Design and Control of an High Maneuverability Remotely Operated Vehicle with Multi-Degree of Freedom Thrusters.

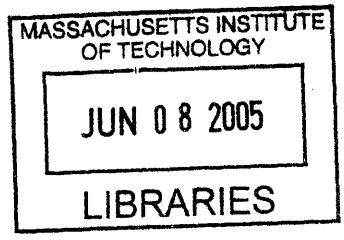
by

Daniel G. Walker

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005



Signature of Author .....
Department of Mechanical Engineering
May 6, 2005

Certified by .....
Franz Hover
Principal Research Engineer
Thesis Supervisor

Accepted by .....
Ernest Cravahlo
Professor of Mechanical Engineering
Chairman, Undergraduate Thesis Committee

Design and Control of a High Maneuverability Remotely Operated Vehicle with Multi-Degree of Freedom Thrusters.

by

Daniel G. Walker

Submitted to the Department of Mechanical Engineering
on May 6, 2005 in Partial Fulfillment of the Requirements for
the degree of Bachelor of Science in Mechanical Engineering

ABSTRACT

This research involves the design, manufacture, and testing of a small, $<1m^3$, <10kg, low cost, unmanned submersible. High maneuverability in the ROV as achieved through a high thrust-to-mass ratio in all directions. One identified solution is moving the primary thrusters in both the pitch and yaw directions. The robot is propelled by a pair of 2 DOF thrusters, and is directly controlled in heave, surge, sway, yaw, and roll. Pitch is controlled through passive buoyancy and, potentially, active manipulation of added mass and gyroscopic effects. This system is compared against a traditional fixed-thruster system in terms of cost, size, weight, and high/low speed performance. Preliminary results indicate that the actuated system can provide an improved thrust-to-mass metric at the expense of increased system complexity. This margin of improvement increases with increasing thruster size. The system has applications in high accuracy positioning areas such as ship hull inspection, recovery, and exploration.

Thesis Supervisor: Franz Hover
Title: Principal Research Engineer

Acknowledgements

The author would like to thank Franz Hover, Tom Consi, The MIT ROV Team, The Department formerly known as Ocean Engineering, and the Ocean Engineering Teaching Laboratory for their continued support, as well as Analog Devices for the donation of accelerometers and rate gyros.

Table of Contents

Table of Figures

Table of Tables

1.0 Introduction

The overarching purpose of this research is to extend the current field of high maneuverability underwater robots. This is an attempt to create a new ROV based on design principles, rather than existing designs. The initial bias is towards a more complex system, as there are already a number of commercially available, relatively simple ROVs.

The robot in this research should be small and highly maneuverable. High maneuverability will be defined as the ability to: hover in one place; adjust position or velocity without turning first; and quickly change direction.

In order to adjust position without turning, the ROV must have active control of every direction in which it is expected to move. For a small inspection-class ROV, the ability to translate in any direction greatly enhances the product usability. Most ROVs of this type have a front camera and gripper payload. The ability to move laterally allows the user to precisely position the ROV payload. User control of pitch and roll are not required for this ROV. This will limit the operations to inspecting mostly vertical walls. The ROV can still be used on complex hull forms, but it will not be able to maintain a constant, normal orientation to the surface.

In order to adjust position or accelerate rapidly, it is desirable to have as little system mass as possible, while having the maximum output force from the thrusters. This is a basic application of Newton's second law. Confounding this, however, stability also effects the system performance. Stability is the system's ability to withstand disturbance inputs. High stability makes precise positioning easier for the user, and is desirable. For this attribute, increased mass is desirable, as a larger mass is more difficult to accelerate by outside forces. However, a similar effect can also be achieved with a more complex control system. The tradeoff here appears to be increased complexity in the control system versus a more sluggish response. Because the control system can be improved later with less effort than reducing mass in the mechanism, a lighter system is selected.

For an actively controlled system, the system should be as small as possible, with thrusters as forceful as possible. For a passive system, the system should be as large as possible, with proportionally larger thrusters.

The generation of numerical specs is somewhat arbitrary, but is loosely based on numbers 1-2X better those from commercially available ROVs, such as those covered in the next sections. The ROV should be approximately $1 ft^3$, have a max speed 5 m/s, be able to change velocity from 1m/s to -1m/s in 1 second, and have a mass of 5kg.

1.1 Background

Unmanned underwater vehicles (UUVs) can generally be divided into long-range and short-range machines. The long-range machines have slender hulls, shaped like a torpedo or modern military submarine. They also usually rely on one large propeller with a series of rudders and control surfaces. These are designed to travel long distances efficiently, and are generally used

for surveying and mapping of large grids in the ocean. The short-range vehicles are more box-like, and have multiple thrusters. These robots are designed to move in several different directions from a standstill. By comparison, the long range robots can only change direction while they are moving. The short-range robots are used in applications such as object recovery, hull inspection, and diver assistance.

In clarification of commonly used acronyms, autonomous underwater vehicles (AUVs) are generally considered to be robots without a tether. Remotely operated vehicles, on the other hand, do have a tether. In industry, AUVs are used to perform long-range tasks, while ROVs are used to perform short-range tasks. Within the prototypical applications, the ROV/AUV definition is logical. However, there are a number of instances where the industry convention can become misaligned with the definition of the acronym. For example, a robot that is remotely controlled by radio or acoustic methods may still be considered an AUV, although it is not autonomous. Classically, remote control is not practical unless using a tether; remote control by acoustic or radio transmissions rarely have the necessary range or bandwidth, so a tether is used. This paper will hold to the industry convention, with clarification as needed.

There are a number of small, maneuverable ROVs on the market. The two most widely used ones, with dry weight less than 25 lbs, are the VideoRay and LBV.

The VideoRay Scout is the smallest platform sold by VideoRay LLC. It has a maximum depth rating of 91m, max speed of 1.9 knots, mass of 3.6 kg, dimensions of 0.35x0.25x0.21m, and costs $5,995.00 including controls and monitors. Higher-end models have greater depth rating, speeds to 2.57 knots, and more operational sensors and accessories. All are based on a three thruster design, with two forward thrusters on the sides, and one vertical thruster in the center.[1]

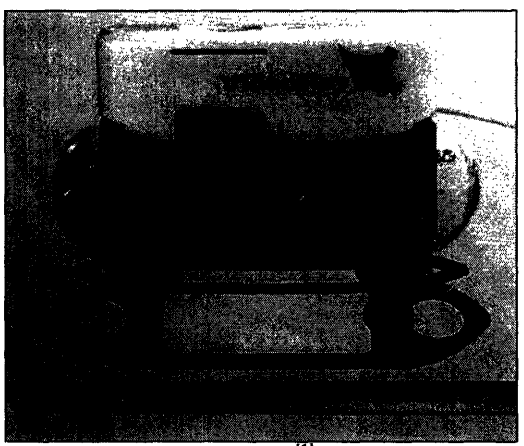


Figure 1: VideoRay Scout [1]

The Sebotix LVB is a similar system made by Cetrax Systems LTD. A 150m rated system starts at $16,182.87 [converted from pounds]. The LBV has 4 thrusters: two forward thrusters, a vertical thruster, and a lateral thruster, for top speeds of 2 knots vertically or laterally, and 4 knots forward. The LBV is larger than the Scout, measuring 0.49x0.26x0.24m and 10kg. Higher end models have depth capacity up to 3,500m [2]

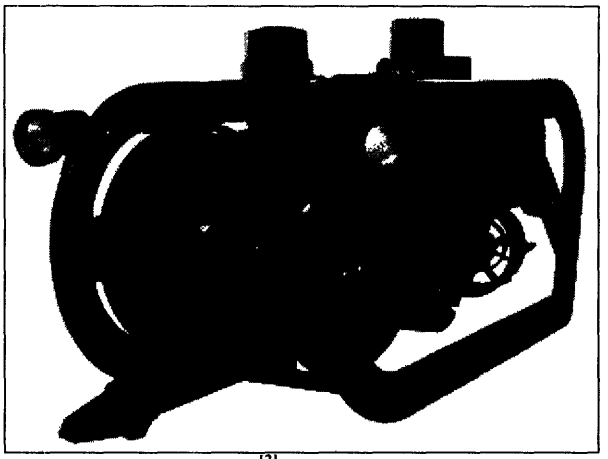


Figure 2: Sebotix LBV [2]

1.2 Relevance of this Problem

One of the traditional applications for underwater vehicles it the mapping of the ocean floor. In addition to basic cartographic data, high resolution sensors can help locate wreckage and archaeological remains. The vehicle typically flies a regular pattern over the floor, taking elevation data as it passes overhead.

Another application for highly maneuverable ROVs is ship hull inspection. This can be part of regularly scheduled preventative maintenance, damage assessment, or searching for contrabands. The VideoRay is already used in searching the ballast tanks of large vessels, saving the cost and danger of draining the tanks or sending in divers. In addition, increased security concerns warrant the inspection of large LNG carriers for attached mines or explosive devices. Commercially available software allows for the detection of anomalies on previously mapped hulls while using a small ROV.

Work-class ROVs can also augment or replace divers for light repairs or coarse manipulation of tools and objects.

The deep oceans are relatively unknown when compared to dry land, or even parts of space. A major contributor to this is the lack of ROVs for ocean-going scientists. For deep submergence, the number of capable ROVs or Human Operated Vehicles is still very small. The National Research Council has identified deep submergence assets as an inadequately met need. However, the development of new ROVs or restructuring of current assets would have a significant cost. [3] The pressure and hostility of the deep waster environment immediately excludes any researcher without access to such a high-end ROV or HOV. Thus ROVs are a basic tool for any such deepwater research.

1.3 Outline

This paper, after the current introduction, analyzes the problem first from a global view. Basic engineering parameters such as power, response time, and geometry are generated from the initial specifications. Next, an overarching design is created to meet these parameters. Each relevant subsystem is designed and analyzed. Next, the system is tested, and performance is measured and compared to the initial specifications. Lastly, conclusions are drawn and future work is discussed.

2.0 Mechanical Design

In a free-flying ROV, the principle axes of motion are surge, sway, heave, pitch, yaw, and roll.

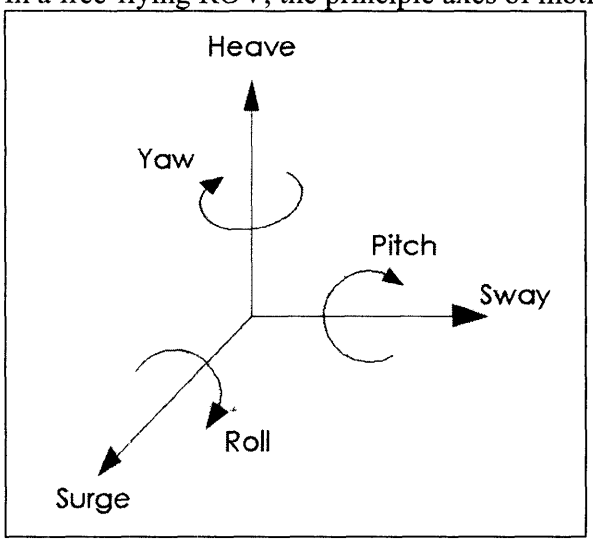


Figure 3: Degrees of Freedom for a Free-Flying Robot

Thruster power is initially suspected as the most sensitive parameter in determining the size and performance of an ROV. Electronics are small and generally constant in size. Power can be supplied through the tether. The only other main component, the payload, creates the performance specifications directly through mass and lift capabilities. Also, payload creates performance specifications indirectly through setting the mission type, such as being highly maneuverable for a hull inspection camera, or having efficient propulsion for long-duration sonar scanning. At the small end of the ROV range, mechanical overhead such as wiring, pressure housings, and structural members also start to play a role in ROV form.

2.0.1 Quantitative Analysis

In order to create a design for the proposed system, the initial suspicion of the thruster as most sensitive parameter is assumed to be correct. Different aspects of the performance specifications are analyzed in terms of the necessary thruster power. A series of different thruster arrangements are created to fulfill the thruster power requirement. Lastly, the arrangements are compared and the suspicion is validated.

The power required from the thrusters are in terms of the total fluid power, and do not consider efficiency losses from the propeller and drive train. These factors are considered during motor selection in section **2.2.4.**

The drag force F on the ROV is determined by the coefficient of drag, C_d, the surface area, A, and the free stream velocity, U.

$$F_{drag} = \tfrac{1}{2} C_d A U^2 \tag{1.1}$$

The drag coefficient is empirically determined using the above relation. where the Reynolds number is a relation between the fluid density ρ, velocity, U, characteristic length, L, and fluid viscosity μ.

$$\mathrm{Re} = \frac{\rho U L}{\mu} \tag{1.2}$$

The Reynolds number for an 0.3m object in water moving at 1 m/s is $3x10^5$, putting the flow in the critical regime.

For initial estimates, the ROV will be modeled as somewhere between a sphere in the laminar regime and a cube. The C_{d_sphere} is then inversely proportional to the Reynolds number for Reynolds numbers less than 10^2, and approximately 0.5 for Reynolds Numbers less than 10^5.[4]

$$\begin{aligned} C_{d\_sphere\_low} &\approx \frac{24}{\mathrm{Re}} \\ C_{d\_sphere\_high} &\approx 0.5 \end{aligned} \tag{1.3}$$

The C_d for a cube is accepted as 1.05 [5] for Reynolds numbers greater than 10^{-1}. Assuming a frontal surface area of $0.1m^2$ (~1 ft^2), or an equivalent diameter of 0.35m, the drag forces are listed in table 1. Assuming that $\rho_{water} = 10^3$; $\mu_{water} = 10^{-3}$

Table 1: Drag Forces on an ROV Hull

Geometric Model	*Drag Equation*	*Drag at U = 0.2 m/s*	*Drag at U = 0.5 m/s*	*Drag at U = 1 m/s*	*Drag at U = 5 m/s*
Sphere Low Speed	$\frac{24 \cdot \mu_{water}(A=0.1)U}{2\rho_{water}(D=0.35)}$	6.85e-7 N	1.71e-6 N	3e-6 N	
Sphere High Speed	$\frac{1}{2}(C_d = 0.5)(A=0.1)U^2$	.001 N	.00625 N	.025 N	0.6875 N
Cube	$\frac{1}{2}(C_d = 1.05)(A=0.1)U^2$	.0021 N	.0131 N	.0525 N	1.3125 N

Using the upper bound cubic model, the max steady-state speed for ROV is a function of the maximum mechanical power output for the thruster motors

$$\begin{aligned} P_{max\_motors} &= F_{drag}(v) \cdot v_{max} \\ P_{max\_motors} &= \tfrac{1}{2} \cdot 1.05 \cdot 0.1 v^2 \cdot v_{max} \\ v_{max} &= \sqrt[3]{\frac{P_{max\_motors}}{0.0525}} \end{aligned} \tag{1.4}$$

To achieve the initially specified max speed of 5 m/s, the thrusters must be capable of supplying

$$P_{max\_motors} = .0525 v^3 = 6.56W \tag{1.5}$$

Drag may limit the performance of the ROV for the specified requirements. For comparison, momentum effects will be examined. The maximum required power output for the motor is rate of energy change required to move the system. Initial requirements specified a change in velocity from 1m/s to -1m/s in 1 second.

$$P_{max\_motor} = \frac{dW_{mechanical}}{dt} = \frac{dE_{mechanical}}{dt} \tag{1.6}$$

$$\begin{aligned} E_{mecanical} &= E_{potential} + E_{kinetic} \\ E_{mecanical} &= \tfrac{1}{2} m v^2 + \tfrac{1}{2} I \omega^2 \end{aligned} \tag{1.7}$$

Because the system is neutrally buoyant, gravitational potential and buoyancy potential energy cancel, leaving only kinetic terms. With constant system mass, the power is

$$P_{max\_motor} = \tfrac{1}{2}(m\frac{d}{dt}v^2 + I\frac{d}{dt}\omega^2) \tag{1.8}$$

Assuming a constant acceleration profile, the specified acceleration requirement is $2m/s^2$. If rotation is held constant, the motion can be modeled as a linear velocity profile.

$$\begin{aligned} &v = 1 - 2t \\ &P_{max\_motor} = \tfrac{1}{2} m \frac{d}{dt}(1-2t)^2 = \tfrac{1}{2} m \frac{d}{dt}(1 - 4t + 4t^2) = m(-2+4t) = -2ma_{max} \\ &\left|P_{max\_motor}\right| = 2 \cdot 2m = 20W \text{ in air} \end{aligned} \tag{1.9}$$

However, underwater, an additional added mass term must be added in order to account for water that is accelerated along with the moving body. Using the cube model for the ROV, the added mass is calculated by summing added mass coefficients over the width of the body. For a cube of side length L

$$m_{11} = 1.51\pi\rho\frac{L^2}{4} \text{ [6]} \tag{1.10}$$

$$m_{added} = \int_0^W 1.51\pi\rho\frac{L^2}{4} dw$$

$$m_{added} = \frac{1.51\pi}{4}\rho L^3 = 1186L^3 = 50.8kg \tag{1.11}$$

The maximum thruster power will now be significantly greater. Revising Eqn 1.8

$$\left|P_{max\_motor}\right| = 4(m + m_{added}) = 4(5 + 50.8) \approx 220W \text{ in water} \tag{1.12}$$

By examining the contributions of drag and momentum to the power requirements, it is seen that the drag specification is less than three percent of the total power requirement. Momentum effects dominate. In order to fulfill the initial specification, however, the ROV must be able to generate this much power in each of the four desired degrees of freedom: surge, sway, heave, and yaw. This is nontrivial.

2.0.2 Iteration of Designs

The previous section developed the requirement for a large power output in each of the four controlled directions. An iterative process is used to create and downselect designs.

In order to generate power in the three translation directions, a thruster must be pointed in that direction at the time of motion. This can be accomplished by a fixed thruster pointed in the direction, or a movable thruster oriented temporarily in that direction. Because the robot moves in all three orthogonal translation directions, the minimum number of fixed thrusters is three. With three orthogonal fixed thrusters, however, the ROV is capable of translating along any vector in the three dimensional space (R^3). Such a fixed-thruster ROV has the ability to immediately generate a translation thrust vector in any direction.

Translation may also be implemented by using thrusters that can move in one or more directions. The minimum number of moving components to achieve 3D translation is three. A single thruster is moved in two orthogonal directions can produce the full space. For a non-over-actuated system (no redundancy), translation components can be thought of as special coordinates. (Fig. 4) Pure translation is possible only when the thrust vector intersects or extends from the center of mass. In the coordinate system analogy, the origin is the center of mass. In this light, the three fixed thrusters are a Cartesian implementation, with magnitudes in Surge – X, Sway – Y, and Heave – Z. Similarly, the single thruster with double actuation can be seen as a spherical implementation. The thruster magnitude provides the radius R, while stacked rotation in latitudinal and longitudinal provides the angles ϕ and θ. Using this discovery to iterate, it is seen that a hybrid system between pure magnitude and rotation can be used: the cylindrical coordinate system. Two thrusters are implemented: a single fixed thruster in the heave – Z direction, and another thruster actuated in a plane normal to the first thruster, providing R from the thruster magnitude and θ from the angle.

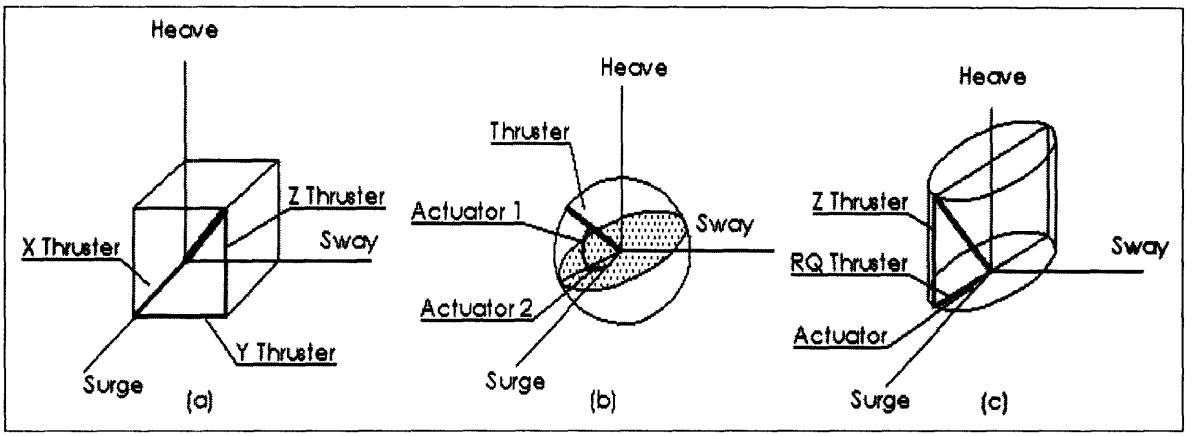


Figure 4: Principal Coordinate Systems and Thruster/Actuator Implementations: (a) Cartesian (b) spherical (c) cylindrical

None of the above systems are capable of producing rotation, however. One additional thruster or actuator must be added in order to achieve the fourth degree of freedom: yaw. For this rotation, the thrust vector must now not intersect or extend from the center of mass. Continuing with exactly-actuated systems, the two boundary designs are apparent. First, using one thruster that is free to move in three directions. Such a design would be a complex combination of translation and rotary actuators. Second, using fixed thrusters: three thrusters are configured as before in the Cartesian arrangement, with an additional thruster parallel to the X or surge thruster, but out of line with the center of mass. A variant on this design is to have two thrusters equidistant from the center of mass. This configuration is found on the LBV. Using reciprocity, however, it is seen that the additional thruster can be anywhere within the XY plane, as long as the thrust vector does not intersect the center of mass. Most conveniently, the alternative configuration has the additional thruster parallel to the Y thruster. This is comprised of two lateral thrusters and one forward thruster. .

Having the two thrusters pointed in the same direction provides the benefit of having twice the thrust capability. This means that smaller thrusters can be used, or additional performance can be gained. Because this is highly desirable, a similar configuration is sought for the two single thrusters. By rotating them 45°, the pair of thrusters is still able to generate a thrust vector anywhere in the YZ plane. In order to generate thrust in the principle directions, however, both thrusters must contribute. To move in the heave (Z) direction, both thrusters must turn on the same direction. To move laterally in the sway (Y) direction, the thrusters must operate in opposite directions: one pulls while the other pushes. The benefit comes with the vector addition of the thrust. Each thruster contributes a projection on the principal axes:

$$\begin{aligned} F_Z &= \sin(45^o)\cdot F_{thruster1} + \sin(135^o)\cdot F_{thruster2} \\ F_{Z\_\max} &= 1.414\cdot F_{thruster\_\max} \end{aligned} \tag{1.13}$$

$$\begin{aligned} F_Y &= \cos(45^o)\cdot F_{thruster1} + \cos(135^o)\cdot(-F_{thruster2}) \\ F_{Y\_\max} &= 1.414\cdot F_{thruster\_\max} \end{aligned} \tag{1.14}$$

In this arrangement, the maximum thrust developed in both directions is nearly one and a half times larger than having both thrusters inline with the principle axes. The cost to the system is in the lost energy from the thrust that cancels out. Twice as much power is required for less than one and a half times the output.

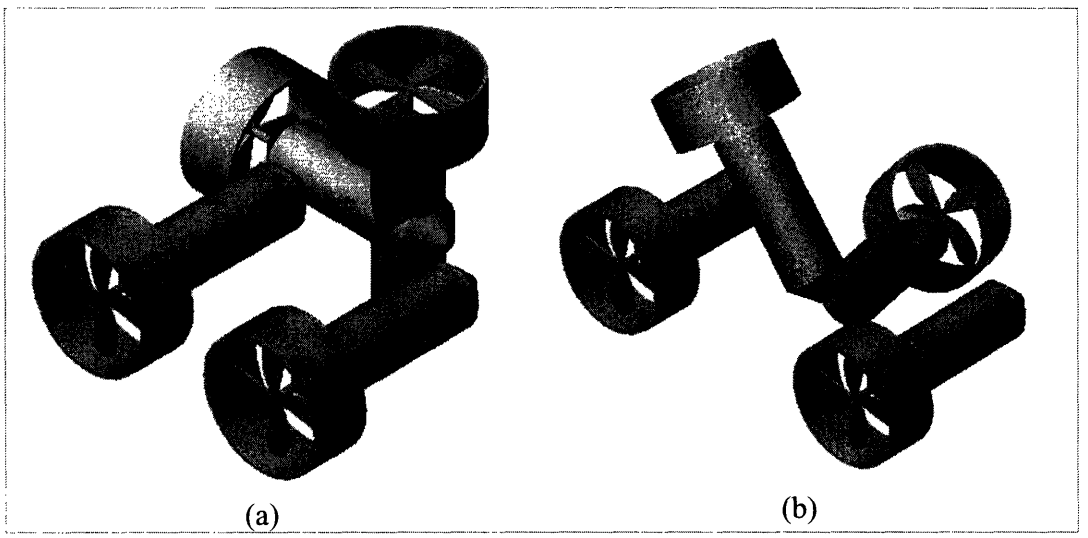


Figure 5: Selected Fixed Thruster Configurations: (a) Orthogonal (b) Vertical V

In between the boundary designs (4 fixed;1 3dof), there are also a number of designs. Although the implementations may vary, the topology is limited to three cases: two fixed thrusters and one 1dof thruster; 1 fixed and one 2dof thruster; two 1dof thrusters. Within these topologies, a number of designs were developed. Most interesting of these is for an ROV with two thrusters, 1dof each. (Fig 6) In this design, one actuated thruster is located inline with the center of mass, providing heave or sway motion, depending on the orientation. The second thruster is actuated in an orthogonal direction, and is located away from the center of mass. When inline with the center of mass, the second thruster provides surge motion. When orthogonal to the center of mass, the second thruster provides yaw rotation.

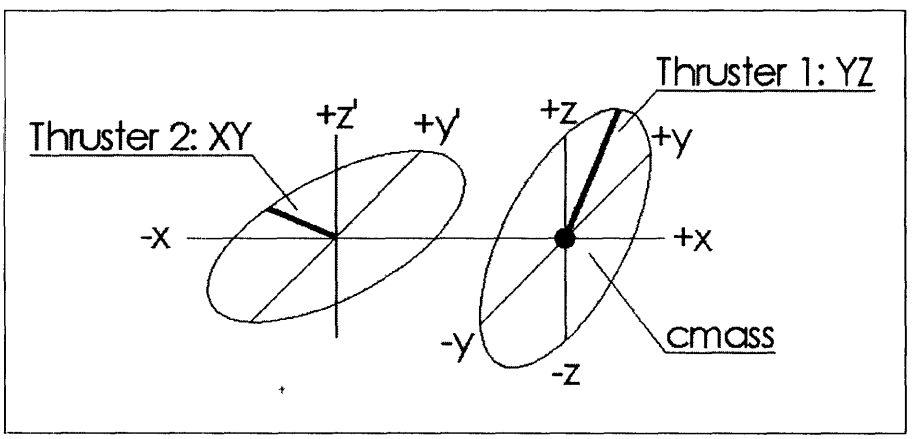


Figure 6: Two Single Degree of Freedom Thrusters

Applying reciprocity, over-actuated systems are now analyzed. In an over-actuated system, the number of actuators exceeds the desired degrees of freedom. These systems are characterized by multiple configurations capable of achieving the same total output vector. This can be beneficial, by allowing the system multiple options to achieve a commanded output. Presumably, one potential configuration will be closer to the current configuration than the other, allowing a faster realization of the command. The control system required to handle this, however, is considerably more complex.

Going to the extreme, we look at possibility of highly over-actuated systems. One design, casually suggested by Dr. Hover during brainstorming, is an robot using surface jets. The design would incorporate thrusting elements over the entire surface of the robot. In one implementation, tubes on the surface would be connected to a central manifold and pumping system. Movement is achieved by pumping in fluid from one half, and pumping out on the other. Rotation could be achieved by integrating surface ports that are not inline with the center of mass: shear thrusters, as it were. With proper design and selection of ports, such a design could be capable of putting out full power in any direction.

Simplifying this idea, but continuing with the same general mechanism, is a robot incorporating a limited number movable surface ports with a central pump. This trades the complexity of the manifold system for the complexity of actuated nozzles at the surface. In order to prevent the need for a manifold, the nozzles are designed to have an 'off' position within the actuation range. This design could also be capable of full power in any direction.

Thrusters are proven components with relatively high efficiencies. A hybrid system is considered by incorporating both thruster and surface port technology. One implementation is considered in detail. (Fig 7) The robot is given a primary direction. Along this axis, a combined thruster/pump mechanism is located. Intakes on both ends of the robot are used to draw in water for the surface nozzles. By switching the direction of one of the intakes, water flows in one intake and out the other. This acts as a thruster, and propels the vehicle most efficiently in the primary direction. By switching this output back to an intake again, water is forced out through the surface nozzles, allowing lateral, vertical, rotary, or combined motions. This design initially appears best suited for fast motion in one direction, with slower positioning and steering in the other directions.

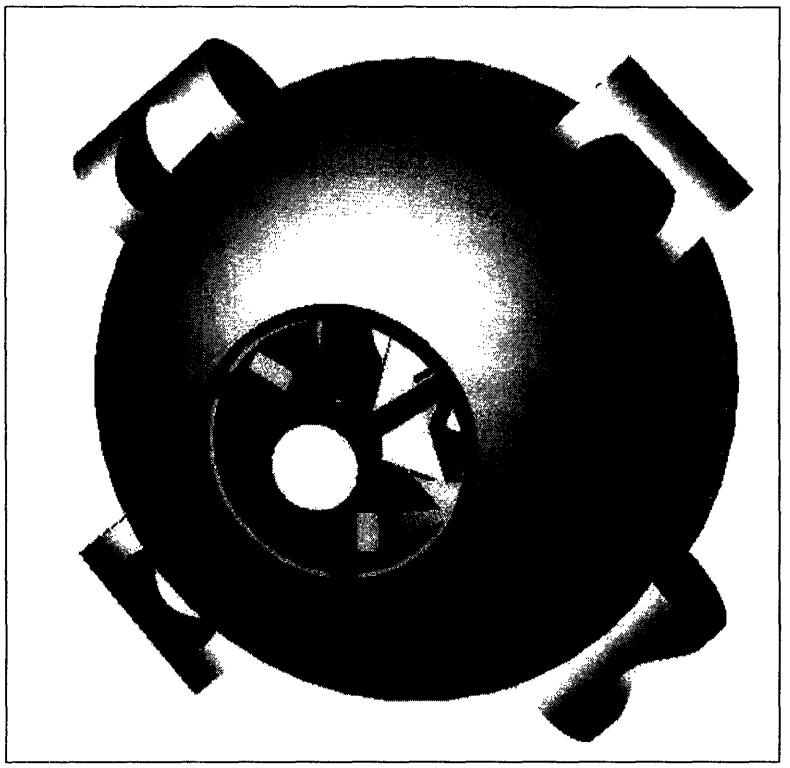


Figure 7: Hybrid Thruster - Surface Port ROV

Moving back to pure thruster-actuator systems, a number of over-actuated systems are examined. First and simplest is a robot using redundant fixed thrusters. For example, an ROV could use two or four vertical thrusters. These could be used to actively control pitch and roll. While the user does not provide input data for these two axes, the computer uses sensors to actively maintain a level orientation. A simpler, over-actuated, arrangement, is to simply send identical control signals to each vertical thruster. The geometry of the robot may make such an arrangement more convenient by allowing fewer thrusters directly inline with the center of mass; only the summed vector from the vertical thrusters must intersect the center of mass. For a heavy lifting mission, it is also desirable to have more vertical thrusters, and so more vertical power. On the opposite end, using a number of smaller thrusters may be less costly than a single large thruster, depending on the thruster design.

Recalling the fixed-V design, a similar design is sought in an over-actuated family. Each thruster should be able to contribute to multiple directions of motion. In order to overcome the power loss from canceling forces, each of the two V-thrusters is actuated, allowing rotation in the roll direction. This allows each thruster to point upwards or sideways for maximum thrust in that direction. The surge/yaw thruster pair remains the same.

In a different design, the surge/yaw thruster pair is replaced by another degree of freedom on each V-thruster. The ROV now has only two thrusters, each capable of moving in full spherical coordinates. (Fig. 8) This design allows each thruster to point in the direction of travel. Neglecting hydrodynamic effects, this design is capable of moving at maximum power in any direction. This arrangement allows for direct control of five principal directions: surge, sway,

heave, yaw, and roll. Pitch is coaxial with the thruster pair in the figure implementation; the ROV cannot pitch because the thrusters have no moment arm in the necessary direction. A complex use of gyroscopic and added mass would have to be used to control this axis. For a set thruster size, additional power can be added with additional thrusters. Adding a third or fourth thruster also allows for direct control over pitch.

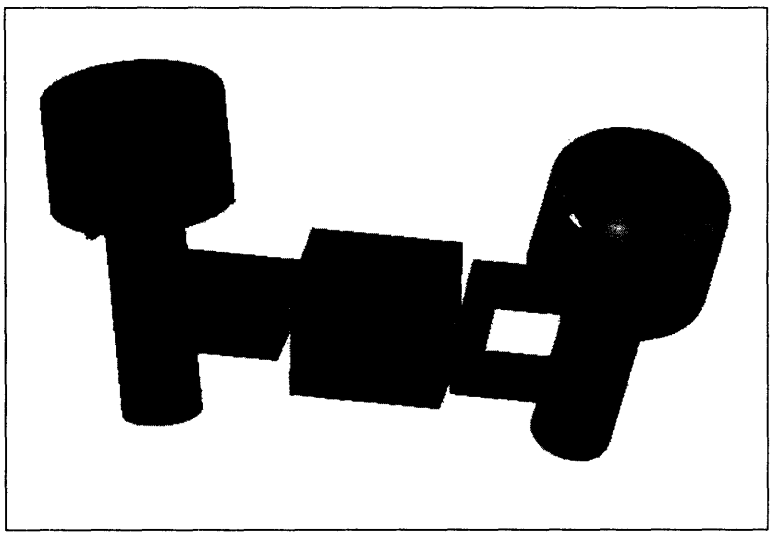


Figure 8: ROV with a Pair of Two Degree of Freedom Thrusters

Leading contenders are selected for evaluation by their relative effectiveness, efficiency, and novelty. The main identified advantages and disadvantages are tabulated below.

Table 2: Strategy Level Design

Design	Advantages	Disadvantages
4 fixed V	Conventional style design incorporates readily available components with existing configurations.	Relatively large wasted power output and unused thrust capability
2 x 2 dof	Maximum thrust available in all directions. Uses existing thruster technology.	Relies on a potentially complex arrangement of bearings structures and actuators.
Surface port	Occupies the least continuous space on the surface area. This allows for more sensors, manipulators, and payload to be on the outside. Uses only one high-power actuator (potentially)	Replaces complexity in thrusters for complexity in valves, pumps, and switches.
Hybrid Surface Port	Modifies existing technology to create a new variant on propulsion. Retains high speed/efficiency capabilities of thruster with low surface area overhead of surface port. Uses few high-power actuators	Potential for large inefficiencies. May not work. Incorporated complexities of both thruster-based and surface-port based.

Analysis of thruster count on system performance:

The first hypothesis is that the mechanisms and structure necessary to create an actuated thruster are negligible compared the thruster itself. This should become increasingly true with thruster size. An actuator for a small mechanism must be built to withstand the same collision forces as a

large mechanism. (e.g. a ¼" strut is much more significant to a ¼" diameter actuator than to a 3" diameter actuator) Thus, small actuators will be overbuilt for operational loads, or those caused by the thruster. A larger actuator has less overhead, as the operational loads are closer to collision loads. In addition, more materials must be used relative to size for the small actuator to accommodate the fixed sized components, such as seals, o-rings, mounting surfaces, and bearings. While special exotic components can combat this, using them will increase system cost.

Although the scaling argument seems to follow logically, it does not prove that the actuator mechanism becomes negligible within a useful range of sizes.

The 2 x 2 design is selected, as it has the potential for drastic performance improvement, while still using existing and proven technology.

2.1 Developed Mechanical Design

After selecting the twin 2dof thruster design, detailed design for the mechanisms was required. The robot is divided by function into four assemblies: thruster, pitch actuator, yaw actuator, and body. The thruster creates the force which propels the robot through the water. The pitch and yaw actuators implement the two degrees of freedom specified. The body holds everything together.

In examining the possibilities for two rotary axes, two seemingly different options appeared. First is the pan and tilt mechanism, typically involving a rotary table and a yoke. Second is a mechanism with concentric rings. [fig.9]. The concentric ring approach has the advantage of working away from the singularity. In the yoke configuration, the system is singular when pointing up. This means that the one of the axes is no longer able to change the position of the output part. In this case, the only differential motion possible is from the yoke. When the thruster is pointing up in the ring configuration, both axes can move the thruster for a full range of differential motion.

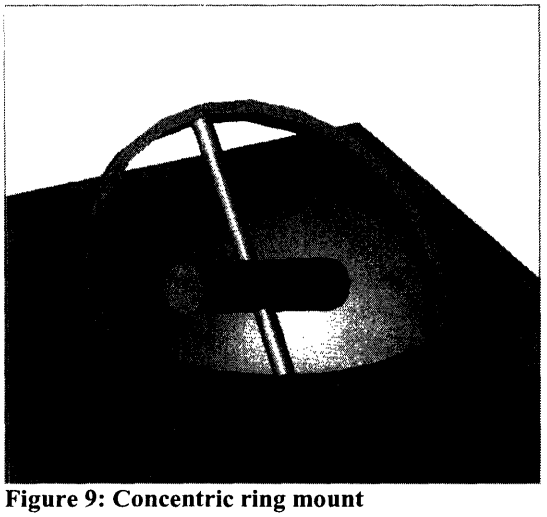


Figure 9: Concentric ring mount

Upon further inspection of the topology, however, it is discovered that the two designs are identical, with the output part mounted differently. In the ring, the thruster is perpendicular to the final axis of rotation, but still lies within the plane of the last ring. The yoke setup is orthogonal to both the axis of rotation and the plane of the yoke. Combining both, the design uses a sideways yoke. The cantilevered structure minimizes the room required while increasing range of motion.

The complete ROV is a pair of these yoke assemblies, joined at the center (Fig.10). No plastic shell or faring is implemented in this prototype.

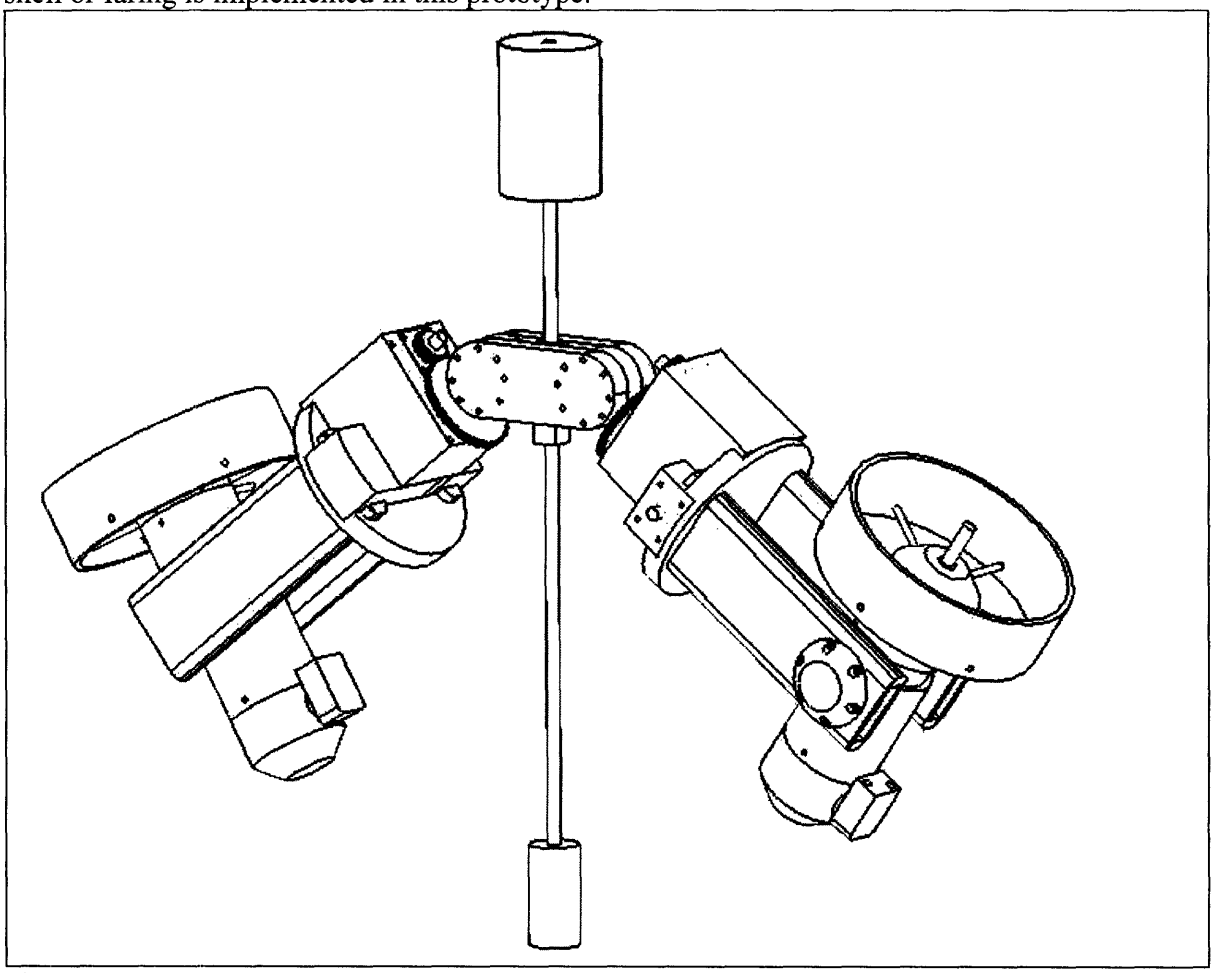


Figure 10: Experimental Implementation of ROV Design

2.2 Thruster Design

The thruster is an assembly that provides thrust by the spinning of a propeller. A thruster typically includes a motor, propeller, and a housing. With the design developed in the beginning of the section, the thrusters for this ROV must have a relatively high power output.

The thrusters should be relatively short, as a minimum to fit within the 0.35m cubic frame. The shorter the thruster, the easier it is to place on or in the ROV. Also, the thruster should be as thin and streamlined as possible, to minimize the drag, and more importantly, the added mass.

The thruster works by spinning a propeller on the order of 500 RPM, so the bearing housing must be able to constrain any shaft. While the operation range cannot be considered high-speed, a reasonably high radial stiffness is still required to account for the high possibility of eccentric loading. This loading could occur from deformities in the propeller or additional material catching on or sticking to the propeller. With a diameter of 4-6", any eccentric mass can cause considerable loading. The primary loading, however, is in the axial direction; the propeller works by pushing the water away in the axial direction.

The entire assembly must be waterproof. The dynamic seal must have as little friction as possible, while still maintaining a reliable seal. The static seals can have as much friction as necessary, with the emphasis on seal reliability over ease of assembly.

2.2.1 Thruster Implementation Overview

The main components are the motor and gearbox, the front spindle, back cap, main tube, propeller, and cowling. The motor and gearbox provide the necessary rotational velocity to spin the propeller in the water. The front spindle is the bearing housing, and constrains the propeller shaft in all directions except inline rotation. The main tube and back cap form the pressure housing with the front spindle. The propeller provides thrust when spun, and the cowling decreases wasted thrust from the propeller.

2.2.2 Front Spindle

The front spindle of the thruster assembly constrains the propeller in all directions except for rotary motion. It also couples the propeller drive shaft to the output of the planetary gearbox. This allows the propeller to spin without applying off-axis or thrust loads on the motor.

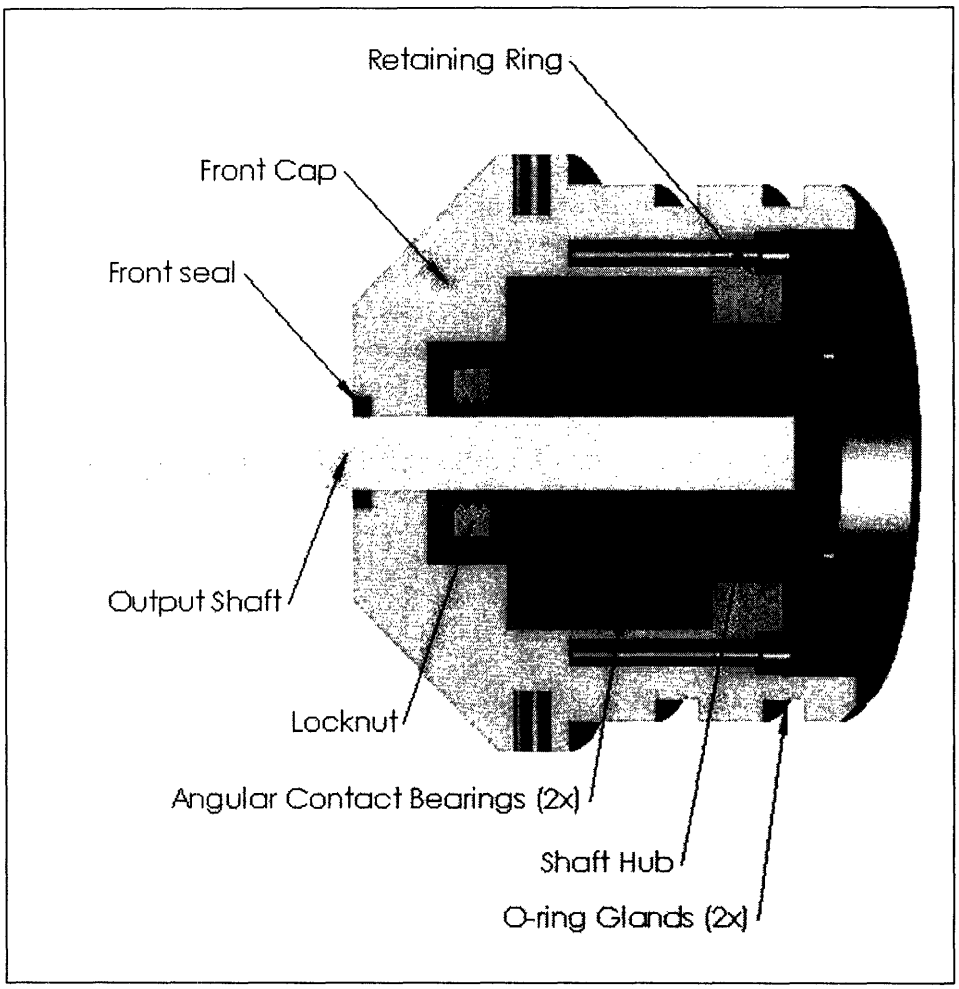


Figure 11: Section view of Front Spindle

The main components of the front spindle are the front cap, two (2) angular contact bearings, retaining ring, spindle hub, spacer, and locknut. The front cap is a polycarbonate housing for the front spindle. It also incorporates the front dynamic seal, and two o-ring glands for the static tube seal. The front cap has a sliding fit (+0.002") for the two bearings, providing constraint in the radial and angular directions. No constraint is required in the inline-rotation direction, as the bearings do not transmit load in this direction. A modest holding torque is supplied in the direction by the sliding fits and axial constraint.

The angular contact bearings are used in a back-to-back arrangement. This allows for increased angular stiffness especially when the bearings are not spaced apart. The lines of contact for the bearings intersect the shaft approximately 1" apart, giving the effective angular stiffness of a pair of radial bearings spaced 1" apart. Because length of the assembly is a major concern, angular contact bearings proved a reasonable solution. The angular contact bearings also provide resistance to thrust loading, and so eliminate the need for a separate pair of thrust bearings. (In the thruster, the primary loading is in the thrust directions).

The outer races of the bearings are retained in the axial direction by the front cap and the retaining ring. The retaining ring has a 6-bolt circle which forces the races into the front cap. A spacer is placed between the outer races to allow pre-tensioning of the bearings via the inner race.

The spindle hub connects the inner races of the bearings to the propeller shaft and the gearbox output. The back is splined to fit into the gearbox output. Because the gearbox does not have integrated bearings, a flexible coupling is not necessary or desired. The inside of the hub is a press fit (-0.005") for the 0.25" propeller drive shaft. The outside of the hub is a sliding fit (+0.002") for the bearing inner races. The front of the hub is threaded to accept a 3/8-24 lock nut. The inner races are retained in the radial and angular directions by the sliding fit. They are retained axially in the back by the shoulder connecting the sliding fit to the spline. They are also retained axially in the front by the locknut. The spacer between the outer races leaves a small space between the inner races. The bearings are pre-tensioned by tightening the locknut and closing this space. Pre-tensioning removes internal clearances in the bearings and improves stiffness and resistance to shock loading.

2.2.3 Motor Mount

The motor is a 150W drill motor from Black and Decker. The original case and clutch were stripped, leaving only the motor and gearbox.

The main tube is machined to a close fit on the gearbox (+0.005"), aligning the gearbox output with the spindle hub. The motor is attached to a mounting block, which aligns the motor with the gearbox input. In addition, the mounting block assures that the motor gear is axially aligned with the planet gears in the gearbox. The keyway on the gearbox is ground using an electric die grinder, as the hardened steel cannot be cut or filed. The keyway on the motor mount is machined to an interference fit (-0.005" for this plastic). The keyway on the inside of the main tube is broached using a custom bushing for the 1.5" ID.

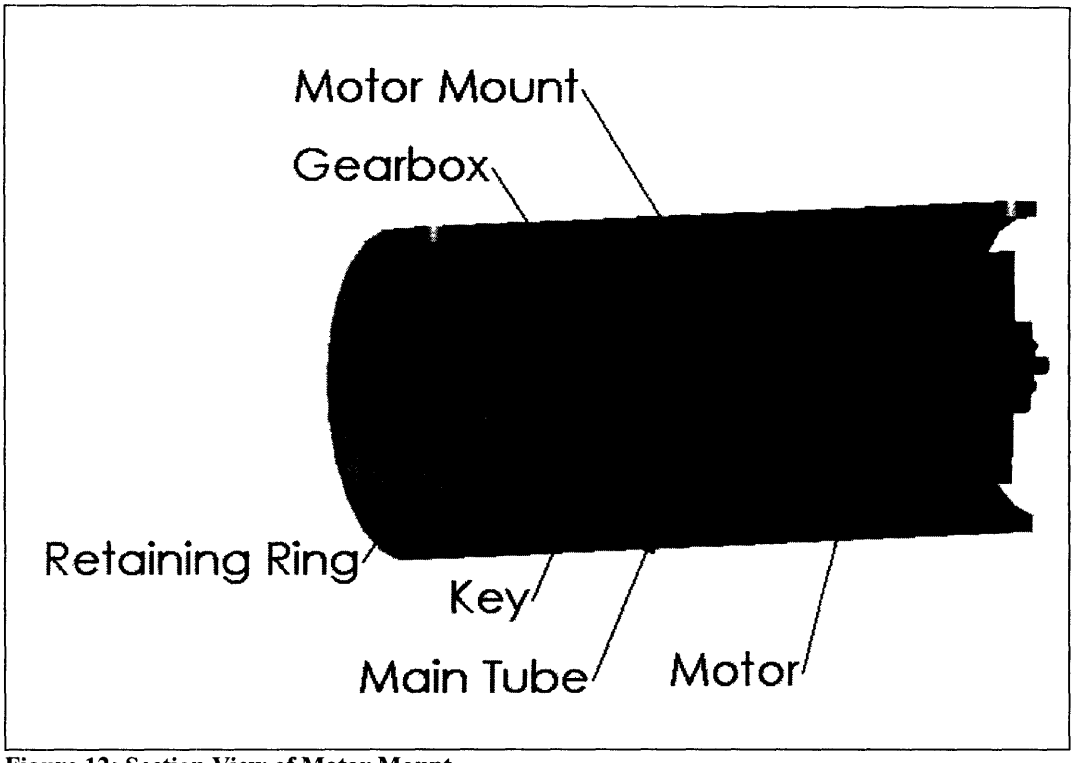


Figure 12: Section View of Motor Mount

The motor mount is constrained radially by a sliding fit (+0.002"). It is constrained axially by a slight shoulder from behind, and by the gearbox in front.

2.2.4 Motor Selection

The mechanical power calculations in **[2.0]** call for a total mechanical power output of ~225W. The output power of the motor must be greater than this, in order to compensate for losses in the power transmission. Table [3] calculates the combined efficiency and the scaling factor required as a result of the efficiency losses. The scaling factor is calculated both before and after the electromagnetic stage, as some prospective motors have power ratings for mechanical output, while others have power ratings only in terms of electric power consumption.

Table 3: Thruster Efficiency

Stage	Efficiency
Electromechanical (EM)	0.80
Gearbox	0.80
Shaft Seal	0.95
Propeller	0.33
Total	0.201
Scaling Factor (w/ EM)	4.98
Scaling Factor (w/o EM)	3.99

Within the power target and budget, a pair of drill motors was selected. They are rated at 300W mechanical combined, running at 7.2V. The system will operate at 12V, allowing a maximum power draw of 800W. Under these conditions for a sustained period, the motor will overheat and potentially melt. A conservative assumption is made that the system will operate at a 50% duty cycle. This brings the motor to 400W output, 33% higher than nominal. In all likelihood the motors will operate closer to 20% duty cycle.

2.3 Pitch Actuator

The pitch actuator is the top of two stacked actuators controlling the thruster. Because the Yaw actuator must move the pitch mechanism. This means that the pitch mechanism should not have excessive mass. The actuator also has parts that stick out of the hull into the water. This means that the upper structure should be as streamlined as possible, to reduce drag and added mass effects. The structure must still be stiff, however, as the entire thrust load from the thruster must be transferred through the actuator to the ROV.

2.3.1. Structure

In order to minimize the drag on the structure in flight, the thruster is mounted on top of two slender spars. These put the thruster away from main hull, giving the thruster a greater range of useful outputs.

The structure is loaded under three conditions. First, the thruster applies a reaction force onto the structure, inline with the thruster. Second, when the thruster is being rotated, the mass and added mass effects apply a torque to the structure inline with the axis of rotation. Third, the assembly contacts foreign objects, applying forces or torques.

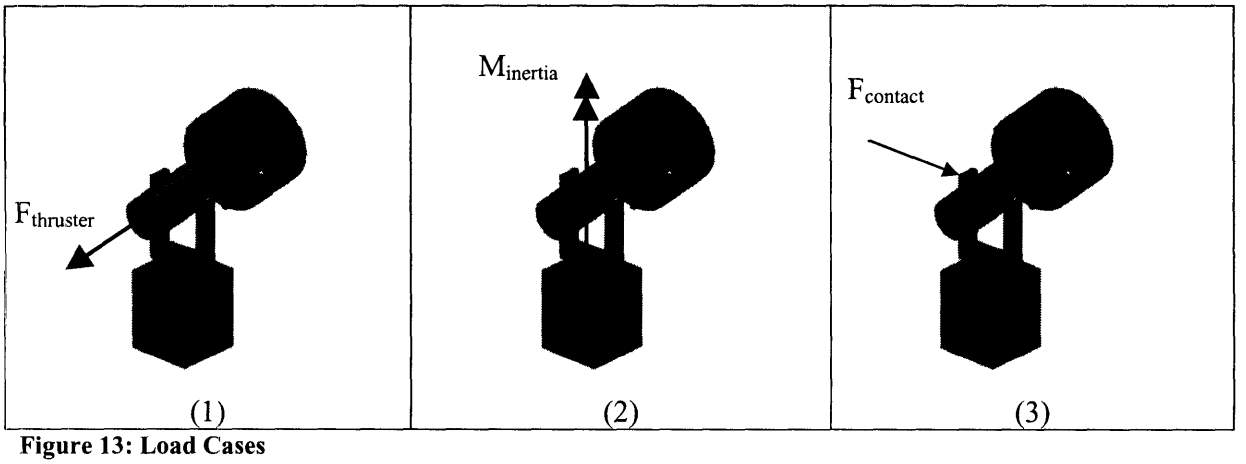


Figure 13: Load Cases

Under the first loading condition, the force is transferred from the thruster to the spar bearings in the radial direction. The spars then take the load in the long direction. Both of these are the stiffest cases. Normal thruster loading sees the stiffest structural response. Under the second

loading condition, the force is transferred from the thruster to the spar bearings, again, in the stiff radial direction. The spars each take the load in the stiff long direction. Each spar takes the load in an opposite direction, though. An additional moment load is also experienced by the spars, however. The slender spars are relatively compliant in torsion. The third load case acts as a lump condition for all remaining loads. Most notable of these cases is loading the thruster across the spars. This transfers force axially through the bearings and to the thin dimension of the spar. This is the most compliant arrangement.

2.3.2. Drive System

The pitch actuator is driven by a timing belt and pulley system. This design allows the motor to be located in the base of the structure, inside the hydrodynamic profile of the vehicle. The belt is located inside one of a pair of slender spars. Only a small bearing, pulley, and pulley housing are required at the output shaft at the top of the spar.

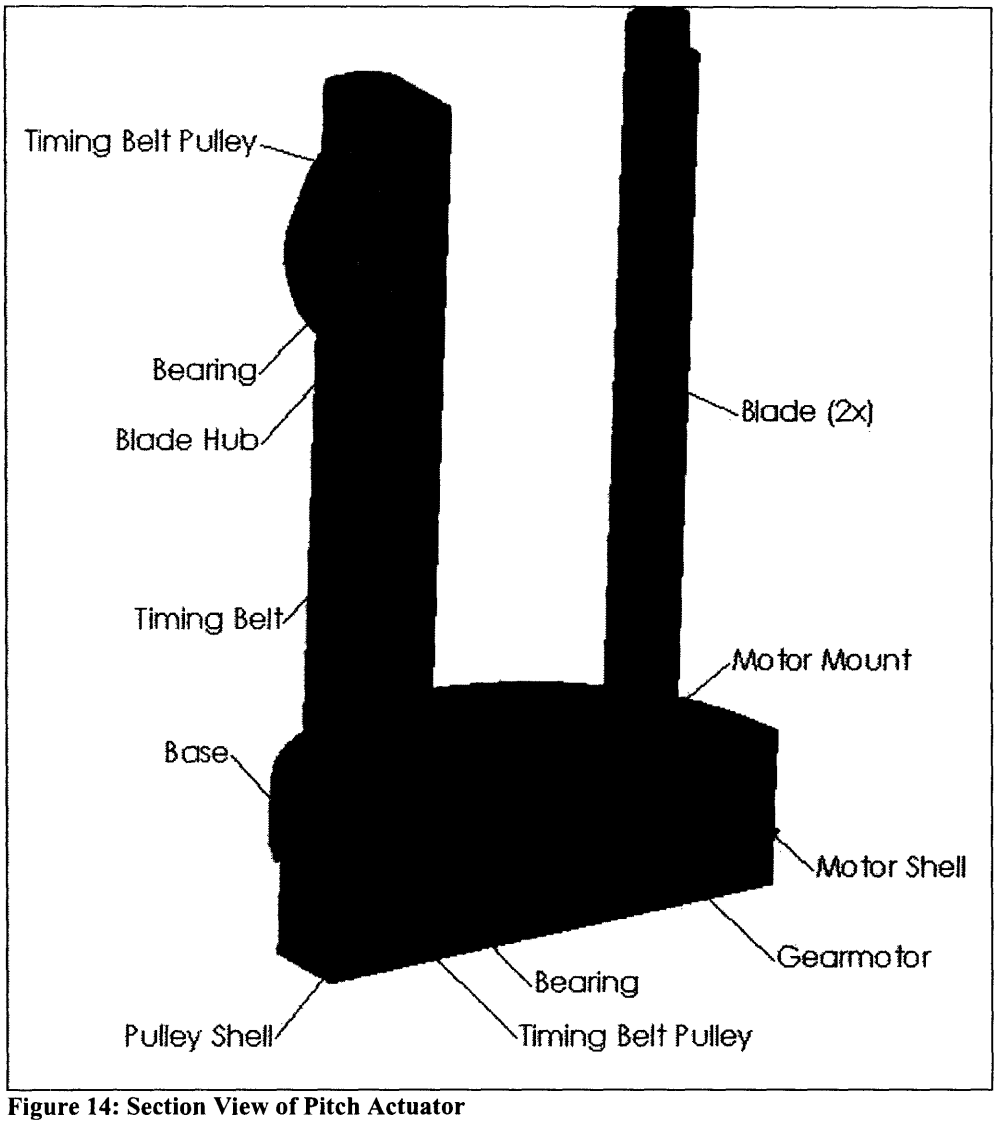


Figure 14: Section View of Pitch Actuator

The lower timing belt pulley shaft is constrained by a pair of radial bearings. The load is purely radial, so no pre-tensioning of the bearing assembly is needed. The pulley is fixed to the shaft. One bearing is constrained between the pulley and the pulley shell. The pulley shell is a detachable cover which contains a bearing seat. That bearing is axially constrained between the seat and the pulley. It is radially constrained by a sliding fit (+0.002") on the outer race. The other bearing is constrained between the motor mount and the pulley. As with the first bearing, the second bearing is constrained axially in one direction and radially by a sliding fit bearing seat. The other axial direction is constrained by the pulley. The pulley shell is fastened to the motor shell by four long 6-32 machine screws. The housing was blanked and machined as a single piece, then cut and finished. This minimizes the problems from misalignment.

The upper timing belt pulley is attached to a shaft press fit into the thruster. One radial bearing in each blade constrains the thruster in the radial direction. Spacers are used to constrain the shaft in the axial direction between the fixed components.

2.4 Yaw Actuator

The yaw actuator needs to be stiffer and stronger than the pitch actuator, because it is the bottom in a pair of stacked actuators. While the pitch actuator only needed to control the mass of the thruster, the yaw actuator must control the mass of both the pitch actuator and the thruster. Also, any angular deflection from the bearings or structure will cause a large deflection at the thruster end of the structure; this is a sine error.

2.4.1. Structure

The yaw structure is manufactured from a single block of acetal. This ensures that the gears are properly spaced, and are not subject to variation from assembly and fastening. The housing block is fastened to the pitch actuator by a set of countersunk machine screws. On the other side, the output shaft is mounted to the anhedral connector (more below) with a press fit.

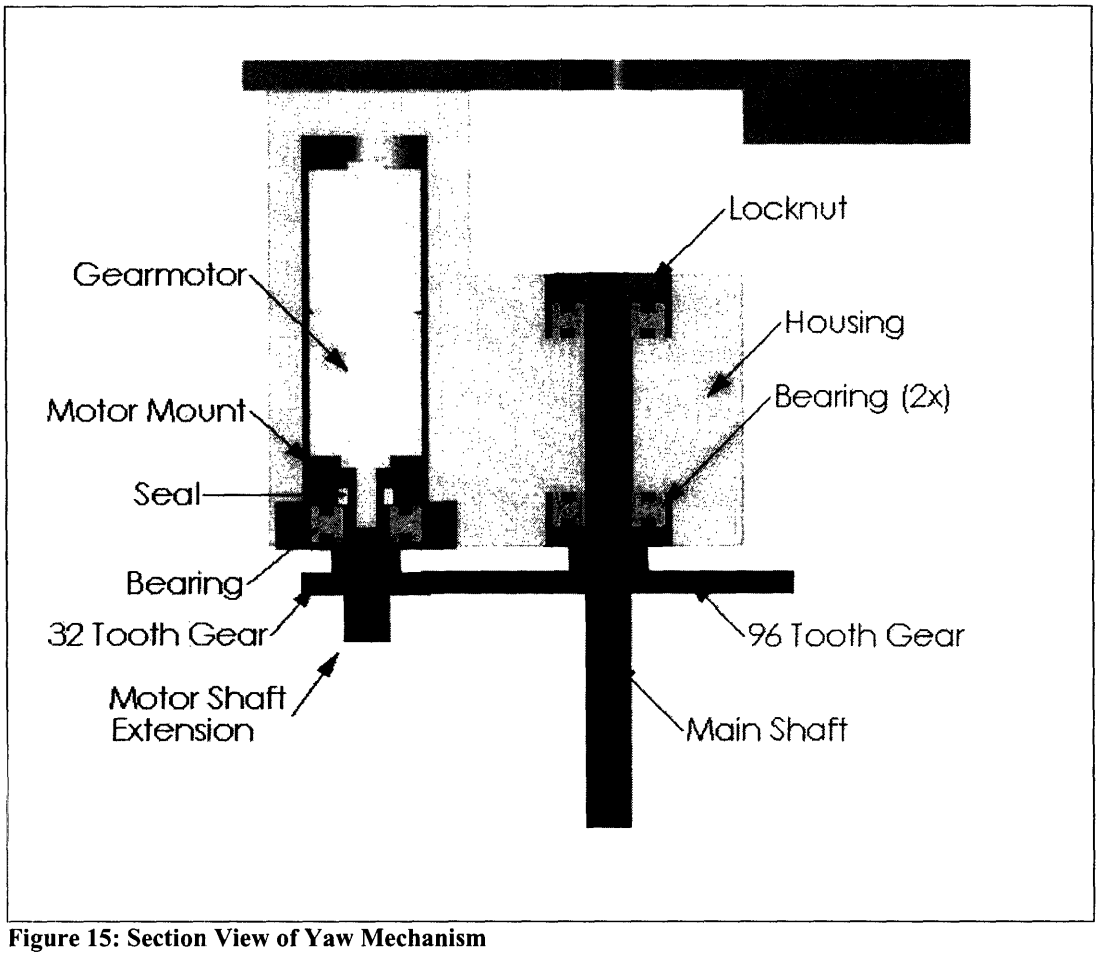


Figure 15: Section View of Yaw Mechanism

2.4.2. Drive System

The drive system is a simple spur gear reduction. Both the drive and driven gear are cantilevered out from the housing block. The output shaft is constrained in the radial and angular directions by a pair of radial bearings. It is constrained axially in one direction by the 96 tooth gear press fit (-0.002") on the shaft. In the other axial direction it is constrained by a locknut. The locknut allows for variable preloading in the axial direction. These ball bearings are not designed for heavy preload, so the applied preload is just enough to eliminate the backlash in the axial direction. The bearings are constrained in the radial and angular directions by a sliding fit (+0.002") between the outer race and the housing. They are each constrained axially in one direction by the counterbored surface of the housing. They are constrained in the other axial direction by the shaft assembly.

The motor shaft extension is press fit (-0.002") directly onto the output shaft of the motor. This constrains the shaft in the radial, axial, and angular directions. An additional bearing is mounted on the shaft extension in order to reduce the moment loading on the motor. This causes the design to be overconstrained. The additional room required to implement a flexible coupling would make the assembly too large to comfortably fit in the system. In order to mitigate the overconstraint, the motor mount is carefully machined, with additional allowances in the screw holes and in the bearing seat. This allows the structure to resist loads while having a degree of compliance to reduce the damage that any misalignment causes.

2.5 Body

The body connects the two thruster assemblies together. Under mission conditions, it would incorporate a selection of sensors and payloads. Instead, it contains dummy ballast and floatation.

2.5.1 Anhedral Connector

Anhedral is the angle from horizontal down to the thruster mounts. This angle is used in order to optimize the range of motion use, in conjunction with modifications to the control scheme.

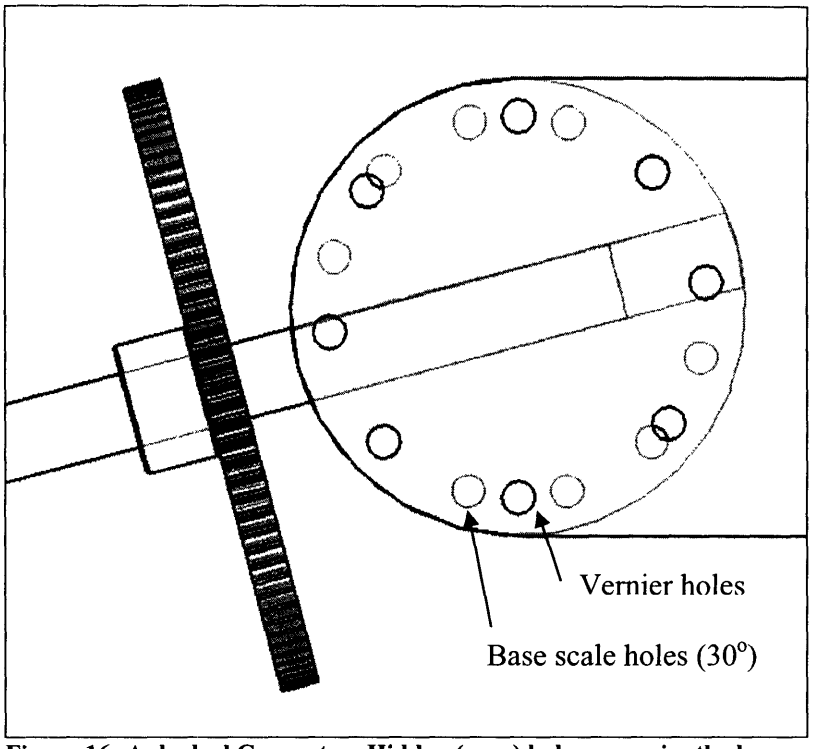


Figure 16: Anhedral Connector: Hidden (grey) holes comprise the base scale in the yaw half. Visible (black) lines comprise the vernier holes in the fixed half.

The anhedral connector allows the selection of different anhedral angles for the thruster assemblies. It uses a vernier type hole arrangement to allow angles from +90 to -90 in 7.5° increments. The half connected to the yaw mechanism has holes drilled at 30° increments. The stationary half has a pair of holes at $\pm$90°. This allows positioning at the base scale of 30° increments. A second pair of holes is drilled at -45° and 135°. This allows positioning at the base scale +15°. A final double pair of holes allows positioning at the base scale $\pm$7.5, or +7.5 and +22.5°.

2.5.2 Ballast and Floatation

A bar extends from the main crosspiece vertically both up and down to mount the buoyancy and ballast, respectively. The bar is manufactured from a 1" piece of sheet aluminum bent into a U-shaped cross-section. Lead weights are used for ballast. Syntactic foam is used as floatation. Syntactic foam is a special hard foam filled with glass bubbles. Although it also has applications in aerospace and rapid prototyping, most syntactic foam is designed as floatation for submarines, offshore rigs, and buoys.[7]

This system passively maintains orientation in pitch and roll. In a neutral position, the ballast and flotation are inline with the center of mass and gravity. Any deviation in the pitch or roll directions forces the floatation and ballast to move out of line with the center of mass and gravity. This creates two moment contributions: one from the ballast, and one from the floatation. The two contributions sum to create a restoring moment, bringing the system back to a neutral state.

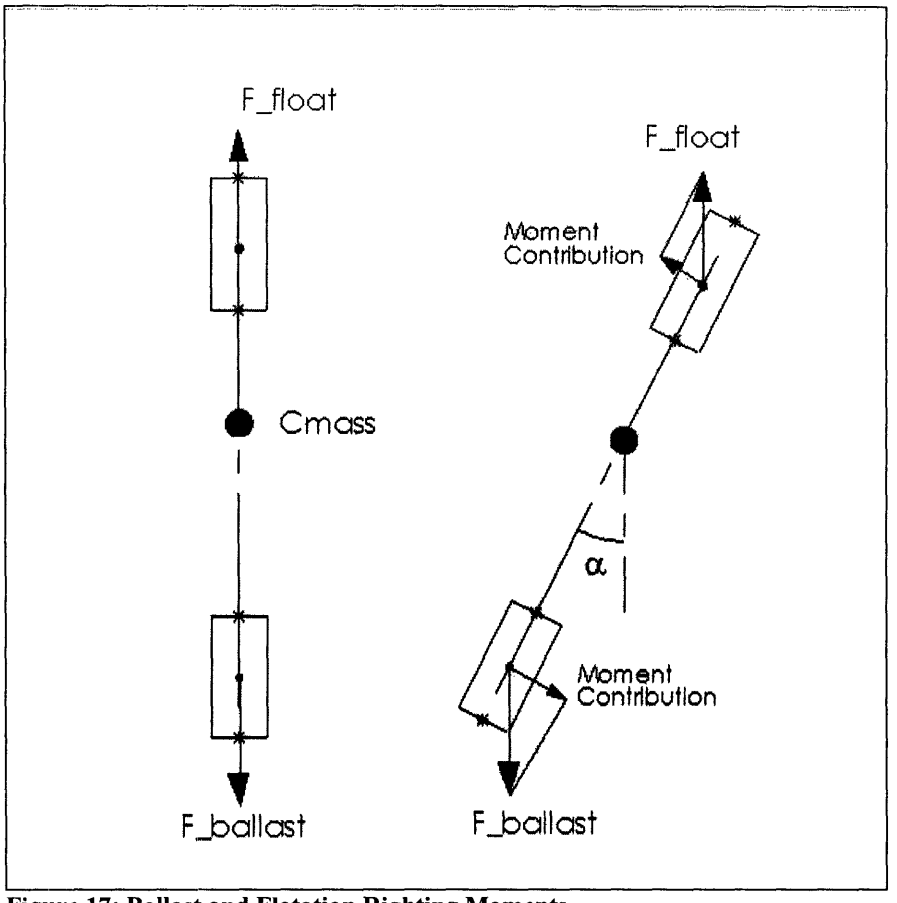


Figure 17: Ballast and Flotation Righting Moments

3.0 Control System Design

The control aspect can be divided into several basic parts. The first part is the servo controller that takes in command angles for each joint, and directs the motors. The second is the kinematic controller that converts high level direction input (move up, down, etc) into the necessary joint angle commands. Last is the user interface that converts human input into high level direction input.

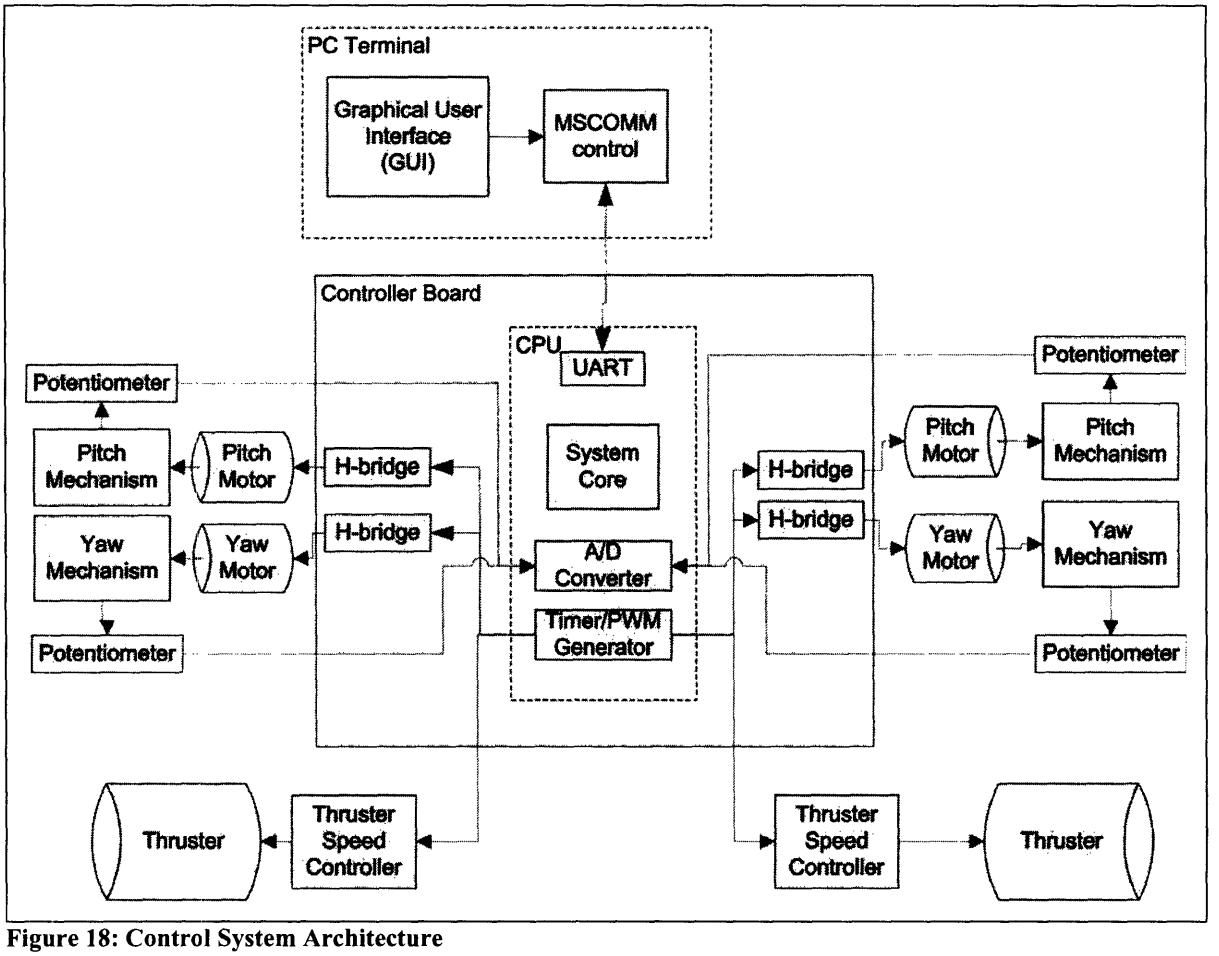


Figure 18: Control System Architecture

3.1 Mechanism Servo Control

The servo controller is the basic feedback loop for the motors, controlling output position. The loop is a PID control loop. Initial estimates of parameters were calculated by geometry. Final parameters were tuned in the operational environment.

Both of the positioning axes are controlled with DC gearmotors, and will have control models. Motor parameters are calculated based on the manufacturer's torque-speed curve (Fig.19). The

torque constant, K_t is equal to the back EMF constant, K_e in SI units, by conservation of energy. This is assumed true for all motors. A single constant, K_e is used in this analysis. This parameter is derived as the relation between voltage and speed.

$$K_e = \frac{V}{\dot{\Theta}} = \frac{6}{210} = 0.028 \frac{V}{RPM} = 0.003 \frac{V}{rad/s} \tag{1.15}$$

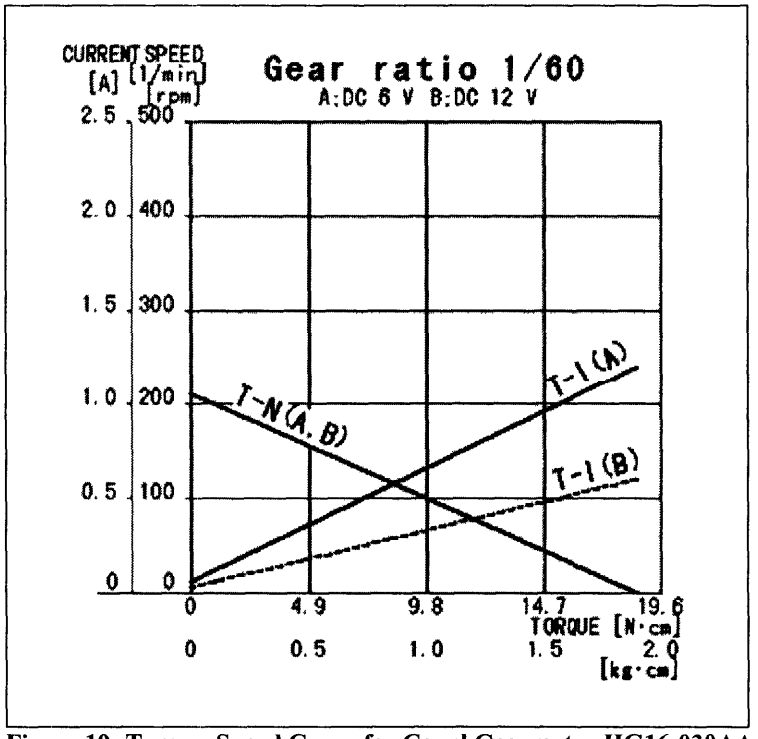


Figure 19: Torque-Speed Curve for Copal Gearmotor HG16-030AA [8]

The motor is controlled using voltage, because this is easy to generate with PWM hardware. The alternative, controlling with current, is significantly more difficult to implement in hardware. The motor acts as small feedback loop (Fig. 20). First, the back EMF is subtracted from the input voltage. The back EMF, V_{EMF}, is related to the output speed Ω through the motor constant K_e.

$$V_{EMF} = K_e \Omega \tag{1.16}$$

The difference between the input voltage and the back EMF is the error signal in the feedback loop, or the remaining voltage which will cause a change in the output of the system. This is fed through the equivalent resistance R and inductance L of the motor to create a current I. Calculation in the Laplace domain greatly simplifies the problem.

$$I_{motor}(s) = \left(V_{input}(s) - V_{EMF}(s)\right) \cdot \frac{1}{Ls + R} \tag{1.17}$$

The torque output from the motor windings is related to the current through the windings again by the constant K_e.

$$T_{motor}(s) = I_{windings}(s) \cdot K_e \tag{1.18}$$

Any disturbance torques, as from the load, are subtracted from this motor torque. The remaining, or net torque, is applied to the mechanical system with an equivalent moment of inertia J and viscous damping C. The output velocity is then

$$\Omega_{motor}(s) = \left(T_{input}(s) - T_{disturbance}(s)\right) \cdot \frac{1}{Js + C} \tag{1.19}$$

The new output velocity feeds back to the initial voltage difference through the motor constant K_e.

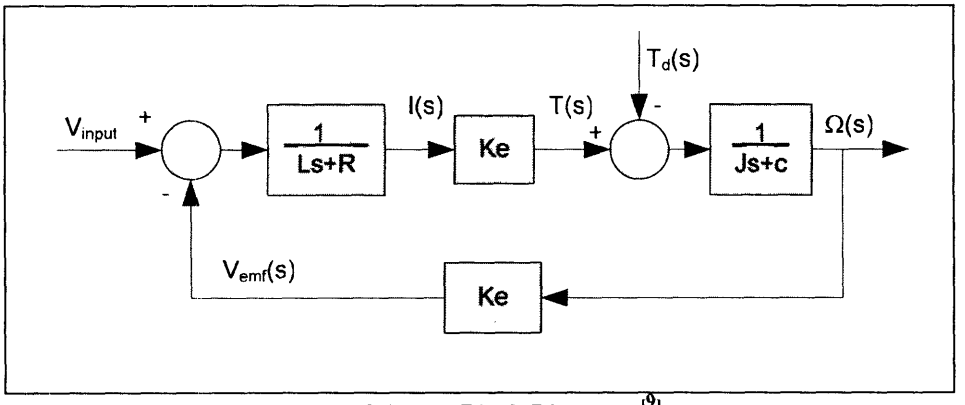


Figure 20: Voltage Controlled DC Motor Block Diagram [9]

The voltage controlled DC motor is part of the larger servomechanism. The output velocity is reduced with a 1:3 gear reduction, symbolically 1:N. The velocity is also integrated by the physical system to position, the variable being controlled. For ease of operation, the gear reduction and integration are modeled as taking place on the motor shaft. A degree of compliance is assumed from the motor shaft to the output shaft. (Fig 16)

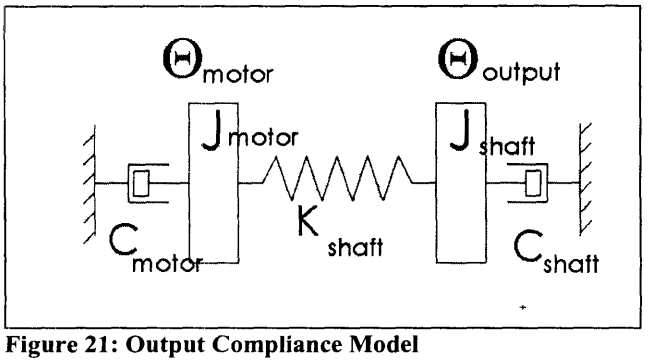


Figure 21: Output Compliance Model

The compliance is modeled as a spring between the position of the motor and the position of the output shaft. The torque on the output shaft is proportional to the difference in angular position for the two shafts.

$$T_{ouput}(s) = K_{shaft\_spring} \cdot \left(\Theta_{motor\_shaft}(s) - \Theta_{output\_shaft}(s)\right) \tag{1.20}$$

This torque is equally applied on the output and motor shafts. The torque applied on the motor shaft takes the form of the disturbance torque into the voltage controlled DC motor. The magnitude is reduced by the gear ratio 1:N. The torque applied on the output shaft acts on the output mechanical system, with equivalent moment of inertia J and viscous damping C. The velocity output is integrated once to yield the position of the output shaft.

$$\Theta_{output}(s) = T_{output}(s) \cdot \left(\frac{1}{Js + C}\right) \cdot \left(\frac{1}{s}\right) \tag{1.21}$$

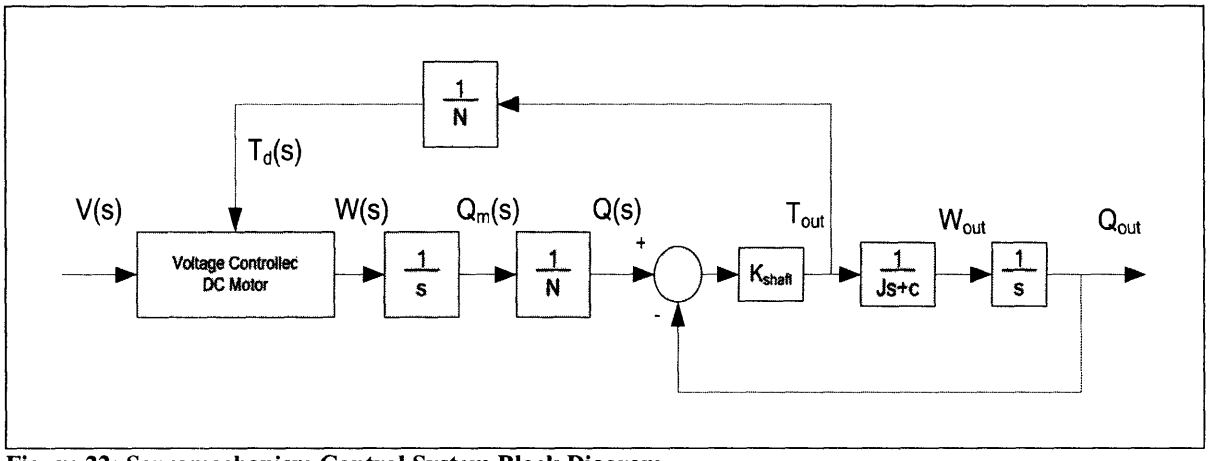


Figure 22: Servomechanism Control System Block Diagram

For both the pitch and yaw mechanisms, it is desirable to develop approximations for the motor and shaft inertias, damping, and stiffness. K_e was found earlier. The inductance L and resistance R for the motor can be directly measured with an LRC meter. The rear reduction ratio is fixed by design. The motor shaft inertia J is approximated as a L = 10cm R = 0.5cm steel (ρ = 7.85g/cm^3) shaft.

$$\begin{aligned} J_{cylinder} &= \tfrac{1}{2} MR^2 \\ J_{shaft} &= \tfrac{1}{2}\left(\rho_{steel} \cdot \pi R^2 L\right) R^2 \end{aligned} \tag{1.22}$$

The damping is measured by recording the no-load torque required to spin the shaft. This is a low-accuracy term. The output shaft inertia is calculated in SolidWorks. The inertia of the yaw direction varies with the position of the thruster in the pitch direction. This is because the pitch direction controls the distribution of thruster mass in relation to the yaw axis centerline.

Table 4: Approximated Servomechanism Parameters

Parameter		Pitch	Yaw
K_e	[V s/rad]	0.003	0.003
L	[H]	0.005	0.005
R	[ohm]	6	6
N	[ ]	3	3
J_{motor}	[kg m^2]	$9.36x10^{-9}$	$9.36x10^{-9}$
C_{motor}	[Nms/rad]	$1.0x10^{1}$	$1.0x10^{1}$
J_{shaft}	[kg m^2]	$2x10^{-3}$	$1.9x10^{-3}+(2x10^{-3}*\theta\_pitch)$
C_{shaft}	[Nms/rad]	0.3	0.1
K_{shaft}	[N/rad]	$1x10^{6}$	$2.3x10^{6}$

-

With the entire physical system modeled, the reduced feedback loop for the system can be developed. A potentiometer measures the position of the output shaft. A prefilter and compensator are implemented to stabilize the system. The design of these components are covered in the following section.

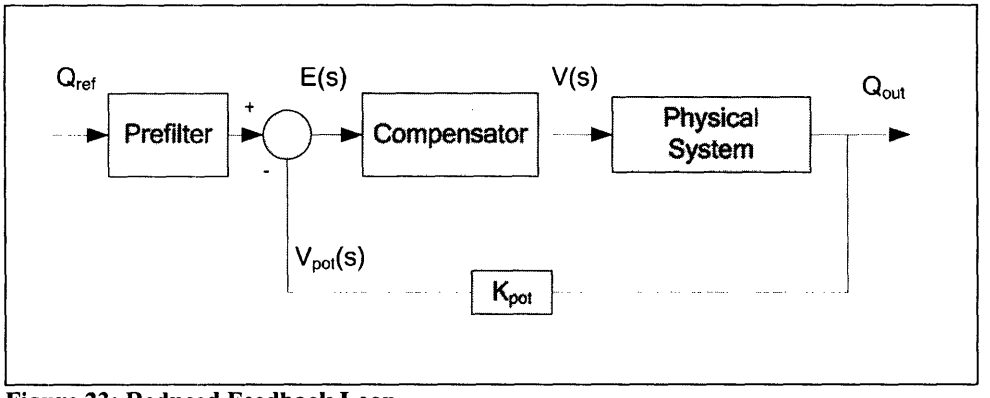


Figure 23: Reduced Feedback Loop

3.1.1 Compensator Design

The compensator is designed with the understanding that it will be implemented in software. A set of discrete time approximations for PID control are implemented. The proportional contribution is a constant, and so it is the same as in the frequency domain. In software, this is simply the present error signal multiplied by a constant.

$$V_p(s) \approx v_p(t) = K_p \cdot e(t) \tag{1.23}$$

The differential contribution is the time derivative of the error signal, scaled by a constant. In software this is approximated as the difference between the present value and the value immediately before, scaled by the length of time between the samples, and scaled again by the constant.

$$V_d(s) \approx v_d(t) = K_d \cdot \frac{[e(t_n) - e(t_{n-1})]}{\Delta t} \tag{1.24}$$

The integral contribution sums is the sum of all differential elements up to the present time. In software, this is implemented as a sum of the error signals from the start of the command to the present time. More practically, the derivative error signal calculated for the differential contribution is added to a storage variable.

$$V_i(s) \approx v_i(t) = K_i \cdot \sum_{j=0}^{t} e(j) \cdot \Delta t \tag{1.25}$$

3.2 Kinematic Controller

The kinematic controller is responsible for translating user commands into joint angles and thruster speeds. This is accomplished with a series of trigonometric and minimization calculations.

3.2.1 Analysis

In order to analyze the kinematics of the system, each joint is given its own coordinate system. Each is defined by a linear translation and rotation from the previous coordinate system. The four systems are, in order, the center of mass, base, yaw, and pitch. The center of mass system is mapped to the global coordinate system X-surge, Y-sway, Z-heave.

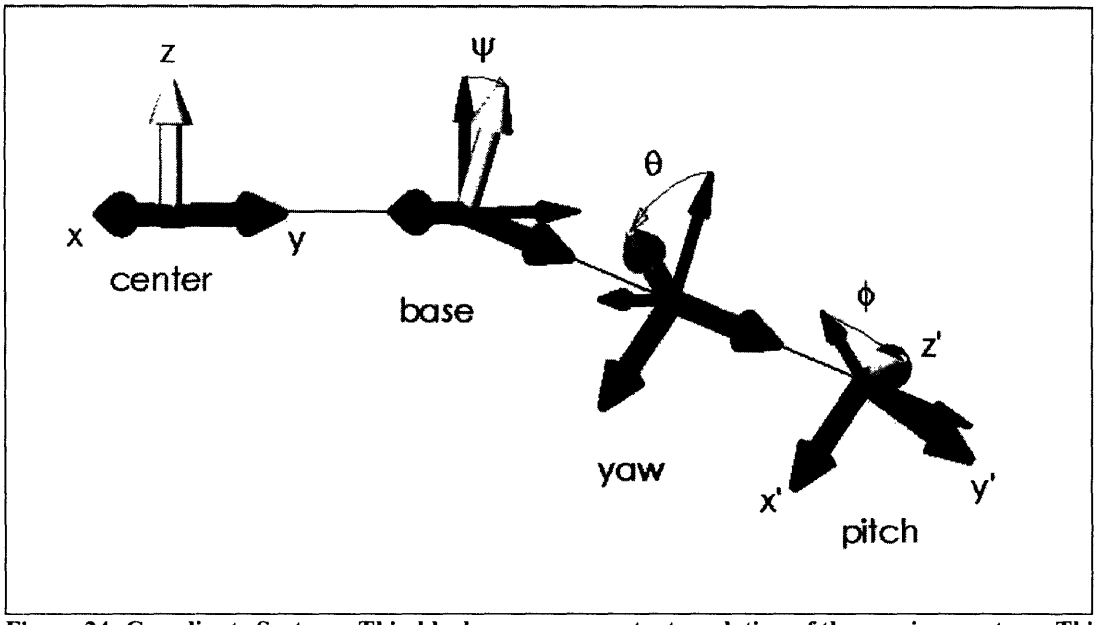


Figure 24: Coordinate Systems. Thin black axes represent a translation of the previous system. Thick axes represent current rotated system.

A series of homogeneous transform matrices (HTMs) are used to calculate the forward kinematics of the robot. HTMs allow relatively easy conversion between coordinate systems with both angular and linear offsets. Beginning with the dihedral (anhedral) angle of the thruster mounts at the base, the base coordinate system from the center (origin) coordinate system is

$$\begin{pmatrix} 1 & 0 & 0 & X_{center\_to\_base} \\ 0 & \cos(\psi) & -\sin(\psi) & Y_{center\_to\_base} \\ 0 & \sin(\psi) & \cos(\psi) & Z_{center\_to\_base} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.26}$$

Next, from the base to the yoke assembly coordinate system

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & X_{base\_to\_yoke} \\ \sin(\theta) & \cos(\theta) & 0 & Y_{base\_to\_yoke} \\ 0 & 0 & 1 & Z_{base\_to\_yoke} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.27}$$

From the yoke to the thruster coordinate system

$$\begin{pmatrix} 1 & 0 & 0 & X_{yoke\_to\_thruster} \\ 0 & \cos(\phi) & -\sin(\phi) & Y_{yoke\_to\_thruster} \\ 0 & \sin(\phi) & \cos(\phi) & Z_{yoke\_to\_thruster} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.28}$$

The three HTMs are combined symbolically into one large 4x4 homogeneous transform matrix which is the total change between the base and thruster coordinate systems.

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta)\cdot\cos(\psi) & \sin(\theta)\cdot\sin(\psi) & \\ \cos(\phi)\cdot\sin(\theta) & \cos(\phi)\cdot\cos(\theta)\cdot\cos(\psi) - \sin(\phi)\cdot\sin(\psi) & -\cos(\phi)\cdot\cos(\theta)\cdot\sin(\psi) - \sin(\phi)\cdot\cos(\psi) & \\ \sin(\phi)\cdot\sin(\theta) & \sin(\phi)\cdot\cos(\theta)\cdot\cos(\psi) + \cos(\phi)\cdot\sin(\psi) & -\sin(\phi)\cdot\cos(\theta)\cdot\sin(\psi) + \cos(\phi)\cdot\cos(\psi) & \\ 0 & 0 & 0 & \cdots \\ \cdots & & & 1 \end{pmatrix} \tag{1.29}$$

$$\begin{pmatrix} \cos(\theta)\cdot\mathrm{Xdih} - \sin(\theta)\cdot\mathrm{Ydih} + 2\cdot\mathrm{Xpan} \\ \cos(\phi)\cdot\sin(\theta)\cdot\mathrm{Xdih} + \cos(\phi)\cdot\cos(\theta)\cdot\mathrm{Ydih} - \sin(\phi)\cdot\mathrm{Zdih} + \cos(\phi)\cdot\mathrm{Ypan} - \sin(\phi)\cdot\mathrm{Zpan} + \mathrm{Ytilt} \\ \sin(\phi)\cdot\sin(\theta)\cdot\mathrm{Xdih} + \sin(\phi)\cdot\cos(\theta)\cdot\mathrm{Ydih} + \cos(\phi)\cdot\mathrm{Zdih} + \sin(\phi)\cdot\mathrm{Ypan} + \cos(\phi)\cdot\mathrm{Zpan} + \mathrm{Ztilt} \\ 1 \end{pmatrix}$$

The thrust value for each thruster is calculated by dividing the initial XYZ input vector by its maximum possible magnitude. These HTMs allow conversion between XYZ coordinates and θ,ϕ coordinates. The thrust calculation gives the output coordinates a full set of θ,ϕ,M coordinates.

In the software implementation, a direct inverse kinematic approach is taken. This allows the trig computation to be kept to a minimum, using only the arctangent during the function call. The first transform is used, although since the anhedral angle does not change throughout the mission, the sine and cosine components are calculated at initialization.

$$\begin{aligned} AnhedralCOS &= \cos(\psi) \\ AnhedralSIN &= \sin(\psi) \end{aligned} \tag{1.30}$$

Then at each software call, only the multiplication is required. The matrix structure is removed for calculation in ANSI C.

$$\begin{aligned} y_1' &= y_1 \cdot AnhedralCOS + z_1 \cdot AnhedralSIN \\ z_1' &= z_1 \cdot AnhedralCOS + y_1 \cdot AnhedralSIN \end{aligned} \tag{1.31}$$

The inverse kinematics consist of two arctangent operations. The first determines the angle θ.

$$\theta_1 = \arctan\left(\frac{z_1}{x_1}\right) \tag{1.32}$$

The second determines the angle ϕ. This angle requires the magnitude of the vector in the XZ plane.

$$\phi_1 = \arctan\left(\frac{\sqrt{x_1^2 + z_1^2}}{y_1}\right) \tag{1.33}$$

The thruster magnitude is initially treated as the average magnitude

$$M_1 = \frac{x_1 + y_1 + z_1}{3} \tag{1.34}$$

If $x_1 \ldots z_1$ are given a maximum of 1, this method only allows maximum thrust with a full component in each direction. The maximum output for thrust purely in the x, y, or z direction is 1/3. A more advanced possibility allows the thrusters to command the maximum output in any direction. This would be implemented by comparing the average magnitude to the maximum possible magnitude for a particular ration of x y and z. In software, a lookup table would be used.

In order to make the ROV usable by unskilled operators, a higher level of control must be implemented. This level takes user commands (X,Y,Z,Θ) for the total robot and generates XYZ, then θ,ϕ,M commands for each thruster. Because the system is over-actuated, there is not a one-to-one mapping of output translation to input commands. Several sets of thruster XYZ coordinates can produce the same ROV X,Y,Z, Θ output. In order to be able to produce a program that is capable of selecting an appropriate set of coordinates, a set of values are minimized. First, the system seeks to use the minimum possible thrust. Second, the system seeks to move as little as possible to the next position.

3.2.2 Low Level Algorithm

The simplest algorithm satisfies the thrust minimization, but does not satisfy the motion minimization. To satisfy force balance, the sum of each thruster contribution must add to the total desired output. To satisfy moment balance, the contributions must be equal for translation. To accomplish this, each thruster takes one-half of the input displacement command in X,Y, and Z. The Θ input is also split, but with one of the halves reversed in sign to create rotation.

$$\begin{aligned} x_1 &= \frac{X}{2} + \frac{\Theta}{2} & x_2 &= \frac{X}{2} - \frac{\Theta}{2} \\ y_1 &= \frac{Y}{2} & y_2 &= \frac{Y}{2} \\ z_1 &= \frac{Z}{2} & z_2 &= \frac{Z}{2} \end{aligned} \tag{1.35}$$

Or in matrix form

$$\begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ \Theta \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$$
$$\begin{pmatrix} \frac{1}{2} & 0 & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ \Theta \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \tag{1.36}$$

3.2.3 Advanced Algorithm

A second, more advanced algorithm takes advantage of running the thrusters in reverse to provide more options for thruster positioning. The system identifies all possible configurations that satisfy the thrust minimization criterion. The closest one to the current configuration is then selected. In this augmented version of the low level system, the second configuration takes the negative of XYZ for both thrusters. An identifier bit is used to mark the coordinates as reverse thrust. The distance from the current position to both possible configuration is calculated as a straight-line displacement.

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}_{Forward} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ \Theta \end{pmatrix} \tag{1.37}$$

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}_{Reverse} = \begin{pmatrix} -\frac{1}{2} & 0 & 0 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ \Theta \end{pmatrix} \tag{1.38}$$

$$\begin{aligned} \left|\Delta_{1\_forward}\right| &= \sqrt{\left(x_{forward} - x_{current}\right)^2 + \left(y_{forward} - y_{current}\right)^2 + \left(z_{forward} - z_{current}\right)^2} \\ \left|\Delta_{1\_reverse}\right| &= \sqrt{\left(x_{reverse} - x_{current}\right)^2 + \left(y_{reverse} - y_{current}\right)^2 + \left(z_{reverse} - z_{current}\right)^2} \end{aligned} \tag{1.39}$$

3.3 User Interface

The user interface is kept to the minimum required for operation. The interface sends basic commands, such as forward and back, and receives a constant stream of system parameters. The topside control is a Visual Basic application using the MSCOMM serial protocol.

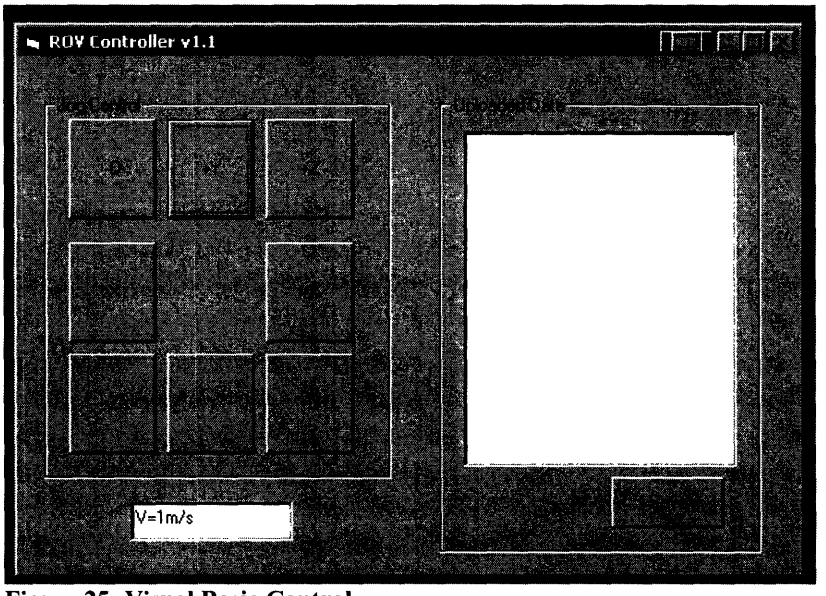


Figure 25: Visual Basic Control

Initial testing is done with a joystick plugged directly into the controller hardware.

3.4 Controller Hardware

The onboard hardware consists of a custom embedded microcontroller system. The controller was previously designed by the author as a leg controlling slave board for an amphibious walking robot. The board has onboard capability for four motors drawing up to 5A each, 32 analog inputs, and two channels of asynchronous serial (RS-232 compatible). A commercial motor driver is used to operate each of the main thrusters.

Table 5: Control Hardware Requirements and Implementation

Area	*System Requirement*	*Hardware Specification/implementation*
Small Motor Driver	4 - 12V, 2A motors (Pitch and yaw for two assemblies)	4, 5A each, 24V max
Large motor Driver	2 12V 15A (two main thrusters)	Commercial RC Motor controller 36V 24A – driven by 2 digital I/O pins on board
Sensor input	4 analog in (position potentiometers) 10 analog in (accelerometers/gyros) 4 sensors (depth, ??????)	32 multiplexed analog inputs to 10 bit A/D converter
Communications	RS-232 Uplink to PC	Hardware UART with RS-232 level shifter, up to 115200 bits per second.
Computation	500 Hz? Processing and servoing	10 MIPS (million instructions per second) for average cycle time of ?? s = ?? Hz

The controller uses a Microchip PIC18F458 programmed in mixed C and assembly as a general purpose microcontroller. Small motors are driven by the next generation Motorola M33887 H-bridge IC. The M33887 has ultra-low on-resistance and fast switching times to minimize the energy dissipated as heat. This allows control of motors at 5A in a 1" x 0.5" x 0.1" (LxWxH) package with only ground plane heat sinking. This allows a tighter board design than traditional H-bridges would allow.

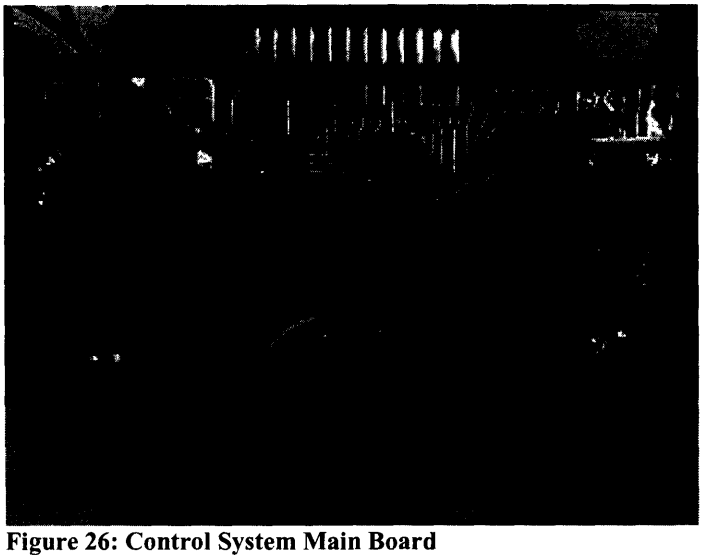


Figure 26: Control System Main Board

For testing, the control system was kept out of the water. This means that the tether carries all 12 motor wires plus a minimum of 6 sensor wires. In a practical implementation, the tether would only carry power and a bi-directional data line, in order to minimize the cost and stiffness of the tether; if the tether is too stiff, the motion of the ROV may be severely impaired. Creating a waterproof control housing is a well defined and solved problem; it was not repeated in this research in an effort to reduce the system complexity and potential modes of failure.

4.0 Experimental Validation

The experimental portion takes two major forms. First, the ability of the mechanism to move into position on command is measured. This includes positioning accuracy including backlash and compliance, and speed/bandwidth. Second, two of the devices are mounted on a hull, and the ability to control motion in all six degrees of freedom is measured. Although only four degrees of freedom have user inputs, the remaining two, pitch and roll may be actively constrained by the control system, and so are also relevant outputs. The positioning/velocity capability of the system is again measured in terms of accuracy and bandwidth.

4.1.0 Single Mechanism: Setup and Procedure

The purpose of the single mechanism test is to measure the performance of the servomechanism, separate from any high level control system.

All joints except the active joint are commanded to hold a center position. The active joint is sent a square wave command. The initial frequency is slow enough to allow a full step response on both the rise and fall of the input signal. During the test, the frequency is increased until the system is unable to respond in any sort of coherent manner. Input and output waveforms are recorded throughout the procedure. For each input frequency, the average final position is calculated. The average final position is plotted against the input frequency to give the closed loop system response. The closed loop transfer function is also approximated from the graph.

The initial setup and test is done in the air, to allow for easy calibration and repair. The robot is mounted upright on a test stand.

Figure 27: Robot mounted on test stand

The second setup is designed to measure the system response while immersed in water. The robot is mounted on its side, with one thruster assembly submerged. By arranging it in this

manner, only one electronic component is submerged: the pitch potentiometer. Although all components have been sealed or otherwise waterproofed, this setup minimizes the potential damage to the system. This also is a better model for a full implementation of the system, where most of the mechanism would be within a plastic shell.

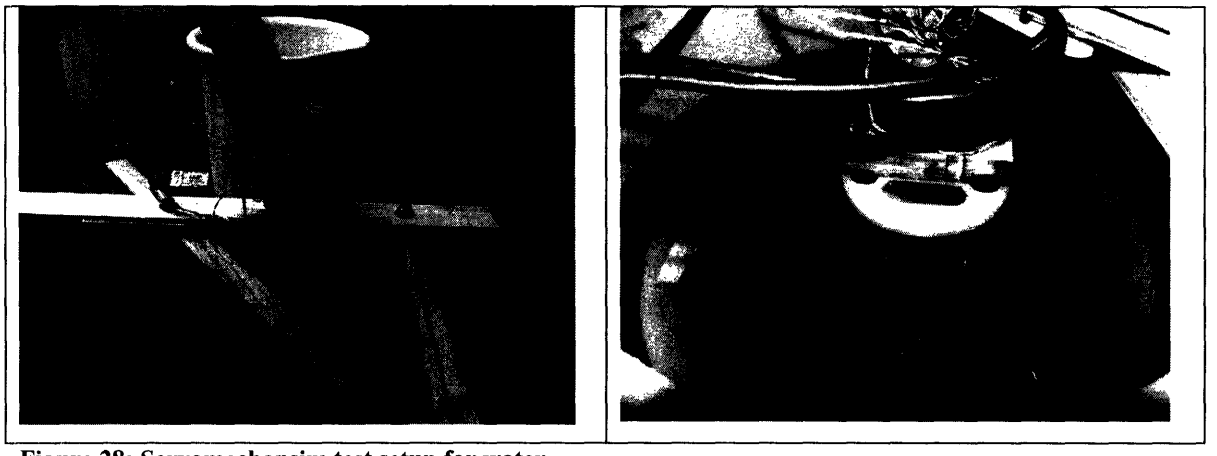


Figure 28: Servomechansim test setup for water

4.1.1 Single Mechanism: Results

The system response is measured for input frequencies ranging from 0.5Hz to 9 Hz, for a angular displacement of 130° for pitch and the and 170° for the yaw, representing a typical motion. Low frequency responses saw a clean matching of input command (Fig.29). High frequency responses showed a significant drop in magnitude, as the actuator was unable to keep up with the commands (Fig.30). Relatively high overshoot on the yaw is a result of the predominantly proportional tuning of the PID loop.

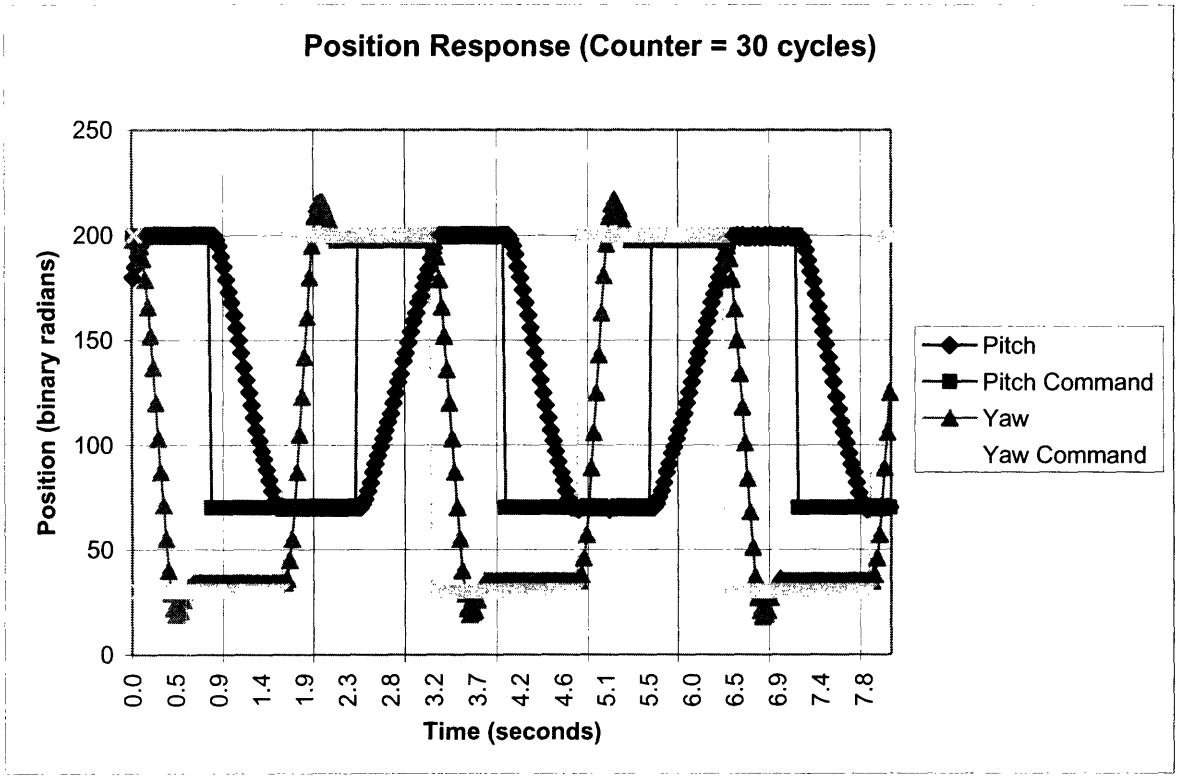


Figure 29: Low frequency input and system response. 30 cycles ~ 0.3 Hz

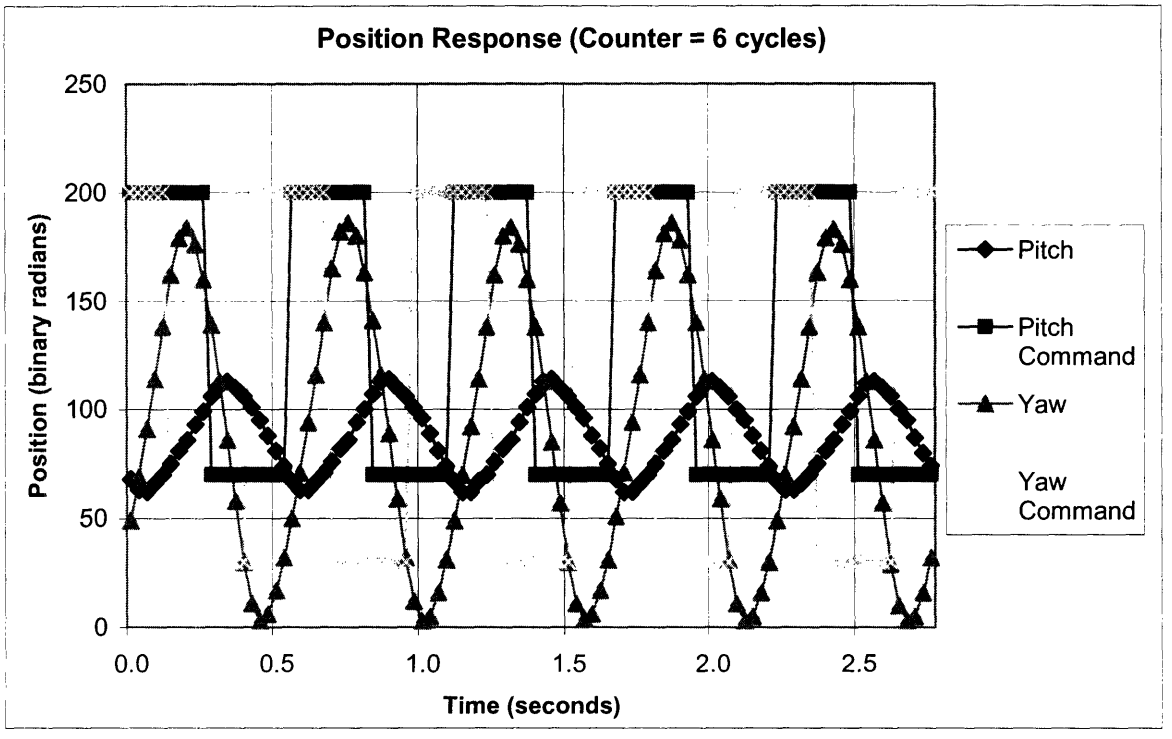


Figure 30: High frequency input and associated magnitude decrease. 6 cycles ~ 2Hz

This data was recorded for 13 different frequencies. The relative magnitude is compared to the initial command magnitude in a frequency response plot (Figs.31,32)

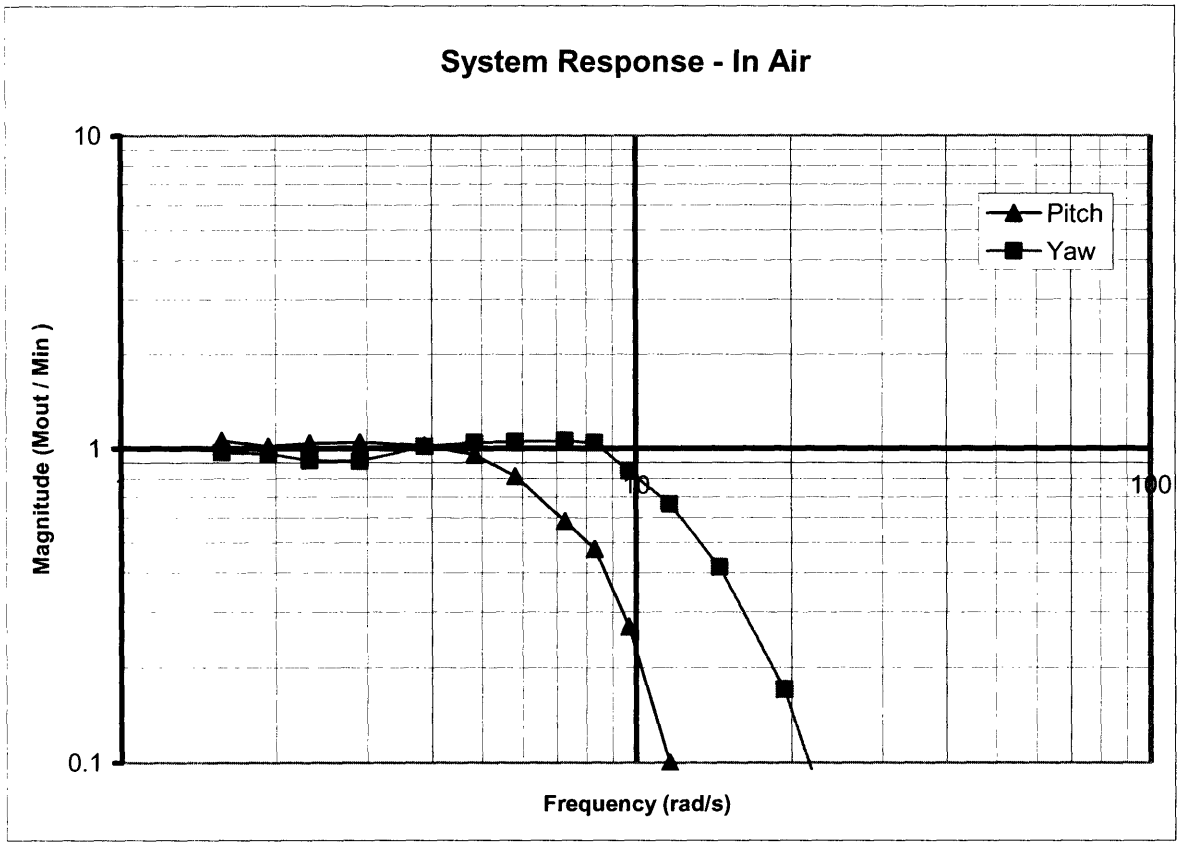


Figure 31: System Response in Air

The frequency response shows a bandwidth of approximately 6 rad/s or ~1 Hz for the pitch actuator, and approximately 10 rad/s or ~2 Hz for the yaw. After these frequencies the performance degrades along a two decade per decade line. The system closed loop transfer function can be modeled in its simplest form as

$$T_{pitch}(s) = \frac{1}{(6+s)^2} \tag{1.40}$$

$$T_{yaw}(s) = \frac{1}{(10+s)^2} \tag{1.41}$$

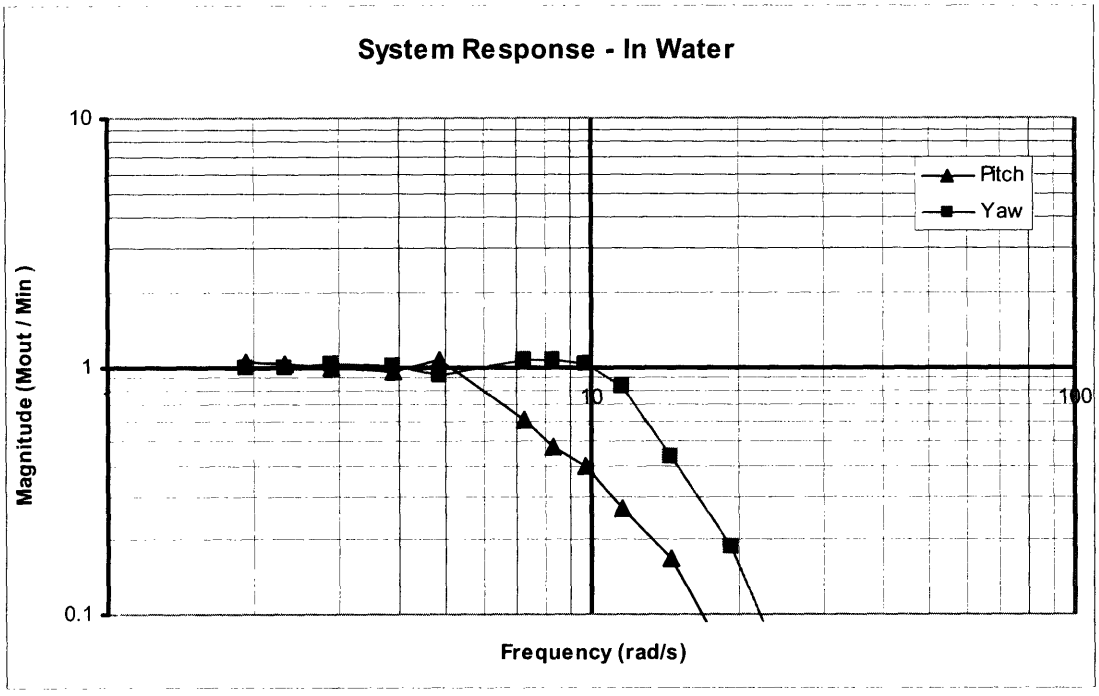


Figure 32: System Response in Water

As before the transfer functions can be approximated.

$$T_{pitch}(s) = \frac{1}{(6+s)^2} \tag{1.42}$$

$$T_{yaw}(s) = \frac{1}{(10+s)^2} \tag{1.43}$$

The waster test was conducted at a higher input voltage, resulting in approximately equal responses for air and water. Both air and water tests were conducted at a voltage just below the voltage at which the system would go unstable. Although the pitch axis has less inertia than the yaw axis, the belt drive mechanism of the pitch actuator has significantly more damping than the gear train on the yaw actuator. This explains the faster response of the yaw actuator.

4.2.0 ROV Mount: Setup and Procedure

The purpose of the ROV mount test is to quantify the performance of an ROV using the novel thruster technology. This performance is the primary motivation for the research. As with the single mechanism test, the accuracy and speed of output position are measured.

The challenge with this procedure is to measure and decouple motion in all six directions. As an example, for surge or forward motion, both the forward speed and the parasitic motion in the

other directions are relevant. The preferred setup is to use a six-axis measurement system, either an inertial measurement unit (3 accelerometers and 3 gyroscopes) or a camera setup. This allows all six directions of interest to be measured simultaneously for any input.

An alternative setup uses a battery of single axis measurements to record the most interesting aspects of the response, or all aspects over a much longer series. Measurement technologies include using string pots, depth sensors, and single axis accelerometers or gyroscopes.

4.2.2 ROV Mount: Results

Failures in the mechanism and control system prevented successful free-swimming tests. Additional tests will be carried out in future work with replacement hardware.

5.0 Analysis / Discussion

Table 6: ROV Performance v. Initial Specs

Parameter	*Initial Specification*	*Prototype Spec*	*Commercial Benchmark*
Mass	5kg	3kg	3kg
Size	0.35m sided cube	0.6x0.2x0.3m	0.35x0.25x0.21m
Max Speed	5m/s	--	2.5m/s
Max Acceleration	1 to -1m/s in 1 second	--	--

Although a full system, free-swimming test was not performed, it is possible to judge the potential performance of the system of the servomechanism test. The mass specification was met. The size specification is met in volume ($V=0.035m^3$, $V_{spec}=0.043m^3$), although the longest dimension is larger than both the commercial and initial specification.

The max speed has no quantifiable data on it, although the power output analysis suggests that the current system is capable. In **2.0.1**, the power lost through drag was estimated as ~6W, rounded up to 10W. The 400W combined mechanical output of the motors can suffer a 40X loss in transmission while still meeting the specification. The estimated loss in transmission was calculated as 4X, although this was for the optimum configuration. Certain thruster arrangements will have a significant obstruction to the water flow, reducing efficiency. The system is still overpowered for the drag spec, although this was identified as not a limiting criterion.

The largest contributor to power loss, and last specification is acceleration. The initial specification called for acceleration from 1m/s to -1m/s in 1 second. By reversing the thrusters, this can be achieved without the servomechanism. Although the motor power output is marginally capable of this spec, the additional 2kg remaining on the mass requirement can be used for larger motors. However, most propellers have a drastically reduced performance in reverse. A better method may be to use a highly efficient propeller in one direction, while rapidly positioning the thruster. In order to meet this requirement, the thruster must, as a minimum, be able to position within the one second. This would leave no time for thruster to work. Ideally, the positioning would take as little time as possible, in order to allow the maximum time for thruster output. With the current servomechanism, the fastest response before significant degradation is approximately 2Hz. Although the pitch actuator is only 1 Hz capable, it operates over half of the range.

With a 2Hz output, half a second is available for thruster output, under worst-case conditions. Because the same energy is required to change the velocities, the power to do it in half the time will be twice as much. This will require larger motors or a relaxation in the initial specification. The current system would be capable of decelerating from 1m/s to -1m/s in 1.5 seconds.

If motors with twice the power output were used, the additional mass would slow the system response down, and so would require additional time for positioning. More powerful thrusters of an equal mass and size would prevent this problem. Because relatively coarse, low efficiency motors were used in the current design, this is a non-trivial solution. Identified alternatives

include high power density model airplane motors from Astroflight, Inc. Assuming for now that this is not possible, An additional 1kg of available system mass will be distributed to the thrusters, 0.5 kg to each. This doubles the current 0.5 kg each. The addition of this mass will increase the pitch moment if inertia by 1.9x, and the yaw moment by 1.5x (accounting for additional mass of structure and drive train). The additional inertia will degrade the positioning performance, although not severely. In examining the experimental results in section 4, it is apparent that the limiting factor in positioning is the steady-state output of the motor. The acceleration and stabilizing periods account for only one fourth of the total motion. Furthermore, a proper retuning of the PID loop should be able to reduce this significantly. Even without a faster stabilization, however, the additional inertia will only increase the acceleration portion of the motion. With a conservative twofold increase in acceleration times, the new total positioning times will be

$$\begin{aligned} t_{new} &= 0.5 \cdot (0.75 + I_{factor} 0.25) \\ t_{new\_\max} &= 0.5 \cdot (1.25) = 0.625s \end{aligned} \tag{1.44}$$

The thrusters will thus have to be only ~2.5X the current output. In summary, the existing servomechanism is capable of meeting the performance specifications with moderately more powerful thrusters.

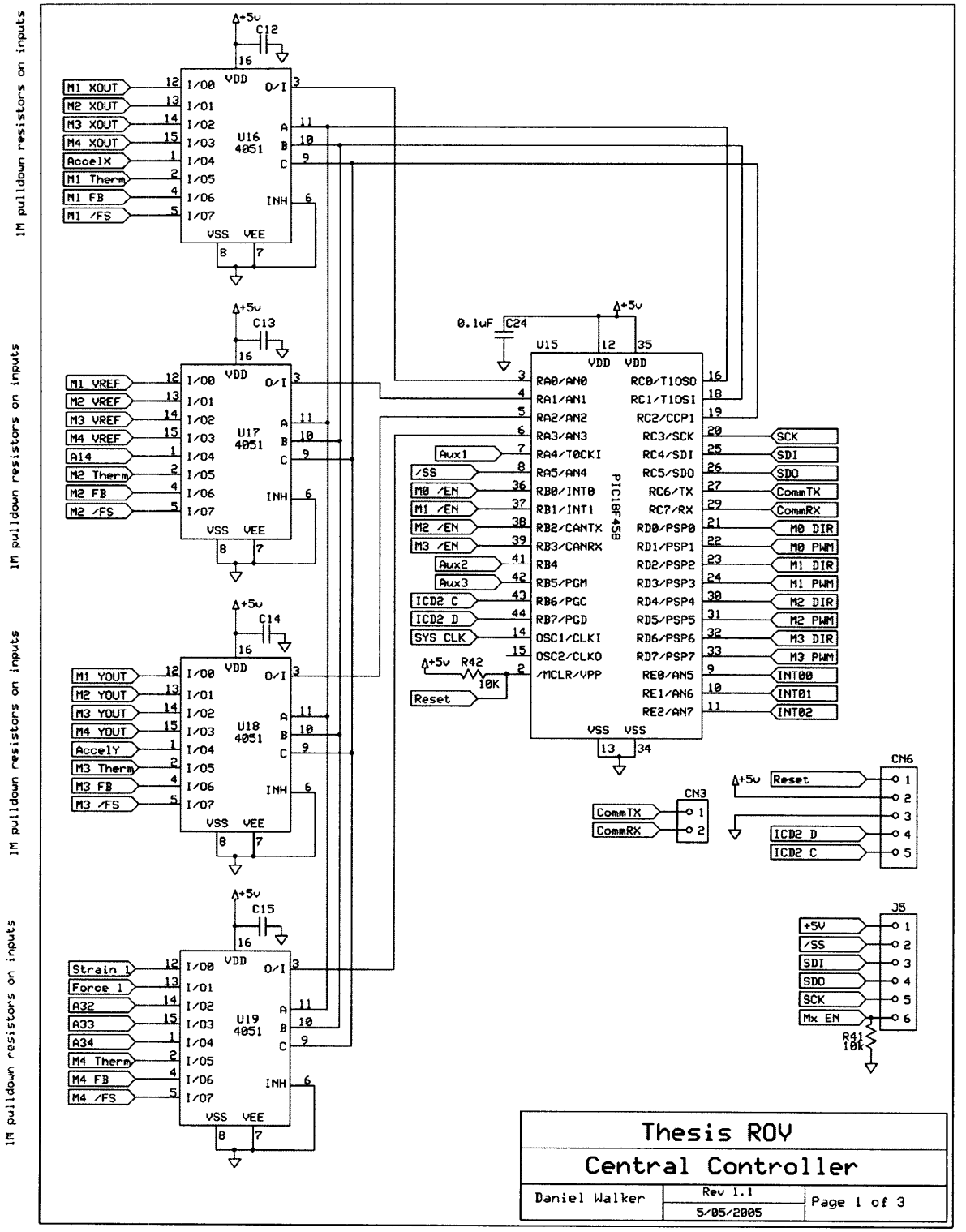
6.0 Conclusion

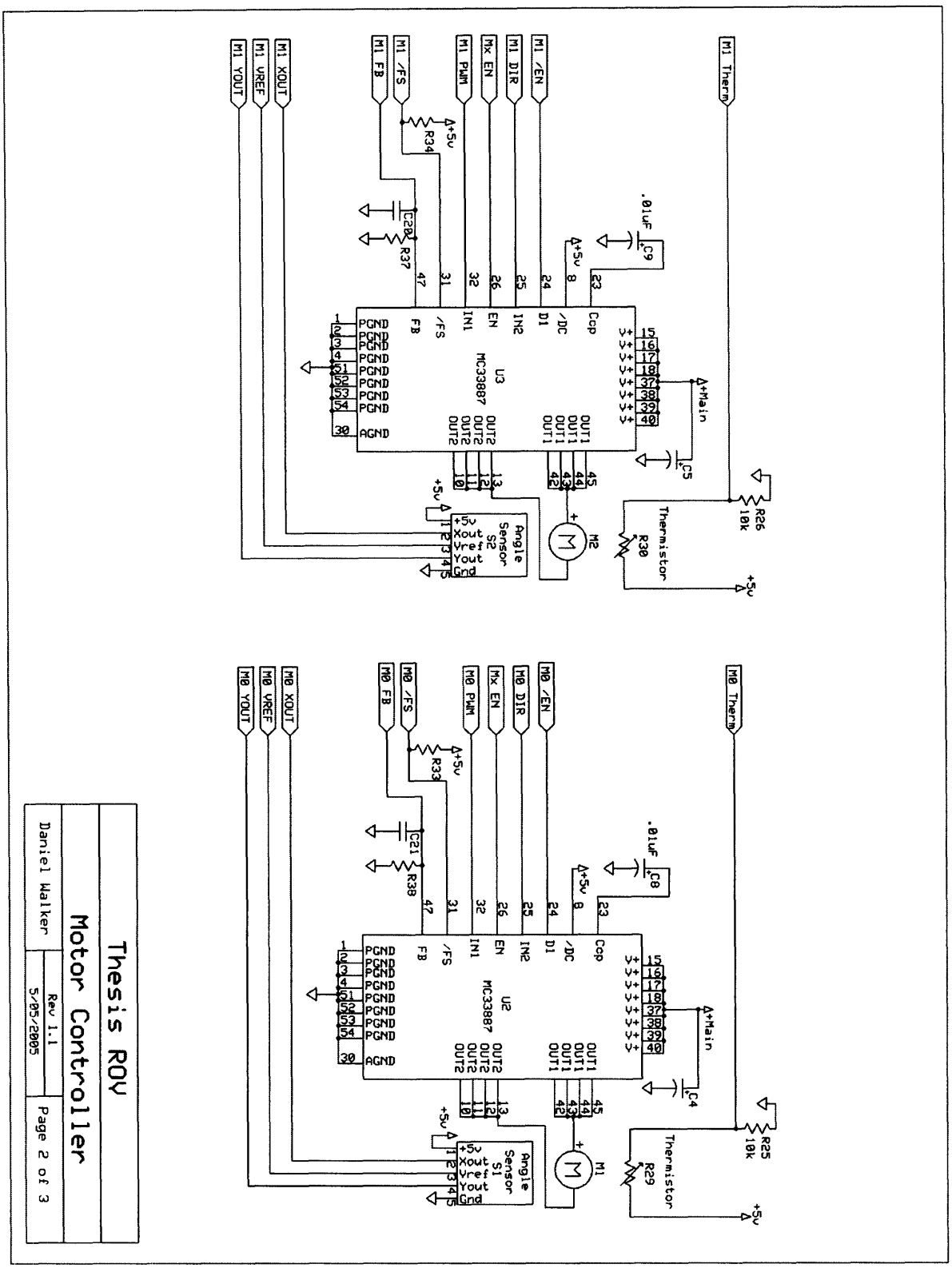
In this research, a highly maneuverable ROV is designed and manufactured. The intent is to create a platform that is useful as a small inspection-class robot. In addition, the research aims to create a design than can be scaled up to larger work-class ROVs. A set of industry-leading specifications was chosen as a design target.

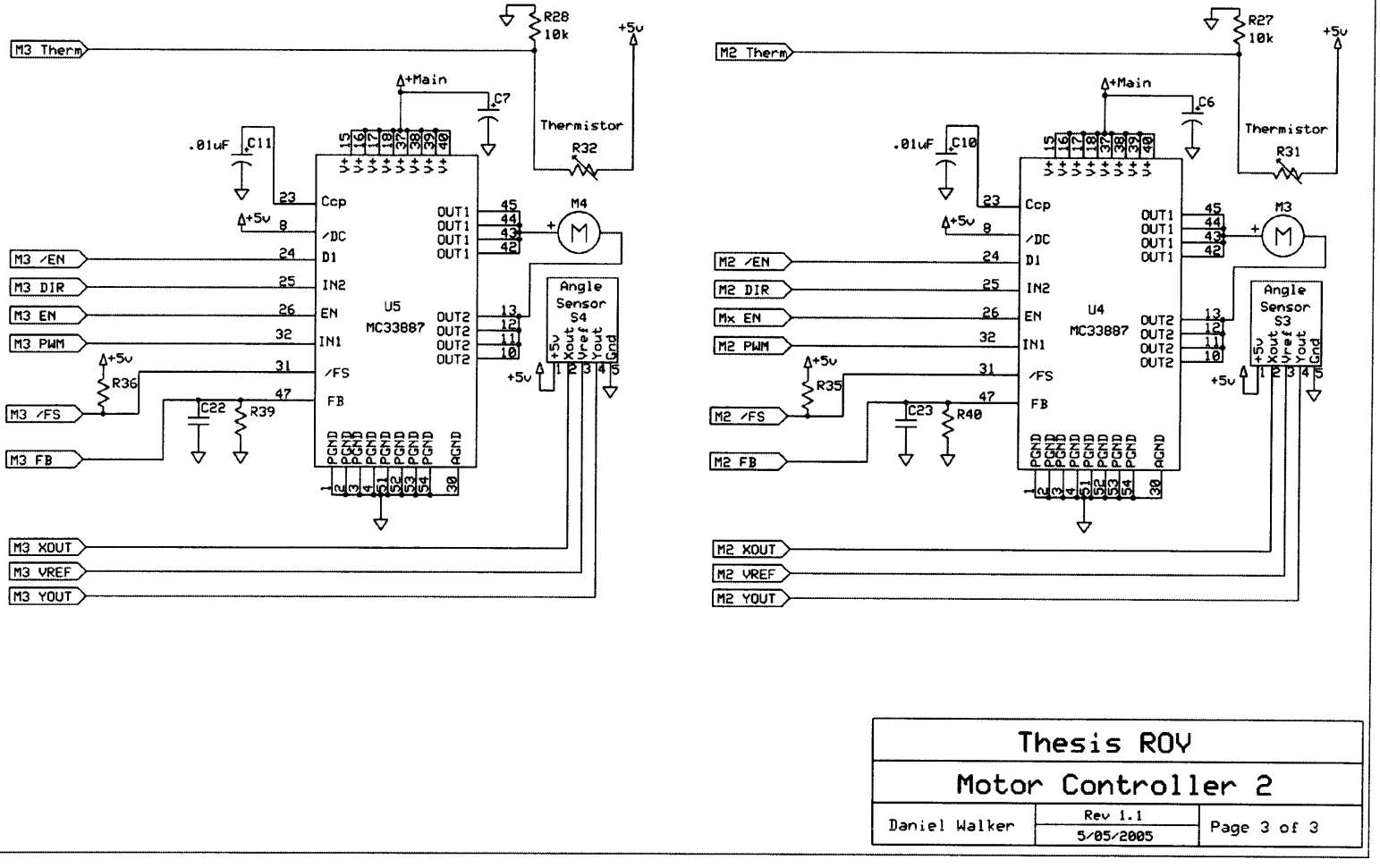
In the first design section, system parameters were examined to determine the relevance of each on the system performance. Thruster power was identified as the primary performance driver. Momentum effects, or acceleration, were identified as the primary draw of this thruster power. In the second section, a variety of possible propulsion arrangements are considered, both traditional and novel. The four designs of most interest include arrangements with: four fixed thrusters; two thrusters with two degrees of freedom each; a surface covered with small jets; and a combination of surface jets and thrusters. The two thruster design was selected because it could provide full power in all direction, and relied on existing technology. In the next section, the detailed mechanical design of the robot and any associated analysis was carried out. In the second design chapter, the control system was designed. A low-level PID controller was implemented in software. A high-level kinematic controller was also created, in order to generate joint output angles from high-level input commands, such as move forward, turn, etc. After this, the next section describes the experimental results from testing. Lastly, the results are analyzed in terms of the initial specifications. Although a full-system test was not performed, the servomechanisms are capable of performing well enough to meet the specification.

Although the closed-loop servo tests proved to be successful, no conclusive data was generated in testing the entire system. A combination of mechanical and control failure contributed to this. The obvious and immediate future work is in constructing a second-generation mechanical assembly, suitable for long-duration testing and operational conditions. In addition the control system should be revised to minimize the system resources used on algebraic and trigonometric operations. Trig look-up tables and clever manipulation of variable types will accomplish this. Additional future research includes finding the optimum anhedral angle, and modifying the control system to take advantage of this. Also, a full-body faring should be implemented to minimize drag during free swim tests. Overall, the design appears very promising, and warrants this future work.

Appendix A: Control System Schematics




M1 Therm
M1 YOUT
M1 VREF
M1 XOUT
M1 /FS
M1 FB
M1 PWM
Mx EN
M1 DIR
M1 /EN
.01uF
C9
+5v
+Main
C5
R26
10k
Thermistor
R30
M2
Angle Sensor S2
Xout
Yout
Gnd
U3
MC33887
Cap
/DC
D1
IN2
IN1
EN
/FS
FB
OUT1
OUT2
AGND
R34
R37
C28
M0 Therm
M0 YOUT
M0 VREF
M0 XOUT
M0 /FS
M0 FB
M0 PWM
Mx EN
M0 DIR
M0 /EN
C8
C4
R25
10k
Thermistor
R29
M1
Angle Sensor S1
U2
R33
R38
C21
Thesis ROV
Motor Controller
Daniel Walker
Rev 1.1
5/05/2005
Page 2 of 3



Thesis ROV		
Motor Controller 2		
Daniel Walker	Rev 1.1 5/05/2005	Page 3 of 3

Appendix B: Control System Code

Thesis.C – Main Program Loop

```
// Thesis Robot Controller
//
//
// Daniel Walker 2003 - 2005


#ZERO_RAM

#include <18F458a.h>
#include <18F458defs.h>

#include <slavedefs.h>
#include <interrupts.h>

long int counter,counter2;
#define  pathlength  4
unsigned int      ComArray[4][pathlength] =      {200,70,70,200          //rpitch
                                                  200,70,70,200,          //lpitch
                                                  30,30,200,200,          //ryaw
                                                  30,30,200,200};         //lyaw

void main(void)
{
        InitializeSlave();
        QuickCalcs();

        while (True)
                {
                for (counter2=0;counter2<pathlength;counter2++)
                {
//              RpitchCom = ComArray[0][counter2];
//              LpitchCom = ComArray[1][counter2];
//              RyawCom = ComArray[2][counter2];
//              LyawCom = ComArray[3][counter2];
                counter = 15;

                ReadJoysticks();
                CalcAngles(Xcom,Ycom,Zcom,Qcom,0.0);

//              ThrusterDuty1 = 133;
//              ThrusterDuty0 = 75;

                        while(counter-->1)
                        {

                        ReadAllAnalog();

                        Lpitch = Lpitch+Lpitchzero;
                        Lyaw = Lyaw+Lyawzero;
                        Ryaw = Ryaw+Ryawzero;
                        Rpitch = Rpitch+Rpitchzero;
```

```
                    fprintf(COM_A,"%5.2f,%5.2f,%5.2f,%5.2f, \n\r",Xcom,Ycom,Zcom,Qcom);

                    DoFeedback(0, CalcFeedback(0,Rpitch,RpitchCom) );
                    DoFeedback(1, CalcFeedback(1,Lyaw,LyawCom) );
                    DoFeedback(2, CalcFeedback(2,Ryaw,RyawCom) );
                    DoFeedback(3, CalcFeedback(3,Lpitch,LpitchCom) );

                    fprintf(COM_A,"%3u,%3u,  %3u,%3u,  %3u,%3u,  %3u,%3u, \n\r",Rpitch,RpitchCom,
        //
        fprintf(COM_A,"%3u,%3u,%3u,%3u\n\r",PWMduty[0],PWMduty[1],PWMduty[2],PWMduty[3]);

                    }

            }

            }

}
```

Slavedefs.h – initializes and defines main functions

```
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=40000000)

#use rs232(baud=115200, xmit=PIN_C6, rcv=PIN_C7, STREAM=COM_A)
#use rs232(baud=38400, xmit=PIN_C4, rcv=PIN_C3,STREAM=COM_B)


#include <slavemath.h>

//IO pins

#define  ActiveLED              PORTE,2

#define    EN0                  PORTB,0
#define    EN1                  PORTB,1
#define    EN2                  PORTB,2
#define    EN3                  PORTB,3

#define    PWM0                        PORTD,1
#define    PWM1                        PORTD,3
#define    PWM2                        PORTD,5
#define    PWM3                        PORTD,7

#define    DIR0                 PORTD,0
#define    DIR1                 PORTD,2
#define    DIR2                 PORTD,4
#define    DIR3                 PORTD,6

#define  Thruster0                     PORTC,3
#define  Thruster1                     PORTC,5

//Analog Ports
unsigned int    joint,ADpin,ADmuxChan;
unsigned int    ADres[4][8];

#define         FaultStatus0            ADres[0][5]
#define         FaultStatus1            ADres[1][5]
#define         FaultStatus2            ADres[2][5]
#define         FaultStatus3            ADres[3][5]

#define         FeedbackTemp0           ADres[0][7]
#define         FeedbackTemp1           ADres[1][7]
#define         FeedbackTemp2           ADres[2][7]
#define         FeedbackTemp3           ADres[3][7]

#define         FeedbackCurrent0                ADres[0][7]
#define         FeedbackCurrent1                ADres[1][7]
#define         FeedbackCurrent2                ADres[2][7]
#define         FeedbackCurrent3                ADres[3][7]

#define         Rpitch                  ADres[3][1]
#define         Ryaw                    ADres[3][2]
#define         Lyaw                    ADres[3][4]
```

```
#define     Lpitch          ADres[3][6]


#define     RFR             ADres[1][1]
#define     Ltheta          ADres[1][2]
#define     LLR             ADres[1][4]
#define     Rtheta          ADres[1][6]

#define     Z_in                        ADres[1][1]
#define     Y_in                        ADres[1][2]
#define     Q_in                        ADres[1][4]
#define     X_in                        ADres[1][6]


unsigned int    Lpitchzero=235,Lyawzero=253,Ryawzero=0,Rpitchzero=10;
unsigned int    LpitchCom=127,LyawCom=127,RyawCom=127,RpitchCom=127;
long int        ThrusterCount,ThrusterDuty0=103,ThrusterDuty1=150,ThrusterPeriod=4000;
unsigned int    ThrusterFW = 129,ThrusterRV = 75, ThrusterMID = 105, ThrusterRange=28;

//PWM commands
unsigned int    PWMcount,PWMperiod=31,PWMduty[4];

//#defineTimerOffset            0x0D0       //2kHz PWM, 203 cycles with overhead 730
#define  TimerOffset            0x0A0       //1kHz PWM, 396 cycles with overhead 550 !thruster
PWM set for this value!
//#defineTimerOffset            0x041       //500 Hz PWM, 775 cycles with overhead 260
//#define      TimerOffset                0x000       //500 Hz PWM, 775 cycles with overhead
260

unsigned int    CommSTAT;
unsigned int    i;
unsigned int    temp;
unsigned int    count;
float           temp2;


//unsigned int          jointpos[4],jointcom[4]={127,127,127,127};
float   err[4],errlast[4];
float   Kp[4]={.5,0.1,0.1,0.15},Kd[4]={0.02,0.01,0.01,0.02},Ki[4] = {0.005,0.002,0.002,0.005};
        //rpitch, lyaw, ryaw, lpitch

float           Vout[4], Voutmax=12.0;
float           D_contribution,P_contribution,I_contribution,delta_t=0.016,Integrator_sum[4];
unsigned int    I_LastCom[4];
float           AnhedralCOS,AnhedralSIN,AnhedralAngle = 0.0;
float           distR1,distR2,distL1,distL2;
float           Xcom,Ycom,Zcom,Qcom;

unsigned int    MotorDISABLE = 0b11111111,MotorTempWarning = 20,MotorTempLimit = 25,
MotorTempWarningQC=0, MotorTempLimitQC;


////////////////////////////////////////////////////////////
//Functions
```

```
//-------------------------------------------------------------//
//      delay routine  - leave as first.  subsequent calls
//-------------------------------------------------------------//

void Delay(float cycles)
        {
        while(cycles>0)
                cycles = cycles-1;
        }



//-------------------------------------------------------------//
//      Initialization
//-------------------------------------------------------------//

void InitializeSlave(void)
        {
        set_tris_a(0b00101111);
        set_tris_b(0b00000000);                         //portb output
        set_tris_c(0b00000000);                         //portc 3, 5 out for thruster pwm
        set_tris_d(0b00000000);                         //portd output
        set_tris_e(0b00000000);

        setup_adc_ports( A_ANALOG);             //porta is all analog
        setup_adc( ADC_CLOCK_INTERNAL );

        fprintf(COM_A,"\n\r\n\r\n\r\n\r\n\r\n\rLink Active\n\r");
//      printf("\n\r\n\r\n\r\n\r\n\r\n\rLink Active\n\r");


        enable_interrupts(INT_TIMER0);          //enable timer interrupt
        enable_interrupts(GLOBAL);                      //set Global Enable

        #asm                                            //clear to enable motor outputs
                bcf     EN0
                bcf     EN1
                bcf     EN2
                bcf     EN3
        #endasm

        #asm
//              clrwdt
                bcf             T0CON,T0PS2
                bcf             T0CON,T0PS1
                bsf             T0CON,T0PS0             //write 0x001 to prescaler

                bsf             T0CON,PSA               //enable prescaler
                bcf             T0CON,5                         // select main clock source

                clrf    TMR0L                   //start timer
```

```
        #endasm

                #asm
                bsf             ActiveLED
                #endasm
                Delay(10000);
                #asm
                bcf             ActiveLED
                #endasm
                Delay(2000);
                #asm
                bsf             ActiveLED
                #endasm
                Delay(2000);
                #asm
                bcf             ActiveLED
                #endasm
                Delay(2000);
                #asm
                bsf             ActiveLED
                #endasm


        }


//---------------------------------------------------------------//
//      read all analog inputs
//---------------------------------------------------------------//

void ReadAllAnalog(void)
        {
        for(ADpin = 0; ADpin <=3; ADpin++)                      //index ADinputs
        for(ADmuxChan = 0; ADmuxChan<=7;ADmuxChan++)    //index   ADpin
                {
                #asm                                                    //select MUX channel on ADmux
                        movlw   0b11111000              //clear select bits
                        andwf   PORTC,f

                        movf    ADmuxChan,w
                        andlw   0b00000111              //kill top bits on ADpin
                        iorwf   PORTC,f                         //write bottom 3 bits

                #endasm

                set_adc_channel( ADpin );
                delay_us(10);
                temp = Read_ADC();
                ADres[ADpin][ADmuxChan] = temp;
                delay_us(100);
                }
        }

//---------------------------------------------------------------//
//send all AD values on UART
```

```
//-----------------------------------------------------------------//

void TXAllAnalog(void)
        {
//      fprintf(COM_A,"\n\r");
        for(ADpin = 0; ADpin <=3; ADpin++)                      //index ADpin
        for(ADmuxChan = 0; ADmuxChan<=7;ADmuxChan++)    //index   MuxChan
                fprintf(COM_A,"ADpin %3u MuxChan %3u =
%3u\n\r",ADpin,ADmuxChan,ADres[ADpin][ADmuxChan]);
                //printf("ADpin %3u MuxChan %3u =
%3u\n\r",ADpin,ADmuxChan,ADres[ADpin][ADmuxChan]);


        }




//-----------------------------------------------------------------//
//      Quick Calcs: setup limit values
//-----------------------------------------------------------------//
void QuickCalcs(void)
        {
//Temperature Thresholds
        //ADout = 25.85*e^{-.0405*[2550/(Temperature+10)]}
        float const     TempIndex[17] =
{119.27,132.17,144.97,157.42,169.29,180.41,190.63,199.88,208.13,215.38,221.70,227.14,231.78,235.78,239.03,24
1.8,244.12};
        float const     TempTail[16] =
{12.9,12.8,12.45,11.87,11.12,10.22,9.25,8.25,7.25,6.32,5.44,4.64,4,3.25,2.77,2.32};

        float QCindex=0,QCremainder=1/*,QCtemp*/;
        QCremainder = modf((((float)MotorTempWarning-20)/5),&QCindex);
        MotorTempWarningQC = TempIndex[QCindex]+QCremainder*TempTail[QCindex];

        QCremainder = modf((((float)MotorTempLimit-20)/5),&QCindex);
        MotorTempLimitQC = TempIndex[QCindex]+QCremainder*TempTail[QCindex];

        AnhedralSIN = SIN(AnhedralAngle);
        AnhedralCOS = COS(AnhedralAngle);
        }



//-----------------------------------------------------------------//
//calculate feedback
//err,temp: local
//-----------------------------------------------------------------//
float CalcFeedback(unsigned int joint, unsigned int jointpos, unsigned int jointcom)
        {
        float D_error;


        err[joint] = (float)jointcom-jointpos;
        D_error =       ( err[joint]-errlast[joint] )/delta_t;
```

```
//      fprintf(COM_A,"%3u, %5.2f \n\r",joint, D_error );


        Integrator_sum[joint] = D_error+Integrator_sum[joint];
        if (jointcom != I_LastCom[joint])
                {
                Integrator_sum[joint] = D_error;
//              fprintf(COM_A,"%3u,%3u,%3u \n\r",joint, jointcom , I_LastCom[joint] );

                }


        D_contribution = Kd[joint] * D_error;
        P_contribution = Kp[joint] * (err[joint]) ;
        I_contribution = Ki[joint] * Integrator_sum[joint];

        Vout[joint] =P_contribution + D_contribution + I_contribution;

        errlast[joint]=err[joint];
        I_LastCom[joint] = jointcom;
        //saturation criterion

        if (Vout[joint] > Voutmax)
                Vout[joint] = Voutmax;
        if (Vout[joint] < -Voutmax)
                Vout[joint] = -Voutmax;

        return Vout[joint];
        }


//-----------------------------------------------------------------//
// implement motor actuation
//
//-----------------------------------------------------------------//
void DoFeedback(unsigned int joint,float Vout)
        {
                if (Vout>=0.0) //positive
                        {
                        PWMduty[joint] = PWMperiod*Vout/Voutmax;

                        switch(joint)
                                {
                                case    0:
                                #asm
                                bcf             DIR0
                                #endasm
                                break;

                                case    1:
                                #asm
                                bcf             DIR1
                                #endasm
                                break;

                                case    2:
```

```
                    #asm
                    bcf             DIR2
                    #endasm
                    break;

                    case    3:
                    #asm
                    bcf             DIR3
                    #endasm
                    break;
                    } //end switch

            }       //end if positive


        if (Vout <0.0) //negative --> pwm is now reversed
                {
                PWMduty[joint] = PWMperiod*(1.0+Vout/Voutmax);

                switch(joint)
                        {
                        case    0:
                        #asm
                        bsf             DIR0
                        #endasm
                        break;

                        case    1:
                        #asm
                        bsf             DIR1
                        #endasm
                        break;

                        case    2:
                        #asm
                        bsf             DIR2
                        #endasm
                        break;

                        case    3:
                        #asm
                        bsf             DIR3
                        #endasm
                        break;
                        } // end switch

                } // end if negative

    PWMduty[joint] = PWMperiod-PWMduty[joint];

    }

//-------------------------------------------------------------------------//
// ensure all otuput coordinates are within specified limits
```

```
//
//-------------------------------------------------------------------------//
#define LpitchMin  50
#define LpitchMax  200
#define RpitchMin  50
#define RpitchMax  200

#define RyawMin  30
#define RyawMax  220
#define LyawMin  30
#define LyawMax  220


void InsideLimits(void)
        {
        if (LpitchCom < LpitchMin) LpitchCom = LpitchMin;
        if (LpitchCom > LpitchMax) LpitchCom = LpitchMax;

        if (RpitchCom < RpitchMin) RpitchCom = RpitchMin;
        if (RpitchCom > RpitchMax) RpitchCom = RpitchMax;

        if (LyawCom < LyawMin) LyawCom = LyawMin;
        if (LyawCom > LyawMax) LyawCom = LyawMax;

        if (RyawCom < RyawMin) RyawCom = RyawMin;
        if (RyawCom > RyawMax) RyawCom = RyawMax;

        }


//-------------------------------------------------------------------------//
//      calc joint angles
//
//-------------------------------------------------------------------------//

void CalcAngles(float X,float Y, float Z_, float Q, float anhedral)
        {
        float x1,x2,y1,y2,z1,z2,f1[2],f2[2],t1[2],t2[2],N1[2],N2[2];


        // initial minimization of thrust
        x1 = X/2 + Q/2;
        x2 = X/2 - Q/2;

        y1 = Y/2;
        y2 = Y/2;

        z1 = Z_/2;
        z2 = Z_/2;


        // modify output to match anhedral coordinate system
        y1 = y1*AnhedralCOS+z1*AnhedralSIN;
        y2 = y2*AnhedralCOS+z2*AnhedralSIN;

        z1 = z1*AnhedralCOS+y1*AnhedralSIN;
```

```
        z2 = z2*AnhedralCOS+y2*AnhedralSIN;


        // begin inverse kinematics (forward thrust)
        t1[0] = atan2(z1,x1)*40+127;
        t2[0] = atan2(z2,x2)*40+127;

        f1[0] = atan2( y1, sqrt( x1*x1 + z1*z1 ) )*40+127;
        f2[0] = atan2( y2, sqrt( x2*x2 + z2*z2 ) )*40+127;

        N1[0] = (x1+y1+z1)/1.5;
        N2[0] = (x2+y2+z2)/1.5;

        // repeat inverse kinematics (reverse thrust)
        t1[1] = atan2(-z1,-x1)*40+127;
        t2[1] = atan2(-z2,-x2)*40+127;
        f1[1] = -f1[0];
        f2[1] = -f2[0];

        N1[1] = -N1[0];
        N2[1] = -N2[0];



        // determine closest arrangemnt for thruster 1

        distL1 = abs(Lpitch-f1[0]) + abs(Lyaw-t1[0]);
        distL2 = abs(Lpitch-f1[1]) + abs(Lyaw-t1[1]);
        distR1 = abs(Rpitch-f2[0]) + abs(Ryaw-t2[0]);
        distR2 = abs(Rpitch-f2[1]) + abs(Ryaw-t2[1]);


        if ( (distL1) < (distL2) )
                {
                LpitchCom = f1[0];          //convert from radians to 127 centered binary radians
                LyawCom = t1[0];
                ThrusterDuty0 = N1[0]*ThrusterRange+ThrusterMID;
//              fprintf(COM_A,"1:0 %3u,%3u,%3u ",(unsigned int)LpitchCom,(unsigned
int)LyawCom,(unsigned int)ThrusterDuty0);
                }
        else
                {
                LpitchCom = f1[1];
                LyawCom = t1[1];
                ThrusterDuty0 = N1[1]*ThrusterRange+ThrusterMID;
//              fprintf(COM_A,"1:1 %3u,%3u,%3u ",(unsigned int)LpitchCom,(unsigned
int)LyawCom,(unsigned int)ThrusterDuty0);
                }

        // determine closest arrangement for thruster 2
        if ( (distR1) < (distR2) )
                {
                RpitchCom = f2[0];
                RyawCom = t2[0];
                ThrusterDuty1 = N2[0]*ThrusterRange+ThrusterMID;
```

```
//                      fprintf(COM_A,"2:0 %3u,%3u,%3u \n\r",(unsigned int)RpitchCom,(unsigned
int)RyawCom,(unsigned int)ThrusterDuty1);

                        }
                else
                        {
                        RpitchCom = f2[1];
                        RyawCom = t2[1];
                        ThrusterDuty1 = N2[1]*ThrusterRange+ThrusterMID;
//                      fprintf(COM_A,"2:1 %3u,%3u,%3u \n\r",(unsigned int)RpitchCom,(unsigned
int)RyawCom,(unsigned int)ThrusterDuty1);

                        }

                InsideLimits();

                        fprintf(COM_A,"%3.0f,%3.0f,%3.0f,%3.0f
\n\r",(float)distL1,(float)distL2,(float)distR1,(float)distR2);


                }




//-------------------------------------------------------------//
// read the joysticks, convert to commands
//
//-------------------------------------------------------------//

void ReadJoysticks(void)
        {
        ReadAllAnalog();
        Xcom = (X_in-127.0)/127.0;
        Ycom = (Y_in-127.0)/127.0;
        Zcom = (Z_in-127.0)/127.0;
        Qcom = (Q_in-127.0)/127.0;

        }
```

Interrupts.h – handles PWM generation

```
#priority ssp,timer0
//-----------------------------------------------------------//
// PWM-Timer0 interrupt routine
//-----------------------------------------------------------//
#INT_TIMER0
void timer_ie(void)
        {
        #asm
                bcf             INTCON, T0IF            //reset overflow flag
                movlw   TimerOffset             //reset timer with offset
                movwf   TMR0L


                movf    PWMcount,w
                subwf   PWMperiod,w
                btfss   STATUS,Z                        //PWMcount /= PWMperiod?
                goto    test

                bcf             PWM0                            //turn all off, reset count
                bcf             PWM1
                bcf             PWM2
                bcf             PWM3
                clrf    PWMcount



test:   movf    PWMcount,w
                subwf   PWMduty[0],w
                btfsc   STATUS,Z                        //PWMcount = PWMduty0?
                        bsf             PWM0                    //turn on 0

                movf    PWMcount,w
                subwf   PWMduty[1],w
                btfsc   STATUS,Z                        //PWMcount = PWMduty1?
                        bsf             PWM1                    //turn on 1

                movf    PWMcount,w
                subwf   PWMduty[2],w
                btfsc   STATUS,Z                        //PWMcount = PWMduty2?
                        bsf             PWM2                    //turn on 2

                movf    PWMcount,w
                subwf   PWMduty[3],w
                btfsc   STATUS,Z                        //PWMcount = PWMduty3?
                        bsf             PWM3                    //turn on 3

                incf    PWMcount,f

                movf    MotorDisable,w
                andwf   PORTB,f

        #endasm

        if (ThrusterCount>ThrusterPeriod)
```

```
        {
        ThrusterCount=0;
        #asm
        bsf Thruster0
        bsf Thruster1
        #endasm
        }

if (ThrusterCount==ThrusterDuty0)
        {#asm bcf Thruster0        #endasm}

if (ThrusterCount==ThrusterDuty1)
        {#asm bcf Thruster1        #endasm}

ThrusterCount++;
return;
}
```

Works Cited

---

[1] Product Literature at http://www.videoray.com/Products/specs.php
[2] Product Literature "LBV Standard System Specifications"
http://www.cetrax.com/detailed%20lbv%20system%20description_dec%202002.pdf
[3] National Research Council of the National Acadamies. *Future Needs in Deep Submergence Science*. Washington, DC. 2004.
[4] Smits, Alexander. *A Physical Introduction to Fluid Mechanics*. p314.
[5] Smits, Alexander. *A Physical Introduction to Fluid Mechanics*. p318.
[6] Techet, A.H. *Reading #6 from 13.012 Hydrodynamics for Ocean Engineers*. p10
[7] Erikson, Ray. *Mechanical Engineering*. "Foams on the Cutting Edge". Jan 1999.
[8] Product Literature "Copal 16mm Gear Motor HG16 Series 60 :1 Ratio"
http://www.copal-usa.com/PDFs/HG16-060.PDF
[9] Palm, William J. *Modeling, Analysis, and Control of Dynamic Systems*. p.185