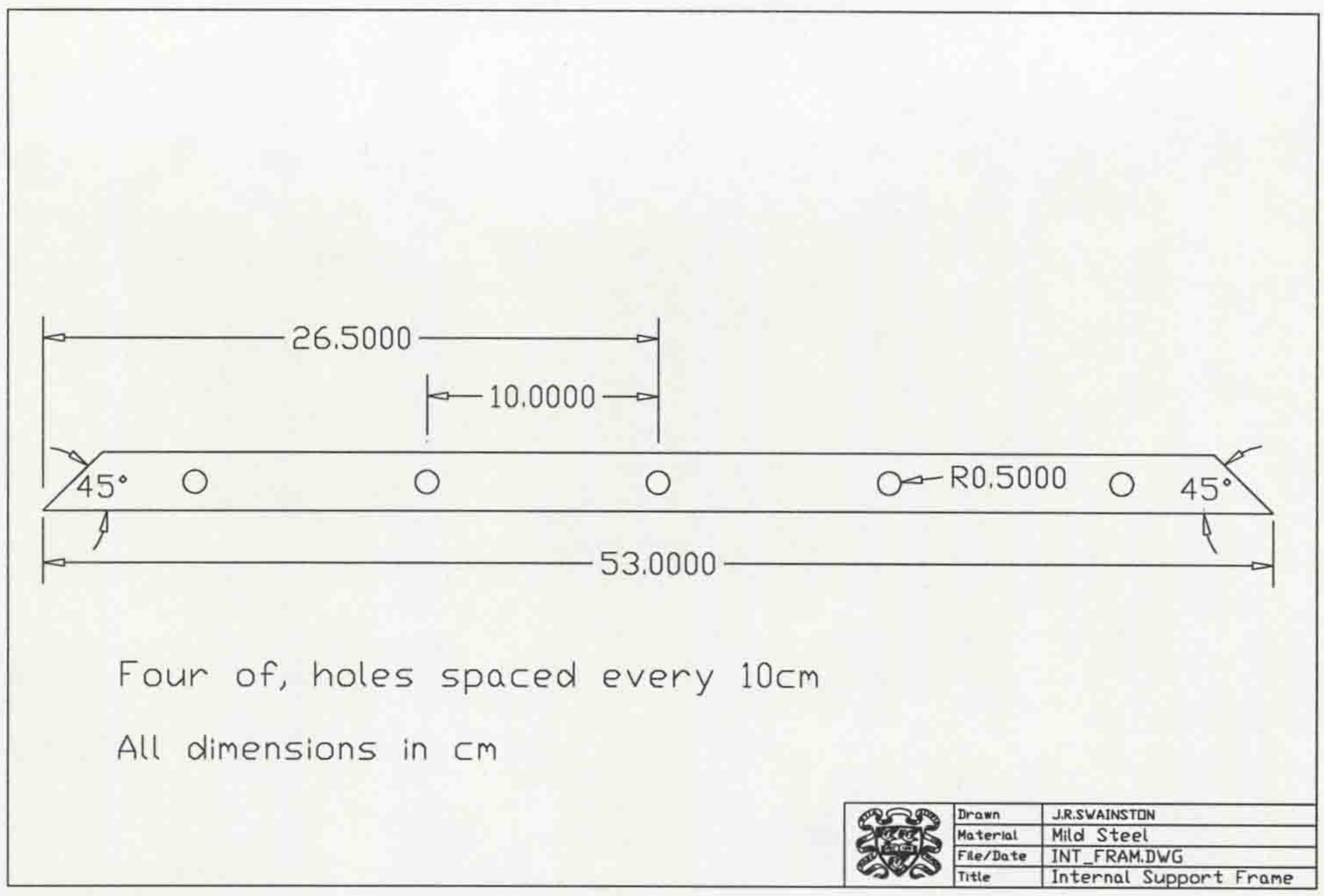

Figure 86, Internal Frame Supports

From the package Autocad the following information can be found for each of the supports shown in Figure 86, Internal Frame Supports:-

```
Ray projection along X axis, level of subdivision: 5.
Mass:            620.029 gm
Volume:          78.88409 cu cm  (Err: 0.1255519)


Bounding box:          X: 0  --  53 cm
                       Y: 0  --  2.54 cm
                       Z: 0  --  0.635 cm

Centroid:              X: 26.5  cm    (Err: 0.04217739)
                       Y: 1.248667  cm    (Err: 0.02243445)
                       Z: 0.3175092  cm    (Err: 0.005747078)

Moments of inertia:    X: 1392.302  gm sq cm (Err: 36.93889)
                       Y: 567639.8  gm sq cm (Err: 1061.336)
                       Z: 568865.6  gm sq cm (Err: 1041.516)
Products of inertia: XY: 20516.56  gm sq cm (Err: 368.6152)
                     YZ: 245.8165  gm sq cm (Err: 6.750574)
                     ZX: 5216.92  gm sq cm (Err: 94.4289)

Radii of gyration:     X: 1.498514  cm
                       Y: 30.25732  cm
```

```
                    Z: 30.28997  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                    I: 363.0654 along [1 4.617489e-17 1.717596e-17]
                    J: 132483.5 along [-1.717628e-17 6.817325e-06 1]
                    K: 132162 along [4.617477e-17 -1 6.817325e-06]
```

10.1.4 Vertical Frame Supports Solid model Information

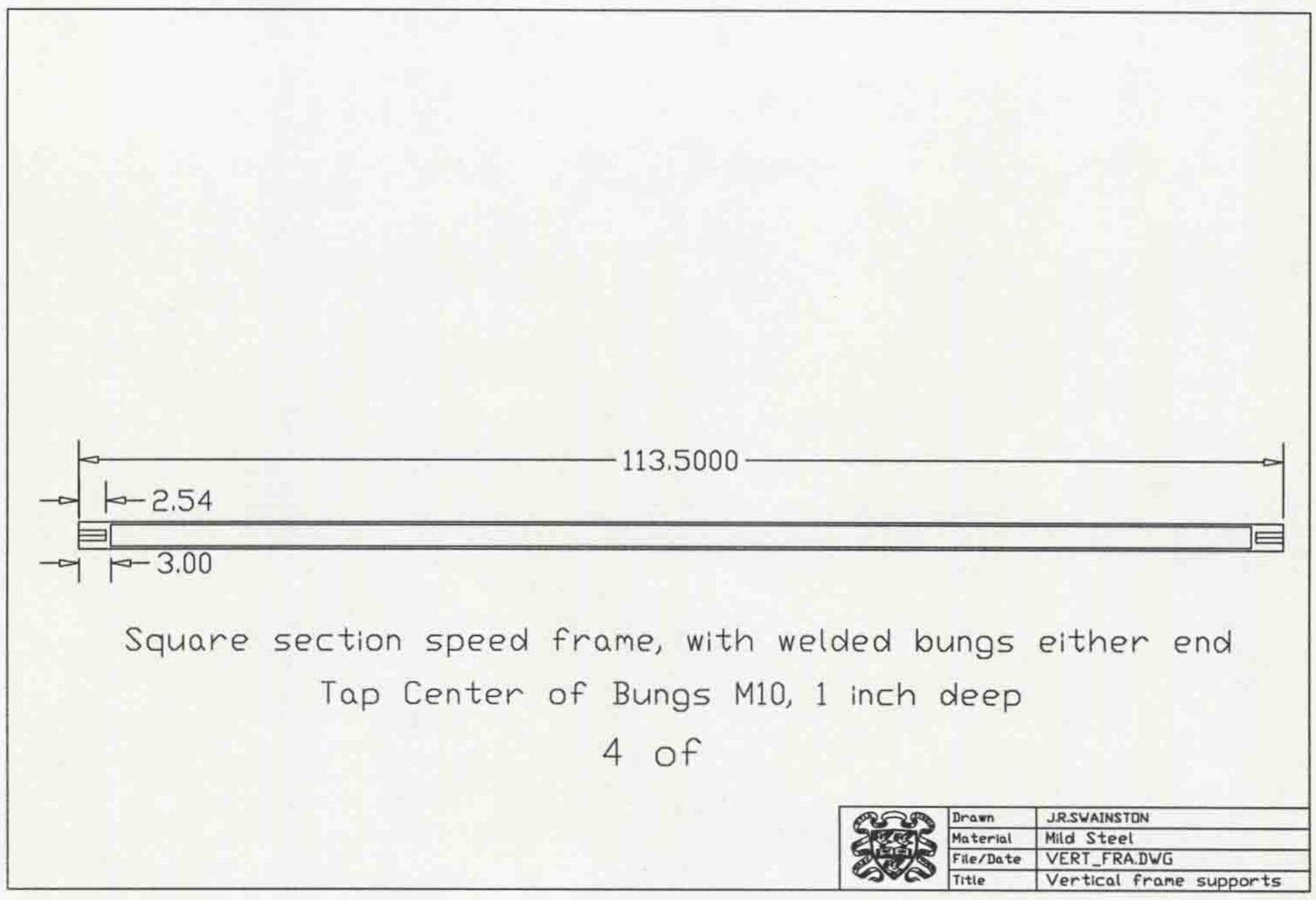

Figure 87, Vertical Frame Supports

Again, from the package Autocad the following information about the vertical struts can be found:-

```
Ray projection along X axis, level of subdivision: 5.
Mass:           2239.605 gm
Volume:         284.9371 cu cm  (Err: 10.50267)

Bounding box:         X: -56.75  --  56.75 cm
                      Y: -1.033678e-14  --  2.54 cm
                      Z: 0  --  2.54 cm

Centroid:             X: 3.885977e-15  cm    (Err: 1.585262e-16)
                      Y: 1.265802  cm    (Err: 0.06499105)
                      Z: 1.280073  cm    (Err: 0.06537019)

Moments of inertia:   X: 11035.31  gm sq cm (Err: 555.171)
                      Y: 2731280  gm sq cm (Err: 81620.98)
```

```
                    Z: 2731212  gm sq cm (Err: 81617.65)
Products of inertia: XY: 1.188395e-11  gm sq cm (Err: 5.669263e-13)
                    YZ: 3635.882  gm sq cm (Err: 241.3337)
                    ZX: 1.153924e-11  gm sq cm (Err: 5.582256e-13)

Radii of gyration:   X: 2.219762  cm
                     Y: 34.92186  cm
                     Z: 34.92143  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                     I: 3777.102 along [1 3.185237e-19 1.463719e-19]
                     J: 2727627 along [8.075041e-21 0.3965127 -0.9180292]
                     K: 2727608 along [-3.504524e-19 0.9180292 0.3965127]
```

10.1.5 Network Box Solid Model Information

```
Ray projection along X axis, level of subdivision: 5.
Mass:            32519.98 gm
Volume:          2860.156 cu cm  (Err: 265.1807)

Bounding box:         X: -55  --  0 cm
                      Y: 0  --  28 cm
                      Z: 0  --  25 cm

Centroid:             X: -27.5  cm    (Err: 2.549675)
                      Y: 14.75213  cm    (Err: 1.699225)
                      Z: 12.35315  cm    (Err: 1.46772)

Moments of inertia:   X: 17436124  gm sq cm (Err: 2042088)
                      Y: 41434571  gm sq cm (Err: 3702217)
                      Z: 44069442  gm sq cm (Err: 3980954)
Products of inertia: XY: -13192817  gm sq cm (Err: 1519616)
                     YZ: 5870072  gm sq cm (Err: 818748)
                     ZX: -11047414  gm sq cm (Err: 1312581)

Radii of gyration:   X: 23.15527  cm
                     Y: 35.69491  cm
                     Z: 36.81236  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                     I: 5396398 along [1 3.499147e-15 3.349018e-15]
                     J: 12405047 along [-3.701722e-15 0.1062108 0.9943436]
                     K: 11872775 along [3.123652e-15 -0.9943436 0.1062108]
```

10.1.6 Sensor Box Solid Modeling Information

From the solid model of the component, the following information is gained:-

```
Ray projection along X axis, level of subdivision: 6.
Mass:            21858.48 gm
Volume:          8065.86 cu cm  (Err: 135.9786)
```

```
Bounding box:          X: 49.15  --  84.85 cm
                       Y: -6.799574e-15  --  25 cm
                       Z: -2.25  --  34.75 cm

Centroid:              X: 67  cm    (Err: 1.129522)
                       Y: 12.42394  cm    (Err: 0.2844893)
                       Z: 16.19247  cm    (Err: 0.3859679)

Moments of inertia:    X: 13218599  gm sq cm (Err: 327172.1)
                       Y: 1.109095e+08  gm sq cm (Err: 1845170)
                       Z: 1.071166e+08  gm sq cm (Err: 1770601)
Products of inertia: XY: 18195090  gm sq cm (Err: 416639.7)
                     YZ: 4397617  gm sq cm (Err: 132558.7)
                     ZX: 23714167  gm sq cm (Err: 565257)

Radii of gyration:     X: 24.59137  cm
                       Y: 71.23188  cm
                       Z: 70.00329  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                       I: 4113439 along [1 -1.561214e-10 -2.346171e-10]
                       J: 7055595 along [1.560802e-10 1 -0.0001754916]
                       K: 5619940 along [2.346445e-10 0.0001754916 1]
```

10.1.7 Thruster Box Solid Modeling Information

The information for the thruster and arm boxes derived from Autocad is:-

```
Ray projection along X axis, level of subdivision: 6.
Mass:            30975.58 gm
Volume:          3940.914 cu cm  (Err: 166.6619)

Bounding box:          X: -27.5  --  27.5 cm
                       Y: -44.5  --  0 cm
                       Z: -10  --  10 cm

Centroid:              X: 0.0001647951  cm    (Err: 0.0002197664)
                       Y: -23.20347  cm    (Err: 1.179885)
                       Z: 0.01324725  cm    (Err: 0.3201849)

Moments of inertia:    X: 24599526  gm sq cm (Err: 1402085)
                       Y: 9927655  gm sq cm (Err: 342762.1)
                       Z: 31653404  gm sq cm (Err: 1563554)
Products of inertia: XY: 34.16327  gm sq cm (Err: 63.00769)
                     YZ: 73367.98  gm sq cm (Err: 253924.8)
                     ZX: -84.6753  gm sq cm (Err: 44.23649)

Radii of gyration:     X: 28.18082  cm
                       Y: 17.9025  cm
                       Z: 31.9669  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                       I: 7922233 along [1 7.563812e-05 -1.112694e-05]
                       J: 14977478 along [1.236472e-05 -0.01641208 0.9998653]
```

```
K: 9926289 along [7.544531e-05 -0.9998653 -0.01641208]
```

10.1.8 Vision Box Solid Modeling Information

The data derived from the vision box is slightly different from the thruster control box, due to its smaller size. As can be seen the smaller size of the box reduces the mass of the vehicle by 4kg compared to the larger thruster/arm box :-

```
Ray projection along X axis, level of subdivision: 6.
Mass:            26050.46 gm
Volume:          3314.308 cu cm  (Err: 136.5324)

Bounding box:          X: -27.5  --  27.5 cm
                       Y: -10  --  10 cm
                       Z: -35.5  --  0 cm

Centroid:              X: 5.178357e-05  cm    (Err: 0.000216961)
                       Y: 0.04381619  cm    (Err: 0.3114463)
                       Z: -18.35252  cm    (Err: 0.9123186)

Moments of inertia:    X: 13523897  gm sq cm (Err: 738802.7)
                       Y: 19497747  gm sq cm (Err: 875931.4)
                       Z: 8363543  gm sq cm (Err: 282471.5)
Products of inertia: XY: 70.78146  gm sq cm (Err: 35.05446)
                     YZ: -56353.03  gm sq cm (Err: 163116.5)
                     ZX: -147.9716  gm sq cm (Err: 62.30694)

Radii of gyration:     X: 22.7847  cm
                       Y: 27.35801  cm
                       Z: 17.91791  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                       I: 4749663 along [1 1.204132e-05 -3.421315e-05]
                       J: 10724095 along [1.152689e-05 -0.9998875 -0.01499656]
                       K: 8362962 along [-3.438989e-05 0.01499656 -0.9998875]
```

10.1.9 Thruster Cowling Solid Modeling Information

The modeling data for the cowling gives:-

```
Ray projection along X axis, level of subdivision: 6.
Mass:            1364.058 gm
Volume:          169.8702 cu cm  (Err: 3.826346)

Bounding box:          X: -5  --  5 cm
                       Y: -5  --  5 cm
```

```
                    Z: -18.5  --  2.85 cm

Centroid:           X: -4.045135e-16  cm    (Err: 3.752764e-17)
                    Y: -0.009602876  cm    (Err: 0.07116063)
                    Z: -8.211695  cm    (Err: 0.2926996)

Moments of inertia: X: 170130.8  gm sq cm (Err: 7520.148)
                    Y: 170126.3  gm sq cm (Err: 7464.19)
                    Z: 14758.49  gm sq cm (Err: 384.9393)

Products of inertia: XY: 1.598289e-13  gm sq cm (Err: 1.106242e-13)
                     YZ: 287.1289  gm sq cm (Err: 1377.845)
                     ZX: 4.872571e-12  gm sq cm (Err: 6.472204e-13)

Radii of gyration:  X: 11.16799  cm
                    Y: 11.16784  cm
                    Z: 3.289308  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                    I: 14757.86 along [5.39438e-18 0.002832807 0.999996]
                    J: 78149.64 along [1 -3.920378e-14 1.056628e-16]
                    K: 78145.73 along [3.920392e-14 0.999996 -0.002832807]
```

10.1.10 Camera System Solid Modeling Information

The solid modeling information obtained from this model is:-

```
Ray projection along X axis, level of subdivision: 6.
Mass:           2085.438 gm
Volume:         769.5343 cu cm  (Err: 21.81147)

Bounding box:       X: 36.82107  --  57.22107 cm
                    Y: -15.5  --  5 cm
                    Z: -5.606249  --  5.393751 cm

Centroid:           X: 47.02107  cm    (Err: 1.332752)
                    Y: -6.578882  cm    (Err: 0.3145257)
                    Z: -0.1691733  cm    (Err: 0.1301494)

Moments of inertia: X: 213810.5  gm sq cm (Err: 9411.933)
                    Y: 4727513  gm sq cm (Err: 133300)
                    Z: 4884992  gm sq cm (Err: 138755.6)
Products of inertia: XY: -645122.1  gm sq cm (Err: 30842.24)
                     YZ: 1995.246  gm sq cm (Err: 2493.508)
                     ZX: -16589.05  gm sq cm (Err: 12762.39)

Radii of gyration:  X: 10.12549  cm
                    Y: 47.61214  cm
                    Z: 48.39866  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                    I: 183868.1 along [-4.31854e-16 0.004842232 0.9999883]
                    J: 123489.5 along [0 -0.9999883 0.004842232]
```

```
                    K: 123489.5 along [1 2.091137e-18 4.318489e-16]
```

10.1.11 Camera Platform Solid Modeling Information

The calculated mass properties of the camera platform are:-

```
Ray projection along X axis, level of subdivision: 6.
Mass:           1337.749 gm
Volume:         170.1971 cu cm  (Err: 5.151945)

Bounding box:         X: 32.94431  --  58 cm
                      Y: -10  --  10.25 cm
                      Z: -21.05569  --  21.05569 cm

Centroid:             X: 33.50344  cm    (Err: 1.38109)
                      Y: -1.454013  cm    (Err: 0.3139714)
                      Z: 0.02588385  cm    (Err: 0.2032814)

Moments of inertia:   X: 197934.9  gm sq cm (Err: 5043.741)
                      Y: 1666409  gm sq cm (Err: 86602.09)
                      Z: 1551143  gm sq cm (Err: 90319.35)
Products of inertia: XY: -59445.87  gm sq cm (Err: 18753.67)
                     YZ: 1082.342  gm sq cm (Err: 1658.574)
                     ZX: 2383.666  gm sq cm (Err: 9532.029)

Radii of gyration:    X: 12.16393  cm
                      Y: 35.2942  cm
                      Z: 34.05168  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                      I: 46695.62 along [0.008629623 0.01000698 0.9999127]
                      J: 196157.2 along [0.9837668 -0.1793265 -0.006695606]
                      K: 163780.5 along [0.1792438 0.9837387 -0.01139205]
```

10.1.12 Laser Canister Solid Modeling Information

The solid modeling data for the mounting device and the laser cylinder is included below:-

Laser Clamp Data-

```
Ray projection along X axis, level of subdivision: 6.
Mass:           1051.627 gm
Volume:         130.9622 cu cm  (Err: 2.436936)

Bounding box:         X: 32.2  --  47.2 cm
                      Y: -16.5  --  -7.5 cm
                      Z: 8  --  12 cm

Centroid:             X: 39.7  cm    (Err: 0.7387347)
                      Y: -8.87609  cm    (Err: 0.1498028)
                      Z: 10.0024  cm    (Err: 0.1894075)
```

```
Moments of inertia:   X: 191673.5  gm sq cm (Err: 3287.818)
                      Y: 1784237  gm sq cm (Err: 33189.76)
                      Z: 1762903  gm sq cm (Err: 32392.14)
Products of inertia: XY: -370573  gm sq cm (Err: 6254.203)
                     YZ: -93364.32  gm sq cm (Err: 1606.814)
                     ZX: 417595.9  gm sq cm (Err: 7907.684)

Radii of gyration:    X: 13.50051  cm
                      Y: 41.19035  cm
                      Z: 40.94335  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                      I: 3607.985 along [1 -3.609728e-15 4.59738e-15]
                      J: 22592.29 along [4.602266e-15 0.001354794 -0.9999991]
                      K: 21565.99 along [3.603496e-15 0.9999991 0.001354794]
```

Laser Cylinder Data -

```
Ray projection along X axis, level of subdivision: 6.
Mass:            2927.363 gm
Volume:          1125.909 cu cm  (Err: 29.18046)

Bounding box:         X: 29.7  --  49.7 cm
                      Y: -8.2  --  8.2 cm
                      Z: 1.8  --  18.2 cm

Centroid:             X: 40.99758  cm    (Err: 1.016304)
                      Y: -0.05958095  cm    (Err: 0.1648782)
                      Z: 10.05259  cm    (Err: 0.3102266)

Moments of inertia:   X: 449021.6  gm sq cm (Err: 15577.16)
                      Y: 5405951  gm sq cm (Err: 129300.2)
                      Z: 5108137  gm sq cm (Err: 121152.3)
Products of inertia: XY: -6882.716  gm sq cm (Err: 19179.99)
                     YZ: -1478.635  gm sq cm (Err: 5174.806)
                     ZX: 1206208  gm sq cm (Err: 35753.72)

Radii of gyration:    X: 12.38498  cm
                      Y: 42.97321  cm
                      Z: 41.77274  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                      I: 153184.2 along [0.9999476 0.007259344 -0.00721122]
                      J: 189850.2 along [0.008163024 -0.9908898 0.1344276]
                      K: 187774.4 along [-0.006169668 -0.1344795 -0.9908972]
```

10.1.13 Laser Mounts Solid Modeling Information

The mass properties for one of the laser mounts is:-

```
Ray projection along X axis, level of subdivision: 6.
Mass:                327.8413 gm
Volume:              40.82706 cu cm  (Err: 3.050958)

Bounding box:         X: 3.104413  --  22.10441 cm
                      Y: 0  --  6 cm
                      Z: -12  --  0 cm

Centroid:             X: 12.61116  cm    (Err: 0.9421433)
                      Y: 4.591362  cm    (Err: 0.3858374)
                      Z: -3.759856  cm    (Err: 0.4095259)

Moments of inertia:   X: 16889.82  gm sq cm (Err: 1829.325)
                      Y: 73585.24  gm sq cm (Err: 5429.323)
                      Z: 72351.74  gm sq cm (Err: 5277.15)
Products of inertia: XY: 18972.15  gm sq cm (Err: 1593.537)
                     YZ: -6738.723  gm sq cm (Err: 793.684)
                     ZX: -15542.52  gm sq cm (Err: 1692.248)

Radii of gyration:    X: 7.177623  cm
                      Y: 14.98178  cm
                      Z: 14.85568  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                      I: 5344.182 along [0.9999994 -0.0009698783 0.0004399428]
                      J: 17115.67 along [0.0008135124 0.9622437 0.2721883]
                      K: 12995.03 along [-0.0006873218 -0.2721878 0.9622439]
```

10.1.14 Manipulator Mount Solid Modeling Information

The solid modelling data for the manipulator mount is shown below:-

```
Ray projection along X axis, level of subdivision: 6.
Mass:                2229.886 gm
Volume:              283.7006 cu cm  (Err: 18.53383)

Bounding box:         X: 642.7574  --  660.7574 cm
                      Y: 177.7636  --  209.7636 cm
                      Z: 354.1964  --  373.05 cm

Centroid:             X: 653.8767  cm    (Err: 42.59973)
                      Y: 193.2574  cm    (Err: 12.61254)
                      Z: 370.4503  cm    (Err: 24.3646)

Moments of inertia:   X: 3.895286e+08  gm sq cm (Err: 25699903)
```

```
                    Y: 1.259526e+09  gm sq cm (Err: 82209409)
                    Z: 1.036948e+09  gm sq cm (Err: 67378865)
Products of inertia: XY: 2.817852e+08  gm sq cm (Err: 18339606)
                   YZ: 1.596403e+08  gm sq cm (Err: 10489234)
                   ZX: 5.401293e+08  gm sq cm (Err: 35427913)

Radii of gyration:    X: 417.9539  cm
                    Y: 751.5577  cm
                    Z: 681.926  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                    I: 113088.7 along [0.02152494 0.9996278 -0.01676025]
                    J: 271849.9 along [0.3123611 0.009200751 0.9499189]
                    K: 226622.2 along [0.9497196 -0.0256822 -0.3120468]
```

10.1.15 Internal Frame Mounts Solid Modeling Information

The design of the internal frame mounts is shown in Figure 88.

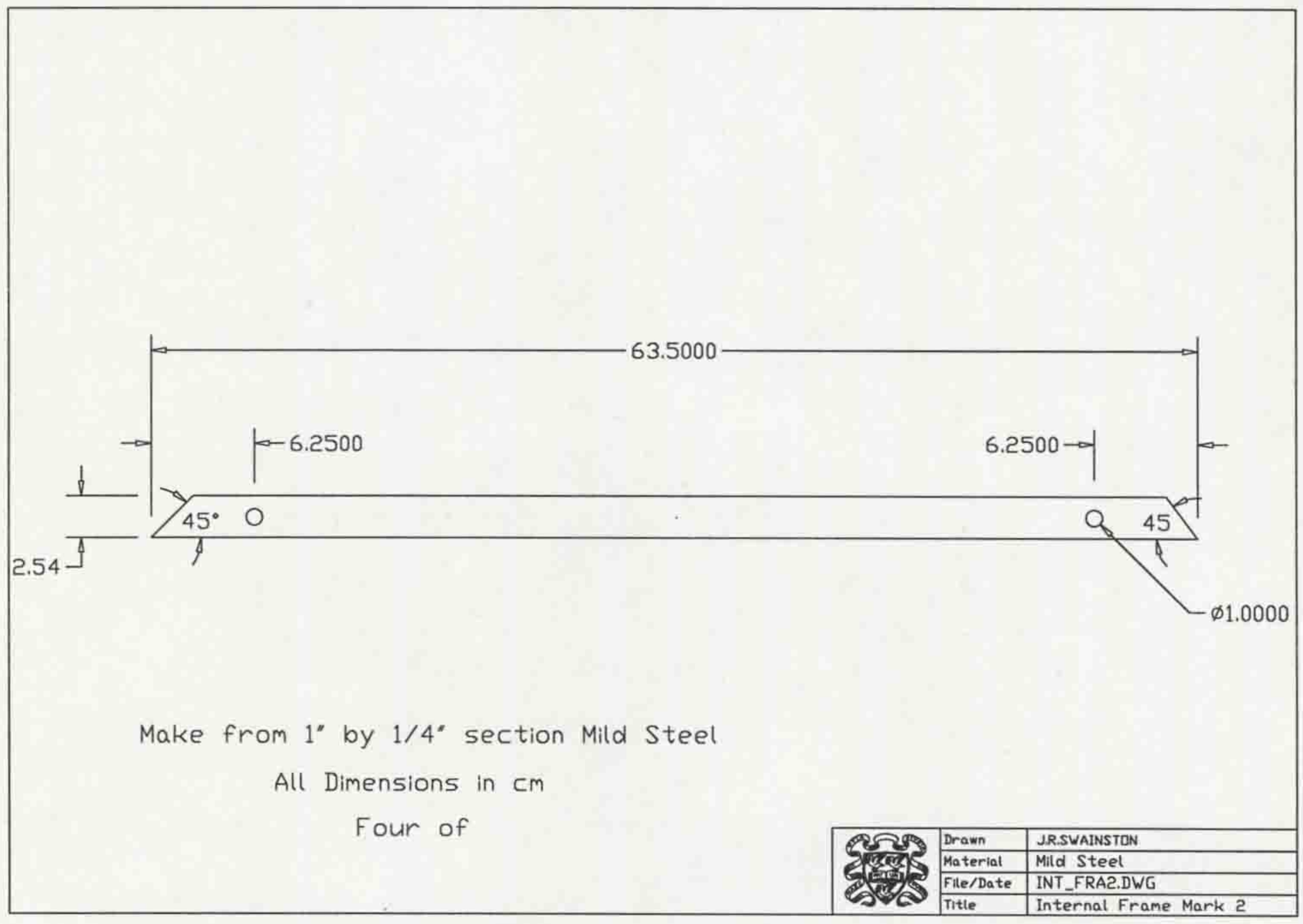

Figure 88, Additional Frame Supports

The solid modeling information for the new frame support pieces is-

```
Ray projection along X axis, level of subdivision: 6.
Mass:           770.3121 gm
Volume:         98.00409 cu cm  (Err: 0.04443989)

Bounding box:         X: 0  --  63.5 cm
                      Y: 0  --  2.54 cm
                      Z: 0  --  0.635 cm

Centroid:             X: 31.90807  cm    (Err: 0.0134373)
                      Y: 1.254456  cm    (Err: 0.01112362)
                      Z: 0.3180467  cm    (Err: 0.002885804)

Moments of inertia:   X: 1732.702  gm sq cm (Err: 22.63126)
                      Y: 1023669  gm sq cm (Err: 553.1149)
                      Z: 1025194  gm sq cm (Err: 538.1632)
Products of inertia: XY: 30885.63  gm sq cm (Err: 274.7931)
                     YZ: 307.3913  gm sq cm (Err: 4.198428)
                     ZX: 7817.263  gm sq cm (Err: 70.91221)

Radii of gyration:    X: 1.499783  cm
                      Y: 36.45411  cm
                      Z: 36.48125  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                      I: 442.5612 along [1 0.0002182469 -2.603556e-07]
                      J: 239708 along [-2.913992e-07 0.0001422386 -1]
                      K: 239317.2 along [-0.0002182469 1 0.0001422387]
```

10.1.16 Internal Frame Solid Modeling Information

.

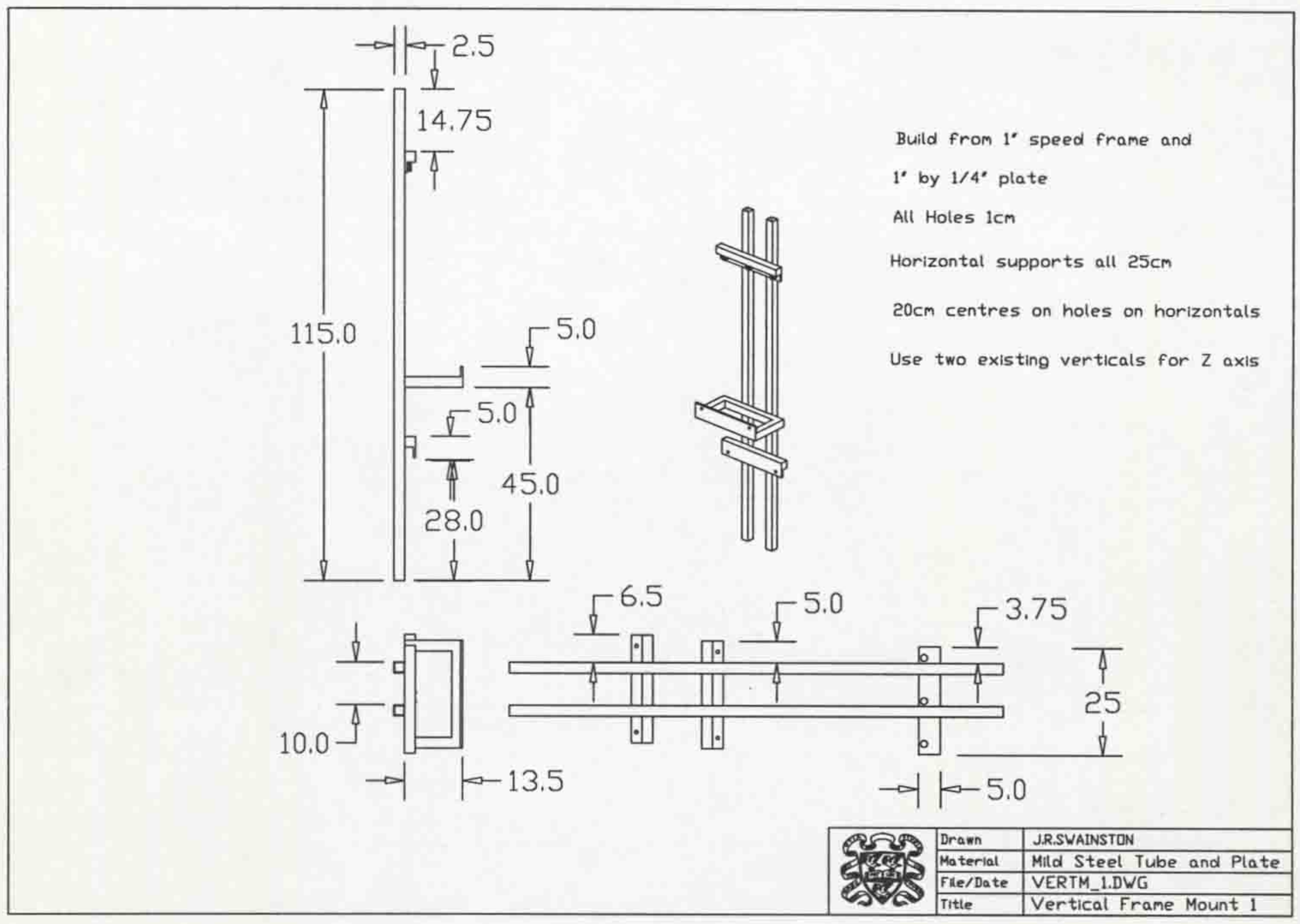

Figure 89, Internal Frame Support 1

```
Ray projection along X axis, level of subdivision: 3.
Mass:            8113.731 gm
Volume:          1032.281 cu cm  (Err: 288.1232)

Bounding box:          X: 27.72325  --  43.72325 cm
                       Y: -92.701  --  -65.201 cm
                       Z: 342.75  --  458.25 cm

Centroid:              X: 32.12645  cm    (Err: 9.55575)
                       Y: -77.73085  cm    (Err: 21.34108)
                       Z: 408.3206  cm    (Err: 113.2679)

Moments of inertia:    X: 1.408612e+09  gm sq cm (Err: 3.900814e+08)
                       Y: 1.367998e+09  gm sq cm (Err: 3.796644e+08)
                       Z: 57700001  gm sq cm (Err: 15944096)
Products of inertia: XY: -20235815  gm sq cm (Err: 5898681)
                     YZ: -2.573522e+08  gm sq cm (Err: 70158887)
                     ZX: 1.060158e+08  gm sq cm (Err: 31092390)

Radii of gyration:     X: 416.6634  cm
                       Y: 410.6128  cm
                       Z: 84.32913  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                       I: 270752.7 along [0.06374717 -0.02555777 -0.9976388]
                       J: 6891192 along [0.6391691 -0.7666842 0.06048278]
```

K: 6816525 along [-0.7664197 -0.6415155 -0.03253821]

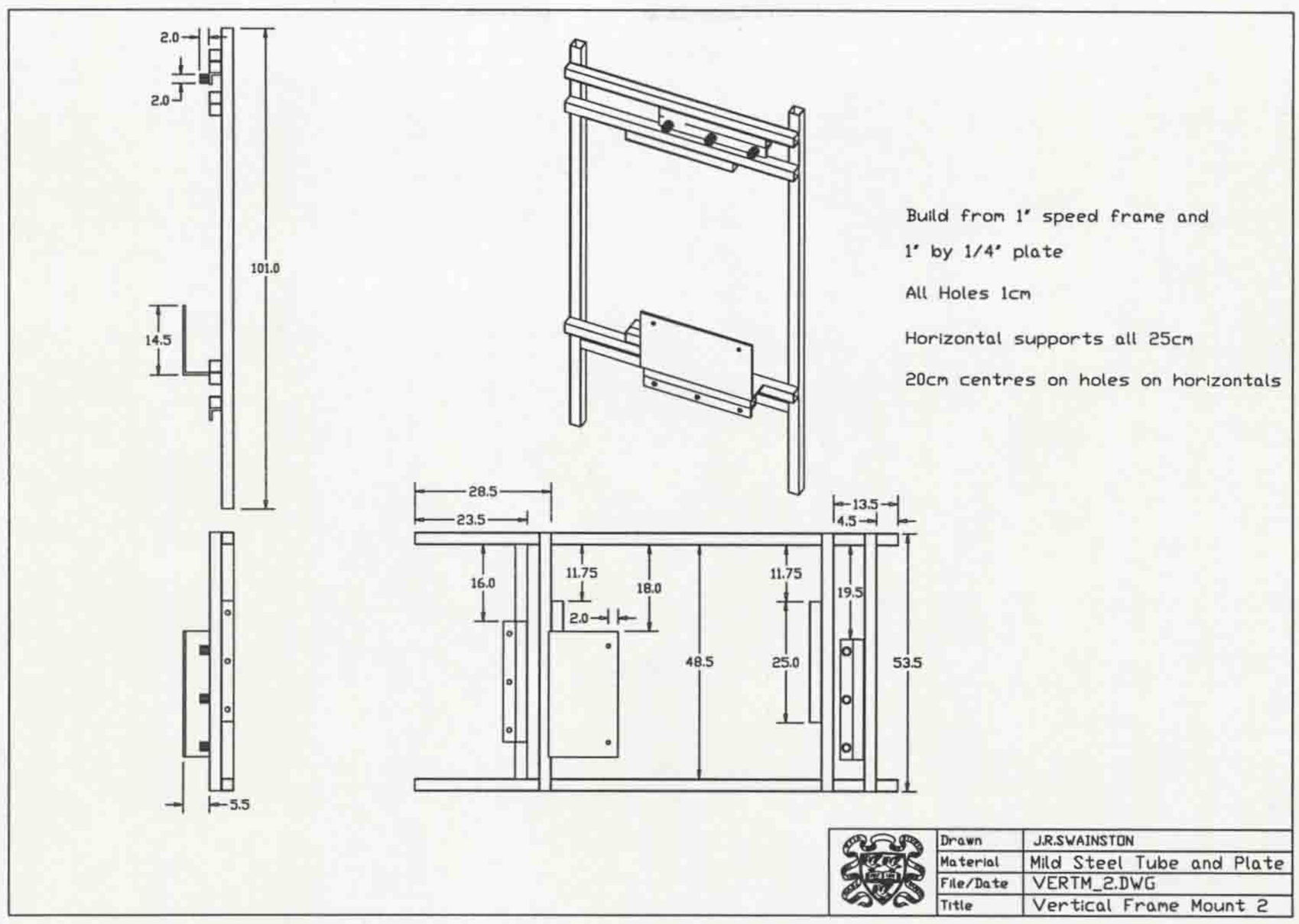

Figure 90, Internal Frame Support 2

The solid modeling data for the internal frame support two is shown below:-

```
Ray projection along X axis, level of subdivision: 7.
Mass:            9008.381 gm
Volume:          1146.104 cu cm  (Err: 22.92603)

Bounding box:          X: -5  --  5.5 cm
                       Y: 409.0782  --  510.0782 cm
                       Z: -8.891759e-14  --  53.5 cm

Centroid:              X: -1.080447  cm    (Err: 0.07121801)
                       Y: 453.384  cm    (Err: 8.954037)
                       Z: 23.90618  cm    (Err: 0.5667984)

Moments of inertia:    X: 1.865923e+09  gm sq cm (Err: 36562457)
                       Y: 7099436  gm sq cm (Err: 198648.3)
                       Z: 1.859046e+09  gm sq cm (Err: 36439516)
Products of inertia: XY: -4426250  gm sq cm (Err: 284115)
                     YZ: 97580634  gm sq cm (Err: 2288833)
                     ZX: -294396.2  gm sq cm (Err: 17970.07)

Radii of gyration:     X: 455.1176  cm
                       Y: 28.07298  cm
                       Z: 454.278  cm
```

```
Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                    I: 1939922 along [0.001796948 -0.9999394 0.0108597]
                    J: 9042062 along [0.999368 0.002181239 0.03547932]
                    K: 7298535 along [-0.03550086 0.01078909 0.9993114]
```

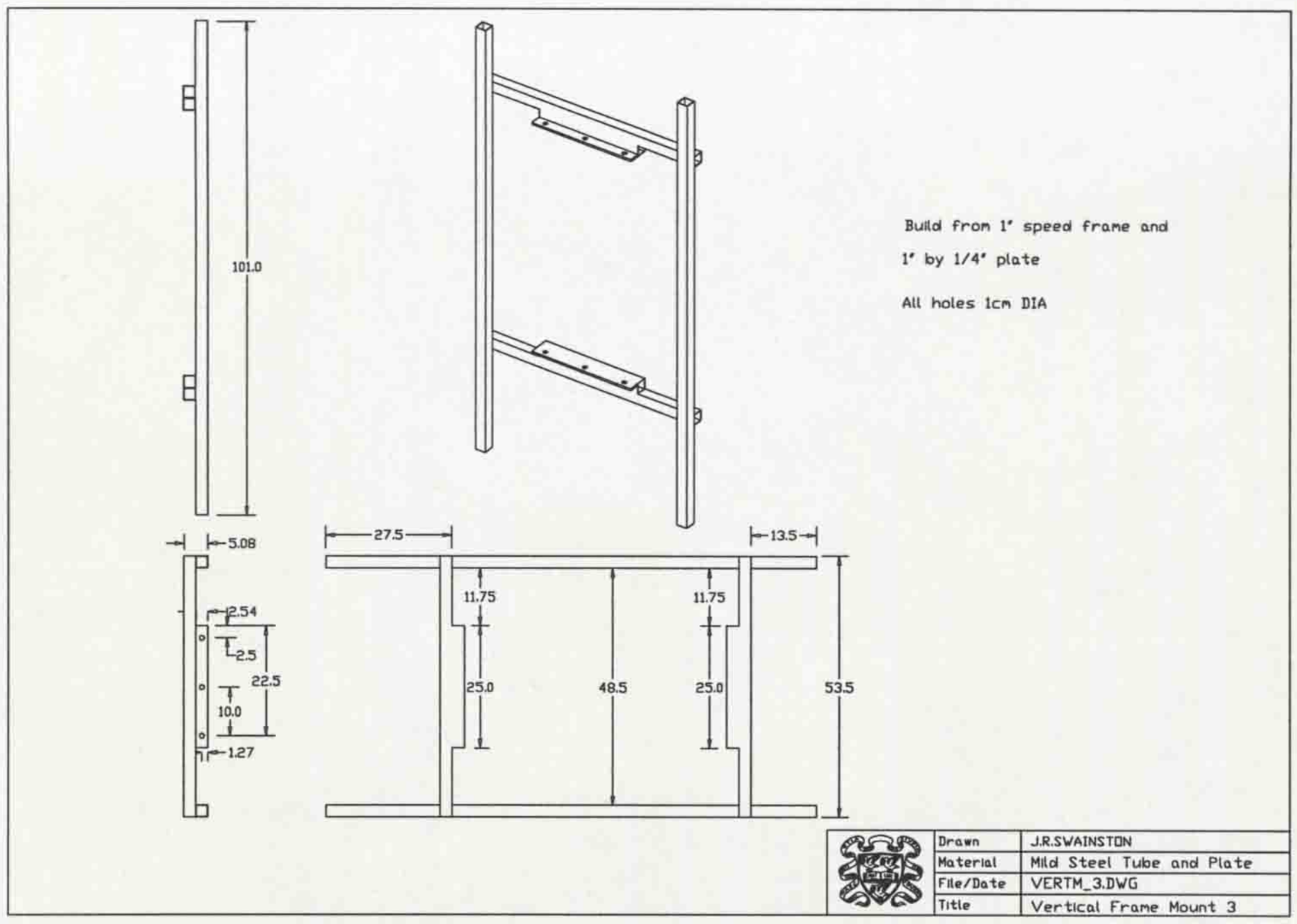

Figure 91, Internal Frame Support 3

The modelling data for the third internal frame support is shown below:-

```
Ray projection along X axis, level of subdivision: 7.
Mass:              3043.973 gm
Volume:            387.2739 cu cm  (Err: 12.80273)

Bounding box:            X: -4.915936e-15  --  5 cm
                         Y: -1.314053e-15  --  53.5 cm
                         Z: 0  --  101 cm

Centroid:                X: 2.794076  cm    (Err: 0.1019524)
                         Y: 25.23636  cm    (Err: 1.020235)
                         Z: 55.12376  cm    (Err: 2.204763)

Moments of inertia:      X: 14931055  gm sq cm (Err: 632238.3)
                         Y: 11905845  gm sq cm (Err: 561230.6)
                         Z: 3087292  gm sq cm (Err: 134261.7)
Products of inertia:    XY: 210495.4  gm sq cm (Err: 9674.844)
                        YZ: 4200761  gm sq cm (Err: 202378)
                        ZX: 460911.9  gm sq cm (Err: 20096.45)

Radii of gyration:       X: 70.03656  cm
```

```
                    Y: 62.54026  cm
                    Z: 31.84699  cm

Principal moments(gm sq cm) and X-Y-Z directions about centroid:
                    I: 1124121 along [0.002988416 0.02238012 -0.9997451]
                    J: 3742962 along [0.999988 0.003823876 0.003074743]
                    K: 2633314 along [0.003891714 -0.9997422 -0.02236843]
```

10.2 Aquasphere Assembly Details

To assemble the Aquasphere for use the following steps must be taken:-

1. Add Vertical Frame Piece 1 (Figure 89, Page 179) to the basic frame (Figure 92, The Basic Frame, Page 184) as in Figure 93, Vertical Frame 1 Inserted, on page 184.
2. Add Vertical Frame Piece 2 (Figure 90, Page 180) to the frame as in Figure 94, Vertical Frame 2 Inserted, on page 185.
3. Add Vertical Frame Piece 3 (Figure 91, Page 181) to the frame as in Figure 95, Vertical Frame 3 Inserted, on page 185.
4. Add the Laser mounts (**Figure 23**, Page 38) to the vehicle as in Figure 96, Laser Brackets Mounted, on page 186.
5. Add Networking box (**Figure 16**, Page 31) to the base of the vehicle as shown in Figure 97, Networking Box Installed, on page 186.
6. Add the Sensor Box (**Figure 17**, Page 32) to the middle of the vehicle as shown in Figure 98, Sensor Box Installed, on page 187.
7. Add the Vision System box to the top of the internal frame as shown in Figure 99, Vision Box Installed, on page 187.
8. Add the thruster box (**Figure 18, Thruster Control Box**, Page 33) to the vehicle's frame as in Figure 100, Thruster Box Installed, on page 188.
9. Add the manipulator box (**Figure 18, Thruster Control Box**, Page 33) to the vehicle's frame as in Figure 101, Manipulator Box Installed, on page 188.
10. Add the camera system (**Figure 20, The Pan-ant-tilt Mechanism**, Page 35) to the vehicle's frame as in Figure 102, Camera System Installed, on page 189.
11. Add the laser system to the vehicle's frame as in Figure 103, Laser Rangefinders Installed, on page 189.
12. Add the thrusters to the vehicle as shown in Figure 104, Thrusters Installed, on page 190.

13. Add the manipulator arm to the vehicle as shown in Figure 105, The Manipulator Arm Installed, on page 190.
14. Add the vehicles buoyancy.
15. Connect all the interconnecting cables.
16. Add the umbilical cables
17. Attach the lifting harness.

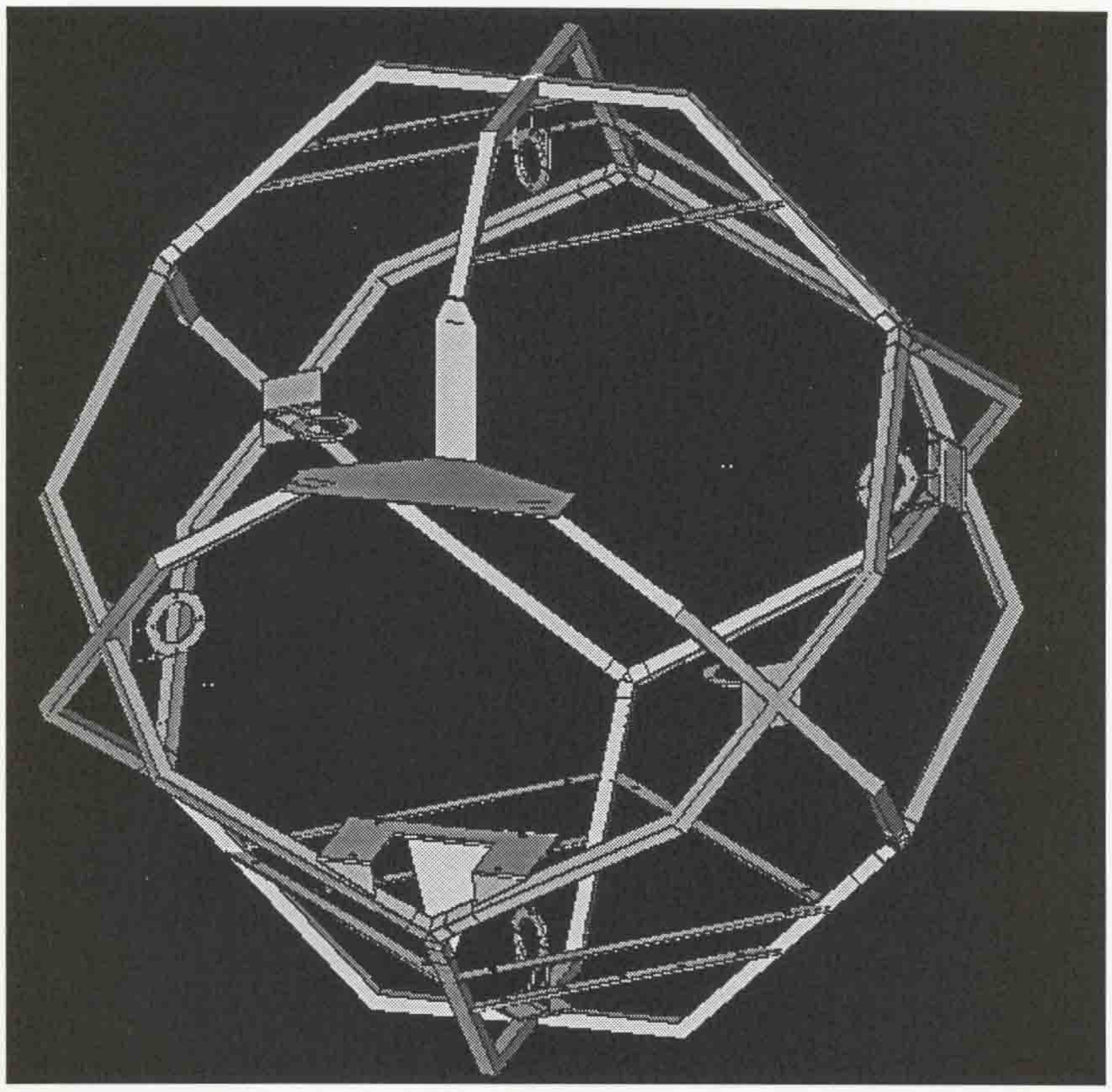

Figure 92, The Basic Frame

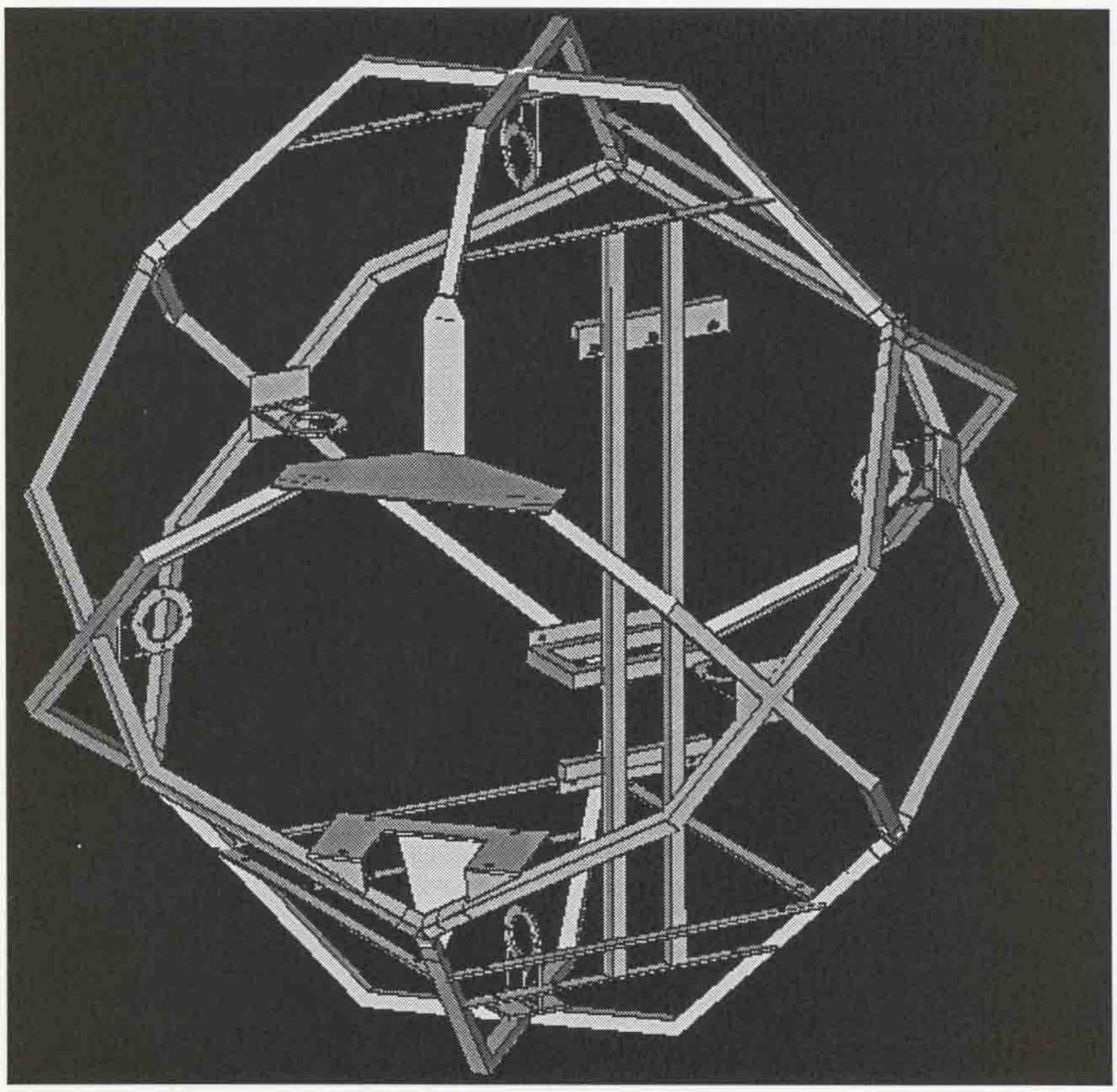

Figure 93, Vertical Frame 1 Inserted

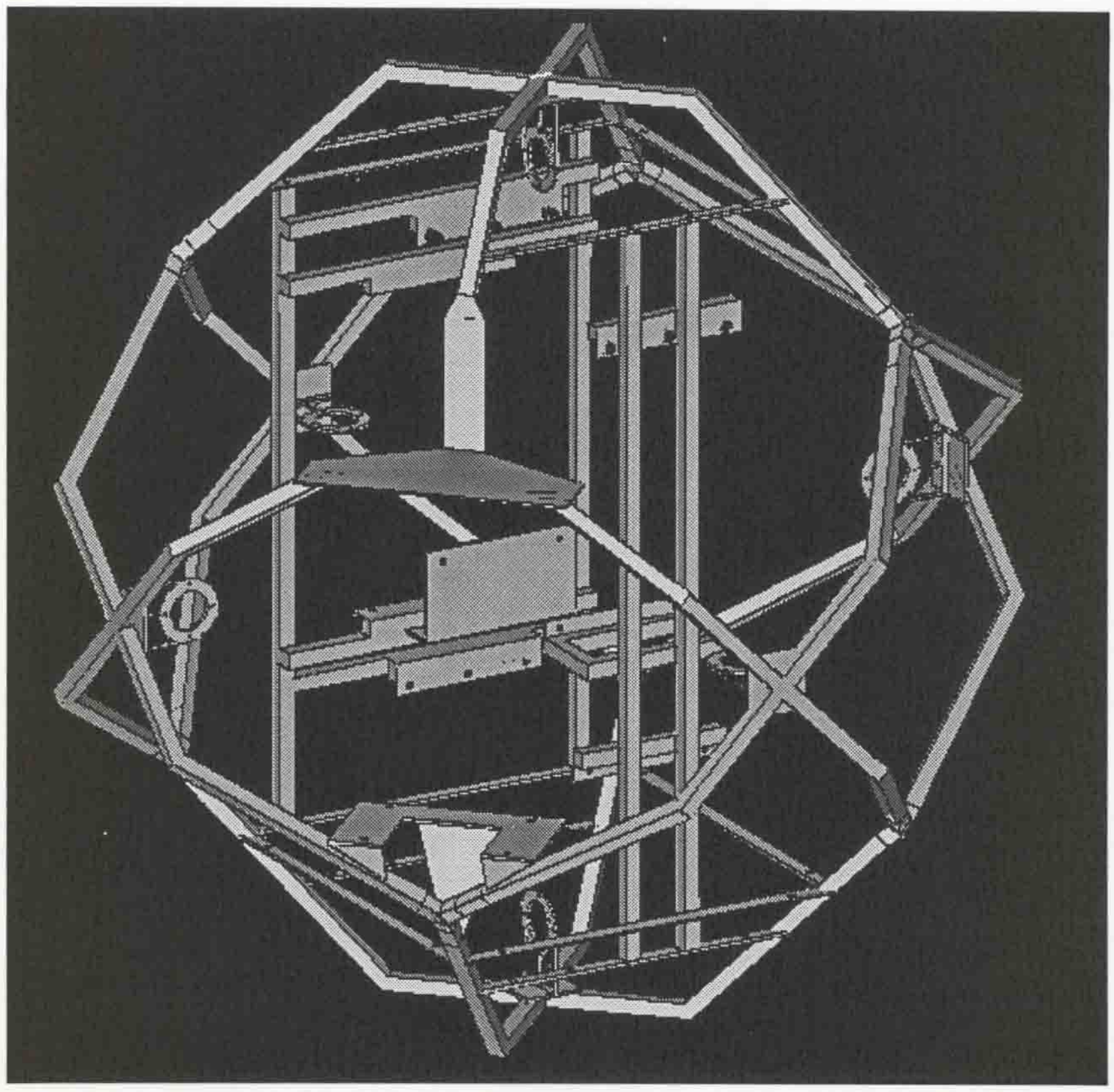

Figure 94, Vertical Frame 2 Inserted

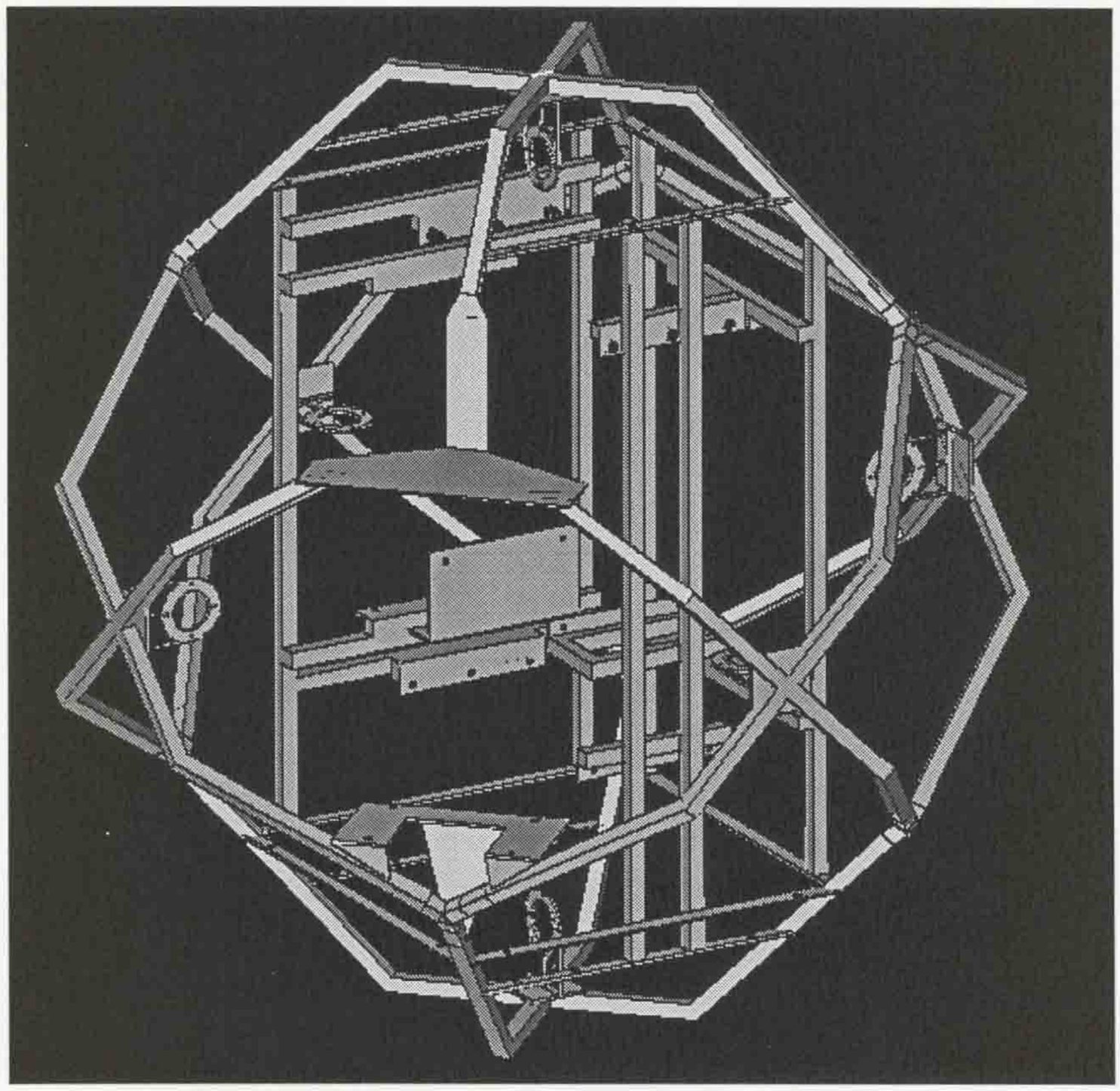

Figure 95, Vertical Frame 3 Inserted

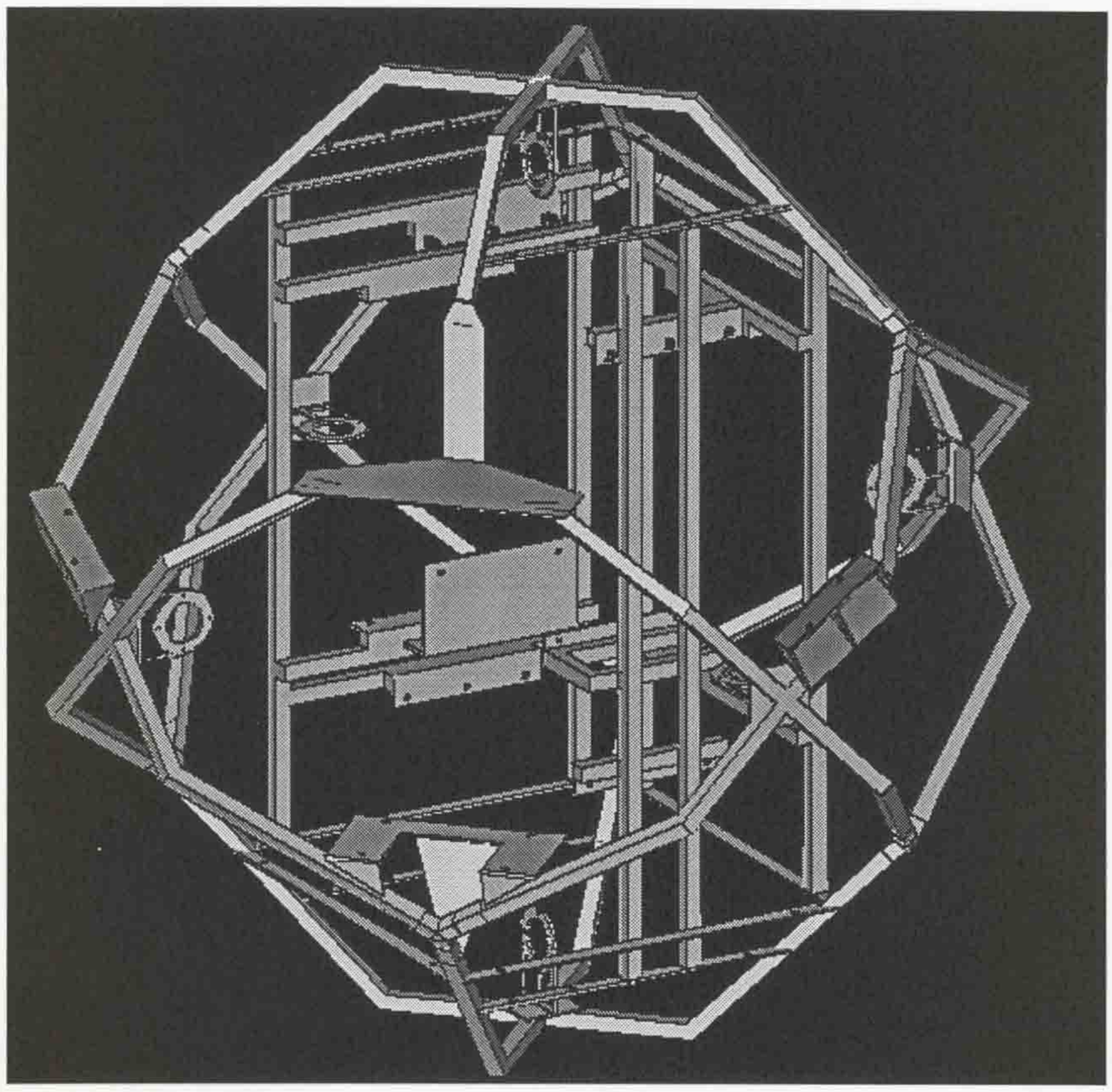

Figure 96, Laser Brackets Mounted

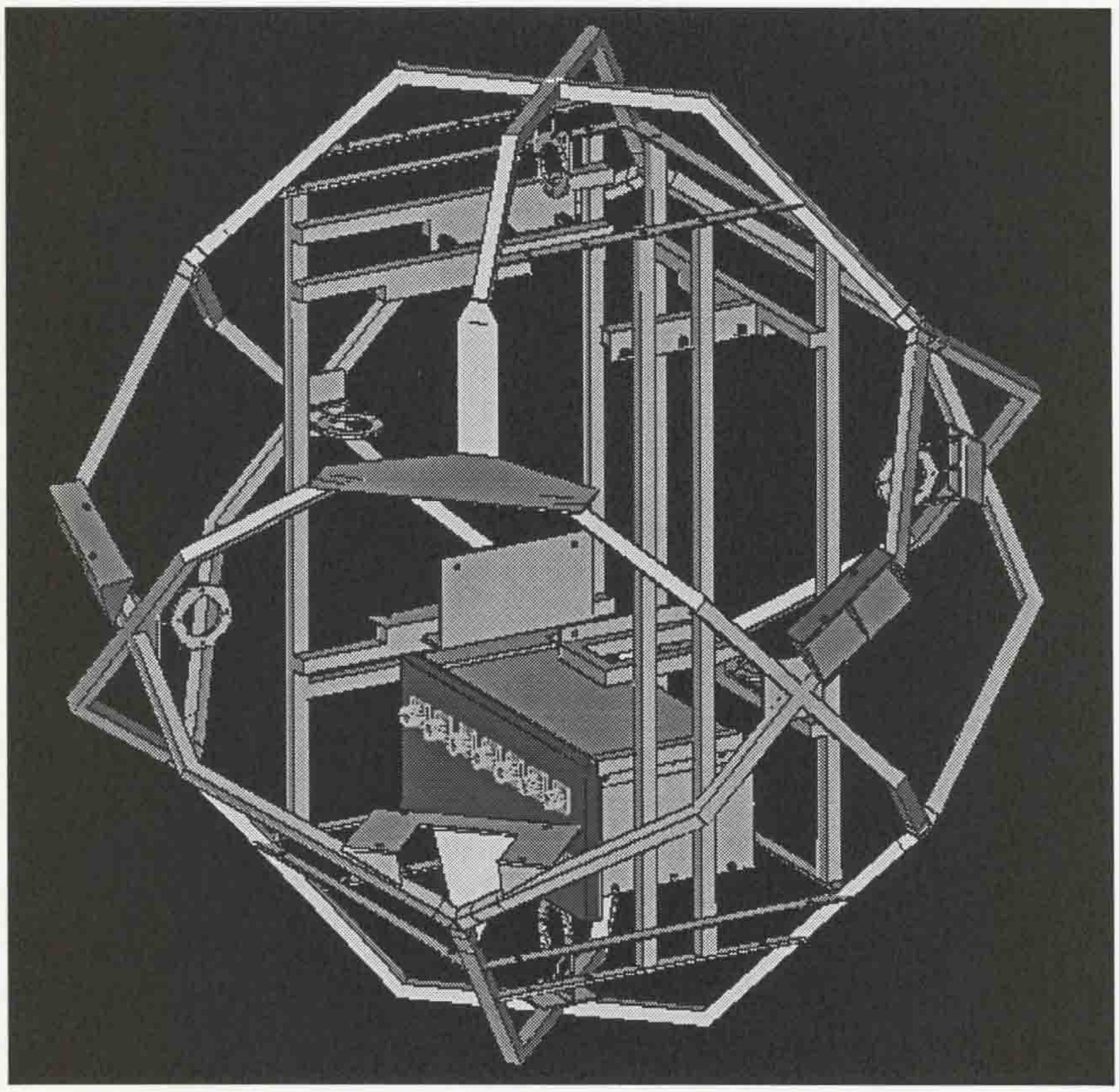

Figure 97, Networking Box Installed

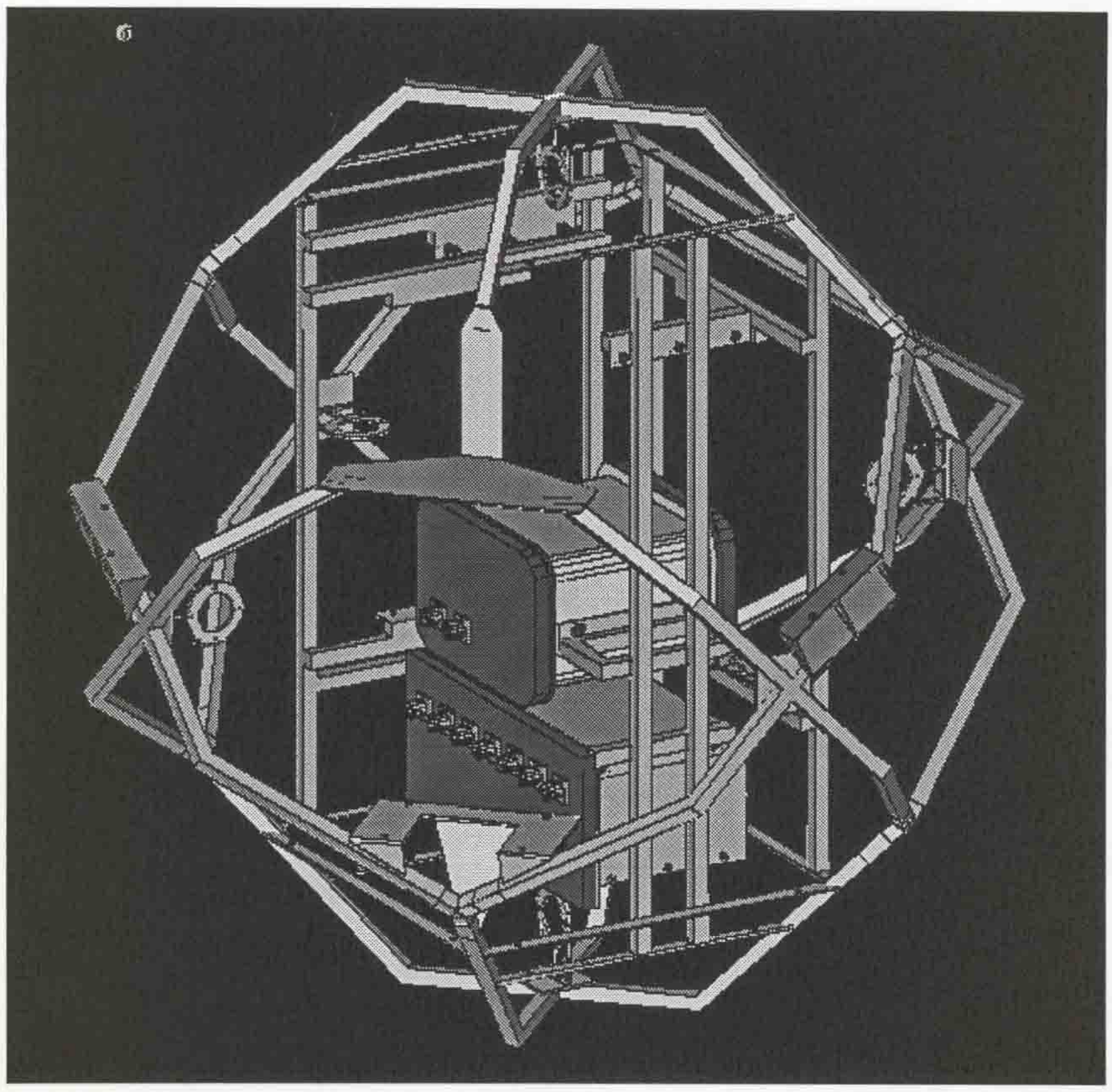

Figure 98, Sensor Box Installed

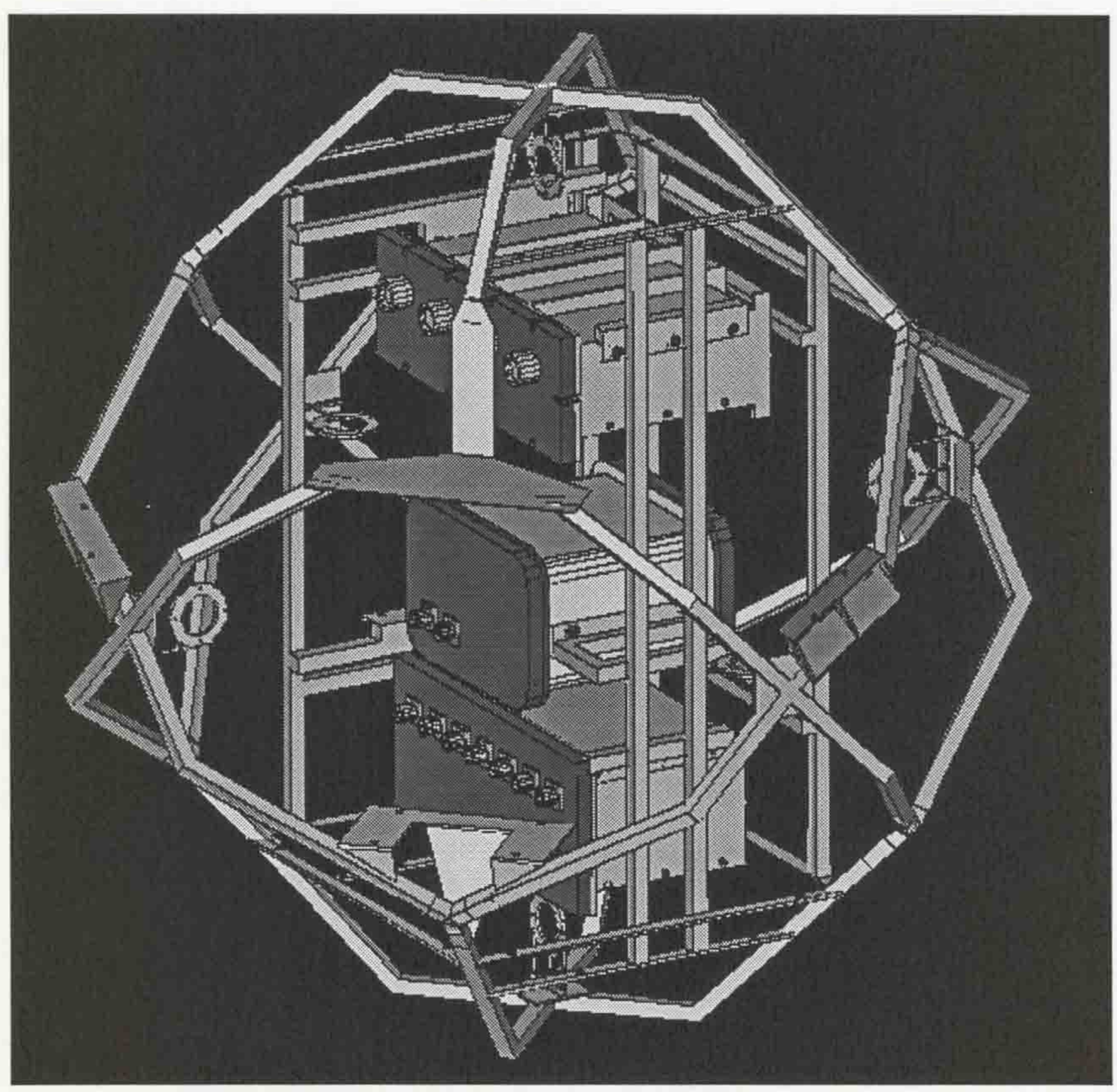

Figure 99, Vision Box Installed

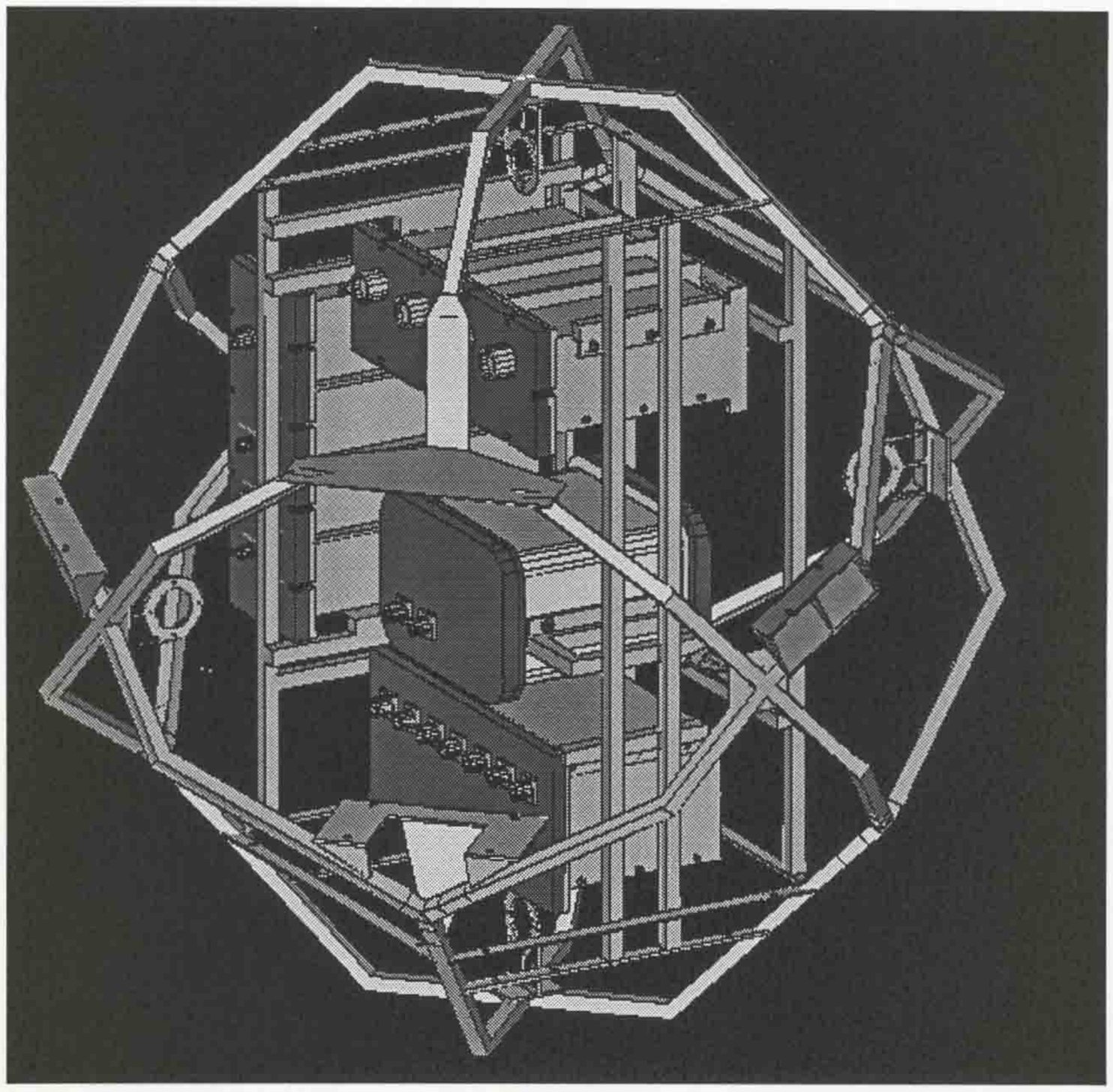

Figure 100, Thruster Box Installed

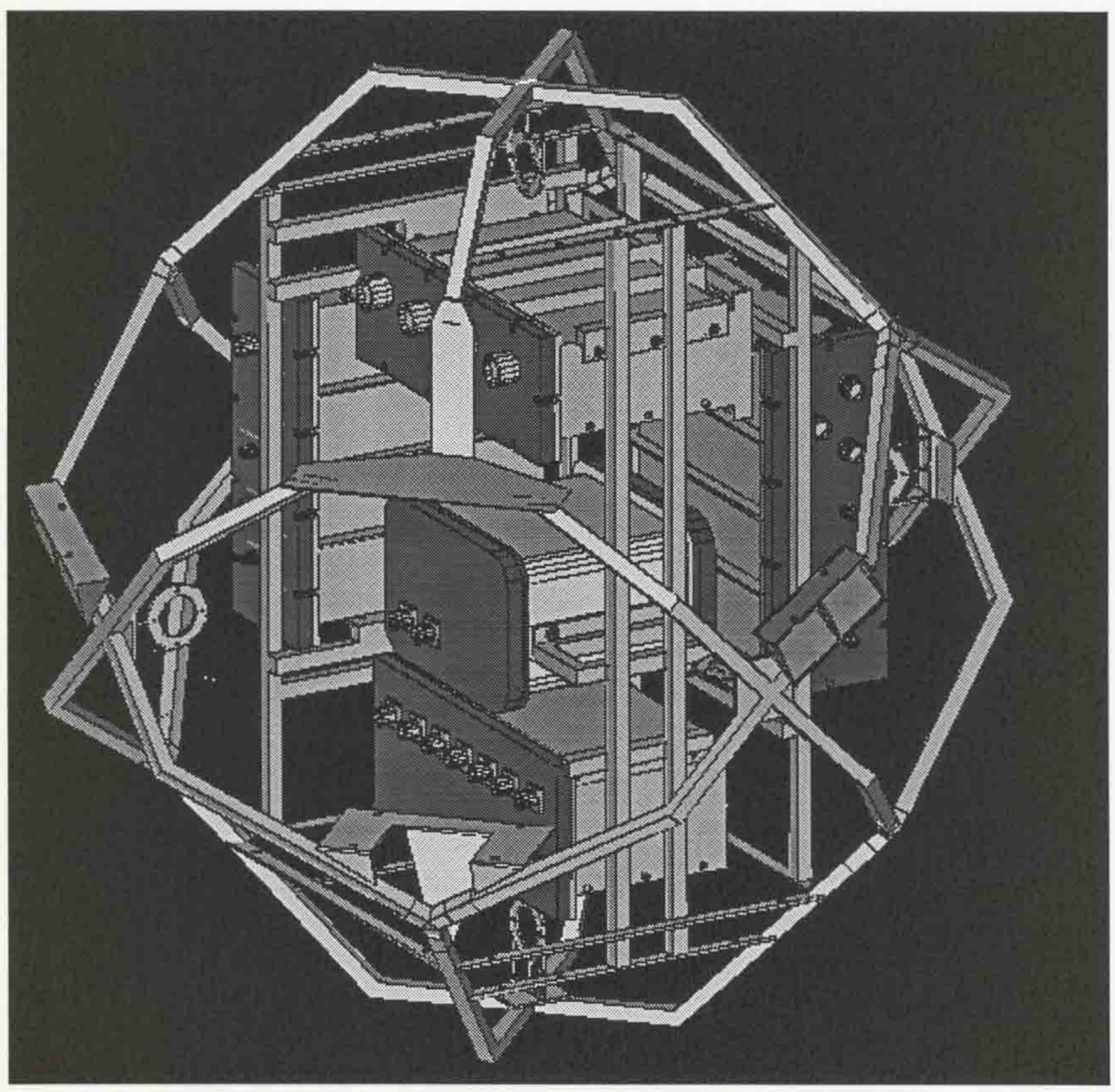

Figure 101, Manipulator Box Installed

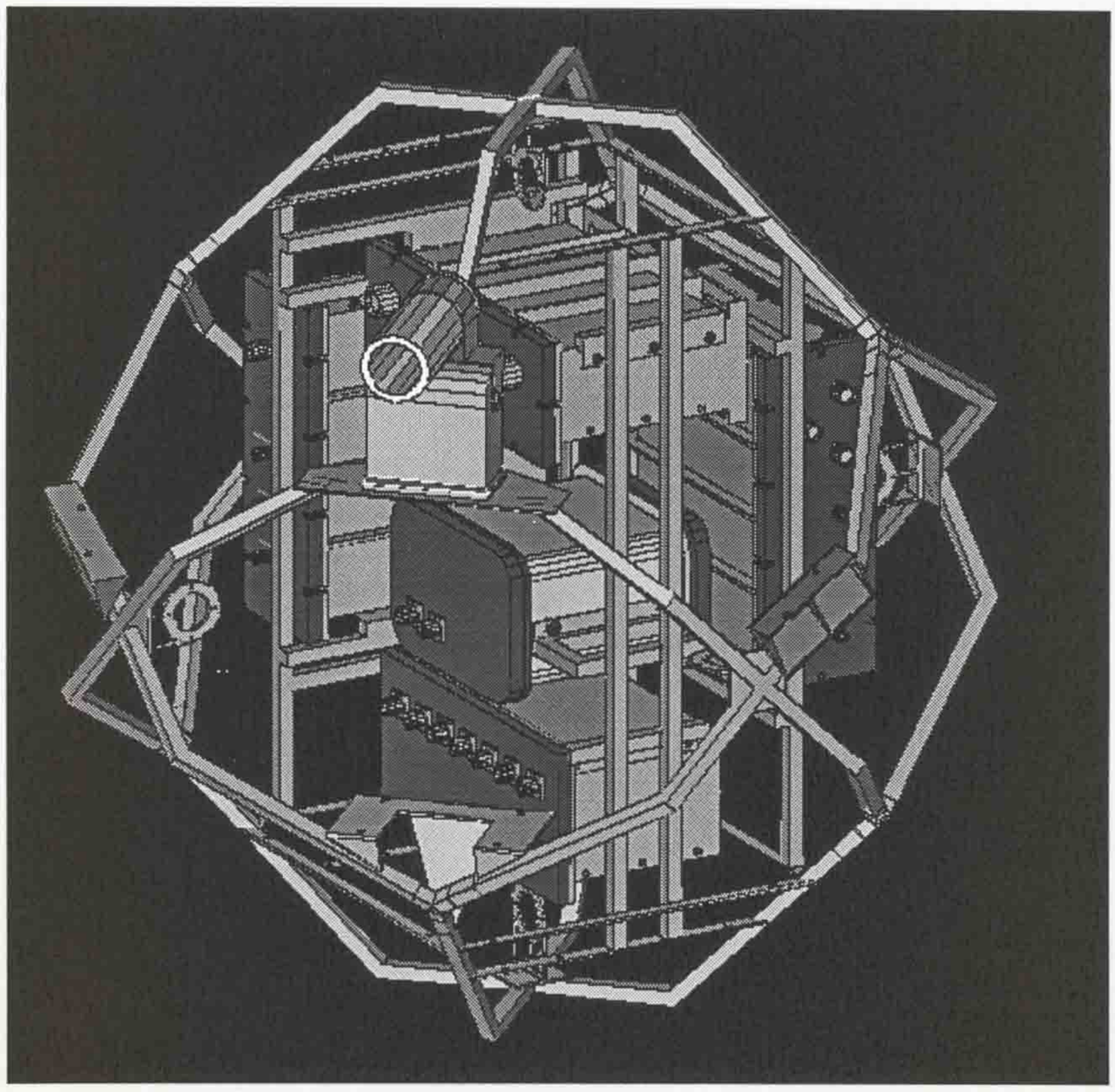

Figure 102, Camera System Installed

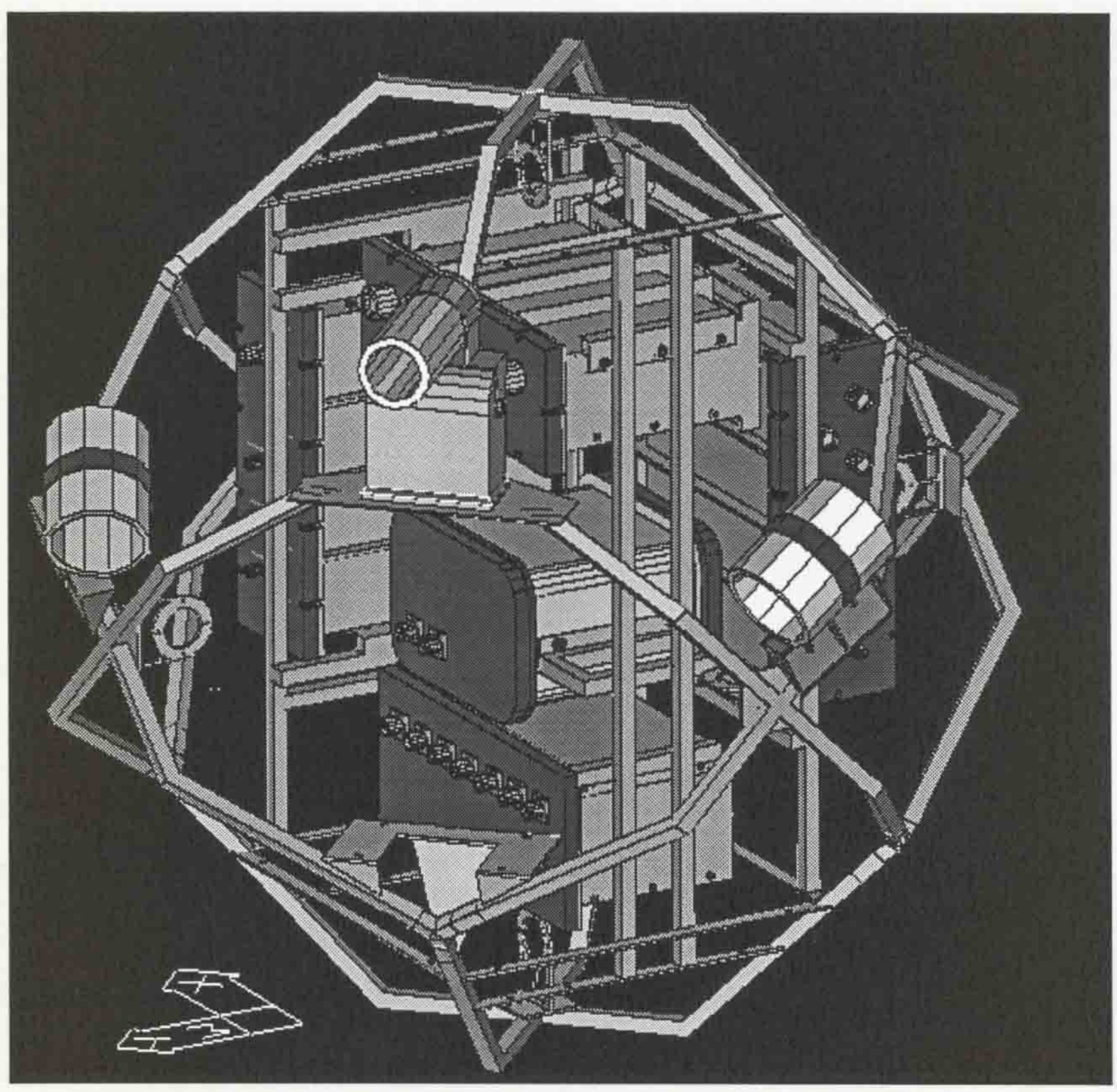

Figure 103, Laser Rangefinders Installed

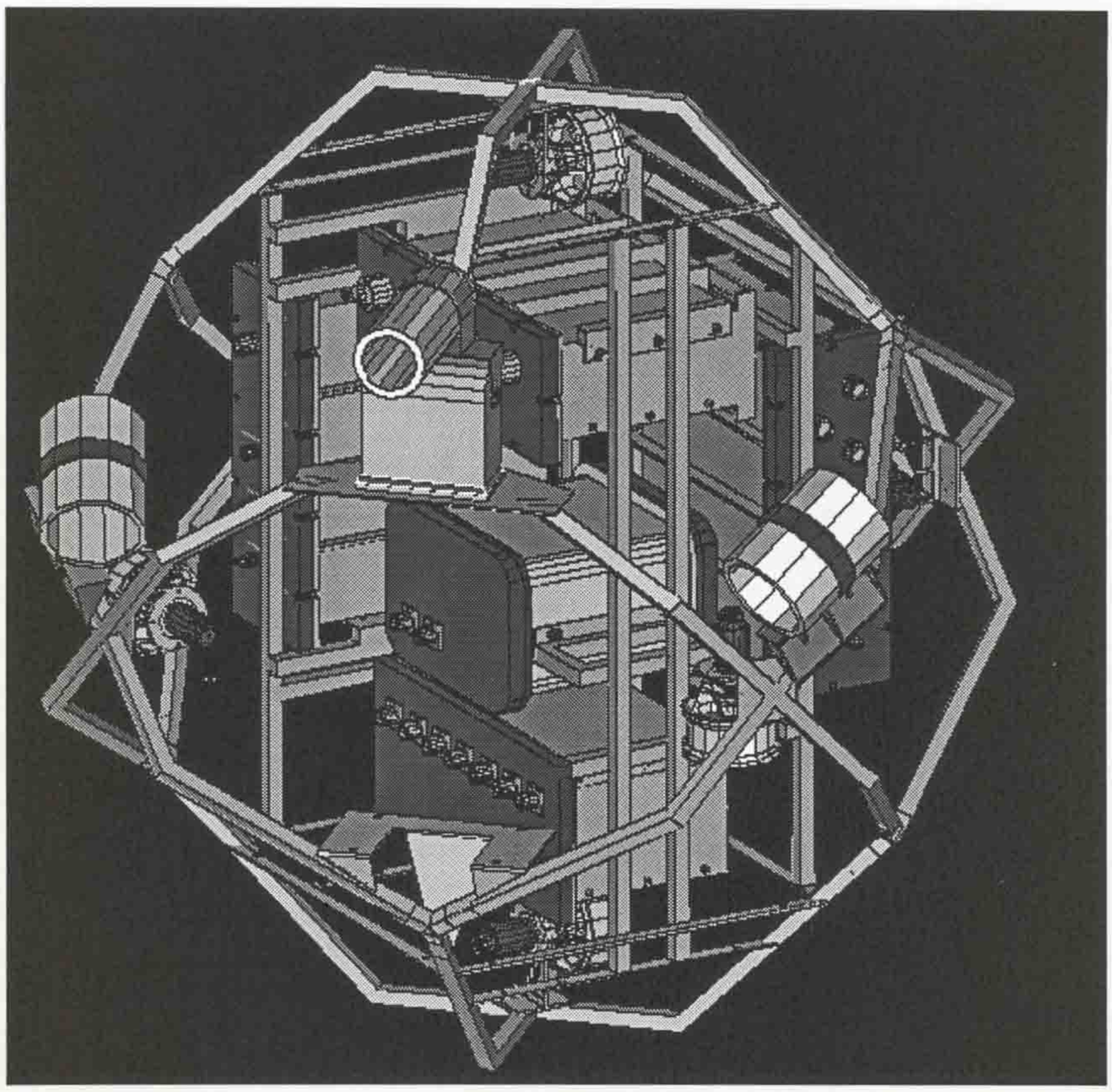

Figure 104, Thrusters Installed

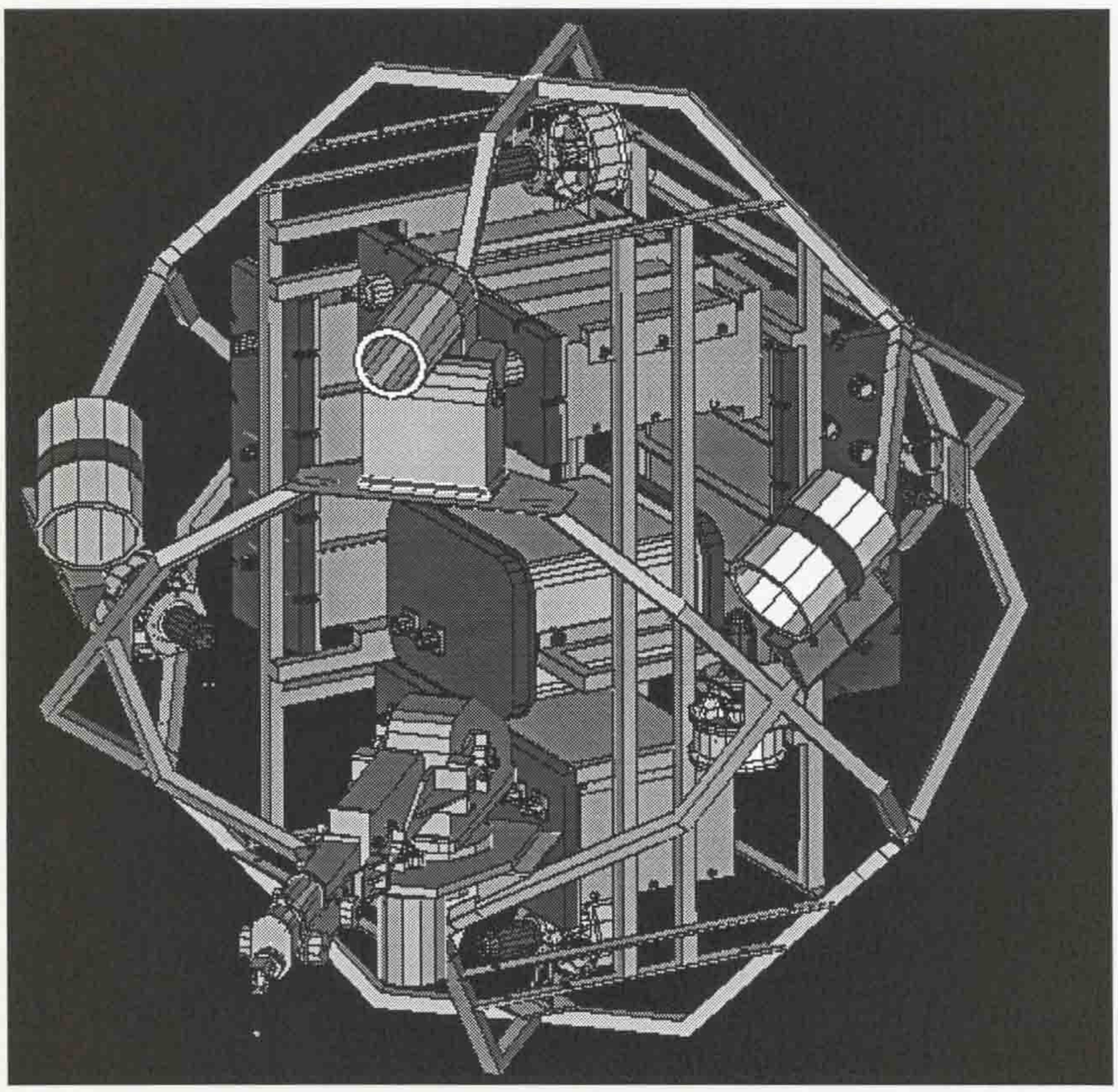

Figure 105, The Manipulator Arm Installed

10.3 Simulink Thruster Model

The following listing is the listing for the thruster model created in Matlab. In order to be able to use the listing the user must have the computer package Matlab with the Simulink toolbox installed.

```
function [ret,x0,str,ts,xts]=tmodel(t,x,u,flag);
%TMODEL is the M-file description of the SIMULINK system named TMODEL.
%       TMODEL has a the following characteristics:
%               1 continuous states
%               31 discrete states
%               0 outputs
%               0 inputs
%               does not have direct feed-through
%               1 sample times
%
%       The block-diagram can be displayed by typing: TMODEL.
%
%       SYS=TMODEL(T,X,U,FLAG) returns depending on FLAG certain
%       system values given time point, T, current state vector, X,
%       and input vector, U.
%       FLAG is used to indicate the type of output to be returned in SYS.
%
%       Setting FLAG=1 causes TMODEL to return state derivatives, FLAG=2
%       discrete states, FLAG=3 system outputs and FLAG=4 next sample
%       time. For more information and other options see SFUNC.
%
%       Calling TMODEL with a FLAG of zero:
%       [SIZES]=TMODEL([],[],[],0),  returns a vector, SIZES, which
%       contains the sizes of the state vector and other parameters.
%               SIZES(1) number of states
%               SIZES(2) number of discrete states
%               SIZES(3) number of outputs
%               SIZES(4) number of inputs
%               SIZES(5) number of roots (currently unsupported)
%               SIZES(6) direct feedthrough flag
%               SIZES(7) number of sample times
%
%       For the definition of other parameters in SIZES, see SFUNC.
%       See also, TRIM, LINMOD, LINSIM, EULER, RK23, RK45, ADAMS, GEAR.

% Note: This M-file is only used for saving graphical information;
%        after the model is loaded into memory an internal model
%        representation is used.

% the system will take on the name of this mfile:
sys = mfilename;
new_system(sys)
simver(1.3)
if (0 == (nargin + nargout))
     set_param(sys,'Location',[77,92,577,393])
     open_system(sys)
end;
set_param(sys,'algorithm',     'RK-45')
set_param(sys,'Start time',    '0.0')
```

```
set_param(sys,'Stop time',     '3')
set_param(sys,'Min step size', '0.001')
set_param(sys,'Max step size', '.01')
set_param(sys,'Relative error','1e-3')
set_param(sys,'Return vars',   '')


%     Subsystem  ['Power Drive',13,''].

new_system([sys,'/',['Power Drive',13,'']])
set_param([sys,'/',['Power Drive',13,'']],'Location',[60,94,325,249])

add_block('built-in/Inport',[sys,'/',['Power Drive',13,'/in_1']])
set_param([sys,'/',['Power Drive',13,'/in_1']],...
               'position',[15,55,35,75])

add_block('built-in/Outport',[sys,'/',['Power Drive',13,'/out_1']])
set_param([sys,'/',['Power Drive',13,'/out_1']],...
               'position',[220,60,240,80])

add_block('built-in/Product',[sys,'/',['Power Drive',13,'/Power Amp']])
set_param([sys,'/',['Power Drive',13,'/Power Amp']],...
               'position',[145,58,175,82])

add_block('built-in/Gain',[sys,'/',['Power Drive',13,'/Gain']])
set_param([sys,'/',['Power Drive',13,'/Gain']],...
               'Gain','0.1',...
               'position',[70,51,100,79])

add_block('built-in/Constant',[sys,'/',['Power Drive',13,'/Power Supply',13,'']])
set_param([sys,'/',['Power Drive',13,'/Power Supply',13,'']],...
               'orientation',3,...
               'Value','25',...
               'position',[110,120,130,140])
add_line([sys,'/',['Power Drive',13,'']],[120,115;120,75;140,75])
add_line([sys,'/',['Power Drive',13,'']],[40,65;65,65])
add_line([sys,'/',['Power Drive',13,'']],[105,65;140,65])
add_line([sys,'/',['Power Drive',13,'']],[180,70;215,70])


%     Finished composite block ['Power Drive',13,''].

set_param([sys,'/',['Power Drive',13,'']],...
               'position',[145,15,175,65])


%     Subsystem  ['Demand',13,'graph',13,''].

new_system([sys,'/',['Demand',13,'graph',13,'']])
set_param([sys,'/',['Demand',13,'graph',13,'']],'Location',[0,59,274,252])

add_block('built-in/Inport',[sys,'/',['Demand',13,'graph',13,'/x']])
set_param([sys,'/',['Demand',13,'graph',13,'/x']],...
               'position',[65,55,85,75])

add_block('built-in/S-Function',[sys,'/',['Demand',13,'graph',13,'/S-function',13,'M-
file which plots',13,'lines',13,'']])
```

```
set_param([sys,'/',['Demand',13,'graph',13,'/S-function',13,'M-file which
plots',13,'lines',13,'']],...
                'function name','sfunyst',...
                'parameters','ax, color, npts, dt',...
                'position',[130,55,180,75])
add_line([sys,'/',['Demand',13,'graph',13,'']],[90,65;125,65])
set_param([sys,'/',['Demand',13,'graph',13,'']],...
                'Mask
Display','plot(0,0,100,100,[83,76,63,52,42,38,28,16,11,84,11,11,11,90,90,11],[75,58,47
,54,72,80,84,74,65,65,65,90,40,40,90,90])',...
                'Mask Type','Storage scope.')
set_param([sys,'/',['Demand',13,'graph',13,'']],...
                'Mask Dialogue','Storage scope using MATLAB graph window.\nEnter
plotting ranges and line type.|Initial Time Range:|Initial y-min:|Initial y-
max:|Storage pts.:|Line type (rgbw-.:xo):')
set_param([sys,'/',['Demand',13,'graph',13,'']],...
                'Mask Translate','npts = @4; color = @5; ax = [0, @1, @2, @3]; dt=-1;')
set_param([sys,'/',['Demand',13,'graph',13,'']],...
                'Mask Help','This block uses a MATLAB figure window to plot the input
signal.  The graph limits are automatically scaled to the min and max values of the
signal stored in the scope''s signal buffer.  Line type must be in quotes.  See the M-
file sfunyst.m.')
set_param([sys,'/',['Demand',13,'graph',13,'']],...
                'Mask Entries','5\/-10\/10\/200\/''y-/g--/c-./w:/m*/ro/b+''\/')


%     Finished composite block ['Demand',13,'graph',13,''].

set_param([sys,'/',['Demand',13,'graph',13,'']],...
                'position',[435,200,465,240])


%     Subsystem  'Demand'.

new_system([sys,'/','Demand'])
set_param([sys,'/','Demand'],'Location',[148,95,506,279])

add_block('built-in/Constant',[sys,'/','Demand/Constant'])
set_param([sys,'/','Demand/Constant'],...
                'orientation',3,...
                'Value','0',...
                'position',[85,120,105,140])

add_block('built-in/Sum',[sys,'/','Demand/Sum'])
set_param([sys,'/','Demand/Sum'],...
                'position',[120,55,140,75])

add_block('built-in/Outport',[sys,'/','Demand/Demand'])
set_param([sys,'/','Demand/Demand'],...
                'position',[285,55,305,75])

add_block('built-in/Saturation',[sys,'/','Demand/Demand Limit'])
set_param([sys,'/','Demand/Demand Limit'],...
                'Lower Limit','-10',...
                'Upper Limit','+10',...
                'position',[190,53,220,77])
```

```
add_block('built-in/Signal Generator',[sys,'/',['Demand/Signal',13,'Generator']])
set_param([sys,'/',['Demand/Signal',13,'Generator']],...
               'Peak','10.000000',...
               'Peak Range','10.000000',...
               'Freq','0.520000',...
               'Freq Range','5.000000',...
               'Wave','Sqr',...
               'Units','Rads',...
               'position',[35,43,80,77])
add_line([sys,'/','Demand'],[95,115;95,70;115,70])
add_line([sys,'/','Demand'],[85,60;115,60])
add_line([sys,'/','Demand'],[145,65;185,65])
add_line([sys,'/','Demand'],[225,65;280,65])


%     Finished composite block 'Demand'.

set_param([sys,'/','Demand'],...
               'position',[20,14,50,66])


%     Subsystem  'Thruster'.

new_system([sys,'/','Thruster'])
set_param([sys,'/','Thruster'],'Location',[4,82,620,421])
open_system([sys,'/','Thruster'])

add_block('built-in/Note',[sys,'/','Thruster/Vb = Kb*w'])
set_param([sys,'/','Thruster/Vb = Kb*w'],...
               'position',[105,300,110,305])

add_block('built-in/Integrator',[sys,'/','Thruster/Integrator'])
set_param([sys,'/','Thruster/Integrator'],...
               'position',[515,50,535,70])

add_block('built-in/Gain',[sys,'/',['Thruster/Motor',13,'Gain']])
set_param([sys,'/',['Thruster/Motor',13,'Gain']],...
               'Gain','218',...
               'position',[460,44,490,76])

add_block('built-in/Note',[sys,'/','Thruster/Jwdot'])
set_param([sys,'/','Thruster/Jwdot'],...
               'position',[450,20,455,25])

add_block('built-in/Note',[sys,'/','Thruster/w'])
set_param([sys,'/','Thruster/w'],...
               'position',[565,165,570,170])

add_block('built-in/Product',[sys,'/',['Thruster/Motor',13,'Constant']])
set_param([sys,'/',['Thruster/Motor',13,'Constant']],...
               'position',[350,43,380,67])

add_block('built-in/Note',[sys,'/','Thruster/w|w|'])
set_param([sys,'/','Thruster/w|w|'],...
               'position',[420,145,425,150])

add_block('built-in/Constant',[sys,'/','Thruster/Ct'])
```

```
set_param([sys,'/','Thruster/Ct'],...
               'orientation',3,...
               'Value','2.23e-3',...
               'position',[324,230,376,260])

add_block('built-in/Gain',[sys,'/','Thruster/Kb'])
set_param([sys,'/','Thruster/Kb'],...
               'orientation',2,...
               'Gain','0.092',...
               'position',[330,271,380,309])

add_block('built-in/Product',[sys,'/','Thruster/Ct*w|w|'])
set_param([sys,'/','Thruster/Ct*w|w|'],...
               'orientation',2,...
               'position',[280,155,320,175])

add_block('built-in/Constant',[sys,'/','Thruster/Km'])
set_param([sys,'/','Thruster/Km'],...
               'orientation',3,...
               'Value','0.087',...
               'position',[308,105,342,125])

add_block('built-in/Product',[sys,'/','Thruster/w *|w|'])
set_param([sys,'/','Thruster/w *|w|'],...
               'orientation',3,...
               'position',[377,165,413,185])

add_block('built-in/Sum',[sys,'/','Thruster/Sum1'])
set_param([sys,'/','Thruster/Sum1'],...
               'inputs','+-',...
               'position',[420,50,440,70])

add_block('built-in/Saturation',[sys,'/','Thruster/Current Limit'])
set_param([sys,'/','Thruster/Current Limit'],...
               'Lower Limit','-20',...
               'Upper Limit','20',...
               'position',[195,23,225,47])

add_block('built-in/Gain',[sys,'/','Thruster/1//Ra'])
set_param([sys,'/','Thruster/1//Ra'],...
               'Gain','1/0.25',...
               'position',[125,16,155,54])

add_block('built-in/Inport',[sys,'/','Thruster/Voltage'])
set_param([sys,'/','Thruster/Voltage'],...
               'position',[15,20,35,40])

add_block('built-in/Sum',[sys,'/','Thruster/Sum'])
set_param([sys,'/','Thruster/Sum'],...
               'inputs','+-',...
               'position',[80,25,100,45])

add_block('built-in/Note',[sys,'/','Thruster/Vin'])
set_param([sys,'/','Thruster/Vin'],...
               'position',[55,10,60,15])
```

```
%     Subsystem  'Thruster/Stiction'.

new_system([sys,'/','Thruster/Stiction'])
set_param([sys,'/','Thruster/Stiction'],'Location',[91,91,584,339])

add_block('built-in/Note',[sys,'/','Thruster/Stiction/|Speed|'])
set_param([sys,'/','Thruster/Stiction/|Speed|'],...
               'position',[270,195,275,200])

add_block('built-in/Note',[sys,'/',['Thruster/Stiction/if |Speed| > 0
Select',13,'Moving Stiction else ',13,'select Static Stiction']])
set_param([sys,'/',['Thruster/Stiction/if |Speed| > 0 Select',13,'Moving Stiction else
',13,'select Static Stiction']],...
               'position',[400,40,405,45])

add_block('built-in/Inport',[sys,'/','Thruster/Stiction/in_2'])
set_param([sys,'/','Thruster/Stiction/in_2'],...
               'Port','2',...
               'position',[15,205,35,225])

add_block('built-in/Outport',[sys,'/','Thruster/Stiction/out_1'])
set_param([sys,'/','Thruster/Stiction/out_1'],...
               'position',[445,90,465,110])

add_block('built-in/Switch',[sys,'/','Thruster/Stiction/Switch'])
set_param([sys,'/','Thruster/Stiction/Switch'],...
               'Threshold','1',...
               'position',[380,84,410,116])

add_block('built-in/Abs',[sys,'/','Thruster/Stiction/Abs'])
set_param([sys,'/','Thruster/Stiction/Abs'],...
               'position',[120,203,150,227])

add_block('built-in/Constant',[sys,'/','Thruster/Stiction/Stuck'])
set_param([sys,'/','Thruster/Stiction/Stuck'],...
               'Value','0',...
               'position',[220,90,240,110])

add_block('built-in/Switch',[sys,'/','Thruster/Stiction/Switch2'])
set_param([sys,'/','Thruster/Stiction/Switch2'],...
               'Threshold','0.1',...
               'position',[295,114,325,146])

add_block('built-in/Switch',[sys,'/','Thruster/Stiction/Switch1'])
set_param([sys,'/','Thruster/Stiction/Switch1'],...
               'Threshold','0.1',...
               'position',[295,59,325,91])

add_block('built-in/Abs',[sys,'/','Thruster/Stiction/Abs2'])
set_param([sys,'/','Thruster/Stiction/Abs2'],...
               'position',[170,118,200,142])

add_block('built-in/Abs',[sys,'/','Thruster/Stiction/Abs1'])
set_param([sys,'/','Thruster/Stiction/Abs1'],...
               'position',[170,63,200,87])

add_block('built-in/Note',[sys,'/','Thruster/Stiction/Speed'])
```

```
set_param([sys,'/','Thruster/Stiction/Speed'],...
                'position',[75,190,80,195])

add_block('built-in/Dead Zone',[sys,'/','Thruster/Stiction/Moving Dead Zone'])
set_param([sys,'/','Thruster/Stiction/Moving Dead Zone'],...
                'Lower_value','-1.8',...
                'Upper_value','1.8',...
                'position',[95,62,125,88])

add_block('built-in/Dead Zone',[sys,'/','Thruster/Stiction/Static Dead Zone'])
set_param([sys,'/','Thruster/Stiction/Static Dead Zone'],...
                'Lower_value','-4.8',...
                'Upper_value','4.8',...
                'position',[95,117,125,143])

add_block('built-in/Inport',[sys,'/','Thruster/Stiction/in_1'])
set_param([sys,'/','Thruster/Stiction/in_1'],...
                'position',[15,95,35,115])

add_block('built-in/Note',[sys,'/',['Thruster/Stiction/if Value is not in
the',13,'dead zone output the ',13,'value, else output 0',13,'(Motor Stuck)']])
set_param([sys,'/',['Thruster/Stiction/if Value is not in the',13,'dead zone output
the ',13,'value, else output 0',13,'(Motor Stuck)']],...
                'position',[305,5,310,10])
add_line([sys,'/','Thruster/Stiction'],[40,105;60,105;60,75;90,75])
add_line([sys,'/','Thruster/Stiction'],[40,105;60,105;60,130;90,130])
add_line([sys,'/','Thruster/Stiction'],[415,100;440,100])
add_line([sys,'/','Thruster/Stiction'],[40,215;115,215])
add_line([sys,'/','Thruster/Stiction'],[155,215;365,215;375,100])
add_line([sys,'/','Thruster/Stiction'],[130,75;165,75])
add_line([sys,'/','Thruster/Stiction'],[205,75;290,75])
add_line([sys,'/','Thruster/Stiction'],[40,105;40,45;220,45;220,65;290,65])
add_line([sys,'/','Thruster/Stiction'],[245,100;260,100;260,85;290,85])
add_line([sys,'/','Thruster/Stiction'],[130,130;165,130])
add_line([sys,'/','Thruster/Stiction'],[205,130;290,130])
add_line([sys,'/','Thruster/Stiction'],[245,100;260,100;260,140;290,140])
add_line([sys,'/','Thruster/Stiction'],[40,105;40,165;250,165;250,120;290,120])
add_line([sys,'/','Thruster/Stiction'],[330,75;345,75;345,90;375,90])
add_line([sys,'/','Thruster/Stiction'],[330,130;345,130;345,110;375,110])


%     Finished composite block 'Thruster/Stiction'.

set_param([sys,'/','Thruster/Stiction'],...
                'position',[280,21,310,74])

add_block('built-in/Sum',[sys,'/','Thruster/Slip'])
set_param([sys,'/','Thruster/Slip'],...
                'orientation',2,...
                'inputs','+-',...
                'position',[165,160,185,180])

add_block('built-in/Outport',[sys,'/',['Thruster/Overall',13,'Thrust']])
set_param([sys,'/',['Thruster/Overall',13,'Thrust']],...
                'orientation',2,...
                'Port','3',...
                'position',[125,160,145,180])
```

```
add_block('built-in/Abs',[sys,'/','Thruster/Abs'])
set_param([sys,'/','Thruster/Abs'],...
               'orientation',2,...
               'position',[475,185,505,205])

add_block('built-in/Note',[sys,'/','Thruster/|w|'])
set_param([sys,'/','Thruster/|w|'],...
               'position',[440,185,445,190])

add_block('built-in/Note',[sys,'/',['Thruster/Propellor',13,'Thrust']])
set_param([sys,'/',['Thruster/Propellor',13,'Thrust']],...
               'position',[225,135,230,140])

add_block('built-in/Gain',[sys,'/',['Thruster/Slip',13,'Coefficient']])
set_param([sys,'/',['Thruster/Slip',13,'Coefficient']],...
               'orientation',3,...
               'Gain','20',...
               'position',[227,205,253,225])

add_block('built-in/Inport',[sys,'/',['Thruster/Vehicle',13,'Speed']])
set_param([sys,'/',['Thruster/Vehicle',13,'Speed']],...
               'orientation',3,...
               'Port','2',...
               'position',[230,250,250,270])

add_block('built-in/Note',[sys,'/','Thruster/Tm'])
set_param([sys,'/','Thruster/Tm'],...
               'position',[395,35,400,40])

add_block('built-in/Gain',[sys,'/','Thruster/alpha'])
set_param([sys,'/','Thruster/alpha'],...
               'orientation',3,...
               'Gain','4.3e-5',...
               'position',[381,105,409,130])

add_block('built-in/Note',[sys,'/','Thruster/TL'])
set_param([sys,'/','Thruster/TL'],...
               'position',[405,80,410,85])

add_block('built-in/Outport',[sys,'/','Thruster/Current'])
set_param([sys,'/','Thruster/Current'],...
               'position',[585,5,605,25])

add_block('built-in/Outport',[sys,'/','Thruster/Speed'])
set_param([sys,'/','Thruster/Speed'],...
               'Port','2',...
               'position',[585,50,605,70])
add_line([sys,'/','Thruster'],[395,100;395,65;415,65])
add_line([sys,'/','Thruster'],[395,160;395,135])
add_line([sys,'/','Thruster'],[105,35;120,35])
add_line([sys,'/','Thruster'],[325,290;55,290;55,40;75,40])
add_line([sys,'/','Thruster'],[40,30;75,30])
add_line([sys,'/','Thruster'],[230,35;255,35;255,15;580,15])
add_line([sys,'/','Thruster'],[385,55;415,55])
add_line([sys,'/','Thruster'],[325,100;325,60;345,60])
add_line([sys,'/','Thruster'],[445,60;455,60])
```

```
add_line([sys,'/','Thruster'],[495,60;510,60])
add_line([sys,'/','Thruster'],[540,60;580,60])
add_line([sys,'/','Thruster'],[395,160;325,160])
add_line([sys,'/','Thruster'],[350,225;350,170;325,170])
add_line([sys,'/','Thruster'],[275,165;190,165])
add_line([sys,'/','Thruster'],[160,170;150,170])
add_line([sys,'/','Thruster'],[240,245;240,230])
add_line([sys,'/','Thruster'],[240,200;240,175;190,175])
add_line([sys,'/','Thruster'],[540,60;555,60;555,225;385,225;385,190])
add_line([sys,'/','Thruster'],[540,60;555,60;555,290;385,290])
add_line([sys,'/','Thruster'],[160,35;190,35])
add_line([sys,'/','Thruster'],[230,35;275,35])
add_line([sys,'/','Thruster'],[315,50;345,50])
add_line([sys,'/','Thruster'],[540,60;550,60;550,145;260,145;260,60;275,60])
add_line([sys,'/','Thruster'],[470,195;460,195;460,205;405,205;405,190])
add_line([sys,'/','Thruster'],[540,60;555,60;555,195;510,195])


%     Finished composite block 'Thruster'.

set_param([sys,'/','Thruster'],...
               'position',[245,27,275,83])

add_block('built-in/Constant',[sys,'/',['Vehicle Speed',13,'']])
set_param([sys,'/',['Vehicle Speed',13,'']],...
               'orientation',3,...
               'Value','0',...
               'position',[195,120,215,140])


%     Subsystem  'Thrust'.

new_system([sys,'/','Thrust'])
set_param([sys,'/','Thrust'],'Location',[0,59,274,252])

add_block('built-in/Inport',[sys,'/','Thrust/x'])
set_param([sys,'/','Thrust/x'],...
               'position',[65,55,85,75])

add_block('built-in/S-Function',[sys,'/',['Thrust/S-function',13,'M-file which
plots',13,'lines',13,'']])
set_param([sys,'/',['Thrust/S-function',13,'M-file which plots',13,'lines',13,'']],...
               'function name','sfunyst',...
               'parameters','ax, color, npts, dt',...
               'position',[130,55,180,75])
add_line([sys,'/','Thrust'],[90,65;125,65])
set_param([sys,'/','Thrust'],...
               'Mask
Display','plot(0,0,100,100,[83,76,63,52,42,38,28,16,11,84,11,11,11,90,90,11],[75,58,47
,54,72,80,84,74,65,65,65,90,40,40,90,90])',...
               'Mask Type','Storage scope.')
set_param([sys,'/','Thrust'],...
               'Mask Dialogue','Storage scope using MATLAB graph window.\nEnter
plotting ranges and line type.|Initial Time Range:|Initial y-min:|Initial y-
max:|Storage pts.:|Line type (rgbw-.:xo):')
set_param([sys,'/','Thrust'],...
               'Mask Translate','npts = @4; color = @5; ax = [0, @1, @2, @3]; dt=-1;')
```

```
set_param([sys,'/','Thrust'],...
                'Mask Help','This block uses a MATLAB figure window to plot the input
signal.  The graph limits are automatically scaled to the min and max values of the
signal stored in the scope''s signal buffer.  Line type must be in quotes.  See the M-
file sfunyst.m.')
set_param([sys,'/','Thrust'],...
                'Mask Entries','5\/-10\/10\/200\/''y-/g--/c-./w:/m*/ro/b+''\/')


%     Finished composite block 'Thrust'.

set_param([sys,'/','Thrust'],...
                'position',[435,140,465,180])


%     Subsystem  'Speed'.

new_system([sys,'/','Speed'])
set_param([sys,'/','Speed'],'Location',[0,59,274,252])

add_block('built-in/Inport',[sys,'/','Speed/x'])
set_param([sys,'/','Speed/x'],...
                'position',[65,55,85,75])

add_block('built-in/S-Function',[sys,'/',['Speed/S-function',13,'M-file which
plots',13,'lines',13,'']])
set_param([sys,'/',['Speed/S-function',13,'M-file which plots',13,'lines',13,'']],...
                'function name','sfunyst',...
                'parameters','ax, color, npts, dt',...
                'position',[130,55,180,75])
add_line([sys,'/','Speed'],[90,65;125,65])
set_param([sys,'/','Speed'],...
                'Mask
Display','plot(0,0,100,100,[83,76,63,52,42,38,28,16,11,84,11,11,11,90,90,11],[75,58,47
,54,72,80,84,74,65,65,65,90,40,40,90,90])',...
                'Mask Type','Storage scope.')
set_param([sys,'/','Speed'],...
                'Mask Dialogue','Storage scope using MATLAB graph window.\nEnter
plotting ranges and line type.|Initial Time Range:|Initial y-min:|Initial y-
max:|Storage pts.:|Line type (rgbw-.:xo):')
set_param([sys,'/','Speed'],...
                'Mask Translate','npts = @4; color = @5; ax = [0, @1, @2, @3]; dt=-1;')
set_param([sys,'/','Speed'],...
                'Mask Help','This block uses a MATLAB figure window to plot the input
signal.  The graph limits are automatically scaled to the min and max values of the
signal stored in the scope''s signal buffer.  Line type must be in quotes.  See the M-
file sfunyst.m.')
set_param([sys,'/','Speed'],...
                'Mask Entries','5\/-10\/10\/200\/''y-/g--/c-./w:/m*/ro/b+''\/')


%     Finished composite block 'Speed'.

set_param([sys,'/','Speed'],...
                'position',[435,75,465,115])
```

```
%     Subsystem  'Current'.

new_system([sys,'/','Current'])
set_param([sys,'/','Current'],'Location',[0,59,274,252])

add_block('built-in/Inport',[sys,'/','Current/x'])
set_param([sys,'/','Current/x'],...
               'position',[65,55,85,75])

add_block('built-in/S-Function',[sys,'/',['Current/S-function',13,'M-file which
plots',13,'lines',13,'']])
set_param([sys,'/',['Current/S-function',13,'M-file which
plots',13,'lines',13,'']],...
               'function name','sfuny',...
               'parameters','ax, color,dt',...
               'position',[130,55,180,75])
add_line([sys,'/','Current'],[90,65;125,65])
set_param([sys,'/','Current'],...
               'Mask
Display','plot(0,0,100,100,[90,10,10,10,90,90,10],[65,65,90,40,40,90,90],[90,78,69,54,
40,31,25,10],[77,60,48,46,56,75,81,84])',...
               'Mask Type','Graph scope.')
set_param([sys,'/','Current'],...
               'Mask Dialogue','Graph scope using MATLAB graph window.\nEnter plotting
ranges and line type.|Time range:|y-min:|y-max:|Line type (rgbw-:*). Seperate each
plot by ''/'':')
set_param([sys,'/','Current'],...
               'Mask Translate','color = @4; ax = [0, @1, @2, @3]; dt = -1;')
set_param([sys,'/','Current'],...
               'Mask Help','This block plots to the MATLAB graph window and can be
used as an improved version of the Scope block. Look at the m-file sfuny.m to see how
it works. This block can take scalar or vector input signal.')
set_param([sys,'/','Current'],...
               'Mask Entries','1\/-20\/20\/''w-/g--/c-./w:/m*/ro/b+''\/')


%     Finished composite block 'Current'.

set_param([sys,'/','Current'],...
               'position',[435,16,465,54])
add_line(sys,[180,40;240,40])
add_line(sys,[280,35;430,35])
add_line(sys,[280,55;375,55;375,95;430,95])
add_line(sys,[280,75;345,75;345,160;430,160])
add_line(sys,[205,115;205,70;240,70])
add_line(sys,[55,40;140,40])
add_line(sys,[55,40;85,40;85,220;430,220])

drawnow

% Return any arguments.
if (nargin | nargout)
        % Must use feval here to access system in memory
        if (nargin > 3)
                if (flag == 0)
                        eval(['[ret,x0,str,ts,xts]=',sys,'(t,x,u,flag);'])
                else
```

```
                        eval(['ret =', sys,'(t,x,u,flag);'])
                end
        else
                [ret,x0,str,ts,xts] = feval(sys);
        end
else
        drawnow % Flash up the model and execute load callback
end
```

10.4 Power Drive Layout

The circuit was designed and laid out using the computer package CADSTAR. A single sided P.C.B. was designed to meet the design criteria detailed above. The input signals and signal power are connected to the P.C.B. by a Molex connector, whilst the power connectors are attached directly to the power rails. Figure 106 shows the layout of the tracks as laid out and designed in the Cadstar package.

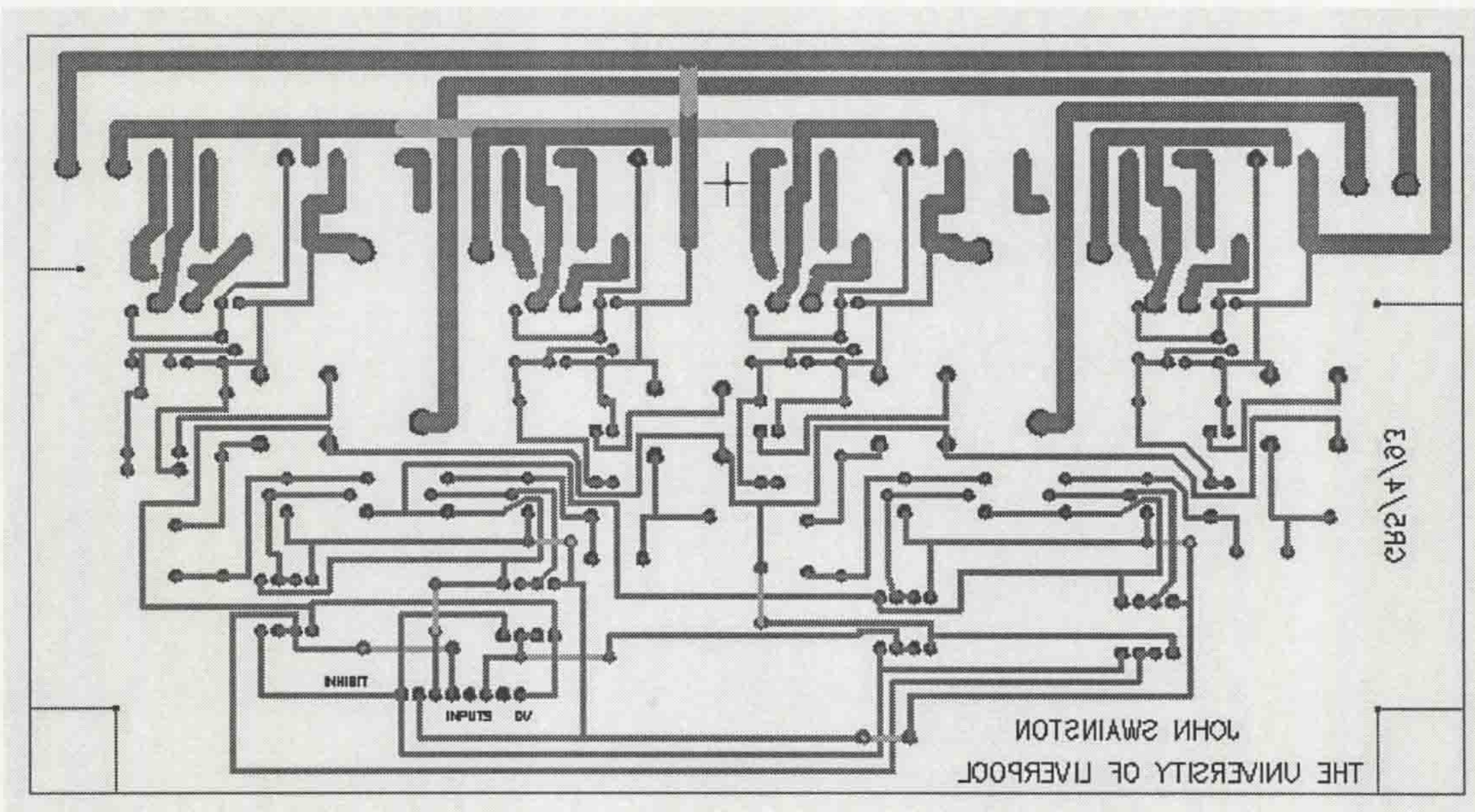

Figure 106, Motor Drive Track Layout

As can be seen from the diagram the power devices are mounted at the top of the board, with the signal drive side of the circuit using the thinner tracks at the base of the diagram

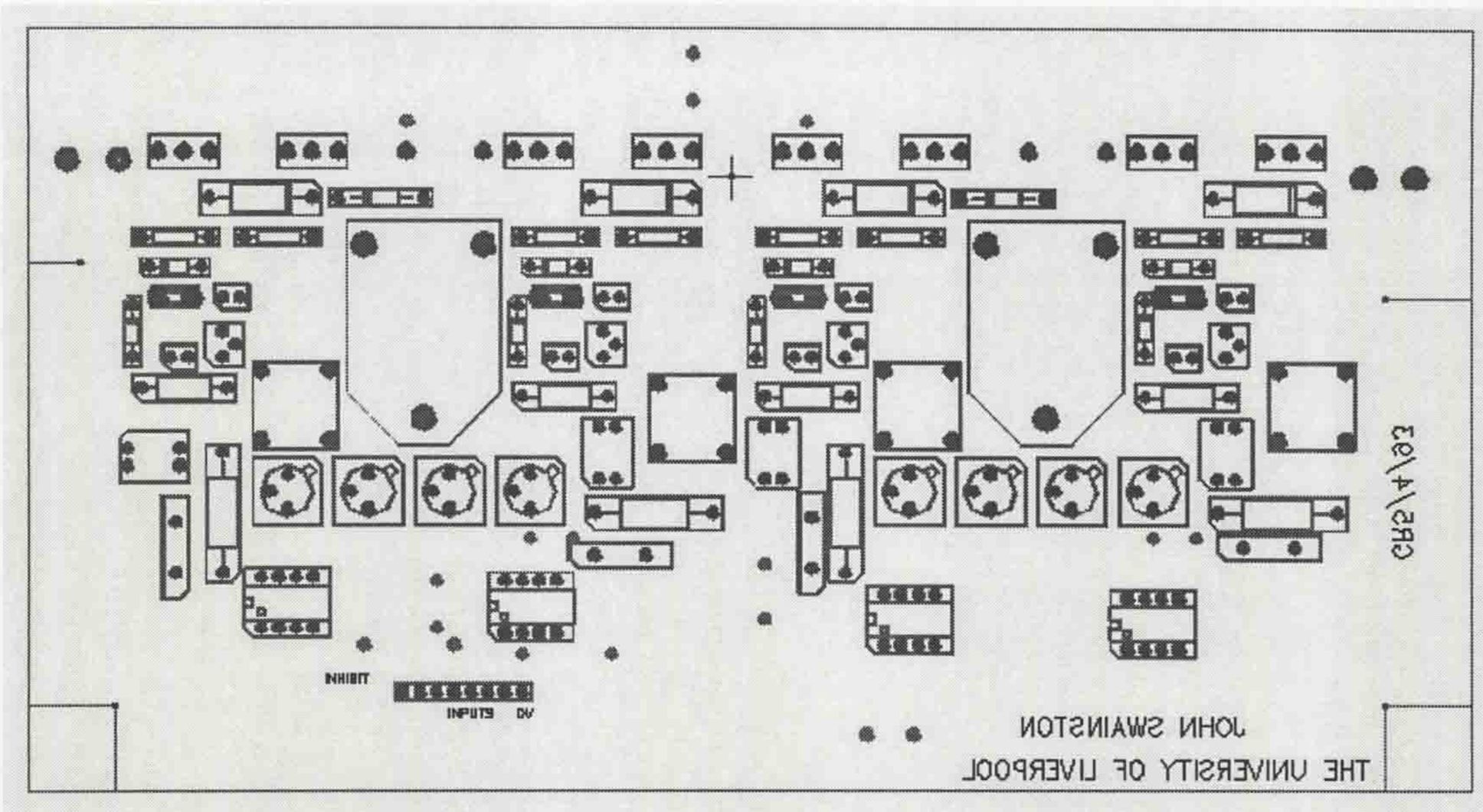

Figure 107, Power Board Component Layout

Figure 107 shows the layout diagram for the components on the power board

10.5 Microcontroller Data

The following data is reproduced from the Intel Reference Manual[4.10].

10.5.1 SCON: Serial Port Control Register

```
SCON    Address = 98H  Reset Value = 0000 0000B
        Bit Addressable

        Bit:    7       6       5       4       3       2       1       0
                SM0/FE  SM1     SM2     REN     TB8     RB8     TI      RI
Symbol  Function
FE      Framing Error bit. This bit is set by the receiver when an invalid stop bit is
        detected. The FE bit is not cleared by valid frames but should be cleared by
        software. The SMOD0 bit must be set to enable access to the FE bit.
SMOD0   is located at PCON6
SM0     Serial Port Mode Bit 0
SMOD0   must = 0 to access bit SM0.   SMOD0 is located at PCON6.
SM1     Serial Port Mode Bit 1
        .
        SM0     SM1     Mode    Description     Baud Rate**
        0       0       0       shift register          FOSC/12
        0       1       1       8-bit UART      variable
        1       0       2       9-bit UART      FOSC/64 or FOSC/32
        1       1       3       9-bit UART      variable
**FOSC = oscillator frequency
SM2     Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1
        then RI will not be set unless the received 9th data bit (RB8) is 1, indicating
        an address, and the received byte is a Given or Broadcast Address. In Mode 1,
        if SM2 = 1 then RI will not be activated unless a valid stop bit was received,
```

and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.

REN Enables serial reception. Set by software to enable reception. Clear by software to disable reception.

TB8 The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.

RB8 In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.

TI Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.

RI Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

10.5.2 CCAPMn: PCA Modules Compare/Capture Registers

```
CCAPMn Address        CCAPM0 0DAH    Reset Value = X000 0000B

       (n = 0--4)     CCAPM1 0DBH
              CCAPM2 0DCH
              CCAPM3 0DDH
              CCAPM4 0DEH

       Not Bit Addressable

       Bit    7      6      5      4      3      2      1      0
              --     ECOMn  CAPPn  CAPNn  MATn   TOGn   PWMn   ECCFn
```

Symbol Function

-- Not implemented, reserved for future use.

ECOMn Enable Comparator. ECOMn = 1 enables the comparator function.

CAPPn Capture Positive, CAPPn = 1 enables positive edge capture.

CAPNn Capture Negative, CAPNn = 1 enables negative edge capture.

MATn Match. When MATn = 1, a match of the PCA counter with this module's compare/capture register causes the CCFn bit in CCON to be set, flagging an interrupt.

TOGn Toggle. When TOGn = 1, a match of the PCA counter with this module's compare/capture register causes the CEXn pin to toggle.

PWMn Pulse Width Modulation Mode. PWMn = 1 enables the CEXn pin to be used as a pulse width modulated output.

ECCFn Enable CCF interrupt. Enables compare/capture flag CCFn in the CCON register to generate an interrupt.

10.5.3 Programmable Counter Array (PCA) Overview

The Programmable Counter Array (PCA) consists of a 16-bit timer/counter and five 16-bit compare/capture modules as shown in Figure 24. The PCA timer/counter serves as a common time base for the five modules and is the only timer which can service the PCA. Its clock input can be programmed to count any one of the following signals:

- oscillator frequency ÷ 12
- oscillator frequency ÷ 4

- Timer 0 overflow
- external input on ECI (P1.2).

Each compare/capture module can be programmed in any one of the following modes:

- software timer/counter
- rising and/or falling edge capture
- high speed output
- pulse width modulator.
- Module 4 can be programmed as a watchdog timer.

When the compare/capture modules are programmed in the capture mode, software timer, or high speed output mode, an interrupt can be generated when the module executes its function. All five modules plus the PCA timer overflow share one interrupt vector (see the PCA Interrupt).
The PCA timer/counter and compare/capture modules share Port 1 pins for external I/O. These pins are listed below. If the port pin is not used for the PCA, it can still be used for standard I/O.

PCA Component	External I/O Pin
16-bit Counter	P1.2 / ECI
16-bit Module 0	P1.3 / CEX0
16-bit Module 1	P1.4 / CEX1
16-bit Module 2	P1.5 / CEX2
16-bit Module 3	P1.6 / CEX3
16-bit Module 4	P1.7 / CEX4

PCA 16-Bit Timer/Counter

The PCA has a free-running 16-bit timer/counter consisting of registers CH and CL (the high and low bytes of the count value). These two registers can be read or written to at any time. The clock input can be selected from the following four modes:

Oscillator frequency ÷ 12	The CL register is incremented at S5P2 of every machine cycle. With a 16 MHz crystal, the timer increments every 750 nanoseconds.
Oscillator frequency ÷ 4	The CL register is incremented at S1P2, S3P2 and S5P2 of every machine cycle. With a 16 MHz crystal, the timer increments every 250 nanoseconds.
Timer 0 overflows	The CL register is incremented at S5P2 of the machine cycle when Timer 0 overflows. This mode allows a programmable input frequency to the PCA.
External input	The CL register is incremented at the first one of S1P2, S3P2 and S5P2 after a 1-to-0 transition is detected on the ECI pin (P1.2). P1.2 is sampled at S1P2, S3P2 and S5P2 of every machine cycle. The maximum input frequency in this mode is oscillator frequency ÷ 8.

CH is incremented after two oscillator periods when CL overflows.
The mode register CMOD contains the Count Pulse Select bits (CPS1 and CPS0) to specify the clock input. This register also contains the ECF bit which enables the PCA counter overflow to generate the PCA interrupt. In addition, the user has the option of turning off the PCA timer during Idle Mode by setting the Counter Idle bit (CIDL). The Watchdog Timer Enable bit (WDTE) will be discussed in a later section.
The CCON register contains two more bits which are associated with the PCA timer/counter. The CF bit gets set by hardware when the counter overflows, and the CR bit is set or cleared to turn the counter on or off. The other five bits in this

register are the event flags for the compare/capture modules and will be discussed in the next section.

CMOD: PCA Counter Mode Register

CMOD Address = 0D9H Reset Value = 00XX X000B
Not Bit Addressable

Bit	7	6	5	4	3	2	1	0
	CIDL	WDTE	--	--	--	CPS1	CPS0	ECF

Symbol Function

CIDL Counter Idle control:
CIDL = 0 programs the PCA Counter to continue functioning during Idle Mode.
CIDL = 1 programs it to be gated off during Idle.

WDTE Watchdog Timer Enable:
WDTE = 0 disables Watchdog Timer function on PCA Module 4.
WDTE = 1 enables it.

-- Not implemented, reserved for future use.*

CPS1 PCA Count Pulse Select bit 1.

CPS0 PCA Count Pulse Select bit 0.

CPS1	CPS0	Selected PCA Input**
0	0	Internal clock, Fosc ÷ 12 (Fosc = oscillator frequency)
0	1	Internal clock, Fosc ÷ 4
1	0	Timer 0 overflow
1	1	External clock at ECI/P1.2 pin (max. rate = Fosc ÷ 8)

ECF PCA Enable Counter Overflow interrupt:
ECF = 1 enables CF bit in CCON to generate an interrupt.
ECF = 0 disables that function of CF.

CCON: PCA Counter Control Register

CCON Address = 0D8H Reset Value = 00X0 0000B
Bit Addressable

Bit	7	6	5	4	3	2	1	0
	CF	CR	--	CCF4	CCF3	CCF2	CCF1	CCF0

Symbol Function

CF PCA Counter Overflow flag. Set by hardware when the counter rolls over. CF flags an interrupt if bit ECF in CMOD is set. CF may be set by either hardware or software but can only be cleared by software.

CR PCA Counter Run control bit. Set by software to turn the PCA counter on. Must be cleared by software to turn the PCA counter off.

-- Not implemented, reserved for future use.

CCF4 PCA Module 4 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.

CCF3 PCA Module 3 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.

CCF2 PCA Module 2 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.

CCF1 PCA Module 1 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.

CCF0 PCA Module 0 interrupt flag. Set by hardware when a match or capture occurs. Must be cleared by software.

PCA Capture/Compare Modules

Each of the five compare/capture modules has six possible functions:

- 16-bit Capture, positive-edge triggered
- 16-bit Capture, negative-edge triggered
- 16-bit Capture, both positive and negative-edge triggered
- 16-bit Software Timer
- 16-bit High Speed Output
- 8-bit Pulse Width Modulator.

In addition, module 4 can be used as a Watchdog Timer. The modules can be programmed in any combination of the different modes.
Each module has a mode register called CCAPMn (n = 0, 1, 2, 3, or 4) to select which function it will perform. Note the ECCFn bit which enables the PCA interrupt when a module's event flag is set. The event flags (CCFn) are located in the CCON register and are set whenever a capture event, software timer, or high speed output event occurs for a given module.
Table 15 shows the combinations of bits in the CCAPMn register that are valid and have a defined function. Invalid combinations will produce undefined results.
Each module also has a pair of 8-bit compare/capture registers (CCAPnH and CCAPnL) associated with it. These registers store the time when a capture event occurred or when a compare event should occur. For the PWM mode, the high byte register CCAPnH controls the duty cycle of the waveform.

PCA Module Modes (CCAPMn Register)

--	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	Module Function
X	0	0	0	0	0	0	0	No operation
X	X	1	0	0	0	0	X	16-bit capture by a positive-edge trigger on CEXn
X	X	0	1	0	0	0	X	16-bit capture by negative-edge trigger on CEXn
X	X	1	1	0	0	0	X	16-bit capture by a transition on CEXn
X	1	0	0	1	0	0	X	16-bit Software Timer
X	1	0	0	1	1	0	X	16-bit High Speed Output
X	1	0	0	0	0	1	0	8-bit PWM
X	1	0	0	1	X	0	X	Watchdog Timer

X = Don't Care

10.6 Network Function Calls

```
/*                                                          */
/*                          SENDNEW.H                       */
/*                                                          */

typedef unsigned char TBuffer[32];    /* Message buffer type.       */
typedef unsigned int (far *s_arrayptr); /* Static screen array pointer. */

                                      /* Prototypes.                */
void  Read_Time(void);                /* Procedure to read system time*/
int   sendpacket(unsigned char*, char, unsigned char);/* Send data  */
                                                      /* packet.    */
int   sendmess(char*, int);           /* Send message.              */
```

```
int   getbyte(void);                  /* Receive a byte.            */
int   getbytew(void);                 /* Receive a word.            */
int   putbyte(char);                  /* Send a byte.               */
void  Reset_Card(void);               /* Reset the thrusters.       */
int   setthruster(char, float);       /* Set the value sent to an   */
                                      /* individual thruster.       */
int   Data_Received(void);            /* Is Data ready to be read ? */
int   receivepacket(char*);           /* Receive a packet of info.  */
void  Reset_DACs(void);               /* A procedure to reset all the */
                                      /* DACs to zero.              */
void  Request_Inhibit(void);          /* Request Inhibits Status.   */
void  Set_Inhibit(int);               /* Sets Inhibits Status.      */
void  Request_Speed(void);            /* Requests speed on paddle.  */
void  Request_Temperature(void);      /* Requests the Current temp. */
void  Request_Depth(void);            /* Request The Vehicles Depth.*/
void  Request_Compass(void);          /* Requests the Current Heading.*/
void  Request_Range(void);            /* Requests the ultrasonic range*/
void  Request_Tilt1(void);            /* Requests tilt 1.           */
void  Request_Tilt2(void);            /* Requests tilt 2.           */
void  Request_Sensors(void);          /* Requests all sensor values.*/
void  Request_Thruster(int);          /* Request Individual Thruster*/
void  Request_Thrusters(void);        /* Request all thruster values*/
void  Request_Thrust_Status(void);    /* Request status of thrusters*/
void  Request_Sensor_Status(void);    /* Request status of sensors. */
void  Request_Gyro(void);             /* Request gyro values.       */
void  Request_Acceleration(int);      /* Requests acceleration.     */
void  Request_Velocity(int);          /* Requests vehicles velocity.*/
void  Request_Position(int);          /* Requests vehicles position.*/
void  ESTOP(void);                    /* Emergency stop command.    */
void  Reset_Accelerometer(void);      /* Resets the accelerometer.  */
void  Absolute_Move_ACK(int);         /* Reports absolute move done.*/
void  Relative_Move_ACK(int);         /* Reports relative move done.*/
void  Reset_Time(void);               /* Resets the current time to 0 */
void  Check_For_Network(void);        /* Procedure to check network.*/
void  Start_Interrupt_Procedures(void); /* Procedure to start interrupt */
                                      /* tied routines.             */
void  Stop_Interrupt_Procedures(void);/* Procedure to stop interrupt*/
                                      /* tied routines.             */
void  Display_Values(void);           /* Procedure to display values. */
void  Check2(void);                   /* Checks for and assigns data.*/
void  Send_System_Status(void);       /* Transmits systems status.  */
void  Request_Pan(void);              /* Requests camera Pan angle. */
void  Request_Tilt(void);             /* Requests camera Tilt angle.*/
void  EnableIRQ (char IRQ);           /* Enable 8259 to generate IRQ.*/
void  DisableIRQ ( char IRQ);         /* Disable 8259 from generat IRQ*/
int   ReadFromBuffer(TBuffer);        /* Reads a message from buffer.*/
void  Assign_Data(void);              /* Convert message to data    */
void  Auto_Request_Defaults(void);    /* Resets request levels default*/
```

10.7 Transputer Network Interface Cards

The transputer interface cards used for this thesis are shown here. As can be seen the cards are standard eurocard size (110 x 160mm) and have a DIN41612 connector. This

modular design allows the cards to be interchanged and added at will, depending on the systems requirements.

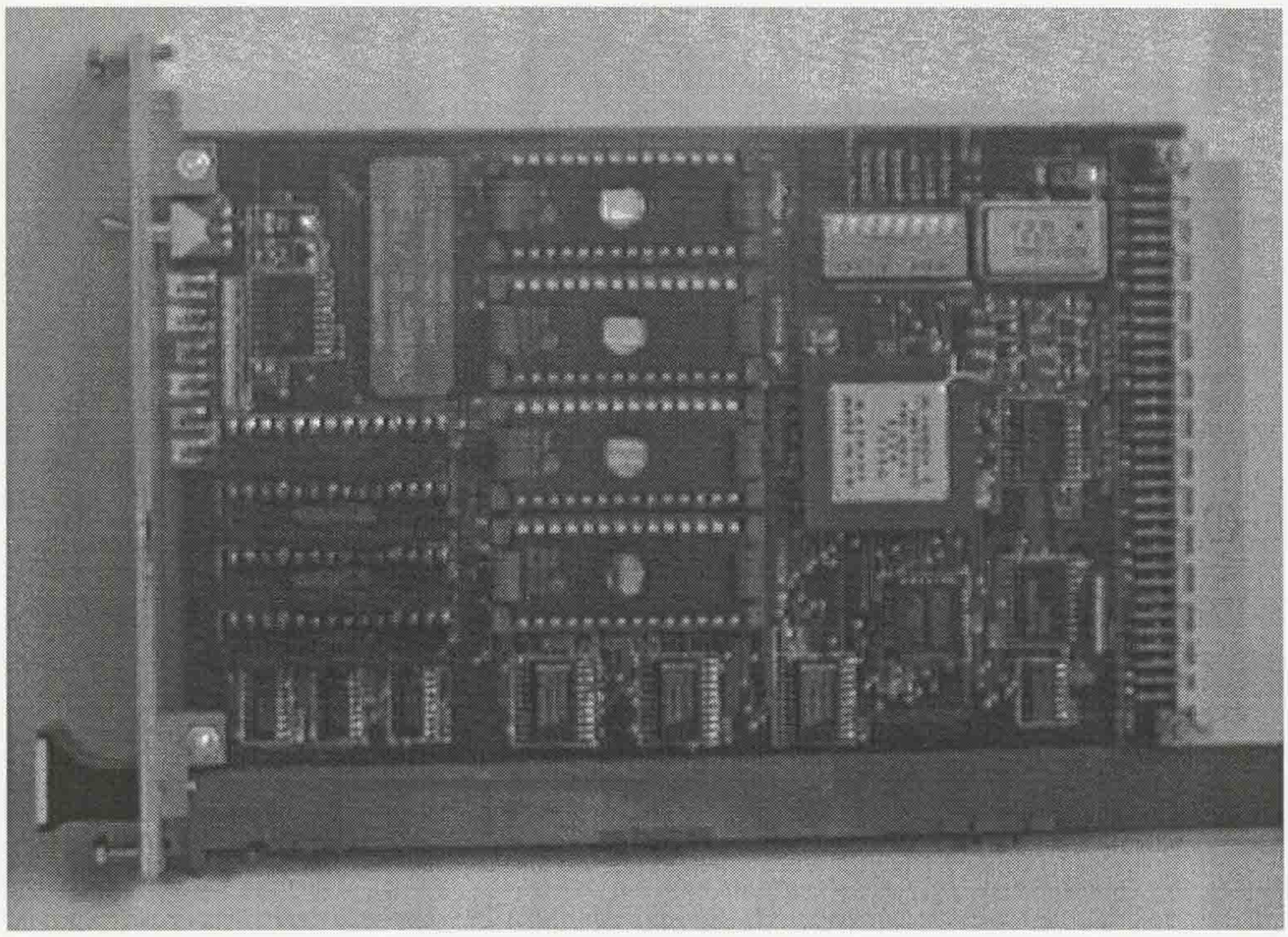

Figure 108, The Transputer Processor Card

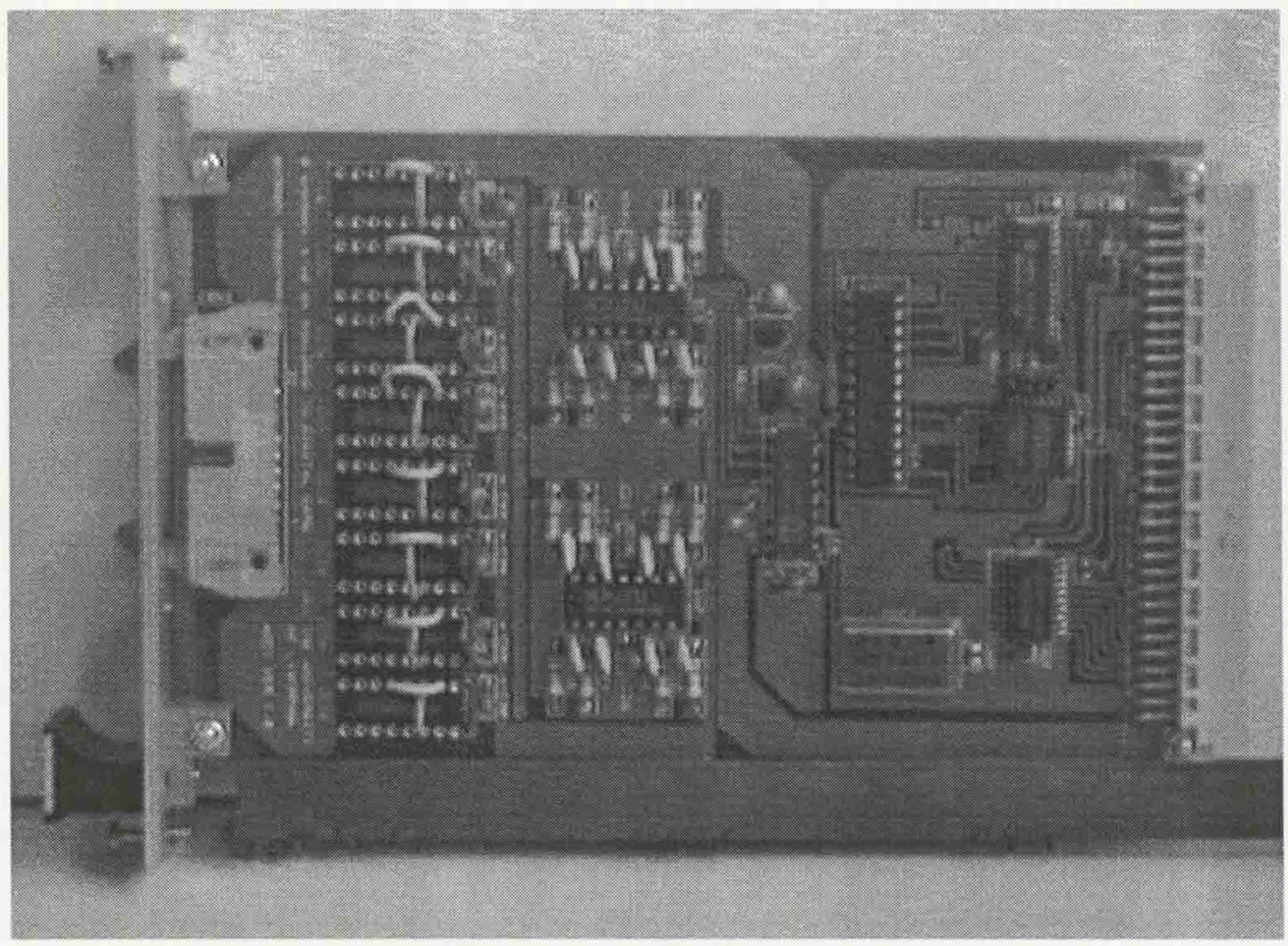

Figure 109, The Analogue to Digital Input Card

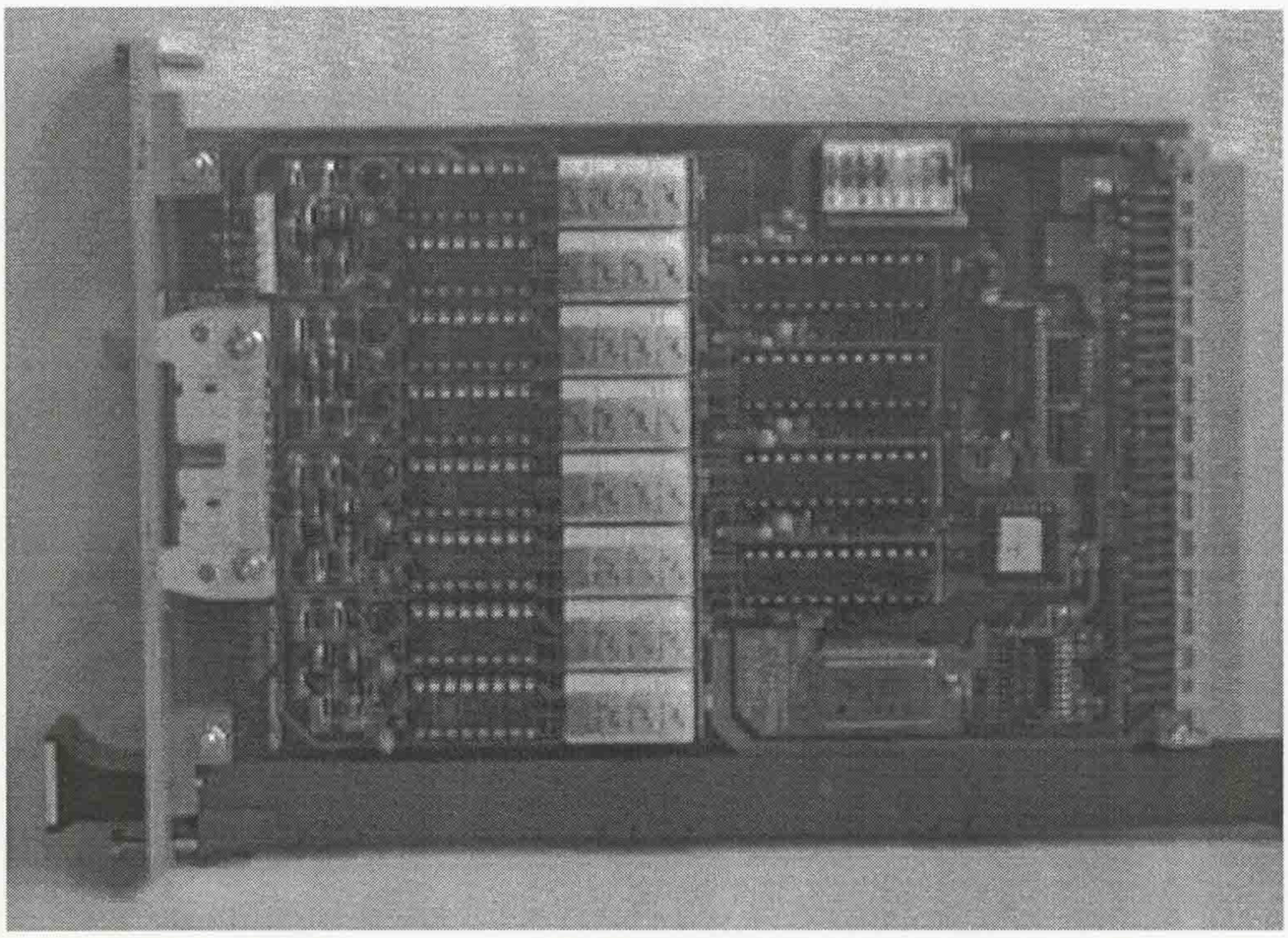

Figure 110, The Digital to Analogue Output Card

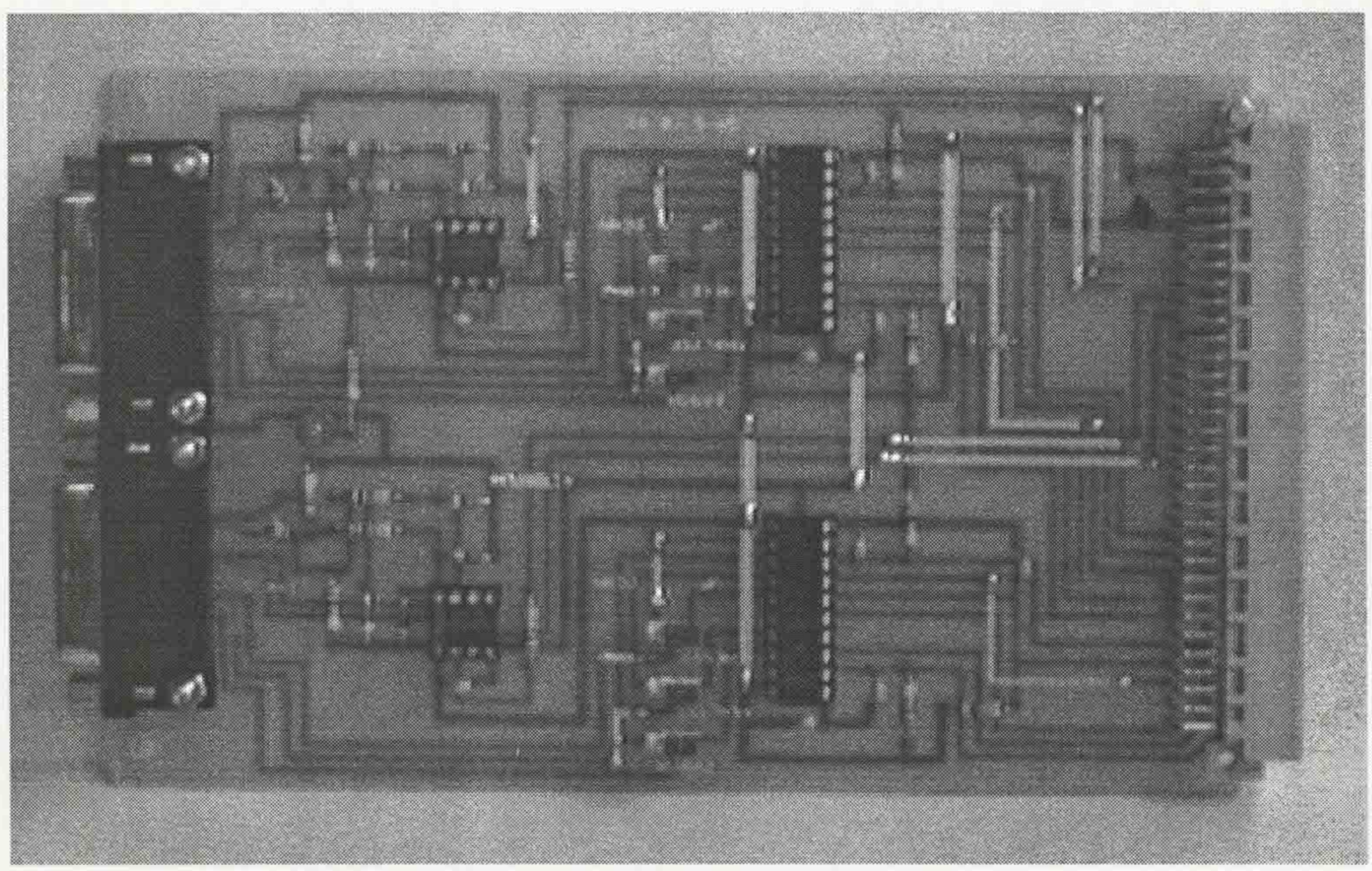

Figure 111, The Link Adapter Card

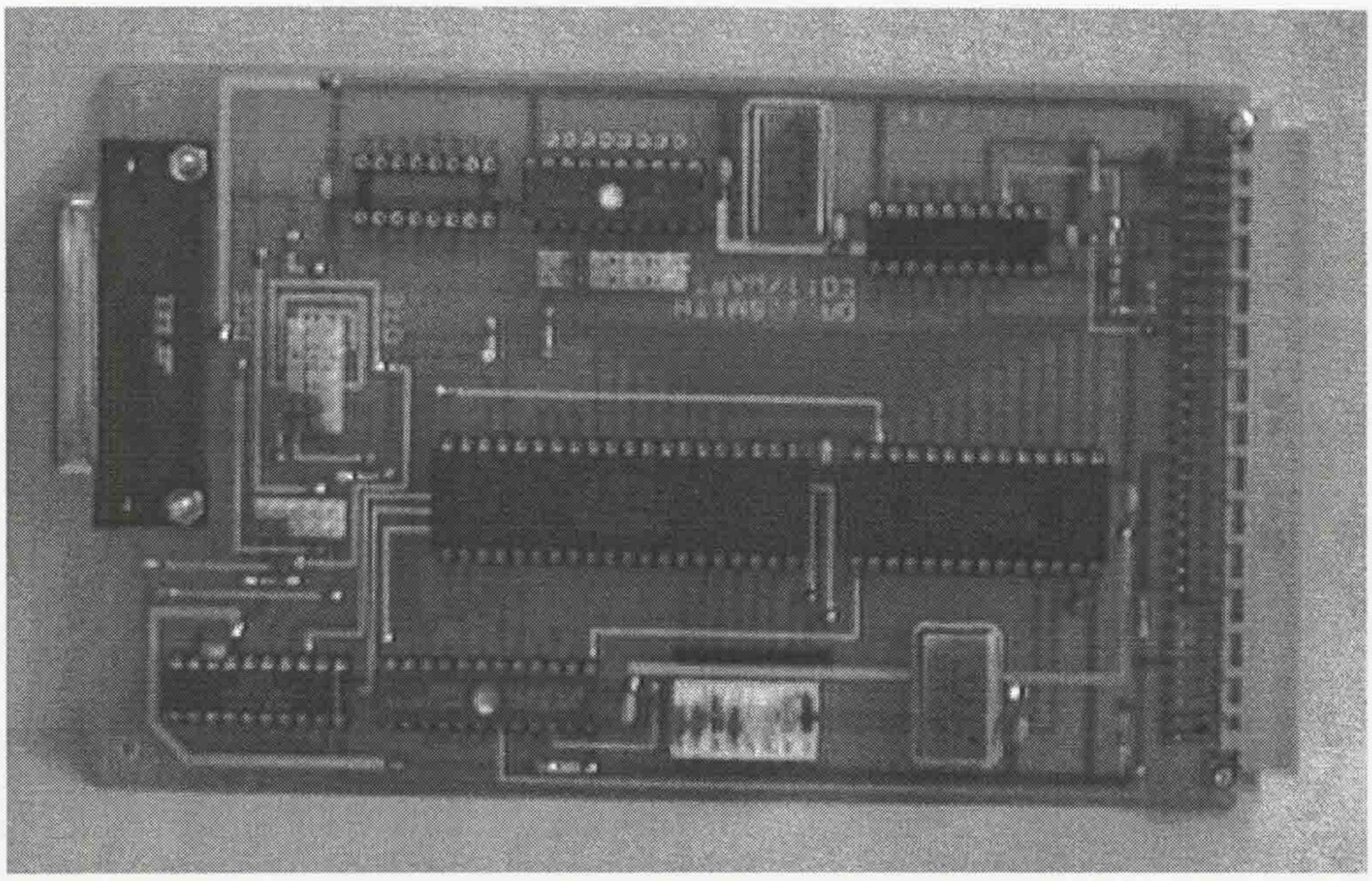

Figure 112, The RS232/NMEA Interface Card

LIVERPOOL UNIVERSITY LIBRARY