



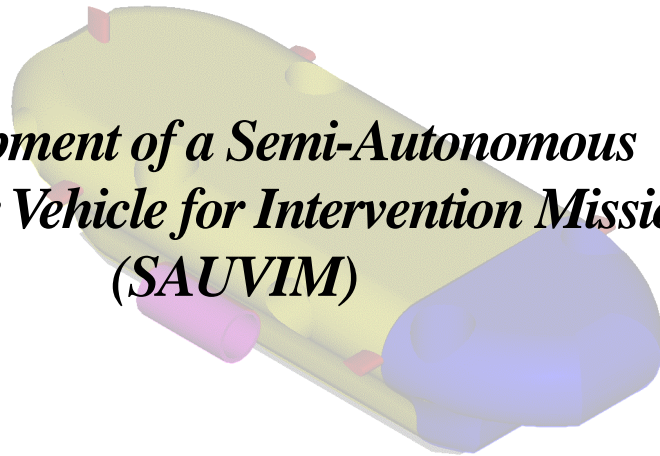
Autonomous Systems Laboratory

2540 Dole St., Holmes 302
Dept. of Mechanical Engineering
University of Hawaii
Honolulu, Hawaii 96822, USA
www.eng.hawaii.edu/~asl

Final Report: Ph



Development of a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM)



Submitted to the Office of Naval Research

Report Documentation Page

Report Date 30062002	Report Type N/A	Dates Covered (from... to) -
Title and Subtitle Development of a Semi-Autonomous Underwater Vehicle for Intervention Missions	Contract Number	
	Grant Number	
	Program Element Number	
Author(s)	Project Number	
	Task Number	
	Work Unit Number	
Performing Organization Name(s) and Address(es) University of Hawaii at Manoa Department of Mechanical Engineering 2540 Dole St., Holmes 302 Honolulu, Hawaii 96822	Performing Organization Report Number	
	Sponsor/Monitor's Acronym(s)	
Sponsoring/Monitoring Agency Name(s) and Address(es)	Sponsor/Monitor's Report Number(s)	
Distribution/Availability Statement Approved for public release, distribution unlimited		
Supplementary Notes The original document contains color images.		
Abstract		
Subject Terms		
Report Classification unclassified	Classification of this page unclassified	
Classification of Abstract unclassified	Limitation of Abstract UU	
Number of Pages 363		

Copyright:

"Copyright ©2002 University of Hawaii, Department of Mechanical Engineering, the Autonomous Systems Laboratory.

Permission to reprint or electronically reproduce any document or graphic in part or in its entirety for any reason is expressly prohibited, unless prior written consent is obtained from the University of Hawaii, Department of Mechanical Engineering, the Autonomous Systems Laboratory and the proper entities."

Summary

Technical Development Report: August 1, 1997 – June 30, 2002

The SAUVIM proposal was submitted under the ONR Annual Announcement of the July 11, 1996 Commerce Business Daily, and the project officially began on August 1, 1997 with an 18-month, \$2.237 million research fund from the Office of Naval Research's Undersea Weapons Technology Program directed by Mr. James Fein. The project was later extended at no cost till October 30, 2000.

The first progress report was submitted to ONR during Mr. Fein's site visit of October 28, 1997. The second progress report was submitted to ONR during the Advisory Committee's (AdCom) site visit of February 24, 1998. The First Annual Progress Report was submitted to ONR in August 1998 and presented during the site visit of September 15-16, 1998. With the departure of Mr. Fein from ONR, Mr. Chris Hillenbrand became the new ONR Program Director for the SAUVIM project. The fourth progress report was submitted during Mr. Hillenbrand's site visit of April 8, 1999. The Second Annual Progress Report was submitted to ONR in August 1999 and presented during the site visit of May 10-11, 2000. A Final Report for Phase I, for work performed from 1997-2000, was submitted in October 2000 and present at the November 14, 2000 site visit. The following official site visit was on October 29, 2001, which coincided with the IEEE Oceans 2001 Meeting in Honolulu. During the 2001 site visit, the SAUVIM vehicle first went in the ocean. During all site visits, each SAUVIM research group gave a presentation of their current progress and demonstration of hardware and software. This is the Final Report for Phase II-A and describes the overall technical development of the project during the 5-year period of 1997-2002. With the departure of Mr. Hillenbrand, Dr. David Drumheller will become the new ONR Program Director for the SAUVIM Project and will be officially introduced at the next site visit which is scheduled for July 18, 2002 where the first shallow water ocean trials will be conducted. Phase II-B will be performed from July 1, 2002 to December 31, 2003.

Objective

The primary research objective is to develop a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM). Unlike the fly-by autonomous underwater vehicles (AUV), SAUVIM will have a manipulator work package. It will require an advanced control system and a precise sensory system to maintain high accuracy in station keeping and navigation.

Background

Most intervention missions - including underwater plug/unplug, construction & repair, cable streaming, mine hunting, and munitions retrieval - require physical contact with the surroundings in

the unstructured, underwater environment. Such operations always increase the level of risk and present more difficult engineering problems than fly-by and non-contact type operations. For these intervention operations, the vehicle requires a dexterous robotic manipulator; thus the overall system becomes a high degree-of-freedom (dof), multi-bodied system from the coupling effects of the vehicle and the manipulator motions. These operations require precise force/torque feedback with high degree of accuracy even in the presence of unknown, external disturbances, i.e. undersea currents. All these issues present very complex engineering problems that have hindered the development of AUVs for intervention missions. Currently, the state-of-the-art in machine intelligence is insufficient to create a vehicle of full autonomy and reliability, especially for intervention missions.

The development of '*undersea robots that can intelligently work with arms than just swim*' will have a great impact on worldwide underwater robotic vehicle technology and provide a cost-effective engineering solution to many new underwater tasks and applications that fly-by type submersibles have not been able to handle. The proposed vehicle – SAUVIM - is in response to the current local and national needs for the development of this technology and will ultimately be useful in many intervention missions. One such application field is the Pacific Missile Ranging Facility (PMRF) in Hawaii.

Development

The SAUVIM project was proposed as a two-phase research and development program. Phase I has three parts: (1) to study the major research components, (2) to develop and integrate the basic software and hardware of SAUVIM, and (3) to test the vehicle in a shallow water environment. Phase II is a continuation and completion of the research and development of Phase I with deep water environment testing.

As stated in the original proposal, the project consists of five major components:

- Adaptive, Intelligent Motion Planning;
- Automatic Object Ranging and Dimensioning;
- Intelligent Coordinated Motion/Force Control;
- Predictive Virtual Environment; and
- SAUVIM Design.

During the Phase I period, there have been approximately sixty people supported by this ONR project. Currently, there are twenty-three people working on the project - 2 faculty members, 7 full-time staff members, 3 part-time staff members, 6 graduate students and 5 undergraduate students. The Advisory Committee was formed to provide technical advice and direction by reviewing research directions and progress, and to provide advice and assistance in exploring potential applications and users. The four-member Advisory Committee consisted of Mr. Fred Cancilliere of the Naval Undersea Warfare Center, Dr. Alexander Malahoff of the University of Hawaii, Dr. Homayoun Seraji of the Jet Propulsion Laboratory, and Mr. Dick Turlington of the Pacific Missile Range Facility. Two additional members - Dr. Paul Yuen of the University of Hawaii and Mr. James Fein, the former ONR Program Director - have been included in the Advisory Committee. The SAUVIM revised organizational chart is shown in Figure A, and a simplified SAUVIM schedule is shown in Figure B.

- Adaptive, Intelligent Motion Planning (AIMP) – The AIMP aims at developing SAUVIM's motion planning, which is intelligent and adaptive in that the system is capable of decision-making at a task or mission level and can deal with unknown or time-varying environment. Motion planning for an AUV can be decomposed into path planning and trajectory generation, although they are not completely independent of each other. Path planning is a computation and optimization of a collision-free path in an environment with obstacles. Trajectory generation is the scheduling of movements for an AUV along the planned path over time. To simultaneously compensate for these objectives, a genetic algorithm (GA) based 3D-motion planner is implemented for both an off-line and on-line cases. An off-line case is when an environment is a known and static, while an on-line case must be capable of modifications in response to dynamic, environmental changes. The utilization of GA-based approach has two advantages: 1) it is adaptive and 2) the dimension of space has less effect on performance than other methods.

The AIMP software has gone through three version upgrades. The first was *Version 1.alpha*, which integrates the off-line and on-line algorithms in C with a graphic user interface using OpenGL. This software version was tested on the Autonomous Systems Laboratory's autonomous underwater vehicle - ODIN. The second was *Version 1.0*, which integrates the path planning and trajectory generation algorithms. The third was *Version 1.1*, which optimizes the original software organization and data structures, and includes a database of mapping data on the main memory. Also, a Software Development Process (SDP) has been developed and implemented to oversee the various developments in software version changes. Several papers have been published in these subjects. A final, optimized version will be tested on the upgraded ODIN (ODIN III) in Phase II-B.

- Automatic Object Ranging and Dimensioning (AORD) – The main objective of the AORD is to develop a multiple sensor configuration to be utilized during SAUVIM's intervention missions. This three-sensor system consists of (1) a laser ranging sensor (LRS), (2) a passive arm sensor (PA) and (3) a manipulator homing sensor (MHS). The laser ranger, the homing sensor, and the passive arm have all been designed and prototyped. According to initial feasibility and prototype tests, all three sensors showed good performance.

The underwater version of the PA has been fabricated and has been assembled. The PA is made of 6061-Aluminum, and it has two three-axis gimballed joints and a single-axis hinge joint. The entire PA structure is compensated with mineral oil. It utilizes the original software developed for the prototype. The kinematics of the PA have been re-verified using various symbolic math packages. It has also been rewired for optimal performance. The PA was simulated with the active arm to conduct feasibility studies in obtaining active manipulation position. The PA will be mounted with the active manipulator in Phase II-B for experimental results.

The underwater versions of the LRS and the MHS are in the process of fabrication and assembly. The camera housings for both systems have been manufactured using 6061 aluminum with vacuum-sealed lens and underwater connectors have been ordered. The software for both systems has been developed using the prototypes.

- Intelligent Coordinated Motion/Force Control (ICM/FC) – The major objective of the ICM/FC is simple yet complex. The control of an AUV and its manipulator is a multi-bodied, dynamic problem of vast unknowns; therefore, this task was subdivided into four sub-tasks, which were Theoretical Modeling (TM), Low-Level Control (LLC), High-Level Control (HLC), and Dry Test Design and Set-up (DTDS). However, with the arrival of the 7-dof, underwater

manipulator, the TM and DTDS were combined to form a common group – Manipulator Control and Test Platform (MCTP). Also, a Localization and Navigation (LN) group was spun-off the LLC group due to the vastness and complexity of the LN material. The LN group is devising a hybrid localization and navigation methodology that will suffice in understanding the geophysical, terrain-matching and dead-reckoning aspects for proper navigation. An integrated data fusion methodology is also being devised to quickly and correctly digest the immense amounts of data from the sensors, which undoubtedly has mass abundance of noise and errors.

The MCTP was recently developed to accelerate the progress in the TM and DTDS sub-tasks. With the acquirement of the Ansaldo 7-dof manipulator and constraints in time, the focus has been changed to the development of the Ansaldo software in conjunction with the manipulator kinematics, dynamics, force-control and coordinated motion control modules. Currently the Ansaldo manipulator runs off the VME bus system using VxWorks and Matlab with Simulink. Development in the “rapid prototyping, graphic software” has been the central point in enhancing the complex, underwater dynamic actions and reactions. The manipulator control codes have been developed to perform simplified force/torque tasks, path optimization around singularity points, and basic collision avoidance techniques. The manipulator development will remain self-contained until its connection to the vehicle prior to wet testing.

The LLC has still has two objectives: 1) to design and develop an advanced vehicle control system for navigation and hovering, and 2) to design and develop an advanced coordinate motion/force control system of the vehicle and manipulator during the intervention mode. However, with the creation of the LN group, the emphasis will be on the integration of the localization and navigation techniques to the basic motion and hovering tasks. The development of the coordinated motion/force control system is being explored from two separate platforms. As the MCTP development continues, the LLC is optimizing the hovering and station-keeping methodologies on the ODIN vehicle. Once both the manipulation and hovering control codes are considered sufficient, a small, 2-dof manipulator will be added to ODIN for coordinated motion/force control experiments. Since the control code developments have been significant, these tests should take place in Phase II-B.

In the LN group for Phase II-A, the focus has been on efforts in obtaining high performance in navigation and hovering, and the development of a localization technique. The navigation and hovering uses the on-board scan sonar, altimeter sonar, inertial navigation unit, Doppler velocity logger and pressure sensors. The localization technique being developed is a hybrid approach combining various techniques such the evidence grid approach. The grid method accumulates occupancy evidence from an array of spatial locations and slowly resolves the ambiguities as the AUV travels. Both the navigation and hovering, and localization techniques are being tested on ODIN.

HLC’s objective is to develop a supervisory control module that will minimize human involvement in the control of the underwater vehicle and its manipulation tasks. This module involves the development of high-level task planning where a mission is always composed of two parts: the goal and the method of accomplishment. In other words, "what do I need to do" and "how do I do it." Following this strategy, a new high-level architecture of vehicle control, named the Intelligent Task-Oriented Control Architecture (ITOCA), is being developed for SAUVIM. ITOCA is an effective and efficient operation running on the VxWorks real-time operating system (RTOS) environment. ITOCA is four layers: a planning layer, a control layer, an execution layer and an evaluation layer. Every mission is broken into many smaller missions and the simplest mission is considered a task. The combination of different tasks in different

sequences accomplishes various missions. Presently, a preliminary, pilot algorithm is being considered and developed. The HLC is one of the major research tasks for Phase II-B.

- Predictive Virtual Environment (PVE) – The PVE is aimed at developing a supervisory monitoring system for SAUVIM to smoothly and realistically integrate mapping data with on-line sensory information even in the midst of delayed and limited information. This virtual reality (VR) based system must also be able to accurately predict the current status and location of the vehicle under these conditions. The development for the PVE has been modular. The various modules are: the SAUVIM Simulation Software (SSS); the SAUVIM Video Overlay Software (SVOS); the Communication Software (CS); and the artificial neural network (ANN) Video Prediction Software (VPS). The SSS has been upgraded from its *Version 1* to *Version 1.1*, which includes the incorporation of a Magellan spaceball mouse, an accurate 3D graphical model of SAUVIM and the Ansaldo manipulator, scene-smoothing methods using interpolation techniques, and an easy-to-use user interface. The SVOS was developed to overlay video images of the seafloor (texture and color) to the graphic images to provide more accurate monitoring of the vehicle, manipulator and environment. The CS for SAUVIM is an extension of the NSF's DVECS project. Currently the DVECS (Distributed Virtual Environment Collaborative Simulator) system uses a cellular phone to communicate the vehicle data from the test-site to the monitoring computer located on campus for data fusion. Experiments are being conducted with the ODIN AUV. The experiments of ODIN are projected via an ElectroHome Marquee 8500 CRT projector coupled with multiple Stereographics (SG) emitters and SG CrystalEyes glasses. Finally, the VPS is in its infancy; however, several ANN methods have been tested for optimal computation time and position accuracy. Experiments have been performed in the laboratory and have generated positive results. Due to the high maintenance costs of SGI workstations, the overall virtual reality and monitoring system, which includes the video prediction, is being transformed to a much more stable and inexpensive personal computing system. This system has been fully designed and the implementation and testing will commence in Phase II-B.
- SAUVIM Design (SD) – This task is still the main objective of the SAUVIM project. It is an effort to design and develop efficient, reliable hardware/software architectures of SAUVIM. Due to the immense demand of this task, it is divided into five sub-tasks, which are Reliable, Distributed Control (RDC), Mission Sensor Package (MSP), Hydrodynamic Drag Coefficient Analysis (HDCA), Mechanical Analysis and Fabrication (MAF), and Mechanical-Electrical Design (MED).

The goal of RDC is to develop a reliable and efficient computing architecture for signal and algorithmic processing of the entire SAUVIM system. The proposed system is a multi-processor system based on a 6U VMEbus and the VxWorks real-time operating system. This system is capable of high processing throughput and fault tolerance. Currently the system consists of two VMEbuses, which are the navigation control system and the manipulator control system. The main VMEbus, or the navigation control system, has two Motorola M68060 CPU boards, a multi-port RS232 interface board, and an I/O board with a Pentium MMX processor based PC104+ board, which is connected via a RS232 port. The navigation control system handles the communication, supervision, planning, low-level control, self-diagnostics, video imaging, etc. The data exchange between the two CPUs is conducted via shared memory. The second VMEbus, or the manipulator control system, has one Motorola M68040 CPU and an I/O board. Two PC104 boards are connected serially to this CPU. The manipulator control system is independent and dedicated to the manipulator control. Many of the hardware components have been tested and are being interface with its respective software systems. Various optimization

changes have been implemented to minimize communication and computation. The overall hardware and software architectures have been completed and integrated. Initial tests for the RTOS architecture has been integrated with the SAUVIM vehicle hardware and tested as individual components. The overall vehicle control will be conducted in near future experiments; and the sensor feedback used to close the loop. This development will continue throughout the vehicle's development process.

The objective of the MSP is to provide semi-continuous records of SAUVIM water depth, temperature, conductivity, computed salinity, dissolved oxygen, magnetic signature of the seafloor, pH and turbidity during the survey mode. In the intervention mode, the MSP also provides compositional parameters at a selected seafloor target, including pumped samples from submarine seeps or vents. The MSP is an independent system with its own PC 104 CPU and its own power supply residing in a separate pressure vessel. All of the sensors have been purchased, and an initial field test at the Loihi Seamount has been conducted. Continual tests are being conducted to optimize the scientific sensor data-gathering capabilities. The communication from the MSP and the vehicle CPUs are being optimized. The MSP has been mounted on the SAUVIM vehicle.

The HDCA is used to determine the hydrodynamic coefficients via a numerical solution of full Navier-Stokes equations using PHOENICS, a commercial computational fluid dynamics (CFD) code. Initial results from the PHOENICS software have produced mixed results. The current vehicle fairing has produced a drag coefficient of 0.40; however, it has not yet been verified. Other CFD software and model testing is being conducted to verify the drag coefficient results before the implementation of the vehicle fairing on SAUVIM. There has been no significant development in this task group. The hydrodynamic coefficients will be obtained through vehicle motion experiments in the near future to aid in simulator developments.

The MAF has three objectives. Its primary goal is to design and fabricate composite pressure vessels with end caps and connector openings for full ocean depths taking stress, buckling, hygrothermal effects, and fatigue analysis into account; and its two secondary goals are to design and fabricate the SAUVIM fairing and to analyze the SAUVIM frame. A thorough analysis and comparison of the Ti-6Al-4V, AS4/Epoxy, and AS4/PEEK pressure vessels manifest the advantage of composite materials in reduction of weight, size and strength. Using these results, a scaled model prototype using AS4/PEEK has been fabricated and tested. A 1/3 sized prototype is being fabricated and will also be tested. For the shallow water vehicle test, a full-sized, fiberglass pressure vessel with aluminum end caps have been manufactured and tested. These vessels are being used to determine the final hardware layout. The aluminum frame has been designed and fabricated. A full-ocean depth pressure vessel of AS4/PEEK has been developed and tested. However, due to several unknowns regarding composite pressure vessels, the vehicle has been equipped with 1000+ meter-depth, aluminum pressure housing. These aluminum housings will be used for the shallow and mid water depth experiments. The initial fairing analysis has been developed and expanded. Fairing optimizations have been considered. Various manufacturing and molding methods have been explored to fabricate the initial fairing in-house. The fabrication will commence in Phase II-B.

The MED is the integration of the mechanical and electrical components for SAUVIM. First, the design specifications were established for the fairing, frame, instrument pressure vessels, buoyancy systems, mission sensor, passive arm and robotic manipulator tasks. Second, after scrutinizing review of SAUVIM's major components - i.e. sensors, actuators and infrastructure - in terms of power consumption, compatibility, weight distribution, buoyancy distribution,

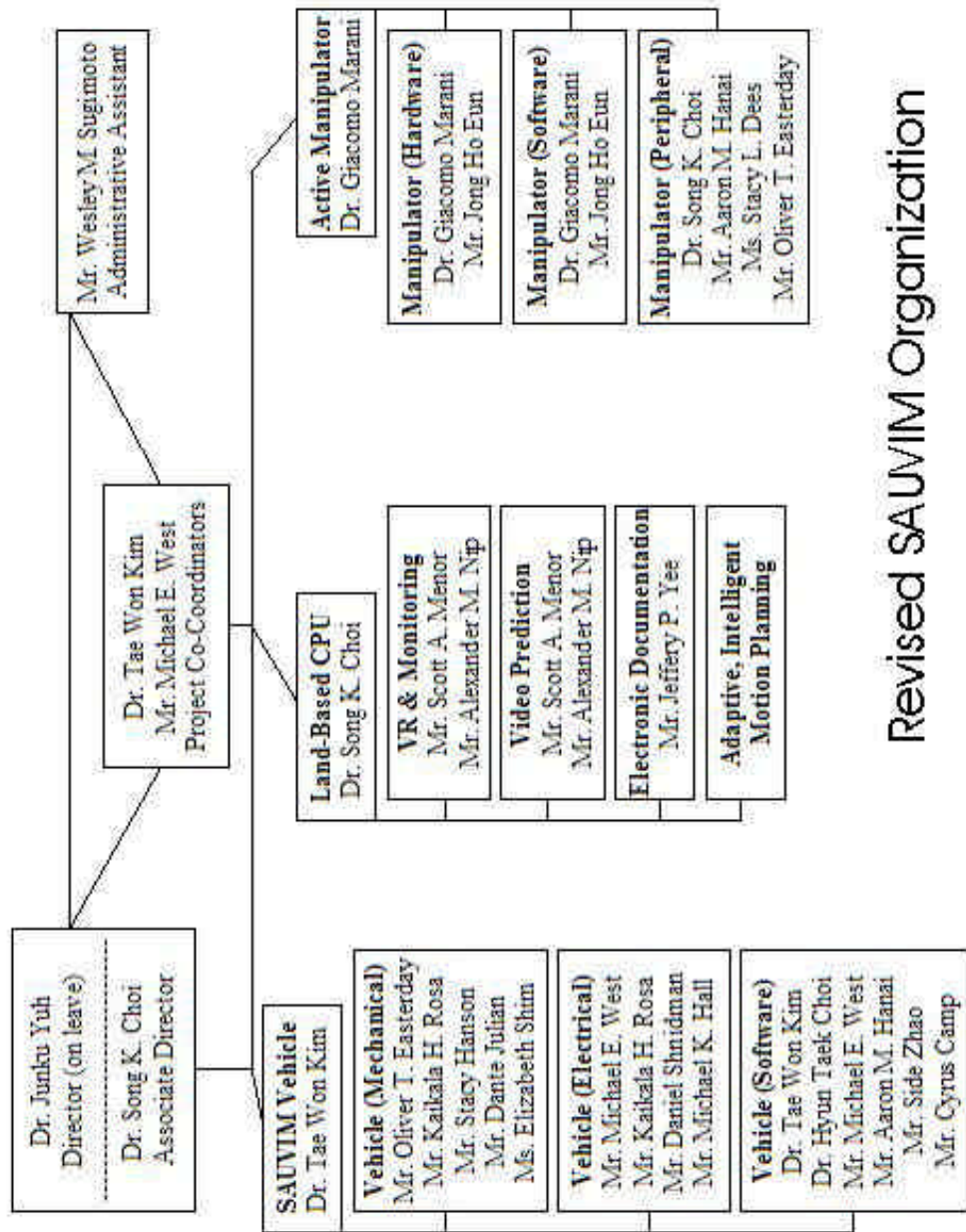
hydrodynamic effects and task effectiveness, all major components have been purchased. Technical drawings of the vehicle frame, fairing, and related sub-structures have been completed. Most of the mechanical and electrical components have been fabricated and integrated with the overall electrical layouts. The vehicle has been wet-tested twice and will be tested again in July 2002. Autonomous shallow water tests will begin in Phase II-B.

The main body of this report is devoted to the detailed descriptions about the major technical developments and achievements during the period of 1997-2002.

SAUVIM Project
June 30, 2002

Table of Contents

	<u>Page</u>
Summary	i
Table of Contents	viii
SAUVIM Revised Organizational Chart	1
SAUVIM Simplified Gantt Chart	2
Project Report	3
• Adaptive, Intelligent Motion Planning	4
• Automatic Object Ranging and Dimensioning	12
• Intelligent Coordinated Motion/Force Control	29
Manipulator Control and Test Platform	30
Low-Level Control	68
Localization and Navigation	97
High-Level Control	105
• Predictive Virtual Environment	106
• SAUVIM Design	124
Reliable, Distributed Control	125
Mission Sensor Package	146
Hydrodynamic Drag Coefficient Analysis	159
Mechanical Analysis and Fabrication	160
Mechanical-Electrical Design	231
References	325
Appendix	335
Publications	350



Revised SAUVIM Organization

Figure A: SAUVIM Revised Organizational Chart

Simplified SAUVIM Schedule

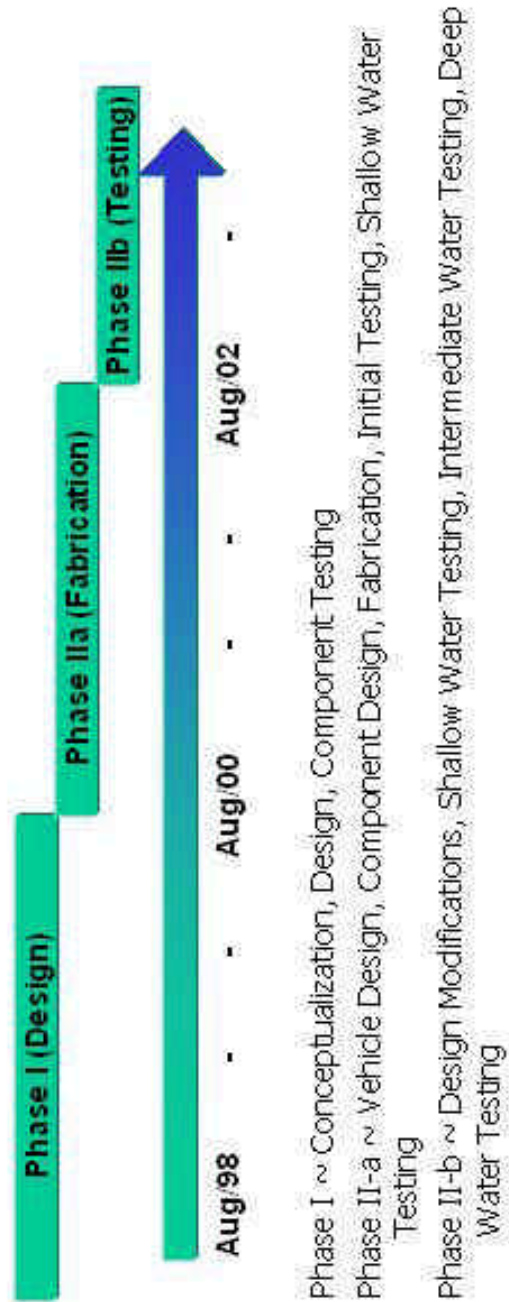


Figure B: SAUVIM: Simplified Gantt Chart

Adaptive, Intelligent Motion Planning (AIMP)

Project Leader(s): Dr. Tae Won Kim & Dr. Song K. Choi
Personnel: - none -
Past Project Leader(s): Dr. Kazuo Sugihara
Past Personnel: Mr. John Smith, Dr. Shenyan Zhen, Mr. Haidong Chang, Ms. Hongshi Chen, Mr. Xihua Xu, Mr. Dwayne Richardson, Mr. Sonny Kim. Mr. Jangwon Lee & Mr. Yongcan Zhang

Objectives

This sub-project aims at developing the motion planning system for SAUVIM. It is intelligent and adaptive in the sense that the system is capable of decision-making at a task or mission level and can deal with an unknown or time-varying environment.

There are three basic objectives.

- To develop an off-line 3D motion planning algorithm.
- To develop an on-line 3D motion planning algorithm.
- To develop an adaptive, intelligent motion planning system by integrating the off-line and the on-line planning algorithms.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

Introduction

Motion planning of an autonomous underwater vehicle (AUV) can be decomposed into path planning and trajectory generation, although they are not completely independent of each other. Path planning is to compute a collision-free path in an environment with obstacles and optimize it with respect to some criterion. Trajectory generation is to schedule the movement of an AUV along the planned path over time. This paper addresses the path planning in 3D space.

An algorithm for path planning is said to be off-line if an environment is a known, static terrain and it computes a path in advance. Otherwise, it is said to be on-line. An on-line algorithm must be capable of modifying a path in response to environmental changes such as a mobile obstacle and detection of an unknown obstacle. We propose a genetic algorithm (GA) that can be used for both off-line and on-line path planning [Sugihara97, Sugihara98, Sugihara99].

The GA-based approach has two advantages. First, it is adaptive in the sense that it can respond to environmental changes and adjust a path “globally” to a new environment. Second, the dimension of space has much less effect on performance in the GA-based approach than others. Since path planning in 3D space is known to be computationally intractable, this makes the GA-based approach more attractive.

Preliminaries

Suppose that an AUV needs to move from a start point s to an end point e in 3D space. By normalizing the unit of each dimension appropriately, consider the cubical space where s and e are on vertical edges located diagonally to each other. Assume that a path between s and e is discretized with

reasonable granularity as a sequence of adjacent cells in an n -by- n -by- n grid corresponding to the 3D space. Without loss of generality, the coordinate system can be defined so that s and e are located at $(0,0,a)$ and $(n-1, n-1, b)$, respectively.

Note that this discretization is applied to only the representation of a path. Map data may be represented in any way as long as we can efficiently access information of a given grid cell such as whether there is an obstacle at the grid cell. There is no restriction on shapes of obstacles.

We consider two types of obstacles: Solid obstacles and hazardous obstacles. A path cannot intersect a solid obstacle while it may intersect a hazardous obstacle at the expense of extra cost in proportion to the hazardous obstacle's weight, which represents various cost factors.

The distance $d(i,j)$ from cell i to its adjacent cell j in the 3D grid is the Euclid distance from the center c_i of cell i to the center c_j of cell j . The length $length(i,j)$ from i to j is defined as $d(i,j)(1+w(c_i,c_j))$, where w denotes the average weight between the locations c_i and c_j . The length of a path between s and e is the sum of lengths between two consecutively adjacent cells on the path.

A problem of path planning in 3D space is defined as follows.

Input: n by n by n grid, start cell $(0,0,a)$, end cell $(n-1,n-1,b)$, obstacles and their weights

Output: A path between cells $(0,0,a)$ and $(n-1,n-1,b)$ such that the length of the path is minimum, subject to the following.

- (a) the path does not intersect any solid obstacle and
- (b) the path meets limitations on the maneuverability of an AUV such as the minimum turning radius

In 2D space, a path is said to be monotone with respect to x -coordinate (x -monotone for short) if no lines parallel to y -axis cross the path at two distinct points, i.e., the projection of the path on x -axis is non-decreasing. Similarly, y -monotone is defined.

In 3D space, a path is said to be xy -monotone if no line parallel to z -axis cross the path at two distinct points. A projection of a path on x - y plane is called xy -projection of the path, which is a path in 2D space. Similarly, xz -projection and yz -projection are defined.

Path Planning GA

A genetic algorithm (GA) [Goldberg89] for an optimization problem maintains a population of individuals, where each individual corresponds to a candidate solution and the population is a collection of such potential solutions. In GA, a binary string commonly represents an individual. The mapping between solutions and binary strings is called a coding. The number of individuals in a population is called the population size. GA repeatedly transforms the population by using a mechanism analogous to biological evolution. The mechanism includes the following steps.

Fitness Evaluation: The fitness corresponding to an optimization criterion (i.e., objective function) is calculated for each individual.

Selection: Some individuals are chosen from the current population as parents, based on their fitness values.

Recombination: New individuals (called offspring) are produced from the parents by applying genetic operators such as crossover and mutation.

Replacement: Some individuals (not necessarily parents in general) are replaced by some offspring.

The population produced at each transformation is called a generation. By giving highly fit individuals more opportunities to reproduce, the population becomes likely to include “good” individuals throughout generations.

There are 4 major components to be designed in a GA:

- coding,
- fitness function,
- configuration of genetic operators, and
- parameters of genetic operators.

The coding used in our GA decomposes a path in 3D space into three projections of the path, namely, xy-projection, xz-projection and yz-projection. Obviously, there exists at least one triple of such 3 projections, which represent an arbitrarily given path in 3D space. However, it is not always true that an arbitrarily given triple of projections represents a unique path in 3D space. To guarantee the uniqueness, we assume the following.

Assumption 1: A path in 3D space is xy-monotone.

Assumption 2: The xy-projection of the path is x-monotone and y-monotone.

Then, a binary string as described below represents each projection. Finally, the resulting 3 binary strings are interleaved bit by bit. The reason for interleaving is that crossover can transform all the projections of a path at the same time.

Since xy-projection is x-monotone (and also y-monotone), it can be represented by a row-wise (or column-wise) sequence of $n-1$ pairs of direction and distance such that each pair specifies a segment of the projection between two consecutive rows (or columns). Thus, xy-projection is coded into a binary string as follows. The first bit β indicates that a path is x-monotone if $\beta=0$ and it is y-monotone if $\beta=1$. A block of $3+\text{ceiling}(\lg n)$ bits represents direction and distance on each column or row. The first 2 bits of each block denote the direction, e.g., 00 (vertical), 01 (upper diagonal), 10 (horizontal) and 11 (lower diagonal) in case of $\beta=0$; 00 (horizontal), 01 (left diagonal), 10 (vertical) and 11 (right diagonal) in case of $\beta=1$. The other bits of the block denote the distance as a signed integer if the direction is 00; otherwise they are ignored.

To interpret binary strings as paths and evaluate their fitness values, we use the following convention, which produces more “valid” paths in generations and hence improves the performance of our GA. If consecutive blocks of a binary string make the corresponding path go beyond boundary cells, that part of the path is regarded as a straight-line short cut along the boundary cells. Note that we leave the binary string as it is, since the non-interpreted sub string may become valid and useful later if genetic operators inherit the sub string to new generations.

In our GA, the initial population is created randomly except some special strings, which correspond to the straight-line or L-shape paths between s and e in the x-monotone and y-monotone representations. Even if a random binary string has large perturbation, the convention of “chopping off a path along the boundary” often makes the random string represents a path consisting of a few line segments.

After we conducted simulation of the GA on our GA Toolkit [Smith96], we have decided the configuration of the GA consisting of the following 3 operators. First, roulette tournament selection with geometric ranking is used to choose parents and mate them. Second, 1-point crossover is applied to the parents. Third, mutation is applied to all individuals bit by bit randomly.

On-line Path Planning

In this section, we discuss how to apply the GA presented in Section 3 to a partially known environment in real time while an AUV is moving. When the GA-based off-line path planning produces a path, a population at the last generation is regarded as the initial population of a GA for on-line path planning.

From the viewpoint of on-line path planning, environmental changes occur due to either update of map data by sensory information or the movement of an AUV. The environmental changes may cause changing the current path in order to avoid collision and improve the path with respect to an optimization criterion in a new environment. Thus, we separate two issues regarding how to incorporate sensory information into the map data and how to update a population of the GA while the AUV is moving.

At every generation in execution of the GA for on-line path planning, the GA refers to the current world model (i.e., the current map data) in order to evaluate the fitness of each individual in the current population. The world model is stored in a database on board and continuously updated with sensory information. The adaptivity of the GA realizes a modification of the current path in response to changes of the world model due to input of sensory information. We are conducting simulation to evaluate how quickly the GA adapts for environmental changes. Simulation study of the GA on our GA Toolkit suggests that the GA keeps a population diverse enough to find an alternative path at the next generation immediately after an environmental change. This feature is very important for SAUVIM.

Trajectory Generation

The path-planning program produces a path represented by a sequence of adjacent cubes in a 3D grid structure. Such a path is intuitively viewed as a corridor, which begins at the start, passes intermediate waypoints, and ends at the destination. Once the path is produced, a smooth curve inside the corridor must be generated (Figure AIMP-1).

Input: The path, the start point, the destination, the initial velocity, and the final velocity.

Output: A curve such that it stays inside the path and its tangent lines at the start and the destination are same as vectors of the initial and final velocities, respectively.

The Hermite curve is used to solve this problem as follows. We sequentially produce a curve between two consecutive waypoints including the start and the destination, beginning from the start. Suppose that a curve is represented in a parametric form with 4 constants a , b , c and d ,

$$p(t) = a t^3 + b t^2 + c t + d$$

where $p(t)$ denotes a vector of 3 coordinates $x(t)$, $y(t)$ and $z(t)$ such that $0 \leq t \leq 1$. With the boundary conditions at the first waypoint ($t=0$) and the second waypoint ($t=1$), $p(t)$ must satisfy the following, where v_1 and v_2 are the velocities at the first and second waypoints, respectively.

$$p(0) = d$$

$$p(1) = a + b + c + d$$

$$p'(0) = v_1 = c$$

$$p'(1) = v_2 = 3a + 2b + c$$

By solving this system of linear equations, we can determine the constant coefficients and hence compute the curve $p(t)$, which is intuitively S-shaped.

Next, the movement of a vehicle on the curve must be scheduled. To develop the first version of software for trajectory generation, we simplify this scheduling problem by assuming that a vehicle's speed changes in the way shown in Figure AIMP-2 and the vehicle's orientation is always same as a tangent line of the curve at the current location.

Input: The generated curve, the initial speed at the start, the final speed at the destination, constant acceleration, constant deceleration, the cruising speed, and the unit time Δ in scheduling.

Output: A sequence of locations for the vehicle to be located on the curve at each time $i \Delta$ where i is a natural number.

In general, choices of a curve and a schedule on it are interrelated. Hence the curve generation and scheduling should be solved together in order to optimize them simultaneously. For example, the maximum cruising speed may depend on the curvature and the maneuverability of a vehicle (e.g., minimum turning radius). In addition, the vehicle's dynamics should be taken into account. This is one of the issues to be investigated in future.

AIMP Software

AIMP Software Version 1.0 alpha

The off-line and on-line path planning algorithms were implemented together in C.

A graphical user interface was implemented by using OpenGL.

Outputs of the path-planning program for both off-line and on-line planning were tested in experiments of the vehicle ODIN in a pool.

AIMP Software Version 1.0

A program for trajectory generation, which generates a smooth curve for a path computed by the path-planning program and schedules the movement of SAUVIM on the curve, was developed in C. Algorithms for trajectory generation will be explained below.

The programs for path planning and trajectory generation were integrated as software for motion planning.

AIMP Software Version 1.1

A database of mapping data on the main memory was implemented and incorporated into the motion planning software.

Major revisions of source code of the AIMP Software Version 1.0 were made, which greatly improved the organization and data structures of the Version 1.0.

Documentation was revised in accordance with *SAUVIM Software Development Process* (which will be explained below).

Screen snapshots in a demonstration of Version 1.1 are shown in Figures AIMP-3 and AIMP-4, where the submarine volcano, Loihi, was used as the initial terrain and unknown obstacles were hypothetically added in the way that they obstructed a path.

Software Development Process

In order to assure the software quality, control version upgrades, and manage the software documentation, we designed and implemented the standardized process of software development described as follows.

Every Week (done by each member)

Take backup of everything of the current software and a progress report together with each member's activity log on ZIP disks. The backup hierarchically consists of the following.

sw_name/	(software name)
ver#/	(version number)
src/	(source code)
doc/	(documents)
data/	(data or examples for testing)
demo/	(binary files or a compressed file for a demo)
report/	(progress reports with activity log)
backup_log	(backup history on the ZIP disk)

Once Every One or Two Months (done by all members)

The latest version of software is tested by a member other than its author(s) as follows.

Try to reinstall the software from the backup from scratch.

Try to reproduce and run a demo of the software.

Give the author comments/suggestions based on this experience.

The author(s) will revise the software including documentation accordingly.

Upon the End of a Term or Substantial Progress Made (done by a supervisor)

Compile new documents and/or revise previous documents.

Create a new version of software.

Certify it as the latest version on the backup and keep it in duplicate.

A version upgrade should be done as follows.

1. Clean up source code of software.

Test whether it works after the cleanup.

Write informative inline comments.

Add version number, author(s)' name(s), date, and copyright at the beginning of every source file.

2. Write the following documentation about the software.

Requirements Specification: Objectives, functionality (what to do, especially, input/output relationships), hardware/software environments, etc.

Design Specification: Overall architecture of your software, module structure, caller/callee relationships with data flow, algorithms, data structures, etc. Use of diagrams is strongly desired.

Reference Manual: Any implementation details, which are important for other programmers to know in order to maintain for correction, improvement and adaptation.

User's Manual with README: Instructions for installation and operation. Use of screen snapshots is strongly suggested.

Testing Document (optional, but desirable): Methods/tools for debug of the software, input/output data in testing, performance evaluation, etc.

In case of an upgraded version, *Upgrade Note* is also required and describes what are modifications, why & how the modifications are made, which parts of software and documents have major changes, etc. With the upgrade note, a person who has some knowledge about the previous version can save time to understand the upgraded version.

Future Tasks (Phase II Tasks)

The following tasks are expected in Phase II:

- To test the AIMP software on SAUVIM in the ocean.
- To investigate integration of the vehicle's dynamics into our GA-based motion planning.

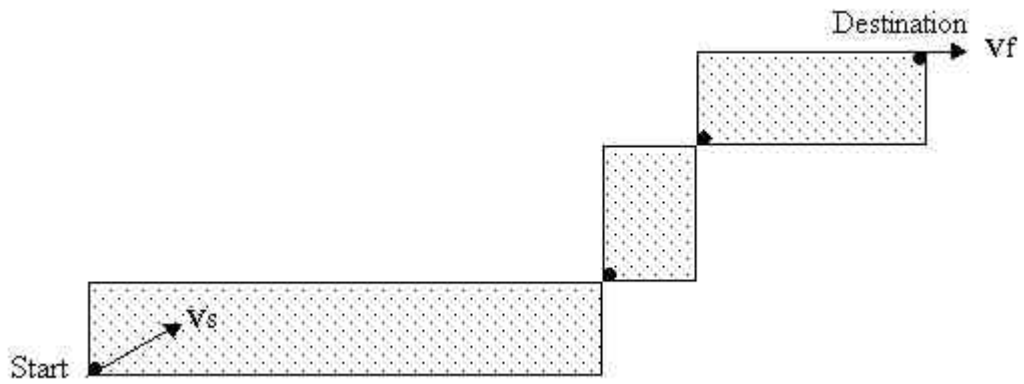


Figure AIMP-1: Inputs for generation of a curve from a path.

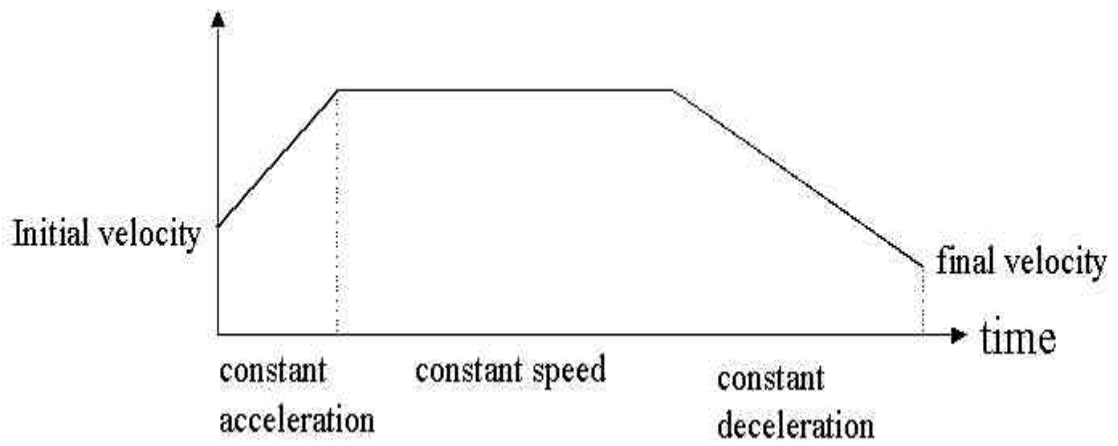


Figure AIMP-2: A schedule of speed on the curve.

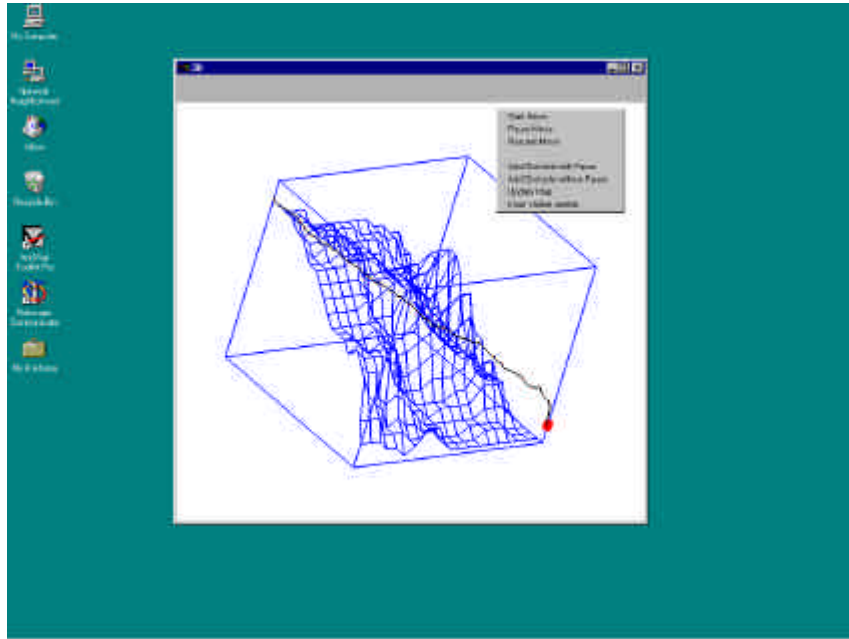


Figure AIMP-3: A screen snapshot of Version 1.1 before unknown obstacles are added.

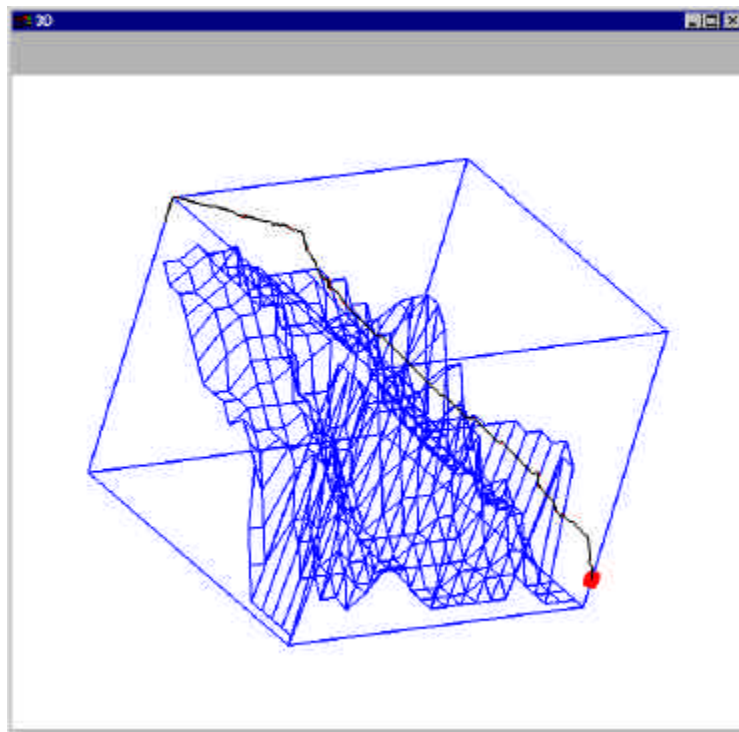


Figure AIMP-4: 3-D output after unknown obstacles are added.

Automatic Object Ranging and Dimensioning (AORD)

Project Leader(s): Dr. Song K. Choi, Dr. Tae Won Kim & Dr. Junku Yuh
Personnel: Mr. Aaron Hanai & Mr. Oliver Easterday
Past Project Leader(s): Dr. Curtis S. Ikehara
Past Personnel: Mr. Marc Rosen, Mr. Mike Kobayakawa, Mr. Henrik Andreasson & Mr. Anders Andreasson

Objectives

To develop a multiple sensor configuration to be utilized during SAUVIM's intervention mission. The configuration will allow accurate vehicle positioning, workspace dimensioning and ranging, and manipulator homing to the task object.

A three-sensor combination is being developed to accomplish this task: 1) a passive arm sensor for vehicle positioning and station keeping, 2) a parallax-based laser ranger for workspace dimensioning and ranging, and 3) a manipulator mounted homing sensor system that will allow accurate homing of the manipulator gripper to the workspace location.

Passive Arm

The passive arm system (PA) is a multi-jointed mechanical arm that utilizes direct kinematics to sense the proximity and orientation between its two ends. Specifically, each axis of each joint senses its current angular position through the use of a potentiometer. One end of the passive arm is mounted and fixed within the forward cavity of the SAUVIM vehicle on the arm-tray, opposite to the Ansaldo robotic arm. The other end of the arm is to be attached to or placed near the task site, by means of an electro-magnet, during intervention tasks. Hence, any changes in the relative proximity between the vehicle and the task site will be sensed by angular displacements in the joints of the PA. The PA will be stored in a crutched position during cruise mode mission phases. The vehicle's robotic arm will be used to crutch and un-crutch the PA as well as position it for use during active arm intervention tasks.

Laser Ranging System

The laser array ranging system is a sensor system designed to provide the vehicle's control system with navigation information relative to a predetermined target at the task site. The operational range of this system is between 1-4 meters and is designed to supplement the longer-range sonar systems. Mounted under the forward nosecone, the sensor prototype consists of one mounted diode laser and one CCD camera hooked to a PC-104 CPU equipped with a frame grabber. Parallax effects result in the laser dot migrating in a known pattern, depending on the distance to the occluding object. The computer can easily calculate ranges based on the relative position of the dot in the frame.

Homing Sensor System

A CCD camera will be attached to the robotic arm above the gripper and seventh joint and will be used to acquire image data for the homing sensor. The homing sensor is tasked to locate and identify shapes. Specifically, it will locate a pre-determined circular bar-code target, which will be used as a reference point, and will determine the distance and angles between the robot arm camera and the

target pattern. The initial sensor is programmed to seek, identify and reference location to circular barcode targets, later development may extend the breadth of targets

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

Passive Arm

The deep-ocean passive arm is pictured in Figure AORD-1. This shows the deepwater version of the assembly. All of the aluminum parts are comprised of aluminum 6061 alloys. Two three-axis gimbal joints are in each of the two canisters. The whole assembly is around 15 lb in weight, dry and unfilled. White #9 mineral oil is the compensating fluid, which will circulate freely throughout the entire arm assembly. This is due to one cavity existing throughout the arm through the hollow arm tubes and gimbal sections, which are enclosed within neoprene bellows.

The wiring diagram for the arm is shown in Figure AORD-2. As seen from the figure the passive arm has eleven wires interconnecting it the rest of SAUVIM. As can be seen in the figure, these breakdown as follows: one signal ground line, one logic power line at 5V, seven pot data return lines, and two lines for the electromagnet power. A 50-ohm resistor is in the logic power line as a current limit should a pot reach a very low resistance. The seven 50-Kohm pots are an open-wiper design from JDK electronics company and these are all hooked in parallel between the logic rails.

How these wires are passed though the articulating joints is shown in photos of the base- and end-canister gimbals as well as elbow joints. Figures AORD-3, 4 and 5 are close-up photographs of the respective joints. From these, it can be seen that the wires are routed from slots cut in the aluminum tubing and are anchored with nylon electrical ties at both ends of the free run. Also, the free runs of wires have polyethylene spiral wrap coiled over the wires. This spiral wrap is to prevent the wires from migrating into pinch points in the joint and therefore getting severed as well as chafing on the slots cut through the arm tubing segments. The strain relief on the joints is also taken up by this armor rather than the underlying wires.

The original flexibility specified for the PA has been preserved in the pressure tolerant version. The range of swing of the base gimbals allows freedom of movement of the base leg of the PA in a cone that is ranged up to 60 degrees off the perpendicular line of the base canister. Redesign of the magnet canister allowed the same range of freedom for the lower leg of the PA. The hinge joint in its final configuration can range from a full extension (180 degrees) to a 60-degree included angle. The arm tubing sections along with the enclosed wire have been made easy to modify in length - this was done as much uncertainty about optimal lengths of the tubing segments is likely to remain until some combined arm experiments are performed.

The arm is hooked up to a 16-bit Data Translation A/D board in a 386 computer for accuracy testing as has a C-based program running. The code in its current form reports back the angles of the seven joints. Like the dry test arm, the forward kinematics solution is being sought. The same model with slight modifications should work as the two arms are topographically equivalent; the only thing that has changed is the distance of some of the segments interconnecting the joints. Figure AORD-6 is a display of the kinematics layout of the deep-ocean arm. Appendix AORD-1 is an overview of the symbolic solution matrices and the end results. The full symbolic solution is shown in Appendix AORD-2, which was obtained by use of the DERIVE software.

The next steps to in the PA development will involve sealing up the units, filling with compensating oil and performing full wet testing of the arm. This will complete the hardware development. Software development that remains is to port the dry arm code to the VME system, adjust some of the arm characteristics in the kinematics array, code optimization and to add a routine to allow for velocity tracking of the magnet head relative to the base. Repeatability and ranging error tests will complete development of this system.

Laser Ranging System

PLA system component is proceeding for the full ocean depth version. A 16-laser submergible array has been built to enable practical testing in both dry and wet setups. Proof-of-concept testing was first performed using a 4-laser array dry setup in a darkened room, Figure AORD-7. This model featured parallel-aligned lasers in a 2 by 2 array setup. The results from the 4-laser array testing are presented in Table AORD-1.

A problem encountered with the 16-laser prototype has been getting the lasers properly aligned which is absolutely necessary to be able to use the same mathematical model used in the 4-laser array setup. Allowing non-parallel lasers could resolve these manufacturing problems. Lasers tilted at specific angles can also work together and give a better resolution than the parallel could. In order to be able to handle non-parallel lasers a new math model was derived that can handle the two angles θ_1 and θ_2 , see Figure AORD-8. Proof-of-concept testing was performed using Matlab-based simulations and image processing simulations. Before using the new math model on the prototype it was first tested on a one- and two-laser set up. Before trying to measure the distance, z ., a calibration has to be made by measuring the two angles θ_1 and θ_2 for each laser, see Table AORD-2. The 16-laser array was then used at distances ranging between 1.85 m and 5.00 m, in a dry setup to try the new mathematical model on existing hardware. Four tests were made using four images, Tz1-Tz4. In each of these the real distance, z_r , was measured and then a Matlab simulation of the model gave the average distance, z_a , measured using the images. Testing results are presented in Table AORD-3.

Derivation of the mathematical model for parallel and skewed lasers was done by trigonometric as well by employment of similar triangle method where a pair of equivalent solutions was derived in both cases. Appendix AORD-3 is the derivation for skewed laser based on trigonometric methods.

The distance between the camera and the laser, known as "D", is important and a greater distance will give a better resolution. The prototype 16-laser array is 61 cm in extent on each side, this size will assure good resolution within the operational range. Resizing, however, will be necessary to enable positioning under the forward nosecone of the SAUVIM. By allowing tilted lasers the lasers does not have to be arranged in an array and a sufficient D can be achieved within the allowed size of the laser ranger.

Since the laser ranger has to be resized the next steep will be to evaluate the best hardware to use and how to set it up. Currently, the 16-laser prototype array employs black and white CCD camera, lights, and both red and green lasers. Evaluations on using a color CCD camera with a zoom and gated function will be carried out. The lasers will be tested regarding operational range under turbid water conditions. When the operational range has been determined the whole laser ranger can be tried underwater. The absorption and scattering processes will also be evaluated and tested which is of great importance when designing a laser ranger for an underwater environment [Caimi95].

The following software modifications are called for: reducing code latency by mapping lasers to slots on the frame grabber board, adding a moving Brownian-motion filter routine in anticipation of stirred sediment particles in the field of view. Lastly, integration of the completed system into the SAUVIM navigation CPU will have to be performed.

Redesigned Laser Ranging System

The laser ranging system has been redesigned as a single-laser sensor prototype scalable to a possible multiple-laser array in the future. Both the 532 nm solid-state diode laser and the monochrome CCD camera are currently housed in separate shallow-water pressure vessels. In its existing configuration, the system can range out to 4 meters in turbid ocean conditions, at a 10 Hz refresh rate, with less than 2 % absolute measured error. In a dry setup, range measurements in excess of 10 meters are attainable, which demonstrates the ability of the present method while concurrently exposing the limitations of the existing hardware. It is the absorptive and scattering effects of the ocean water that restrict the maximum range of the laser ranging system.

Triangulation computations, based on mathematical models of the previous-generation systems, were replaced by a simple system calibration technique that must be performed prior to mission deployment. Essentially, the position of the detected laser dot in the camera's field of view is a function of the vehicle's range from the target. Upon calibration, a number of sample data points are collected, and a spline curve is derived that accurately approximates the range function. The resulting coefficients are then stored. At the task site, the calibration coefficients are recalled, and efficient range measurements are derived from the camera data by means of the interpolation of a simple polynomial function.

The current laser dot detection algorithm is written for maximum speed considerations as well as for functionality in turbid underwater conditions, such as the sandy ocean floor stirred up by the thrusters of the SAUVIM vehicle. For overall system speed, the usage of image processing techniques from the previous-generation system was minimized in favor of hardware filtering to improve the camera signal. A narrowband interference filter is currently implemented as a replacement of a prior software-filtering scheme.

Increased maximum range should be attainable with the addition of a different camera lens, as well as an upgrade to a more powerful laser. Expansion of the system to a laser array is a consideration that would provide orientation information, although at the cost of lower measurement refresh rates.

Homing Sensor System

Homing sensor prototype hardware fabrication is complete. Figure AORD-9 displays the components that comprise the MHS. In the center of the photograph is the PC-104 computer that is equipped with a video frame grabber, behind it and to the left is the pressure tolerant housing and bracket that clamp onto the Ansaldo arm above the wrist joint. To the right of the computer is the camera and white LED that will be used for task illumination while behind are some printed copies of the task site targets the MHS will home on. Close-up photographs of the camera and pressure housings are given in Figures AORD-10 and 11, respectively. It will be noted that the pressure housing is on a bracket that allows azimuth and elevation adjustments, this will assist in checking for optimal orientation settings as the MHS gets integrated with the Ansaldo arm controller and computer. The acrylic ring and lens mounted to the camera are to allow illumination from white LED diodes located behind the camera to

diffuse and light the immediate vicinity. The lens is to narrow down the aperture of the camera as is it was found during code development that resolution was lost do to a wide camera image area.

Alpha code has been developed and loaded onto the sensor system that gives range to the targets as well as the angle off of the line centered and perpendicular to the target pattern. This code is developed in C and is resident on the PC-104 computer to avoid loading of the SAUVIM VME computers.

The major tasks that remain for these systems are to test and optimize the alpha code as well integrate the system onto the Ansaldo arm system. Testing of the homing sensor in the dry lab setup will proceed before integration onto the SAUVIM vehicle.

Future Tasks (Phase II Tasks)

- Wet test; load forward kinematics model into C-code; and integrate to VME Arm Control CPU I/O board. Test to quantify system accuracy and repeatability.
- Resizing the pressure housings for the diode lasers, the camera housing and building a new frame to hold the lasers and the camera.
- Finishing modifications of software, modify from dry-test setup 4-array to operational multi laser setup.
- Install homing sensor camera and vessel onto the Ansaldo arm, add circular barcode detection and tracking logic to skeletal image frame grabbing code. Integrate system into the DTDS setup. Test to quantify system accuracy and repeatability.



Figure AORD-1: Passive Arm (PA)

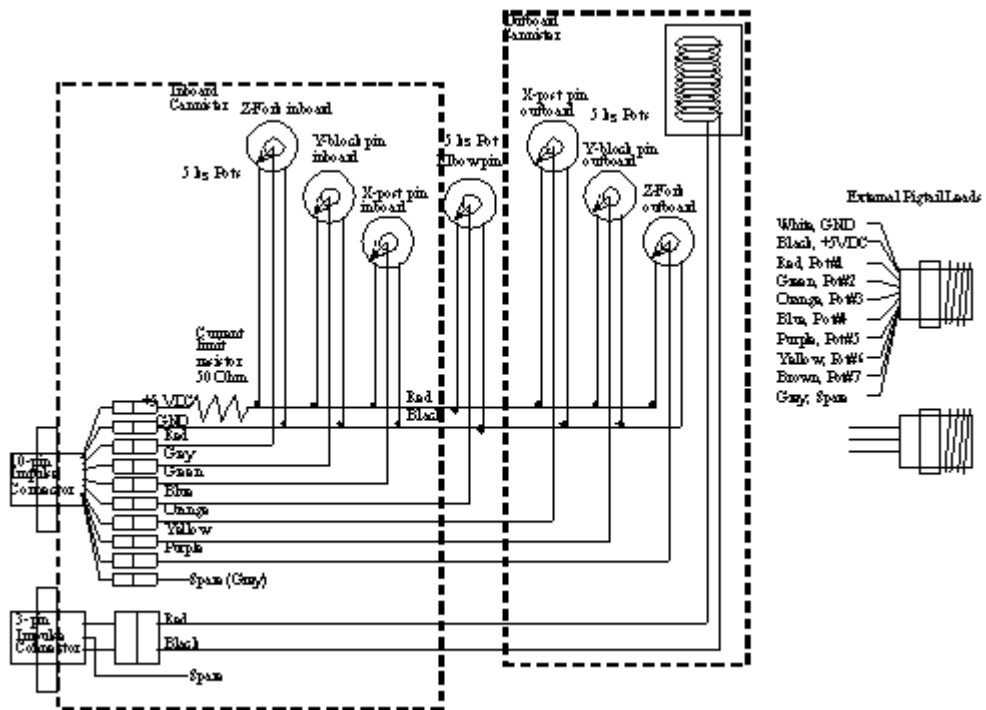


Figure AORD-2: PA wiring diagram

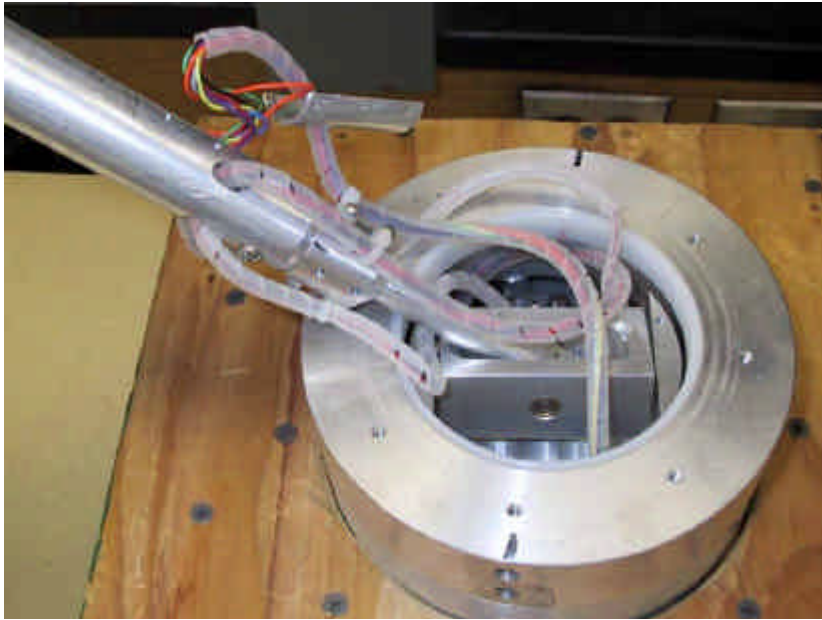


Figure AORD-3: PA close-up base gimbal joint

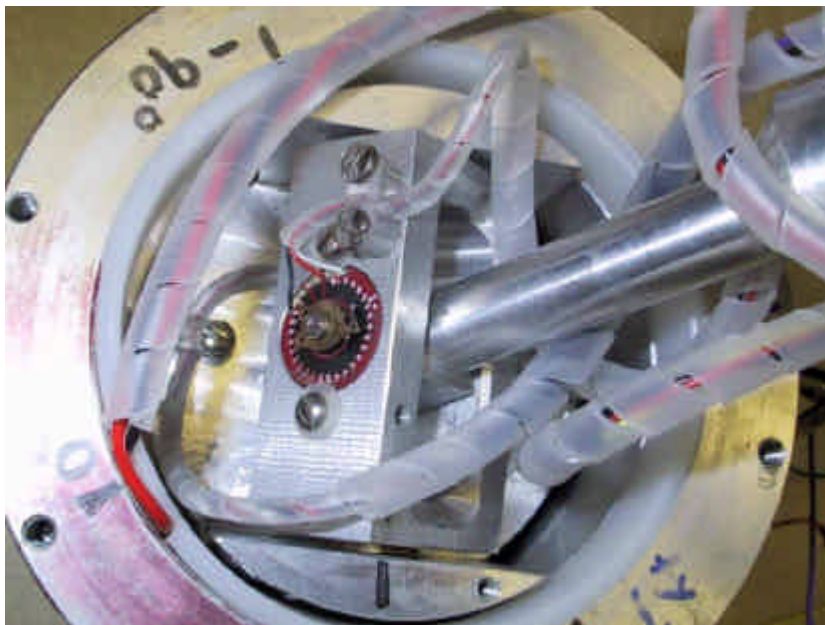


Figure AORD-4: PA close-up end gimbal joint

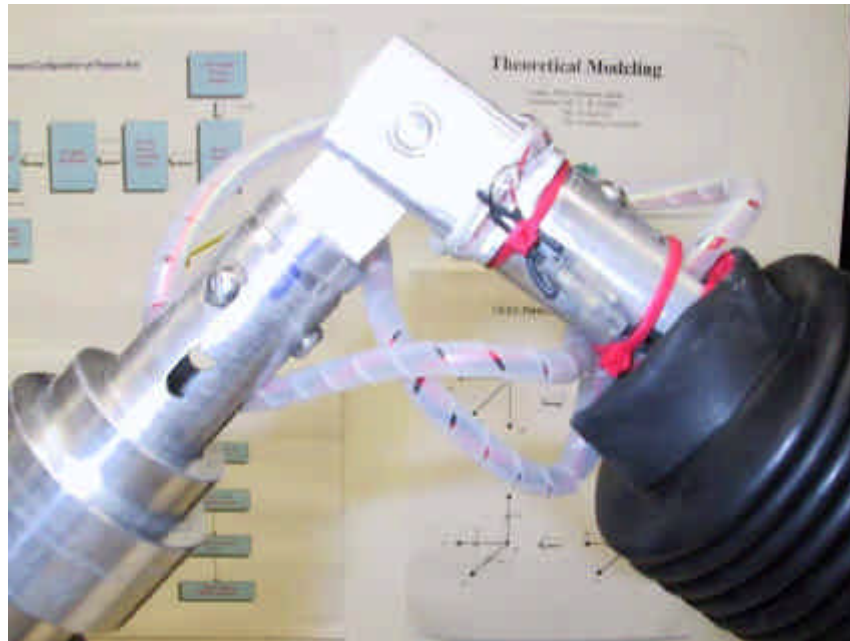


Figure AORD-5: PA close-up elbow joint

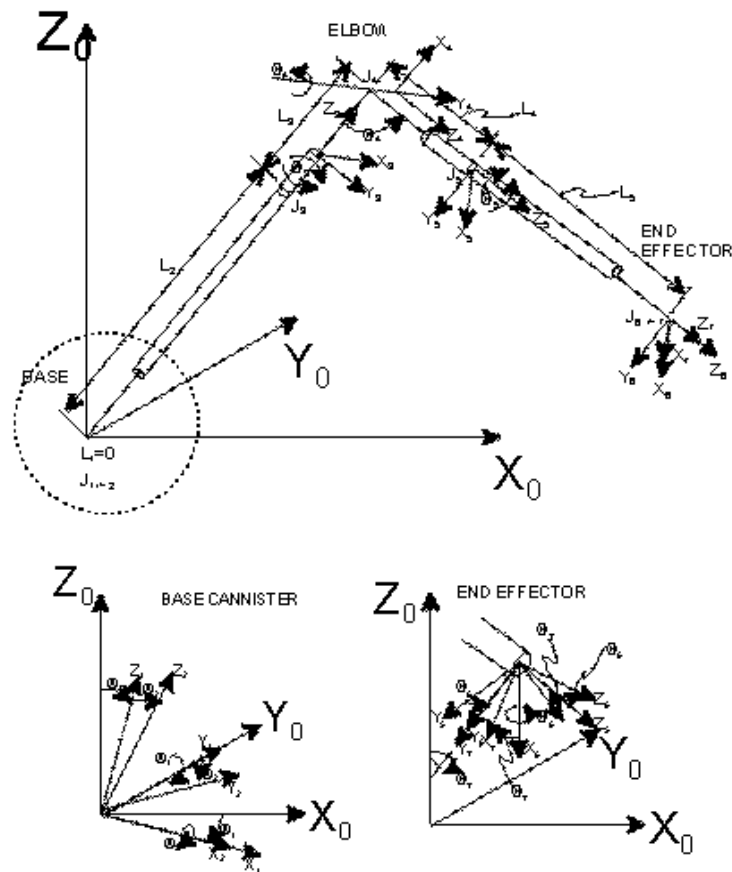


Figure AORD-6: Arm kinematics

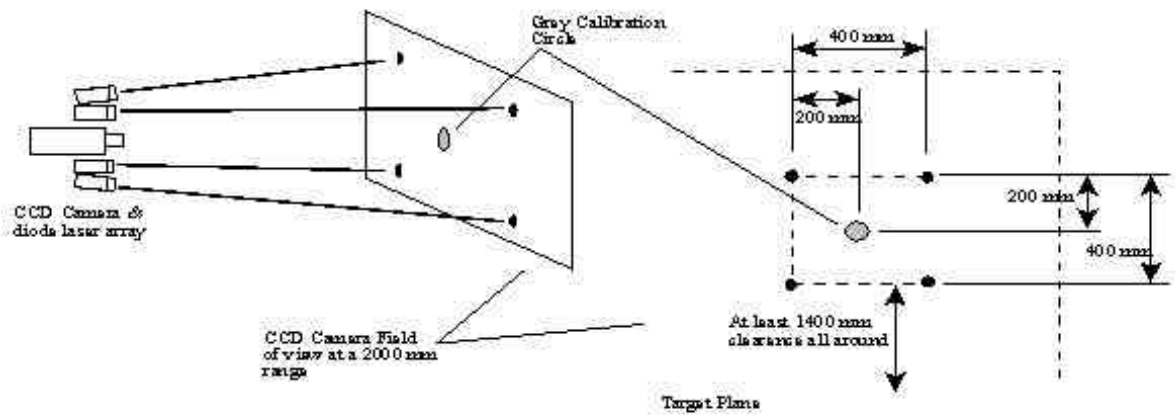


Figure AORD-7: The four-laser array set-up.

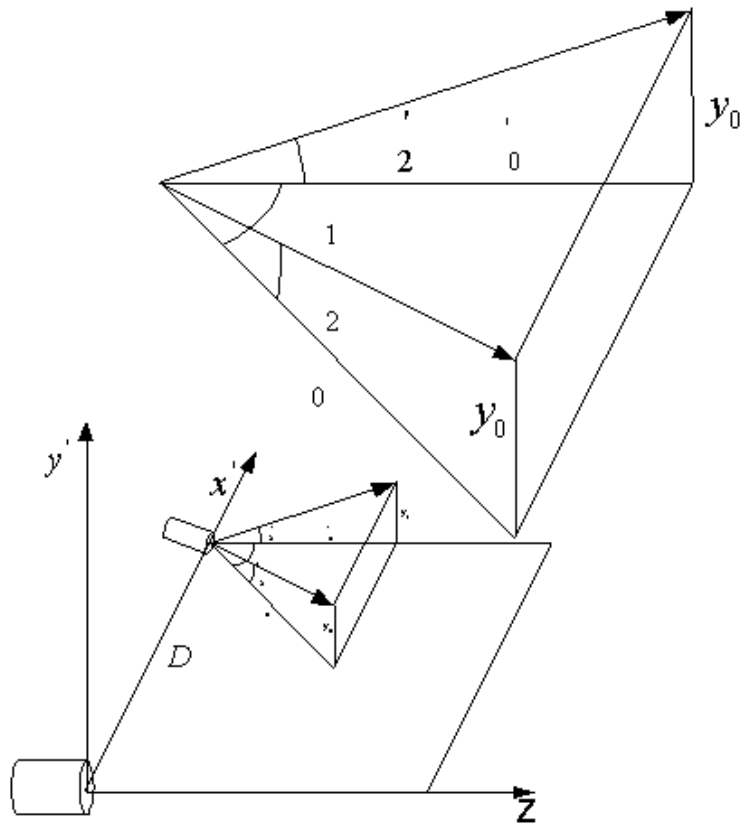


Figure AORD-8: Geometry of non-parallel laser

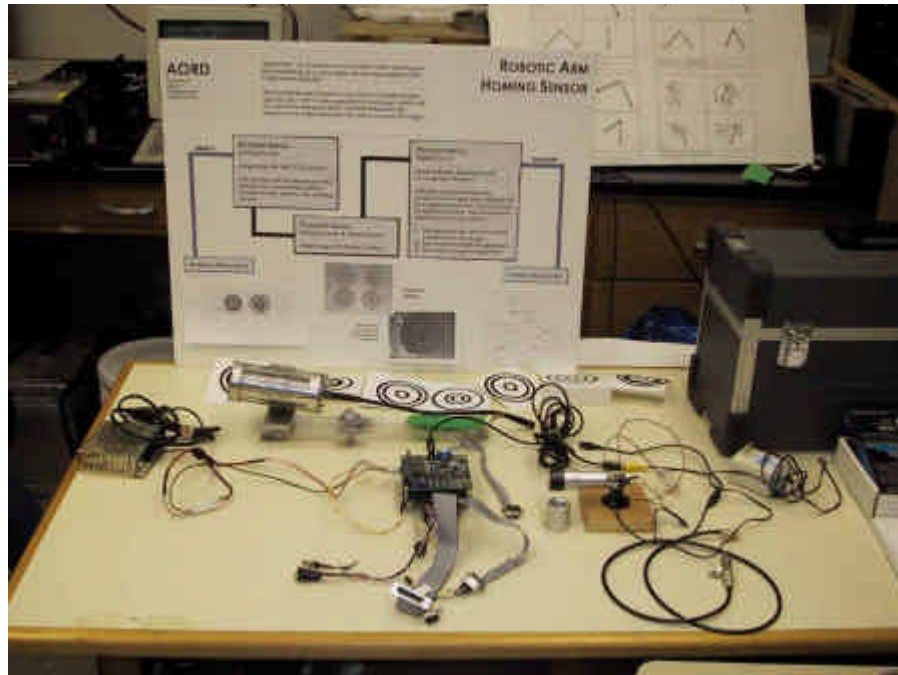


Figure AORD-9: Entire MHS System.



Figure AORD-10: MHS camera.



Figure AORD-11: MHS pressure vessel.

Table AORD-1: Results from four-laser array set-up

Range to Surface (cm)	Distance of Beam Separation (cm)	Angular Size of Image Field (degrees)	Angular Separation of Dots (True) (degrees)	Number of Pixels separating dots (pixels)	Range Precision across 1 pixel (cm)
10.0	4.0	30.0	22.62	386.0	0.03
25.0	4.0	30.0	9.15	156.1	0.16
50.0	4.0	30.0	4.58	78.2	0.64
100.0	4.0	30.0	2.29	39.1	2.56
150.0	4.0	30.0	1.53	26.1	5.75
200.0	4.0	30.0	1.15	19.6	10.23
250.0	4.0	30.0	0.92	15.6	15.98
300.0	4.0	30.0	0.76	13.0	23.01
400.0	4.0	30.0	0.57	9.8	40.91

Table AORD-2: Calibration of the 16-laser array

Image	z_a (m)	z_r (m)
Tz1.bmp	3.4969	3.50
Tz2.bmp	4.5151	4.51
Tz3.bmp	2.1677	2.17
Tz4.bmp	1.8538	1.85

Table AORD-3: Results from four images taken at different distances

Laser no.	θ_1 (average)	θ_2 (average)
1	-0.254	-0.590
	-0.523	-0.976
	-1.080	-0.675
	-1.083	-0.053
	0.309	-0.852
	-0.560	-0.680
	-1.195	0.074
	-1.174	0.491
	0.212	0.344
	0.333	0.241
	-0.340	0.786
	-1.252	0.176
	0.465	0.245
	0.182	0.488
	0.498	1.216
16	-0.519	0.799

Appendix AORD-1: Passive Arm Forward Kinematics Matrix

i	\forall_{i-1}	a_{i-1}	di	2i	transforms-form of 2
1	0	0	0	2 ₁	rotation about Y ₀
2	0	0	l ₂	2 ₂	rotation about X ₁ , translation along Z ₂
3	0	0	l ₃	2 ₃	rotation about Z ₂ , translation along Z ₃
4	0	0	l ₄	2 ₄	rotation about Y ₃ , translation along Z ₄
5	0	0	l ₅	2 ₅	rotation about Z ₄ , translation along Z ₅
6	0	0	l ₆	2 ₆	rotation about X ₅
7	0	0	l ₂	2 ₇	rotation about Y ₆

Transformation Matrices:

$$\begin{aligned}
 {}^0_1T &= \begin{bmatrix} cq_1 & 0 & -sq_1 & 0 \\ 0 & 1 & 0 & 0 \\ sq_1 & 0 & cq_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^1_2T &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_2 & -sq_2 & 0 \\ 0 & sq_2 & cq_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^2_3T &= \begin{bmatrix} cq_3 & -sq_3 & 0 & 0 \\ sq_3 & cq_3 & 0 & 0 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^3_4T &= \begin{bmatrix} cq_4 & 0 & -sq_4 & 0 \\ 0 & 1 & 0 & 0 \\ sq_4 & 0 & cq_4 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^4_5T &= \begin{bmatrix} cq_5 & -sq_5 & 0 & 0 \\ sq_5 & cq_5 & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^5_6T &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cq_6 & -sq_6 & 0 \\ 0 & sq_6 & cq_6 & l_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^6_7T &= \begin{bmatrix} cq_6 & 0 & -sq_6 & 0 \\ 0 & 1 & 0 & 0 \\ sq_6 & 0 & cq_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^0_7T &= {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T {}^6_7T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Note: Projection of \mathbf{k} along z_7 is normal to the magnet face..

Position of end gimbals wrist:

Note "c" is shorthand for "cos" and "s" is shorthand for "sin")

As to gimbals wrist position in the world coordinates the three vectors are:

$$p_x = \{l_4 + l_5\} \left[-(c_1c_3 - s_1s_2s_3)s_4 - s_1c_2c_4 \right] - s_1c_2(l_3 + l_2)$$

$$p_y = l_5 [c_2s_3s_4 - s_2c_4] + l_4 [-c_2s_3s_4 - s_2c_4] - s_2(l_3 + l_4)$$

$$p_z = \{l_4 + l_5\} \left[-(s_1c_3 + c_1s_2s_3)s_4 + c_1c_2c_4 \right] + c_1c_2[l_2 + l_3]$$

Appendix AORD-2: Passive Arm DERIVE symbolic solution

$[T](1 \rightarrow 0) = [["c1", -"s1", 0, 0], ["s1", "c1", 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]$
 $[T](2 \rightarrow 1) = [[1, 0, 0, 0], [0, "c2", -"s2", 0], [0, "s2", "c2", "l1"], [0, 0, 0, 1]]$
 $[T](3 \rightarrow 2) = [["c3", 0, -"s3", 0], [0, 1, 0, 0], ["s3", 0, "c3", 0], [0, 0, 0, 1]]$
 $[T](4 \rightarrow 3) = [["c4", 0, -"s4", 0], [0, 1, 0, 0], ["s4", 0, "c4", "l2"], [0, 0, 0, 1]]$
 $[T](5 \rightarrow 4) = [["c5", 0, -"s5", 0], [0, 1, 0, 0], ["s5", 0, "c5", "l3"], [0, 0, 0, 1]]$
 $[T](6 \rightarrow 5) = [[1, 0, 0, 0], [0, "c6", -"s6", 0], [0, "s6", "c6", 0], [0, 0, 0, 1]]$
 $[T](7 \rightarrow 6) = [["c7", -"s7", 0, 0], ["s7", "c7", 0, 0], [0, 0, 1, "l4"], [0, 0, 0, 1]]$
 $[T](1 \rightarrow 0) * [T](2 \rightarrow 1) * [T](3 \rightarrow 2) * [T](4 \rightarrow 3) =$
 $[["c1", -"s1", 0, 0], ["s1", "c1", 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]*$
 $[[1, 0, 0, 0], [0, "c2", -"s2", 0], [0, "s2", "c2", "l1"], [0, 0, 0, 1]]*$
 $[["c3", 0, -"s3", 0], [0, 1, 0, 0], ["s3", 0, "c3", 0], [0, 0, 0, 1]]*$
 $[["c4", 0, -"s4", 0], [0, 1, 0, 0], ["s4", 0, "c4", "l2"], [0, 0, 0, 1]]$

$[T](4 \rightarrow 0) =$
 $[["c1"*"c3"*"c4"- "c1"*"s3"*"s4"+ "c3"*"s1"*"s2"*"s4"+ "c4"*"s1"*"s2"*"s3",$
 $- "c2"*"s1",$
 $- "c1"*"c3"*"s4"- "c1"*"c4"*"s3"+ "c3"*"c4"*"s1"*"s2"- "s1"*"s2"*"s3"*"s4",$
 $"c3"*"l2"*"s1"*"s2"- "c1"*"l2"*"s3"],$
 $[- "c1"*"c3"*"s2"*"s4"- "c1"*"c4"*"s2"*"s3"+ "c3"*"c4"*"s1"- "s1"*"s3"*"s4",$
 $"c1"*"c2",$
 $- "c1"*"c3"*"c4"*"s2"+ "c1"*"s2"*"s3"*"s4"- "c3"*"s1"*"s4"- "c4"*"s1"*"s3",$
 $- "c1"*"c3"*"l2"*"s2"- "l2"*"s1"*"s3"], ["c2"*"c3"*"s4"+ "c2"*"c4"*"s3",$
 $"s2", "c2"*"c3"*"c4"- "c2"*"s3"*"s4", "c2"*"c3"*"l2"+ "l1"],$
 $[0, 0, 0, 1]]$

$[T](1 \rightarrow 0) * [T](2 \rightarrow 1) * [T](3 \rightarrow 2) * [T](4 \rightarrow 3) * [T](5 \rightarrow 4) * [T](6 \rightarrow 5) * [T](7 \rightarrow 6) =$
 $[["c1", -"s1", 0, 0], ["s1", "c1", 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]*$
 $[[1, 0, 0, 0], [0, "c2", -"s2", 0], [0, "s2", "c2", "l1"], [0, 0, 0, 1]]*$
 $[["c3", 0, -"s3", 0], [0, 1, 0, 0], ["s3", 0, "c3", 0], [0, 0, 0, 1]]*$
 $[["c4", 0, -"s4", 0], [0, 1, 0, 0], ["s4", 0, "c4", "l2"], [0, 0, 0, 1]]*$
 $[["c5", 0, -"s5", 0], [0, 1, 0, 0], ["s5", 0, "c5", "l3"], [0, 0, 0, 1]]*$
 $[[1, 0, 0, 0], [0, "c6", -"s6", 0], [0, "s6", "c6", 0], [0, 0, 0, 1]]*$
 $[["c7", -"s7", 0, 0], ["s7", "c7", 0, 0], [0, 0, 1, "l4"], [0, 0, 0, 1]]$

$T(7 \rightarrow 0) =$
 $[["c1"*"c3"*"c4"*"c5"*"c7"- "c1"*"c3"*"c4"*"s5"*"s6"*"s7"-$
 $"c1"*"c3"*"c5"*"s4"*"s6"*"s7"- "c1"*"c3"*"c7"*"s4"*"s5"-$
 $"c1"*"c4"*"c5"*"s3"*"s6"*"s7"- "c1"*"c4"*"c7"*"s3"*"s5"-$
 $"c1"*"c5"*"c7"*"s3"*"s4"+ "c1"*"s3"*"s4"*"s5"*"s6"*"s7"-$
 $"c2"*"c6"*"s1"*"s7"+ "c3"*"c4"*"c5"*"s1"*"s2"*"s6"*"s7"+$
 $"c3"*"c4"*"c7"*"s1"*"s2"*"s5"+ "c3"*"c5"*"c7"*"s1"*"s2"*"s4"-$
 $"c3"*"s1"*"s2"*"s4"*"s5"*"s6"*"s7"+ "c4"*"c5"*"c7"*"s1"*"s2"*"s3"-$
 $"c4"*"s1"*"s2"*"s3"*"s5"*"s6"*"s7"- "c5"*"s1"*"s2"*"s3"*"s4"*"s6"*"s7"-$
 $"c7"*"s1"*"s2"*"s3"*"s4"*"s5",$
 $- "c1"*"c3"*"c4"*"c5"*"s7"- "c1"*"c3"*"c4"*"c7"*"s5"*"s6"-$
 $"c1"*"c3"*"c5"*"c7"*"s4"*"s6"+ "c1"*"c3"*"s4"*"s5"*"s7"-$
 $"c1"*"c4"*"c5"*"c7"*"s3"*"s6"+ "c1"*"c4"*"s3"*"s5"*"s7"+$
 $"c1"*"c5"*"s3"*"s4"*"s7"+ "c1"*"c7"*"s3"*"s4"*"s5"*"s6"-$
 $"c2"*"c6"*"c7"*"s1"+ "c3"*"c4"*"c5"*"c7"*"s1"*"s2"*"s6"-$
 $"c3"*"c4"*"s1"*"s2"*"s5"*"s7"- "c3"*"c5"*"s1"*"s2"*"s4"*"s7"-$
 $"c3"*"c7"*"s1"*"s2"*"s4"*"s5"*"s6"- "c4"*"c5"*"s1"*"s2"*"s3"*"s7"-$
 $"c4"*"c7"*"s1"*"s2"*"s3"*"s5"*"s6"- "c5"*"c7"*"s1"*"s2"*"s3"*"s4"*"s6"+$
 $"s1"*"s2"*"s3"*"s4"*"s5"*"s7",$
 $- "c1"*"c3"*"c4"*"c6"*"s5"- "c1"*"c3"*"c5"*"c6"*"s4"-$
 $"c1"*"c4"*"c5"*"c6"*"s3"+ "c1"*"c6"*"s3"*"s4"*"s5"+$
 $"c2"*"s1"*"s6"+ "c3"*"c4"*"c5"*"c6"*"s1"*"s2"-$
 $"c3"*"c6"*"s1"*"s2"*"s4"*"s5"- "c4"*"c6"*"s1"*"s2"*"s3"*"s5"-$

"c5"*"c6"*"s1"*"s2"*"s3"*"s4",- "c1"*"c3"*"c4"*"c6"*"14"*"s5"-
 "c1"*"c3"*"c5"*"c6"*"14"*"s4"- "c1"*"c3"*"13"*"s4"-
 "c1"*"c4"*"c5"*"c6"*"14"*"s3"- "c1"*"c4"*"13"*"s3"+
 "c1"*"c6"*"14"*"s3"*"s4"*"s5"- "c1"*"12"*"s3"+
 "c2"*"14"*"s1"*"s6"+ "c3"*"c4"*"c5"*"c6"*"14"*"s1"*"s2"+
 "c3"*"c4"*"13"*"s1"*"s2"- "c3"*"c6"*"14"*"s1"*"s2"*"s4"*"s5"+
 "c3"*"12"*"s1"*"s2"- "c4"*"c6"*"14"*"s1"*"s2"*"s3"*"s5"-
 "c5"*"c6"*"14"*"s1"*"s2"*"s3"*"s4"- "13"*"s1"*"s2"*"s3"*"s4"],
 ["c1"*"c2"*"c6"*"s7"- "c1"*"c3"*"c4"*"c5"*"s2"*"s6"*"s7"-
 "c1"*"c3"*"c4"*"c7"*"s2"*"s5"- "c1"*"c3"*"c5"*"c7"*"s2"*"s4"+
 "c1"*"c3"*"s2"*"s4"*"s5"*"s6"*"s7"- "c1"*"c4"*"c5"*"c7"*"s2"*"s3"+
 "c1"*"c4"*"s2"*"s3"*"s5"*"s6"*"s7"+ "c1"*"c5"*"s2"*"s3"*"s4"*"s6"*"s7"+
 "c1"*"c7"*"s2"*"s3"*"s4"*"s5"+ "c3"*"c4"*"c5"*"c7"*"s1"-
 "c3"*"c4"*"s1"*"s5"*"s6"*"s7"- "c3"*"c5"*"s1"*"s4"*"s6"*"s7"-
 "c3"*"c7"*"s1"*"s4"*"s5"- "c4"*"c5"*"s1"*"s3"*"s6"*"s7"-
 "c4"*"c7"*"s1"*"s3"*"s5"- "c5"*"c7"*"s1"*"s3"*"s4"+
 "s1"*"s3"*"s4"*"s5"*"s6"*"s7",
 "c1"*"c2"*"c6"*"c7"- "c1"*"c3"*"c4"*"c5"*"c7"*"s2"*"s6"+
 "c1"*"c3"*"c4"*"s2"*"s5"*"s7"+ "c1"*"c3"*"c5"*"s2"*"s4"*"s7"+
 "c1"*"c3"*"c7"*"s2"*"s4"*"s5"*"s6"+ "c1"*"c4"*"c5"*"s2"*"s3"*"s7"+
 "c1"*"c4"*"c7"*"s2"*"s3"*"s5"*"s6"+ "c1"*"c5"*"c7"*"s2"*"s3"*"s4"*"s6"-
 "c1"*"s2"*"s3"*"s4"*"s5"*"s7"- "c3"*"c4"*"c5"*"s1"*"s7"-
 "c3"*"c4"*"c7"*"s1"*"s5"*"s6"- "c3"*"c5"*"c7"*"s1"*"s4"*"s6"+
 "c3"*"s1"*"s4"*"s5"*"s7"- "c4"*"c5"*"c7"*"s1"*"s3"*"s6"+
 "c4"*"s1"*"s3"*"s5"*"s7"+ "c5"*"s1"*"s3"*"s4"*"s7"+
 "c7"*"s1"*"s3"*"s4"*"s5"*"s6",
 - "c1"*"c2"*"s6"- "c1"*"c3"*"c4"*"c5"*"c6"*"s2"+
 "c1"*"c3"*"c6"*"s2"*"s4"*"s5"+ "c1"*"c4"*"c6"*"s2"*"s3"*"s5"+
 "c1"*"c5"*"c6"*"s2"*"s3"*"s4"- "c3"*"c4"*"c6"*"s1"*"s5"-
 "c3"*"c5"*"c6"*"s1"*"s4"- "c4"*"c5"*"c6"*"s1"*"s3"+ "c6"*"s1"*"s3"*"s4"*"s5",
 - "c1"*"c2"*"14"*"s6"- "c1"*"c3"*"c4"*"c5"*"c6"*"14"*"s2"-
 "c1"*"c3"*"c4"*"13"*"s2"+ "c1"*"c3"*"c6"*"14"*"s2"*"s4"*"s5"-
 "c1"*"c3"*"12"*"s2"+ "c1"*"c4"*"c6"*"14"*"s2"*"s3"*"s5"+
 "c1"*"c5"*"c6"*"14"*"s2"*"s3"*"s4"+ "c1"*"13"*"s2"*"s3"*"s4"-
 "c3"*"c4"*"c6"*"14"*"s1"*"s5"- "c3"*"c5"*"c6"*"14"*"s1"*"s4"-
 "c3"*"13"*"s1"*"s4"- "c4"*"c5"*"c6"*"14"*"s1"*"s3"- "c4"*"13"*"s1"*"s3"+
 "c6"*"14"*"s1"*"s3"*"s4"*"s5"- "12"*"s1"*"s3"],
 ["c2"*"c3"*"c4"*"c5"*"s6"*"s7"+ "c2"*"c3"*"c4"*"c7"*"s5"+
 "c2"*"c3"*"c5"*"c7"*"s4"- "c2"*"c3"*"s4"*"s5"*"s6"*"s7"+
 "c2"*"c4"*"c5"*"c7"*"s3"- "c2"*"c4"*"s3"*"s5"*"s6"*"s7"-
 "c2"*"c5"*"s3"*"s4"*"s6"*"s7"- "c2"*"c7"*"s3"*"s4"*"s5"+ "c6"*"s2"*"s7",
 "c2"*"c3"*"c4"*"c5"*"c7"*"s6"- "c2"*"c3"*"c4"*"s5"*"s7"-
 "c2"*"c3"*"c5"*"s4"*"s7"- "c2"*"c4"*"c7"*"s3"*"s5"*"s6"-
 "c2"*"c5"*"c7"*"s3"*"s4"*"s6"+ "c2"*"s3"*"s4"*"s5"*"s7"+ "c6"*"c7"*"s2",
 "c2"*"c3"*"c4"*"c5"*"c6"- "c2"*"c3"*"c6"*"s4"*"s5"-
 "c2"*"c4"*"c6"*"s3"*"s5"- "c2"*"c5"*"c6"*"s3"*"s4"- "s2"*"s6",
 "c2"*"c3"*"c4"*"c5"*"c6"*"14"+ "c2"*"c3"*"c4"*"13"-
 "c2"*"c3"*"c6"*"14"*"s4"*"s5"+ "c2"*"c3"*"12"-
 "c2"*"c4"*"c6"*"14"*"s3"*"s5"- "c2"*"c5"*"c6"*"14"*"s3"*"s4"-
 "c2"*"13"*"s3"*"s4"+ "11"- "14"*"s2"*"s6"],
 [0,0,0,1]]

Appendix AORD-3: Derivation of the math model for skewed lasers

$$\begin{aligned} x' &= x_0 - \tan q_1 \cdot z, \{x_0 = D\} \Rightarrow \\ x' &= D - \tan q_1 \cdot z \end{aligned} \quad (1)$$

$$\begin{aligned} y' &= y_0 + \tan q_2 \cdot z = y_0 + \frac{\tan q_2}{\cos q_1} z, \{y_0 = 0\} \Rightarrow \\ y' &= \frac{\tan q_2}{\cos q_1} z \end{aligned} \quad (2)$$

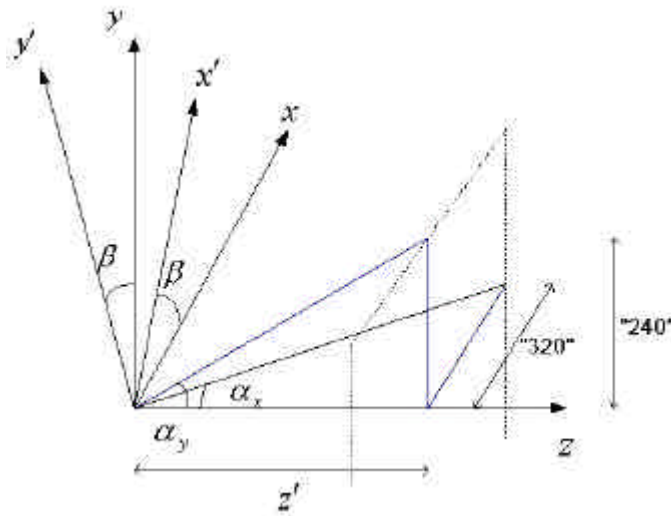


Figure A

Note that distance “320” and “240” in figure is only valid if $z = z'$

Conversion from coordinate system x', y' to x, y , se figure AORD-3

$$x = \cos b \cdot x' - \sin b \cdot y' \quad (3)$$

$$y = \sin b \cdot x' + \cos b \cdot y' \quad (4)$$

Eq. (1) in (3) and eq. (2) in(4)

$$x = \cos b(D - \tan q_1 \cdot z) - \sin b \left(\frac{\tan q_2}{\cos q_1} \right) z \quad (5)$$

$$y = \sin b(D - \tan q_1 \cdot z) + \cos b \left(\frac{\tan q_2}{\cos q_1} \right) z \quad (6)$$

X' and Y' is the actual pixel value from the frame grabber. Get the origin in the middle and the y-axis in the right direction.

$$X = X' - 320 \quad (7)$$

$$Y = -Y' + 240 \quad (8)$$

Since the resolution is 640 x 480

To convert the pixel coordinates to the real coordinate system a constant $\tan \mathbf{a}$ is used, $\tan \mathbf{a}$ is different in x and y directions and is camera specific and have to be measured. In case of a zoom camera, $\tan \mathbf{a} = f(\text{zoom} - \text{value})$. Skewing of the camera has not been under consideration in this model.

$$x = X \cdot \tan \mathbf{a}_x \cdot z \quad (9)$$

$$y = Y \cdot \tan \mathbf{a}_y \cdot z \quad (10)$$

Use eq. (5) and eq.(6) to get \mathbf{q}_1 and use \mathbf{q}_1 to calculate \mathbf{q}_2 . These two equations is used to calibrate the lasers by measure X, Y and z.

$$\mathbf{q}_1 = \tan^{-1} \left(\frac{\cos^2 \mathbf{b} \cdot D - x \cdot \cos \mathbf{b} + \sin^2 \mathbf{b} \cdot D - y \cdot \sin \mathbf{b}}{z} \right) \quad (11)$$

$$\mathbf{q}_2 = \tan^{-1} \left(\frac{\cos \mathbf{q}_1 \cdot D(y + \sin \mathbf{b} \cdot \tan \mathbf{q}_1 \cdot z - \sin \mathbf{b} \cdot D)}{\cos \mathbf{b} \cdot z} \right) \quad (12)$$

We now can solve out z_x and z_y since the only variable that is unknown is X and Y .

$$z_x = \frac{\cos \mathbf{q}_1 \cdot \cos \mathbf{b} \cdot D}{X \cdot \tan \mathbf{a}_x \cdot \cos \mathbf{q}_1 + \cos \mathbf{b} \cdot \tan \mathbf{q}_1 \cdot \cos \mathbf{q}_1 + \sin \mathbf{b} \cdot \tan \mathbf{q}_2} \quad (13)$$

$$z_y = \frac{\cos \mathbf{q}_1 \cdot \sin \mathbf{b} \cdot D}{Y \cdot \tan \mathbf{a}_y \cdot \cos \mathbf{q}_1 + \sin \mathbf{b} \cdot \tan \mathbf{q}_1 \cdot \cos \mathbf{q}_1 - \cos \mathbf{b} \cdot \tan \mathbf{q}_2} \quad (14)$$

The resolution for z_x and z_y depends on \mathbf{b} therefore they have to be valued differently. Since the resolution for z_x is best (z_y worst) when $\mathbf{b} = 0$ or $2\mathbf{p}$, z_y is best (z_x worst) when $\mathbf{b} = \pm \frac{\mathbf{p}}{2}$ and $\cos^2 \mathbf{b} + \sin^2 \mathbf{b} = 1$.

$$z = z_x \cdot \cos^2 \mathbf{b} + z_y \cdot \sin^2 \mathbf{b} \quad (15)$$

Intelligent, Coordinated-Motion/Force Control (ICM/FC)

Project Leader(s): Dr. Giacomo Maranni, Dr. Tae Won Kim, Dr. Hyun Taek Choi, Mr. Michael West & Dr. Junku Yuh

Past Project Leader(s): Dr. Song K. Choi, Dr. Kazuo Sugihara & Dr. Nilanjan Sarkar

The main technical development of the ICM/FC group is described in the following sections: Manipulator Control and Test Platform, Low-Level Control, Localization and Navigation, and High-Level Control. The Manipulator Control and Test Platform is the combined sections of the previous Theoretical Modeling and Dry Test Design Set-Up. The Localization and Navigation is a separation from the Low-Level Control due to the vastness and complexity of the research area.

Manipulator Control and Test Platform (MCTP)

Project Leader(s):	Dr. Giacomo Marani
Personnel:	Mr. Jin Hyun Kim, Mr. Jong Ho Eun, Ms. Stacy L. Dees & Mr. Jangwon Lee
Past Project Leader(s):	Dr. Song K. Choi, Dr. Tae Won Kim, Dr. Junku Yuh & Dr. Nilanjan Sarkar
Past Personnel:	Mr. Tommaso Bozzo, Mr. Gang Cheng, Ms. Jing Nie, Mr. Mike Kobayakawa, Mr. Mark Fujita, Dr. Gyoung H. Kim & Mr. Tarun Podder

Objectives

The primary purpose of SAUVIM is to perform underwater intervention mission with a limited human presence. The problem in designing and implementing a control system for its manipulator is to ensure a reliable motion everywhere in the workspace, avoiding collision, system instabilities and/or unwanted motions. Actually the human intervention is limited or sometimes absent: in that case we must ensure the completion of the required task when it is, almost theoretically, feasible. This is a high priority requirement especially when approaching to high depth, given the high cost requirement for such intervention operations.

The project and realization of Maris 7080 underwater manipulator, an affiliate package of SAUVIM project, has been developed with a particular care of the above objectives, including a well-structured software developing style in order to simplify as much as possible further expansions of the overall system.

The prototyping version of the control software of the previous phase has been migrated to a more flexible and stable release. The last includes a better simulation server capable of a more accurate simulation of the overall arm system than the previous version and an advanced socket-based communication system with an integrated command interpreter.

In addition, our tasks included a preliminary development of the passive sensor system, in order to acquire information of the target position.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

1. Kinematics, inverse kinematics & redundancy resolution:
 - Resolved Motion Rate Controller
 - Position prior orientation
 - Kinematic singularities avoidance
 - Manipulability optimization
 - Algorithmic singularities avoidance (with position prior orientation)
 - Stability optimization in proximity of singular points
 - Comparative experiments between controllers
 - Joint limits handling
 - Collision avoidance (simulation only)
2. Code Optimization and simulation speed-up:
 - New numerical algorithm for pseudo-inversion

- Optimization of procedures for deriving kinematical quantities (All-in-one procedure)
 - New communication socket protocol with automatic disconnection/reconnection feature
 - New non-prototyping task scheduler for VxWorks OS, based on the Real Time Workshop's core
 - Optimization of board drivers. The new non-prototyping version is suitable of use under xBus (Maris Command Interface) and is written in stand-alone Ansi-C language.
3. Servo-actuator system:
- Joints Saturation Guard (to ensure a correct tracking)
 - Elmo controllers integration constant identified (to ensure a constant voltage/velocity ratio for each motor)
 - Motor failure detector
4. Data exchange Bus and Command Interface (xBus):
- Basic protocol and software structure definition (investigation of Finite State Machine software tools)
 - TCP-IP based client-server communication system, with multi-point connection feature
 - Integrated command interpreter suitable of running high-level scripts through the client interface
 - Trajectory planner
5. Graphic interface and Multimedia Development Environment (MarisGL):
- Internal link with Tornado for fast development
 - Vehicle frame integration
 - Motor control dialogs
 - Enhancement of some features (DirectX support for mouse and keyboard, for a completely wireless driving solution)
 - Added support for commands interface
 - Added support for switching between the simulation server and the actual system, compatible with the future arm programming language.
 - Added motors failure detector and real-time link with motors reflecting the actual status of enable flags and sensor selector.

Kinematics, inverse kinematics & redundancy resolution.

A control system for autonomous robotic manipulators must ensure a reliable motion everywhere in the workspace, avoiding collision, system instabilities and/or unwanted motions. Because the human intervention is limited or absent, our research was focused on an autonomous control system as reliable as possible. In order to avoid excessive loosing of performances or, at the limit, a lock in singularity, we choose a robust and efficient singularity avoidance approach.

The algorithm is shown to have the property that the task priority is assigned dynamically: considering the measure of manipulability (index function) as a further task, the approach is suitable for avoiding kinematic singularities, with also a good performance near the singular configurations.

For a given manipulation variable, a singularity-free motion may be usually achieved with an off-line path planning. This approach requires an a-priori knowledge of all the singular configurations of the

manipulator. So, it is difficult to apply in online control schemes. Our approach, based on a real-time evaluation of the measure of manipulability, does not present the above drawback.

Moreover, changing the measure of manipulability with a different index function (such as the minimum distance between the arm and the obstacles) the same approach is suitable for avoiding collision, as well as the joint limits.

Resolved Motion Rate Control

The kinematic output of a generic robotic manipulator is usually represented by a so-called *manipulation variable* $\mathbf{r} \in \mathfrak{R}^m$. A manipulation variable may be, but it is not limited to, the position and orientation of the end-effector.

The relationship between \mathbf{r} and the joint angles \mathbf{q} is represented by the following equation:

$$\mathbf{r} = \mathbf{f}(\mathbf{q}) \quad (1.1)$$

Differentiating with respect to time, the relationship between $\dot{\mathbf{r}}$ and $\dot{\mathbf{q}}$ is given by:

$$\dot{\mathbf{r}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (1.2)$$

$$\mathbf{J}(\mathbf{q}) = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \in \mathfrak{R}^{m \times n} \quad (1.3)$$

where $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix of the manipulation variable \mathbf{r} . In resolved motion rate control **Error! Reference source not found.**, we compute $\dot{\mathbf{q}}$ for a given $\dot{\mathbf{r}}$ and \mathbf{q} by solving the linear system (1.2). In the general case, this is done by using the pseudoinverse of the Jacobian matrix as follow **Error! Reference source not found.**:

$$\dot{\mathbf{q}} = \mathbf{J}^\#(\mathbf{q})\dot{\mathbf{r}} + [\mathbf{I}_n - \mathbf{J}^\#(\mathbf{q})\mathbf{J}(\mathbf{q})]\mathbf{y} \quad (1.4)$$

where $\mathbf{J}^\#(\mathbf{q}) \in \mathfrak{R}^{n \times m}$ is the pseudoinverse of $\mathbf{J}(\mathbf{q})$, $\mathbf{y} \in \mathfrak{R}^n$ is an arbitrary vector and $\mathbf{I}_n \in \mathfrak{R}^{n \times n}$ indicates an identity matrix.

A singular point is defined as the joint configuration vector value $\bar{\mathbf{q}}$ where $\mathbf{J}(\bar{\mathbf{q}})$ is not full rank. Its pseudoinverse $\mathbf{J}^\#(\bar{\mathbf{q}})$ is not defined at such configuration. Moreover, in the neighbor of singular points, even a small change in $\dot{\mathbf{r}}$ requires an enormous change in $\dot{\mathbf{q}}$, which is non-practically feasible in real manipulators and also dangerous for the structure.

The *singularity-robust inverse* (Nakamura) is a classical and simple way to overcome this drawback. It consists in adding a regularization term acting only in the neighbor of the singularities:

$$\mathbf{J}^\# = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \mathbf{I}\mathbf{I})^{-1} \quad (1.5)$$

However, this does not guarantee that the manipulator does not fall into a singularity configuration, from where the successive departure may actually require all a set complex maneuverings, possibly joined with wide movements of the arm itself. Especially in autonomous systems, where the human intervention is limited or absent, the possible presence of the above drawback naturally suggests avoiding all the singular configurations.

Measure of manipulability

The first step in avoiding singularities is to locate themselves in the joint space. Yoshikawa proposed a continuous measure that evaluates the kinematic quality of robot mechanism:

$$mom = \sqrt{\det[\mathbf{J}\mathbf{J}^T]} \quad (1.6)$$

mom takes a continuous non-negative scalar value and becomes equal to zero only when the Jacobian matrix is not full rank. As a matter of fact, the singular value decomposition of the Jacobian matrix is:

$$\mathbf{J}(\mathbf{q}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (1.7)$$

where \mathbf{U} and \mathbf{V} are orthogonal matrixes and $\mathbf{\Sigma}$ is a diagonal matrix whose diagonal elements are the ordered singular values of \mathbf{J} :

$$\mathbf{\Sigma} = \text{diag}(\mathbf{s}_1, \dots, \mathbf{s}_n) \in \mathfrak{R}^{m \times n} \quad (1.8)$$

Substituting eq. (1.7) into eq. (1.6) results in:

$$mom = \sqrt{\det(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T)} = \sqrt{\det(\mathbf{\Sigma}\mathbf{\Sigma}^T)} = \prod_{i=1}^n |\mathbf{s}_i| \quad (1.9)$$

being \mathbf{U} and \mathbf{V} orthogonal matrixes.

Equation (1.9) shows that mom is exactly the product of the singular values of \mathbf{J} and can be regarded as a distance from singularity.

Derivative of measure of manipulability

As will be explained in the next section, it is necessary to find the derivative of eq. (1.6) with respect to the joint configuration vector \mathbf{q} .

Indicating with $j(i, j)$ the element (i, j) of the Jacobian matrix \mathbf{J} , the derivative of mom with respect to the i -th component of \mathbf{q} is given by:

$$\frac{\partial mom}{\partial q_k} = \sum_{i,j} \frac{\partial mom}{\partial j(i, j)} \frac{\partial j(i, j)}{\partial q_k} = \sum_{i,j} \left[\frac{\partial mom}{\partial \mathbf{J}} \right]_{(i,j)} \frac{\partial j(i, j)}{\partial q_k}$$

where $\frac{\partial f}{\partial \mathbf{X}}$ is a matrix whose (i, j) element is $\frac{\partial f}{\partial x_{i,j}}$.

Thus:

$$\begin{aligned} \frac{\partial mom}{\partial q_k} &= \sum_{i,j} \left[\frac{1}{2\sqrt{\det[\mathbf{J}\mathbf{J}^T]}} \frac{\partial \det[\mathbf{J}\mathbf{J}^T]}{\partial \mathbf{J}^T} \right]_{(i,j)}^T \frac{\partial j(i, j)}{\partial q_k} = \\ &= \sum_{i,j} \left[\frac{1}{2\sqrt{\det[\mathbf{J}\mathbf{J}^T]}} 2 \det[\mathbf{J}\mathbf{J}^T] \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1} \right]_{(i,j)}^T \frac{\partial j(i, j)}{\partial q_k} \end{aligned}$$

where the relation

$$\frac{d}{d\mathbf{X}} (\det(\mathbf{X}^T \mathbf{X})) = 2 \det(\mathbf{X}^T \mathbf{X}) \cdot \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \quad (2.1)$$

has been used. Assuming that \mathbf{J} is of full rank we have:

$$\frac{\partial mom}{\partial q_k} = \sum_{i,j} \left[\sqrt{\det[\mathbf{J}\mathbf{J}^T]} \mathbf{J}^\# \right]_{(i,j)}^T \frac{\partial j(i,j)}{\partial q_k} =$$

Finally:

$$\frac{\partial mom}{\partial q_k} = mom \sum_{i,j} \left[\mathbf{J}^{\#T} \right]_{(i,j)} \left[\frac{\partial \mathbf{J}}{\partial q_k} \right]_{(i,j)} \quad (2.2)$$

Equation (2.2) shows that we can express the derivative of measure of manipulability with respect to some already known quantities, as mom itself and the pseudoinverse of Jacobian matrix \mathbf{J} . The derivative of each element of \mathbf{J} with respect to q_k may be easily computed symbolically and is less computationally expensive than the Jacobian itself. This is the only added cost for computing $\partial mom / \partial q_k$.

Singularity avoidance for RMRC

For a given manipulation variable, a singularity-free motion may be usually achieved with an off-line path planning. This approach requires an a-priori knowledge of all the singular configurations of the manipulator.

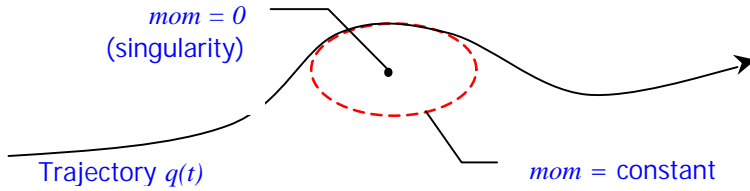


Figure MCTP-1. Singularity-free path in a generic two-dimensional joint space.

The proposed method, based on a real-time evaluation of the measure of manipulability, allows moving along a singularity-free path for a generic manipulator whose singular configurations are not preliminarily known.

The basic idea is to circumscribe singularities by moving, when approaching to them, on a hyper-surfaces where the measure of manipulability is constants. Figure MCTP-1 shows this concept in a generic bi-dimensional joint space \mathcal{R}^2 .

For now, let's consider $\mathbf{y} = 0$ in equation (1.4) (null motion absent):

$$\dot{\mathbf{q}} = \mathbf{J}^\#(\mathbf{q}) \dot{\mathbf{r}} \quad (3.1)$$

The differential $\partial mom(\mathbf{q})$ of the measure of manipulability is given by:

$$\partial mom(\mathbf{q}) = \frac{dmom(\mathbf{q})}{d\mathbf{q}} \partial \mathbf{q} = \frac{dmom(\mathbf{q})}{d\mathbf{q}} \mathbf{J}^\# \partial \mathbf{r} \quad (3.2)$$

In order to have $\partial mom(\mathbf{q}) = 0$, eq. (3.2) implies that the given task must be orthogonal to the vector

$$\frac{d mom(\mathbf{q})}{d \mathbf{q}} \mathbf{J}^\# \quad (3.3)$$

or, equivalently, that $\dot{\mathbf{r}}$ must lie on the surface defined by:

$$\left\{ \mathbf{x} \in \mathfrak{X}^m : \left(\frac{d mom(\mathbf{q})}{d \mathbf{q}} \mathbf{J}^\# \right) \cdot \mathbf{x} = 0 \right\} \quad (3.4)$$

Let \mathbf{n}_m be the unitary vector orthogonal to the surface (3.4):

$$\mathbf{n}_m = \frac{\left(\frac{\partial mom}{\partial \mathbf{q}} \mathbf{J}^\# \right)^T}{\left\| \frac{\partial mom}{\partial \mathbf{q}} \mathbf{J}^\# \right\|} \quad (3.5)$$

Consequently, the projection of the given task is:

$$\dot{\mathbf{r}}_p = \dot{\mathbf{r}} - (\dot{\mathbf{r}} \cdot \mathbf{n}_m) \mathbf{n}_m \quad (3.6)$$

As stated above, such projection must be done only when approaching to singularities. At that aim, it is necessary to introduce in eq. (3.6) a weight as follows:

$$\dot{\mathbf{r}}_p = \dot{\mathbf{r}} - (\dot{\mathbf{r}} \cdot \mathbf{n}_m) \mathbf{n}_m k(mom, \bar{m}) \quad (3.7)$$

where $k(mom, \bar{m})$ is a positive, well shaped function of the measure of manipulability, to be equal to zero for values of mom greater than a predefined threshold \bar{m} and equal to 1 for values of mom smaller than $\bar{m}/2$:

$$k(m, \bar{m}) = \begin{cases} 1 & \bar{m} < m \\ -8 \frac{2m^3 - \frac{9}{2}m^2\bar{m} + 3\bar{m}^2m - \frac{5}{8}\bar{m}^2}{\bar{m}^3} & \frac{\bar{m}}{2} < m < \bar{m} \\ 0 & m < \frac{\bar{m}}{2} \end{cases} \quad (3.8)$$

Figure MCTP-2 shows an example of that function for $\bar{m} = 0.04$.

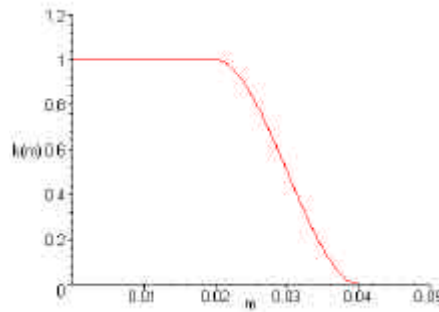


Figure MCTP-2. **Shape function.**

Notice that the first derivative equal to zero in correspondence of those two points. This allows to progressively lying down the task derivative $\dot{\mathbf{r}}$ on the surface where mom is constant, without introducing instabilities on the controller when closing the loop.

However, when mom is already smaller than the value on the surface, eq. (3.7) doesn't guarantee to escape from within the volume enclosed by the same. Moreover, numerical errors may introduce a small derive term driving the task below the surface.

In order to avoid the above drawback, a third term has been introduced in eq. (3.7):

$$\dot{\mathbf{r}}_p = \dot{\mathbf{r}} - (\dot{\mathbf{r}} \cdot \mathbf{n}_m) \mathbf{n}_m k(mom, \bar{m}) + k\left(mom, \frac{\bar{m}}{2}\right) \mathbf{n}_m \quad (3.9)$$

It produces a recalling action toward the surface, starting when $\dot{\mathbf{r}}_p$ has no more components along the gradient (that is when $mom < \bar{m}/2$).

Finally, we must ensure to leave from the surface by acting the task correction in (3.9) only when the scalar product $\dot{\mathbf{r}} \cdot \mathbf{n}_m$ is positive, that is when the measure of manipulability is decreasing:

$$\dot{\mathbf{r}}_p = \dot{\mathbf{r}} - \dot{\mathbf{r}}_{corr} \quad (3.10)$$

where:

$$\dot{\mathbf{r}}_{corr} = \frac{1 - \text{sign}(\dot{\mathbf{r}} \cdot \mathbf{n}_m)}{2} (\dot{\mathbf{r}} \cdot \mathbf{n}_m) \mathbf{n}_m k(mom, \bar{m}) + k\left(mom, \frac{\bar{m}}{2}\right) \mathbf{n}_m \quad (3.11)$$

Thus equation (3.1) becomes:

$$\dot{\mathbf{q}} = \mathbf{J}^\#(\mathbf{q})(\dot{\mathbf{r}} - \dot{\mathbf{r}}_{corr}) \quad (3.12)$$

with $\dot{\mathbf{r}}_{corr}$ given by the (3.11) and:

$$\mathbf{n}_m = \frac{\left(\frac{\partial mom}{\partial \mathbf{q}} \mathbf{J}^\#\right)^T}{\left\|\frac{\partial mom}{\partial \mathbf{q}} \mathbf{J}^\#\right\|}, \quad \frac{\partial mom}{\partial q_k} = mom \sum_{i,j} [\mathbf{J}^{\#T}]_{(i,j)} \left[\frac{\partial \mathbf{J}}{\partial q_k}\right]_{(i,j)} \quad (3.13)$$

Extension to inverse kinematics with task priority

Equation (3.12) has been derived considering the inverse kinematics of one manipulation variable. Let's now extend the study to the inverse kinematics taking account of the priority of the subtasks. The formulation is based upon the Nakamura task-priority based method, with position prior orientation. The goal is to avoid singular configurations for the first manipulation variable.

Let the first manipulation variable $\mathbf{r}_1 = [x \quad y \quad z]^T \in \mathfrak{R}^3$ be the position in \mathfrak{R}^3 of the end-effector.

Likewise, let the second manipulation variable $\mathbf{r}_2 = [R_x \quad R_y \quad R_z]^T \in \mathfrak{R}^3$ be the orientation of the end-effector. Thus we have:

$$\dot{\mathbf{r}}_1 = \mathbf{J}_1(\mathbf{q}) \dot{\mathbf{q}} \quad (4.1)$$

$$\dot{\mathbf{r}}_2 = \mathbf{J}_2(\mathbf{q}) \dot{\mathbf{q}} \quad (4.2)$$

where $\mathbf{q} \in \mathfrak{R}^n$ is the robot configuration vector; $\mathbf{J}_1(\mathbf{q}) \in \mathfrak{R}^{3 \times n}$ and $\mathbf{J}_2(\mathbf{q}) \in \mathfrak{R}^{3 \times n}$ are, respectively, the linear and rotational Jacobian of the system.

The inverse kinematics taking account of the priority of the subtasks is [2] (see section 0):

$$\dot{\mathbf{q}} = \mathbf{J}_1^\# \dot{\mathbf{r}}_1 + \hat{\mathbf{J}}_2^\# \cdot (\dot{\mathbf{r}}_2 - \mathbf{J}_2 \mathbf{J}_1^\# \dot{\mathbf{r}}) + (\mathbf{I}_n - \mathbf{J}_1^\# \mathbf{J}_1) (\mathbf{I}_n - \hat{\mathbf{J}}_2^\# \hat{\mathbf{J}}_2) \mathbf{z} \quad (4.3)$$

$$\hat{\mathbf{J}}_2 = \mathbf{J}_2 (\mathbf{I}_n - \mathbf{J}_1^\# \mathbf{J}_1) \quad (4.4)$$

where $\mathbf{J}_1^\#(\mathbf{q}) \in \mathfrak{R}^{n \times 3}$ is the pseudoinverse of $\mathbf{J}_1(\mathbf{q})$, $\hat{\mathbf{J}}_2^\#(\mathbf{q}) \in \mathfrak{R}^{n \times 3}$ is the pseudoinverse of $\hat{\mathbf{J}}_2(\mathbf{q})$, $\mathbf{z} \in \mathfrak{R}^n$ is an arbitrary vector and $\mathbf{I}_n \in \mathfrak{R}^{n \times n}$ indicates an identity matrix.

Let's consider $\mathbf{z} = 0$ in equation (4.3) (null motion absent):

$$\dot{\mathbf{q}} = \mathbf{J}_1^\# \dot{\mathbf{r}}_1 + \hat{\mathbf{J}}_2^\# \cdot (\dot{\mathbf{r}}_2 - \mathbf{J}_2 \mathbf{J}_1^\# \dot{\mathbf{r}}) \quad (4.5)$$

The measure of manipulability mom_1 of the first manipulation variable \mathbf{r}_1 is given by:

$$mom_1 = \sqrt{\det[\mathbf{J}_1 \mathbf{J}_1^T]} \quad (4.6)$$

The differential $\partial mom_1(\mathbf{q})$ of the measure of manipulability is given by:

$$\partial mom_1(\mathbf{q}) = \frac{dmom_1(\mathbf{q})}{d\mathbf{q}} \partial \mathbf{q} = \frac{dmom_1(\mathbf{q})}{d\mathbf{q}} \left((\mathbf{J}_1^\# - \hat{\mathbf{J}}_2^\# \mathbf{J}_2 \mathbf{J}_1^\#) \partial \mathbf{r}_1 + \hat{\mathbf{J}}_2^\# \partial \mathbf{r}_2 \right) \quad (4.7)$$

In order to have $\partial mom_1(\mathbf{q}) = 0$, eq. (4.7) implies:

$$\begin{aligned} \left(\frac{dmom_1(\mathbf{q})}{d\mathbf{q}} (\mathbf{J}_1^\# - \hat{\mathbf{J}}_2^\# \mathbf{J}_2 \mathbf{J}_1^\#) \right) \cdot \partial \mathbf{r}_1 &= 0 \\ \left(\frac{dmom_1(\mathbf{q})}{d\mathbf{q}} \hat{\mathbf{J}}_2^\# \right) \cdot \partial \mathbf{r}_2 &= 0 \end{aligned} \quad (4.8)$$

or, equivalently, that $\dot{\mathbf{r}}_1$ and $\dot{\mathbf{r}}_2$ must be respectively orthogonal to the vectors:

$$\mathbf{n}_1 = \frac{\left(\frac{dmom_1(\mathbf{q})}{d\mathbf{q}} (\mathbf{J}_1^\# - \hat{\mathbf{J}}_2^\# \mathbf{J}_2 \mathbf{J}_1^\#) \right)^T}{\left\| \frac{dmom_1(\mathbf{q})}{d\mathbf{q}} (\mathbf{J}_1^\# - \hat{\mathbf{J}}_2^\# \mathbf{J}_2 \mathbf{J}_1^\#) \right\|}, \mathbf{n}_2 = \frac{\left(\frac{dmom_1(\mathbf{q})}{d\mathbf{q}} \hat{\mathbf{J}}_2^\# \right)^T}{\left\| \frac{dmom_1(\mathbf{q})}{d\mathbf{q}} \hat{\mathbf{J}}_2^\# \right\|} \quad (4.9)$$

Consequently, the projections of the given tasks are:

$$\dot{\mathbf{r}}_{1p} = \dot{\mathbf{r}}_1 - (\dot{\mathbf{r}}_1 \cdot \mathbf{n}_{1m}) \mathbf{n}_{1m} \quad (4.10)$$

$$\dot{\mathbf{r}}_{2p} = \dot{\mathbf{r}}_2 - (\dot{\mathbf{r}}_2 \cdot \mathbf{n}_{2m}) \mathbf{n}_{2m} \quad (4.11)$$

Finally, equation (4.3) becomes:

$$\dot{\mathbf{q}} = \mathbf{J}_1^\# (\dot{\mathbf{r}}_1 - \dot{\mathbf{r}}_{1corr}) + \hat{\mathbf{J}}_2^\# \cdot (\dot{\mathbf{r}}_2 - \dot{\mathbf{r}}_{2corr} - \mathbf{J}_2 \mathbf{J}_1^\# (\dot{\mathbf{r}}_1 - \dot{\mathbf{r}}_{1corr})) \quad (4.12)$$

where (see equation (3.11)):

$$\dot{\mathbf{r}}_{icorr} = \frac{1 - \text{sign}(\dot{\mathbf{r}}_i \cdot \mathbf{n}_{im})}{2} (\dot{\mathbf{r}}_i \cdot \mathbf{n}_{im}) \mathbf{n}_{im} k(mom_1, \bar{m}) + k\left(mom_1, \frac{\bar{m}}{2}\right) \mathbf{n}_{im} \quad (4.13)$$

with \mathbf{n}_1 and \mathbf{n}_2 defined as in (4.9).

Application example

Equation (4.12) has been used in order to avoid singularities for the first manipulation variable (position). This, of course, doesn't guarantee a singularity-free path for the second manipulation variable (orientation) and the manipulation may fall into a configuration where one or more singular values of the Jacobian $\hat{\mathbf{J}}_2$ are zero.

In the current implementation, in order to pseudo-invert $\hat{\mathbf{J}}_2(\mathbf{q})$, see eq. (4.4)., we used the weighted singularity-robust inverse, as follows:

$$\hat{\mathbf{J}}_2^\# = \mathbf{W} \hat{\mathbf{J}}_2^T (\mathbf{J}_2 \mathbf{W} \hat{\mathbf{J}}_2^T + \mathbf{I} \mathbf{I}_n)^{-1} \quad (5.1)$$

The weight $\mathbf{W} \in \mathfrak{R}^{n \times n}$ is a diagonal matrix whose diagonal elements $w_{i,i}$ are given by:

$$w_{i,i} = \left(1 - k \left(mom_2, \bar{m}_2\right) |n_{3i}|\right)^2 \quad (5.2)$$

being n_{3i} the i-th component of the unitary vector

$$\mathbf{n}_3 = \frac{\begin{pmatrix} \frac{\partial mom_2}{\partial \mathbf{q}} \end{pmatrix}^T}{\left\| \frac{\partial mom_2}{\partial \mathbf{q}} \right\|} \quad (5.3)$$

where:

$$mom_2 = \sqrt{\det \left[\hat{\mathbf{J}}_2 \hat{\mathbf{J}}_2^T \right]} \quad (5.4)$$

and $k(mom, \bar{m})$ defined as in equation (3.8).

This allows to progressively attenuate, when approaching to a small values of mom , the components of $\dot{\mathbf{q}}$ responsible of the greatest change in the rate of measure of manipulability.

The value of \mathbf{I} is chosen according to the distance of $\hat{\mathbf{J}}_2$ from its singular configuration:

$$\mathbf{I} = k(mom_2, \bar{m}_2) \quad (5.5)$$

and is acting only when mom_2 is lower than the constant limit \bar{m}_2 .

Figure MCTP-3 and Figure MCTP-4 show a simulative result (see also the movie clip). In this example the given task is a circle partially enclosed in the workspace: the manipulator tries to follow the path, giving the highest priority to the position when running far from a singular configuration. Instead, when approaching to the boundary of the workspace, the first manipulation variable task is performed with an error necessary to keep the measure of manipulability equal to the lower limit $\bar{m}/2$ (**Error! Reference source not found.**).

A comparison with Nakamura singular-robust pseudo-inversion shows that our method has a faster error recovery. As a matter of fact, because the measure of manipulability is never zero, we can use

the exact pseudo-inversion minimizing the (projected) task error. The last differs from the original task only when this is necessary for limiting *mom*.

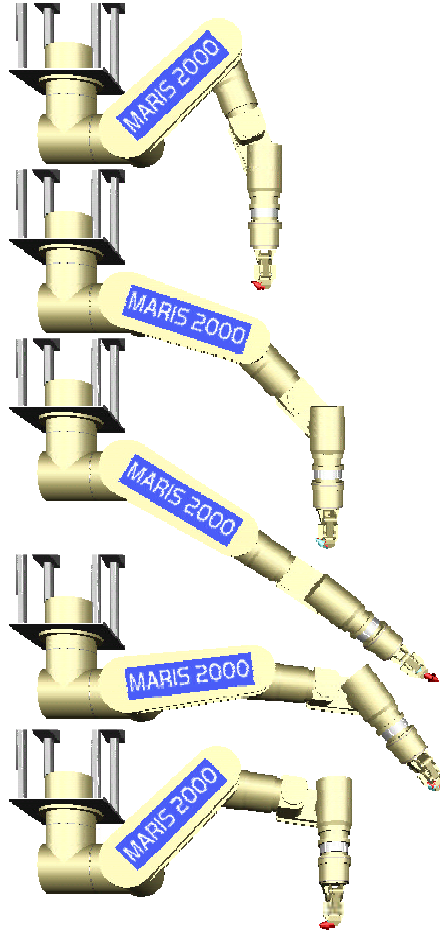


Figure MCTP-3. **MARIS 7080 Underwater Manipulator: task sequences.**

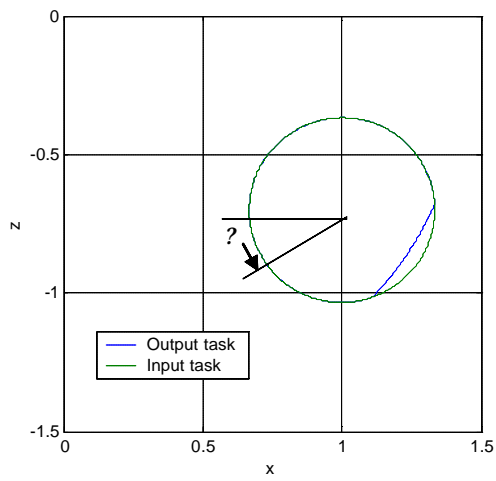


Figure MCTP-4. **Linear task in the x-z plane.**

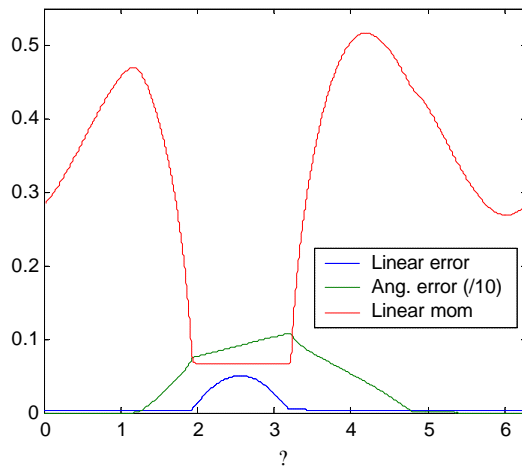


Figure MCTP-5. **Task errors and measure of manipulability, with respect to the angle q of the circle.**

Notice that it is possible to regard the proposed method as a kind of dynamic priority-changing algorithm. As shown in Figure MCTP-5, when the measure of manipulability of the given task approaching to zero, the highest priority is given to the distance form singularity.

Notes on the inverse kinematics taking account of the priority of the subtasks

In autonomous robotic systems, the subtask decomposition between position and orientation is advantageous, because it will enlarge the reachable workspace of the first-*priority* manipulation variable (usually position) by allowing incompleteness for the second priority subtask.

Here, a slightly different formulation of the problem is proposed. This mathematical formulation is the one actually used for the most recent version of Maris controller.

Let the first manipulation variable $\mathbf{r}_1 = [x \ y \ z]^T \in \mathfrak{R}^3$ be the position in \mathfrak{R}^3 of the end-effector:

$$r_1 = f_1(\mathbf{q}) \quad (5.6)$$

where $\mathbf{q} \in \mathfrak{R}^n$ is the robot configuration vector and \mathbf{r}_1 is the position of the end-effector. The differential relationship of (5.6) is:

$$\dot{r}_1 = \mathbf{J}_1(\mathbf{q}) \dot{\mathbf{q}} \quad (5.7)$$

Likewise, let the second manipulation variable $\mathbf{r}_2 = [R_x \ R_y \ R_z]^T \in \mathfrak{R}^3$ be the orientation of the end-effector:

$$r_2 = f_2(\mathbf{q}) \quad (5.8)$$

$$\dot{r}_2 = \mathbf{J}_2(\mathbf{q}) \dot{\mathbf{q}} \quad (5.9)$$

$\mathbf{J}_1(\mathbf{q}) \in \mathfrak{R}^{3 \times n}$ and $\mathbf{J}_2(\mathbf{q}) \in \mathfrak{R}^{3 \times n}$ are, respectively, the linear and rotational Jacobian of the system. Equation (5.7) has an infinite variety of solution of $\dot{\mathbf{q}}$, whose general solution is obtained using the pseudoinverse solution of the Jacobian matrix:

$$\dot{\mathbf{q}} = \mathbf{J}_1^\#(\mathbf{q}) \dot{r}_1 + [\mathbf{E}_n - \mathbf{J}_1^\#(\mathbf{q}) \mathbf{J}_1(\mathbf{q})] \mathbf{y} \quad (5.10)$$

where $\mathbf{J}_1^\#(\mathbf{q}) \in \mathfrak{R}^{n \times 3}$ is the pseudoinverse of $\mathbf{J}_1(\mathbf{q})$, $\mathbf{y} \in \mathfrak{R}^n$ is an arbitrary vector and $\mathbf{E}_n \in \mathfrak{R}^{n \times n}$ indicates an identity matrix.

Substituting eq. (5.10) into eq. (5.9) we obtain:

$$\dot{r}_2 = \mathbf{J}_2 \left[\mathbf{J}_1^\# \dot{r}_1 + (\mathbf{E}_n - \mathbf{J}_1^\# \mathbf{J}_1) \mathbf{y} \right] \quad (5.11)$$

$$\mathbf{J}_2 (\mathbf{E}_n - \mathbf{J}_1^\# \mathbf{J}_1) \mathbf{y} = \dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^\# \dot{r}_1 \quad (5.12)$$

If the exact solution of \mathbf{y} exists, eq. (5.12) implies that the second manipulation variable can be realized. The exact solution doesn't generally exist. However, we can obtain \mathbf{y} that minimize $\|\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^\# \dot{r}_1\|$ in the least square sense by using again the pseudoinverse:

$$\mathbf{y} = \hat{\mathbf{J}}_2^\# \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^\# \dot{r}_1) + (\mathbf{E}_n - \hat{\mathbf{J}}_2^\# \hat{\mathbf{J}}_2) \mathbf{z} \quad (5.13)$$

$$\hat{\mathbf{J}}_2 = \mathbf{J}_2 (\mathbf{E}_n - \mathbf{J}_1^\# \mathbf{J}_1) \quad (5.14)$$

Finally, substituting eq. (5.13) into eq. (5.10) we obtain:

$$\dot{\mathbf{q}} = \mathbf{J}_1^\# \dot{r}_1 + (\mathbf{E}_n - \mathbf{J}_1^\# \mathbf{J}_1) \hat{\mathbf{J}}_2^\# \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^\# \dot{r}_1) + (\mathbf{E}_n - \mathbf{J}_1^\# \mathbf{J}_1) (\mathbf{E}_n - \hat{\mathbf{J}}_2^\# \hat{\mathbf{J}}_2) \mathbf{z} \quad (5.15)$$

Let now introduce a third manipulation variable $r_3 \in \mathfrak{R}$:

$$r_3 = f_3(\mathbf{q}) \quad (5.16)$$

$$\dot{r}_3 = \mathbf{J}_3(\mathbf{q}) \dot{\mathbf{q}} \quad (5.17)$$

Substituting equation (5.15) into (5.17) yields:

$$\dot{r}_3 = \mathbf{J}_3(\mathbf{q}) \left[\mathbf{J}_1^{\#} \dot{r}_1 + (\mathbf{E}_n - \mathbf{J}_1^{\#} \mathbf{J}_1) \hat{\mathbf{J}}_2^{\#} \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^{\#} \dot{r}_1) + (\mathbf{E}_n - \mathbf{J}_1^{\#} \mathbf{J}_1) (\mathbf{E}_n - \hat{\mathbf{J}}_2^{\#} \hat{\mathbf{J}}_2) z \right] \quad (5.18)$$

$$\mathbf{J}_3 (\mathbf{E}_n - \mathbf{J}_1^{\#} \mathbf{J}_1) (\mathbf{E}_n - \hat{\mathbf{J}}_2^{\#} \hat{\mathbf{J}}_2) z = \dot{r}_3 - \mathbf{J}_3 \mathbf{J}_1^{\#} \dot{r}_1 - \mathbf{J}_3 \hat{\mathbf{J}}_2^{\#} \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^{\#} \dot{r}_1) \quad (5.19)$$

$$z = \hat{\mathbf{J}}_3^{\#} \left[\dot{r}_3 - \mathbf{J}_3 \mathbf{J}_1^{\#} \dot{r}_1 - \mathbf{J}_3 \hat{\mathbf{J}}_2^{\#} \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^{\#} \dot{r}_1) \right] \quad (5.20)$$

$$\hat{\mathbf{J}}_3 = \mathbf{J}_3 (\mathbf{E}_n - \mathbf{J}_1^{\#} \mathbf{J}_1) (\mathbf{E}_n - \hat{\mathbf{J}}_2^{\#} \hat{\mathbf{J}}_2) = \mathbf{J}_3 (\mathbf{E}_n - \mathbf{J}_1^{\#} \mathbf{J}_1 - \hat{\mathbf{J}}_2^{\#} \hat{\mathbf{J}}_2) \quad (5.21)$$

Thus:

$$\dot{\mathbf{q}} = \mathbf{J}_1^{\#} \dot{r}_1 + \hat{\mathbf{J}}_2^{\#} \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^{\#} \dot{r}_1) + (\mathbf{E}_n - \mathbf{J}_1^{\#} \mathbf{J}_1) (\mathbf{E}_n - \hat{\mathbf{J}}_2^{\#} \hat{\mathbf{J}}_2) \hat{\mathbf{J}}_3^{\#} \left[\dot{r}_3 - \mathbf{J}_3 \mathbf{J}_1^{\#} \dot{r}_1 - \mathbf{J}_3 \hat{\mathbf{J}}_2^{\#} \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^{\#} \dot{r}_1) \right] \quad (5.22)$$

$$\begin{aligned} & \mathbf{J}_1^{\#} \dot{r}_1 + \hat{\mathbf{J}}_2^{\#} \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^{\#} \dot{r}_1) + \hat{\mathbf{J}}_3^{\#} \left[\dot{r}_3 - \mathbf{J}_3 \mathbf{J}_1^{\#} \dot{r}_1 - \mathbf{J}_3 \hat{\mathbf{J}}_2^{\#} \cdot (\dot{r}_2 - \mathbf{J}_2 \mathbf{J}_1^{\#} \dot{r}_1) \right] = \\ & = \mathbf{J}_1^{\#} \dot{r}_1 + \hat{\mathbf{J}}_2^{\#} (\dot{r}_2 - \mathbf{J}_2 \dot{\mathbf{q}}_1) + \hat{\mathbf{J}}_3^{\#} \left[\dot{r}_3 - \mathbf{J}_3 (\dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_2) \right] \end{aligned} \quad (5.23)$$

$$\dot{\mathbf{q}}_1 = \mathbf{J}_1^{\#} \dot{r}_1, \quad \dot{\mathbf{q}}_2 = \hat{\mathbf{J}}_2^{\#} (\dot{r}_2 - \mathbf{J}_2 \dot{\mathbf{q}}_1)$$

Equation (5.23) suggests a recursive idea:

$$\begin{cases} \dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + \hat{\mathbf{J}}_i^{\#} (\dot{r}_i - \mathbf{J}_i \dot{\mathbf{q}}_{i-1}) \\ \hat{\mathbf{J}}_i = \mathbf{J}_i \mathbf{J}_i^n \\ \mathbf{J}_i^n = \mathbf{J}_{i-1}^n - \hat{\mathbf{J}}_{i-1}^{\#} \hat{\mathbf{J}}_{i-1} \end{cases}, \quad \begin{cases} \dot{\mathbf{q}}_0 = \mathbf{0} \\ \mathbf{J}_0 = \mathbf{0} \\ \mathbf{J}_0^n = \mathbf{I}_n \end{cases} \quad (5.24)$$

that is the final form implemented within Maris control software.

Algorithmic singularities

As seen in par. 0, the above approach has been used to avoid singular configurations for the first manipulation variable, which happen when $\hat{\mathbf{J}}_1 = \mathbf{J}_1$ in equation (5.24) is not full rank. However, for $i > 1$, equation (5.24) holds only when $\hat{\mathbf{J}}_i$ is full rank. Conversely, the pseudo-inverse $\hat{\mathbf{J}}_i^{\#}$ of $\hat{\mathbf{J}}_i$ is not defined and we fall in algorithmic singularity.

In the last part of our research we were able to solve this problem with an extension of the above methodology. Even though the preliminary results were good, the study is still under development. For this reason, the algorithmic singularity avoidance approach will be fully illustrated in one of the next technical report.

Collision and joint limits avoidance

With a different choice of the index function, the above approach has been proved valid to avoid obstacle collisions. The new index function is the minimum distance between the arm and the obstacle, computed simplifying each solid with a bounding box (see Figure MCTP-6). In this way the procedure for computing the minimum distance is simple and efficient.

Its derivative, however, has not a symbolic closed form and in our experiments has been computed numerically. Moreover, it presents several discontinuities resulting in some stability problems. For this reason, the experiment has been done only in simulation mode. The result is shown in the attached clip - here, the first (and only) manipulation variable is the 3 *dof* end-effector position.

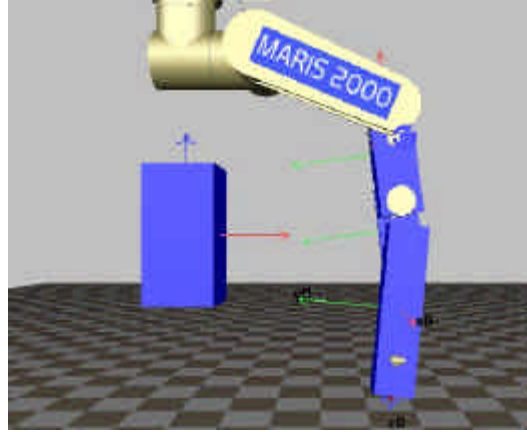


Figure MCTP-6. **Bounding boxes for computing the minimum object distance.**

Another index suitable of use with the precedent method is a potential function of the joint position, taking large values when the joints are far from their limits. Indicating with q_{iMin} and q_{iMax} the mechanical limits of the i -th joint, its potential function is:

$$Jpf_i = -4 \frac{q_i^2 - q_i q_{iMin} - q_i q_{iMax} + q_{iMin} q_{iMax}}{(q_{iMax} - q_{iMin})^2} \quad (6.1)$$

The global index is:

$$Jpf = \prod_{i=1}^{dof} Jpf_i \quad (6.2)$$

Figure MCTP-7 shows a plot of eq. (6.1) for $q_{iMin} = -p$ and $q_{iMax} = p$.

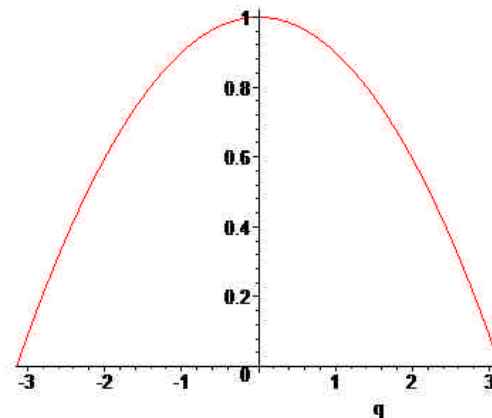


Figure MCTP-7. **The joint potential function with $q_{iMin} = -p$ and $q_{iMax} = p$.**

Code Optimization and speed-up

All the experiment described in the SAUVIM - Phase I report were made trough a fast prototyping approach in order to focus on the algorithmic aspect. For the release version, as the one currently

under development, has been necessary migrating to a less redundant framework in order to reduce the overall sample time and also to allow a more flexible and expandable structure. The Code Optimization and speed-up process involved a new numerical algorithm for pseudo-inversion, a drastic optimization of procedures for deriving kinematical quantities (all-in-one procedure) and a new non-prototyping task scheduler for VxWorks OS, based on the Real Time Workshop's core. Moreover, all the board software drivers have been rewritten in stand-alone Ansi-C language and are suitable of use under xBus (Maris Command Interface).

Numerical algorithm for pseudo-inversion

In order to speed-up the computation of the measure of manipulability:

$$mom = \sqrt{\det[\mathbf{J}\mathbf{J}^T]}, \quad \mathbf{J} \in \mathfrak{R}^{m \times n} \quad (7.1)$$

we can use some property of the product of the matrix

$$\mathbf{M} = \mathbf{J}\mathbf{J}^T \quad (7.2)$$

Because \mathbf{M} is a symmetric semi-positive definite matrix, it has a special, more efficient, triangular decomposition: the Cholesky decomposition **Error! Reference source not found.** We can factorize the matrix \mathbf{M} of equation (7.2) as follow:

$$\mathbf{L}\mathbf{L}^T = \mathbf{M} \quad (7.3)$$

where \mathbf{L} is a lower triangular matrix. The components of \mathbf{L} are:

$$L_{i,i} = \sqrt{m_{ii} - \sum_{k=1}^{i-1} L_{i,k}^2} \quad (7.4)$$

$$L_{j,i} = \frac{1}{L_{i,i}} \left(m_{i,j} - \sum_{k=1}^{i-1} L_{i,k} L_{j,k} \right), \quad j = i+1, i+2, \dots, n \quad (7.5)$$

Thus we have:

$$mom = \sqrt{\text{trace}(\mathbf{L})} \quad (7.6)$$

The operation's count is about $m^3 / 6$ multiplication and subtractions, with also $m+1$ square roots. In our application, it takes about 0.4 ms on a 68060 Motorola CPU (40 MHz).

A similar approach has been used to pseudo-inverting the Jacobian matrix. In this way, the overall time for computing equation (4.12) is less than 2 ms on the above hardware.

All the simulative tests were made with the aid of *Robosim*, a software package for simulation and fast prototyping in robotics applications.

Servo Actuator System

The VME-based computing architecture (see Figure MCTP-8) uses a FORCE 68060 board to implement the control scheme, which in turn provides the references to the Elmo motor drivers through the AD/DA board a MATRIX MD-DAADIO. The development of the software driver for the motor section has been done with particular care to linearity problems, safety issues and robustness. The following paragraphs illustrate with more details the features of the *Motors Driver Module*.

Hardware Motor Controller unit

Elmo NBA (Figure MCTP-9) are a miniature PWM, full wave, three phase servo amplifier designed for high performance brushless servo motors.

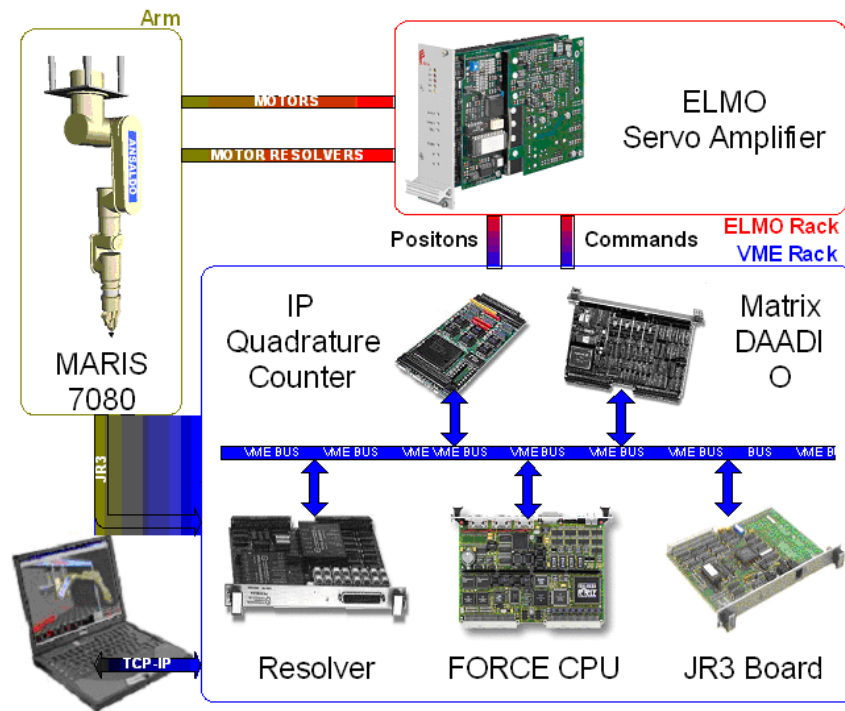


Figure MCTP-8. The manipulator hardware structure.

They utilize power MOSFETs and Surface Mounting Technology which contribute to their high efficiency and compact design. Motor drives are constructed from two PCBs mounted on a heat sink plate. The lower board contains the power switching transistors, which drive the motor, control section, terminals for both the power stage and control stage, and the protection logic. The upper PCB contains the switch mode power supply.

The main features are:

- Internal DC to DC converter allows operation from a single supply
- 20 KHz switching frequency
- 97% efficiency
- Output voltage is up to 90% of Output voltage is up to 90% of input voltage
- Zero deadband
- Motor current monitor
- Operation in velocity or current mode
- DC supply 40-195 V
- Current limits 6/12 A (continuous/peak)
- Size rack 3U/8T
- Weight 0.7 Kg Operating temperature 0 - 50°C
- Storage temperature -10 - +70°C

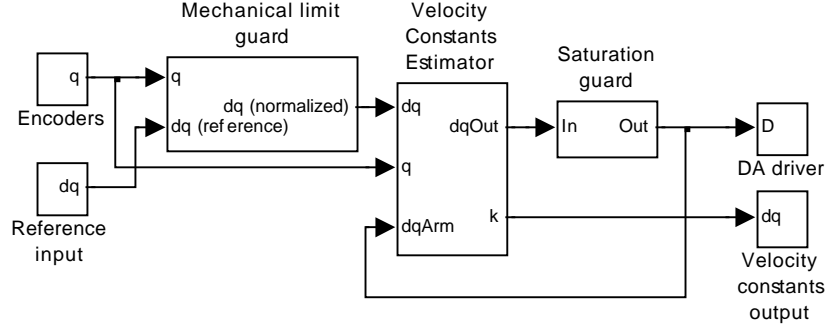


Figure MCTP-10. **Block diagram of Motors Driver (joint section).**

Mechanical Limit Guard

This block scales the velocity when any joint approaches to its mechanical stop. It normalizes *every* joint velocity even if only a joint in approaching to the end of its stroke. Internally, the block estimates the time of impact and tries to avoid it. If t_{ii} , the *time-to-impact*, is less than a predefined value t_{max} , the entire vector of velocities is scaled by a function of the same t_{ii} . Analytically we have:

$$v_{i out} = \begin{cases} 0 & q_i \notin [q_{imin}, q_{imax}] \\ v_{i corr} & q_i \in [q_{imin}, q_{imax}] \end{cases} \quad (9.1)$$

where:

$$v_{i corr} = \begin{cases} \left(\frac{t_{ii}}{t_{max}} \right)^2 \left(3 - 2 \frac{t_{ii}}{t_{max}} \right) v_{i in} & t_{ii} < t_{max} \\ v_{i in} & t_{ii} \geq t_{max} \end{cases} \quad (9.2)$$

The time-to-impact is estimated as follows:

$$t_{ii} = \min_i \frac{\Delta q_i}{v_{i in}} \quad (9.3)$$

with:

$$\Delta q_i = \begin{cases} q_{i min} - q_i & v_{i in} < 0 \\ q_{i max} - q_i & v_{i in} \geq 0 \end{cases} \quad (9.4)$$

The vectors \mathbf{q}_{min} and \mathbf{q}_{max} of the actual joint's mechanical limit are:

$$\mathbf{q}_{min} = \mathbf{a} [-3.0369 \quad -2.9845 \quad -3.7699 \quad -2.5838 \quad -3.1241 \quad -2.2515 \quad -3.3685] \quad (9.5)$$

$$\mathbf{q}_{max} = \mathbf{a} [2.9496 \quad 2.9845 \quad 0.6196 \quad 2.5831 \quad 3.1416 \quad 2.2166 \quad 2.8274] \quad (9.6)$$

For the safety constant \mathbf{a} we chose the value 9/10.

Velocity Constants Estimator

Actually, the Elmo servo amplifiers implement a hardware velocity loop PID controller. The angular velocity of each motor is related to the board control input voltage V_{iref} by the relation:

$$\mathbf{w}_i(t) \text{ (rad / sec)} = k_{vi}(t) V_{iref}(t) \quad (10.1)$$

where k_{vi} is the board velocity constant. Notice that the main difference with the equation (8.1) is that the velocity constant is different for each board and can be (slowly) time-variant. Actually it depends mainly by the gear ratio (see table MCTC-1) and by the tunable constants of the hardware PID controller.

In order to have a linear behavior, as in equation (8.1), we first need to estimate all the k_{vi} constants. At the given time instant k , from eq. (10.1) we have:

$$(k_i)_k = \frac{(\mathbf{w}_i)_k}{(V_{iref})_k} \cong \frac{(q_i)_{k+1} - (q_i)_k}{\Delta t} \frac{1}{(V_{iref})_k} \text{ [rad / (s \cdot V)]} \quad (10.2)$$

or, similary:

$$(k_i)_{k-1} = \frac{(\mathbf{w}_i)_{k-1}}{(V_{iref})_{k-1}} \cong \frac{(q_i)_k - (q_i)_{k-1}}{\Delta t} \frac{1}{(V_{iref})_{k-1}} \text{ [rad / (s \cdot V)]} \quad (10.3)$$

Taking a mean value over a window of n samples, we have:

$$\bar{k}_i = \sum_{j=1}^n (k_i)_{k-j} \quad (10.4)$$

Thus, from eq. (8.1), we have:

$$V_{iref} = \frac{\bar{k}_i}{k_i} \mathbf{w}_{iref} \quad (10.5)$$

In other words, given a reference value \mathbf{w}_{iref} , in order to have the actual angular velocity on the joint as in (8.1), the voltage reference input to the i -th board must be as in eq. (10.5), and this is the correction performed by the block *Velocity Constants Estimator*.

In the actual implementation, in order to compute the mean value (10.4), only the samples k_i included in a predefined range are considered. Values not verifying the relation

$$|k_i - \bar{k}_i| \leq \bar{k}_i \quad (10.6)$$

are discharged. \bar{k}_i is the (fixed) nominal mean value of the i -th board-joint pair. Notice that eq. (10.6) allows to consider the value 0 as a valid sample, giving to the mean value the opportunity to reflect a failure of the motor as it reaches zero. In the last case, eq. (10.5) cannot be used to compute the voltage reference. Instead, if a motor breaks, we use the a-priori known mean value constant \bar{k}_i as the current one:

$$\bar{k}_i = \begin{cases} \sum_{j=1}^n (k_i)_{k-j} & \sum_{j=1}^n (k_i)_{k-j} > \frac{\bar{k}_i}{2} \\ \bar{k}_i & \sum_{j=1}^n (k_i)_{k-j} \leq \frac{\bar{k}_i}{2} \end{cases} \quad (10.7)$$

In this way, a drive is supposed to break when the absolute value of its integrative constant is lower than the limit $\frac{\bar{k}_i}{2}$.

Finally, the following tips have been used:

- 1) If the voltage reference V_{iref} appearing in eq. (10.2) is under 10 times the resolution of the DA converter, we do not consider it valid for a speed measure.
- 2) For a more stable behavior at $t = 0$, the history window for computing the mean value of eq. (10.4) has been initialized to the nominal mean value \bar{k}_i .

Joint	Total Ratio	Reduction	Joint	Total Ratio	Reduction
J1	100	$\frac{100}{19}$	J5	160	$\frac{80}{18}$
J2	100	$\frac{100}{19}$	J6	160	$\frac{70}{22}$
J3	100	$\frac{100}{19}$	J7	120	$\frac{45}{18}$
J4	160	$\frac{80}{30} \frac{56}{23}$	EE	8mm / turn	

Table MCTC-1. Joint's gear ratio.

DA driver

This block maps the input integer(s), between 0 and 4095, to the output voltage(s) (between $-10V$ and $9.9951171875V$) of the MATRIX DAADIO board:

$$V_{out} = 20 \frac{D - 2048}{4096}, \quad 0 \leq D \leq 4095 \quad (11.1)$$

Motor saturation

Actually the module of motors speed is limited to the maximum value correspondent to the maximum input voltage allowed by the Elmo controller board. Because this saturation may affect each joint in a different way, the manipulator may fail to track a reference velocity input. The *SatGuard* function

ensures a correct track by attenuating, proportionally, all the joint velocities when one is approaching to its saturation value. Given \dot{q}_{max} the maximum of the absolute values of the reference components \dot{q}_i :

$$\dot{q}_{max} = \max_i |\dot{q}_i| \quad (12.1)$$

The normalized value of each component is given by:

$$\dot{q}_{in} = \begin{cases} \dot{q}_i & \dot{q}_{max} \leq \dot{q}_{lim} \\ \dot{q}_i \frac{\dot{q}_{lim}}{\dot{q}_{max}} & \dot{q}_{max} > \dot{q}_{lim} \end{cases} \quad (12.2)$$

where \dot{q}_{lim} is the maximum value allowed for \dot{q} . The plot in Figure MCTP-11 shows the behavior of equation (12.2) for $\dot{q}_{lim} = 10$.

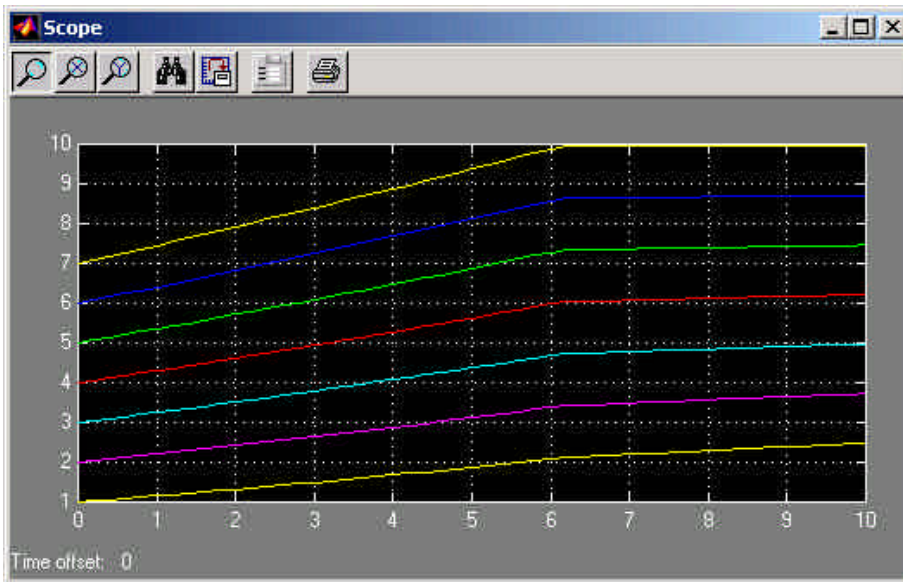


Figure MCTP-11. *SatGuard* behavior.

Data exchange Bus and Command Interface (xBus)

The ability of exchanging data with the external environment is one of the most important aspects of Maris Control System. Data includes manipulator sensor monitor (output) and command operation (input), as for instance those for enabling the motors, calibrating the sensors, operating the task controller or for using the arm in teleoperation mode.

The first prototyping version of Maris Control Software, as described in the Phase 1 report, included a simple TCP-IP communicator block capable of transferring an array of float from an to the arm.

Although efficient for testing purposes, it presents some limitation when fail-safe and high-level command operations are required, as in autonomous systems.

Those objectives have been reached with *xBus*, which includes extended capabilities such as those described above.

xBus is a TCP-IP based client-server communication system, with an integrated interpreter suitable of running high-level scripts through the client interface. The server can accept any number of client connections, each of any can count on a error-robust communication protocol capable of auto-reconnection in case of a temporary network failure. This is important in a hostile environment, when the communication media does not allows safe and durable connections (acoustic modems).

Figure MCTP-12 shows a block diagram of the entire *xBus* system; the next subsections will explain with more details the contents of each sub-system.

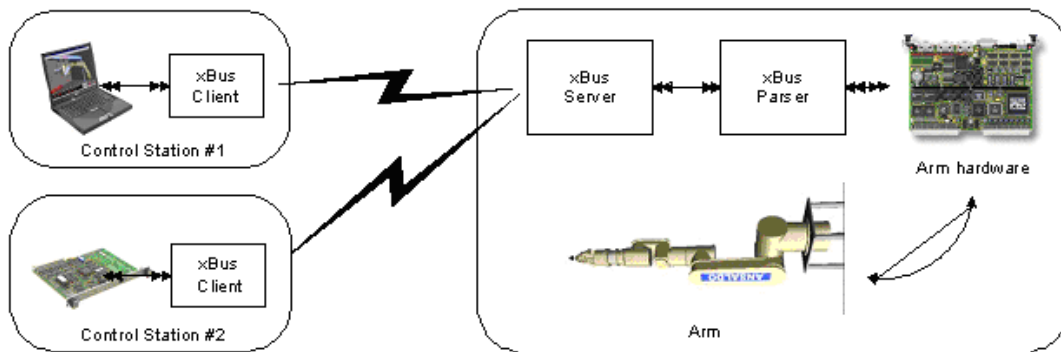


Figure MCTP-12. **xBus basic block structure.**

xBus Server

Most of the network-based servers (such as FTP servers or HTTP servers) use a multi-thread approach for handling each client connection. *xBus*, on the other side, uses a different concept in order to accomplish the requirements of the command interpreter. As a matter of fact, the last is shared between each connections and a parallel multi-thread approach would result in synchronization problems.

Internally *xBus Server* is a meta-state machine, or a set of finite state machine. These machines, one for each client connection, are called sequentially so that each one can request a different command execution to the parser. It is matter of the parser allowing or not the execution of each command, according to its priority with respect the already running ones.

Figure MCTP-13 shows the internal structure of *xBus Server*. The main loop first monitors any incoming connection request: upon its arrival, the server instantiates a dedicated finite state machine.

Each state machine is described by the following pseudo-language, borrowed from the FSM description of Imatix's Libero software package (state-event-action):


```

(-- ) Write-Timeout                -> Socket-Reading
    + Set-Write-Timeout-Error
    + Pop-Command-From-Read-Buffer
    + Process-Read-Buffer
    + Read-Data

```

Closing-Socket :

```

(-- ) Shutdown-Successful          ->
    + Do-Close-Socket
    + Terminate-The-Program
(-- ) Wsaeprogress1-Error          -> Closing-Socket
    + Shutdown-Send
(-- ) Wsaeprogress2-Error          -> Closing-Socket
    + Shutdown-Both
(-- ) Shutdown-Error              ->
    + Do-Close-Socket
    + Terminate-The-Program

```

It is possible to recognize three main states: Socket-Reading, Socket-Writing and Closing-Socket.

The first, Socket-Reading, basically waits for incoming data from the client. After the Read-Complete event, the actions Execute-Command, Prepare-Write-Buffer, Init-Timeout and Socket-Write are taken, right before entering the state Socket-Writing. The first action consists in sending the request to the local parser server, while the last sends the answer to the client after the command request has been executed.

A Read-Complete event can happen only when the appropriate command structure has been received. It consists in a header followed by a variable-length data field, representing the command code and argument. Figure MCTP-14 shows schematically the structure of the data packet. The header contains also a timestamp that will be sent back to the client, in order to correctly associate the server answer to the client request.

A presettable internal timer generates an error if any reading or writing operation could not be executed within a certain amount of time. The error results in a forced disconnection followed by a reconnection attempt by the client (see next section). Upon any disconnection (graceful or forced), the correspondent server state machine is destroyed and removed from the machine list.

Forced disconnection may happen even for other kind of socket errors (for example when loosing the carrier of the acoustic modem during a mission), and are always followed by a reconnection attempt. For example, during the execution of any task, it is possible to unplug and successively re-plug the network cable: the only consequence is the loss of control by the client during the time that the cable is disconnected.

The multi-client feature is necessary to monitor (and eventually correct) the behavior of the arm from a second connection when the system is controlled by the main vehicle CPU (connected through a different client).

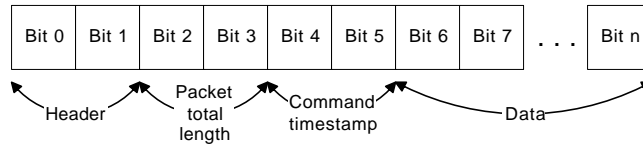


Figure MCTP-14. Packed data format.

Software structure

The kernel of xBus Server is written in Ansi-C language, including its vectorial state machine. This allows easily compiling the source code on a different platform, such as Windows (for the simulation server) or VxWorks (the actual arm controller).

A particular care has been taken in order to simplify the interface with the rest of the software. The use involves the following four functions (for a detailed explanation see the Maris API Reference, not included in this report):

```
int xBusServerInit(void);           // General initialization
int ProcessConnections(void);      // Process connection requests
int SERV_VSM_MainLoop (void);     // Main FSM loop.
void xBusServerTerminate(void);    // Termination
```

In this way, the simplest server code structure is clean and simple as follows:

```
...

#include "xBusServer.h"

// Init
xBusServerInit();

// Main loop
while (!ExitFlag)
{
    // Do process connections
    ProcessConnections();
    // Execute the Finite State Machine list
    SERV_VSM_MainLoop();
}

// Uninitialize (disconnect all clients and de-allocate memory)
xBusServerTerminate();
```

The header file xBusServer.h contains several user-defined settings, as for example the connection port, buffer sizes and timeout for reading and writing operations.

xBus Client

Each of the control stations in Figure MCTP-15 contains one instance of xBus communication client. It can be regarded as the access point for the entire arm control system. The control station #1 can be, for example, the ground control station while the #2 is usually the main vehicle CPU. As noted above, on the server side the parser is shared between all the client connections: in this way, each station can reflect the *actual* status of the parser, allowing the ground station to monitor the activity of the main vehicle CPU (with respect the arm part).

Internally the client is still a finite state machine, described by the following pseudo-code:

Not-Connected:

```
(-- ) Socket-Connected          -> Processing-Commands
      + Process-Requests
(-- ) Connect-Error-Wouldblock  -> Not-Connected
      + Connect-To-Server
(-- ) Connect-Error-Isconn      -> Processing-Commands
      + Process-Requests
(-- ) Connect-Other-Errors      -> Not-Connected
      + Connect-To-Server
(-- ) Init-Terminate-Sequence   -> Closing-Socket
      + Set-Terminate-Flag
      + Shutdown-Both
```

Processing-Commands:

```
(-- ) Ready-To-Process          -> Processing-Commands
      + Process-Requests
(-- ) New-Command-Pending       -> Socket-Writing
      + Init-Timeout
      + Prepare-Write-Buffer
      + Socket-Write
(-- ) Command-Successful        -> Processing-Commands
      + Set-Processing-Successful
      + Process-Requests
(-- ) Init-Terminate-Sequence   -> Closing-Socket
      + Set-Terminate-Flag
      + Shutdown-Send
(-- ) Socket-Errors             -> Closing-Socket
      + Prepare-For-Closing-Socket
      + Shutdown-Both
```

Socket-Writing:

```
(-- ) Write-Complete            -> Socket-Reading
      + Init-Timeout
      + Reset-Read-Buffer
      + Read-Data
(-- ) Write-Incomplete          -> Socket-Writing
      + Socket-Write
(-- ) Error-Ewouldblock         -> Socket-Writing
      + Socket-Write
(-- ) Other-Errors              -> Closing-Socket
      + Shutdown-Both
```

```

(-- ) Write-Timeout                -> Closing-Socket
    + Set-Write-Timeout-Error
    + Shutdown-Both

Socket-Reading:
(-- ) Error-Ewouldblock            -> Socket-Reading
    + Read-Data
(-- ) No-Data-To-Read              -> Socket-Reading
    + Read-Data
(-- ) Read-Complete                 -> Processing-Commands
    + Set-Processing-Successful
    + Process-Requests
(-- ) Read-Incomplete              -> Socket-Reading
    + Read-Data
(-- ) Bad-Header                    -> Socket-Reading
    + Find-New-Header
(-- ) Read-Zero-Bytes               -> Closing-Socket
    + Shutdown-Both
(-- ) Other-Errors                  -> Closing-Socket
    + Shutdown-Both
(-- ) Read-Timeout                  -> Closing-Socket
    + Set-Read-Timeout-Error
    + Shutdown-Both
(-- ) Dummy                          -> Socket-Reading
    + Process-Read-Buffer

Closing-Socket:
(-- ) Shutdown1-Successful          -> Closing-Socket
    + Reset-Read-Buffer
    + Read-Post-Shutdown-Data
(-- ) Shutdown2-Successful          -> After-Close-Socket
    + Do-Close-Socket
(-- ) Read-Zero-Bytes               -> After-Close-Socket
    + Do-Close-Socket
(-- ) Shutdown-Error                -> After-Close-Socket
    + Do-Close-Socket
(-- ) Wsaeprogress1-Error           -> Closing-Socket
    + Shutdown-Send
(-- ) Wsaeprogress2-Error           -> Closing-Socket
    + Shutdown-Both
(-- ) Error-Ewouldblock             -> Closing-Socket
    + Read-Post-Shutdown-Data
(-- ) No-Data-To-Read               -> Closing-Socket
    + Read-Post-Shutdown-Data
(-- ) Read-Incomplete               -> Closing-Socket
    + Read-Post-Shutdown-Data
(-- ) Other-Errors                  -> After-Close-Socket
    + Do-Close-Socket

After-Close-Socket:
(-- ) Ok                            -> Not-Connected
    + Connect-To-Server
(-- ) Error                          ->

```

+ Terminate-The-Program

This is a little more complicated than the server machine. As a matter of fact, the client is responsible of several important issues as auto-reconnection feature (in case of any socket error, timeout, etc.) and processing the command requests from the extern.

The Not-Connected state, for example, executed cyclically the Connect-To-Server action in order to ensure the automatic connection when the client is disconnected. The Processing-Commands state is entered upon a successful connection and here the client is waiting of command execution requests from the extern. These requests can be carried-out by means of the several dedicated functions. For example, the function `xbpcGetQ` for retrieving the arm position or `xbpcDisableAll` which disables all the motors (see the next section). A command request triggers the following cycle:

- 1) Initialize the timeout counter
- 2) Encode data in the write buffer
- 3) Execute a socket-write action
- 4) Wait for server answer
- 5) Check for execution errors

Any error encountered before the actions 3 and 4 would result in a forced disconnection followed by a reconnection attempt.

xBus Client software structure

Likewise the server component, the kernel of xBus Client is written in Ansi-C language. This still allows easily compiling the source code on a different platform, such as Windows (for the simulation server) or VxWorks (inside the main vehicle CPU).

An expandable subset of interface functions is available for executing each of the arm commands. These functions are:

```
int xbpcGetQ(float* q);
    Retrieve the arm configuration vector (joint angles and gripper
    position)

int xbpcGetJr3Data(float* fm);
    Returns the force-moment data from the JR3 sensor

int xbpcExchangeArmData(sArmData* pArmData);
    Send and receive in one-shot all the arm sensor data, including
    the motor enable status, motor health and the teleoperation
    input data

int xbpcDisableAll();
    Disable all the motors

int xbpcEnableAll();
    Enable all the motors
```

```

int    xbpLoadPathPoints(float    Points[][PATH_POINT_SIZE],    int
nPoints);
    Load a path sequence points

int xbpZeroLengthCmd(short Cmd);
    Executes the command coded in cmd such as, for example:
    XBPC_CALIBRATE_ENCODER
    XBPC_CALIBRATE_JR3
    XBPC_SET_RESOLVER_MODE
    XBPC_SET_ENCODER_MODE
    XBPC_PARK
    XBPC_STOP

```

For a detailed explanation see the Maris API Reference, not included in this report. Currently under development, a special command will allow to execute a script in a predefined language. This is particularly useful for automating particular task and represents a key point in the future expansion of the system.

xBus Parser

The ability of interpreting and executing different kind of command is a feature of the *xBus Parser* block of Figure MCTP-15. Likewise the server and clients part, its kernel is still based on a Finite State Machine. This allows us to execute both “single-shot” commands (like `xbpcEnableAll` for enabling the motors) and command that require internal states (like `XBPC_START_PATH` for tracking a sequence of point in the task space).

The internal pseudo-code description of the parser FSM currently is:

Idle:

```

(--) Parser-Ready          -> Idle
    + Set-Parser-Ready
(--) Cmd-Park              -> Parking
    + Do-Stop-All-Motors
(--) Cmd-Set-Teleop-Mode   -> Teleoperated-Mode
    + Start-Teleop-Mode
(--) Cmd-Start-Tracking-Mode -> Tracking-Mode
    + Init-Tracking-Mode
    + Go-To-Next-Point

```

Tracking-Mode:

```

(--) Tracking-Incomplete  -> Tracking-Mode
    + Track-Current-Point
(--) Tracking-Complete    -> Idle
    + Set-Parser-Ready

```

Teleoperated-Mode:

```

(--) Teleop-Mode-Ok       -> Teleoperated-Mode
    + Teleoperate

```

Parking:

```

(--) Stop-Incomplete      -> Parking
    + Check-For-Stop-Completion

```

```

(-- ) Stop-Complete          -> Parking
    + Resume-The-Motors
    + Park
(-- ) Parking-Incomplete    -> Parking
    + Park
(-- ) Parking-Complete      -> Idle
    + Set-Parser-Ready

```

Defaults:

```

(-- ) Cmd-Stop              -> Stopping-The-Motors
    + Do-Stop-All-Motors

```

Stopping-The-Motors:

```

(-- ) Stop-Incomplete       -> Stopping-The-Motors
    + Check-For-Stop-Completion
(-- ) Stop-Complete         -> Idle
    + Set-Parser-Ready

```

Each state corresponds to a dedicated command. Note that the overall structure allows an easy expansion for further command implementation.

The simulation server

The modular structure of Figure MCTP-15 has an important consequence. Substituting the arm system (hardware and manipulator) with an equivalent mathematical model, we can transparently switch between the real system and its simulated model, without affecting the structure of each control station. This is done selecting, in the client side, the appropriate IP address of the server. Figure MCTP-15 shows the block diagram of this concept.

In our system, the arm model has been implemented via Simulink (The Mathworks, Inc). xBus Server and Parser's code have been compiled as well within a c-mex s-function and thus embedded in a custom Simulink block. This process allows testing and simulating every aspect of the control system before running it on the actual arm. The results we obtained in the simulation environment were quite close to those belonging to the actual arm.

Trajectory planner

As seen in the 0 paragraph, one of the operation modes of xParser is “*Tracking*”, which allows to drive the end-effector through a predefined path.

The path is defined by a succession of points X_i , each of one representing a generalized position in the task space:

$$X_{ei} = [x_i \quad y_i \quad z_i \quad roll_i \quad pitch_i \quad yaw_i]^T \quad (12.3)$$

The rotation parameters indicate the orientation of the end-effector with respect the main frame. The trajectory between X_i and X_{i+1} has been planned preserving the continuity of the task velocity, in order to avoid excessive stress to the arm structure. **Error! Reference source not found.** illustrates a one-dimensional velocity profile that accomplishes the above requirements.

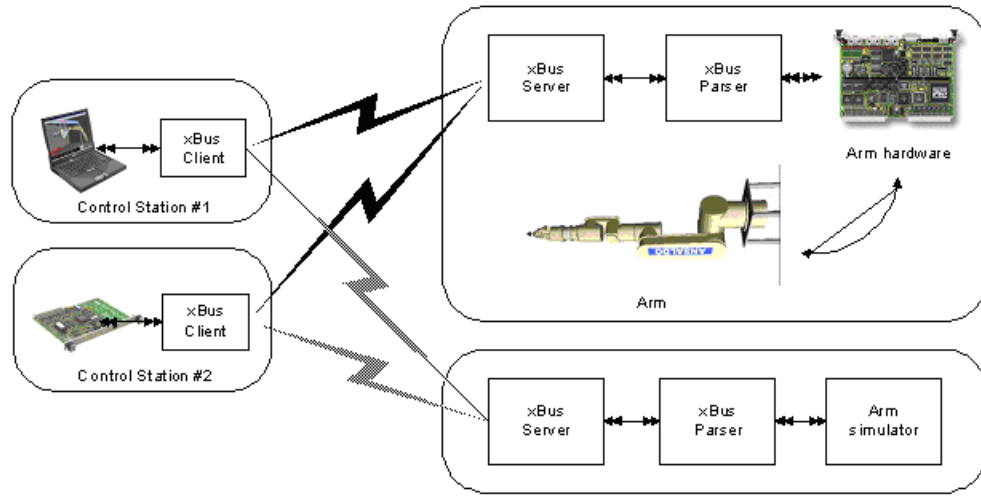


Figure MCTP-15. Simulation server concept

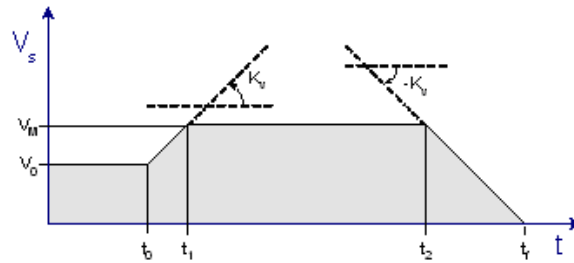


Figure MCTP-16. Trajectory generator: trapezoidal velocity profile.

Given the input data for the profile as s_0 (the initial position), s_f (final position), V_{s0} (initial velocity), V_m (maximum velocity), K_v (acceleration) and t_0 , the output trajectory is:

$$s(t) = \begin{cases} s_0 & t = t_0 \\ s_0 + \int_{t_0}^t V_0 + K_v \cdot (t - t_0) dt & t_0 \leq t < t_1 \\ s_1 + \int_{t_1}^t V_m dt & t_1 \leq t < t_2 \\ s_2 + \int_{t_2}^t V_m - K_v(t - t_2) dt & t_2 \leq t < t_f \\ s_f & t \geq t_f \end{cases} \quad (12.4)$$

where:

$$s_1 = s_0 + \int_{t_0}^{t_1} V_0 + K_v \cdot (t - t_0) dt \quad (12.5)$$

$$s_2 = s_1 + \int_{t_1}^{t_2} V_m dt \quad (12.6)$$

$$s_f = s_2 + \int_{t_2}^{t_f} V_m - K_v(t - t_2) dt \quad (12.7)$$

Expanding above integrals we have:

$$s(t) = \begin{cases} s_0 & t = t_0 \\ s_0 + \frac{1}{2} K_v \cdot (t^2 - t_0^2) + V_0 \cdot (t - t_0) - K_v t_0 \cdot (t - t_0) & t_0 \leq t < t_1 \\ s_0 + \frac{1}{2} K_v \cdot (t^2 - t_0^2) + V_0 \cdot (t_1 - t_0) - K_v t_0 \cdot (t_1 - t_0) + V_m \cdot (t - t_1) & t_1 \leq t < t_2 \\ s_0 + \frac{1}{2} K_v \cdot (t_1^2 - t_0^2) + V_0 \cdot (t_1 - t_0) - K_v t_0 \cdot (t_1 - t_0) + V_m \cdot (t_2 - t_1) - \\ - \frac{1}{2} K_v \cdot (t^2 - t_2^2) + V_m \cdot (t - t_2) + K_v t_2 \cdot (t - t_2) & t_2 \leq t < t_f \\ s_f & t \geq t_f \end{cases} \quad (12.8)$$

where:

$$t_1 = t_0 + \frac{V_m - V_0}{K_v} \quad (12.9)$$

$$t_2 = t_f - \frac{V_m}{K_v} \quad (12.10)$$

$$t_f = \frac{2s_f K_v - 2s_0 K_v + 2t_0 K_v V_m + 2V_m^2 - 2V_m V_0 + V_0^2}{2K_v V_m} \quad (12.11)$$

With a particular set of input parameters, it is possible that $t_2 \leq t_1$. In this case, we cannot use the trapezoidal shape for the velocity profile. Instead, a triangular profile is suitable (**Error! Reference source not found.**), because the above condition ensure that maximum velocity will be smaller than V_m .

In this case we have:

$$V_{se} = \frac{(t_f - t_0) K_v + V_{s0}}{2} \quad (12.12)$$

$$t_e = \frac{t_f + t_0}{2} - \frac{V_{s0}}{2K_v} \quad (12.13)$$

$$t_f = \frac{-V_{s0} + t_0 K_v + \sqrt{2V_{s0} + 4s_f K_v - 4s_0 K_v}}{K_v} \quad (12.14)$$

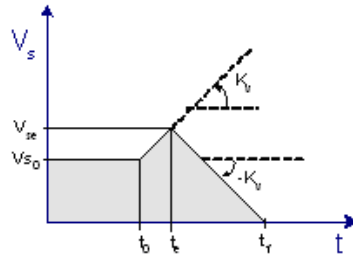


Figure MCTP-17. Trajectory generator: triangular velocity profile.

Figure MCTP-18 and Figure MCTP-19 show a numerical example, for the one-dimensional case.

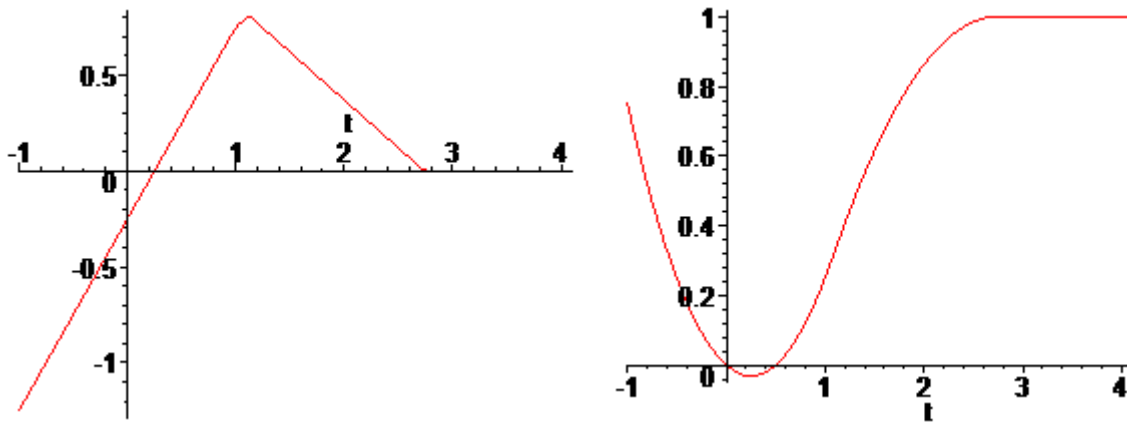


Figure MCTP-18. Trajectory generator example: triangular case with $t_0 = 1$,

$$s_0 = 0.25, s_f = 1, V_{s_0} = 0.75 \text{ and } K_v = 0.5.$$

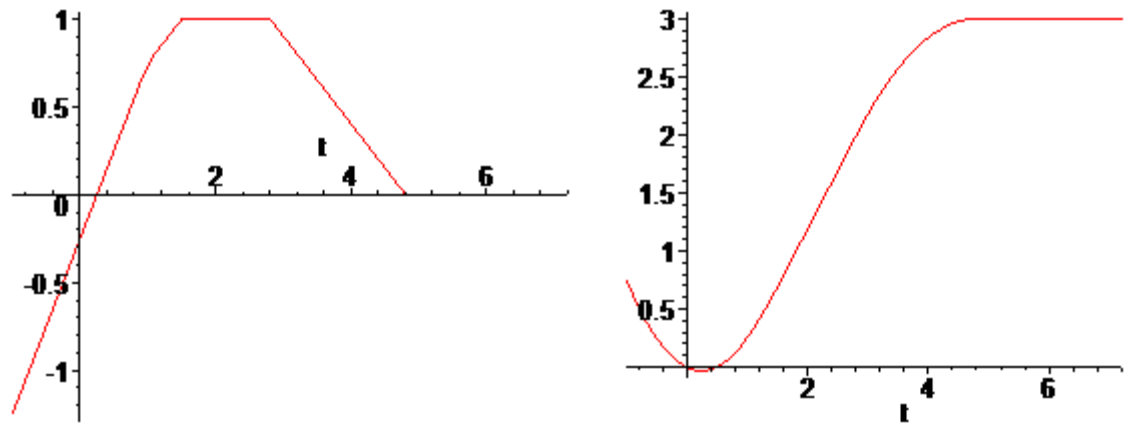


Figure MCTP-19. Trajectory generator example: trapezoidal case with $t_0 = 1$,

$$s_0 = 0.25, s_f = 3, V_{s_0} = 0.75 \text{ and } K_v = 0.5.$$

The extension to the multidimensional vector (12.3) is done considering the following considerations.

Let R_0 be the initial rotation matrix of the end-effector with respect the main frame. Likewise, let R_f be the final posture. It is possible to find a direction r_i around which R_0 must be rotated of \mathbf{q} radians in order to obtain R_f . Thus we can define the four element vector

$$AB_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 0 \end{bmatrix}, \quad AB_f = \begin{bmatrix} x_f \\ y_f \\ z_f \\ \mathbf{q} \end{bmatrix}, \quad (12.15)$$

and the parameter:

$$s_f = |AB_f - AB_0| \quad (12.16)$$

obtaining:

$$AB(t) = AB_0 + (AB_f - AB_0) \frac{s(t)}{s_f} \quad (12.17)$$

where $s(t)$ is obtained integrating the trapezoidal or triangular profile as above. The rotation matrix $R(t)$ can be computed from the fourth component of $AB(t)$, say $\mathbf{q}(t)$, by rotating R_0 around r_i of $\mathbf{q}(t)$ radians.

Graphic interface and Multimedia Development Environment (MarisGL)

MarisGL is a complete station monitor and a development environment for the manipulator. By integrating an internal link with Tornado, MarisGL automates all the operation necessary for downloading and running the code on the FORCE CPU board. With the introduction of the vehicle frame graphic model, it allows to simulate the natural environment of the arm and to test the collision avoidance algorithm.

Most of the work dedicated to its realization involved a massive code development, and its description is beyond the scope of this report. Hence, only some of the main aspect are introduced.

Software architecture

MarisGL is a Win32-based application (not portable to other operating systems), developed under Microsoft Visual C++ 6 with the aid of the Microsoft Foundation Classes (MFC) framework and the OpenGL graphic library. Figure MCTP-20 shows its internal block structure: a summarized description of each block follows.

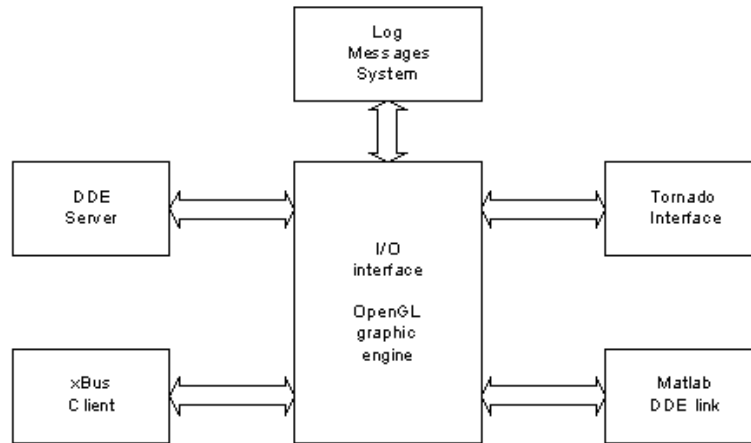


Figure MCTP-20. **MarisGL application basic structure.**

I/O Interface (OpenGL Graphic Engine)

This is the kernel of the application, responsible of all the user interactions with Maris data.

The output part consists in an OpenGL reconstruction of the arm, together with its environment. The last can be either the simulation room (arm alone) or the SAUVIM vehicle (useful for visualizing the arm-vehicle interactions, see Figure MCTP-21 or the movie clip). The OpenGL reconstruction of the vehicle has been done using its Autocad® as source, preserving in this way its original dimension.

The lower side of the animation window guests some virtual instruments showing the information coming from the arm sensors (see Figure MCTP-22 for more details).

The input section contains a Microsoft DirectX® driver for mouse and keyboard, as well a support for external position control devices (such as joysticks). Using the keyboard as input device, the entire control of Maris 7080 manipulator can be totally performed on a notebook, eventually equipped with a WAN device for a completely wireless driving solution (that is the actual configuration in our laboratory).

DDE Server

Most of the MarisGL commands are externally accessible via DDE link. This block is responsible of instantiating a DDE server that will process this kind of external commands.

In the general case, a DDE server uses a three-level hierarchy - *service name* (also called *application name*), *topic name*, and *item name* - to uniquely identify a unit of data the server can exchange during a conversation.



Figure MCTP-21. **OpenGL reconstruction of part of the vehicle and the manipulator.**

For the establishment of the DDE link with MarisGL, the above identifiers are:

Service name: MarisGL
Topic name: Update
Item name: StringMode

A client can use the XTYP_EXECUTE transaction to cause MarisGL DDE server to execute a command of the following set:

DrawScene	Perform a refresh of the entire OpenGL scene
DrawIntroScene	Starts an introduction demo
Maximize	Maximizes the application window
Minimize	Iconizes the application window
Restore	Restores the application windows to its size
ShowCollisionTarget	Shows the box-shaped obstacle (for collision avoidance test purpose)
ShowVehicle	Go to the vehicle visualization mode
ShowSimRoom	Go to the Simulation room mode (arm alone)
BringToFront	Bring MarisGL on the top of other windows
RunTarget	Run the controller on the specified target

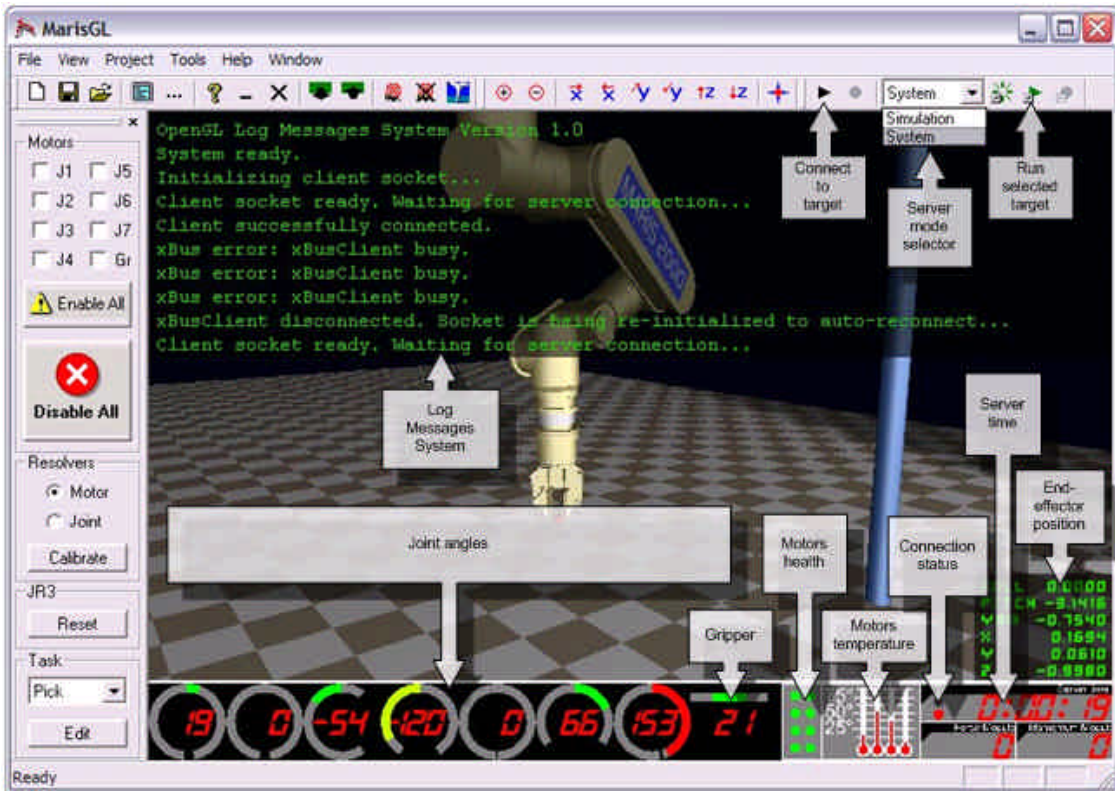


Figure MCTP-22. MarisGL main window: sensors monitor location.

xBus Client

This is the communication client with the actual arm. For more details see the xBus section.

Tornado link

Tornado®, by WindRiver, is a stand-alone tool for developing VxWorks-based applications. MarisGL incorporates a link with Tornado in order to establish a link with the target machine, thus downloading the compiled VxWorks application. This is done using the *Target Server* and *Wind Shell* command line tools included in Tornado.

Matlab link

Some functionalities of MarisGL require the use of Matlab®. This happens, for example, when running the target in simulation mode, as the simulation server is a Simulink® model. The Matlab link is a DDE client able to open a communication channel with Matlab and thus exchanging commands or data.

Log Messages System

Most of the activities of both MarisGL and the actual arm system are monitored by the application kernel, which delivers their status to the user through the *GLLog Messages System* (Figure MCTP-22). It is an OpenGL-based text and/or vocal (using Microsoft Speech SDK) interface useful to retrieve extended information (such as connection information, command execution results, etc.).

Working with projects

One of the capabilities of MarisGL is to work with different project configuration. A project contains information about the target IP address, the VxWorks executable, the Simulink model of the simulation server and its sample time (for using with the new VxWorks task scheduler). This allows switching between different demos simply changing the project.

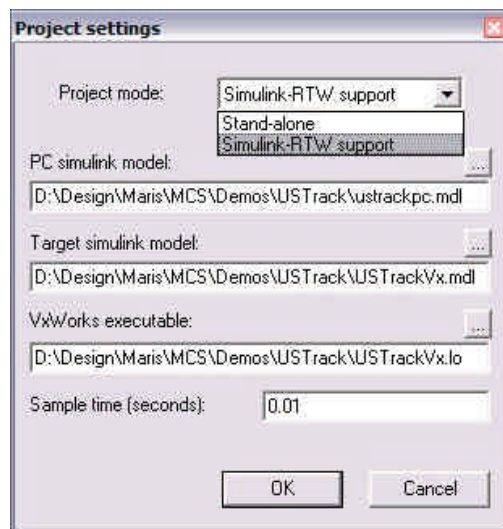


Figure MCTP-23. **Project configuration dialog.**

Low-Level Control (LLC)

Project Leader(s):	Dr. Hyun Taek Choi
Personnel:	Mr. Side Zhao & Mr. Ove Stapnes
Past Project Leader(s):	Dr. Junku Yuh, Dr. Song K. Choi & Dr. Tae Won Kim
Past Personnel:	Ms. Jing Nie, Mr. Eric Kardash & Mr. Michael West

Objectives

- To design an advanced vehicle control for navigation and hovering; and
- To design an advanced coordinated motion/force control of the vehicle and manipulator during the intervention mode.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

- Development of model-based robust control for positioning control;
- Development of Adaptive backstepping control for Heave control;
- Development of ODIN-III AUV to test control algorithm;
- Extensive simulation and testing on ODIN-III AUV

Structure of low-level control

The Autonomous Underwater Vehicle (AUV) system is a complex nonlinear time-varying multivariable system in the presence of strong external disturbances. In order for the AUV to implement underwater missions, the AUV's control system must have the robustness not only to its own nonlinearity, but also to the environmental uncertainties such as the sea current and the reaction force from the manipulator intervention.

Several advanced control algorithms have been proposed to control the underwater vehicle. Among them are sliding mode control [Cunha95, Healey93, Dougherty90], neural network based control [Ishii94], Fuzzy logic control [DeBitetto94], hybrid adaptive control [Tabaii94] and non-regressor-based adaptive control [Yuh00]. These algorithms showed good performance. However, this doesn't mean that these algorithms work well under every situation. In some situations, the fact that a control algorithm has advantage might be disadvantage in other situations. For example, disturbance rejection algorithm will be good to keep the position but it will not be good to move from point to point because it will spend limited energy on unnecessary action. Therefore, for multi-purpose AUV, different types of control algorithms should be considered according to different situations.

Mode Switching Control scheme

Each situations of being AUV can be defined as a mode, for example positioning, point to point moving. Also the control objective and control specification can be defined for each mode. From this procedure, we can choose and design a controller for each mode. After that, we can switch controller for each mode to get the best performance during whole operation time of vehicle. Here, we defined three modes for basic operation, those are positioning, tracking, and point-to-point moving, and we'll be able to add another mode for another purpose of operation.

In this report, we considered that adaptive disturbance observer as a disturbance rejection control scheme for positioning mode, and adaptive backstepping control as a tracking control for heave control.

Disturbance Rejection Control

The design of a high performance position control system for underwater vehicle is not an easy task. For this, controller should be able to cope with various internal/external disturbances. Some disturbance rejection control schemes were proposed as a kind of robust control. One of them is disturbance observer.

Disturbance Observer

Disturbance observer (DOB) proposed by Onishi has strong disturbance rejection property. By estimating the modeling error and the disturbance and then feeding back these into the system input, DOB makes the whole system behave as a nominal system. Because DOB regards the difference between the output of actual system and the output of model as equivalent disturbance applied to the nominal model, its structure and design method is simple compared to other robust control scheme as shown in Figure LLC-1.

Therefore, DOB has been widely used to many industrial applications. However, to get nominal model of underwater vehicle is difficult. So, we choose a mathematical desired model as a nominal model. And we're going to find an experimental desired model.

Robust Internal-loop Compensator

Even though DOB has advantages, it has important design constraint because it uses the inverse nominal model. Therefore, it must involve a low-pass filter to make the inverse model to be implementable. Based on general two-loop structure for model-based robust control, robust internal-loop compensator (RIC) was proposed as shown in Figure LLC-2. It doesn't use inverse model and doesn't have any design constraint. It will be an important point considering the noisy environment of underwater vehicle like sonar sensor. Using this design flexibility, we can design controller to get rid of noise getting from each sensor. And we can add some adjusting algorithm to compensate nominal model. Because DOB and RIC generate the control action to eliminate the difference between actual system and nominal model regardless control reference, a big difference causes wasting of energy. So, it will be helpful to save energy

Adaptive Control

Lots of researchers have resorted to adaptive controllers to cope with the parameter uncertainties of the AUV system. These schemes can be roughly divided into two categories, regressor based and non-regressor based. The former category schemes require an a priori knowledge of the system to be controlled. It assumes that the order of the system is known. When the complexity increases and the system is difficult or impossible to be modeled, the performance will deteriorate significantly. Moreover, the computation expense for parameter estimation by the regressor-based adaptive control approach will increase as the number of unknown system parameters increases. Therefore lots of researchers have been studying non-regressor based adaptive control schemes, among which Yuh and Choi proposed one scheme that uses the bound estimation method, which estimates the bounds of the parameter matrices of the robot dynamic system or their combinations, and then these estimates are

used to adjust control gains. This scheme does not need any parametric knowledge of the system, and its stability has been proved using Lyapunov method.

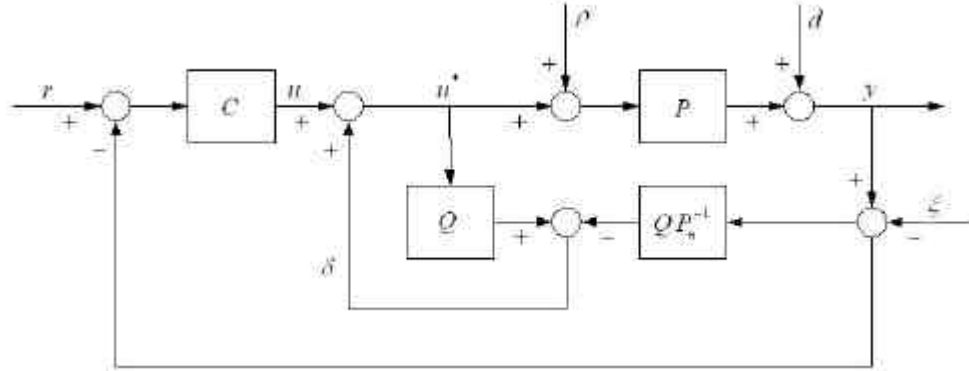


Figure LLC-1 Diagram of DOB based Control System

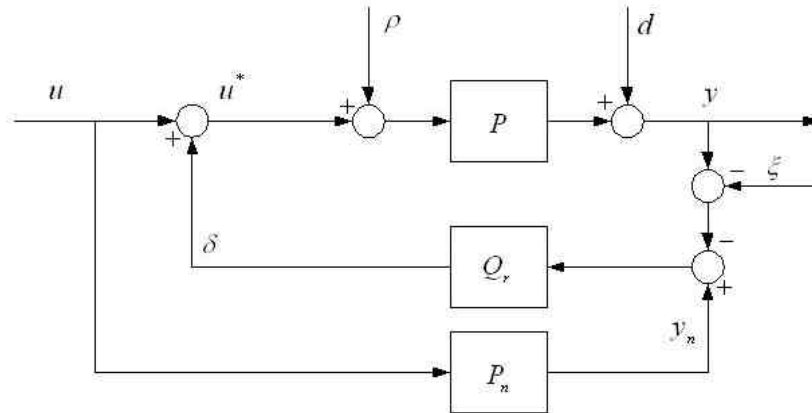


Figure LLC-2 Robust Internal-Loop Compensator

Adaptive Control based on Disturbance Observer

The scheme proposed in this report is an extension of the adaptive control system developed by Yuh and Choi. The disturbance observer is used to simplify the nonlinear underwater vehicle system with uncertainties into a simple model with residual disturbance error due to time delay of the low-pass filter. Based on this simplified model, the adaptive control system is designed. By introducing the disturbance observer, the adaptive controller becomes much simpler than the previous ones. At the same time, the adaptive controller adapts to any changes in performance due to disturbance observance error resulted from the time delay of the low-pass filter in constructing the disturbance observer and provides better performance than other linear control approaches with the disturbance observer.

The proposed controller has a very simple structure and is computationally efficient. The only information required to implement this scheme is the number of degrees-of-freedom and actuator inputs of the system. The adaptive control law estimates parameters defined by combinations of bounded constants of the parameter matrices of the nominal model with disturbance estimation error, instead of each unknown parameter of the actual system model. No computation for updating the robot dynamic model is needed. Therefore, it is easy to implement the control scheme to a class of

dynamic systems. The tracking errors can asymptotically converge to zero by the Lyapunov method. Effectiveness of the presented control system was investigated by computer simulation with Omni-Directional Intelligent Navigator (ODIN).

Algorithm description

The vehicle dynamics can be described as follows:

$$M\dot{v} + C(v)v + D(v)v + g(\mathbf{h}) = \mathbf{t} \quad (\text{LLC-1})$$

$$\dot{\mathbf{h}} = J(\mathbf{h})v \quad (\text{LLC-2})$$

where $M = M_{RB} + M_A$, $C(v) = C_{RB}(v) + C_A(v)$, M_{RB} is the rigid-body inertia matrix, M_A is the added inertia matrix; $C_{RB}(v)$ is the rigid-body Coriolis and centripetal matrix, $C_A(v)$ is the hydrodynamic Coriolis and centripetal matrix, $D(v)$ is the hydrodynamic damping and lift matrix, $g(\mathbf{h})$ is the gravitational forces and moments vector, \mathbf{t} is the control inputs vector, v is the velocity vector in the vehicle-fixed frame, \mathbf{h} is the earth-fixed displacement vector and J is the transformation matrix between the vehicle-fixed frame and the earth-fixed frame.

The dynamics above can also be written in earth-fixed frame as follows:

$$M_h(\mathbf{h})\dot{\mathbf{h}} + C_h(v, \mathbf{h})\mathbf{h} + D_h(v, \mathbf{h}) + g_h(\mathbf{h}) = \mathbf{t}_h \quad (\text{LLC-3})$$

where $M_h(\mathbf{h}) = J^{-T}(\mathbf{h})M_v J^{-1}(\mathbf{h})$, $C_h(v, \mathbf{h}) = J^{-T}(\mathbf{h})[C(v) - MJ^{-1}(\mathbf{h})j(\mathbf{h})]J^{-1}(\mathbf{h})$, $\mathbf{t}_h = J^{-T}(\mathbf{h})\mathbf{t}_v$, $D_h(v, \mathbf{h}) = J^{-T}(\mathbf{h})D(v)J^{-1}(\mathbf{h})$, $g_h(\mathbf{h}) = J^{-T}(\mathbf{h})g(\mathbf{h})$.

Figure LLC-3 shows the overall control system that has the inner loop compensator of the disturbance observer inside the dotted square and the outer loop of the adaptive controller. The system with the disturbance observer inside the dotted box becomes a simple nominal model with disturbance estimation error, which will be controlled by the adaptive controller.

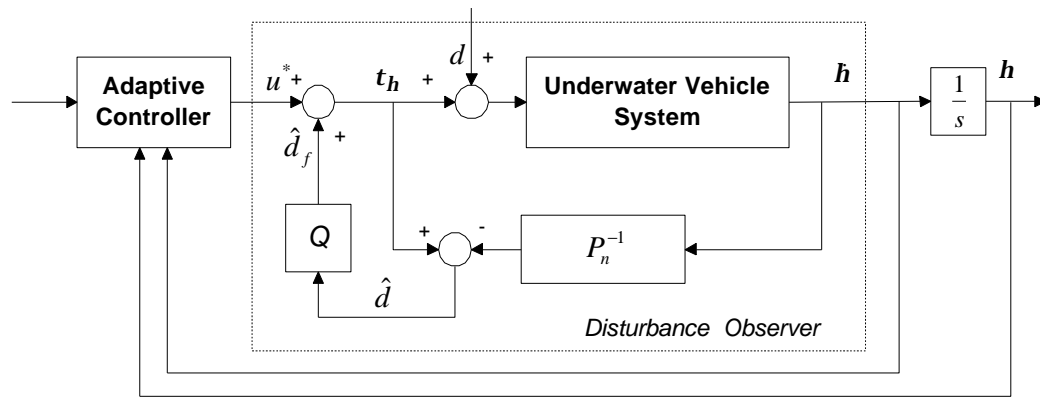


Figure LLC-3 Overall control system

Disturbance Observer

As shown in Figure LLC-3, the control input \mathbf{t}_h is computed by

$$\mathbf{t}_h = u^* + \hat{d}_f \quad (\text{LLC-4})$$

where u^* is the output of the adaptive controller and \hat{d}_f is the filtered estimate of \hat{d} :

$$\hat{d}_f = Q\hat{d} \quad (\text{LLC-5})$$

where Q is a low-pass filter, and \hat{d} is the estimate of the equivalent disturbance:

$$\hat{d} = \mathbf{t}_h - P_n^{-1}\mathbf{h} \quad (\text{LLC-6})$$

where P_n represents a nominal model of the underwater vehicle system whose output is \mathbf{h} .

If Q is an identity matrix, from Equations (LLC-4 - LLC-6), the system inside the dotted line becomes the nominal model:

$$\mathbf{h} = P_n u^* \quad (\text{LLC-7})$$

However, the disturbance observer cannot be implemented with $Q = I$ since P_n^{-1} is not realizable by itself. Therefore, the relative degree of Q must be equal or greater than that of P_n .

For $Q \neq I$, the filtered estimate can be expressed by

$$\hat{d}_f = \hat{d} + \Delta\hat{d} \quad (\text{LLC-8})$$

where $\Delta\hat{d}$ is the difference between \hat{d}_f and \hat{d} , due to the time delay of the low-pass filter.

Therefore, from Equations (LLC-4, LLC-6, and LLC-8),

$$\mathbf{h} = s^{-1}P_n(u^* + \Delta\hat{d}) \quad (\text{LLC-9})$$

Referring to Equation (LLC-3), if one chooses $P_n^{-1} = sM_n$, where M_n is a constant nominal inertial matrix, the dotted block in Figure LLC-3 can be represented by the following simple model:

$$M_n \mathbf{h} + h = u^* \quad (\text{LLC-10})$$

where $h = -\Delta\hat{d}$.

Adaptive Controller

We now design an adaptive controller for the nominal model of (LLC-10) with the tracking error vector e defined as

$$e = \mathbf{h}_d - \mathbf{h} \quad (\text{LLC-11})$$

where \mathbf{h}_d is a desired value of \mathbf{h} .

The parameter matrices of the nominal model are assumed to be bounded as

$$\|M_n^{-1}\| \leq \mathbf{a}, \quad \|h\| \leq \mathbf{b}_1 / \mathbf{k}, \quad \mathbf{I}_{\min}(M^{-1}) > \mathbf{g}, \quad (\text{LLC-12})$$

where $\|\cdot\|$ represents Euclidean norm, and \mathbf{a} , \mathbf{b}_1 , \mathbf{k} and \mathbf{g} are positive constants.

Instead of mathematically proving Equation (LLC-11), we will present how to estimate new parameters defined by

$$\mathbf{q}_i = \frac{\mathbf{a}\mathbf{b}_i}{\mathbf{g}}, \quad i=1,2,3, \quad (\text{LLC-13})$$

where $\mathbf{b}_2 = \mathbf{b}_3 = \frac{\mathbf{c}}{\mathbf{a}}$ and $\mathbf{c} \geq 1$ is a constant.

Consider the following control law

$$u^* = K_0 + K_1 \mathbf{k} + K_2 \dot{e} + K_3 e = \sum_{i=0}^3 K_i \Phi_i, \quad (\text{LLC-14})$$

where $\Phi_0 = \mathbf{h}_d$, $\Phi_1 = \mathbf{k}$, $\Phi_2 = \dot{e}$, $\Phi_3 = e$, and K_i are control gain matrices with $K_0 = M_n$. From Equations (LLC-10) and (LLC-14), the error equation can be obtained as follows:

$$\ddot{e} = M_n^{-1} \{(h/\mathbf{k} - K_1)\mathbf{k} - K_2 \dot{e} - K_3 e\} = M_n^{-1} \sum_{i=1}^3 (P_i - K_i) \Phi_i \quad (\text{LLC-15})$$

where $P_1 = h/\mathbf{k}$, $P_2 = P_3 = 0$.

Theorem: The tracking error e asymptotically converge to zero and the estimate of the parameters converge to a certain bounds with the following adaptive controller:

$$K_i = \frac{\hat{\mathbf{q}}_i \tilde{e} \Phi_i^T}{\|\tilde{e}\| \|\Phi_i\|} \quad i = 1, 2, 3 \quad (\text{LLC-16})$$

$$\dot{\hat{\mathbf{q}}}_i = f_i \|\tilde{e}\| \|\Phi_i\| \quad (\text{LLC-17})$$

where f_i are positive constants, $\hat{\mathbf{q}}_i$ are estimates of \mathbf{q}_i , and

$$\tilde{e} = \dot{e} + \mathbf{s} e, \quad (\text{LLC-18})$$

where \mathbf{s} is a positive constant satisfying $\mathbf{s} \leq \mathbf{c}$.

Proof: Consider the following Lyapunov function candidate:

$$V = \frac{1}{2} \tilde{e}^T \tilde{e} + \frac{1}{2} e^T e + \frac{1}{2} \sum_{i=1}^3 f_i^{-1} \mathbf{g}(\mathbf{q}_i - \hat{\mathbf{q}}_i)^2 \quad (\text{LLC-19})$$

Differentiating Equation (LLC-19) along Equation (LLC-15) with respect to time yields

$$\begin{aligned} \dot{V} &= \tilde{e}^T \ddot{e} + \mathbf{s} \tilde{e}^T \dot{e} + e^T \dot{e} - \sum_{i=1}^3 f_i^{-1} \mathbf{g}(\mathbf{q}_i - \hat{\mathbf{q}}_i) \dot{\hat{\mathbf{q}}}_i \\ &= \left[\tilde{e}^T \left(M_n^{-1} \sum_{i=1}^3 P_i \Phi_i \right) + \mathbf{s} \tilde{e}^T \dot{e} + e^T \tilde{e} - \mathbf{s} e^T e - \sum_{i=1}^3 f_i^{-1} \mathbf{g} \dot{\hat{\mathbf{q}}}_i \right] + \left[-\tilde{e}^T M_n^{-1} \sum_{i=1}^3 K_i \Phi_i + \sum_{i=1}^3 f_i^{-1} \mathbf{g} \dot{\hat{\mathbf{q}}}_i \right] \end{aligned} \quad (\text{LLC-20})$$

With the adaptive controller Equations (LLC-16), (LLC-17) and $\mathbf{s} \leq \mathbf{c}$, and $\mathbf{c} \geq 1$ the equation in the first bracket of Equation (LLC-20) becomes

$$\begin{aligned} &\tilde{e}^T \left(M_n^{-1} \sum_{i=1}^3 P_i \Phi_i \right) + \mathbf{s} \tilde{e}^T \dot{e} + e^T \tilde{e} - \mathbf{s} e^T e - \sum_{i=1}^3 f_i^{-1} \mathbf{g} \dot{\hat{\mathbf{q}}}_i \\ &= \tilde{e}^T M_n^{-1} P_1 \Phi_1 - \mathbf{a} \mathbf{b}_1 \|\tilde{e}\| \|\Phi_1\| + \mathbf{s} \tilde{e}^T \dot{e} - \mathbf{c} (\|\tilde{e}\| \|\dot{e}\| + \|\tilde{e}\| \|e\|) + e^T \tilde{e} - \mathbf{s} e^T e \\ &\leq \left(\|M_n^{-1}\| \|P_1\| - \mathbf{a} \mathbf{b}_1 \right) \|\tilde{e}\| \|\Phi_1\| + (\mathbf{s} - \mathbf{c}) \|\tilde{e}\| \|\dot{e}\| - (\mathbf{c} - 1) \|\tilde{e}\| \|e\| - \mathbf{s} e^T e \\ &\leq -\mathbf{s} e^T e \end{aligned} \quad (\text{LLC-21})$$

and the equation in the second bracket becomes

$$\begin{aligned}
& -\tilde{e}^T M_n^{-1} \sum_{i=1}^3 K_i \Phi_i + \sum_{i=1}^3 f_i^{-1} \hat{\mathbf{g}} \hat{\mathbf{q}}_i \\
& = \sum_{i=1}^3 \left(-\frac{\tilde{e}^T M_n^{-1} \tilde{e}}{\tilde{e}^T \tilde{e}} + \mathbf{g} \right) \|\tilde{e}\| \|\Phi_i\| \hat{\mathbf{q}}_i \\
& \leq \sum_{i=1}^3 \left(-\mathbf{1}_{\min}(M_n^{-1}) + \mathbf{g} \right) \|\tilde{e}\| \|\Phi_i\| \hat{\mathbf{q}}_i \leq 0
\end{aligned} \tag{LLC-22}$$

From Equations (LLC-20), (LLC-21), and (LLC-22), \dot{V} is reduced to

$$\dot{V} \leq -\mathbf{s} e^T e \tag{LLC-23}$$

which is negative for all $e \neq 0$. Therefore, the tracking error will asymptotically go to zero.

It is noted from Equation (LLC-17) that estimates of parameters increase but never decrease as time goes by while the control gains from Equation (LLC-16) may vary in both directions. It may result in overestimates of the parameters that are the combinations of the bounded constants of system parameter matrices as defined by Equation (LLC-13). Overestimation will not cause stability problem but may cause degraded performance of the overall control system. To solve this problem, the following modification is adopted:

$$\hat{\mathbf{q}}_i = \begin{cases} f_i \|\tilde{e}\| \|\Phi_i\| & \text{if } \|\tilde{e}\| \|\Phi_i\| \geq \mathbf{m}_i \\ -\mathbf{e} \hat{\mathbf{q}}_i + f_i \|\tilde{e}\| \|\Phi_i\| & \text{if } \|\tilde{e}\| \|\Phi_i\| < \mathbf{m}_i \end{cases} \tag{LLC-24}$$

It is also noted that the direct use of the controller of Equation (LLC-15) would generate large control input signals and extreme chattering phenomena at near zero value of denominator. To avoid this problem, the following controller is used instead of Equation (LLC-16):

$$K_i = \begin{cases} \frac{\hat{\mathbf{q}}_i \tilde{e} \Phi_i^T}{\|\tilde{e}\| \|\Phi_i\|} & \text{if } \|\tilde{e}\| \|\Phi_i\| > \mathbf{d}_i \\ \frac{\hat{\mathbf{q}}_i \tilde{e} \Phi_i^T}{\mathbf{d}_i} & \text{if } \|\tilde{e}\| \|\Phi_i\| \leq \mathbf{d}_i \end{cases} \tag{LLC-25}$$

where $i=1,2,3$ and \mathbf{d}_i is a positive constant. The control gain described by Equation (LLC-25) may not guarantee the asymptotic stability but tracking errors are bounded by small numbers depending on \mathbf{d}_i .

Simulation and its result

ODIN is a 6-degree-of-freedom autonomous underwater robot developed by Autonomous System Laboratory, University of Hawaii. It is a closed-framed sphere shaped vehicle, which makes its dynamics in each direction identical. It has 8 thrusters, 4 horizontal and 4 vertical, which make it capable of maneuvering with 6 degrees-of-freedom. Two control schemes are tested. One is PID + DOB and the other is Adaptive DOB. They have exactly the same DOB. To test the disturbance rejection performance of the DOB, the following disturbances are added as external disturbance.

$$d_n(\mathbf{h}) = D * (\sin(14\mathbf{p}t) + \sin(38\mathbf{p}t)) * [1 \ 1 \ 0 \ 0 \ 0 \ 0]^T \tag{LLC-26}$$

Here two prime frequencies of 7Hz and 19Hz are selected to make a periodical signal with a long period. Figure LLC-4 shows the time-amplitude graph of the disturbance with $D = 200$.

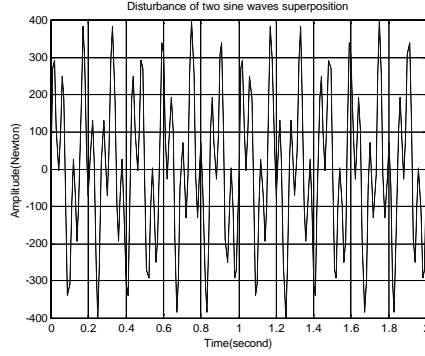


Figure LLC-4 Disturbance of Two Sine Waves Superposition

The trajectory of the vehicle is the straight lines connecting the following four points consequently: (2.5, 2.5, 0.27), (2.5, 2.5, 2.0), (2.5, 4.5, 2.0) and (4.5, 4.5, 2.0). In each phase, trapezoidal acceleration plan is used.

To show the tracking and regulating error, the following symbol is defined:

$$e_{sq} = \sqrt{e_x^2 + e_y^2 + e_z^2} \quad (\text{LLC-27})$$

where e_x , e_y and e_z are the tracking errors in x, y and z directions. In the PID + DOB control scheme, the feedback gains are set as $k_p = 0.1$, $k_I = 0.0002$, $k_D = 25.0$. While in the Adaptive DOB

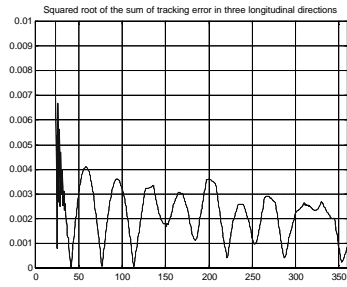
control scheme, $\mathbf{s} = 0.1$, $\mathbf{f} = [0.1 \ 0.1 \ 0.1 \ 0.02 \ 0.02]^T$, $\mathbf{d}_i = 100$, $K = 50$, $\mathbf{e} = [0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01]^T$, $\mathbf{m} = [0.001 \ 0.001 \ 0.001 \ 0.001 \ 0.001]^T$,

$K_I = [0 \ 0 \ 0.0001 \ 0 \ 0 \ 0]^T$. The low-pass filter used in the DOB is $Q = \frac{3ts + 1}{t^3 s^3 + 3t^2 s^2 + 3ts + 1}$ where t is the filter time constant, $1/t$ represents the cut-off frequency of the low-pass filter and it varies in different cases, the sampling period used in digitalization is $T = 0.01$.

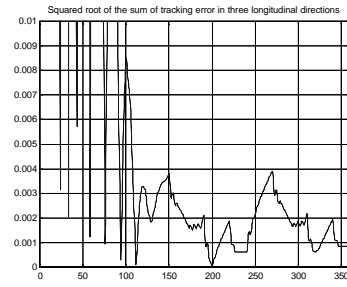
The performance of the two control schemes are tested using the following five cases:

- Case 1: $D = 0.0$ and $t = 0.000$.
- Case 2: $D = 200.0$ and $t = 0.001$.
- Case 3: $D = 0.0$ and $t = 0.012$.
- Case 4: $D = 200.0$ and $t = 0.012$.
- Case 5: $D = 0.0$ and $t = 0.02$.

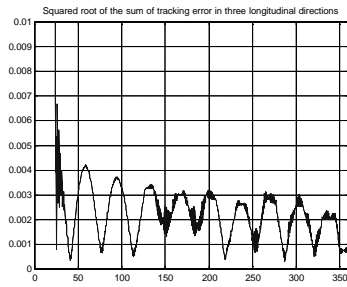
The simulation results are shown in Figure LLC-4 and LLC-4. It is shown that:



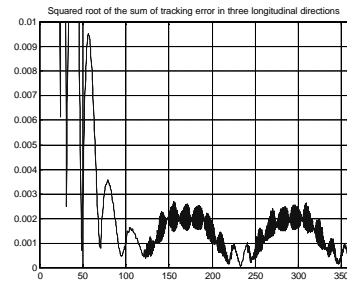
a.1. $D = 0.0, t = 0.001$



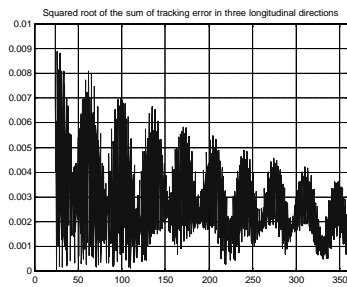
b.1. $D = 0.0, t = 0.001$



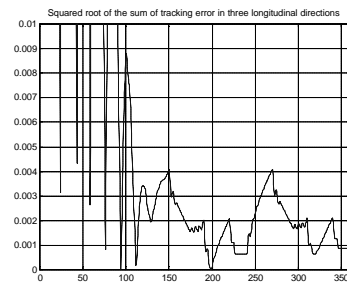
a.2. $D = 200.0, t = 0.001$



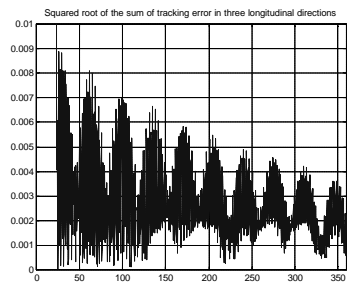
b.2. $D = 200.0, t = 0.001$



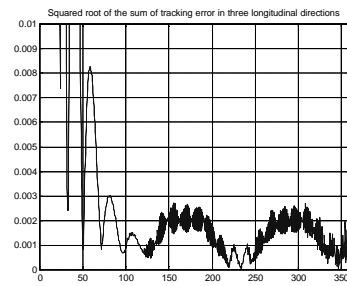
a.3. $D = 0.0, t = 0.012$



b.3. $D = 0.0, t = 0.012$



a.4. $D = 200.0, t = 0.012$



b.4. $D = 200.0, t = 0.012$

a. PID + DOB control scheme

b. Adaptive DOB control scheme

Figure LLC-5 Simulation Result Comparison of e_{sq} under Different Cases

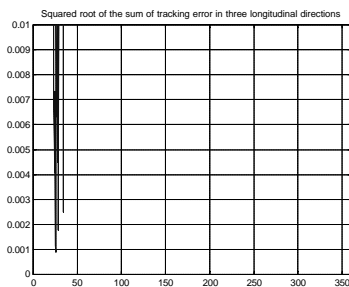
First, comparisons between two figures in each of the following sets: (Figure LLC-5 a.1 and a.2), (a.3 and a.4), (b.1 and (b.2), and (b.3 and b.4) show that the DOB achieves disturbance cancellation successively.

Second, comparisons between two figures in each of the following sets: (Figure LLC-4 a.1 and a.3), (a.2 and a.4), (b.1 and b.3), and (b.2 and b.4) show that with the increase of t , which means the decrease of the disturbance estimate precision, there is little tracking and regulating error observed in the case of adaptive DOB control scheme, while in the case of PID + DOB control scheme, the tracking and regulating performances deteriorate significantly. This is because in the case of PID + DOB, the control gains are fixed. Thus when the disturbance estimate deteriorates, the performance deteriorates too. While in the case of Adaptive DOB, the gains tune themselves with the performance index \tilde{e} , till the performance meets the predefined requirements.

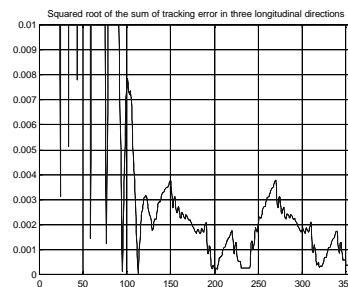
Third, Figure LLC-5 shows that when the t increases further, the Adaptive DOB controller still keeps the nearly same performance, while the PID + DOB controller diverges. Becomes unstable. This is because the non-regressor based adaptive controller adapts its gains according to the system's performance. When $\|\tilde{e}\|\|\Phi_i\| < m_i$, the gains cease to increase. However, since PID+DOB uses fixed gains, when the system disturbance estimate error increases, the PID + DOB controller may become unstable as shown in Figure LLC-5. Therefore, the proposed adaptive DOB controller provides robustness with respect to disturbances and nearly same performance for a wider range of the DOB filter, Q .

Conclusion

A new adaptive DOB controller for underwater robotic vehicles was presented. It was observed from the simulation results that DOB provides robustness with respect to disturbances and modeling errors even though the overall control performance depends on the DOB filter when a linear controller with fixed gains is used as an outer-loop controller. However, using the presented adaptive controller as an outer-loop controller, the overall control performance remains nearly same for different DOB filters. This controller is very attractive especially for controlling AUVs that would operate in unstructured environments without human interventions that would allow adjusting control gains when its performance degrades.



a. PID + DOB when $D = 0.0$ $t = 0.02$



b. Adaptive DOB when $D = 0.0$ $t = 0.02$

Figure LLC-6 Comparison of e_{sq} Using Different Control Schemes

Adaptive Backstepping Control

Backstepping design gives the control system designer a variety of choices. He can incorporate the nonlinearities in the feedback loop by utilizing the damping in the system in several ways. He can also cancel them altogether. The stability proof follows by choosing Lyapunov functions recursively. These properties make the design flexible and well suited for AUV control. The nonlinearities are usually poor or little known, we want to exploit the system's natural damping characteristics.

The name backstepping originates from the block diagram, where the designer add and subtracts the stabilizing function $\mathbf{a}(x)$ to the input of the system, see Figure LLC-7. The system is then augmented with an additional integrator before the regulator and the subtracted \mathbf{a} is moved to the other side of this, and thus becomes $\dot{\mathbf{a}}(x)$. This procedure gives the impression of "stepping back" the integrator, hence the name.

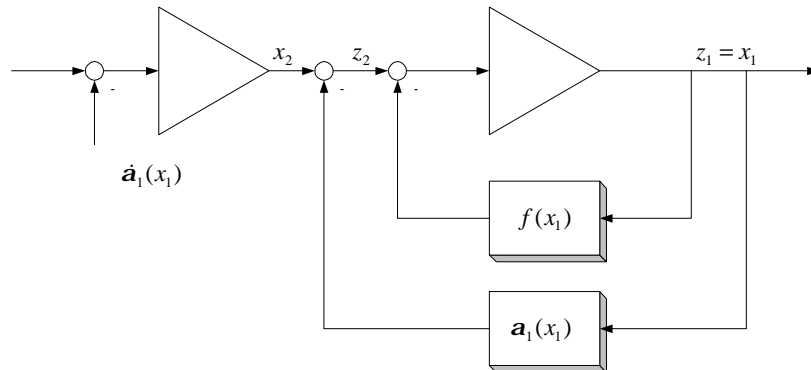


Figure LLC-7 A simple nonlinear system

The key idea of the concept is to start with a system which is stabilizable with a known feedback law for a known Lyapunov function, and then to add to its input an integrator as seen from Figure LLC-7. For this augmented system a new stabilizing feedback law is designed and shown to be stable for a new Lyapunov function. This recursive and structured manner makes it easy to design complex controllers with a simple way to establish a solid stabilizing proof.

Algorithm Description

Adaptive backstepping is an adaptive enhancement of the backstepping scheme. In the adaptive backstepping algorithm, an adaption part is integrated into the control structure. Standard adaption technique is then used to estimate uncertainties in the system equation and provide better feedback to the controller. This feedback can be amplified by design and thus act like nonlinear damping.

Parametric Strict-Feedback Systems

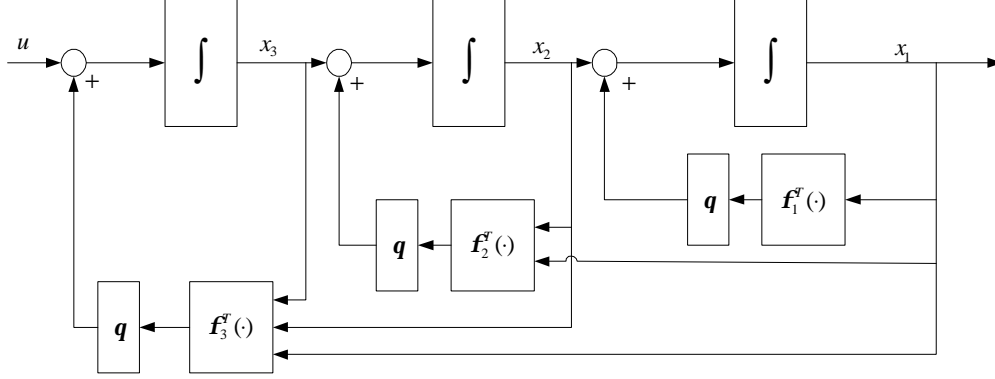


Figure LLC-8 A parametric-strict feedback system

A parametric strict-feedback system is a system that can be written in a parametric strict-feedback form:

$$\begin{aligned}
 \dot{x}_1 &= x_2 + \mathbf{f}_1^T(x_1)\mathbf{q} \\
 \dot{x}_2 &= x_3 + \mathbf{f}_2^T(x_1, x_2)\mathbf{q} \\
 &\dots \\
 \dot{x}_{n-1} &= x_n + \mathbf{f}_{n-1}^T(x_1, \dots, x_{n-1})\mathbf{q} \\
 \dot{x}_n &= \mathbf{b}(x)u + \mathbf{f}_n^T(x)\mathbf{q}
 \end{aligned} \tag{LLC28}$$

A controller for this system is proposed in the following theorem in Krstic et al (1995):

Theorem 1

$$\begin{aligned}
 u &= \frac{1}{\mathbf{b}(x)}\mathbf{a}_n(x, \mathbf{q}_1, \dots, \mathbf{q}_n) \\
 \dot{\mathbf{q}}_i &= \Gamma \left(\mathbf{f}_i - \sum_{j=1}^{i-1} \frac{\partial \mathbf{a}_{i-1}}{\partial x_j} \mathbf{f}_j \right) \mathbf{z}_i, i = 1, \dots, n
 \end{aligned} \tag{LLC29}$$

where $\mathbf{q}_i \in \mathfrak{R}^p$ are multiple estimates of \mathbf{q} . The variables z_i and $\mathbf{a}_i, i = 1, \dots, n$ are defined by the following recursive expressions (with design constants $c_i > 0$ and $z_0 = \mathbf{a}_0 \equiv 0$ used for notational convenience):

$$\begin{aligned}
 z_i &= x_i - \mathbf{a}_{i-1}(x_1, \dots, x_{i-1}, \mathbf{q}_1, \dots, \mathbf{q}_{i-1}) \\
 \mathbf{a}_i &= -c_i z_i - z_{i-1} - \left(\mathbf{f}_i - \sum_{j=1}^{i-1} \frac{\partial \mathbf{a}_{i-1}}{\partial x_j} \mathbf{f}_j \right) \mathbf{q}_i \\
 &+ \sum_{j=1}^{n-1} \left[\frac{\partial \mathbf{a}_{i-1}}{\partial x_j} x_{j+1} + \frac{\partial \mathbf{a}_{i-1}}{\partial \mathbf{q}_j} \Gamma \left(\mathbf{f}_j - \sum_{k=1}^{j-1} \frac{\partial \mathbf{a}_{j-1}}{\partial x_k} \mathbf{f}_k \right) \mathbf{z}_j \right]
 \end{aligned} \tag{LLC30}$$

This overparameterized adaptive controller guarantees global boundedness of $x(t)$, $\mathbf{q}_1(t), \dots, \mathbf{q}_n(t)$ and regulation of $x_1(t)$ and $x_i(t) - x_i^e, i = 2, \dots, n$ where $x_i^e = -\mathbf{q}^T \mathbf{f}_{i-1}(0, x_2^e, \dots, x_{i-1}^e)$

Proof

The proof is found in Krstic et al (1995) and Stapnes (2002). The design algorithm is thus on the following recursive form:

Compute z_1, \mathbf{q}_1 and \mathbf{a}_1

From this, compute the partial derivatives of \mathbf{a}_1

Compute z_2, \mathbf{q}_2 and \mathbf{a}_2

Compute the partial derivatives of \mathbf{a}_2

(...)

Compute z_n, \mathbf{q}_n and \mathbf{a}_n

Since $u \equiv \frac{1}{\mathbf{b}(x)} \mathbf{a}_n$, u is now given

AUV Heave Control design

If we only consider feedback from velocity in z-direction, assuming the cross couplings in the total vessel model to be small, assume the vehicle is completely submerged, i.e. the buoyancy term is constant and small attitude angles, we get the simplified dive equation (Fossen 1994):

$$m\dot{w} + d_t |w|w - \left(mg - \frac{4}{3} \mathbf{pr}^3 \mathbf{r}g \right) = \mathbf{t}_z \quad (\text{LLC31})$$

Since we assume small attitude angles and therefore $\dot{w} \approx \dot{z}$, we can simplify this equation to

$$m\dot{z} + d_t |\dot{z}| \dot{z} - mG = \mathbf{t}_z \quad (\text{LLC32})$$

$$\text{where } G \equiv g - \frac{4}{3m} \mathbf{pr}^3 \mathbf{r}g. \quad (\text{LLC33})$$

The heave equation can now be written on a modified parametric strict feedback form:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{\mathbf{t}_z}{m} + G + \mathbf{f}\mathbf{q} \\ &= x_3 + \mathbf{f}\mathbf{q} \\ \dot{x}_3 &= u \end{aligned} \quad (\text{LLC34})$$

with $\mathbf{f} = |\dot{z}|z$, $\mathbf{q} = -\frac{d_t}{m}$ and the control signal $\mathbf{t}_z = m \int_0^t u dt - mG$.

By using the controller from theorem 1, we can design a SISO controller with nonlinear damping and integrator augmentation by 4 steps. The step details can be found in Stapnes (2002)

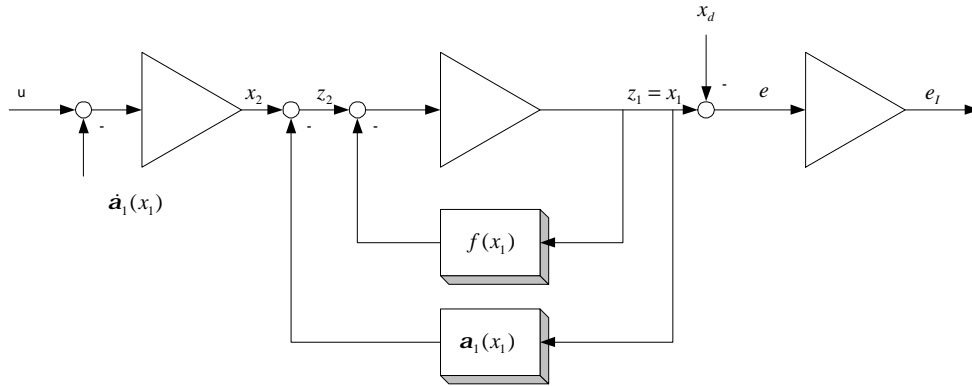


Figure LLC-9 A general system augmented with an integrator

Since $\mathbf{b}(x) = 1$, the control signal $u \equiv \mathbf{a}_4$ is found to be

$$\mathbf{a}_4 = -c_4 z_4 - z_3 + \frac{\partial \mathbf{a}_3}{\partial x_3} \mathbf{f}_3 \mathbf{q}_4 + \frac{\partial \mathbf{a}_3}{\partial x_1} x_2 + \frac{\partial \mathbf{a}_3}{\partial x_2} x_3 + \frac{\partial \mathbf{a}_3}{\partial x_3} x_4 + \frac{\partial \mathbf{a}_3}{\partial \mathbf{q}_3} \Gamma \mathbf{f}_3 z_3 \quad (\text{LLC35})$$

The total control input is thus $\mathbf{t}_z = m \int_0^t \mathbf{a}_4 dt - m\mathbf{G}$.

The system can easily be extended to MIMO control by using the concept of Vectorial Backstepping and is done in Stapnes (2002)

Simulation Results

The simulations were conducted with a cascaded 6 DOF MIMO controller. The heave motion was controlled by the adaptive backstepping controller, while the other DOFs were controlled by a simple sliding mode controller implemented by Mr. Side Zhao at ASL. A PID controller from was used as benchmark controller in the simulations to evaluate the backstepping controller's performance.

Step disturbance rejection

A step disturbance can be caused by an unexpected collision to an object, such as corals or debris. A step disturbance analysis can also be a measurement on how the system adjust to changes in weight distribution, which can be caused by for instance loss of tools during an underwater operation.

During this simulation, a disturbance identical to 10 N was added for 50 seconds, and then released. Figure LLC-11 shows the position error

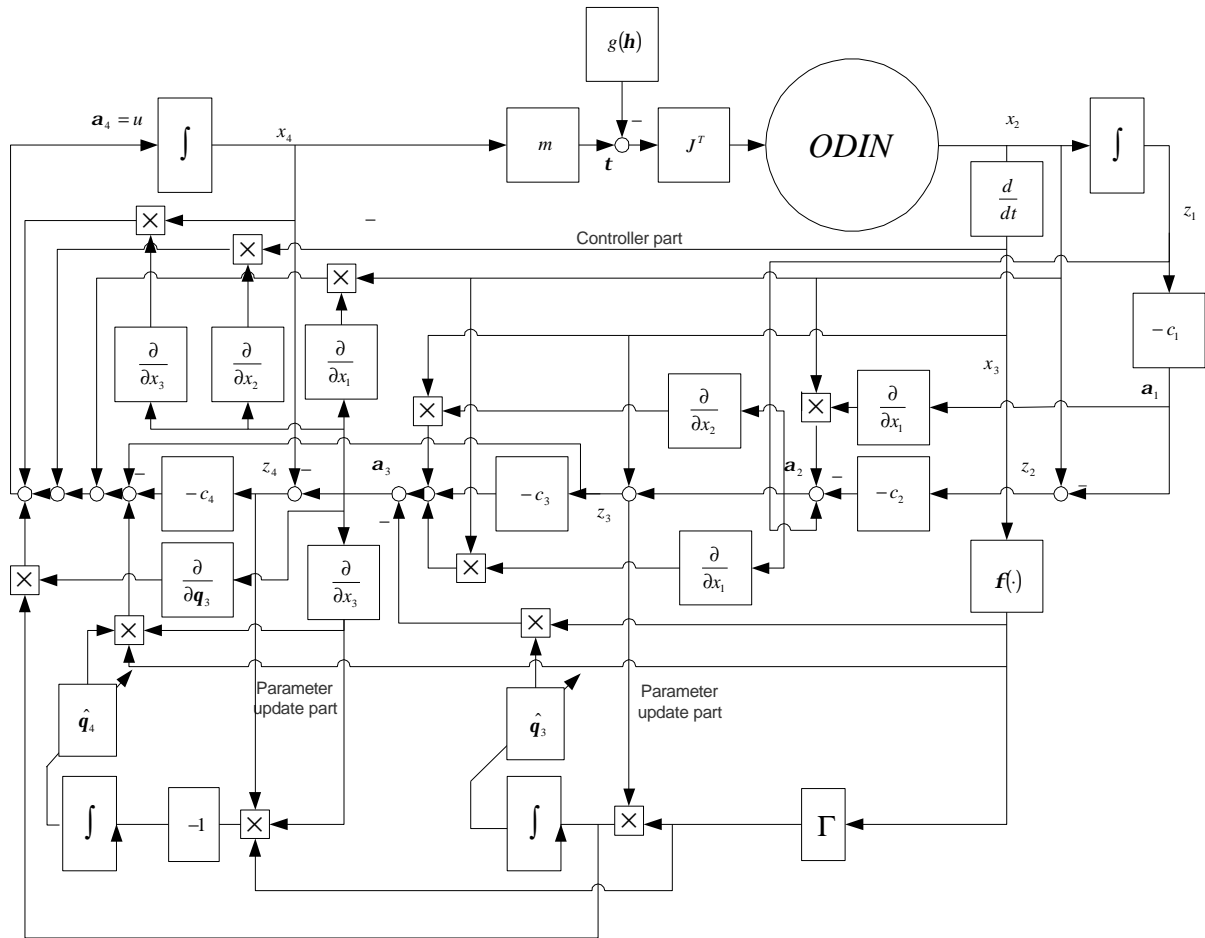


Figure LLC-10 The integrator augmented controller

We see that the backstepping controller's step disturbance rejection performance is slightly better than the PID controller. The backstepping controller is faster and has less overshoot. However, the work performed by the thrusters is more aggressive, and therefore more power consuming.

General disturbance rejection

It is useful to investigate the controller's ability to reject general disturbances since they always will exist. These disturbances can be interpreted as a combination from a lot of different sources, such as unmodelled effects in the mathematical equation of the vehicle, currents, and so forth.

The disturbance vector is modelled as a superposition of two slowly-varying sinusoidal waves, which could represent current components in z-direction and unmodelled dynamics, and a smaller white noise component.

The solid line is the backstepping controller, while the dash-dot is the PID controller

A comparison between the backstepping controller and the benchmark PID position error is shown in Figure LLC-12. The backstepping controller is performing better than the PID and rejects the disturbance added.

Adaption performance

Since the regressor is based upon the velocity of the vessel, it will become singular after ODIN has reached and stabilized on a stationary position and the parameter update will cease.

The Figures LLC-13 and LLC-14 show that the actual effect of the adaption is small. The controller seems to perform quite well without the adaption turned on, although the settling is somewhat smoother with adaption active, as the first 50 seconds of simulation shows.

By selecting another initial value, the parameter estimate oscillated in phase with the dive. This suggests that the a priori estimate of \mathbf{q} is of some importance for the convergence of the adaption.

Nonlinear Damping

One of advantages with backstepping is the utilization of nonlinear damping. By amplifying the natural damping, we can damp a system without a precise knowledge of the system.

Figures LLC-16 and LLC-17 show that it is the vehicle's behavior in the first seconds of the simulations that is improved by adding the nonlinear term. The characteristic starting peak from the previous sections is removed by adding the \mathbf{k} -term. This is in compliance with the theoretical results in Krstic et al (1995). In these Figures, Solid line shows backstepping controller

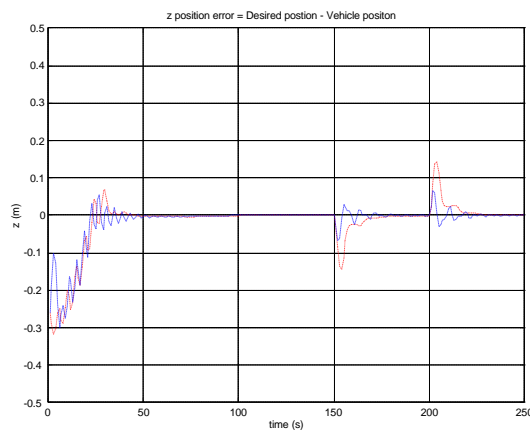


Figure LLC-11 Z position error after step disturbance turned on at $t = 150$ [s], and off at $t = 200$ [s].

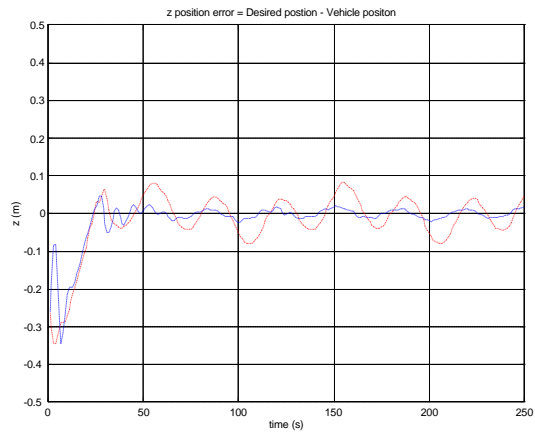


Figure LLC-12 Error in z-direction with added disturbance.

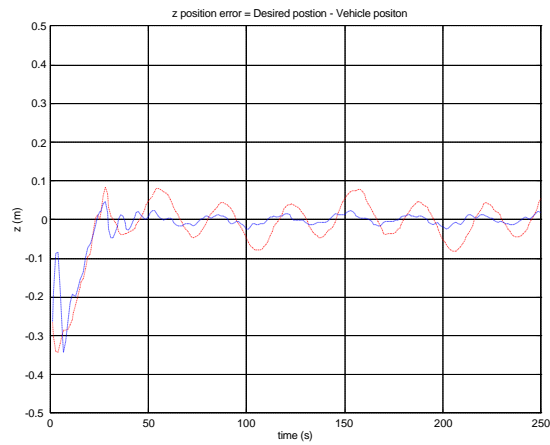


Figure LLC-13 Z position error with adaption turned on

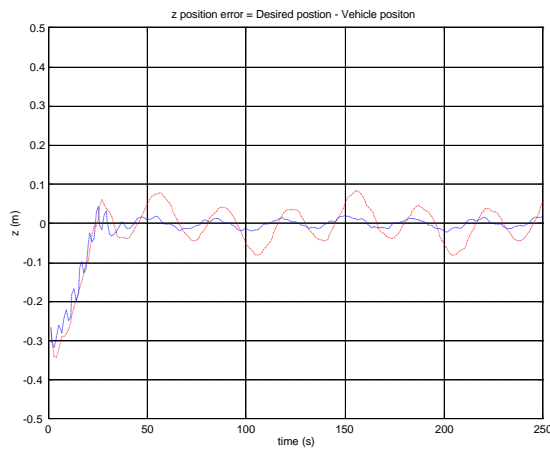


Figure LLC-14 Z position error with adaption turned off

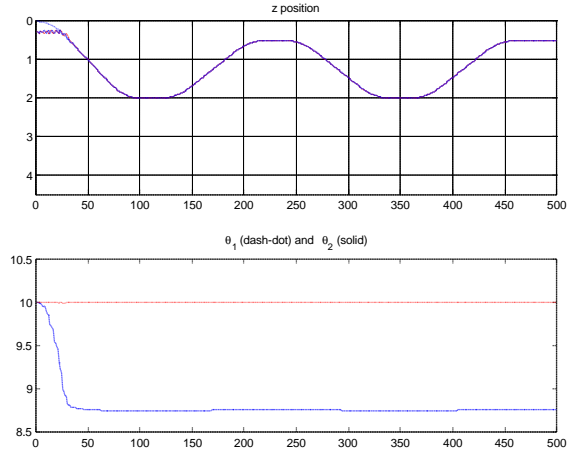


Figure LLC-15 Dive profile compared with parameter convergence with $q_0 = 1/15 \approx 0.07$

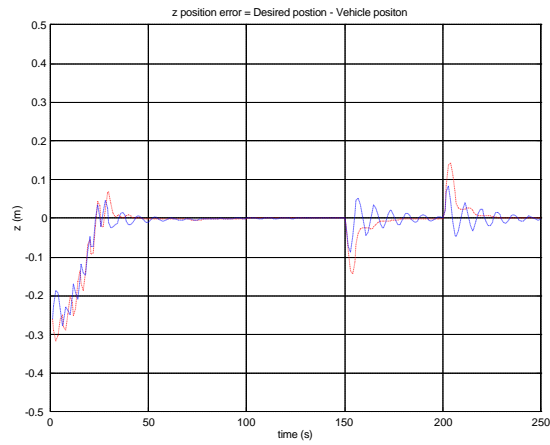


Figure LLC-16 z position error with $k = 0$ and $\Gamma = 0.01$

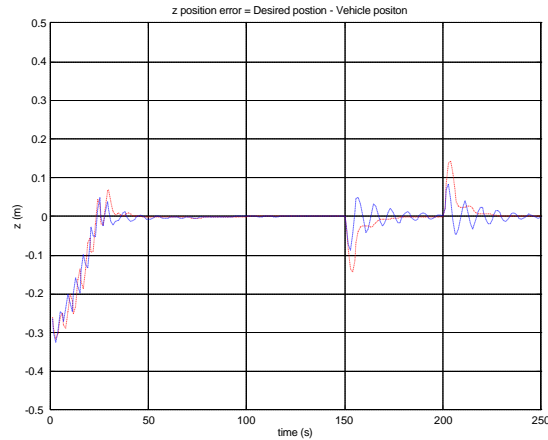


Figure LLC-17 z position error with $k = 1$ and $\Gamma = 0.01$

Conclusion

The overall impression from the simulations presented is that the adaptive backstepping controller suppressed added disturbances satisfactorily, and had performance characteristics roughly as a nonlinear PID. The controller also had faster response and less deviation with step disturbance than the PID controller who were used for comparison purposes.

The adaptation part didn't seem to have a significant effect on the system and played little or no role in the overall controller performance during the simulations. The nonlinear amplification term slightly improved the transient performance of the vehicle, but had little effect in the overall performance.

Thruster output with adaptive backstepping controller was more aggressive than the PID output. This is because the controller is quicker to counteract for disturbances. The drawback is more power being consumed and thus shorter battery life.

The backstepping controller is not very intuitive and therefore difficult to tune without specialized tuning techniques. Oscillations seemed to prevail until point of instability, which can be explained by the low sampling frequency (5 Hz) used in the simulations. Sampling frequency also limits the gain selection and instability could be a problem. The gains can only be chosen in a certain window, which is enveloped by instability or no action at all.

The complexity of the regulator design increases exponentially with number of steps. This makes a vectorial backstepping approach attractive, since the recursive design steps are taken for matrices and vectors, and not scalars.

ODIN-III – System

In order to develop a low-level control algorithm, we use ODIN (Omni directional Intelligent Navigator) instead of SAUVIM. In this phase, we upgraded ODIN to make new environment for developing control algorithm. Originally, ODIN-II had VME based system and VxWorks real time operating system, and communicated with notebook computer using RS-232C Serial Communication. However, under these environments, we were not able to change or modify control algorithm outside lab, because we needed a special compiler installed in Workstation and EPROM needed to save the

algorithm. And we had to open vehicle to charge batteries. It meant that we had to spend a lot of time to prepare test for developing control algorithm.

To lessen these burdens, we modified ODIN. VME and VxWroks based system are changed with PC104 and Windwos2000 plus real time extension. And, communication method between vehicle and notebook computer are changed with TCP/IP. Whole software for vehicle operation and monitoring are also changed with windows based software. Therefore, we can easily monitor whole data coming from vehicle and can modify control algorithm in the notebook computer with usual C++ compiler, Visual C++. After than, execution file can be transferred to ODIN through TCP/IP. Additionally, we can charge batteries using charging connector without disassemble of vehicle. Therefore, we can save a lot of time to develop new control algorithm.

Main feature of ODIN-III is PC104 with AD/DA board and Windows2000 with real time extension. This structure gives a lot of advantages, for example, graphic user interface, new software developing environment. And batteries and whole electrical wires are changed. Specially, the plug between sonar head and vehicle body is changed to make new battery charging method. And, some internal sensors (temperature and battery sensors) are changed.

However, three important external sensors, e.g. sonar sensor, inertial measurement unit (IMU), pressure sensor, and actuators, e.g. thrusters, amplifiers, and stepper motor are same to ODIN-II.

Figure LLC-18. shows ODIN-III and Table LLC-1 describes new main components in ODIN-III.

Table LLC-1. Description of new main components

No	Name	Features
1	PC104-PLUS Cool Roadrunner II made by LIPPERT	All in One CPU Board: Pentium Compatible CPU 300 MHz 256 MB RAM, 20 GB External HDD, Ethernet Port, ISA and PCI Bus
2	AD Board Diamond –MM-AT made by Diamond systems corporation	16 analog inputs, 2 analog output 12-bit D/A resolution, Programmable gain, 24 digital I/O line, Auto-calibration
3	DA Board RUBY-MM-1612 made by Diamond systems corporation	16 analog voltage outputs, 12-bit D/A resolution, Simultaneous update of all DACs, 24 digital I/O line
4	Real Time Extension RTX 5.0 made by VenturCom	Add real time capability to Windows2000

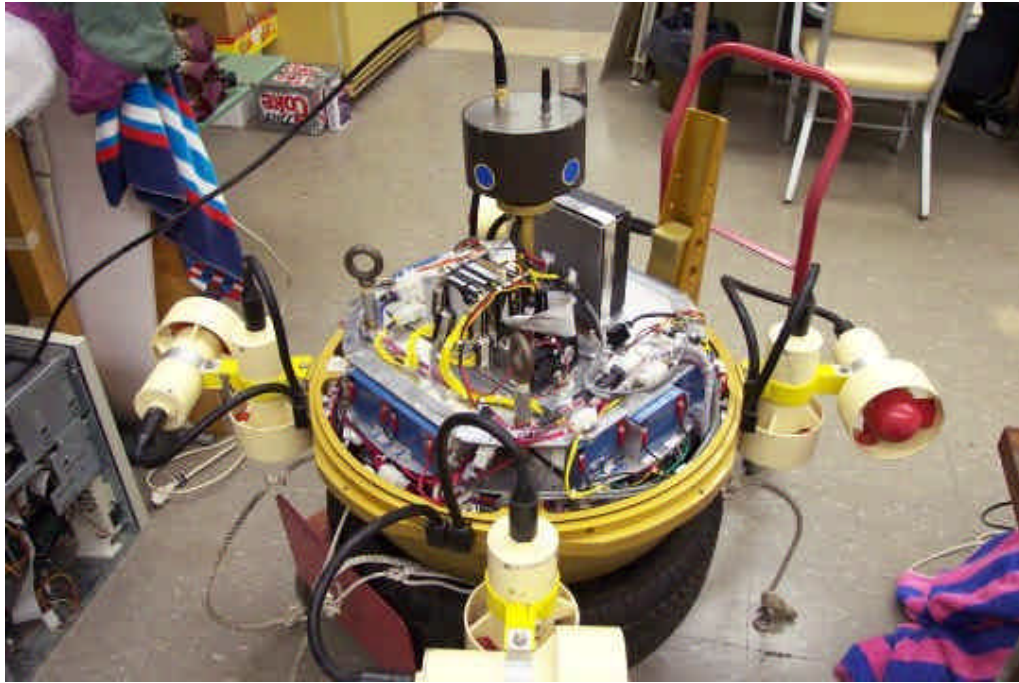


Figure LLC-18 ODIN-III with PC104

ODIN-III – Software

Software for ODIN-III is designed in Visual C++ with MFC environment. This consists of three modules, `auv.exe` operated in vehicle, `station.exe` operated in notebook computer, and control Dll operated in both. The control Dll (Dynamic Link Library) is the most important point in this scheme. Basically, Dll can be loaded or unloaded without a recompile or a re-link of main software. A control algorithm being developed is implemented in control Dll, which means control algorithm independent with the other two softwares. Therefore, to develop control algorithm of vehicle, we can design and test various control algorithms without loss of time. Figure LLC-19 shows concept of this structure.

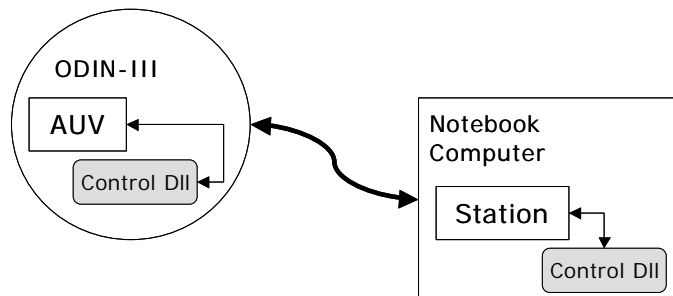


Figure LLC-19 Concept of software structure

AUV.exe

As main software of vehicle, it has three threads with windows message loop as shown Figure LLC-20. All commands for operating vehicle come from notebook computer, called station. And all

information getting from sensors, result of command, and acknowledgment will be sent to station. Thus, it has only two menus, disconnect TCP/IP and exit for unusual situation. Actually, when station disconnects TCP/IP, this software disconnects TCP/IP automatically.

After connection with station using TCP/IP, message handling loop always check and analysis a command, then call a function for carrying the command. And after connection RS-232C serial communication for IMU, IMU thread check and save IMU data. During vehicle is working, all sensor raw data and manipulated data, e.g. 8 sonar signals, pressure sensor, 9 IMU signals, 2 temperature signals, 2 batteries monitor signals, position from walls or obstacles, depth, and so on are sent to station using Send info. Thread.

There are three control modes in this software.

Open-loop Remote Control Vehicle (ROV) mode is to control vehicle using motion command of station without any controller in vehicle. It is working without Dll. In this mode, OROV thread generate thruster input at every 300msec, where OROV command comes from station keyboard. This thread is non real time thread.

Closed loop ROV mode is to control vehicle using motion command for forward/backward motion, left/right motion, and up/down motion with roll, pitch, and yaw control algorithm.

Autonomous Underwater Vehicle (AUV) mode is to control vehicle using given trajectory with full control system.

Closed loop ROV Dll and AUV Dll are transferred from station using TCP/IP like FTP, and then automatically loaded in this software. These Dlls have real time thread for control. Details about these Dll will be given later.

This software operates stepper motors in three directions, forward/backward, left/right, and up/down to keep the vehicle balance before starting control.

Figure LLC-21 shows view of this software. In this window, we can check current IMU data and current vehicle status for debug.

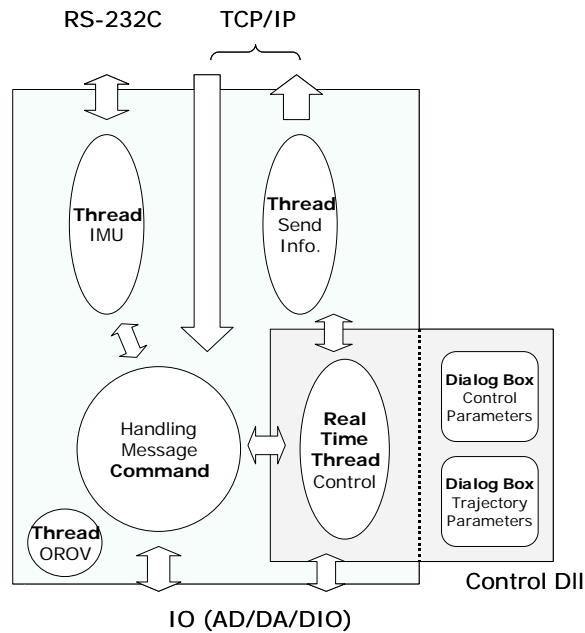


Figure LLC-20 Structure of AUV software



Figure LLC-21 AUV software

Station.exe

As main control panel, this software displays all data coming from vehicle and send command to vehicle. And this give a lot of dialog boxes for user interface. Figure LLC-23 shows structure of station software. It has two threads with message handling loop. Message handling loop send commands getting from user input in dialog boxes and get acknowledgement or result of command. Display thread display all data coming from vehicle on the screen and Data save thread save the data in memory using the dynamic memory allocation method. After finishing moving, these data in memory can be saved in specific file in HDD.

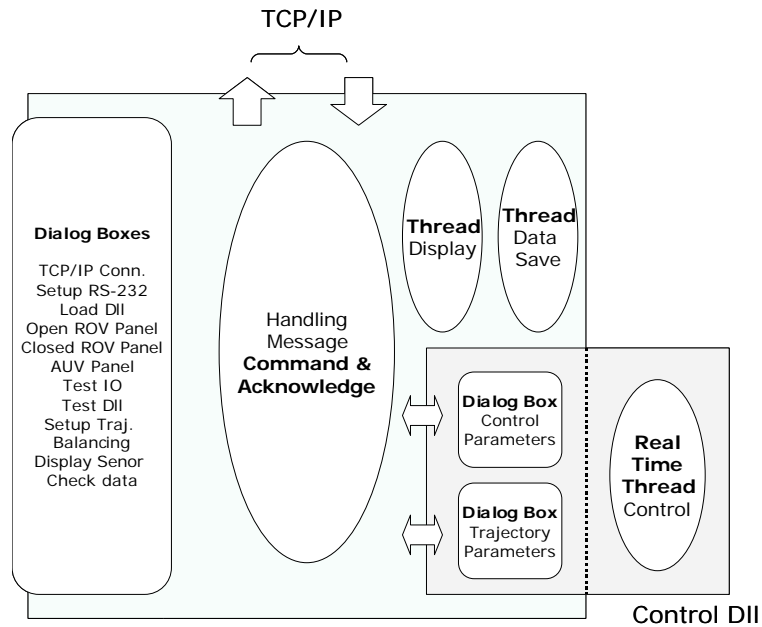


Figure LLC-22 Structure of station software

As shown in Figure 23, it has five menus, e.g. ignition, operation, test, and option. Each menu has 3 - 5 sub menus. All commands getting from these menus will be sent to vehicle, and in some case, it wait acknowledgement from vehicle and display this result on screen.

Ignition menu is related to basic setup for working vehicle, and consists of connection/disconnection of TCP/IP, connection/disconnection of serial communication, load/unload of DII, and Exit.

Operation menu is related to vehicle operation, and these sub-menus only can be selected after connecting TCP/IP. First sub-menu, balance is to control stepper motor in vehicle to keep vehicle balance. Second sub-menu, OROV is to control vehicle in the Open-loop ROV mode. In this mode, keyboard commands directly send to thrusters of vehicle. Third and fourth sub-menus can be only selected when each DII are loaded, respectively. CROV menu generates keyboard command just like OROV. And additionally we can set orientation of vehicle with two cases. One case is to keep current orientation and the other case is to set specific angle of roll, pitch, and yaw as a references. Therefore, this CROV DII has three controllers for roll, pitch, and yaw. Control parameters of each controller can be changed in option menu. AUV menu give same method to set orientation of vehicle and trajectory planner to make a reference position for vehicle motion. Details about trajectory planner will be described later.

Test menu is related to check system IO, DII before loading, data file. Test IO sub menu only can be selected after connecting TCP/IP. In this menu, we can check the performance of all Input output (IO) including analog digital converter (ADC), digital analog converter (DAC), and Digital IO, and current configuration of IO as well.

Option menu is related to set parameters for environment or change control parameters for each DII. Stepper motor sub menu is for setting of stepper motor operation, for example speed, home position.

And sensor display is for setting of calibration data of temperature sensor or battery monitor sensor, and pressure sensor. CROV sub menu only can be selected when CROV Dll is loaded because this dialog box comes from Dll. In this menu, we can change control parameters of roll, pitch, and yaw. Default control methods for three are PID. Of course, we can implement advanced control algorithm in CROV Dll. For AUV, there are trajectory set sub menu and AUV sub menu, which only can be selected when AUV Dll is loaded, because these two dialog boxes comes from AUV Dll. In trajectory set menu, parameters used for making trajectory, for example maximum acceleration, maximum velocity can be changed for three directions, X, Y, and Z. In the AUV dialog boxes, all control parameters of each controller implemented in AUV Dll can be changed. Default control methods are PID. Of course, we can modify AUV Dll to implement advanced control algorithm. On the screen, several graphic objects are used to make that user get information easily. And it supplies 6 temporary variables to allow display on screen.

Control Dll

As mentioned above, control Dll scheme allows us to develop advanced control algorithm without understanding and modifying whole software. And, it save a lot of time to prepare for testing each control algorithm. Therefore, control algorithm designer can concentrate on algorithm itself.

There are two types control Dlls, CROV Dll and AUV Dll. CROV Dll is for orientation control only, which has only control parameter dialog box for these three controllers. On the other hand, AUV Dll has two dialog boxes for control parameters for six controllers and trajectory parameter.

Functions in control Dll can be put into two parts. One part is for AUV.exe; Functions in this part are related real time thread, control algorithms, and trajectory planner, which are called in AUV.exe. The other part is for dialog box. These functions are called in station.exe.

When Dll are loaded in station.exe, its all functions are checked to confirm of Dll. After that, it is transferred to AUV.exe, and loaded.

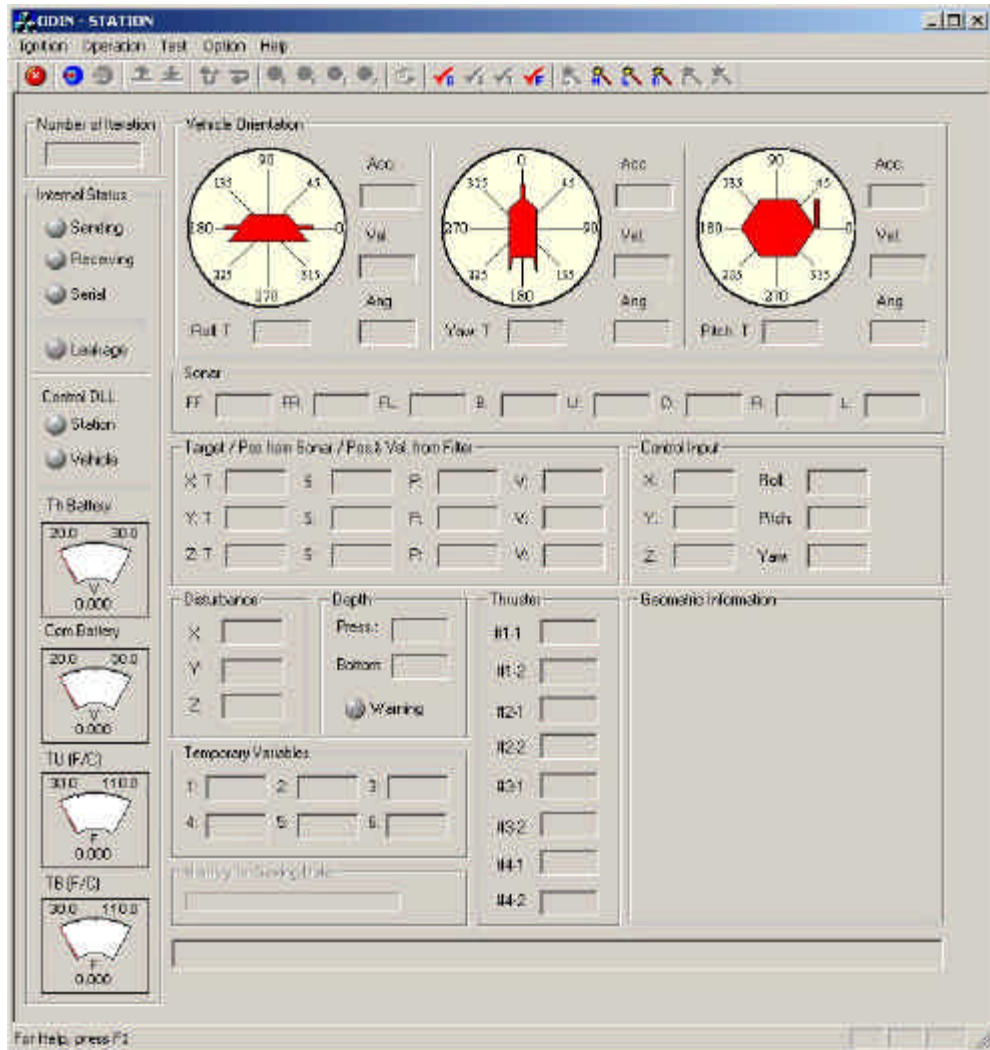


Figure LLC-23 Station software

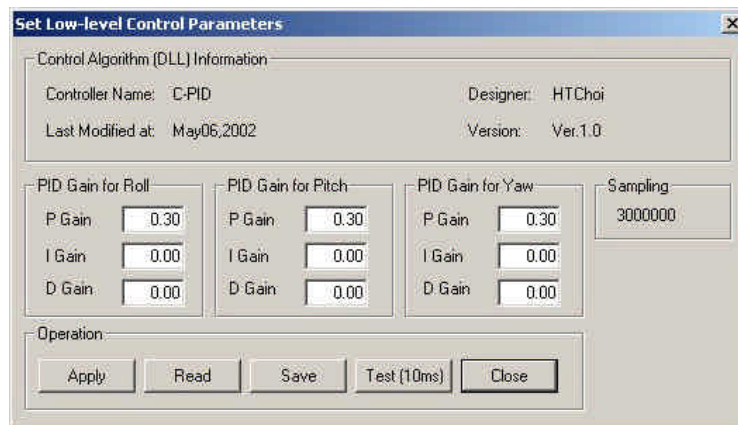


Figure LLC-24 Dialog box for setting Control Parameters in CROV mode

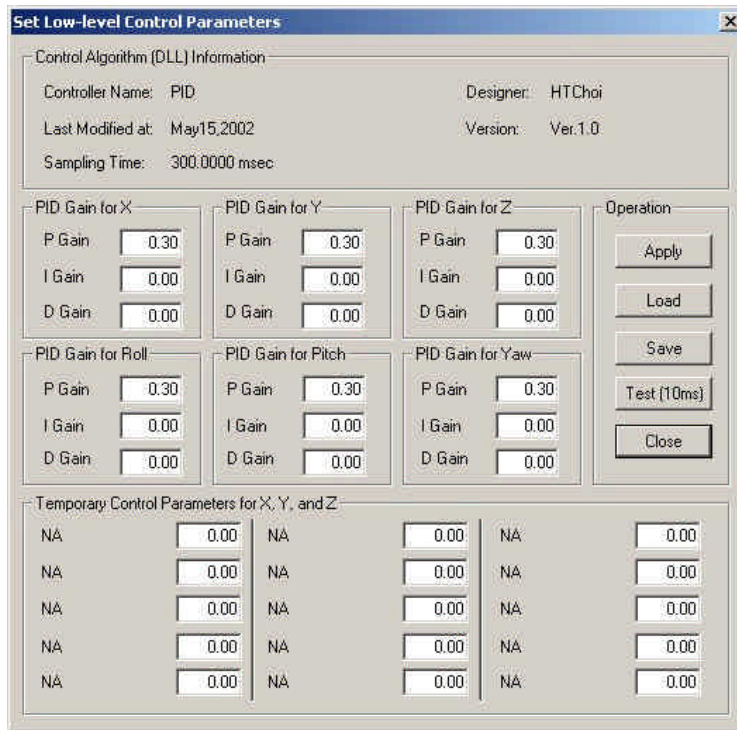


Figure LLC-25 Dialog box for setting Control Parameters in AUV mode

Trajectory Planner

This trajectory planner is a usual trajectory planner. It can generate reference positions in X, Y, and Z at every sampling time. For this, AUV.exe gets sequence of target position sent from station.exe. And, the planner generates reference position between target positions using give maximum acceleration and maximum velocity. Now, only trapezoidal type planner is implemented. This planner can compensate discretization error, which occur each target position because switching time between maximum acceleration and 0 acceleration and between 0 acceleration and deceleration might not be exact sampling time as shown in Figure LLC-26. To solve this problem, before generating reference position, maximum acceleration and maximum velocity are adjusted slightly to be switching time on the sampling time using the constraint that two areas which mean distance before/after adjusted, must be same in geometric sense as shown in Figure LLC-27.

Additionally, this planner can make diagonal motion in X and Y plane regardless each distance. For this, in case of different distance in X and Y, coordinator divide the maximum acceleration and velocity of short distance with the ratio of distance between X and Y under assumption of X and Y thruster power are same. Now, only diagonal motion in X and Y are considered because thruster power of Z direction is different to that of X or Y. For diagonal motion in 3D, distance and power of each axis must be considered at the same time.

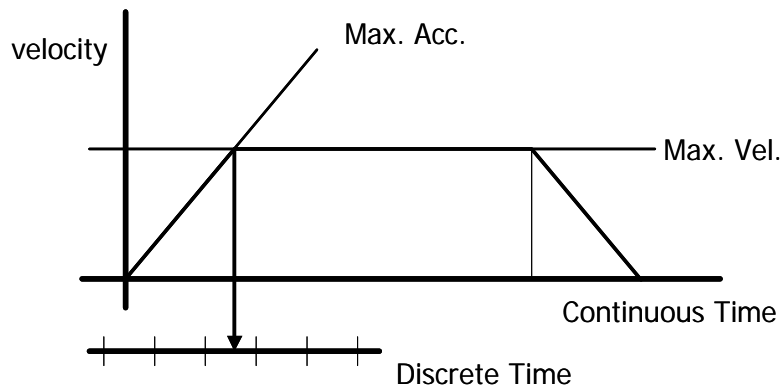


Figure LLC-26 Discretization error in trajectory planner

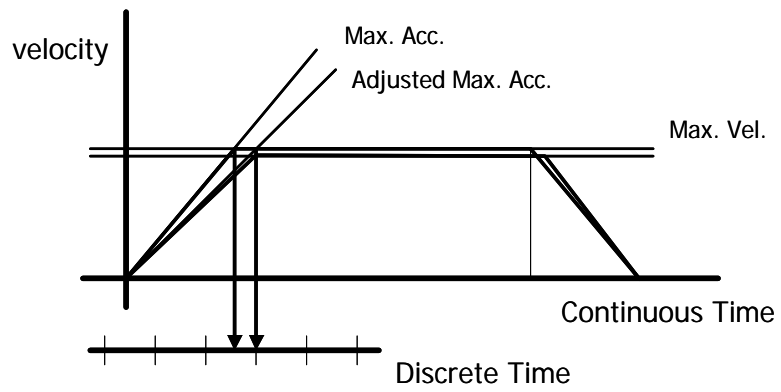


Figure LLC-27 trajectory planner using adjusted parameters

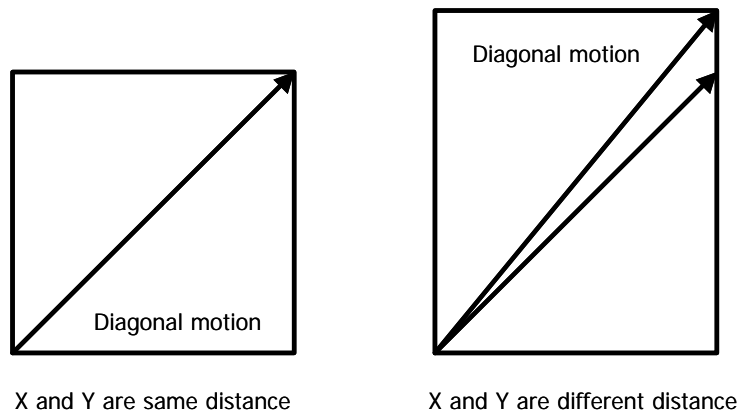


Figure LLC-28 Diagonal motion

Simulation Environment

Simulink version: Former version of simulator was built in Matlab m-file. It was not an easy environment to develop control algorithm even though the m-file version is running a little bit faster than the Simulink version. Now, we can design control algorithm using drag and drop method in Simulink environment. To save simulation time, some parts used mex-file (using C language). Core of simulator is state space equations itself, and all parameters related to simulation condition, for example, sampling time, w/ or w/o disturbance, and feedback information are changeable using menu. We're going to add another module to make this realistic simulator after getting data from experiment test.

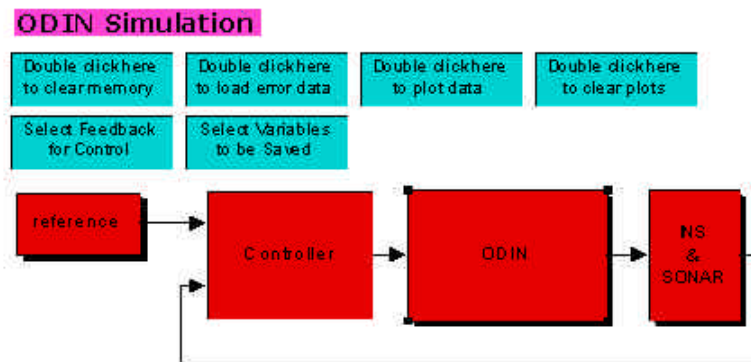


Figure LLC-29 ODIN-III Simulator under Matlab/Simulink

Localization and Navigation (LN)

Project Leader(s): Mr. Michael West
Personnel: Mr. Kaikala H. Rosa, Mr. Scott A. Menor, Mr. Daniel Shnidman & Mr. Mike Hall
Past Project Leader(s): Dr. Junku Yuh, Dr. Tae Won Kim & Dr. Song K. Choi
Past Personnel: - none -

Objectives

The design and implementation of an underwater navigation system for the SAUVIM vehicle.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

The harsh, forbidding, and dark environments in which AUV must work make it difficult for the vehicles sensory system to determine its position. The success of future AUV's will be the ability to accurately navigate (determine the vehicle within geodetic or relative coordinates) and localize (determine the vehicles specific distance from some fixed point) itself in this underwater domain. The underwater world limits the types of sensors available, as compared to that of above the water surface sensors and their vehicles. Electromagnetic energy propagates very little in water and, thus, instruments such as the Global Positioning System (GPS) has limited use in water. However, if truly autonomous systems are to be developed, good navigation sensory information is needed in order to achieve mission goals and provide safe operation.

The current practice for AUV navigation typically involves the use of at least one of the following three types of techniques: Acoustic Transponder Navigation, Dead Reckoning and Inertial Navigation, and Geophysical Navigation. Acoustic navigation usually involves acoustic energy beacons at different baselines: long, short, and ultra-short. Navigation by dead reckoning is defined as determining the location by knowing ones speed and heading. Geophysical navigation uses Earths geophysical traits such as magnetic field, gravity, or bathymetry. An overview of these systems is discussed in Blidberg and Jalbert (1995), Leonard and others (1998) and Babb (1990). The navigational requirements for the sensors involved with the underwater systems is described in Stambaugh and Thibault(1992) and Hutchison et al. (1990).

Acoustic Transponder Navigation

The most widely used scheme for underwater navigation is the acoustic baseline techniques; and, several papers have described their use in AUV systems (Austin 1994, Vickery 1998, Carof 1994, Austin and Stokey 1998, Smith and Kronen 1997, and Watson et. al. 1998). Vickery (1998a) gives a comprehensive overview of acoustic positioning systems, their configurations, and operational considerations. Vickery (1998b) continues his overview of acoustic positioning systems with new concepts and future concepts. Long baseline (LBL) navigation consists of an array of transponders, which are deployed in surveyed locations. The vehicle sends out an acoustic signal that is then returned by each beacon as it is received. Position is determined by measuring the travel time between the vehicle and each beacon, measuring or assuming the local speed profile, and knowing the geometry of the beacon array (Leonard and others (1998)). Short baseline (SBL) usually has the array transponders on the surface ship; and, thus, the baseline length is shorter than that of a LBL system. Smith and Kronen (1997) illustrate the use of an SBL system with the Ocean Explorer vehicle. With

an ultra-short baseline (USBL) system, the vehicle has a multi-element receiver array that enables it to measure the angle as well as the range to an acoustic beacon. By measuring the arrival time difference of a single sonar ping between two or more hydrophones, the bearing from the vehicle to the beacon can be determined (Leonard and others 1998). USBL systems have had increased interest recently, for the cost is much lower than the LBL or SBL configurations. Austin (1994), Watson *et al.* (1998), Black and Butler (1994) and Austin and Stokey (1998) show USBL systems with increased accuracy for long range missions. The range of these systems is shown in Table 8. Errors for these baseline systems can arise as a result of out of range limitations, water temperature variations, poor placement of transponders, and bias errors (Blidberg and Jalbert 1995). This type of navigation system is a very mature system and can provide positioning to an absolute accuracy of 5m and a relative accuracy of < 2m. Several manufacturers of acoustic navigation sensors are shown in Table 9. A more detailed comparison of a couple manufacturers, Sonatech and Desert Star, is shown in Table 10.

Dead Reckoning and Inertial Navigation

Dead reckoning is by far the most established navigation system. The system usually involves magnetic heading sensor and a velocimeter in order to measure the vehicles velocity. However, the main problem with vehicles in the ocean medium is currents acting upon the vehicle which are not detected by the sensors. Ocean currents can be quite severe, especially in relation to a slow moving AUV; and, thus, the position calculation can be quite poor.

Inertial navigation determines the vehicle position by integrating the acceleration twice in time. However, error accumulates as a result of what is called “Drift” which is the result of change in the output rotation rate sensor with time due to effects such as temperature, magnetic fields, aging, and wear of components (Blidberg and Jalbert 1995). Barbour and Schmidt (1998) and Huddle (1998) discuss the current state of the technology in Inertial Navigation Sensor (INS). Specific INS sensors are discussed in Cox and Wei (1994), and Murphy (1998). INS navigation may be improved through a Kalman filter by position updates, such as GPS if working near the surface or by velocity updates. A comparison of three INS manufacturers is shown in Table 11

Velocity updates can be achieved by acoustic velocity log sensors that are split into two types: Doppler Velocity Log (DVL) and Correlation Velocity Log (CVL). Table 12 shows some different manufacturers of Velocity Logs. DVL systems use the Doppler principle with respect to reflected echoes off the bottom of the ocean to calculate the motion with respect to the ocean floor. Doppler underwater navigation is described in Jorgensen *et. al.* (1994). A detailed comparison of three manufactures of DVL systems is shown in Table 13. The typical configuration for the CVL system is the Spatial Correlator where hydrophones placed under the vehicle in a given array geometry receive the echoes from each transmitted pulse. The cross-correlation measurements between the various echoes are analyzed to estimate the distance traveled by the vehicle during the time of the given pulse interval. Correlation velocity log systems are further illustrated in Griffith and Bradley (1998), and Grose (1992).

As stated earlier, inertial navigation systems are subject to drift and are usually compensated with either position or velocity information. The integration of this information by the use of Kalman filtering can improve the system perform tremendously. Packaged sensors which include both inertial and velocity log sensors have been developed (Trimble 1998). The DARPA UUV achieved navigation performance of 0.01% of distance traveled using an integrated INS/DVL system (Leonard *et al* 1998).

Geophysical Navigation

Still in their infancy, the research into geophysical sensors has shown some encouraging results. The geophysical sensors are similar to the types of senses that the animals use to navigate. The most common of these types of navigation techniques are magnetic field, gravity based, and terrain matching. Each of these techniques is based upon the matching of sensory information with a given, “a priori”, map of the environment.

The irregular geomagnetic fields that span the earth everywhere could be mapped for the area in which the vehicle is to operate. Magnetic field strengths could be measured and compared with a map of the magnetic topography. If two more magnetometers were to be placed in a known location on the underwater vehicle, then the displacement of the two curves measure could directly indicate ground speed (Blidberg and Jalbert 1995). However, the magnetic fields change from day to night and are affected tremendously by magnetic storms. Complicated algorithms would be needed in order to compensate for these changes and man made ones. Currently, fielded AUV systems do not use this type of navigation.

Earth gravitational field maps have been collected by the US Navy for the calibration of inertial navigation systems (Leonard *et al.* 1998). Using these “a priori” maps of the earth’s gravitation fields, a vehicle’s position may be obtained. Jarcitano and others (1990) presented an INS aided by gravity based navigation scheme for AUV’s using the Bell Aerospace Textron Gravity Gradiometer System (GGS). Goldstein and Brett (1998) used the Bell GGS in a non-powered towed underwater vehicle with some very intriguing results. The main drawbacks of these types of systems are the cost, size and complexity of the sensors.

The most studied of the types of geophysical navigation is the position determination via bathymetric maps (Bergam *et al.* 1993, Cristi *et al.* 1995, Johnson and Hebert 1996, McLaren and Kuo 1996, Auran and Silvan 1995, Carpenter 1998 and Feder *et al.* 1998). Since vision systems are extremely limited even at shallow depths, terrain of the ocean floor is typically obtained through the use of multibeam sonar. The sonar map of the sea floor is then matched with an “a priori” bathymetric map. McLaren and Kuo (1996) used a geometrical algorithm to process an imaging sonar sensors data. The AUV’s environment included mainly vertical and inclined cylinders at know positions. Auren and Silven (1995) created a 3D occupancy grid of the environment. Cristi *et al.* (1995) used sonar data in order to construct a potential function. The gradient of the potential function is used as an error signal in a simple LMS algorithm for the recursive correction of the estimated vehicle’s position. In many cases the vehicle will be in an unknown environment, Feder *et al.* (1998) and Carpenter(1998) attempts to address this problem with Concurrent Mapping and Localization (CML). The goal of CML is to have the AUV build a map of the environment that is operating and to use this map to navigate in real time.

SAUVIM Navigation

The Localization and Navigation system for SAUVIM is still early in its development. From the time SAUVIM is deployed and submerges (figure LN-1 and LN-2), it will need to navigate to the target autonomously. After the vehicle is completely under the water (figure LN-3), the navigation system must determine its location. Using the information provides by the ocean terrain, the vehicle will navigate along the ocean bottom (figure LN-4 and LN-5).

Currently, we are investigating different techniques using Geophysical navigation. First, since SAUVIM will be operating in areas where we will have full bathymetric knowledge, we are looking at map matching techniques using an a priori map of the environment. Some of the techniques that we are looking at include map matching of the 2-D contour map using the Hausdorff Distance, map matching of 3-D elevation map using an occupancy map, and map matching of a 3-D mesh map.

The first step in our development of the Navigation system is to design a simulation environment to test our algorithms. We are developing an OpenGL environment interfaced with a Simulink model of our system. This will allow us to develop our algorithms for navigation and test them within a simulated environment without operating expense of the real system. Figure LN-6 illustrates the top page of the SAUVIM navigation simulation within Simulink.

The next step in the Navigation systems development is the testing of the algorithms on a land based robot (Magellan) and a smaller AUV. We have purchased a land based robot, the Magellan, to help us develop our navigation algorithms. The Magellan includes a scanning laser which will provide data which is similar to that of a profiling sonar. We have also converted a small ROV system, the C'Cat, into a small AUV system. We have placed on it the Imaging sonar and DVL which will be used on the larger SAUVIM vehicle. This will allow us to test our algorithms in the University of Hawaii pool which again will alleviate the expense of the large system testing. The Magellan robot and C'Cat ROV is shown in figure LN-7 and figure LN-8, respectively.



Figure LN-1 – Deployment of the SAUVIM vehicle



Figure LN-2 – SAUVIM submerges into the ocean environment

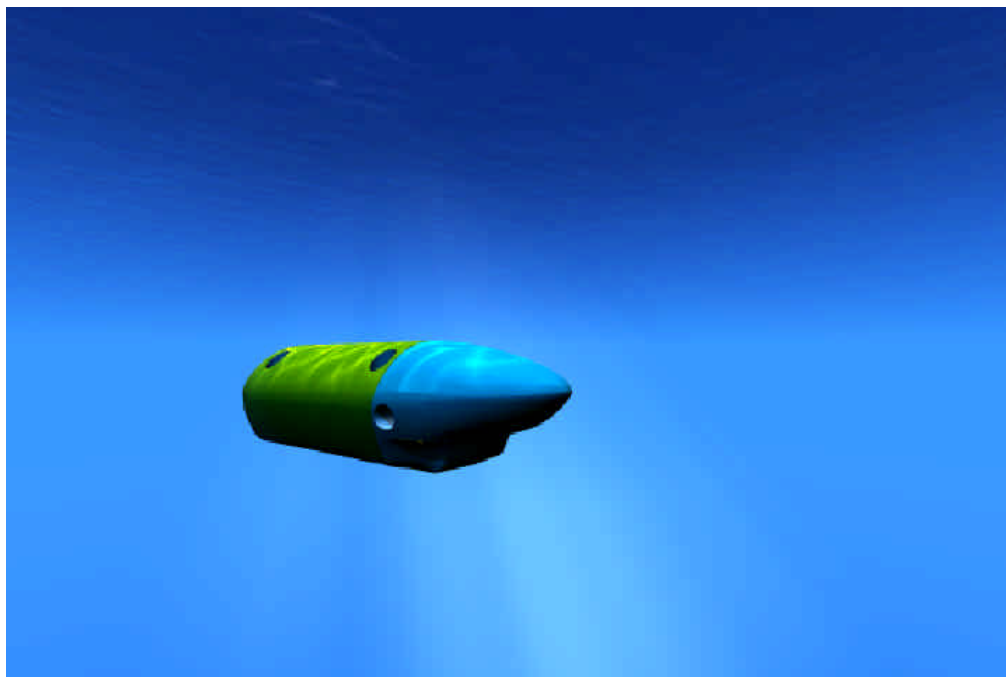


Figure LN-3 – SAUVIM is under the surface of the ocean and must navigate to target.

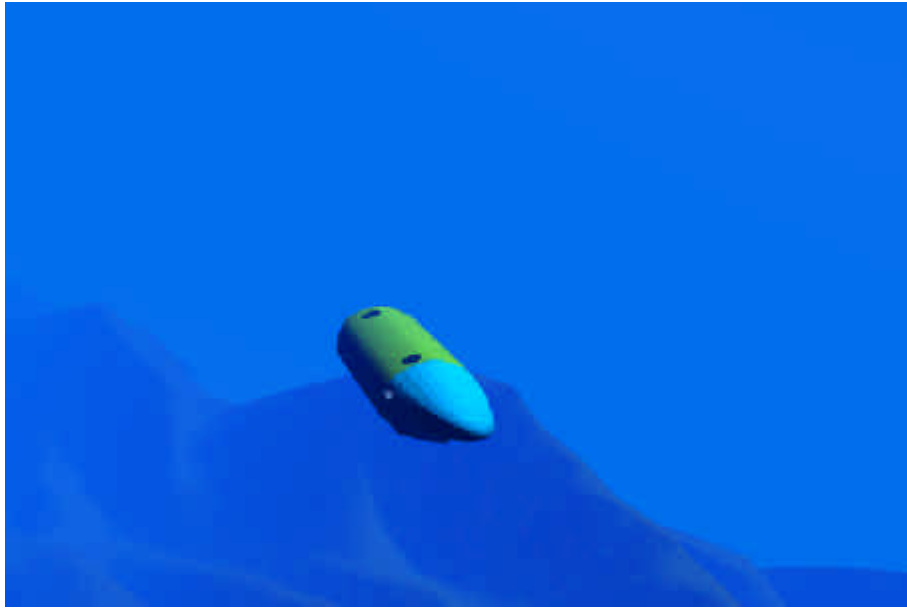


Figure LN-4 – SAUVIM approaching bottom of the ocean and determines position



Figure LN-5 – SAUVIM moving along the surface of the ocean floor.

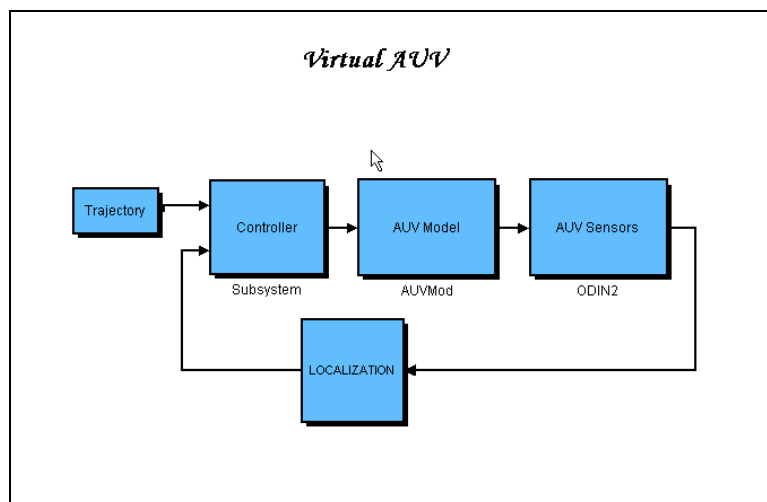


Figure LN-6 – SAUVIM simulation of the Navigation system using Simulink and OpenGL.



Figure LN-7 – The Magellan robot with a SICK scanning laser system

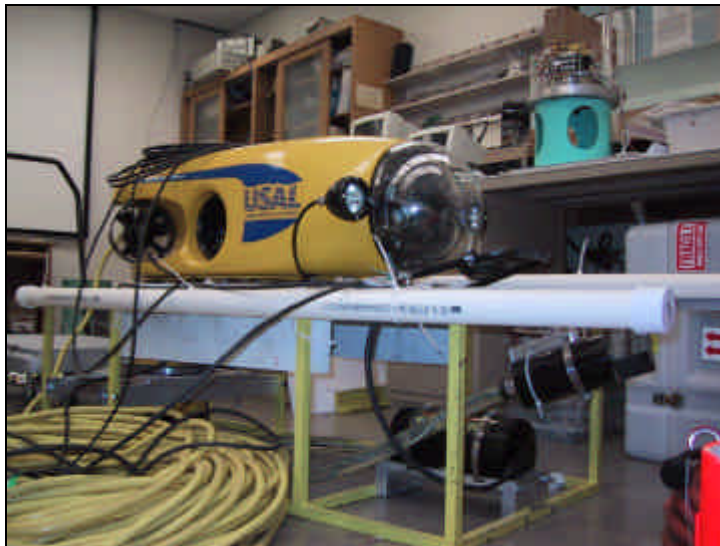


Figure LN-8 – The C'Cat ROV with an Imagenex imaging sonar and a RDI DVL.

- Auran, P.G. and Silven, O. 1995. "Ideas for Underwater 3D Sonar Range Sensing and Environmental Modeling." *CAMS'95, May*, Trondheim, Norway, pp. 284-290.
- Austin, T.C. 1994. The application of spread spectrum signaling techniques to underwater acoustic navigation. *Proceedings on IEEE AUV Technology*, pp. 443-449.
- Austin, T.C., and Stokey, R., Relative Acoustic Tracking, *Sea Technology*, Mar. 1998, pp. 21-27.
- Babb, R., AUV Navigation for Underwater Scientific Surveys. *Sea Technology*, Dec. 1990. pp. 25-32.
- Bergem, O., Andersen, C.S., and Christense, H.I. Using Match Uncertainty in the Kalman Filter for a Sonar Based Positioning System. In *Proceedings of the Eighth Scandinavian Conference on Image Analysis*, volume 1, pages 405-410, Tromsø, Norway, 1993. IAPR.
- Black, M.R. and Butler, B. 1994. Arctic ocean trials of trackpoint ultrashort baseline acoustic positioning system. *Proceedings on IEEE AUV Technology*, pp. 297-302.

- Blidberg, D. R. and Turner, R. 1995. Mission planner, In J. Yuh (ed.). *Underwater Robotic Vehicles: Design and Control*, Albuquerque: TSI.
- Barbour, N., and Schmidt, G., Inertial Sensor Technology Trends, *IEEE Symposium on Autonomous Underwater Vehicle Technology*. Aug. 1998. pp. 55-62.
- Carpenter, R.N., Concurrent Mapping and Localization with FLS, *IEEE Symposium on Autonomous Underwater Vehicle Technology*. Aug. 1998. pp. 133-148.
- Cristi, R., Caccia, M., Veruggio, G., and Healey, A.J. 1995. A Sonar Approach to AUV Localization. *CAMS'95, May*, Trondheim, Norway, pp. 291-298.
- Goldstein, M.S., and Brett, J.J., Precision Gravity Gradiometer/AUV System, *IEEE Symposium on Autonomous Underwater Vehicle Technology*. Aug. 1998, pp. 167-174.
- Hutchison, B.L., Skov, B.E., Bradley, S.E., Rowe, F.D., A System Approach to Navigation and Piloting Small UUV's, *AUV 90*, pp. 129-136.
- Johnson, A.E. and Hebert, M, Seafloor Map Generation for Autonomous Underwater Vehicle Navigation, *Autonomous Robots*, Vol. 3, 1996, pp. 145-168.
- Jorgensen, K.V., Grose, B.L, and Crandall, F.A, Doppler Precision Underwater Navigation, *Sea Technology*, Mar. 1994, pp. 63-66.
- Leonard, J.J., Bennett, A.A., Smith, C.M., Feder, H.J.S., Autonomous Underwater Vehicle Navigation, *MIT Marine Robotics Laboratory Technical Memorandum*, 1998.
- Smith, S.M., and Kronen, D., Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation. 1997. pp. 714-720.
- Stambaugh, J.S., and Thibault, R.B., Navigational requirements for Autonomous Underwater Vehicles. *Navigation: Journal of the Institute of Navigation*. Vol. 39, No. 1, Spring 1992. pp. 79-91

High-Level Control (HLC)

Project Leader(s): Dr. Tae Won Kim
Personnel: - none -
Past Project Leader(s): Dr. Junku Yuh, Dr. Kazuo Sugihara & Dr. Song K. Choi
Past Personnel: Mr. Side Zhao, Ms. Jing Nie & Mr. Zhi Yao

Objectives

The objective is to develop an event-driven supervisory control module that minimizes human involvement in the control of underwater manipulation tasks.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

From a human viewpoint, a mission is always composed of two parts: the goal and the method of accomplishment. Other words, "what do I need to do" and "how do I do it". Following this strategy, a new architecture of vehicle control, named Intelligent Task-Oriented Control Architecture (ITOCA), is developed for SAUVIM. ITOCA is an effective and efficient operation running on the VxWorks real-time operating system (RTOS) environment. ITOCA is four layers: Planning Layer; Control Layer; Execution Layer and Evaluation Layer. Every mission is broken into many smaller missions, and the simplest mission is considered a task. The combination of different tasks in different sequences accomplishes various missions.

The planning layer consists of the plan supervisor and task supervisor. The plan supervisor decomposes the given mission into the sequence of several sub-goals. The task supervisor sequences task modules for each sub-goal. Common sub-goals and task modules are pre-programmed in database along with a world model that is continuously updated using sensor data. The control layer handles various control actions, such as the adaptive and intelligent motion control and the manipulator force/position control, based on the sequence of tasks. The execution layer includes permanent tasks independent of specific missions. These tasks are interrupt handling, shared memory control, navigation sensor handling, servo control, and communication. The evaluation layer checks the status of the vehicle/manipulator performance by comparing actual performance based on sensor feedback with desired performance given by the plan supervisor and task supervisor. This layer makes a decision for the modifications of sub-goal planning and task sequences or the suspension of the mission from fatal errors.

Advancements, in the SAUVIM software modules of motion planning and control, include: off-line and on-line motion planning modules based on GA; a low-level control module using a new non-regressor based adaptive control scheme; and a redundancy resolution control module for the vehicle and manipulator system.

Future Tasks (Phase II Tasks)

- Refinement of ITOCA and development of generic ITOCA command language.
- Software implementation of preliminary version of ITOCA.
- Testing of ITOCA and refinement..
- Implementation in to the SAUVIM software.

Predictive Virtual Environment (PVE)

Project Leader(s):	Dr. Song K. Choi & Mr. Scott Menor
Personnel:	Mr. Alexander Nip, Mr. Alberto Brunete, Ms. Tammy Yamauchi & Mr. Jeffery P. Yee
Past Project Leader(s):	Dr. Kazuo Sugihara & Dr. Stephen Itoga
Past Personnel:	Mr. Zhenyu Yang, Mr. Jiwen Liu, Mr. Steve Timcho, Ms. Lori Yokota, Ms. Jennifer Saito, Mr. Brandon Higa & Mr. Xiandong Su

Objectives

This sub-project aims at applying virtual reality (VR) to the construction of the predictive virtual environment that presents a supervisor with the current situation of SAUVIM as accurately and realistically as possible.

The basic four objectives are:

- To develop software for data fusion of map data and online sensory information;
- To develop software to smooth out a jerky virtual environment due to delayed information from limited bandwidth of communication;
- To develop a learning algorithm for prediction of the current situation from the delayed information acquired by SAUVIM; and
- To integrate the above software modules and interface them with SAUVIM communication for the PVE.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

I. INTRODUCTION

Design, fabrication and analysis of autonomous underwater vehicles (AUVs) are complex, expensive, and time-consuming. The unpredictable and hazardous underwater environment is extremely unforgiving and remote. Due to limitations in communication, an AUV must operate in fully or near autonomous modes. This requirement immensely complicates the development, diagnosis, and evaluation of AUV's many subsystems. To ensure reliability in these systems, it is necessary to test and retest them under severe or extreme conditions in a controlled laboratory environment before operational or sea trial deployment. In addition, since many military, scientific, and commercial tasks in open-oceans often require multi-national participation, it becomes a necessity to rehearse these operations before the actual operation, thus establishing operational strategy and ensuring the success of the operation without releasing proprietary or secured materials. Therefore, the **Distributed Virtual Environment Collaborative Simulator (DVECS)** becomes the ultimate tool for these needs.

Several universities have conducted research in the graphic simulator arena. To mention a few, they are: (a) the Naval Postgraduate School and their NPS AUV Integrated Simulator for their NPS AUV [1]; (b) the University of Tokyo and their Multi-Vehicle Simulator for their Twin-Burger AUV [2]; and (c) the Autonomous Undersea Systems Institute and their Cooperative AUV Development Concept [3]. These systems were developed on IRIX (a and b) and Intel based Win32 (c) environments running OpenGL protocols.

Much of DVECS is based and developed on the graphic test platform architecture for AUV by Yuh, Adivi and Choi [4], and the SGI GL based 3-dimensional graphics by Choi, Yuh and Takashige [5].

DVECS utilizes a similar architecture of a combined hierarchical and heterarchical structure of the previous test platforms along with UT's DVECS simulation system; but DVECS differs by utilizing a variety of different wireless communications methods – RF links, commercial cellular telephones, wireless Ethernet, wireless LAN, and asynchronous transfer mode – for data transfer.

II. OVERVIEW

At this stage, we have completed work on the initial design and implementation of DVECS and have begun preliminary work on a Next Generation DVECS (DVECS TNG (The Next Generation)). The bulk of this report refers to the completed DVECS and not DVECS TNG.

DVECS is an integrated environment for mission planning, simulation and monitoring. It includes the ability to generate a 3D simulation with realistic physical modeling and can emulate communication and sensor systems present on real vehicles.

There are several fundamental design issues with DVECS that have, along with a new design goal of producing a central server-less simulation environment, prompted us to begin development on DVECS TNG. First, DVECS communication uses TCP/IP socket streams and requires continuous streams for the duration of a simulation. Second, extending the current communication protocol to add new features requires rewriting, recompiling and retesting several communication modules. Finally, reconfiguring DVECS, in its current incarnation, typically requires a substantial portion of the program to be recompiled.

TCP/IP streams are intended to provide reliable, in-order two-way communication between two programs. This, in itself, isn't a major issue as reliable communication is desirable as is receiving information in the order it was sent. The problem it creates, however, is that if any single packet is lost, there is considerable overhead required to detect and recover from that loss and no future packet may enter the system before the missing packet has been successfully received. This has the consequence of causing a noticeable delay and synchronization problems between components of the simulation. On small scale, closed and dedicated networks, such as LANs, dropped packets are rare so this isn't a significant issue BUT on larger scale, open and undedicated networks, such as the Internet, this can be a significant problem and is unacceptable for real-time use.

Communication between DVECS TNG components will be based on a Remote Procedure Call (RPC) like protocol implemented using Simple Object Access Protocol (SOAP), an XML data exchange format that can be tunneled through a variety of networking protocols, including HTTP, SMTP and TCP/IP streams. SOAP has many benefits over our current custom stream based protocol. Extending or modifying the communication protocol will simply require editing an XML document and could be done on the fly without even needing to re-launch any of the DVECS TNG components. The RPC-like model will eliminate the issues with communication delays resulting from recovery from dropped packets and will make transactions atomic, 'all-or-none' events, avoiding bugs and synchronization issues due to incomplete or corrupt communication.

DVECS TNG will be written entirely in Java and should be able to execute on almost any currently available computing platform with minimal 3D hardware, CPU performance, memory and networking requirements. Using the JVM rather than native executables does incur some performance penalty,

particularly with respect to graphics performance, but our initial tests indicate that communication performance can actually be faster in Java than un-optimized, compiled C/C++ code and we have been able to achieve acceptable frame rates on modest hardware so we believe that the benefits of using Java justify the performance penalties.

Finally, the design decision to use a central server-less, Peer-to-Peer (P2P) architecture for DVECS TNG will afford greater flexibility and stability in simulations. Ad hoc virtual networks would be constructed with efficient routing between nodes and plug-able prediction and interpolation modules to compensate for communication delays and bandwidth limitations.

III. COMMUNICATION

Communication within DVECS

DVECS is a distributed virtual environment for collaborative simulation and monitoring. Communication between components within DVECS is based on a client-server model using TCP/IP socket streams.

DVECS Socket

TCP/IP uses ‘sockets’ as connection points that provide endpoints for communication within and between systems on a network. Sockets can have any 16-bit, unsigned integer port number, providing 65,536 possible connection endpoints for a any given machine. Communication within DVECS takes place over socket ‘streams’ (SOCK_STREAM), reliable, 2-way connections. Alternative communication models exist, including UDP Datagrams (DGRAMs), which use small packets and are transmitted without explicitly making connections and with no guarantee of arrival. DGRAMs have a significant advantage over streams, in that they require significantly less network and computational overhead than streams and, unlike streams, they do not have a significant transmission rate penalty if a few packets are lost, making them more appropriate for applications such as streaming telemetry data, audio or video, where timely delivery is more important than reliable transmission of *every* packet. Therefore, the choice to use streams in DVECS provides reliable transmission of data, but at the price of causing significant delays over unreliable networks or large distances, effectively limiting the DVECS core applications to distribution over small, reliable networks. To overcome this limitation, for long distance transmission, DVECS includes ‘remote’ programs that use UDP DGRAMS instead of TCP Streams, allowing reasonable performance even when components, such as vehicles, are distributed around the world (this is discussed further in the section on ‘remote’ communication).

DVECS Server

Servers within DVECS are programs that, as their name implies, provide information and services to other programs. A general DVECS server, like any TCP/IP SOCK_STREAM server, is first to attempt to create a socket and ‘bind’ it to a specific port. If the server cannot create the socket and bind it to the desired port, it returns error messages and quits. If the socket is successfully created and bound to the desired port, then the server ‘listens’ to the port until another program, a ‘client’, makes a connection request. Once a client makes a connection request, the server sets up another socket just for communication with that particular client and the server ‘forks’ off another process to receive and respond to client requests.

DVECS Client

As mentioned in the previous section, an DVECS client is a program that makes TCP/IP SOCK_STREAM connections with a DVECS server. Many programs within DVECS include both

client and server functionality, that is that they will be a client for one set of programs and a server for others. DVECS clients include programs such as user interfaces, managers and entities.

DVECS Daemon

DVECS ‘daemons’ are programs that run in the background and connect to DVECS sessions to provide a variety of services. Generically, daemons are programs that provide infrastructure support for DVECS virtual worlds and the DVECS universe. Daemon activities include establishing, maintaining and terminating communication connections, providing simulated physical realism such as gravity, viscosity, ultrasonic range finders and communication links, cameras and other sensors.

DVECS Attendant

DVECS ‘attendants’ are omniscient observers of DVECS virtual worlds. Attendants can view any portion of the world, including mission mode, external vehicle states (ie – thruster / actuator data; but not the internal processing / control state of vehicles) and other attendants. Attendants are primarily designed for use as generic user interface modules that can be run from anywhere on the internet with a route to the machine where a DVECS virtual world is running.

DVECS Managers

DVECS uses a distributed daemon control system to construct virtual worlds. By design, a central daemon, ‘universe’ is called (by inetd) when a connection request is made to a specific port (9800, by default). A TCP/IP socket stream connection is then established between a component of DVECS and the ‘universe’. Within a DVECS universe, there can be several virtual ‘world’s, each of which can contain its own set of ‘entities’, including such things as vehicles, obstacles and representations of real world objects. In DVECS’s design, it is assumed that only one ‘world’ exists per machine. When a ‘world’ is created, it communicates with the ‘universe’ daemon and listens to a port (7000) for TCP/IP connection requests. Realism is added to DVECS by means of specific ‘managers’, daemons that provide information and services to entities within an DVECS virtual world. Currently, DVECS world managers include ‘collision’, which provides collision detection for entities within a world, ‘vision’, which provides sensor (SONAR / camera) data, ‘ucomlink’, which provides simulated ultrasonic communication between entities within a world and ‘urfinder’, which provides simulated SONAR ranging data between entities and ‘vskernel’ / ‘vsrespect’, which manage vehicle states.

The DVECS ‘Universe’

At the root of DVECS’s design hierarchy is the ‘universe’, a daemon that establishes and maintains relationships between subcomponents of a distributed DVECS environment. Universe is called whenever a connection attempt is made to a specific port (9800, by default) and universe sets up appropriate sub connections between DVECS components and passes state and other information between managers, observers and other entities.

A DVECS Virtual ‘World’

The next hierarchical level is the DVECS virtual ‘world’. Within a single DVECS universe, there may be several, independent virtual worlds. By design, only one virtual world may exist per machine, however the individual components within the world can be distributed anywhere in the world as long as there is an Internet route between them.

DVECS Managers: vskernel and vsrespect

Within an DVECS virtual world, vehicle state is controlled and maintained by two managers, Vehicle State Kernel (vskernel) and Vehicle State Respective (vsrespect). Vskernel and vsrespect both provide information and issue state change requests for the current mission mode (pause; run or reset), vehicle

position / orientation (x, y, z, roll, pitch, yaw) and (vsrespect) provides information specific to each entity (such as thruster state; sensor data; etc). Information and services provided by vskernel and vsrespect are used by 'attendants', observers within an DVECS world and by user interfaces to monitor and control DVECS virtual worlds.

DVECS Managers: System Config ('sconfig')

The DVECS System Configuration manager logs into the DVECS universe and provides system configuration services to DVECS virtual worlds.

DVECS Managers: Common Sense ('csense')

Communication between DVECS components is facilitated by the 'csense' manager. Csense provides a database of known entity types and gives a communication interface between them. With csense, communication is limited to a string of numbers which themselves contain no information about their significance but the numbers can be mapped to their actual meaning relative to DVECS entities. This ability is particularly useful in bandwidth-limited environments, including such things as underwater ultrasonic communication links.

DVECS Managers: Collision

3D entities within DVECS are given solidity by the 'collision' manager. Collision detects collisions between entities within an DVECS virtual world and uses impact surface normal vectors to compute simulate elastic collisions and prevent objects from passing through one another.

DVECS Managers: Ultrasonic Communication Link (ucomlink) and Ultrasonic Communication Quality (ucomqual)

Ucomlink provides simulated underwater ultrasonic communication links between vehicles. Propagation delay, signal attenuation and other effects of transmission through liquid media can be simulated. Ucomqual provides simulated communication quality decay to underwater ultrasonic communication links provided by ucomlink. Effects such as signal loss and noise can be modeled by ucomqual and a reasonable approximation of the limitations of underwater ultrasonic communication links can be provided by the combination of ucomlink and ucomqual.

DVECS Managers: Ultrasonic Range Finder (urfinder)

The DVECS manager, 'urfinder' provides simulated underwater SONAR range sensor data for entities within DVECS virtual worlds. Urfinder data can be as simple as minimal distance between a SONAR sensor and an object in direct line with it or as complicated as is necessary and possible given the constraints of algorithm design and compute and network time requirements.

DVECS Managers: Vision

The DVECS manager, 'vision' provides simulated optical camera viewpoints for use as sensor data for underwater vehicles and for use as viewpoints for observers monitoring DVECS worlds.

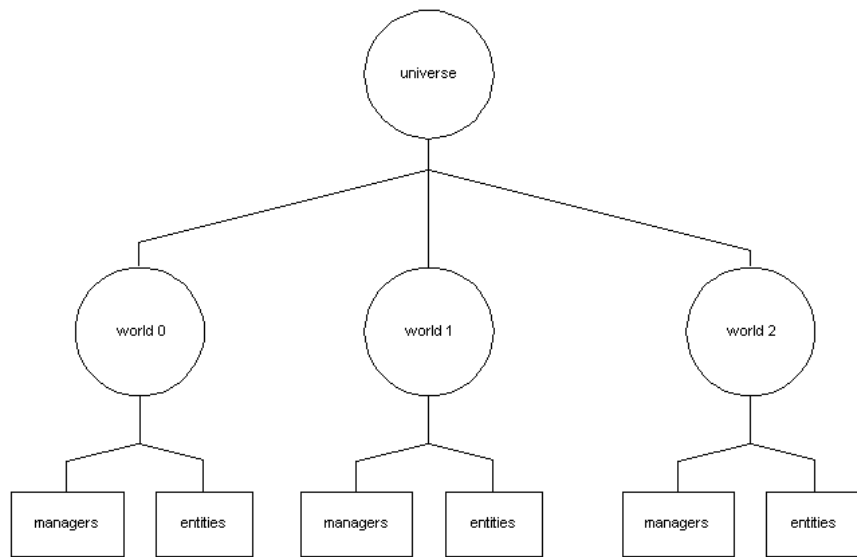


Figure PVE-1. DVECS hierarchy (with 3 virtual worlds)

Communication between vehicles and DVECS

Introduction.

One key feature of the DVECS is that multiple simulated and real systems must be able to interact with one another. In order to accomplish this, communication between the various components is essential. Within the DVECS environment, TCP socket streams and UDP socket datagrams convey messages from one component to another. These messages or data are used in the various sub-systems - graphics module, numerical module, navigation module, etc. - to exactly determine the position of the AUV within the virtual environment. In addition, a simple network reflector interface that allows a simulated entity within the DVECS to be controlled by telemetry data from an external component is used to garner data from the UH AUV, ODIN (Omni-Directional Intelligent navigator) [Choi95]. This interface can be used to monitor a physical or simulated vehicle either directly over a network or with an additional software reflector if the communications system does not include network support.

Wireless communication between a vehicle in the field (ODIN) and the DVECS is accomplished by a serial data link that connects to an intermediate interface that relays data between our TCP packets and the serial link. It would also be possible by a direct network connection over a wireless link with an interface either on board the AUV or at the test site. This wireless link is under study at present time. Either system is completely transparent to the DVECS and the communications delay created by adding additional reflectors is typically negligible.

Typically, for ODIN, the transfer of data between the test vehicle and the monitoring laptop is via a radio frequency (RF) modem. The transfer of data from the monitoring laptop to the DVECS is via a wireless Ethernet connection nowadays, but it will be possible by wireless cell phone in the future.

For interactive vehicle testing in a hybrid, simulated synthetic environment, combining the actual and the virtual sensor measurements generates the synthetic sensor data. This is easiest for range data, such as from sonar, in which the synthetic range is simply the minimum of the actual and virtual sensor data.

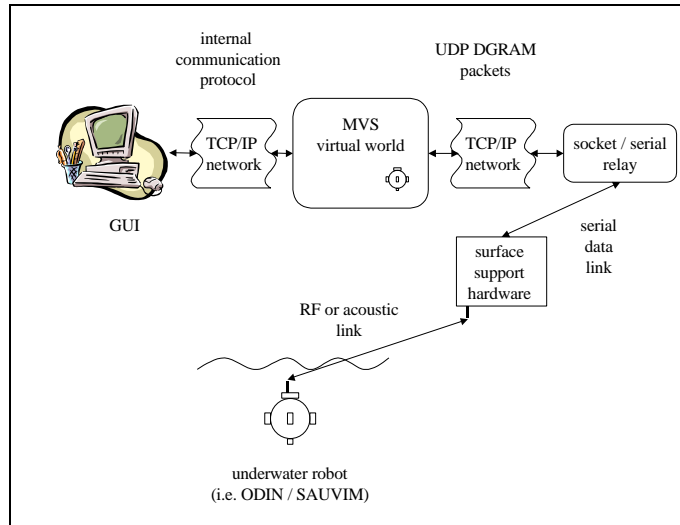


Figure PVE-2. DVECS hierarchy (with 3 virtual worlds)

Finally, multiple simulated or physical entities can interact over a networked environment without requiring them to share code or knowledge of each other's capabilities so proprietary algorithms can be tested in a common environment without making them public. This allows for collaboration from many different sectors of the underwater community that wish to "evaluate" their AUV in conjunction with pre-tested AUVs, as shown in the next figure.

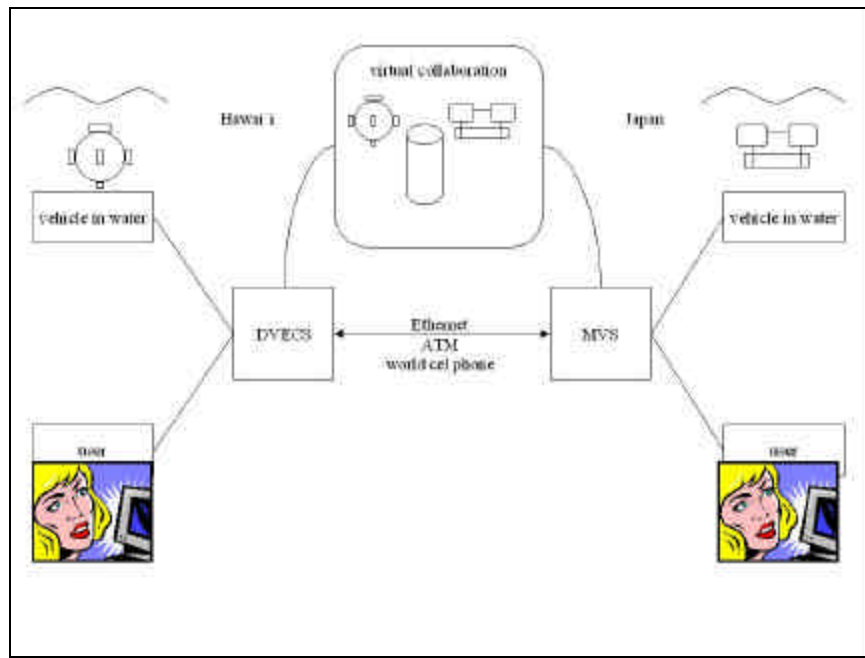


Figure PVE-3. DVECS hierarchy (with 3 virtual worlds)

Packet

The packets used in this communication are UDP datagrams with the structure:

<	X	Y	Z	rx	ry	rz	>
---	---	---	---	----	----	----	---

The reason of choosing UDP rather than TCP is that the cost of losing a packet is much more smaller than the cost of communication latency. The next packet can replace the lost one, or using a filter it can be predicted.

Where x, y and z are the position and rx, ry, and rz the roll, pitch and yaw. Every packet starts with the symbol “<” and ends with “>”. All parameters are floats converted to strings, so the length of the packets varies from one packet to another depending on the values that x,y,z,...etc take.

Communication.

Every certain amount of time (about 0.2 seconds) the vehicle (ODIN) sends its position and orientation to a laptop via serial communication. This information is then put together in a UDP packet and sent via network to a computer (for example “o2”, an sgi workstation). The reason why the packets are not send directly to the computer which the GUI is running in is to free it from this tasks, because the graphical tasks consumes a lot of cpu time. In o2 the program tbRemote is running. At the first of moment of running this program it logs into DVECS (that is running for example in visual, another sgi workstation) and establishes a communication link between both machines.

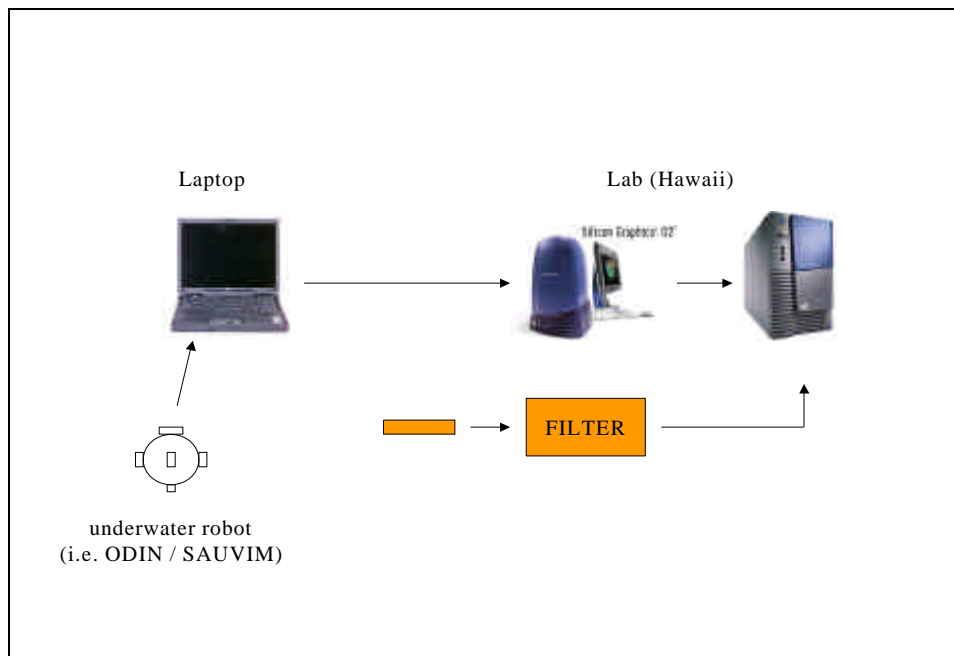


Figure PVE-4. DVECS hierarchy (with 3 virtual worlds)

When a packet arrives to o2, it is filtered and then send again to visual, where the coordinates are extracted and a new position and orientation for ODIN is set. The filter is described in another section of this paper, but basically it checks that the values are within some limits (there have been no

transmission errors). If any error has occurred, a predictive one interchanges the wrong value, based on the previous packets received.

IV. SIMULATION

V. INTERACTION WITH REAL WORLD ENTITIES

In addition to being a simulation environment, DVECS, in its current and next generation incarnations, allows real world entities to have a presence within the simulation environment, which simulated entities can respond to and, in principle, semi-synthetic sensor data can be generated to allow entities existing in the real world to interact with simulated entities.

VI. USER INTERFACE

The user interface is a set of classes used by both the graphical and the text user interfaces. It includes many functions used to interact with DVECS, like log in and out, initialization, join and leave a world, create a new user, add and remove agents, deal with some of the managers (collision, comlink, respect...etc), send messages, send and get world information, set passwords...etc. Some of this functions are overridden by the classes in the GUI and TUI. The user interface is a set of classes used by both the graphical and the text user interfaces. It includes many functions used to interact with DVECS, like log in and out, initialization, join and leave a world, create a new user, add and remove agents, deal with some of the managers (collision, comlink, respect...etc), send messages, send and get world information, set passwords...etc. Some of this functions are overridden by the classes in the GUI and TUI.

Purpose.

The purpose of the GUI is to create an environment where the user can easily interact with DVECS. Some of the tasks that it is possible to perform are:

- control the vehicle as if watching it directly.
- get and send data from/to the vehicle in a simple way.
- create obstacles and different environmental situations (fog, light...etc)

Description

The DVECS software is a multi-layered C++ program modularized by its subsystems and utilizes the inheritance properties and so is GUI. It uses OpenGL graphics libraries to generate the background, vehicle and obstacles, and uses Open Inventor 3D toolkit protocols to create the 3-dimensional, virtual images.

The DVECS consists of three windows. They are the Main View Window, Main Menu Bar Window, and Main Control Panel Window (previous figure).

The Main View displays the virtual environment with the vehicle being tested. The background environment can be modified to represent an area being used, such as the UH dive well, or can be used pre-mapped seafloor data to represent a specific deployment area. The window allows instantaneous change in viewpoints and magnification.

The Main Menu Bar allows access to the background, multiple vehicles or obstacles. This is a simple, pull-down menu layout allowing for access to a specific object's properties - dimension, location and attributes.

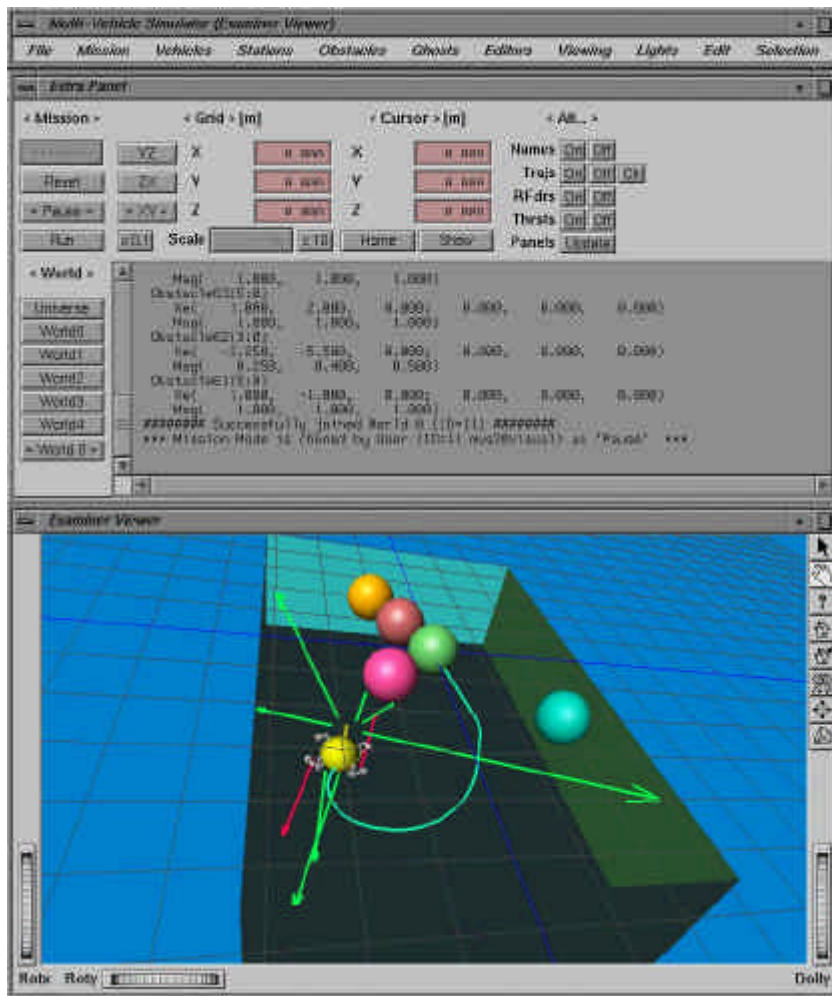


Figure PVE-5. DVECS hierarchy (with 3 virtual worlds)

The Main Control Panel allows access to different environments or simulations, mission controls (start, stop, pause), placement of grid layout, modification of object labels, and control of the sensor and thruster data. It also allows monitoring of system messages, warnings, etc.

How it works

The GUI is composed by several programs, which after compilation and linking are put together into the DVECSGUI2 executable file.

The first one, the one that includes "main", is DVECSGUI.c. It checks the command line for the right number of parameters, starts openInventor, logs into the managers and then enters a loop which lasts until the program is finished.

The graphical user interface, as any other program in DVECS, has to login into the managers, which can be running in a different machine. That is why it is necessary to specify the host name and the port where you want to connect to. The command line is: DVECSGUI2 [host [port]].

Within this loop, many functions and procedures are called. To explain what happen here, a description of the classes will be given.

DVECSGraphicUI: this is the most important class. It includes most of the functions used to interact with DVECS, like log in and out, initialization, join and leave a world, create a new user, add and remove agents, send messages, send and get world information, set passwords...etc. One interesting function is init(), which creates three child processes to run each of the different windows that form the GUI.

DVECSGuiAlbac, DVECSGuiP150, DVECSGuiPW45, DVECSGuiR1, DVECSGuiTB1 are classes for each of the vehicles.

DVECSGuiMonolith, DVECSGuiStation: Station class.

DVECSGuiDefs: includes definitions and information used by other classes.

DVECSGuiMenu: menu bar.

DVECSGuiNB: Node Base class for DVECSGUI program

DVECSGuiPoly, DVECSGuiSeabed, DVECSGuiSimple, DVECSGuiGhost, DVECSGuiObstacle, DVECSGuiSeabed, DVECSGuiGCombi, DVECSGuiGLines, DVECSGuiPB, DVECSGuiPP, DVECSGuiVehicle: these classes allows the user to create objects, ghosts, seabeds...etc and to set their properties and parameters. Next two pictures show examples of a seabed and several objects.

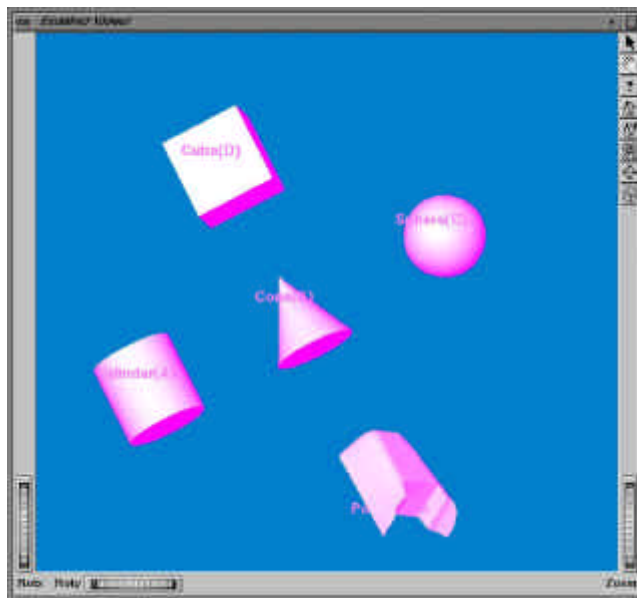


Figure PVE-6. DVECS hierarchy (with 3 virtual worlds)

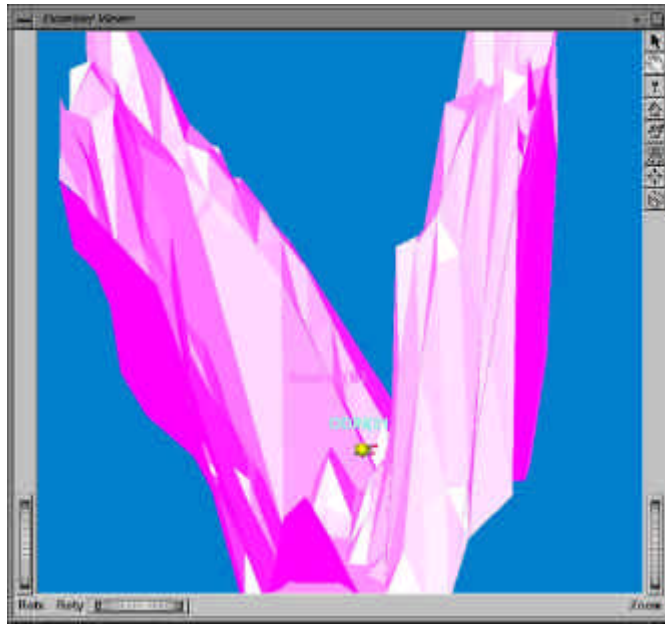


Figure PVE-7. DVECS hierarchy (with 3 virtual worlds)

DVECSGuiWP: World panel information and definitions.

DVECSGuiSmB: Submenu Base class for DVECSGUI program

DVECSGuiRF: Range Finder. In the graphical window it represents the green arrows in (see next picture).

DVECSGuiRexec: Allows execution in a remote host.

DVECSGuiThs: Thrusters. In the graphical window it represents the red arrows (see next picture).

DVECSGuiUser: User Panel

DVECSGuiUtil: Basic utilities for the gui.

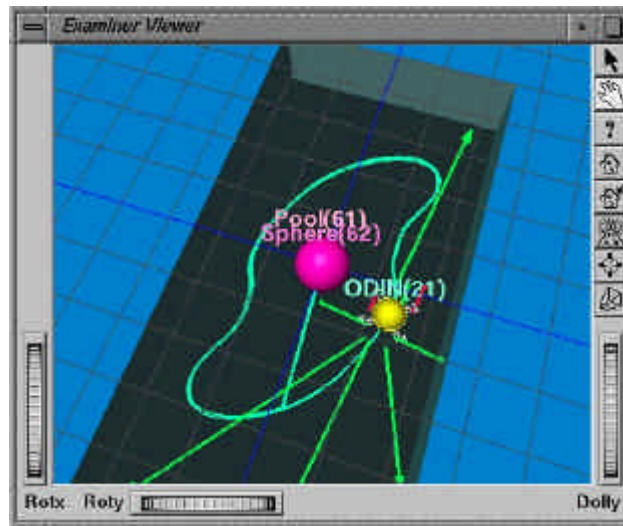


Figure PVE-8. DVECS hierarchy (with 3 virtual worlds)

Text User Interface (TUI)

The text user interface performs the same tasks as the GUI, but in a text window. This allows remote users to access the world information without the need of having a graphical environment. It presents the same information than the GUI but in numbers.

```
winterm
MVS: Multi-Vehicle Simulator >>==== World 1 (Run) ===== 06/28 21:04

[

---(Input Window)---
mmode Run
done

---(Guidance Window)---
Mag( 1.000, 1.000, 1.000)
Obstacle62(3:0) Xc( -3.250, -5.500, 0.000; 0.000, 0.000, 0.000)
Mag( 0.250, 0.400, 0.500)
Obstacle61(5:0) Xc( 1.000, -1.000, 0.000; 0.000, 0.000, 0.000)
Mag( 1.000, 1.000, 1.000)
*** Mission Mode is changed by mvs2(12) as 'Reset'. ***
*** Mission Mode is changed by mvs2(12) as 'Pause'. ***
*** Mission Mode is changed by mvs2(12) as 'Run'. ***

---(Receiver Window)---
```

Figure PVE-9. DVECS hierarchy (with 3 virtual worlds)

Other programs.

Vehicles

There are different kinds of vehicles: ODIN, PW45, Pteroa150, R1... Each of these vehicles can be running in the water or as a simulation. Any of them can join a world in DVECS and interact with others vehicles and objects, no matter if they are in the water or not. The important thing is that in DVECS all vehicles have the same physical properties.

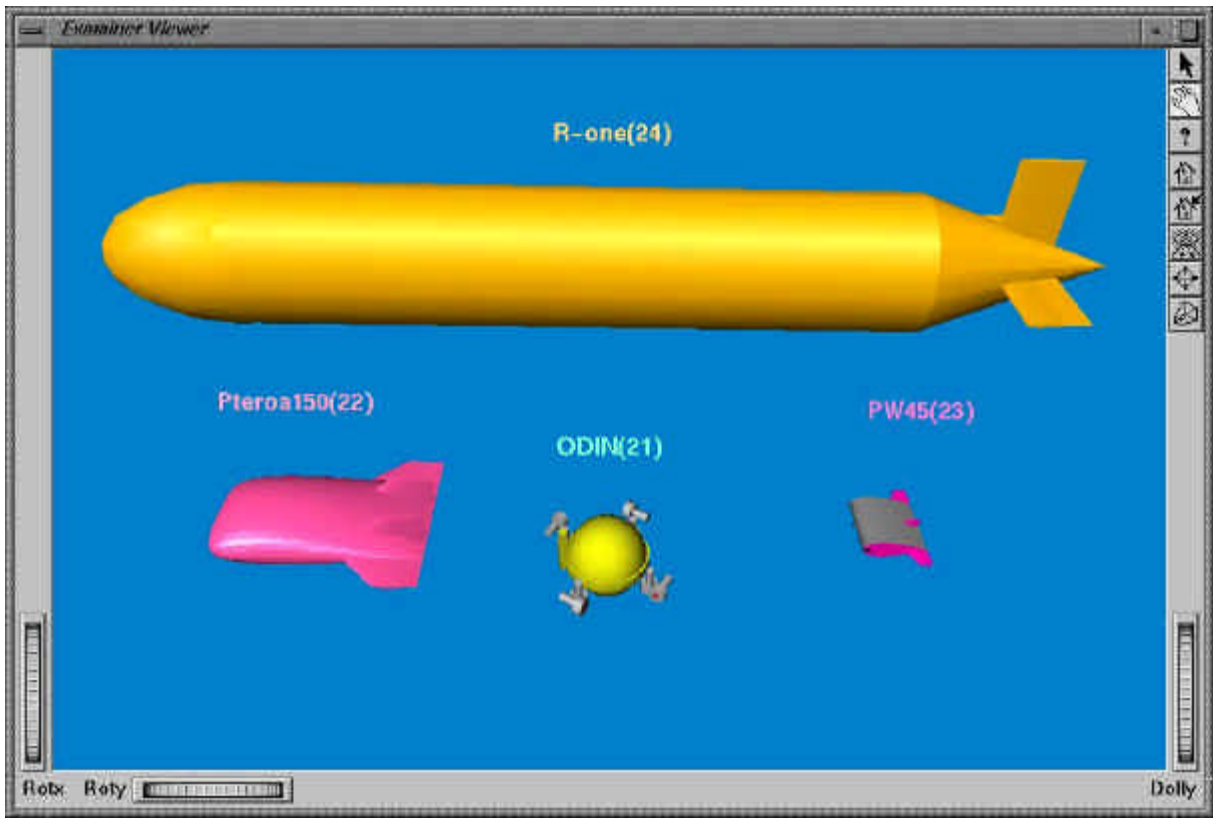


Figure PVE-10. DVECS hierarchy (with 3 virtual worlds)

There are two major parts in every vehicle: the graphics and the behaviour. The graphics are designed in the gui, as mentioned before (DVECSGuiP150, DVECSGuiPW45, DVECSGuiR1, DVECSGuiTB1). The vehicles are drawn using basic shapes to build more complicated models.

The behaviour of the vehicles depends on the situation. If DVECS is running in a simulation mode, each vehicle has a behaviour that describes its trajectory. This is described on `tb1nav`, `r1nav`, `pw45nav`, `p150nav`.... If DVECS is in remote mode, the position and orientation is received from the real vehicle via the programs `tbRemote`, `pw45remote`...etc, as described in other part of this report.

All the vehicles have the same structure, and are based in the class `Nav`.

Miscellaneous

The rest of the programs of the gui are not so relevant and have a secondary roll. Most of them provide support for communication or buffering. They are be divided in the following directories:

- Itimer: internal timer for the simulation.
- MB: communications routines within the gui.
- Misc: various functions concerning buffering, remote execution procedures and log files.
- Obstacle: obstacle creation and manipulation.
- Station: Station creation and manipulation.
- Twindow: terminal windows.
- XP: generic classes for entities.

VII. VIDEO SEGMENTATION, FRAME INTERPOLATION AND PREDICTION

VIII. FILTERING

When transmitting data in a network it is probable that some of the packets get lost. If the purpose is to visualize these data in real time, one needs a system to detect when an error has happened, in order to avoid both the computation of the wrong values and the ugly visual effect that it involves.

Other use of the filter is to make the trajectory smoother. Sometimes the difference between samples makes the vehicle jump in the virtual environment. It is possible to use a filter to gradually increment or decrement the difference between samples.

The filtering is composed of two stages: the first one is to compute an estimated value for the position and orientation of the vehicle at that time. The second one is to compare the received value with the estimated value and select the one that is more likely to be the correct one.

The behavior of the filter is similar to a low pass filter, in the sense that it does not allow values that differ significantly from prediction to go through. Instead of eliminating these values, it substitutes them with the estimated values calculated from the previous ones.

Every time that a new packet is received, an estimated value for the position and the orientation is calculated. The received and estimated values are compared, and the closer to the previous value is chosen. However, no matter which value has been chosen, the received value is the one that is stored and used for following computations. In this way, if the vehicle takes a different trajectory than the estimated one, the new values will be stored little by little in the buffer and make the estimated values change as the same time, with only some samples of difference.

The algorithm used to calculate the predictive values is based in the equation of the uniform accelerated movement:

$$x[n] = x[n-1] + \bar{v} \cdot \Delta n + \frac{\bar{a}}{2} \cdot \Delta^2 n$$

in this equation, $x[n]$ represents the array of samples (position or orientation) received, and v and a are the mean values of the velocity and acceleration calculated as follows:

During the first samples (30 in this example), the data received are stored in $x[n]$, one variable for each value of position and orientation. From that moment on, every time a packet is received, new values of v and a are calculated, and an estimated value for the position and orientation is calculated. This value is then compared with real one. If the difference is bigger than a certain value (adjustable, call it sensitivity of the filter), the estimated value is chosen, if not, the real one is chosen.

The figure 15 is a diagram in which it is possible to see a comparison between the received data and the estimated ones. Values out of range are discarded, and sudden changes in the trajectory are smothered.

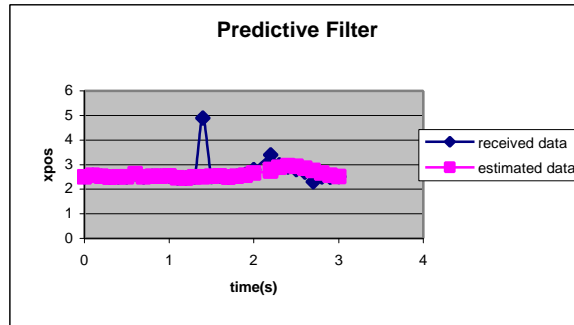


Figure PVE-11. Predictive filter data.

The Figure PVE-11 and Figure PVE-12 correspond to two different filter sensitivity situations. In the first one two kinds of errors are manifested: (1) the first - packet damages - is a big jump in one of the coordinates. This type of errors occur when a TCP/IP packet is lost or damaged; and (2) the second one - physical problems under the water - is an exceptionally fast movement of the vehicle, due to communication noise.

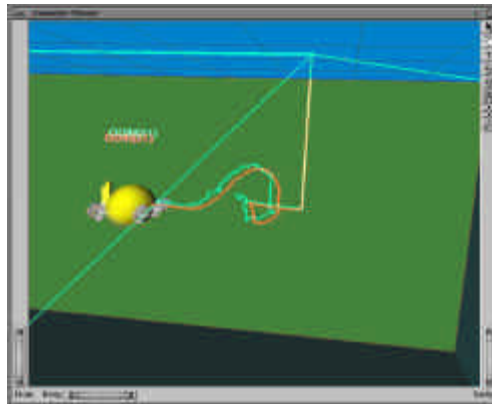


Figure PVE-12. Filtered data with high sensitivity overlaid on unfiltered data.

IX. SENSOR DATA MAPPING

We have developed a simple system to combine sensor data, specifically from a video camera, with surface map data to provide a texture map to enhance realism of our virtual environment.

One naïve approach to texture reconstruction on a surface is to use simple back projection, effectively ray tracing in reverse. One problem with this approach is that it can be too computationally intensive for on-line use. We have implemented a simple and elegant workaround that gives most of the benefits of reverse ray tracing while having the advantage of taking minimal processing time.

The following assumptions were made:

- *Undistorted data* – Ideally, the sensor system should behave approximately like an optical camera in a vacuum.
- *2D sensor data* – Sensor data should be in the form of 2D ‘images’.

- *2D surface map data* – Map data should be in the form of a 2D surface, such as that provided by side-scan SONAR.
- *Position/orientation are known* – The position of the vehicle (and thus the sensor) should be known to reasonable accuracy.

These assumptions lead to a simplification in algorithm that can be overviewed as:

- Capture an image and transmit it to the virtual environment.
- Combine sensor data to determine the approximate position and orientation of the camera in 3-space when the image was captured (details are beyond the scope of this paper).
- Choose a coordinate system for the patch of the surface visible to the camera based on the view plane.
- Apply the camera image as a texture to the visible patch of the surface using the camera's coordinate system.

Several limitations were observed.

- Using captured images directly as texture maps means that only data from one image can be used for a given patch of the map and that processing such as global color correction and distortion removal cannot be done directly.
- Texture map data of surfaces at a large angle to the camera or through murky water will be low quality.
- Surface normal have to be considered so that texture is not applied to surfaces facing away from the camera viewpoint.
- Adequately detecting and ignoring hidden surfaces when applying texture can significantly increase computational requirements.

To limit these shortcomings, several enhancements were made. Rather than use image data directly, standard 2D image processing algorithms can be used to warp captured data and combine several samples into a texture map for a region of the surface; and for each region of the map surface, data can be selected from several samples in order to choose the ‘best’ view of the region.

Selection of a sensor-space coordinate system for the visible patch of a surface map is straightforward. We simply seek to project the map into the viewplane (sensor-space) such that a vector originating from the viewpoint and ending at a point in the patch of the surface map will intersect the viewplane at the point corresponding to the appropriate point in sensor-space for the point on the map. Given a viewpoint, P and a viewplane, L, choose a Cartesian coordinate system with origin at P and with the z-axis perpendicular to L (i.e. – the standard dot product of any vector in L with any vector pointing entirely in the z-direction is identically zero). Now, choose x and y axes such that they correspond with the rectilinear coordinate system used for sensor-space on the view plane. We find that for a point, X, on the surface of the map, the sensor-space parameterization is

$$\tilde{X} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \frac{l}{z} \begin{pmatrix} x \\ y \end{pmatrix}$$

where l is the distance from the viewpoint to the view plane in the z-direction.

We can now directly apply the sensor data to the map surface as a texture map using this parameterization.

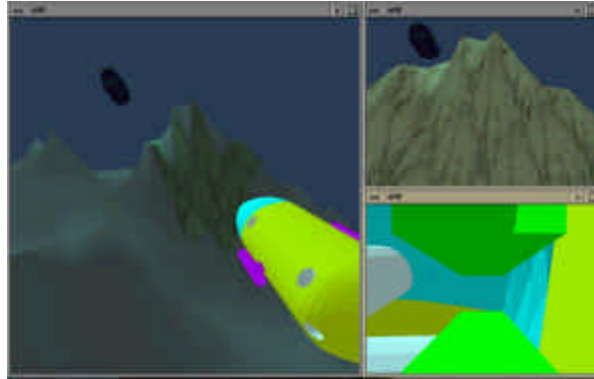


Figure PVE-13. Texture mapping of the image data from a simulation of SAUVIM (upper right corner) onto a surface map (left).

Our method offers many of the benefits of reverse ray tracing and closely approximates it but benefits from significantly decreased computational requirements and can be performed in real time with graphics hardware that is already widely available.

X. REFERENCES

- [1] D.P. Brutzman, Y. Kanayama & M.J. Zyda, "Integrated Simulation for Rapid Development of Autonomous Underwater Vehicles," Proceedings of the IEEE Oceanic Engineering Society AUV92 Conference, Jun. 1992.
- [2] Y. Kuroda, K. Aramaki, T. Fujii & T. Ura, "A Hybrid Environment for the Development of Underwater Mechatronic Systems," Proceedings of the 1995 IEEE 21st International Conference on Industrial Electronics, Control and Instrumentation, Nov. 1995.
- [3] S.G. Chappell, R.J. Komerska, L. Peng & Y. Lu, "Cooperative AUV Development Concept (CADCON) – An Environment for High-Level Multiple AUV Simulation," Proceedings of the 11th International Symposium on Unmanned Untethered Submersible Technology, Aug. 1999.
- [4] J. Yuh, V. Adivi & S.K. Choi, "Development of a 3D Graphic Test Platform for Underwater Robotic Vehicles," Proceedings of the 2nd International Offshore and Polar Engineering Conference, Jun. 1992.
- [5] S.K. Choi, J. Yuh & G.Y. Takashige, "Omni-Directional Intelligent Navigator," Underwater Robotic Vehicle: Design and Control, TSI Press, NM, 1995.
- [6] K. Kawaguchi, C. Ikehara, S.K. Choi, M. Fujita, W.C. Lee & J. Yuh, "Design of an Autonomous Underwater Robot: ODIN II," The Proceedings of the First International Symposium on Intelligent Automation and Control, FRANCE, May 1996.
- [7] S.K. Choi & J. Yuh, "Experimental Study of a Learning Control System with Bound Estimation for Underwater Robots," J. of Autonomous Robots, Mar. 1996.

SAUVIM Design (SD)

Project Leader(s): Dr. Tae Won Kim, Mr. Oliver T. Easterday, Mr. Michael E. West & Dr. Song K. Choi

Past Project Leader(s): Dr. Curtis S. Ikehara, Dr. Junku Yuh, Dr. Mehrdad Ghasemi Nejhad, Dr. Gary McMurtry, Dr. Pan-Mook Lee, Dr. Farzad Masheyekhi, Dr. Gyoung H. Kim & Mr. Gus Coutsourakis

The main technical development of the SD group is described in the following sections: Reliable, Distributed Control, Mission Sensor Package, Hydrodynamic Drag Coefficient Analysis, Mechanical Analysis & Fabrication and Mechanical-Electrical Design. The Mechanical-Electrical Design group will absorb the Mechanical Analysis & Fabrication group in the near future, since most of the research components have been completed.

Reliable Distributed Control (RDC)

Project Leader(s): Dr. Tae Won Kim
Personnel: Dr. Hyun Taek Choi, Mr. Alberto Brunete & Mr. Alexander Nip
Past Project Leader(s): Dr. Pan-Mook Lee, Dr. Curtis S. Ikehara, Dr. Song K. Choi & Dr. Gyoung H. Kim
Past Personnel: Mr. Jang-Won Lee, Mr. Michael West & Mr. Tuan M. Hyunh

Objectives

The objective is to develop a reliable & efficient computing architecture for signal and algorithmic processes of the entire SAUVIM system.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

- Graphic User Interface (GUI) system was designed and implemented.
- Sensor/Actuator setup and calibration modes were designed and implemented.
- Hierarchical S/W structure, incorporating real-time processing and event handling, was upgraded.
- Third party device drivers were interfaced under VxWorks environment.
- Interface between multi-CPU's (MC68060's, Pentiums, and Stamp II-sx's) was implemented via Ethernet and serial communication link.
- Task Description Language (TDL) was designed and implemented with VxWorks. The communication between TDL and supervisory controller is under development.

■ Structure of SAUVIM Control System

As shown in Figure RDC-1, the SAUVIM controller consists of multiple CPU boards and I/O boards to distribute tasks among multiple components. The basic idea of design is to make multiple components work in harmony and perform specific tasks based on their capabilities. The systems communicate via VME buses or Ethernet lines depending on the time dependencies of tasks. The entire controller system will be installed in two separate pressure vessels based on control objects. The first pressure vessel will contain hardware components for navigation of the vehicle. The second pressure vessel will house underwater manipulator controller and related components. These two pressure vessels are connected with Ethernet cables. Though Ethernet link is known to have a little delay in communication, we found that the communication delay is negligible in our configuration.

■ Navigation Control System Hardware

Navigation control system consists of six CPU boards and multiple I/O boards. Two of CPU boards are Force Computer SYS68K-60Ds based on Motorola MC68060 processor. The four other boards are PC104+ boards based on Intel Pentium MMX processor. Two MC68060 CPU boards are installed in a 6U VME chassis with VME bus. The PC104+ boards communicate with other CPU boards via Ethernet link. Two MC68060 boards handle main tasks for vehicle navigation and PC104+ boards support those two MC68060 boards. Two multifunctional AD/DA/DIO boards are used for interfacing vehicle's sensors and actuators. One intelligent multi-port serial communication board is installed to handle communication between CPU chassis and sensors with RS-232 or RS-485 interfaces. A small network hub will be installed in the navigation pressure vessel to handle Ethernet communications

among all CPU boards in control system. Because of the network hub, two MC68060 boards can have Ethernet connection beside the VME bus connection.

The Force SYS68K-60D board has a shared DRAM onboard. The shared memory is used for task synchronization and data exchanges between two navigation CPUs.

A frame grabber card is installed on one of PC104+ boards to capture images from external cameras around vehicle. Up to 7 cameras will be connected to the frame grabber with on-board video mux and in-house video mux board. Another two PC104+'s receive data from scan sonar of which transmits speed is up to 115.2Kbps via RS-485 interface. A RS-485 to RS-232C converter is used to convert RS-485 signal from scan sonar to RS232C in PC104+. The other PC104+ board is in charge of controlling laser range detector and serial communication link to the main MC68060 processor board

Each multifunctional analog and digital I/O board, MD-DAADIO, has 32 ADCs, 8 DACs, and 48 digital I/O channels. The ADC and the DI port A (8 bits) of the MD-DAADIO can be driven with an interrupt service routine. Compared to the previous configuration, another multifunctional I/O board has been added for future expandability. Specific pin layout is finished, but it can be changed during vehicle construction and/or test.

The intelligent serial communication board, MVC16, has 16 serial communication channels. These channels can be set up as RS-232, RS-422 or RS-485 using jumpers. In our configuration, MVC16 board has 12 RS-232 channels and 4 RS-485 channels. Since this board has own processor with 128Kbyte buffer memory, it does not require main CPU board's processor time to handle communication.

■ Manipulator Control System

Similar with navigation control system, manipulator control system has its own pressure vessel and power supply. The robot controller uses one Force SYS68K-60D processor board. One PC 104 boards is installed for homing devices. The PC104 for homing device is installed in the manipulator pressure vessel because of space limit of other pressure vessel housing navigation control system. The manipulator control system communicates with the navigation control system via Ethernet. In previous reports, we mentioned about Ethernet's inherent communication delay, but it is found negligible after simulation and experiment. The same multifunctional analog and digital I/O board is used to control brushless DC servo motors of the manipulator. Two IP quadrature counters are installed on a carrier board in the VME chassis for detecting seven resolver signals from the motors and one encoder signal from hall sensor in the gripper. A 6 degree of freedom force/torque sensor, JR3, is mounted at the wrist of the robot manipulator, and its controller is installed in the VME bus.

Figure RDC-1 shows the I/O boards and the external components for the manipulator controller. The homing sensor is interfaced with the RS-232 ports of the manipulator controller.

The manipulator control architecture is developed by the Theoretical Modeling and Low Level Control group.

■ Sensors

- **Attitude Heading Reference System: AHRS-BA303**

AHRS is a low-cost reference navigation sensor. It uses a solid state gyro system for an attitude gyro and a slaved heading gyro. It corrects errors with a closed loop system and adjusts biases from earth rotation and instrument offsets automatically. The attitude and heading signals are compared with two vertical reference pendulums and a triaxial fluxgate magnetometer to derive short-term absolute errors. To get the reliable data, moving average and min/max cancellation methods are used in low-level sensor handling routine in control software. The detailed specification of AHRS-BA303 is provided in Table RDC-1.

- **Altimeter: Tritech PA200**

SAUVIM will be equipped with seven range sonar sensors, Tritech PA200. One is for altitude (vertical) and the others are for range measuring. These sensors have RS-485 multi-drop serial communication interfaces. And, star topology is used for physical connection, because it doesn't affect the rest of the connection, and it's easy to add and remove nodes. Table RDC-2 shows the specification of PA200 sensors.

- **Electronic Compass Sensor: TCM2**

TCM2 is an electric compass sensor module. It has a three-axis magnetometer and two-axis tilt sensor. In addition to compass heading, the TCM2 supplies pitch, roll, magnetic field data and temperature information. This sensor can be used as a backup sensor for the AHRS-BA303 sensor. And, it also uses moving average and min/max cancellation methods to have noise immunity. The detailed specification of TCM2 is shown in Table RDC-3.

- **Scan sonar: Imagenex 881 high resolution imaging sonar**

The Imagenex sonar is an image scanning sonar. It will provide scanned images around the vehicle. The scanned images can be used for obstacle avoidance or target detecting. The sonar consists of two parts. One is a sonar module with a rotating sonar head. The other is a digital signal processing module, which processes sonar signal and transmits processed data via RS-485 interface. Two modules are connected with an oil-filled underwater cable. The processing module is connected to the pressure vessel of the navigation control system with a 4-conductor underwater cable. Table RDC-4 shows specification of the Imagenex 881 sonar. For the forward and backward scanning, two sonars will be installed at head and tail of vehicle. Since the communication speed for scan sonar is so fast that Pentium-based PC/104+ is used to handle communication data and image processing.

■ **Software Architecture**

There are several objective of software design for the SAUVIM. First, the whole software system is designed to be modularized so that anyone can implement his or her own control algorithm easily and additional functions can be easily added. Second, the tasks should be distributed among processor boards. The tasks should be performed in harmony with other tasks. Third, the system should provide fault-tolerant and/or fault-recovery functions to guarantee retrieval of the vehicle in case of emergency.

To fulfill those requirements, the whole software was designed to have three layers, *Application layer*, *Real-time layer*, and *Device layer*, as shown in Figure RDC-2.

The first layer of SAUVIM software, *Application Layer*, consists of application software, application task manager, and sub-task modules. Application software includes hardware independent high-level modules such as interface module for human operators, interpreter for task description language, and control algorithm for SAUVIM. Actual processing module for application software is sub-task module that includes all software modules for high-level processing. The application task manager is in charge of connecting between the application software and the sub-task modules by using task managing method such as creation and deletion of tasks, and communication and synchronization between tasks.

The second layer of SAUVIM software, *Real-Time Layer*, consists of system configurator, real-time object manger, and real-time operating system. The main role of system configurator is mapping between hardware-independent application software and real hardware-related modules. It can map real world human-friendly data to actual value for specific hardware. It can also map actual raw sensor data to human-friendly meaningful value such as depth or speed of vehicle. The real-time object manager provides system services tuned to the domain of object-oriented application. These services include management of object-oriented modules for real-time tasks. And the real-time object manager is used on top of a commercial RTOS(real-time operating system). Most of commercial RTOS vendors support and provide this real time object-related modules as an option.

The third layer of SAUVIM software, *Device Layer*, is the only hardware dependent part. Most of software modules in this layer are directly connected to hardware to send command data for actuators and get actual data from sensors. Instead of actual hardware device driver, virtual device driver can be used to emulate hardware for testing purpose or isolate software from hardware.

● **Navigation Control System Software**

The entire software is being developed based on commercial 32-bit real-time operating systems, VxWorks, and Windows 2000. Two different operating systems are selected based on the capabilities of hardware and cost. As shown in Figure RDC-3, tasks are distributed in processor boards based on hierarchical software architecture.

The primary CPU board has several main functions. First, it harmonizes requests and responds from different systems distributed in multiple pressure vessels. For example, when the manipulator control system requests the navigation control system to move the vehicle after failing to reach an object within arm range, it responds to the request and determines what to do. Second, it reports status of vehicle to the supervisor using communication link. Third, it performs high-level control like path planning and task planning. It plans tasks based on the task description language. Based on the interpretation of the task description language, it executes necessary tasks or requests other processor boards to perform necessary functions to fulfill its objective.

The second navigation CPU synchronizes tasks between two processor boards with its built-in shared memory and communicates with other boards via VME bus or Ethernet. It collects and keeps data required to operate vehicle in the shared memory. It provides the data in response to internal request or external request from other processor boards. The second CPU mainly processes device handling and data handling routines. It communicates with external devices using I/O device drivers for specific hardware. Current status of external devices is saved in the shared memory in the second CPU board for the first navigation CPU.

A PC/104+ with frame grabber handles image capturing from external cameras. It stores data from navigation CPU's and PC104+'s during mission on a hard disk drive. And, two PC104+'s gather data from scan sonars and execute image interpretation algorithm. These data can be reviewed after tasks. PC/104+ system uses Windows 2000 for its operating system.

■ Task Description Language (TDL)

TDL is designed to provide high-level task describing tool like a command script language or MatLab script language. It can handle complex tasks with pre-defined simple commands. The detail syntax of TDL is summarized in Appendix RDC-A.

TDL is implemented by PC-based interpreter/compiler developing tools, *bison & flex*, that are almost same as traditional UNIX-based tools, *lex & yacc*. It is known that *bison & flex* can generate more optimal C/C++ code than *lex & yacc*. Since communication link between PC and VME might be slow, two interpreters are implemented in PC and VME, respectively. One in PC performs syntax check, and the other in VME executes actual command. After checking syntax in PC, task description file is downloaded to VME and run.

■ SAUVIM User Interface System (UIS)

UIS consists of three parts, *Monitoring System*, *Setup mode*, and *Calibration mode*. Monitoring System displays current status of vehicle sensors/actuators. Setup mode handles power control of sensors/actuators, software interrupt control of sensors, and operation of actuators. And Calibration mode provides interactive human-machine interface to calibrate difference between actual I/O values and digitized values in I/O interfacing boards.

● Monitoring System (MS)

MS shows current status of sensors/actuators with many indicators, such as buttons, graphs, and simple graphics. The major indicators for vehicle control are mainly located in the center of screen to provide fast and easy information acquisition, and status displays are deployed off the center of screen. In normal condition, most indicators look dark. But, in case of abnormal condition, it becomes bright red to help user identify the condition. The overall display of MS is shown in Fig.RDC-4.

● Setup mode

Setup helps to handle power control of sensors/actuators and interrupt from sensors. Unimplemented, unused, or abnormal devices can be eliminated from overall vehicle control loop by disabling them in Setup mode. It makes control S/W flexible and adaptable, regardless of number of sensors/actuators

used. Once Navigation CPU gets disabled sensors/actuators information, actual data from selected devices are ignored, and redundant data from other devices will be used. The display of Setup mode is shown in Fig. RDC-5.

● Calibration mode

Calibration mode is used to calculate mapping parameters between actual analog data and processed digital data in computer. Since the mapping is almost linear, 2nd order line fitting is used for positive and negative value calibration. All analog values from A/D or to D/A have to be calculated with their own calibration data, which are calculated in Calibration mode. Fig. RDC-6 shows the display of Calibration mode.

■ SAUVIM Simulation System

Since most of sensors for SAUVIM can be operable in the water, simulation system is prepared to confirm H/W and S/W in dry condition. Simulation system has identical H/W configuration of actual SAUVIM controller except sensors. It has same CPU boards, I/O boards, and a communication board as used in SAUVIM control system. And, line/port assignment in I/O board is same as the actual system. But, each sensor signal is emulated with in-house control box, which has same output range of actual sensor. As shown in Fig.RDC-7, control boxes are used for emulating sensor signals such as depth sensor, battery voltage and current, thruster feedback current, and so on. After confirming H/W and S/W with the simulation system, overall H/W and S/W can be easily portable to actual system, without change of any part.

For visual checking, LED's are used to confirm digital I/O, as shown in Fig. RDC-8. And, to simulate battery/arm tray, stepper motor with STAMP II-sx, one chip micro-controller, is used as shown in Fig.RDC-9. Since STAMP II-sx can be programmed by serial connection, it is very convenient to modify the program in the field.

Future Tasks (Phase II Tasks)

- 1) Integration and test of low- and higher-level control routine into existing control architecture.
- 2) Upgrade and test of robust communication between navigation control system and manipulator control system.
- 3) Actual water test with current software architecture.
- 4) Refinement and debugging of software.
- 5) Program and run task with TDL.
- 6) Water test with setup/calibration mode in the field.

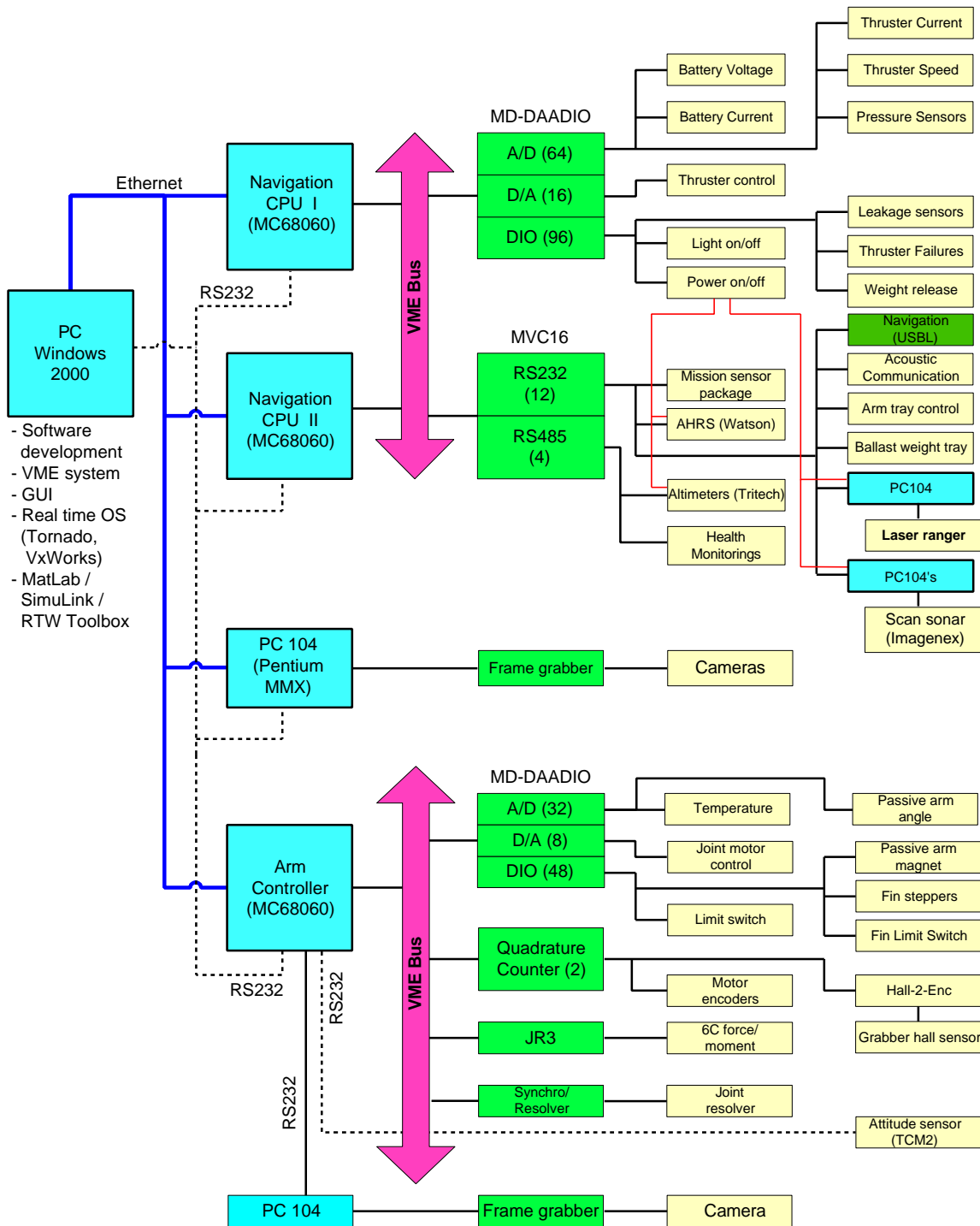


Figure RDC-1. System diagram of SAUVIM.

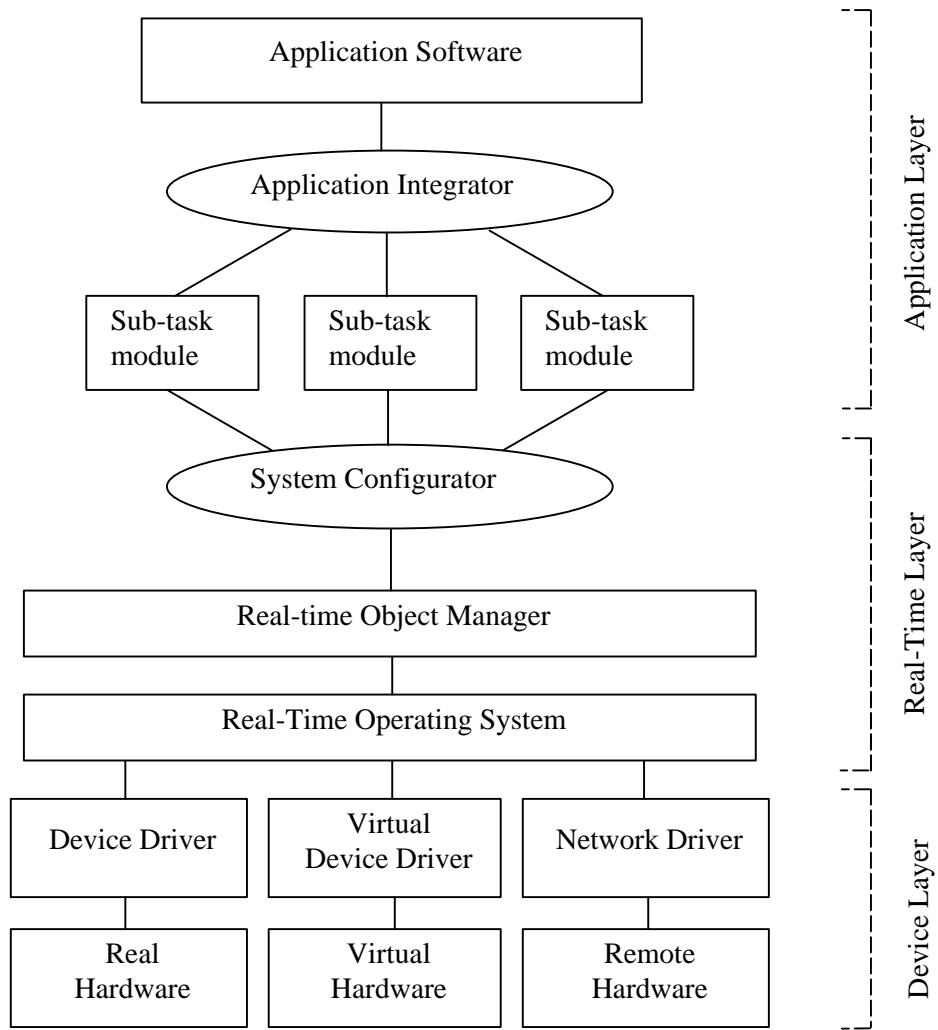


Figure RDC-2. Software hierarchy for SAUVIM controller

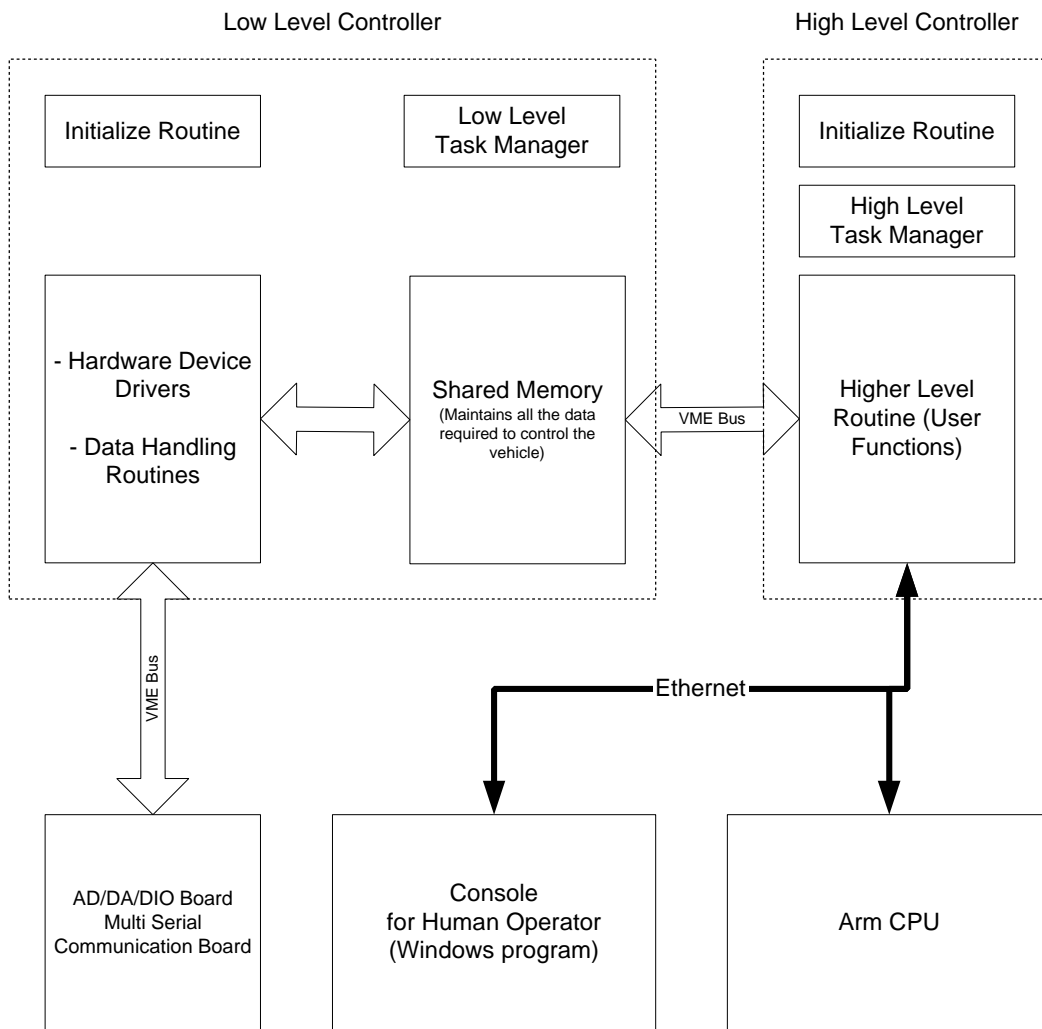


Figure RDC-3. Software architecture of the navigation control system.

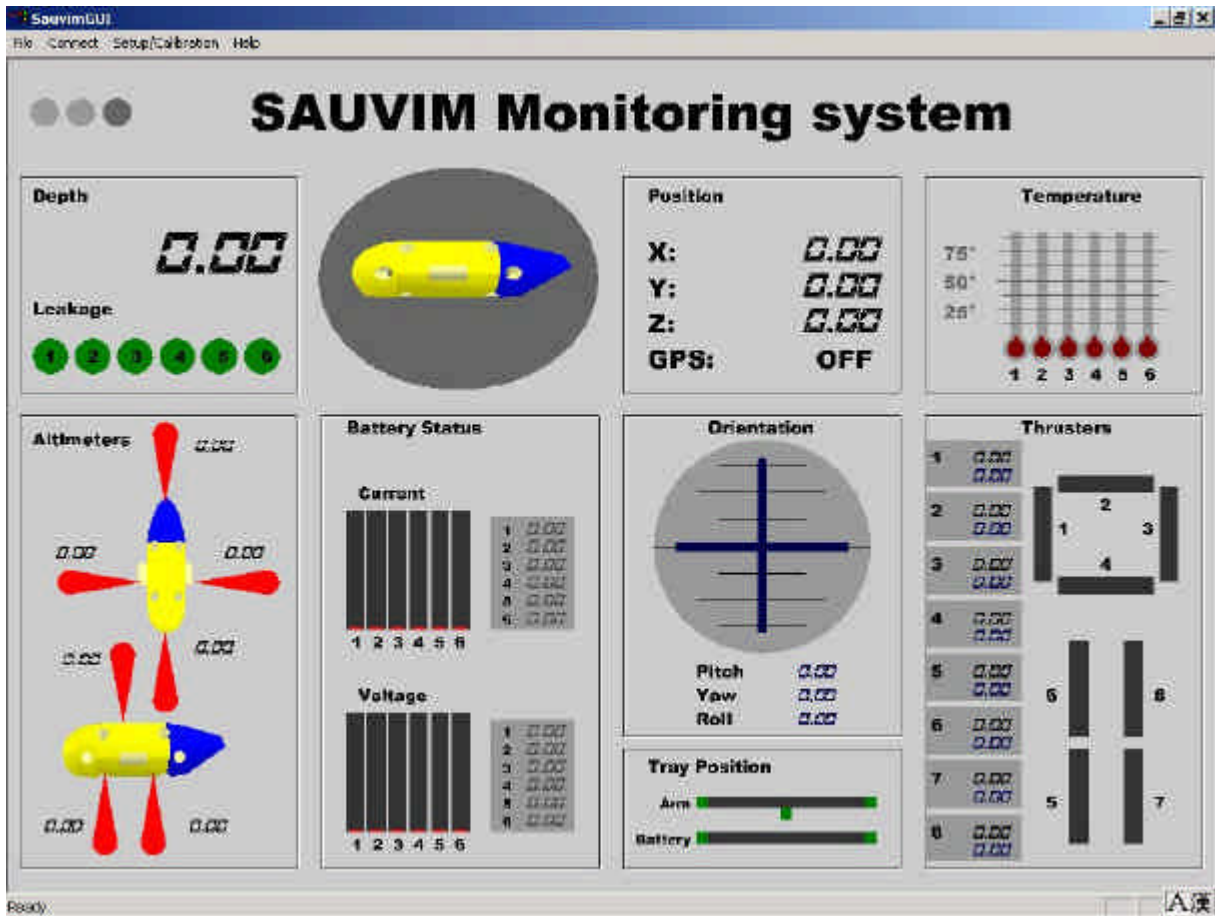


Figure RDC-4. Display of Monitoring System

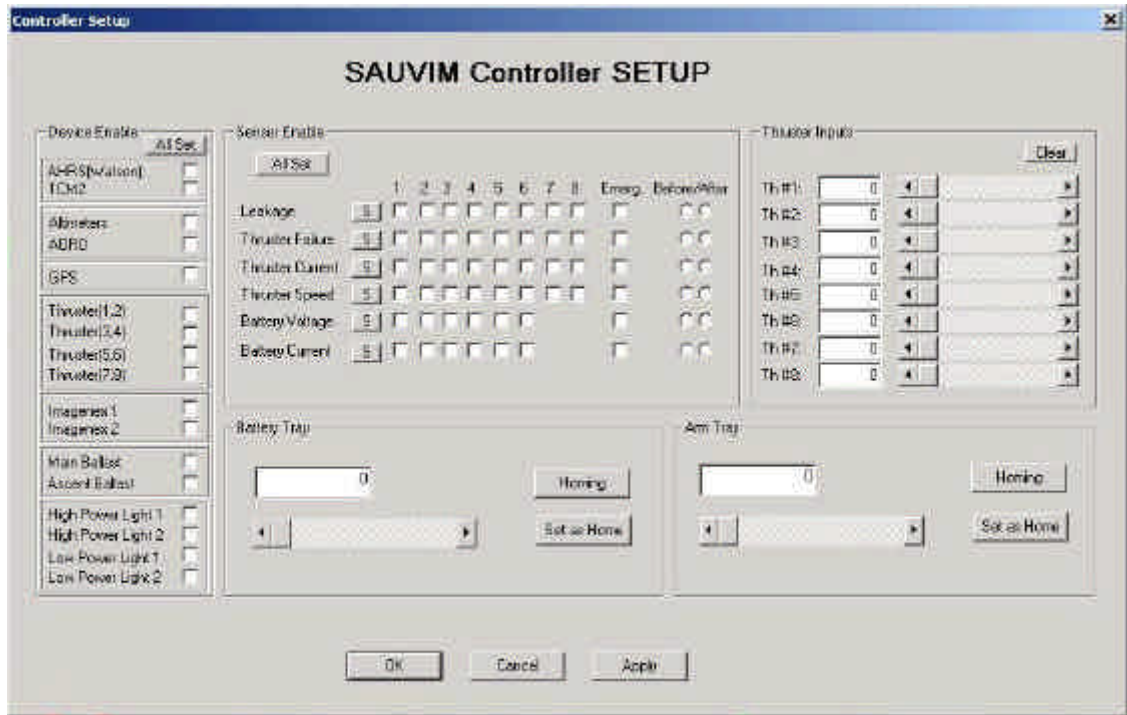


Figure RDC-5. Display of Setup mode

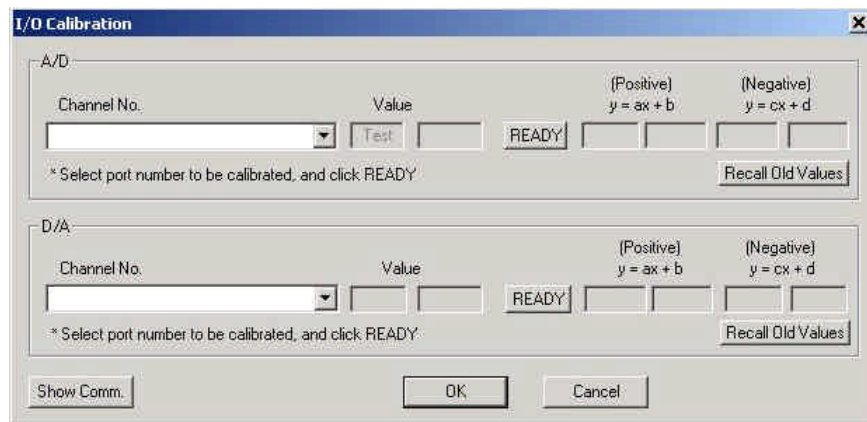


Figure RDC-6. Display of I/O Calibration mode

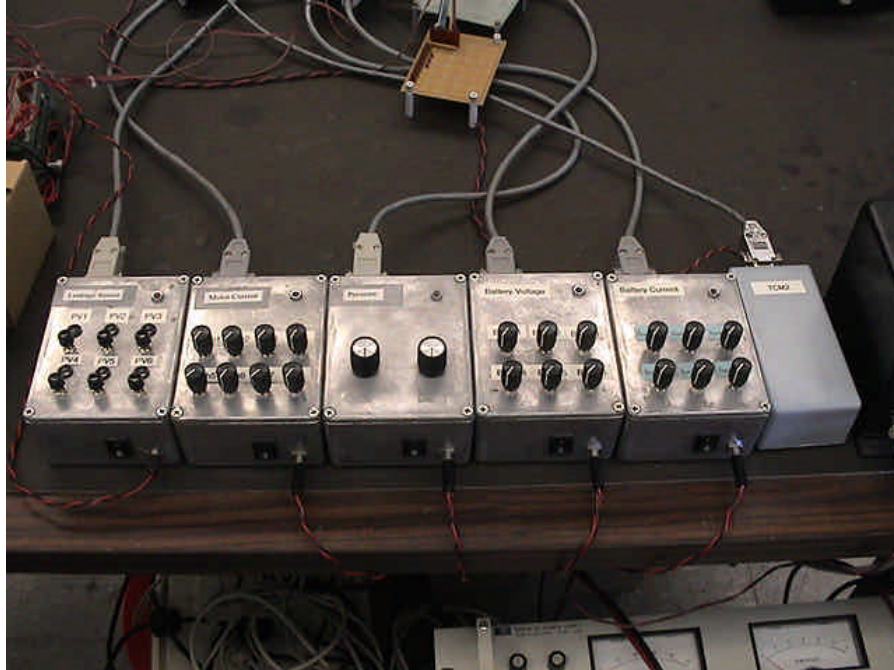


Figure RDC-7. Control boxes for simulation system

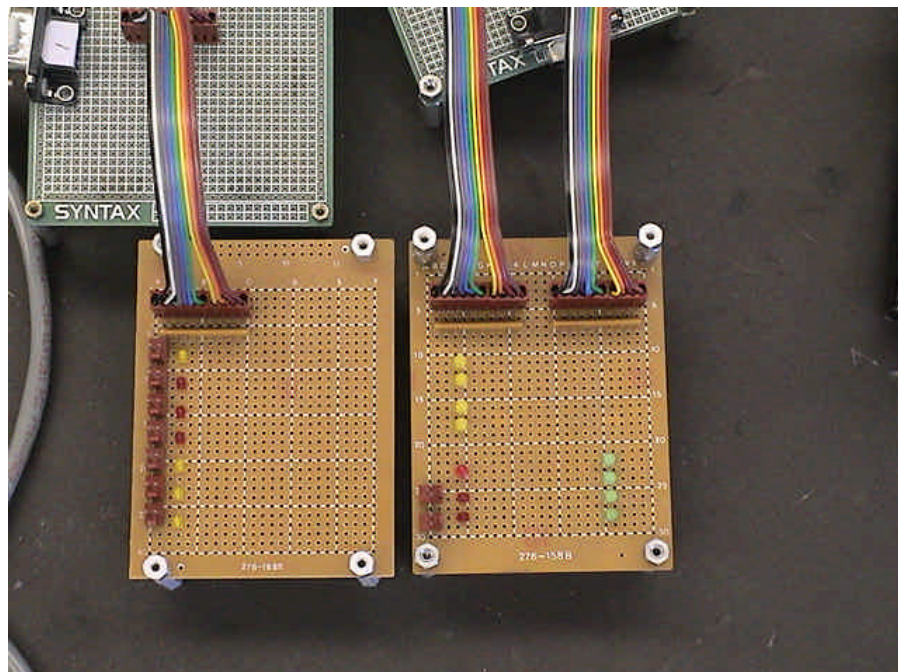


Figure RDC-8. LED boards for Digital I/O

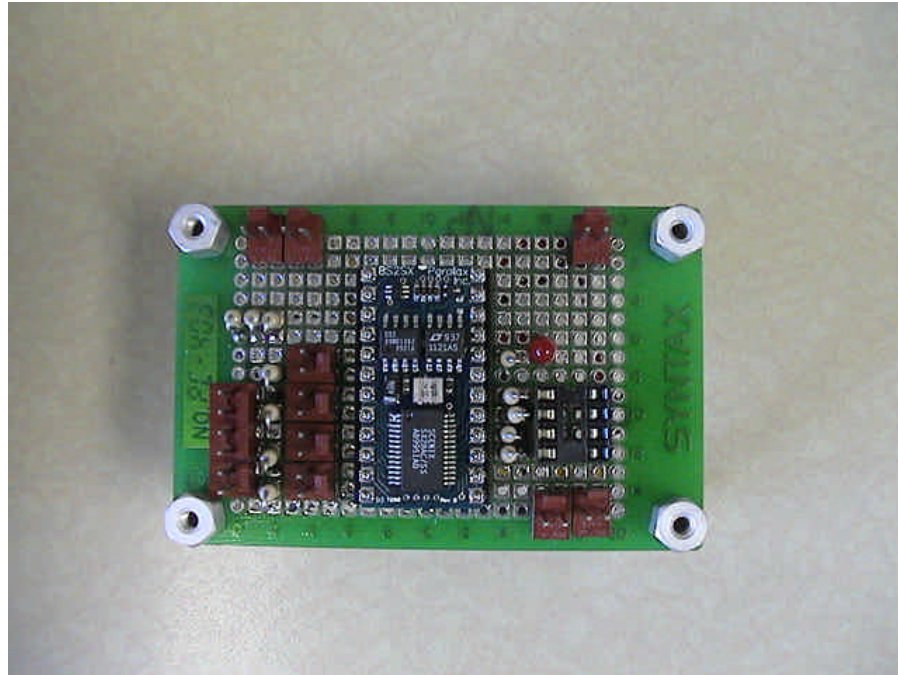


Figure RDC-9. STAMP II-sx for stepper motor controller

Table RDC-1. Specification of AHRS-BA303

Item	Range	Accuracy	Sensitivity	Remarks
Pitch rate	$\pm 100^\circ/\text{s}$	Static: $\pm 0.2^\circ/\text{s}$ Dynamic: $\pm 2\%$ digital $\pm 6\%$ analog	10 $^\circ/\text{s}/\text{V}$	Positive for nose up
Roll rate	$\pm 100^\circ/\text{s}$			Positive for roll to right
Yaw rate	$\pm 100^\circ/\text{s}$			Positive for right turn
Heading rate	$\pm 100^\circ/\text{s}$			Positive for right turn
Bank	$\pm 180^\circ$	Static: $\pm 0.5^\circ/\text{s}$ Dynamic: $\pm 2\%$	18 $^\circ/\text{V}$	Positive for bank to right
Elevation	$\pm 90^\circ$			Positive for nose up
South heading	0 - 360 $^\circ$	Static: $\pm 1^\circ/\text{s}$ Dynamic: $\pm 2\%$	18 $^\circ/\text{V}$	S=0V, E=-5V, W=5V, N= $\pm 10\text{V}$
North heading	0 - 360 $^\circ$			N=0V, E=5V, W=-5V, S= $\pm 10\text{V}$
Velocity input	-400 – 400 Km/hr		40 Km/hr/V	Forward velocity
Error correction time	15 seconds			

Table RDC-2. Specification of Tritech PA200

Frequency and beam width	200 kHz and 20 degrees
Measurement range	100 meters
Operating depth	6800 meters
Input voltage	12 VDC
Interface	RS-485, 9600 bps, 8 data bits, 1 stop bit, no parity
Head RS-485 Termination	220 Ω (Sensor A only)
Command	*, or 'A', 'B', 'C', 'D', 'E', 'F', 'G'

Table RDC-3. Specification of Precision Navigation TCM2

Heading information	Accuracy when level	$\pm 0.5^\circ$ RMS
	Accuracy when tilted	$\pm 1^\circ$ RMS
	Resolution	0.1°
	Repeatability	$\pm 0.1^\circ$
Tilt information	Accuracy	$\pm 0.2^\circ$
	Resolution	0.1°
	Repeatability	$\pm 0.2^\circ$
	Range	$\pm 20^\circ$
Magnetic field information	Accuracy	$\pm 0.2 \mu\text{T}$
	Resolution	$0.01 \mu\text{T}$
	Repeatability	$\pm 0.2 \mu\text{T}$
	Range	$\pm 80 \mu\text{T}$
Temperature information (sensor is uncalibrated)	Accuracy after calibration	$\pm 1^\circ\text{C}, \pm 2^\circ\text{F}$
	Resolution	$1^\circ\text{C}, 2^\circ\text{F}$
	Range	-20°C to 70°C
Power requirement	Supply voltage	+5 VDC regulated 6 to 18 VDC unregulated
	Current	Standard mode: 15-20 mA Low-power mode: 7-13 mA Sleep mode: 2.5 mA
Interface	Digital	RS-232C, NMEA0183
	Analog	0-5V linear, 19.53 mV resolution (256 discrete levels), 0-5 quadrature (sine and cosine)

Table RDC-4. Specification of Imagenex 881

Frequency	675 kHz
Transducer	Imaging/profiling
Power supply	22 – 48 VDC at 1 Amp max.
Interface	RS-485 (115200 bps, 8 data bits, 1 stop bit, no parity)
Operating range	6000meters
Measurement range	5 – 200 meters 15 – 600 feet Default: 50m (150ft)
Sector size	Scan with angle Sector mode: 0 to 180° in 3° increments. Default: 180° Polar mode: 0 to 360° in 3° increments Default: 360°
Speed	Step size angle Slow: 0.3°/step Med: 0.6°/step Fast: 0.9°/step Faster: 1.2°/step Fastest: 2.4°/step Default: fast
Transmit pulse length	0 to 255 μs in 5 μs increments

APPENDIX RDC-A. Syntax of SAUVIM Task Description Language

1. Vehicle Motion

- 6 DOF motion
- Use relative/absolute Pose
- Position/ Orientation/ Pose
- Speed (SP)

◆ Relative motion (w.r.t. body-fixed frame)

movev.r *position_variable* [with sp=*speed*]

movev.r *pos1, pos2, pos3* [, *ori4, ori5, ori6*] [with sp=*speed*]

◆ Absolute motion (w.r.t. earth-fixed frame)

movev.a *position_variable* [with sp=*speed*]

movev.a *pos1, pos2, pos3* [, *ori4, ori5, ori6*] [with sp=*speed*]

pos = float, integer, * (current position ; valid for absolute motion)

ori = float, integer, * (current orientation ; valid for absolute motion)

speed = float, integer [absolute value]

(float, integer) + % [relative value]

◆ Fin motion

fin *#number, angle* [with sp = *speed*]

number = integer (1 ~ 3)

angle = integer (-90 ~ 90)

speed = float, integer (default = pre-defined speed)

2. Robot Motion

- 7 DOF
- Relative/Absolute Pose in 6 DOF (in Cartesian Space)
- Relative/Absolute Pose in 7 DOF (in Joint Space)
- Position/ Orientation/ Pose
- Speed (SP)
- By-pass level (PL)

◆ Relative motion (in Cartesian space)

mover.r *position_variable* [with sp=*speed*]

mover.r *pos1, pos2, pos3* [, *ori4, ori5, ori6*] [with sp=*speed*]

◆ Absolute motion (in Cartesian space)

mover.a *position_variable* [with sp=*speed*]

mover.a *pos1, pos2, pos3* [, *ori4, ori5, ori6*] [with sp=*speed*]

◆ Relative motion (in Joint space)

movej.r *joint variable* [with sp=*speed* [%]]
movej.r *ang1, ang2, ang3, ang4, ang5, ang6, ang7* [with sp=*speed* [%]]

- ◆ Absolute motion (in Joint space)

movej.a *joint variable* [with sp=*speed* [%]]
movej.a *ang1, ang2, ang3, ang4, ang5, ang6, ang7* [with sp=*speed* [%]]

pos = float, integer, * (current position ; valid for absolute motion)
ori = float, integer, * (current orientation ; valid for absolute motion)
ang = float, integer, * (current orientation ; valid for absolute motion)
speed = float, integer [absolute value]
(float, integer) + % [relative value]

- ◆ Force control command : T.B.D.

- ◆ Gripper command : T.B.D.

3. Sensing

- Analog
 - Input
 - PV temperature
 - Motor temperature
 - Motor current
 - Depth/Pressure
 - Output
 - Thruster command

ain #*port number*
aout #*port number, floating value*

- Digital
 - input
 - Limit switch
 - Leakage sensor
 - Output
 - Weight release
 - light

dio #*port number*
dout #*port number, binary value*

- Communication-based sensors
 - INS
 - Altimeters

➔ Also, it is possible to use *internal state variable* to read/write the data

4. Arithmetic Operations

- Basic Operations : +, -, /, *, %, DIV
- Trigonometrical Functions : sin, cos, tan, asin, acos, atan, hsin, hcos, htan
- Etc : sqrt, log, exp
- Use () to change operation priority

5. Logical Operations

- Byte-wise operations: and (&&), or (||), not (!)
- Bit-wise operations : and (&), or (|), not (~), ex-or (^)
- Comparison : <, >, <=, >=, ==, !=

6. Flow Operations

- IF (*logical expression*)
statements
ELSE
statements
ENDIF
- DO
statements
WHILE (*logical expression*)
- WHILE (*logical expression*)
statements
ENDWHILE
- FOR (initialization; condition ;)
statements
ENDFOR
- SWITCH (values)
CASE values : statements
BREAK
DEFAULT : *statements*
BREAK
ENDSWITCH

7. File Operations

- Open/Close
 - o open.file number file type filename
 - o close.file number
 - o close.* → close all open files
- ex) open.1w data.dat → open *data.dat* for writing with file ID #1
- close.2 → close file ID #2
- close.* → close all files

- Read/Write data
 - **write.file number variables/IO data [with period = time]**
 - **read.file number variables**
- File management
 - **type filename**
 - **dir [filename]**
 - **del filename**
 - **copy filename1 filename2**

8. ETC

- Define/Alias
 - **define name value**
- Change status of internal flags
 - **set flags ON/OFF**
 - **get flags**

flags

 - **stop_e** : stop at error (default)
 - **stop_w** : stop at warning
 - **verbose_e** : display error message (default)
 - **verbose_w** : display warning message
 - **coord_a** : absolute coordinate
 - **coord_r** : relative coordinate
 - **debug** : display debug message (default : off)
- Set/get warning or error range of internal state variables
 - **set internal state variable warning/error at (lower limit: upper limit)**
 - **get internal state variable**

internal_state_variable : pre-defined variable (see Variable)
lower limit, upper limit : integer, float
- *//* : remark
- **printf()** : print out. Usage is same as C.
- **goto [number]**
- Variable
 - Use the first character in the variable as an identifier of the variable type.
 - **I***** : integer variable
 - **F***** : floating variable
 - **L***** : logic variable
 - **J***** : joint variable

- C*** : Cartesian variable

Ex) I_var = 34
 F_value = 34.56
 L_value = 1
 J_angle = (23.4, 32.34, 0.32, 78.3, 28.3, 39.5, 34.2)
 C_pos = (10.2, 32.32, 45.3, 10.2, 1.23, 0.34)

- Use array with []
- Pre-defined internal state variables

ain_pv temp = ain#1
ain_mot temp = ain#2
ain_mot cur = ain#3
ain_depth = ain#4
ain_battery = ain#5

about 1 ~ 8 = about#1 ~ 8
→ thruster 1~8

din_fin1 limit h = din#1
din_fin1 limit n = din#2
din_fin1 limit l = din#3

for INS

ins bank, ins elev[ation], ins head[ing], ins xv, ins yv, ins zv

for Altimeter

alti[eter] 1 ~ 4

- Pre-defined internal variables

speed.v = value [%]

speed.r = value [%]

- Wait or delay process
 - **delay time**
time : msec
- Save/Load program file
 - **save [filename]**
 - **load filename**
- Start interpreter
 - **run [filename]**

Mission Package Sensors (MSP)

Project Leader(s): Dr. Tae Won Kim, Dr. Song K. Choi & Mr. Oliver T. Easterday
Personnel: - none -
Past Project Leader(s): Dr. Gary McMurtry
Past Personnel: Mr. Yann Douyere, Mr. Alan Parsa & Mr. Max D. Cremer

Objectives

The SAUVIM Mission Sensor Package for Phase 1 is designed to provide semi-continuous records of AUV water depth (pressure), water temperature, conductivity, computed salinity, dissolved oxygen, pH and turbidity for at least eight hours. These parameters as well as the magnetic signature of the seafloor can be acquired by the SAUVIM in survey mode. In intervention mode, the Mission Sensor Package will provide AUV water depth (pressure) and the water temperature and compositional parameters at a selected seafloor target, including pumped samples from submarine seeps or vents.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

The system-wiring layout is presented in Figure MSP-1. Ambient seawater or submarine vent/seep waters enters the Teflon sensor plenum through a short length of Teflon tubing, which contains a marine anti-foulant. The Teflon entry nozzle screens and faces the forward direction of the AUV, allowing waters to passively enter the sensor housing when the AUV is running or to be pumped across the sensors when the AUV is station keeping. Otherwise, the Sea Bird Electronics impeller pump remains off to conserve power. The Ocean Sensors model OS 200 CTD is a small, compact and low power instrument used to acquire data in remote oceanic; the sensor head is 2.25 inches in diameter (OD) and will house the conductivity cell, thermistor temperature probe, pressure, pH and dissolved oxygen sensors. The ranges for the sensors are: pressure, 0 - 6000 dB; temperature, -2 - 100° C; conductivity, 0.5 - 65 mS/cm; salinity (computed), 2 - 42 PSU; pH, 0 - 14 pH units; dissolved oxygen, 0 - 15 ml/l. All sensors are rated to 6000 meters.

The Ocean Sensors OS-200 CTD electronics have been custom modified to slave to our CPU (via RS-232 link) and allow up to eight additional analog inputs. Three of these analog inputs are currently used to retrieve pH, turbidity and dissolved oxygen data. Additional space within the command or CPU housing will be set aside for future sensor additions. The Sea Tech Light Back-Scattering Sensor (LBSS) will be measuring by a high-gain (to 33mg/l) the particle concentrations or turbidity levels in the surrounding. The LBSS or nephelometer will be externally mounted to the AUV so that its frame will not obstruct the light emitting diodes during its time of operation.

The PC/104 CPU module, manufactured by AMPRO computer Inc, provides for a 25 MHz 386SX processing power in a compact, pre-configured subsystem module. The CPU card is interfaced with a serial communication card (Emerald-MM PC/104 Module from Diamond Systems Corporation) that offers two independent RS-232 serial ports, which are used to converse directly with the OS-200 CTD and the magnetometer. The computer operates in a DOS-type environment. The DOS version was updated from v5.x to v6.22 in order to provide the POWER.EXE features that allow battery power conservation. Several C-based programming files were created to communicate with the sensors. Information extracted from the sensors are directly saved in the 64-MB flash card from M-Systems

The magnetic signature of the seafloor will be measured with an Applied Physics Systems model 544 Micro Angular Orientation Sensor. The unit contains both a 3-axis fluxgate magnetometer and a 3-axis accelerometer. These sensors are sampled by an internal ADC and microprocessor subsystem, which outputs 16-bit digital data representing the magnetometer and accelerometer readings via an RS-232 cable to the CPU. Ideally, to minimize the AUV magnetic background, this small (0.75" x 0.75" x 4.6") sensor should be placed as far away from magnetic-field generating devices (e.g., motors, spinning propellers, circuit boards, hard disks) as practicable. To date, we plan placement of this sensor in the nose faring of the AUV. The angular orientation sensor is housed within a 1-inch OD cylindrical pressure vessel (6000-m capable) made low-Fe grade-2 titanium.

For a 6000-m depth capability, we have constructed a 7071-grade titanium pressure housing for the external power supply, Intel 386-based CPU and associated electronics, and the internal battery. Currently, we use a PC/104 card stack for the CPU and serial I/O, and a 128 MB Quantum IDE hard drive for program and data storage. System power will be provided at 12 VDC within the main body of the AUV. Communications (system command uploads and downloads data) to the surface and other AUV CPUs will be via RS-232 link. We plan to mount the CTD and CPU housings of the mission sensor package within the AUV forward of the battery pods and high enough above the skids to minimize damage on the seafloor. The magnetometer will reside within the faring of the AUV body as far forward as possible to isolate it from the motors. The nephelometer will point down immediately below the CTD and CPU housings. When station keeping, the AUV manipulator will be able to pull the nozzle out toward any vent or seep for better sampling.

SOFTWARE DESCRIPTION

Programs:

There are a total of 8 files (*.cpp) that operate the mission sensor package.

They are:

1. MAIN.CPP
2. SAUVIM.CPP
3. MENU.CPP
4. DIGITAL.CPP
5. MAGNETO.CPP
6. ANALOG.CPP
7. CTD.CPP
8. MAG.CPP

Figure MSP-2 provides for the general layout of the entire program and presents the interactions between the different files.

Starting and Ending the Program:

The path to the main executable file (SAUVIM.EXE in the root directory) has been inserted in the AUTOEXEC.BAT file. This allows the program to start automatically after the boot procedure of the computer. Similarly, it can be stopped at anytime when the user issues the CTRL-BREAK command followed by the carriage return key. Once the program starts it automatically program the magnetometer by issuing the following command:

```
0L \r
0SD \r
```

The first command allows to enable low level writes while the second tells the magnetometer to send the data. The data will return under the following format:

```
-----  
MX:+0.16124 AX:+0.82113  
MY:+0.22124 AY:-0.56370  
MZ:-0.14426 AZ:+0.00341  
T: +16.842  
-----
```

Where MX, MY, MZ correspond to the 3 axis fluxgate magnetometer values and AX, AY, AZ to that of the 3 axis accelerometer. The values are given with respect to a Cartesian referential frame. In addition, the temperature is given in degree Celsius.

Once the value of the magnetometer is successfully extracted. The program closes the serial communication port with the magnetometer and sequentially opens a new serial port with the CTD. Just like the magnetometer, the program sends a set of command to the CTD in order to retrieve any information of interest. The following program is issued to the CTD:

```
-----  
P: \r  
10 PRINT LPRT TEMP PRESS COND AS128 AS129 AS132 \r  
20 WAIT 00:00:01 \r  
30 IF LPRT <60 THEN GOTO 10 \r  
40 END \r  
E \r  
R: \r  
-----
```

The first line in the series programs the CTD to start a new program in Loop: (i.e. P B). The next line tells the CTD to print to screen the temperature, pressure, conductivity, Dissolved Oxygen (AS128), pH (AS129) and turbidity (AS 132) [line 10]. There is a waiting time of 1 second between each sampling [line 20]. The statement on line 30 causes the loop to continue until 59 samples have been retrieved and saved onto the hard drive [line30]. Line 40 tells the program to end upon retrieval of the 59 samples. Finally the last two lines tell the CTD to exit from the programming mode, and start running loop B.

Following the CTD data retrieval, the program closes communication port with the CTD and reopen a port with the magnetometer to extract once more the magnetic, acceleration and temperature data. Once the data from the magnetometer are extracted and the communication port closed with all the devices, the program goes to sleep for 10 minutes.

The retrieved data are automatically saved in the root directory of the system's HD onto the files CTDATA.dat and MAGNETO.dat, which respectively correspond to the data gathered from the CTD and the magnetometer.

Since the SAUVIM computer was designed to interact at will with the MSP computer, it is possible to access any of the electronic equipment at any given time in order to retrieve CTD or magnetometer data. To do so the SAUVIM computer has to issue commands 1 to 3.

- Issuing 1 will program the CTD to retrieve one single value of temperature, pressure and conductivity. These values are
- Issuing 2 will program the CTD to retrieve one single value of Dissolved Oxygen, pH and turbidity.
- Issuing 3 will program the Magnetometer to get the 3-axis fluxgates, the 3-axis accelerometers, and the temperature value.
- Issuing a ^z will allow the program to go back to its normal mode of acquisition with two minutes of sampling with the CTD and Magnetometer, and 10 minutes in sleeping mode.

Note: the data retrieved from the SAUVIM computer will NOT be saved in the MSP computer.

As opposed to the temperature, pressure, and conductivity, which values are automatically converted in engineering forms within the CTD, dissolved oxygen, turbidity and pH values are given in raw forms. Therefore, it was necessary to convert the raw values as they were gathered. ANALOG.CPP allows the conversion to takes place so that anyone can get a feeling of the values as they are collected in real time.

The governing equation to change the raw data to engineering ones are given below for each of the channels:

```
f2=(i2*GAIN128)+OFF128;      //DO2
f3=(i3*GAIN129)+OFF129;      //PH
f4=(i4*GAIN132)+OFF132;      //TURBIDITY
```

Where the offset and gains are given and are part of the calibration values for each of the sensors.

The dissolved oxygen, pH and turbidity data that are saved in the file CTDATA.dat onto the MSP hard disk are in raw form. These data can be converted into their respective engineering format using the application AUTO16SB.EXE application located in the MSP hard drive. When the program AUTO16SB.EXE is started, one must be sure that the STATUS.CAL file must be present in the same directory. The latter contains the necessary offset and gain values used for the conversion. Please refer to Figure MSP-3 to see the organization of the program

Calibration file

The file STATUS.CAL contains the offset and gains used by the CTD to convert raw data to engineering values. As mentioned earlier, the CTD was not programmed by the vendor to provide engineering values from the analog channel output (i.e. DO2, pH and turbidity). The basic equation for the conversion is given in the previous section. This file is very important, as it allows to getting accurate data within the instruments resolution and accuracy. The calibration file for our CTD looks like the following:

```
-----
OS SERIAL #: 430
ROM VERSION: 2.0, FRI FEB 26 12:29:46 1988

CH NOFF NGAIN 128 10 25 -1.010202E+01 5.939151E-04 mg/L
0 0 200 129 128 255 1.619271E+01 -4.440469E-04 pH
1 2 14 130 128 255 7.501447E+00 -3.648914E-04 V
2 128 255 131 128 255 7.507731E+00 -3.649589E-04 V
3 1 150 132 128 255 7.500877E+00 -3.649074E-04 V
```



```

4   255   255 133 128 255  7.518998E+00 -3.649128E-04 V
5   128   255 134 128 255  7.484714E+00 -3.649360E-04 V
6    2     5 135 128 255  7.487344E+00 -3.649119E-04 V

COND   -1.096916E-01  8.105237E+01  0.000000E+00  0.000000E+00
TEMP   -1.165107E+01 -2.094517E+01  1.270573E+00 -1.534750E-01
PRESS   5.000000E+02  1.025981E+05  0.000000E+00  0.000000E+00
( . . . . )

```

Since no sensors behave exactly the same way, the offset and gain values should be changed accordingly in case of a sensor replacement. Please contact the vendor at Ocean Sensors (section 6 in this report).

SYSTEM DESCRIPTION

Orientation Sensor:

The Model 544 system contains both a 3 axis fluxgate magnetometer and a 3 axis accelerometer. These sensors are sampled by an internal A to D converter and microprocessor subsystem which outputs 16 bit digital data representing the magnetometer and accelerometer readings. The system can also be configured to transmit the roll, pitch and azimuth orientation angles of the Model 544 system. These angles are calculated before transmission from the accelerometer and magnetometer sensor output data.

Communication with the 544 system is accomplished by means of a bi-directional serial data link which can be configured to be TTL compatible or RS232 compatible. The system baud rate is user programmable up to a maximum of 9600 baud. Commands to the 544 and data from the 544 are both in the form of ASCII characters. A high speed binary communications protocol is also available, and can be enabled by the user.

The Model 544 scale factors, zero bias factors and alignment angles are measured by placing the system in precision rotational and magnetic field applying fixtures. Scale and offset calibration factors are typically measured over the 0 to 70°C temperature range. The integral microprocessor corrects for alignment, scale and offset factors at any given temperature before outputting data. The system calibration data is stored in the system EEROM and is directly accessible to the user.

The magnetometer noise level is 5×10^{-6} Gauss and the accelerometer noise level is 2×10^{-4} Gee. The maximum data throughput is approximately 3 readings/sec if all 6 outputs are transmitted. When viewed as a roll, pitch and yaw sensor, the temperature compensated 544 system has an overall accuracy of $\pm 0.5^\circ$ for roll and pitch and $\pm 1.5^\circ$ for azimuth.

Light Back-Scattering Sensor:

The light Back-Scattering Sensor projects light into the sample volume using two modulated 880nm Light Emitting Diodes. Light back-scattered from the suspended particles in the sample volume is measured with a solar-blind silicon detector. A light stop between the light source and light detector prevents the measurement of direct transmitted light so that only back-scattered light from suspended particles in water are measured.

The sensor has two ranges permitting the user to measure nearly all suspended particle concentrations found in open ocean or coastal water. Range of the measurement of suspended particle concentration in water will be approximately 30 mg/l in High-Gain is selected. If Low-Gain is selected full scale will be a factor of 3.3. high of approximately 100 mg/l.

Do not remove the protective cap during installation and wiring of the light back-scattering sensor. The protective cap will prevent accidental damage to the sensor surface and is used to test the device during the installation and wiring process. The cap is design to allow reflected light to be measured permitting sensor operation to be verified with the protective cap in place.

Sensor Wiring

The sensor is supplied with five-conductors bulkhead connector. The LBSS should be operated with a power supply capable of delivering 5 mA of current a nominally +12VDC. The sensor voltage output is 0 to +5VDC with an output impedance of 1000 ohms. The sensor has two gain ranges, high and low. For the user who chooses to select gain manually, high gain is selected by leaving pin 3 open. Low gain is selected by connecting pin 3 to power ground. Gain can be remotely controlled by the user equipment. Pin 3 is tied to +V, through a 100K ohm pull-up resistor permitting a logic high to select high gain and logic low to select low gain. Gain can also be controlled manually if a special cable is fabricated.

Sensor Care

The LBSS housing material is ABS plastic filled with black epoxy and the optical windows are clear epoxy. The LBSS sensor may be cleaned with soap and water or alcohol. Use a on-abrasive paper or cloth to clean the sensor front surface to avoid scratching the clear epoxy windows. Clean the optical windows before and after use in water.

Sensor installation

The LBSS is designed to measure light scattering in the forward hemisphere relative to the front surface to the sensor. The sensor should be installed at the outer most diameter of the users system. This should be done such that light scattering objects will be behind the plane defined by the front surface of the LBSS. The LBSS should also be placed as far away from light reflective objects as possible. If significant zero offset is observed in the LBSS data for very clean water then the LBSS is probably not mounted correctly.

Submersible Pump:

The SBE 5T pump module is a compact unit consisting of a centrifugal pump head and a long-life, brushless, DC, ball bearing motor contained in a titanium pressure housing useable to 10,500 meters (34,400 ft). The pump impeller and electric drive motor are coupled magnetically through the housing, providing high reliability by eliminating moving seals. The SBE 5T is a primary component in our SBE 9plus CTD Underwater Unit and SBE 25 SEALOGGER CTD. It is also used as optional equipment on the SEACAT (SBE 16plus and 19plus) product line. The pump flushes water through the conductivity cell at a constant rate, independent of the CTD's motion, improving dynamic performance. The pump may also be suitable for custom applications, where pressure heads are less than 300 cm of water and flow rates are less than 100 ml/sec. The SBE 5T is configured for various applications by selecting either standard or low voltage options, and one of several motor speed options. Speed options of 1300, 2000, 3000, or 4500 rpm have been established to meet flow rate

requirements for Sea-Bird applications. Other speeds can be set by adjusting a potentiometer. Motor speed and pumping rate remain nearly constant over the entire input voltage range (less than 1% change in speed for a one volt change in supply voltage). The unrestricted flow rate with no head is approximately 100 ml/second at 2000 rpm. Flow changes are nearly linear with changes in speed. With unlimited power supply current, turn-on surge is approximately 1.8 A (maximum), which drops to steady state in about 0.25 seconds. If power supply current is limited to approximately 200 mA, the motor will come up to speed in about 0.30 seconds. A series diode is installed in the input power line to prevent damage if the wires are accidentally reversed.

OS200 CTD Sensor:

The OS200 CTD is small, lightweight, low-power instrument used to acquire conductivity/temperature/depth data in remote oceanic, estuarine and fresh water environments

Sensor Care and Maintenance:

The conductivity cell must be properly wetted before using. This can be accomplished by soaking the CTD sensors overnight in a bucket of water or flushing by lowering the CTD to 10 to 20 meters depth a couple of times. Failure to wet the cell will result in improper output of the conductivity sensor. A non-ionic wetting solution will greatly improve wetting. Wetting solution are readily available through chemical supply stores. Each time the CTD is removed from the water, rinse the sensor site with fresh water and replace the red plastic sensor guard cap. If temperatures will be above freezing, the wet cap can be filled with seawater and placed over the sensor guard and cap. Allow some room in the wet cap for thermal expansion. Do not expose the sensors to temperatures below freezing when the instrument and sensors are wet. The freezing of water in the conductivity sensor will damage the cell. If temperature below freezing is encountered, remove any water from the conductivity cell. This is easily accomplished by gently blowing into the cell volume. Do not stick any object into the pressure transducer orifice as the thin (less than 1 mm) membrane for this sensor can be easily damaged. Avoid large shocks or vibrations to the instrument.

The conductivity cell is very fragile and can be broken by a curious prodding finger. Do not attempt to remove the ceramic barrel from the conductivity cell. It is there to limit the size of the conductivity and has been glassed onto the sensor. Remove batteries for long term (more than 90 days) storage.

O-ring Inspection Procedure:

O-ring inspections as a regular part of the care and maintenance of most oceanographic instrumentation, and the OS200 is no exception. Even small items such as specks of dirt or a strand of hair, if link across and end cap O-ring, can cause the instrument to flood and render it unusable. The OS200 has 7 static O-ring seals contained within the instrument, and each one is necessary to maintain the watertight integrity of the unit. Therefore, the following inspections should be performed prior to re-making any O-ring seal that has been broken. Except for the CT sensor, all O-rings should be removed from their groove when performing an inspection. The back of the CT is too fragile to remove the O-ring unless absolutely necessary, and therefore an in-place inspection should be made. Visually and by touch, inspect the O-ring for nicks, cuts and abrasions to its surface. Also look for any permanent deformation like flattening or pinching. Stretch the O-ring to look for material cracking or fine cuts that would not otherwise be visible. Inspect both the O-ring groove and mating surface, looking for pitting, scratches or foreign material. Clean the surfaces with a lint free cloth and Q-Tip, if possible. Clean and re-lubricate O-rings using generous amounts of silicone (grease, oil or spray), or any other O-ring lubricant specifically designed for use with Buna-N (Butyl Nitrile) O-Rings in static, long-term saltwater applications. Carefully replace the O-ring in its groove. Once it is

completely in its groove, give it a slight twist (or run your finger pad all the way around, pressing into the O-rings) to evenly distribute both the O-ring tension and the lubricant.

Mechanical Design

a. Pressure housing

The pressure housing is designed for operation to 6000 meters and composed of three major components. The largest portion is referred to as the pressure case. The pressure case material is 6061-T6 aluminum. Each end of the pressure case is sealed with an end cap. One end cap is referred to as the sensor end cap the other as the fitting end cap. Inserted into the sensor end cap are the conductivity, temperature and pressure sensors. The lifting end cap has a lifting strap. The lifting end cap also has the underwater connector to establish communication in the RS-232 protocols.

b. End Cap O-ring Seals

Around the circumference of each end cap are two O-ring seals (Parker Number 2-030 material: Buna-N). Each time the end cap is removed the O-rings should be inspected for damage (small cuts or abrasions), proper lubrication and debris. If the O-ring is not lubricated a silicone based lubricant (Dow Corning 111 lubricant) should be applied to the O-rings prior to end cap insertion into the pressure vessel. Care must be taken to avoid any debris or foreign particles at the surface of the O-rings or in the O-ring Grooves. Any particle large enough to be felt with your finger or seen by the caked eye can potentially cause leakage at these locations at high pressures.

c. Sensor End Cap Seals

The pressure sensor and the integral conductivity/temperature sensor each have an O-ring seal. Under normal circumstances the user will never encounter these seals. If for any reason the sensors must be removed, proper care must be observed. Ocean Sensors does not recommend that the user attempt to remove the sensors, unless having previously consulted with Ocean Sensors.

d. Seal Maintenance

Two holes are bored into the pressure sensor end cap. One hole is for the CT sensor and the other is for the pressure transducer. The CT sensor seal and the pressure transducer seal are each competed by the use of a single O-ring. Both sensors are removed with O-ring remaining on the sensor. If a sensor has been removed from the sensor end cap, check for and remove any debris from the O-ring. The CT O-ring Parker number is 2-012. The O-ring for the pressure transducer is a non-standard size and can be acquired through Ocean Sensors. The sensor seals require little maintenance and should be performed by Ocean Sensors.

MSP Computer:

Ampro computers, Inc, manufactured the computer.

The model is the CoreModule/3Sxi

Some of the basic features are:

- PC compatible motherboard architecture

- 25MHz 386SX CPU

- The motherboard has 4MB of RAM

- 2-serial ports providing for RS-232 communication protocol w/ selectable baud

- Offers modular PC/104 expansion bus

To alimnt the need of the project, a 4-channel multi-protocol PC/104 compatible module serial card was installed in order to provide serial communications access to and from the sensors (i.e. CTD, orientation sensor, and turbidity sensor).

Data Logger pressure housing:

For a 6000-m depth capability, the engineers at the Engineering Support Facility have constructed (1) 7071-grade titanium pressure housing, which will host the PC/104 computer plus associated electronics, and the internal 12-Volt lead-acid battery. To date, we plan to mount the CTD and CPU housings of the mission sensor package within the AUV forward of the battery pods and high enough above the skids to minimize damage from the seafloor. The dimensions of the pressure case are: 12” (L) – 7.37” (OD)

Orientation pressure case:

The pressure barrel for the magnetometer was completed at the Engineering Support Facility. Grade-2 titanium was chosen because of its low Iron content as well as it capability to withstand well above the project design conditions, which is about 6000 meters. The dimensions of the pressure case are: 5” (L) – 1.2” (OD)

SYSTEMS SPECIFICATION

Orientation Sensor:

<u>Noise level Magnetometer</u>	<u>5 x 10⁻⁶ G</u>
<u>Noise level Accelerometer</u>	<u>2 x 10⁻⁴ Gee</u>
<u>Linearity</u>	<u>±0.1% FS</u>
<u>Angular accuracy</u>	
<u>Roll and pitch</u>	<u>±0.5°</u>
<u>Azimuth</u>	<u>±1.5°</u>
<u>Axis alignment</u>	<u>±0.2°</u>
<u>Alignment of axes with package reference frame</u>	<u>±0.2°</u>
<u>Power</u>	<u>+5V @ 71 ma or +7 to +12V @ 71 ma</u>
<u>Communications</u>	<u>RS232</u>
<u>Weight</u>	<u>or TTL @ 9600 baud</u>
<u>Size</u>	<u>50 gms</u>
<u>Leads Flying leads 6" long</u>	<u>0.80" x 0.75" x 4.6"</u>

LIGHT BACK-SCATTERING Sensor:

<u>Range</u>	
<u>High-Gain</u>	<u>30 mg/l</u>
<u>Low-Gain</u>	<u>100 mg/l</u>
<u>Resolution (0.01% of full scale)</u>	<u>3 µg/l</u>
<u>Sensor Output</u>	<u>0 ~ 5 VDC</u>
<u>Time Constant</u>	<u>= 0.1 Seconds</u>
<u>Power</u>	<u>9~28 VDC @ 17 mA</u>
<u>Sensor Turn On Time</u>	<u>1 seconds</u>
<u>Temperature Stability</u>	<u>0.5%, 0-50°C</u>
<u>Depth</u>	<u>6000 meters</u>
<u>Weight</u>	<u>0.26 kg in air, 0.13 kg in water</u>

<u>Size</u>	<u>3.2 cm, (1.25") D, 14 cm, (5") L</u>
<u>Material</u>	<u>ABS Plastic housing filled with epoxy</u>
	<u>Clear epoxy optical windows</u>

SUBMERSIBLE PUMP:

<u>Standard input range (#3 winding, 1300-3000 rpm)</u>	<u>10-18 VDC</u>
<u>Standard input range (#3 winding, 4500 rpm)</u>	<u>13-18 VDC</u>
<u>Low input range (#5 winding, 1300-2000 rpm only)</u>	<u>6-16 VDC</u>
<u>Weight</u>	<u>0.7 kg in air, 0.3 kg in water</u>

OS200 CTD:

<u>SERIAL NUMBER</u>	<u>SN-430</u>
<u>Communication Standard</u>	<u>RS-232 (9600 BAUD)</u>
<u>Communication Option</u>	<u>20 mA loop (1200 BAUD)</u>
<u>Data format</u>	<u>ASCII or Binary</u>
<u>Depth</u>	<u>6000 meters</u>
<u>Pressure Vessel Material</u>	<u>Anodized Aluminum standard</u>
<u>Power</u>	<u>Alkaline 7.5 VDC</u>
<u>Dimensions</u>	<u>Diameter 5.7 cm (2.25 in)</u>
	<u>Length 70 cm (28 in)</u>
<u>Weight</u>	<u>2.2 kg in air, 0.6 kg in water</u>
<u>Keyboard selectable values</u>	<u>Conductivity</u>
	<u>temperature</u>
	<u>Pressure</u>
	<u>Salinity</u>
	<u>PH</u>
	<u>Dissolved Oxygen</u>
<u>Maximum Scan Rate (C,T,D)</u>	<u>145 per second</u>
<u>Maximum Rate (single channel)</u>	<u>500 per second</u>

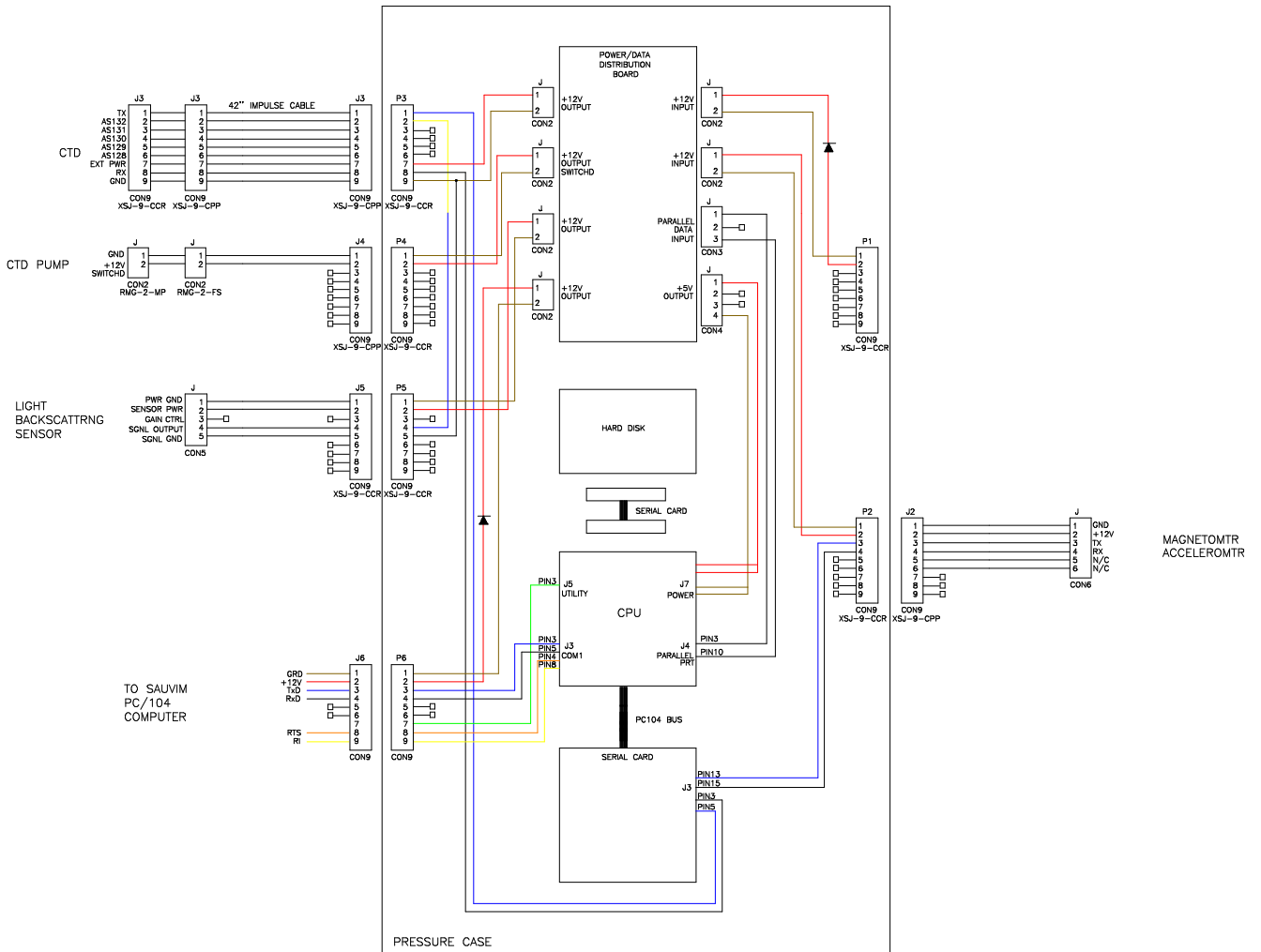


Figure MSP-1. MSP Wiring Diagram

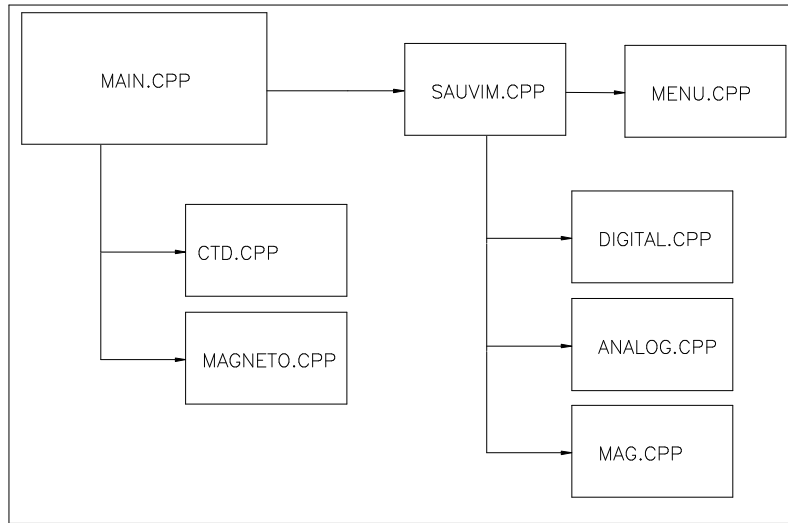


Figure MSP-2. Program Layout

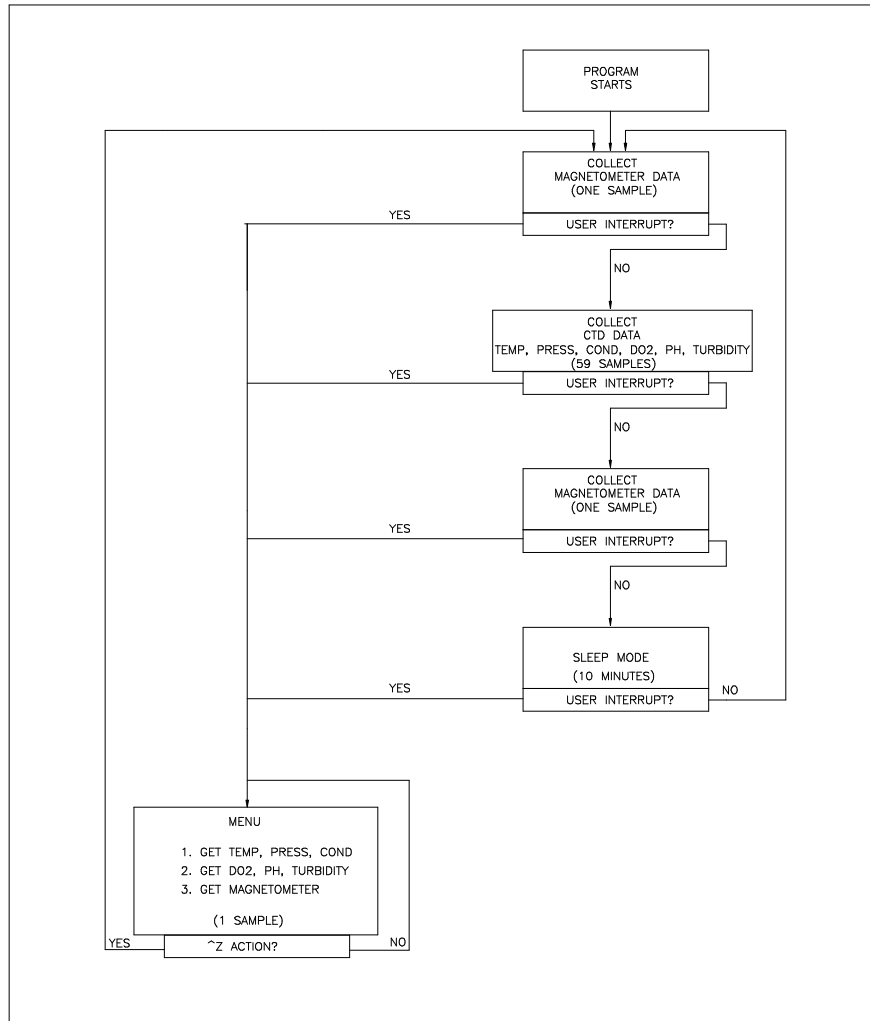


Figure MSP-3. Program Organization

Hydrodynamic Drag Coefficient Analysis (HDCA)

Project Leader(s): Dr. Song K. Choi & Mr. Oliver T. Easterday
Personnel: - none -
Past Project Leader(s): Dr. Farzad Masheyekhi, Dr. Junku Yuh & Dr. Curtis S. Ikehara
Past Personnel: Mr. Brian S.C. Lau

Objectives

- Determination of the hydrodynamic coefficient via numerical solution of full Navier-Stokes equations using commercial CFD code, PHOENICS.
- Provide design recommendations for the vehicle fairing from the hydrodynamic results.
- Perform experiments to verify and confirm the CFD results.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

The CHAM PHOENICS software generated various hardware and software errors when an attempt was made to numerically obtain a reasonable drag number for a known object. Further testing of the software will have to be conducted to confirm and verify other standardized numerical results with empirical data in references.

However, using standardized methods to calculate the coefficient of drag (C_d) of an object or a vehicle, the formula,

$$C_d = F/0.5*\rho*A*V^2,$$

where F is the force in the direction of the flow direction being tested, ρ is the fluid density, A is the frontal area of an object or vehicle, and V is the fluid velocity was used. For the SAUVIM, a coarse grid of $10x10x10$ as provided by CHAM was used, and setting ρ as 998 kg/m^3 , $V = 3\text{m/s}$, and SAUVIM frontal area of 10 m^2 , the software generated a drag coefficient of 0.40.

Future Tasks (Phase II Tasks)

- Fabricate SAUVIM model for testing.
- Compare CFD results to actual test data.
- Correct and modify CFD codes for future use.

Mechanical Analysis and Fabrication (MAF)

Project Leader(s): Mr. Oliver T. Easterday & Dr. Song K. Choi
Personnel: - none -
Past Project Leader(s): Dr. Mehrdad Ghasemi Nejhad
Past Personnel: Dr. Ali Yousefpour, Mr. Eric Sung, Mr. Bruce Flegal, Mr. Robert Ng, Mr. Mark Uyema, Mr. Saeid Pourjalali, Ms. Melanie Yamauchi & Mr. Reid Takaiya

Objectives

Mechanical Analysis and Fabrication (MAF) group is responsible for designing, analyzing, manufacturing, and testing of pressure vessels and flooded fairing as well as analyzing the metallic frame of the vehicle.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

1. DESIGN AND ANALYSIS OF PRESSURE VESSEL

To design the pressure vessels for the SAUVIM project various finite element analyses for different modes of failure have to be performed. The structure is mainly analyzed for stress and buckling mode of failure. Design constraints of the pressure vessels are: an effective length of 18", an inner diameter of 13", and a design pressure of 8,976 psi, which corresponds to 20,013 feet (6,100 meters) depth. It should be noted that the operating depth of the vehicle is 19,685 feet (6,000 meters), which corresponds to a pressure of 8,840 psi. Figure MAF-1 shows the development methodology for the pressure vessels employed herein. Two modes of failure are considered, and they are i) stress and ii) buckling analyses taking hygrothermal effects into account. Ti-6Al-4V, Graphite/Epoxy, and APC-2/AS4 are three candidate materials for the pressure vessels. The results of the analyses reveal the most desirable material and required thickness of the pressure vessels. Next, end-caps are designed. Then, detailed analysis of the pressure vessels and the end-caps are performed. Finally, the pressure vessel and end-caps are fabricated and tested.

1.1. Candidate Materials and Material Properties

Ti-6Al-4V, Graphite/Epoxy, and APC-2/AS4 are three candidate materials for the pressure vessels. Ti-6Al-4V is a popular material for manufacturing pressure vessels for underwater vehicles. It is a lightweight material, and due to its excellent corrosion resistance, it is the best metallic material for marine structures. Ti-6Al-4V has a relatively high strength-to-weight ratio, good high-temperature properties, and high mechanical performance (Askeland84). Table MAF-1 gives the mechanical properties of Ti-6Al-4V.

APC-2/AS4 thermoplastic composite and Graphite/Epoxy thermo set composite are the two candidate composite materials. Both of these materials have high specific properties, good corrosion resistance, and low density. Table MAF-2 gives unidirectional mechanical properties for these materials (ICI Thermoplastic Composite92, Mallick93).

1.2. Analysis Procedure

Figure MAF-2 shows the analysis procedure for designing a pressure vessel. Stress and buckling analyses taking hygrothermal (i.e., moisture and temperature) and end-cap effects into account are performed to achieve an optimum wall thickness for the pressure vessel. The following explains the analysis procedure flowchart in Figure MAF-2. Stress analysis is used for the structure to determine the required wall thickness for the pressure vessel according to the design constraints at room temperature, 70°F. A stress factor of safety of around two is required. Von Mises failure criterion is considered for the metallic pressure vessel. For the composite pressure vessel, Maximum Principal Stress criterion is considered. The stress analysis is also performed considering thermal effects, i.e., extreme temperatures of 32°F and 140°F, for the metallic pressure vessel. The hygrothermal effects, i.e., extreme temperatures of 32°F and 140°F as well as the maximum moisture absorption, are considered for the stress analysis of composite pressure vessels. In the case of the metallic pressure vessel, the moisture effect is zero. Next, non-linear buckling analysis is performed considering hygrothermal and end-cap effects to investigate the stability of the pressure vessel under the design constraints. To perform the non-linear buckling analysis, an eigenvalue buckling analysis is first employed to determine the first buckling mode shape and bifurcation pressure of the pressure vessel. These results are then used in the non-linear buckling analysis. Buckling factor of safety of at least two is required for the pressure vessel. Finally, the non-linear stress analysis considering hygrothermal and end-cap effects are also performed to investigate the linear nature of the problem.

1.2.1. Hygrothermal Effects

The effects of temperature and moisture (hygrothermal) were considered in the design of the pressure vessels. Two extreme cases were considered. For the metallic pressure vessel, the extreme temperatures were a) 32°F and b) 140°F and the moisture effect was zero. For the composite pressure vessel, in addition to the thermal effect, the effect of moisture absorption was also considered. Coefficient of Moisture Expansion (CME) is not an input material property for a composite material in ANSYS finite element software. To apply the effect of moisture absorption, the CME was modeled as an equivalent Coefficient of Thermal Expansion, CTE, and is introduced here as the Moisture Equivalent Coefficient of Thermal Expansion, MECTE.

1.2.2. Moisture Equivalent Coefficient of Thermal Expansion, MECTE

The MECTE was evaluated and then combined with the CTE of the composite material to give a Modified Coefficient of Thermal Expansions (MCTE), which was used in the analysis. The hygrothermal effects in principal directions are dilatational and can be shown by the following equation:

$$\varepsilon_i = \alpha_i \Delta T + \beta_i \Delta m \quad (i=1, 2, 3) \quad (1)$$

where

- ε_i = Strain due to hygrothermal effects
- α_i = Coefficient of thermal expansion, CTE
- β_i = Coefficient of moisture expansion, CME
- ΔT = Change of temperature
- Δm = Change in percentage weight

It can be assumed that the ε_i is equal to the MCTE, $\bar{\alpha}_i$, times ΔT (see Equation 2):

$$\varepsilon_i = \bar{\alpha}_i \Delta T = \alpha_i \Delta T + \beta_i \Delta m \quad (i=1, 2, 3) \quad (2)$$

By dividing Equation 2 by ΔT , the MCTE, which also includes the effect of moisture absorption, MECTE, can be obtained as,

$$\bar{\alpha}_i = \alpha_i + \beta_i \frac{\Delta m}{\Delta T} \quad (i=1, 2, 3) \quad (3)$$

The MCTE for the extreme temperatures and maximum moisture absorption were determined using Equation 3. In Equation 3, the LHS is MCTE, the first term on the RHS is CTE, and the second term on the RHS is MECTE. The maximum value for Δm is 0.23% for APC-2 (thermoplastic resin) and 5% for Epoxy (thermo set resin) (ICI Thermoplastic Composite92, Mallick93). α_i and β_i for the APC-2/AS4 and Graphite/Epoxy are given in Table MAF-2 (ICI Thermoplastic Composite92, Mallick93). Table MAF-3 gives the MCTE at 140°F and 32°F, for both composite systems.

1.3. Stress Analysis

Finite Element Analysis (FEA) was performed to determine the required thickness of the pressure vessel. Due to the axi-symmetric geometry, material properties, loading, and boundary conditions, ten degrees wedge of the pressure vessel was modeled (see Figure MAF-3). Modeling wedge section of the pressure vessel reduces the computational time and memory requirements significantly. A cylindrical coordinate system was used with its origin at the center of the pressure vessel and $X \equiv R$, $Y \equiv \theta$, and $Z \equiv Z$. Axi-symmetric boundary conditions were applied at $Y=0^\circ$ and $Y=10^\circ$ planes, a symmetric boundary condition was applied at the $Z=0$ plane (see Figure MAF-3). Hydrostatic pressure was applied. Also, the end-cap was simulated as radial simply supported boundary and axial hydrostatic pressure load conditions at the other end of the model.

1.3.1. Effects of End-cap

A contoured-end plug-supported end-cap was considered for the pressure vessel. A large radius circular taper, R , is incorporated into the end-cap to control both bending and shear stresses near the cylinder ends (see Figure MAF-4). The cylinder ends can conform to the circular tapered section as pressure increases which reduces the stress concentration at the ends and enhances the performance of the pressure vessel (Leon95; Yousefpour00a).

The contoured-end plug-supported end-cap was modeled as radial displacements at the cylinder end. A parametric study was performed to determine the optimum radius circular taper, R , based on the stress factor of safety (Yousefpour00a). The length of the tapered section was also optimized and found to be 1.5". This length was also required to place two radial O-rings for sealing purposes. To calculate 'R', an edge simply supported pressure vessel with 21" in length was modeled and analyzed under a hydrostatic (i.e., radial and axial) design pressure of up to 10,300 psi corresponding to 7,000 meters depth. Table MAF-4 gives the selected thickness for the pressure vessels, which were determined by trial-and-error based on a factor of safety of around 2. The radial displacement, d , at 1.5" from the cylinder end was determined (see Figure MAF-4). Figures MAF-5, MAF-6, and MAF-7 show the radial displacements from the end to the mid-length of the 21" long Ti-6Al-4V, APC-2/AS4,

and Graphite/Epoxy cylinders with edge simply supported boundary condition, respectively. Table MAF-4 also gives the selected thickness and the corresponding ‘d’ values for the candidate materials. Although the maximum deflections of all pressure vessels, at the mid-length (see Figures MAF-5, MAF-6, and MAF-7), were close, however the ‘d’ and thickness values for the Ti-6Al-4V were smaller than those for the composite pressure vessels. This is believed to be due to a higher flexural rigidity of the Ti-6Al-4V pressure vessel. $D_{zz}=3.041 \times 10^6$ lb-in for the composite and $D_{zz}=Eh^3/12(1-\nu^2)=3.7 \times 10^6$ lb-in for Ti-6Al-4V, in this case. Simple trigonometry was applied to calculate ‘R’ for the given ‘d’. For parametric analysis, different percentages of ‘d’ were considered and corresponding R’s were calculated and applied to the finite element model as radial displacement boundary conditions. The curve of the end-plug is part of a circle with radius R. Figure MAF-8 shows the end-cap and its plug circle with radius R. Assume the X-Y coordinate system at the center of the circle. The equation of circle can be written as:

$$X^2 + Y^2 = R^2 \quad (4)$$

Since $X=a$ and $Y=R-d$ is a point on the circle, Equation 4 can be written as:

$$a^2 + (R - d)^2 = R^2 \quad (5)$$

After simplifying Equation 5, the radius R can be obtained from the following equation:

$$R = \frac{a^2 + d^2}{2d} \quad (6)$$

It should be noted that ‘a’ is equal to 1.5” and ‘d’ is radial displacement at 1.5” from the cylinder end. Table MAF-5 gives the radius of the end-cap circular taper, R, for different percentages of ‘d’ for the candidate materials. Zero percentage of ‘d’ means no tapered on the end-cap. Von Mises failure criterion was considered for metallic pressure vessel (Shigley89). For the composite pressure vessel, Maximum Principal Stress criterion was considered (Vinson87; Mallick93). Figures MAF-9, MAF-10 and MAF-11 show the stress factor of safety of the pressure vessels as a function of the different percentages of ‘d’ for the candidate materials. The results revealed an optimum ‘R’ for each pressure vessel. Maximum factor of safety of 1.96 was obtained for Ti-6Al-4V pressure vessel at the circular tapered radius of 420”, which corresponds to 20% ‘d’ value (see

Table MAF-5). For the APC-2/AS4 and AS4/Epoxy the maximum factor of safety was 1.96 ($R=131$ ” and $d=40\%$) and 1.90 ($R=135$ ” and $d=40\%$), respectively. The results clearly show that the optimum radius circular taper, R, which is incorporated into the end-caps, increases the performance of the pressure vessel, when the optimum percentage of ‘d’ value is selected.

1.3.2. Stress Analysis of Ti-6Al-4V Pressure Vessel

Table MAF-6 presents the results of stress analyses for the Ti-6Al-4V pressure vessel with the optimum R, 420”, and wall thickness of 1.35” at 32°F, 70°F, and 140°F under a design pressure of 10,300 psi. Desired stress factor of safety was achieved for each case. The results revealed that the thermal effects did not cause significant effects on the performance of the pressure vessel. The maximum Von Mises stress and strain occur at the end-cap and maximum deflection happens at the mid-length for different temperatures (see Table MAF-6).

Figures MAF-12 and MAF-13 show the strain and stress distributions through the thickness at the mid-length of the Ti-6Al-4V pressure vessel at 70°F, respectively. The results of stress had the same trend as the verification model for the metallic pressure vessel. Both axial and hoop strain were compressive however the radial strain was tensile although the whole structure was under hydrostatic pressure (see Figure MAF-12). This was due to the axial pressure and Poisson's ratio effects. Compressive axial, hoop, and radial stresses should cause compressive axial, hoop, and radial strains. However, the compressive radial strain, due to compressive radial stress (see Figure MAF-12), was smaller than axial pressure and Poisson's ratio effects. The results would be the extensional radial strain through the thickness at the mid-length of the pressure vessel. It should also be mentioned that radial stress satisfies the radial stress boundary conditions at the inner (zero stress) and outer (external pressure) radii.

1.3.3. Stress Analysis of Composite Pressure Vessels

A symmetric sub-laminate configuration of $[90/90/0/0/90/90]_s$ was chosen for the composite pressure vessels. The symmetric cross-ply lay-up configuration for the composite pressure vessels eliminates extension-bending, extension-shear, and bending-twisting couplings in the structure. The 2:1 hoop-to-axial ply ratio is chosen since the theoretical hoop stress for metallic pressure vessel is twice of the axial stress (Hyer, 1989).

Table MAF-7 gives the stress analysis results for APC-2/AS4 and Graphite/Epoxy pressure vessels considering hygrothermal effects. Modified Coefficient of Thermal Expansion (MCTE) was used for 32°F and 140°F cases with the maximum moisture effect (see Table MAF-3). The results revealed that the hygrothermal effects did not cause significant difference on the performance of the APC-2/AS4 pressure vessel. However, the hygrothermal effects were severe on the performance of the Graphite/Epoxy. This was due to the high moisture absorption of the Epoxy in the water, i.e., $\Delta m=5\%$. The maximum radial and axial stresses and strains occur at the end-cap of the pressure vessel. The maximum hoop stress and strain happen at the mid-length of the composite pressure vessels. The maximum displacement occurs in the axial direction of the composite pressure vessels. These locations were the same at different temperatures.

Figures MAF-14, MAF-15, MAF-16, and MAF-17 show the strain and hoop, axial, and radial stress distributions through the thickness at the mid-length of the APC-2/AS4 pressure vessel under a design pressure of 10,300 psi, respectively. The results had the same trend as Hyer's solution (Hyer88; Yousefpour00a). It should be noted that the radial strain was tensile at the inner radius and became compressive at the outer radius (see Figure MAF-14). However, the radial strain was tensile for the metallic pressure vessel through the thickness (see Figure MAF-12). For the same reason that explained for the metallic pressure vessel, the radial strain was tensile close to the inner radius. However, the compressive radial strain, due to the hydrostatic pressure, overcomes the Poisson's ratio effect close to the outer radius. This resulted in the compressive radial strain close to the outer radius. The discontinuity of the radial strain was due to the layer interfaces and element boundaries (see Figure MAF-14). In the hoop direction, fibers in 90-degree layers took most of the loads and 0-degree fibers carried less loads (see Figure MAF-15), and in the axial direction, fibers in 0-degree layers took most of the load and 90-degree fibers carried less loads (see Figure MAF-16). It should be noted that the right choice of 2:1 fibers in the hoop direction resulted in an efficient structure where all fibers in hoop (90° layers in Figure MAF-15) and axial (0° layers in Figure MAF-16) directions carry equivalent loads, unlike a metallic structure where stress in hoop direction is twice of that in axial direction. Radial stresses also satisfy radial stress boundary conditions. Stress discontinuities, which

are due to the lay-up configuration, are clearly observed in Figures MAF-15, MAF-16, and MAF-17. Figures MAF-18, MAF-19, MAF-20, and MAF-21 show the strain and stress distributions of the Graphite/Epoxy composite pressure vessel. Similar trend as APC-2/AS4 pressure vessel was observed for the Graphite/Epoxy pressure vessel. Mesh convergence study was performed for all stress analyses with a convergence criterion of 5%.

1.4. Buckling Analysis

Buckling phenomenon occurs in a pressure vessel under a hydrostatic pressure when most of its strain energy, which is stored as membrane energy can be converted to the bending energy (Bushnell85), which requires large deflections. Two types of buckling analyses: a) bifurcation buckling or eigenvalue buckling analysis and b) non-linear buckling analysis were performed in this work.

1.4.1. Eigenvalue Buckling Analysis

In the eigenvalue buckling analysis, the theoretical buckling pressure for an ideal linear elastic structure is calculated. At the eigenvalue buckling pressure (the bifurcation pressure), the deformation starts to follow a new pattern on the load-deflection curve, which is different from the pre-buckling pattern (Bushnell85).

The finite element eigenvalue buckling analysis of the metallic and composite pressure vessels were performed using ANSYS, 1999. The wall thicknesses of the metallic and composite pressure vessels were the same as those were determined in the stress analyses. Figure MAF-22 shows a typical first buckling mode shape of the Ti-6Al-4V pressure vessel. A typical first buckling mode shape of the composite pressure vessels, i.e., APC-2/AS4 and Graphite/Epoxy, is shown in Figure MAF-23.

Due to the axi-symmetric geometry, material properties, loading, and boundary conditions, one-sixth and one-fourth (to include the snap-through portion within the model) of the metallic and composite pressure vessels were modeled and are given in Figures MAF-24 and MAF-25, respectively. A cylindrical coordinate system was used with its origin at the center of the pressure vessel and $X \equiv R$, $Y \equiv \theta$, and $Z \equiv Z$. Axi-symmetric boundary conditions were applied at the $Y=0^\circ$ and $Y=120^\circ$ planes and symmetric boundary condition was applied at the $Z=0$ plane for the Ti-6Al-4V pressure vessel (see Figure MAF-24). For the composite pressure vessels, axi-symmetric boundary conditions were applied at the $Y=0^\circ$ and $Y=180^\circ$ planes and symmetric boundary condition was applied at the $Z=0$ plane (see Figure MAF-25). The effect of the end-cap was modeled as radial plug simply supported boundary condition at the other end (see Figure MAF-24 and Figure MAF-25). Figures MAF-26 and MAF-27 show the typical first modal shape of one-sixth and one-fourth of the Ti-6Al-4V and composite pressure vessels under external radial pressure, respectively. Table MAF-8 gives the bifurcation pressure of the pressure vessels with pre-determined thicknesses that were obtained from the stress analyses. The bifurcation pressure of the pressure vessels was considerably higher than a design pressure of 10,300 psi.

1.4.2. Non-Linear Buckling Analysis

Non-linear buckling pressure can be evaluated using non-linear stress analysis by observing the first change in the slope (i.e., stiffness of the structure) in the load-deflection curve (Bushnell85). The wall thickness of the pressure vessels that were calculated in the stress analyses was used in non-linear buckling analysis. Also, same optimized tapered end-cap boundary conditions established in the stress

analysis was used in this section. To employ the non-linear buckling analysis small out-of-plane perturbations were applied to the model (geometric imperfections) to make the structure unstable as the pressure increases. The perturbations can be small out-of-plane forces, or specified displacements (see Figures MAF-24 and MAF-25). The mode shape obtained from the eigenvalue buckling analysis was used to predict the location and magnitude of the perturbations to stimulate the desired buckling response (see Figures MAF-26 and MAF-27). The non-linear buckling analysis was performed for Ti-6Al-4V, APC-2/AS4, and Graphite/Epoxy taking hygrothermal and end-cap effects into account, and was found that all three materials had a minimum buckling factor of safety of 3.6. The end-cap was modeled as radial tapered simply supported (for the circular tapered plug portion) and axial external pressure under a hydrostatic pressure.

1.5. Non-Linear Stress Analysis

Non-linear stress analysis was applied to the pressure vessel models that were developed from linear stress analysis to investigate the linear behavior of the structure. The results of the non-linear stress analyses for the Ti-6Al-4V pressure vessel is given in Table MAF-9. Table MAF-6 gives linear stress analysis results. The results are identical. Table MAF-10 gives the results on the non-linear stress analysis for the composite pressure vessels. Table MAF-7 gives linear stress analysis results. The results were in good agreement. Generally, the non-linear stress analysis results revealed the linear nature of the stress problem.

1.6. Hygrothermal Effects on the Pressure versus Radial Displacement

Typical pressure vs. radial displacement curve at the mid-length of Ti-6Al-4V, APC-2/AS4, and Graphite/Epoxy pressure vessels taking hygrothermal and end-cap effects into account are shown in Figures MAF-28, MAF-29, and MAF-30, respectively. It can be observed that the thermal effects did not cause significant difference on the performance of the metallic pressure vessel. The thermal effects just shifted the stability curve parallel to the room temperature (70°) stability curve either upward (for 140°F) or downward (for 32°F) (see Figure MAF-28). These results were also true for the APC-2/AS4 pressure vessel. However, hygrothermal effects in the APC-2/AS4 pressure vessel were less than thermal effects in the metallic pressure vessel. At a given pressure, the radial displacement is lower for higher temperature since the effect of higher moisture and temperature in the radial direction is the opposite of the hydrostatic pressure. The hygrothermal effects were more severe in the Graphite/Epoxy pressure vessel. This was due to high moisture absorption, $\Delta m=5\%$, of the Epoxy resin (see Figure MAF-30).

1.7. Material Selection

APC-2/AS4 was chosen as the material for the pressure vessel. This material has following characteristics: i) high strength and stiffness, ii) low coefficient of moisture absorption, iii) excellent corrosion and solvent resistant, iv) good thermal conductivity in fiber direction, v) low density, vi) high fracture toughness, vii) high damage tolerance, viii) high impact resistant, ix) good fatigue resistance, x) recyclable, xi) reprocessible, xii) repairable, xiii) reshapable, xiv) reformable, and xv) ease of fabrication (ICI Thermoplastic Composite, 1992). In addition, the extra damping and lower electromagnetic observable inherent in the APC-2/AS4 further enhance its performance (ICI Thermoplastic Composite, 1992). These features are desirable in order to decrease the weight of the vehicle, increase the speed and operating depth, and increase the serviceability and survivability of the vehicle, and provide a safe operation. The comparison of the APC-2/AS4 and Ti-6Al-4V pressure vessels shows that although the wall thickness of the APC-2/AS4 pressure vessel is greater than Ti-

6Al-4V but its density is almost three times less than that for Ti-6Al-4V (see Table MAF-11). The dry weight of APC-2/AS4 pressure hull is 91 lbs and that for Ti-6Al-4V pressure hull is 204 lbs. This result shows that the APC-2/AS4 pressure hull is 2.24 times lighter than Ti-6Al-4V pressure hull. In the water, disregarding the weight of the end-caps (which would be the same for both), the APC-2/AS4 composite pressure hull is weight-less and has the buoyancy force (upward) of 58 lbs. However, the Ti-6Al-4V pressure hull wet weight (downward) in the water is 69 lbs. Lower weight of the APC-2/AS4 pressure vessel reduces the total dry weight of the vehicle and the energy consumption of the vehicle (mainly from batteries) as well as the cost. Roughly for every wet lb. of the vehicle 3 dry lbs. of synthetic foam is needed to maintain a buoyant vehicle. The APC-2/AS4 pressure hull has the same weight as the Graphite/Epoxy pressure hull. The hygrothermal effects are negligible for the APC-2/AS4 pressure vessel (see Table MAF-3 and Figure MAF-29). The hygrothermal effects did not change the factor of safety of the APC-2/AS4 pressure vessel (see Table MAF-7 and Figure MAF-30) considerably. However, the hygrothermal effects were severe in Graphite/Epoxy pressure vessel (see Figure MAF-7 and Figure MAF-30). This was due to the higher moisture absorption of Epoxy (thermo set resin) over that of APC-2 (thermoplastic resin). The thermal effect is more considerable in the metallic pressure vessel than hygrothermal effects in the APC-2/AS4 pressure vessel (see Tables MAF-6 and MAF-7), and they were both affected less compared with Graphite/Epoxy (see Figures MAF-28, MAF-29, and MAF-30). Manufacturing of APC-2/AS4 pressure vessel is comparable to the Ti-6Al-4V and the final composite product needs less machining than the Ti-6Al-4V pressure vessel. It should be noted that the material waste during the manufacturing for Ti-6Al-4V pressure hull is much more severe than that for the APC-2/AS4. Manufacturing of thick APC-2/AS4 pressure hull is easier than that of thick Graphite/Epoxy pressure hull. The Graphite/Epoxy is a thermo set composite and due to its exothermic property, it is not suitable for the manufacture of thick-section composite structures. Also, since the thermo set pressure hull needs to be cured in the autoclave after the winding, the final product gains large residual stresses, which reduce the performance of the structure. However, the thermoplastic composite has endothermic property, and an in-situ thermoplastic filament winding technique can be used to manufacture the thick-section thermoplastic pressure hull. The consolidation can be achieved during the winding using in-situ heat sources and a consolidation pressure roller. On-line consolidation considerably reduces the residual stresses in the structure (Nejhad92a, Nejhad92b, Nejhad94), which yields final products with better quality and performance. No post-curing is necessary for the thermoplastic composite when is processed in-situ. In addition, in a thermo set wet filament winding, particularly for thick section composites, the tension build-up causes a fiber migration towards the mandrel which, in turn, leads to fiber waviness under pressure, due to the slacks created by the fiber migration. The fiber waviness is not desirable since it can trigger fiber micro buckling under local compressive loads, which is the case for a pressure vessel under external hydrostatic pressure. A micro buckling phenomenon can lead to a premature catastrophic failure of the pressure vessel.

1.8. Design of Titanium End-caps

End-caps are required to close the ends of the APC-2/AS4 pressure vessel. Under ocean hydrostatic pressure, the pressure vessel and end-caps are under radial and axial pressure, and they mutually affect each other. Finite Element Analysis (FEA) was performed to study the stress distributions in metallic end-caps with maximum six holes for connectors and one hole for a vacuum valve, i.e., a total maximum holes of seven. The diameter of each hole was one inch. Due to axi-symmetric geometry, material properties, loading, and boundary conditions, 51.43 degrees wedge of the end-cap was modeled in the FEA. A cylindrical coordinate system was used, $X \equiv R$, $Y \equiv \theta$, and $Z \equiv Z$, (see Figure MAF-31). Axi-symmetric boundary conditions were applied at $Y=0^\circ$ and $Y=51.43^\circ$ planes. Hydrostatic pressure was applied at the exposed surfaces of the end-cap. Also, simply supported

boundary condition was applied at the axial mating surfaces of the end-cap and pressure hull. The radial stress calculated from the pressure vessel analysis was applied at the radial mating surfaces of the end-cap and pressure hull. Von-Mises failure criterion (Shames, 1989) was considered for the design of the metallic end-cap. The end-cap is made of Ti-6Al-4V (see Table MAF-1) and is designed for 10,300 psi with a stress factor of safety of about two. Figure MAF-31 shows the Von-Mises stress distribution for the end-cap. The maximum stress of 60,731 psi occurs at the holes. Figure MAF-32 shows the deformation of the end-cap. The maximum deformation of 0.022 inches happens at the center of the end-cap. The dimensions of the end-cap and locations of the connectors are shown in Figure MAF-33.

1.9. SAUVIM Pressure Vessel Design

In the previous section APC-2/AS4 was selected as the material of choice for the deep ocean pressure vessel up to a design depth of 7,000 m (corresponding to 10,300 psi). The SAUVIM vehicle will operate up to 6,000 m depth. Therefore, this section presents the design of the SAUVIM pressure vessels for a design depth of 6,100 m, which corresponds to 8,976 psi. The pressure vessels for the SAUVIM vehicle have two different sizes. The design constraints of the pressure vessels are: total lengths of 21" and 19", an inner diameter of 13", and a design pressure of 8,976 psi which corresponds to 20,013 feet (i.e., 6,100 meters) depth. Symmetric sub-laminate configuration of $[(90/90/0/0/90/90)]_4$ is chosen for the composite pressure vessel. The symmetric cross-ply lay-up configuration for the composite pressure vessels eliminates extension-bending, extension-shear, and bending-twisting couplings in the structure. The 2:1 hoop-to-axial ply ratio is chosen since the theoretical hoop stress for metallic pressure vessel is twice of the axial stress (Hyer, 1988). The thickness of a lamina is 0.0055". Stress analysis was performed to determine the required thickness of the pressure vessels with tapered end-cap for SAUVIM vehicle. The thickness was determined by trial-and-error based on a factor of safety of around 2. The thickness of the pressure vessel was selected to be 1.188". The material property of APC-2/AS4 is given in Table 3.2. APC-2/AS4 material is selected employing in-situ thermoplastic composite filament winding technique to manufacture the pressure hull.

A contoured end-plug end-cap was considered (see Figure MAF-4). The goal of using this end-cap is to improve the performance of SAUVIM pressure vessels by reducing the bending and shear stresses at the ends. A parametric study is performed to determine the optimum value for the tapered radius for the 21" and 19" pressure vessels (see section 2.3.1). The length of the plug is optimized and found to be 1.5" (Yousefpour00a). This length was also required to place two radial O-rings for sealing purposes. Stress analysis was performed to determine the required thickness of the pressure vessels with tapered end-cap for SAUVIM vehicle. The thickness was determined by trial-and-error based on a factor of safety of around 2. The thickness of the pressure vessel was selected to be 1.188". The methods were explained in section 2.3.1. For the tapered end-caps, the radial displacements, d , at 1.5" from the end of the 21" and 19" cylinders were 0.024" (see Figure MAF-34). The values were obtained from edge simply supported model (see section 2.3.1). Table MAF-12 gives the taper radius of the end-cap with the corresponding stress factor of safety for different percentage of 'd'. Zero percentage of 'd' means the end-cap has no tapered radius and yields identical plug boundary conditions.

Maximum factor of safety of 1.67 was obtained for 21" and 19" pressure vessels at the circular tapered radius of 115" which corresponds to 40% 'd' value (see Table MAF-12). The results clearly show that the optimum radius circular taper was incorporated into the end-cap, enhanced the performance of the pressure vessel compared with a plug end-cap with no taper. Table MAF-13 gives

the stress analysis results for APC-2/AS4 pressure vessels with optimum tapered radius, 115" with a wall thickness of 1.188" at 32°F, and 70°F, and 140°F under a design pressure of 8,976 psi. The results revealed that the hygrothermal effects did not cause significant difference on the performance of the pressure vessels. For the pressure vessels with tapered end-caps, the maximum radial and axial stresses and strains occur at the end-cap

As mentioned before, in the eigenvalue buckling analysis, the theoretical buckling pressure for an ideal linear elastic structure is calculated. At the eigenvalue buckling pressure (the bifurcation pressure), the deformation starts to follow a new pattern on the load-deflection curve, which is different from the pre-buckling pattern (Bushnell, 1985). First mode shapes of the 21" and 19" composite pressure vessels for SAUVIM are shown in Figures MAF-35 and MAF-36, respectively. Due to the axi-symmetric geometry, material properties, loading, and boundary conditions, one-sixth of the composite pressure vessels were modeled (see Figure MAF 37 and MAF-38). Table MAF-14 gives the bifurcation pressure of the pressure vessels for SAUVIM with pre-determined thickness (i.e., 1.188") that was obtained from the stress analyses. The bifurcation pressures of the pressure vessels were considerably higher than the design pressure of 8,976 psi.

As mentioned before, non-linear buckling pressure can be evaluated using non-linear stress analysis by observing the first change in the slope (i.e., stiffness of the structure) in the load-deflection curve (Bushnell85). The wall thickness of the pressure vessels that were calculated in the stress analyses was used in the non-linear buckling analysis. Also, same optimized tapered end-cap boundary conditions established in the stress analysis was used in this section. Since the eigenvalue buckling pressure was very high, the non-linear buckling analysis was performed up to 17,952 psi, which gave a minimum buckling factor of safety of two for the pressure vessel. The pressure-displacement curve of the 21" and 19" APC-2/AS4 pressure vessels with optimum tapered radius taking hygrothermal effects into account are shown in Figures MAF-39 and MAF-40, respectively. The results show that the pressure vessels did not lose their stability up to 17,952 psi. It can be observed that the hygrothermal effects did not cause significant difference on the performance of the pressure vessels.

Non-linear stress analysis was applied to the pressure vessel models for SAUVIM to present the linear behavior of the structure. Table MAF-15 shows the results on the non-linear stress analysis for the composite pressure vessels. The results were in good agreement (see Table MAF-13). Generally, the non-linear stress analysis results revealed the linear nature of the stress problem.

1.10. Design of Titanium End-cap for SAUVIM

One size end-cap was designed for 21" and 19" pressure vessels with optimum tapered radius for SAUVIM. Finite Element Analysis (FEA) was performed to determine the stress distributions in metallic end-caps with maximum six holes for connectors and one hole for vacuum valve, i.e., a total maximum holes of seven. The diameter of each hole was one inch. The end-cap was made of Ti-6Al-4V and designed for 8,976 psi with a stress factor of safety of about 2. Figure MAF-41 shows the Von-Mises stress distribution of the end-cap. The maximum stress of 60,880 psi occurs at the holes. Figure MAF-52 shows the deformation of the end-cap. The maximum deflection of 0.024 inches happens at the center of the end-cap. The dimensions of the end-cap and locations of the connectors are shown in Figure MAF-43.

1.11. Scaled Thermoplastic Composite Pressure Vessel and End-caps

The analyses for the metallic and composite pressure vessels revealed that APC-2/AS4 pressure vessel in general has advantages over the Ti-6Al-4V and Graphite/Epoxy pressure vessels. An APC-2/AS4 scaled pressure vessel was designed. The length and inner diameter of the scaled pressure vessel were chosen to be approximately one-third of those for the main pressure vessel - the design of which was reported earlier. The length and inner diameter of the scaled pressure vessel were 6.625" and 4.18", respectively. The thickness of the scaled pressure vessel was fixed and chosen to be 0.24" in order to have a thick wall pressure vessel. A symmetric cross-ply sub-laminate configuration of $[90/90/0/0/90/90]_s$ was chosen for the composite scaled pressure vessel to avoid extension-bending, extension-shear, and bending-twisting couplings in the structure. The 2:1 hoop-to-axial ply ratio is chosen since the theoretical hoop stress for metallic pressure vessel is twice of the axial stress (Hyer88). Finite element analyses were performed on the scaled model pressure vessel using ANSYS software. Due to axi-symmetric geometry, material properties, loading, and boundary conditions, ten degrees wedge of the circular cylinder was modeled in the FEM stress analysis. Cylindrical coordinate system was considered. Modeling a wedge portion of the pressure vessel reduces the computational time and memory requirements significantly. The stress analysis revealed that the scaled model, with end-caps in place and modeled as simply supported boundary conditions, could sustain a pressure up to 3,500 psi with a stress factor of safety of about 2.2 using Maximum Principal Stress failure criterion (Vinson87). For the scaled model a plug-supported end-cap was used. Figures MAF-44, MAF-45, and MAF-46 show the radial, axial, and hoop stress distributions, respectively. Also, Figures MAF-47, MAF-48, and MAF-49 show the radial, axial, and hoop strain distributions, respectively.

The maximum radial stress and strain occur at the end-cap of the scaled pressure vessel, i.e., -5,665 psi and 0.00372 in/in, respectively. The maximum axial stress and strain occur at the end-cap of the scaled pressure vessels, i.e., -11,365 psi and -0.00308 in/in, respectively. The maximum hoop stress of -51,682 psi, and maximum strain of -0.00252 in/in, happens at the inner surface at the mid-length of the scaled pressure vessels. The maximum displacement of 0.00716" occurs in the axial direction of the composite pressure vessels. At the mid-length of the scaled pressure vessel, the radial stress and strains are -3,119 psi and 0.00134 in/in, respectively. The axial stress and strain are -50,801 and -0.00214 in/in, respectively, at the mid-length of the scaled pressure vessel. The hoop stress and strain are -52,924 psi and -0.00251 in/in at the mid-length of the scaled pressure vessel. The maximum deflection at the mid-length is 0.00527".

Figure MAF-50 shows the first mode of the buckling shape for the scaled pressure vessel. Due to the axisymmetric geometry, material properties, loading, and boundary conditions, one-sixth of the scaled pressure vessel (i.e., one-third circumferentially and one-half axially) was modeled and the cylindrical coordinate system was considered (see Figure MAF-51). The eigenvalue buckling pressure for the pressure vessel was about 13,942 psi. Non-linear buckling analysis (ANSYS99) was performed on the structure to investigate the stability of the scaled pressure vessel up to 7,000 psi pressure, with end-caps in place modeled as radial simply supported and axial external pressure boundary conditions. The mid-length pressure-deflection curve of the APC-2/AS4 scaled pressure vessel is shown in Figure MAF-52. It was found that the scaled pressure vessel would not lose its stability up to 7,000 psi.

A plug-supported end-cap with a hole at the center for a connector was considered for the scaled pressure vessel. The end-cap was made of stainless steel and designed for 3,500 psi with a factor of safety of about three. The stainless steel has Young's Modules of 30 Msi, Poisson's ratio of 0.3, and yield strength of 40 Ksi. Figure MAF-53 shows the Von-Mises stress distribution of the scaled end-cap. The maximum stress of 13,086 psi occurs at the center holes. Figure MAF-54 shows the displacement of the scaled end-cap. The maximum displacement is radial and happens at the center of

the end-cap around the hole and is 0.00054 inches. The dimensions of the end-cap are shown in Figure MAF-55. The end-cap has two radial O-rings for sealing.

2. Manufacturing of the Pressure Vessel

The APC-2/AS4 thermoplastic composite was chosen as the material system for the manufacture of composite pressure vessels for the SAUVIM underwater vehicle. This was due to superior mechanical properties, performance, and ease of fabrication of the APC-2/AS4 thermoplastic composite over the Graphite/Epoxy and Ti-6Al-4V, as explained earlier. The Cytec Fiberite, Inc produces the APC-2/AS4. This composite is available in form of unidirectional tape with various width and grade (ICI Thermoplastic Composite92). To manufacture the pressure vessel a symmetric sub-laminate configuration of $[90/90/0/0/90/90]_s$ was chosen. The pressure vessel was fabricated using in-situ thermoplastic composite filament winding/tape-laydown. Unidirectional tapes with thickness of 0.005" and width of 0.24" were used.

2.1. Manufacturing Equipment and Set-up

Figures MAF-56 and MAF-57 show the schematic of the in-situ thermoplastic composite filament winding/tape-laydown set-up for the scaled and main pressure vessels, respectively. The tensioner assembly consists of tape supply roller, tensioner rollers, sensing unit cable, air cylinder and festoon. The purpose of applying suitable tension in the tape during winding is to achieve a good compaction during fabrication, and increase the quality of final composite parts. Next, the tape is guided through the bracket system. A bracket system is designed to hold the pay-out eye system, compaction system, and nip-point heater. The tape passes through the pay-out eye system and winds over the mandrel. The nip-point infrared heater is located in such a way to melt the incoming composite tape as well as the surface of the substrate (Werdermann89) (see Figure MAF-56). For the main pressure vessel set-up, three local infrared heaters were used. Two of the local heaters melt the substrate and the third one melts the incoming tape and partially substrate (see Figure MAF-57). The bracket system is mounted on the translation stage, which gives the translational motion to the bracket system, and the mandrel, on one end, is connected to the rotary motor, which gives rotary motion (see Figures MAF-56 and MAF-57). A motion controller can be programmed to control the motion of the rotary and translational motor to yield a desired winding speed and path. The photographs of the filament winding set-up for the scaled and main pressure vessels are given in Figures MAF-58 and MAF-59.

The compaction system consists of an air cylinder, a pressure gage, and a compaction roller. The compaction roller is made of stainless steel with diameter of 3" and thickness of 0.9". The compaction roller can apply pressures up to 1,180 lb/linear-inch on the lay-down point to facilitate the bonding between the layers. The mandrel of the scaled pressure vessel is made of stainless steel and that for the main pressure vessel is made of aluminum. The mandrel for the scaled pressure vessel is 9" long with 4.12" outer diameter. The mandrel for the main pressure vessel is 36" long with an outer diameter of 12.92". The mandrels are tapered with a slope of 0.056 degrees to slip off the mandrel the wound thermoplastic composite part after its manufacturing. The mandrel is placed between two couplings, which are attached to a bearing at each end, and the whole system is connected to a rotary motor. The translational and rotational motions of the translation stage and rotary motor are controlled by a multi-dimensional motion programmable controller (Aerotech92). The program was written in G-code machine language to wind the tape on the mandrel. The winding speed for the scaled pressure vessel was 157 in/min. The scaled pressure vessel was manufactured in 8 hours. The winding speed of the main pressure vessel was 212 in/min. The time of manufacturing of the main pressure vessel is about 90 hours.

Two infrared strip heaters are set up at both sides of the mandrel to maintain the temperature of mandrel and composite shell around 50°C-70°C (see Figures MAF-56 and MAF-57) to facilitate the processing (Nejhad93). The temperature of the infrared heaters can be adjusted by power controllers. To monitor and control the temperature of the mandrel and composite shell a thermocouple and an infrared pyrometer were used.

2.2. Fabrication Methodology and Steps

The pressure vessel was manufactured using in-situ thermoplastic filament winding and tape placement technique. This method consists of three steps: (i) pre-heat the composite tape and mandrel/substrate, (ii) apply enough heat at the lay-down point during winding, and (iii) apply enough pressure on the lay-down point for consolidation purpose (Werdermann89; Aerotech92; Sonmez97). In the first step, the incoming pre-preg composite tape is preheated by passing through the preheating infrared heaters, and then is wound/laid-down on the mandrel (Nejhad97). Infrared heaters on sides of the mandrel also keep the temperature of the mandrel and composite substrate around 50°C-70°C and below the glass transition temperature to facilitate winding (Nejhad93; Sonmez97). While the tape is being wound/laid-down on the mandrel, the local nip-point infrared heater supplies heat to the mating surfaces of the incoming tape and the substrate. The temperature at the lay-down point should be around processing temperature in order to melt the matrix. The processing temperature for APC-2 matrix is around 450°C (ICI Thermoplastic Composite92). The processing temperature is controlled by the winding speed and heat intensity, which can, in turn, affect the size of the processing window (Nejhad91a, Nejhad91b). The larger the size of the processing window, the easier it is to control the processing and quality of the manufactured parts (Nejhad93). Finally, the compaction roller should apply enough pressure on the lay-down point to consolidate the incoming composite layer to the previous layer on the substrate to facilitate the consolidation procedure. Heat flux, tape/tow/substrate preheating, winding speed, and pressure are four most important parameters for in-situ manufacturing of thermoplastic composites. These parameters must be adjusted accurately to manufacture high quality structure and achieve high production rate by enlarging the processing window and operating within it.

To determine the processing condition for the manufacturing of the APC-2/AS4 pressure vessel, a case study was performed (Yousefpour99, Yousefpour00b). In this study, mechanical performances of APC-2/AS4 thermoplastic composite C-ring samples with different processing conditions were investigated and experimental results were compared with numerical results using Finite Element Method (FEM). The manufactured samples had final average inner radius of 2.13", thickness of 0.11" and width of 0.26". The effects of tape preheating, mandrel/substrate preheating, and on-line consolidation pressure on the mechanical performances of the parts were studied. Mandrel/substrate preheating was found to be necessary for good quality manufacturing. Ten sets of samples, with five samples per set, were manufactured using in-situ thermoplastic composite filament winding. For the first five sets, tape preheating below Glass Transition Temperature (T_g) at 110°C was used, however the consolidation pressure for various sets was 30, 70, 105, 140, and 175lb/linear-inch. Same pressures were used for the next five sets while the tape was preheated above T_g at 170°C. T_g for APC-2 is 140°C. C-ring tests were performed to evaluate failure stress, strain, and deflection of C-rings at room temperature. All C-ring tests were performed on an Instron machine. Samples failed in compression at inner radius. It was found that samples with tape preheating below T_g had superior mechanical performance than those with tape preheating above T_g . This could be due to twisting of filaments within the tape during preheating which led to the low quality of samples and possible immature failure. Also, it was found that samples made with consolidation pressures of 70 and 105

lb/linear-inch had better mechanical performances than other samples. It is believed that consolidation pressures of 70 and 105 lb/linear-inch were optimum consolidation pressures and caused good mechanical performances and quality. Scanning Electron Microscopy was conducted on the samples for quality control. The manufactured samples were found to be uniform microscopically. Non-linear Finite Element Analysis (FEA) associated with contact element was performed to simulate the C-ring testing and determine the mechanical performance of the C-rings. It was found that the results of FEA were in good agreement with the experimental and analytical results. The preheating and consolidation pressure affect the quality of the parts, which, in turn, affects the failure load and strength that can be measured experimentally and be used in the FEA. It can be concluded that using FEA in conjunction with failure load as input to model can present the real mechanical performance of the parts. These results were used to determine the optimal processing parameters for the manufacture of the APC-2/AS4 pressure vessels.

3. Testing of Scaled Pressure Vessel

An APC-2/AS4 scaled pressure vessel was designed for the hydrostatic pressure of 3,500 psi. The length and inner diameter of the scaled pressure vessel were chosen to be approximately one-third of the main pressure vessel for the SAUVIM. The length and inner diameter of the scaled pressure vessel were 6.625" and 4.18", respectively. The thickness of the scaled pressure vessel was fixed and chosen to be 0.24" in order to have a thick-walled pressure vessel. A symmetric sub-laminate configuration of $[90/90/0/0/90/90]_s$ was chosen for the composite scaled pressure vessel. The 2:1 hoop-to-axial ply ratio is chosen since the theoretical hoop stress is twice of axial stress (Hyer, 1988). Figure MAF-60 gives a photograph of the scaled pressure vessel with its stainless steel end-caps and tie-rods. The scaled pressure vessel was instrumented with eight strain gages. Figure MAF-61 shows the strain gage locations. Four strain gages were located close to the cylinder end to monitor the strain near the end closure. Two of those strain gages were positioned in the axial and the other two in the hoop direction. Other four strain gages were located at the mid-length of the pressure vessel with two strain gages in the axial and two in the hoop direction. The scaled pressure vessel was tested under hydrostatic pressure up to 3,500 psi, in the Hawaii Institute of Geophysics Laboratory Pressure Chamber with maximum pressure testing capability of 10,000 psi. After attaching the strain gages and wiring them, each wire was connected to a 16-pin connector, which was located on one of the scaled pressure vessel's end-cap (see Figure MAF-62). Another 16-pin connector was attached to the end-cap of the pressure chamber (see Figure MAF-63), which had wires out from the other side of the pressure chamber's end-cap. The wires were connected to a Signal Conditioning Board. The Signal Conditioning Board was connected to a Data Acquisition Board, which was installed on a PC. LabView software was used to collect the strain gage data. The two connectors were connected by a cable. The connectors and the cable were designed specifically for high-pressure applications. Figure MAF-64 shows the photograph of the scaled pressure vessel, the high-pressure chamber, and the connection cable. The scaled pressure vessel was placed in the pressure chamber and pressurized. The pressure was raised with an increment of 350 psi every 5 minutes with a dwell time of 10 minutes per step. After staying 30 minutes at the maximum pressure, the pressure was dropped to 1,750 psi and again increased to 3,500 psi with the same manner as before. After staying for 30 minutes at the maximum pressure, the pressure was released to the atmosphere pressure and the pressure vessel was taken out. At the end of the test, the pressure vessel was intact and no leak was observed. Figure MAF-65 gives the photograph of the scaled pressure vessel after the test.

Figures MAF-66 and MAF-67 show the comparison of the axial and hoop strain results, respectively, from the experiment and FEA for the strain gages close to the end-cap. There are good agreements between the experimental and numerical results. These figures show that the experimental strain results are slightly smaller than the FEA results. This reveals that the FEA results are accurate or somewhat conservative. The same conclusions can be made for the axial and hoop strain results from experiment and FEA at the mid-length of the scaled pressure vessel shown in Figures MAF-68 and MAF-69, respectively. The main pressure vessel will be tested in near future.

4. Design, Analysis, Manufacture, and Test of Shallow Water Composite Pressure Vessels Using E-glass/Epoxy Woven for SAUVIM

For the Phase-I of the SAUVIM project, the vehicle will be operated and tested in the shallow water (less than 300 feet under water). Six E-glass/Epoxy shallow water composite pressure vessels with internal length of 18" and inner diameter of 13" were designed for an external hydrostatic design pressure of 165 psi. Buckling and stress finite element analyses were performed for the design of the pressure vessels. An eigenvalue buckling analysis was performed to determine a bifurcation buckling pressure and a modal shape of the structure for a wall-thickness of 0.304 inches. These results were used to perform a non-linear buckling analysis. The buckling pressure was determined to be 215 psi. Stress analysis was performed to investigate the stress response of the structure with the wall-thickness of 0.304" under the design pressure. Maximum stress criterion was used and a stress factor of safety of 11.95 was achieved. End-caps were designed using Aluminum 6061-T6 employing Von Mises criterion. The end-caps have seven holes, which are used to place the connectors and vacuum bolt. The stress factor of safety of two was achieved for the end-caps (Ng00a; Yousefpour00).

Tube roll-wrapping with wet-laying technique was used to fabricate the pressure vessels. This technique consisted of several steps, namely set-up preparation, fabric impregnation, fabric rolling, shrink taping, curing/cooling, and post-processing (Ng00a, Ng00b). The total time of manufacturing was 7 hours for each pressure vessel. The final products needed minor machining. The final length, inner diameter, and thickness of the pressure vessels were 19.5", 13", and 0.324", respectively. Aluminum 6061-T6 end-caps with 1.125" thickness were fabricated. An axial washer and two radial O-rings were used to seal the pressure vessel/end-caps interfaces. The pressure vessels and end-caps were assembled using six tie-rods. Six pressure vessels were tested at the design pressure of 165 psi inside a high-pressure water-filled chamber. The pressure vessels were intact and no leakage was observed. Figure MAF-70 shows a shallow water pressure vessel with its end-caps and tie rods (Ng00a).

5. Finite Element Analysis of the Frame

Two SAUVIM groups, namely, Mechanical-Electrical Design (MED) and Mechanical Analysis and Fabrication (MAF) groups were involved in the design, analysis, and test of the frame for SAUVIM project. MED group was responsible to identify the required components and develop the preliminary conceptual design of the frame based on the most desired component layout as well as the fabrication considerations of the frame. Several frame designs were proposed at this stage. Material selection and FEA of the frame were MAF group responsibilities. MAF group performed FEA on the frame structure and studied the behavior of the frame structure under different loading conditions. MAF group recommended the necessary changes in the design of the frame to the MED group according to the FEA study. The MED group implemented the changes and came up with a modified design for the frame. This sequence was repeated till the optimum design was achieved. The MAF group presented the final FEA results and determined the final size and shape of all required structural members of the

frame to the MED group. Then, the MED group sent the drawing of the frame with all required information to a machine shop for fabrication. Finally, MED group performed a number of non-destructive tests such as X-ray radiography and penetration tests to evaluate the quality and strength of the critical locations such as welded joints.

5.1. Mechanical Design of the Frame

The frame of an underwater vehicle serves as a carriage on which thrusters, batteries, pressure vessels, robotic arms, foams, and accessories are firmly attached. Therefore, all loads on the vehicle are carried by the frame. Three types of loads are applied to the frame - static loads (e.g., weight of the components and frame itself), bouncy forces, and dynamic loads (e.g., the thruster forces). To design the frame for SAUVIM, several steps were taken. The following gives these steps:

- Identification of the components and component layout for the vehicle (see Figure MAF-71).
- Development of a conceptual design for the frame using component layout.
- Selection of material.
- Analysis of the frame structure using Finite Element Method.
- Fabrication of the frame.

5.2. Components Identification

The MED group identified all the required components that would be installed on the frame. Table MAF-16 shows a list of major components for the vehicle. After identifying the components, a few component layouts were proposed. The component layout presents the locations of the components on the frame and is a benchmark for the design of the frame. A few issues were considered during the component layout design. First, it was desired to place components such that the applied loads on the frame would be distributed on the structure uniformly. Second, the components, such as pressure vessels and batteries, which needed to be accessible, were located at reachable locations for possible services or exchanging.

5.3. Conceptual Design of the Frame

The MED group, based on their desired component layout, proposed a preliminary design of the frame. In the component layout, locations, weights, and volumes of the components were specified. Figure MAF-72 shows a preliminary design of the frame based on the desired component layout. Figure MAF-73 shows the final frame design. The following issues were considered during the final frame design:

- Use of unnecessary members in the structure was avoided. This would otherwise increase the weight of the vehicle, which is not desirable.
- Creation of unnecessary complicated joints in the structure was avoided. This could otherwise increase the fabrication cost of the frame.
- Drilling any holes in the loaded members was avoided. This could otherwise decrease the strength of the loaded members around the holes, leading to a possible premature failure.
- Creation of any air pockets inside/between members was avoided. Any air pocket could otherwise act as a pressure vessel and might cause immature failure of the structure, especially for deep ocean applications.

- Attachments were designed to install the components onto the frame. The attachments were welded to the frame and components will be bolted or riveted to the attachments.

5.4. Candidate Materials/Materials Selection

Aluminum, steel, and titanium were three candidate materials for the frame.

Table MAF-17 gives the mechanical properties of the candidate materials.

To select the appropriate material, the following issues were considered (Ashby80):

- Price and availability of the material. The material has to be cost effective and available in the market. It is economical to design the members of the frame using standards or stock sizes.
- Strength, stiffness, and density of the material. The material properties of the frame depend on the choice, heat treatment, and processing of the material. Choosing a proper material can reduce the weight and increase the strength of the underwater vehicle structure.
- Weldment strength of the material. Welding is structurally sound joining technique for fabrication of the frame. It is especially important to know the strength of the welded region for the selected material. For example, the strength of the Aluminum reduces approximately 30-40% around the welded region (Pickering97).
- Corrosion. The material to be selected for the frame should have a good corrosion resistance since the frame operates in a highly corrosive environment, i.e., seawater. The underwater frame can be subjected to different types of corrosions, e.g., crevice corrosion, galvanic corrosion, pitting corrosion, stress corrosion, cracking corrosion, etc. (Jones92).

Aluminum 6061-T6 was the selected material for the frame for the following reasons. This material is available in the market in different sizes and is less expensive than steel and titanium. Fabrication cost of the aluminum frame is much less than titanium frame. Aluminum is lighter than titanium and steel. It has relatively good corrosion resistance. However, after each deployment, vehicle has to be washed with tap water. The main weakness of this material is its low-weldment strength.

5.5. Finite Element Analysis (FEA)

Finite element analysis is a powerful tool for the frame design because it involves all the factors influencing the behavior of the frame, such as geometry, material properties, loads, and boundary conditions (Huebner95). Thus, it leads to an excellent design in terms of strength, stiffness, and economic efficiency. The most efficient method to analyze the frame using finite element method is using 3-D space frame employing 1-D beam element with appropriate cross-sections and material properties. This method saves tremendous computational time and memory requirements, and gives sufficient information for the reliable design of the frame. Finite element analysis was performed to analyze the frame using I-DEAS finite element software. In the pre-processing step, two noded beam elements with different cross-sectional dimensions were used. The material properties of aluminum were given to the software (Table MAF-17). The model of the frame was created according to the final design (see Figure MAF-74).

In the solution step, the load and boundary conditions were applied to the model and the solution was initiated. As mentioned before, the frame of the underwater vehicle carries the load of the vehicle components. The magnitudes of the loads have to be known by the designer. These loads can be static loads, which act as concentrated or distributed loads, buoyancy forces, which appear when the vehicle is placed in water, and dynamic loads such as thruster loads. Close interaction exists between the applied loads, the frame, and its supports. The loads on the frame produce stresses on the supports, which lead to the deformation of the members. Therefore, locations and the number of the supports can dictate the stress distributions on the frame and play critical roles for the serviceability and survivability of the frame. Four load and boundary condition cases were considered. Each case had two sub-cases, i.e., a) retracted robotic arms and b) extended robotic arms. First, it was assumed that the vehicle was supported on the ship or ground (dry condition). The major loads on the frame were the weight of the frame and the weight of the components, which were attached to the frame. These are all static loads. The best locations for the supports were found to be under each column of the frame (see Figure MAF-75). Second, it was assumed that the vehicle was lifted for deployment purpose. The loading conditions (lifting condition) were the same as previous case but the supports were located at the lifting points (see Figure MAF-76). As the vehicle was placed in the water and released from-lifting mechanism, the loads and boundary conditions of the frame would change (wet condition). In this case, the loads applied on the vehicle were a combination of the weight of the frame and components as well as the buoyancy forces due to foams and buoy equipment. The net weight of the vehicle is zero or slightly greater than zero. The frame itself was under two equal but opposite loads, i.e., weights of the vehicle and buoyancy forces. The frame in reality was not supported. To avoid rigid body motions, it was assumed that the frame was supported at the four corners (see Figure MAF-77). The load conditions for the frame during operation were the same as previous case, but the maximum thruster forces were added to the load conditions (i.e., wet/dynamic condition).

The stress results were studied to modify the design by reinforcing the weak regions of the structure, eliminating unnecessary members, and changing the size and type of the beam members. It should be noted that the stress factor of safety was calculated based on the weldment strength of the aluminum (i.e., 12,000 psi). Due to the existence of different loads and boundary conditions, the frame was designed in such a way that it was sustained under different load and boundary conditions. The results of maximum stresses and displacements, and factor of safety of the frame are given in Table MAF-18. The minimum factor of safety occurred for wet/dynamic analysis with arms at rest position and was found to be 4.41. It should also be mentioned that to roughly a count for possible dynamic loading of the frame during lifting and deployment all loads on the frame were doubled prior to the analysis.

5.6. Fabrication of the Frame

The final step was the fabrication of the frame. All the members were drawn with all specifications and sent for fabrication by MED group. The designed frame was fabricated at the Hawaii Shipyard Machine Shop in Honolulu, HI. X-ray Radiography and penetration tests were performed on the welded regions to verify the quality of the weldment area by MED group.

6. Procedures for the Design, Manufacture, and Assembly of Composite Flooded Fairing

A composite flooded fairing is designed for the SAUVIM vehicle to reduce the drag force during traveling in water and protect the internal equipment in case of a possible collision or impact. The design of the fairing is at its beginning stage. Figure MAF-78 shows the initial design of the fairing.

The SAUVIM vehicle would travel on an average speed of 3 knots. The design of the fairing is peculiar, as the frontal part of the fairing should be made sacrificial. This is because under catastrophic crush situation, the frame and its containment should not be damaged at all. The frontal part should be able to sustain the force and absorb the energy during the impact, and be designed with the concept of disposable. The material system used for the fairing, supports, and joints should be indifferent to seawater environment with the capability to resist corrosion. It should be built using light-weighted materials in order to save energy and improve the ease of maneuvering. To manufacture the fairing, first, a foam model would be built according to the size and shape of the suggested fairing. Next, two sections of the female molds (left and right) would be built on top of the foam using wet composite lay-up and vacuum bagging method. Next, the woven composite male molds (i.e., the fairing) would be built on the inner surface of the female molds using similar technique. To keep the outer surface of the male mold (i.e., the fairing) smooth, the composite fabrics have to be wet-laid in the female molds since the inner surface of the female mold is very smooth. After the composite male molds (i.e., the fairing) are built, it is then cut into several desired segments. The joining mechanism between fairing segments will utilize a special locking mechanism.

Composite is chosen as the material system for the fairing and the joints (i.e. mounting points on the fairing for the supports on the frame). In comparison with metals such as steel or aluminum, polymeric composite materials have high strength and stiffness to weight ratio. They have good corrosion resistance, with low manufacturing costs, especially when the fairing is not of a straightforward shape. Graphite and Kevlar are chosen as the fiber materials in the form of the plain weave hybrid fabric for the fairing. Graphite is known as a material of lightweight, high strength and stiffness. Meanwhile, Kevlar is known for its toughness and good abrasion and impact resistance characteristics. For the matrix material, epoxy resin is chosen. The attachments between the fairing and the frame would utilize aluminum supports on the frame and Graphite/Epoxy plain weave woven composite wet-laid joints on the fairing. The reason for choosing aluminum as the material for the supporting bars as well as for the frame is because it is inexpensive, easy to modify, light, and corrosion resistant. The joint materials were chosen as woven Graphite composite since Kevlar is relatively weak under compression, and the joints are under bending stresses and not a direct impact. The manufacturing technique would also be different for the joints. They will be molded using the wet lay-up method. The 3-D effective orthotropic homogeneous composite properties of the Graphite-Kevlar/Epoxy hybrid woven as well as the Graphite/Epoxy plain weave plies are obtained given the material properties of the individual materials (see Table MAF-19) using 3-D Crimp model and (0/90) Cross-ply model (Ng00a). For the Graphite-Kevlar/Epoxy hybrid woven ply, two sets of lamina materials are needed to determine the 3-D effective material properties. The properties of each lamina are listed in Table MAF-19. The 3-D effective orthotropic properties of Graphite-Kevlar/Epoxy hybrid and Graphite/Epoxy plain-weave woven plies are obtained using 3-D Crimp model (Ng, et al., 2000a) and are given in Table MAF-20.

For the initial design, the thickness is determined as 0.2414 mm for each woven ply, and the composite fairing stacking sequence is proposed to be multiples of $[0_G/90_K/90_G/0_K/0_K/90_G/90_K/0_G]_n$. This orientation is prospective, yet meet all the criteria such as symmetric and balanced even for a satin weave. It should be mentioned that a $(0_G/90_K)$ represents a Graphite-Kevlar/Epoxy hybrid woven ply. The effective properties of Graphite-Kevlar/Epoxy plain weave material given in Table MAF-19 were used for this stacking sequence to obtain the overall effective properties of this static sequence to be used in the LS-DYNA crash simulation. The fairing is comprised of eight composite segments (see Figure MAF-79). The front and back is comprised of 3 pieces each, i.e., 1 piece on top and 2 pieces on the bottom. There will be 2 composite pieces covering the middle part of the submersible, as one on the left-hand side and the other on the right hand side. The fairing segments in

Figure MAF-79 are numbered in the order of assembly. Based on this fairing configuration, the locations of joints and supports can be determined. As a preliminary decision, there will be 40 support bars and 40 joints equally distributed and held tight between the frame and the fairing (see Figure MAF-80). Figure MAF-80 also gives the order and assembly process of the fairing segments on the frame.

6.1. Utilization of DYNA-3D for Composite Flooded Fairing Crash Simulations

DYNA3D, an explicit transient three-dimensional non-linear finite element analysis package, has been utilized in the crash simulation of composite structures (Nejhad97). DYNA3D is used for analyzing large deformation dynamic response of inelastic solids and structures. A contact impact algorithm permits gaps and sliding with friction along material interfaces. Spatial discretization of the model is achieved by the use of 8-node brick or 4-node tetrahedral solid elements, 8-node solid shell elements, 4-node shell elements, 2-node beam elements, truss elements, membrane elements, and rigid bodies and discrete elements. The equations of motion are integrated in time by the central difference method.

For aluminum, MAT 24 was chosen to be the material of frame for the crash simulation. According to LS-DYNA (1999) keyword user's manual, it is an elasto-plastic material with an arbitrary stress versus strain curve, and arbitrary strain rate dependency can also be defined. Also, failure based on a plastic strain or a minimum time step size can be defined. For this material type, the values of the mass density for the material, Young's modulus, Poisson's ratio, Yield strength, and tangent modulus should be defined.

For the Carbon-Kevlar/Epoxy hybrid plain-weave woven composite, MAT 54 was chosen to be the material of fairing for the crash simulation. This model considers failure under compression. Chang-Chang (LS-DYNA99) failure theory is used during the crash simulation. Thus, it is only valid for thin shell elements. For this material type, the mass density of the material, Young's modulus in x, y and z direction, Poisson's ratio ν_{yx} , ν_{zx} , ν_{zy} , shear modulus G_{xy} , G_{yz} , G_{zx} should be defined. Also, the tensile and compressive strengths in both longitudinal and transverse directions are to be defined. The in-plane shear strength should also be defined. According to the ICI publication, the tensile compressive, and shear strengths for Carbon/Epoxy plain weave composite are 75ksi, 65 ksi, and 7.7 ksi respectively. For Kevlar/Epoxy plain weave composite, the tensile, compressive, and shear strengths are 60 ksi, 25 ksi, and 5 ksi respectively. Those values are the more conservative one. Thus, based on those values, the tensile, compressive, and shear strengths for Carbon-Kevlar/Epoxy hybrid plain weave are 67.5 ksi, 45 ksi, and 6.35 ksi respectively given the stacking sequence of [0/90/90/0] using Rule of Mixture. For the Carbon/Epoxy plain weave composite, the tensile, compressive and shear strengths are 75 ksi, 65 ksi and 11 ksi respectively. The failure criterion was based on maximum principal stress criterion to obtain a factor of safety.

For the Carbon/Epoxy plain-weave composite, MAT 54 was chosen to be the material of fairing for the crash simulation. The same parameters apply to this model as mentioned before. The failure criterion was also based on maximum principal stress criterion to obtain a factor of safety.

The majority of the computational time for DYNA3D crash simulation is contributed to the momentum transfer that occurs with impacting bodies. For impact, a master and a slave surface are defined for the contact interface. They are made up of a list of nodes and element faces. With the contact type set to 13, the program automatically checks the nodes and elements that would participate in the simulation. It checks with complex algorithm, for penetration through the contact interface

between the master and slave surfaces. Forces are placed at the nodes of the master and slave surfaces for momentum transfer upon penetration. After the contact interface process, the decelerations are updated and the kinematics boundary conditions are applied. Then, the velocities are updated and the finite element process loops back. The iteration speed is based on the time step, which in turn is based on the length of the smallest element in the whole model. During the initiation of the crash simulation, it will warn the user for the necessary time step size.

In this research, the frontal part of the fairing is meant to be sacrificial and replaceable. This is because the important equipment such as batteries, pressure vessels that house the electronic equipment and control systems are held within the frame. Therefore, the fairing has to be able to sustain the impact in the event of a crash, yet be able to absorb all the impact energies before the crash has any effects on the frame. In other words, the frame should be unharmed when the whole submersible attains the overall velocities of 0 m/s in the event of a crash. The submersible is considered as crashworthy if it fails in a controlled manner, has the ability to absorb the kinetic energy of the crash, and is able to maintain some “survival” space around the frame. Nevertheless, the main drawback of utilizing composites in the structural members of a vehicle is their inherent brittleness. Metals such as steel and aluminum are able to absorb high amounts of energy by plastic deformation. Metals can undergo extensive slip without nucleation and propagation of cracks so that they are able to endure high strains before failure occurs. Unlike metals, composites fail in a brittle mode with no plastic deformation. Composites such as carbon can only endure strains between 1 to 3 percent before failing in a brittle mode (Hull83). As a result, Kevlar fibers are used to make up for the brittleness of carbon fibers, and in return, carbon fibers are used to make up for the low compressive properties of the Kevlar fibers. Therefore, the choice of Carbon-Kevlar/Epoxy hybrid woven composite is optimum given its mechanical and environmental properties compared with metals. Since the duration of the SAUVIM does not exceed eight hours the moisture absorption of the composite fairing and joint are less than 0.2% (Lundgren99; Shen81), and hence are negligible here.

Future Tasks (Phase II Tasks)

- Test the main thermoplastic composite pressure vessel and compare the strain results with analysis (FEA).
- Manufacture and test five more deep ocean thermoplastic composite pressure vessels.
- Finalized the design and analysis of the composite flooded fairing with its joints.
- Manufacture and test the composite flooded faring with its joints.

Table MAF-1. Mechanical Properties of Ti-6Al-4V

Property	Value	Property	Value
E (Msi)	16.5	S_y (Ksi)	120
G (Msi)	6.15	a (10⁻⁶/°F)	4.80
n	0.34	r (lb/in³)	0.16

Table MAF-2. Unidirectional Mechanical Properties of the APC-2/AS4 and Graphite/Epoxy

Property	APC-2/AS4	Graphite/Epoxy	Property	APC-2/AS4	Graphite/Epoxy
E₁ (Msi)	20.0	20.6	E₂ (Msi)	1.48	1.50
G₁₂ (Msi)	0.82	1.04	n₁₂	0.28	0.27
a₁₁(10⁻⁶/°F)	-0.1	-0.50	a₂₂(10⁻⁶/°F)	13.3	15.0
b₁₁	0	0.01	b₂₂	0.3	0.2
S_{11T} (Ksi)	300	210	S_{11C} (Ksi)	175	170
S_{22T} (Ksi)	12.5	9.0	S_{22C} (Ksi)	28.4	29.0
S₁₂ (Ksi)	27.3	8.7	r (lb/in³)	0.057	0.057

Table MAF-3. Modified Coefficient of Thermal Expansion (MCTE) at 140°F and 32°F for Composite Systems

MCTE	140°F		32°F	
	APC-2/AS4	Graphite/Epoxy	APC-2/AS4	Graphite/Epoxy
a₁₁(1/°F)	-1.00E-07	6.64E-06	-1.00E-07	-1.37E-05
a₂₂(1/°F)	2.32E-05	1.58E-04	-4.86E-04	-2.48E-04

Table MAF-4. Thickness and ‘d’ Values for the Pressure Vessels

	Thickness (in)	d (in)
Ti-6Al-4V	1.35	0.0134
APC-2/AS4	1.68	0.0214
Graphite/Epoxy	1.68	0.0208

Table MAF-5. Radius of the End-cap Circular Taper, R, for Various Percentage of ‘d’ for Candidate Materials

d	d=0.0134" R_{Ti-6Al-4V} (in)	d=0.0214" R_{APC-2/AS4} (in)	d=0.0208" R_{Graphite/Epoxy} (in)
100%	84	53	54
90%	93	58	60
80%	105	66	68
70%	120	75	77
60%	140	88	90
50%	168	105	108
40%	210	131	135
30%	280	175	180
20%	420	263	270
10%	841	526	541
0%	∞	∞	∞

Table MAF-6. Maximum Stress, Strain, and Deflection Results of the Ti-6Al-4V Pressure Vessel

Temperature	Stress (psi)	Strain (in/in)	Deflection (in)	Factor of Safety
32°F	64,742	0.0053	0.024	1.85
70°F	61,292	0.0050	0.022	1.96
140°F	58,149	0.0047	0.020	2.06

Table MAF-7. Maximum Stress, Strain, and Displacement Results of the APC-2/AS4 and Graphite/Epoxy Pressure Vessels

	APC-2/AS4			Graphite/Epoxy		
	32°F	70°F	140°F	32°F	70°F	140°F
Radial Stress (psi)	-9,722	-9,723	-10,697	-13,130	-9,723	-14,289
Axial Stress (psi)	-11,950	-11,884	-13,078	-21,370	-11,903	-24,018
Hoop Stress (psi)	-75,406	-75,171	-77,302	-93,391	-77,067	-96,549
Radial Strain (in/in)	-0.0059	-0.0059	-0.0058	-0.0047	-0.0060	-0.0053
Axial Strain (in/in)	-0.0044	-0.0042	-0.0057	-0.0119	-0.0041	-0.0134
Hoop Strain (in/in)	-0.0037	-0.0037	-0.0037	-0.0046	-0.0037	-0.0048
Displacement (Axial) (in)	0.037	0.038	0.035	0.027	0.037	0.028
Factor of Safety	1.97	1.96	2.03	1.30	1.90	1.15

Table MAF-8. Bifurcation Pressure of the Pressure Vessels

Material	Bifurcation Buckling Pressure (psi)	Wall Thickness (in)
Ti-6Al-4V	100,422	1.35
APC-2/AS4	44,740	1.68
Graphite/Epoxy	47,426	1.68

Table MAF-9. Non-Linear Maximum Stress, Strain, and Displacement Results of the Ti-6Al-4V Pressure Vessel

Temperature	Max. Stress (psi)	Max. Strain (in/in)	Max. Deflection (in)	Factor of Safety
32°F	64,246	0.0052	0.023	1.87
70°F	60,846	0.0049	0.022	1.97
140°F	57,951	0.0047	0.020	2.07

Table MAF-10. Non-Linear Maximum Stress, Strain, and Displacement Results of the APC-2/AS4 and Graphite/Epoxy Pressure Vessels

	APC-2/AS4			Graphite/Epoxy		
	32°F	70°F	140°F	32°F	70°F	140°F
Radial Stress (psi)	-9,727	-9,728	-10,652	-13,187	-9,729	-14,342
Axial Stress (psi)	-11,978	-11,869	-13,126	-21,366	-11,930	-24,013
Hoop Stress (psi)	-75,541	-75,306	-77,444	-93,543	-77,198	-96,712
Radial Strain (in/in)	-0.0059	-0.0059	-0.0059	-0.0048	-0.0060	-0.0054
Axial Strain (in/in)	-0.0044	-0.0042	-0.0057	-0.0119	-0.0041	-0.0134
Hoop Strain (in/in)	-0.0037	-0.0037	-0.0037	-0.0046	-0.0038	-0.0048
Displacement (Axial) (in)	0.037	0.038	0.035	0.027	0.037	0.028
Factor of Safety	1.96	1.95	2.03	1.29	1.89	1.15

Table MAF-11. Summary of the Analysis Results

Material	Ti-6Al-4V	APC-2/AS4	Graphite/Epoxy
Stress Factor of Safety	1.96	1.96	1.90
Minimum Buckling Factor of Safety	3.60	3.60	3.60
Wall Thickness (in)	1.35	1.68	1.68
Density (lb/in³)	0.16	0.056	0.056
Weight (lbs)	204	91	91

Table MAF-12. Radius Circular Taper and Factor of Safety of SAUVIM Pressure Vessels

d	APC-2/AS4-19" d=0.024 (in)		APC-2/AS4-21" d=0.024 (in)	
	R (in)	Factor of Safety	R (in)	Factor of Safety
100%	46	0.78	46	0.78
90%	51	1.08	51	1.08
80%	58	1.57	58	1.57
70%	66	1.61	66	1.61
60%	77	1.64	77	1.64
50%	92	1.66	92	1.66
40%	115	1.67	115	1.67
30%	153	1.64	153	1.64
20%	230	1.45	230	1.45
10%	460	1.18	460	1.18
0%	∞	1.00	∞	1.00

Table MAF-13. Maximum Stress, Strain, and Displacement Results of the APC-2/AS4 Pressure Vessels with Optimum Tapered Radius for SAUVIM

	APC-2/AS4-19"			APC-2/AS4-21"		
	32°F	70°F	140°F	32°F	70°F	140°F
Radial Stress (psi)	-9,887	-9,655	-11,237	-9,887	-9,655	-11,237
Axial Stress (psi)	-12,276	-12,139	-13,710	-12,276	-12,139	-13,711
Hoop Stress (psi)	-88,735	-88,565	-90,112	-88,735	-88,565	-90,112
Radial Strain (in/in)	-0.0050	-0.0051	-0.0048	-0.0050	-0.0051	-0.0048
Axial Strain (in/in)	-0.0052	-0.0050	-0.0065	-0.0052	-0.0050	-0.0065
Hoop Strain (in/in)	-0.0043	-0.0043	-0.0044	-0.0043	-0.0043	-0.0044
Displacement (Axial) (in)	0.038	0.039	0.036	0.042	0.042	0.040
Factor of Safety	1.67	1.67	1.72	1.67	1.67	1.73

Table MAF-14. Bifurcation Pressure of the Pressure Vessels for SAUVIM

Material	Bifurcation Buckling Pressure (psi)	Wall Thickness (in)
APC-2/AS4-21"	32,911	1.188
APC-2/AS4-19"	34,504	1.188

Table MAF-15. Non-Linear Maximum Stress, Strain, and Displacement Results of the 19" and 21" APC-2/AS4 with Optimum Tapered Radius Pressure Vessels

	APC-2/AS4-19"			APC-2/AS4-21"		
	32°F	70°F	140°F	32°F	70°F	140°F
Radial Stress (psi)	-9,883	-9,648	-11,233	-9,883	-9,649	-11,233
Axial Stress (psi)	-12,370	-12,235	-13,754	-12,371	-12,235	-13,754
Hoop Stress (psi)	-89,001	-88,830	-90,386	-89,004	-88,834	-90,390
Radial Strain (in/in)	-0.0050	-0.0051	-0.0048	-0.0050	-0.0051	-0.0048
Axial Strain (in/in)	-0.0052	-0.0050	-0.0065	-0.0052	-0.0050	-0.0065
Hoop Strain (in/in)	-0.0043	-0.0043	-0.0044	-0.0043	-0.0043	-0.0044
Displacement (Axial) (in)	0.038	0.039	0.036	-0.042	0.042	0.040
Factor of Safety	1.67	1.66	1.72	1.67	1.66	1.72

Table MAF-16. List of Major Components of the SAUVIM Vehicle with various Components Estimated Weight

Componets	Weight (lbs)
Cameras	109
Arms (two)	216
Batteries	1,459
Ballasts	1,260
Pressure Vessels	1,742
Thrusters	219
Fins	125
Foam	6,140
Fairing	929
Frame	1,200
Total Weigth	13,400

Table MAF-17. Mechanical Properties of the Candidate Materials for the Frame

	Tensile Strength (Ksi)	Yield Strength (Ksi)	Young's Modulus (Msi)	Density (lb/in³)
Aluminum	35-45	31-40	10	0.10
Steel	56-72	41-55	30	0.26
Titanium	130-144	120-134	16.5	0.16

Table MAF-18. Results of Maximum Stresses, Displacements, and Factor of Safety for the Frame

	Arms at the Extended Position			Arms at the Rest Position		
	$\delta_{max}(in)$	$\sigma_{max}(psi)$	Factor of Safety	$\delta_{max}(in)$	$\sigma_{max}(psi)$	Factor of Safety
Dry Analysis	0.007	1,270	9.45	0.007	1,840	6.52
Lifting Analysis	0.022	2,350	5.11	0.020	2,350	5.11
Wet Analysis	0.017	1,200	10.00	0.017	1,610	7.45
Wet/Dynamic Analysis	0.021	2,240	5.36	0.020	2,720	4.41

Note: The factor of safety is calculated based on the weldment strength of Aluminum (i.e., 12,000 psi)

Table MAF-19. Unidirectional Lamina Properties of Kevlar-49/Epoxy and Graphite/Epoxy (Mallick93)

Property	Kevlar-49/Epoxy	Graphite/Epoxy
E ₁ (Pa)	7.59E+10	1.448E+11
E ₂ (Pa)	5.52E+09	9.655E+09
E ₃ (Pa)	5.52E+09	9.655E+09
G ₁₂ (Pa)	2.07E+09	5.862E+09
G ₁₃ (Pa)	2.07E+09	5.862E+09
G ₂₃ (Pa)	1.54E+09	3.462E+09
v ₁₂	3.40E-01	2.50E-01
v ₁₃	3.40E-01	2.50E-01
v ₂₃	4.71E-01	4.07E-01
Density (g/m ³)	1.38E+06	1.58E+06

Table MAF-20. 3-D Effective Orthotropic Properties of Graphite-Kevlar/Epoxy Hybrid and Graphite/Epoxy Plain-Weave Woven Plies using 3-D Crimp Model (Ng00a)

3-D Effective Properties from Crimp Model	Graphite/Epoxy in Weft direction	Kevlar/Epoxy in Weft direction	Graphite-Kevlar/Epoxy lamina (Graphite in x-direction)	Graphite/Epoxy lamina
Ex (GPa)	27.16	18.74	27.16	31.22
Ey (GPa)	--	--	18.74	31.22
Ez (GPa)	7.82	6.98	7.40	9.29
vxy	0.02	0.04	0.02	0.02
vxz	0.44	0.53	0.44	0.45
vyz	--	--	0.53	0.45
Gxy (GPa)	3.37	2.77	3.07	5.08
Gxz (GPa)	3.78	2.48	3.78	4.46
Gvz (GPa)	--	--	2.48	4.46

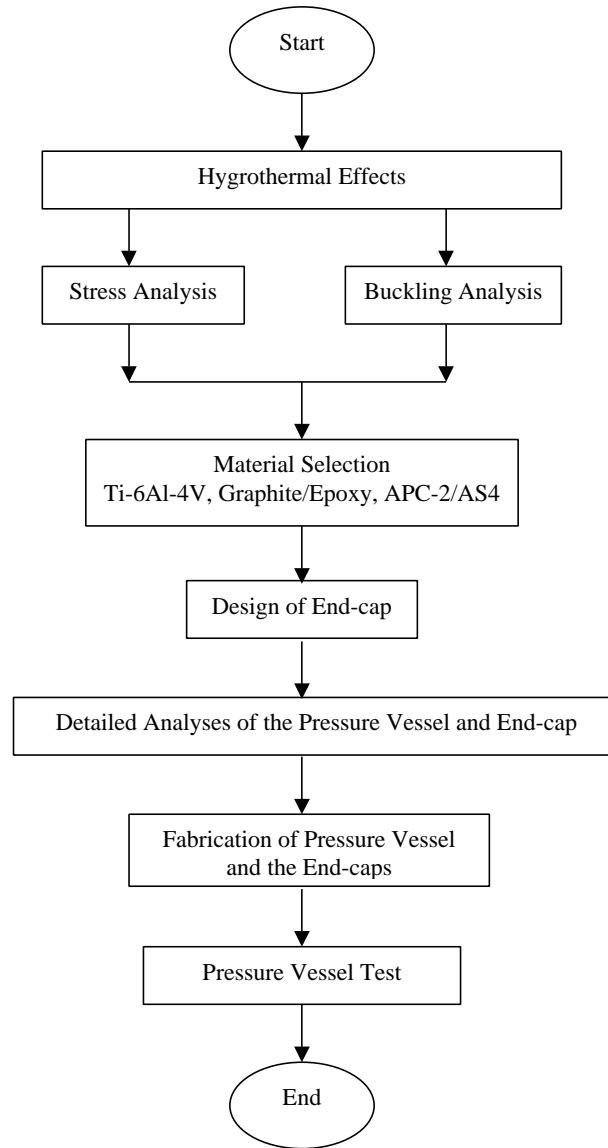


Figure MAF-1. Development Methodology of the Pressure Vessels

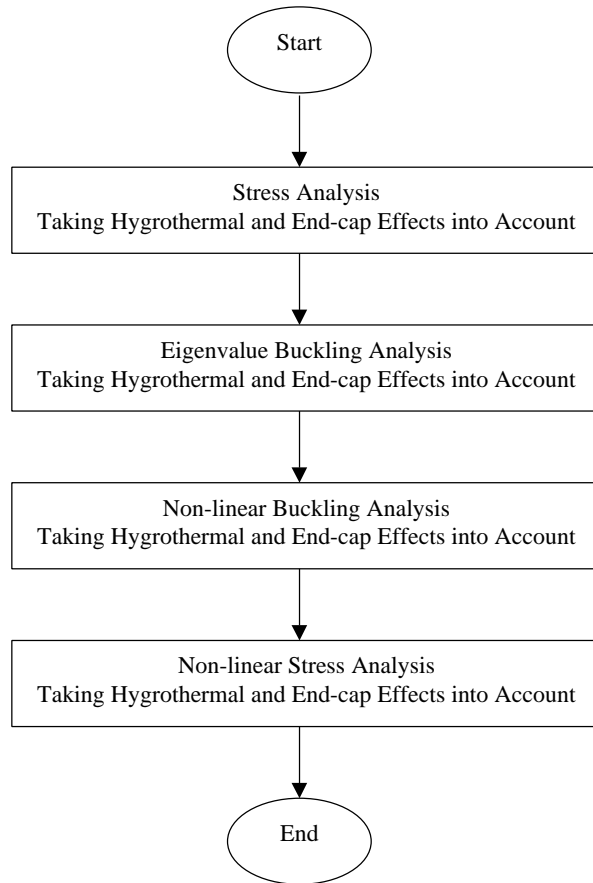


Figure MAF-2. Analysis Procedure for the Pressure Vessels

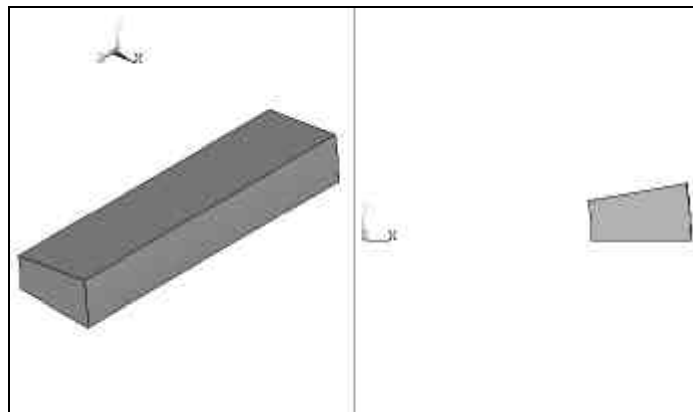


Figure MAF-3. Model of Ten Degrees Wedge of the Pressure Vessel in Stress Analysis

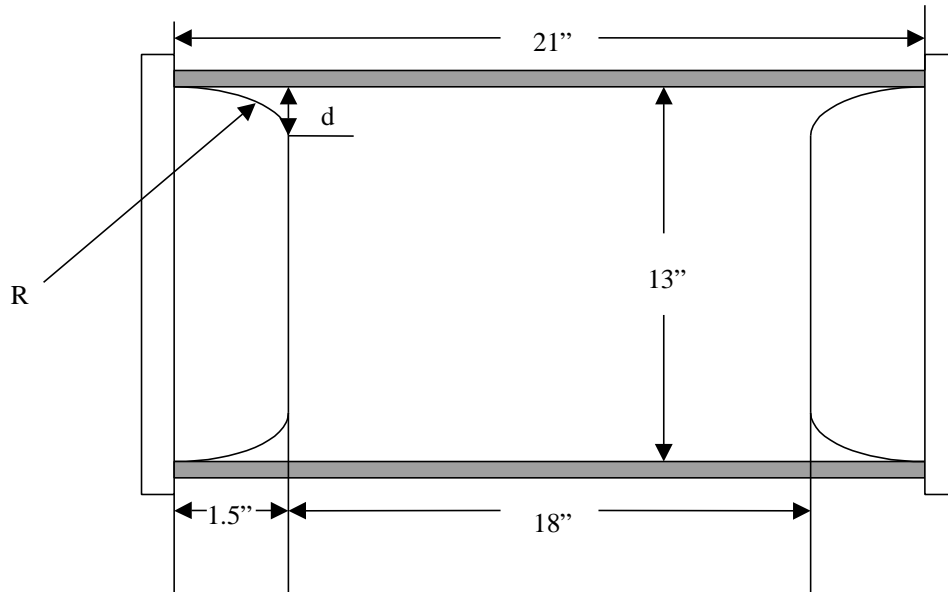


Figure MAF-4. Schematic of the Pressure Vessel with the Contoured-End Plug-supported End-Cap

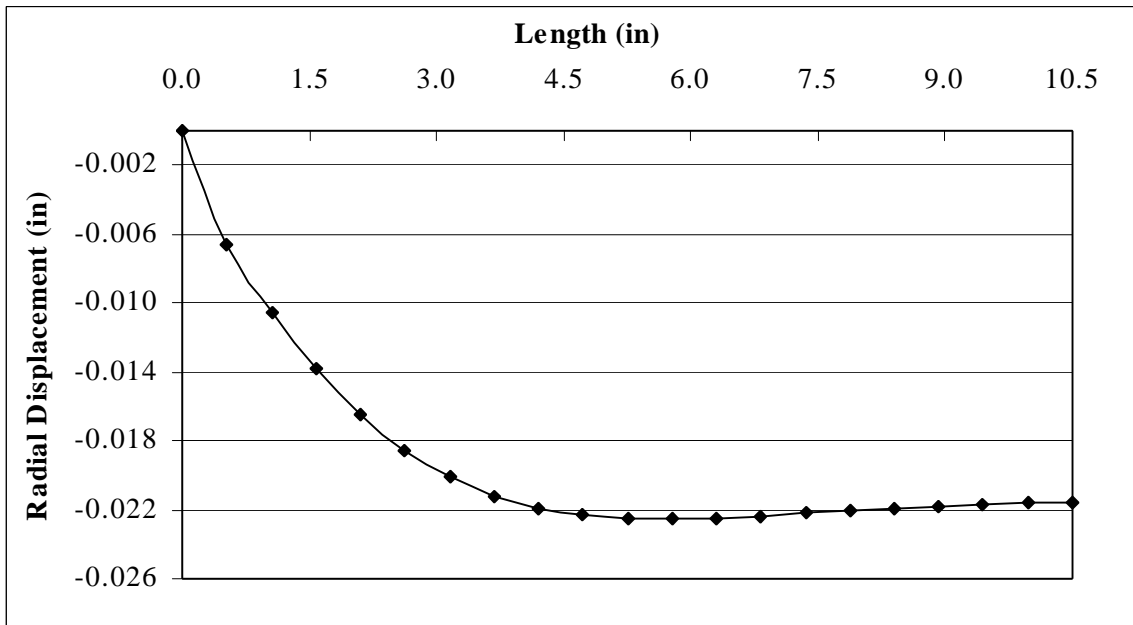


Figure MAF-5. Radial Displacement from the Cylinder End to the Mid-length of the Ti-6Al-4V Pressure Vessel

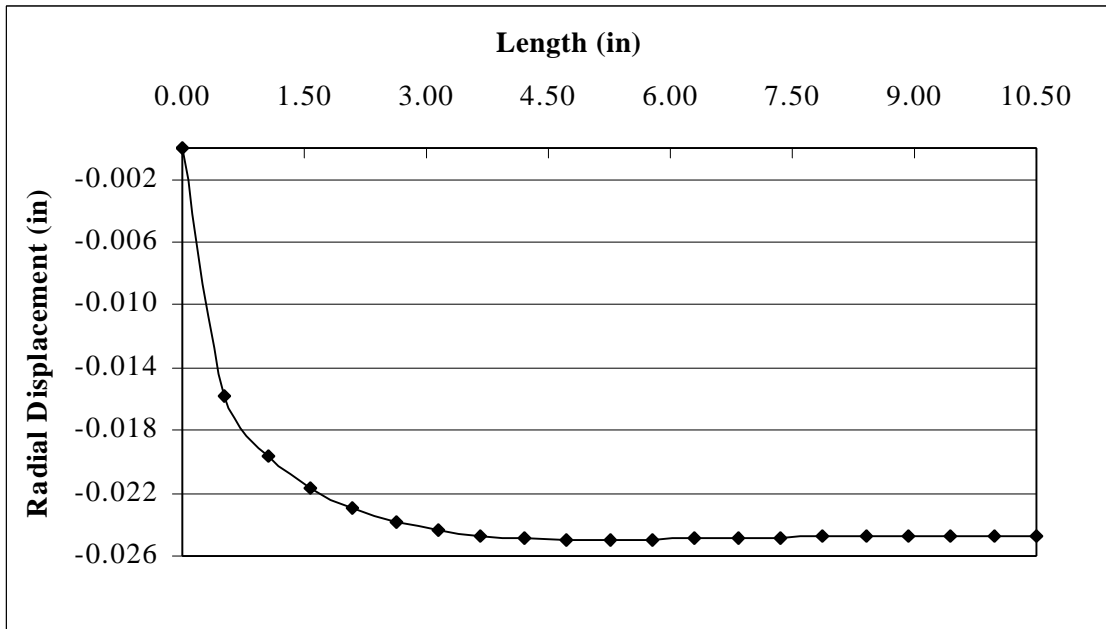


Figure MAF-6. Radial Displacement from the Cylinder End to the Mid-length of the APC-2/AS4 Pressure Vessel

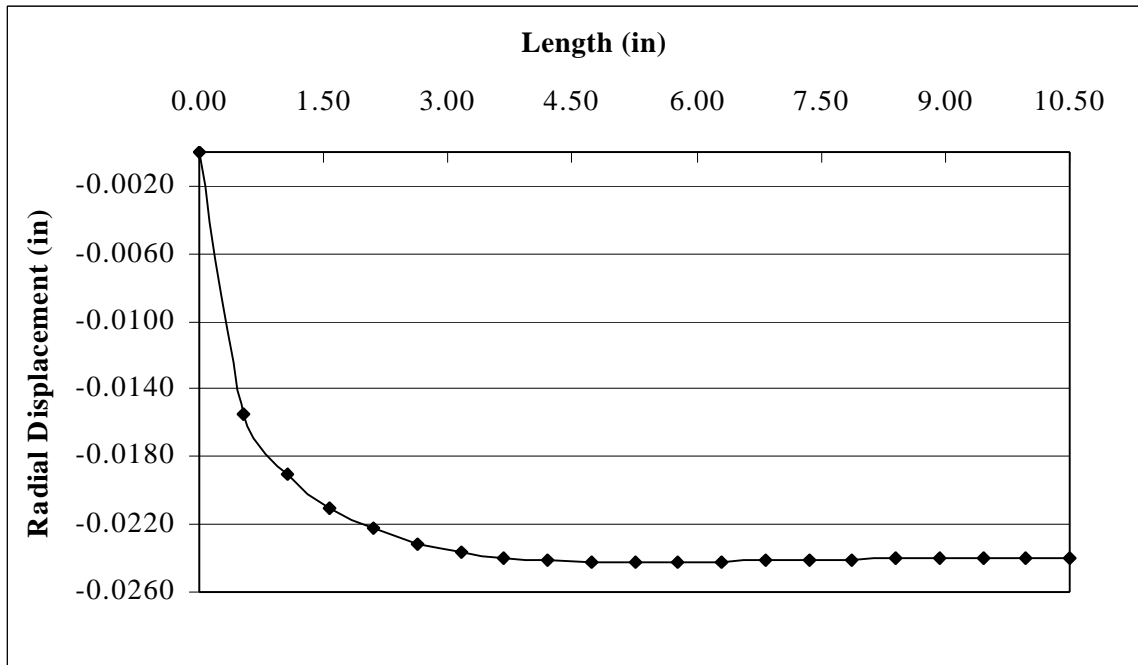


Figure MAF-7. Radial Displacement from the Cylinder End to the Mid-length of the

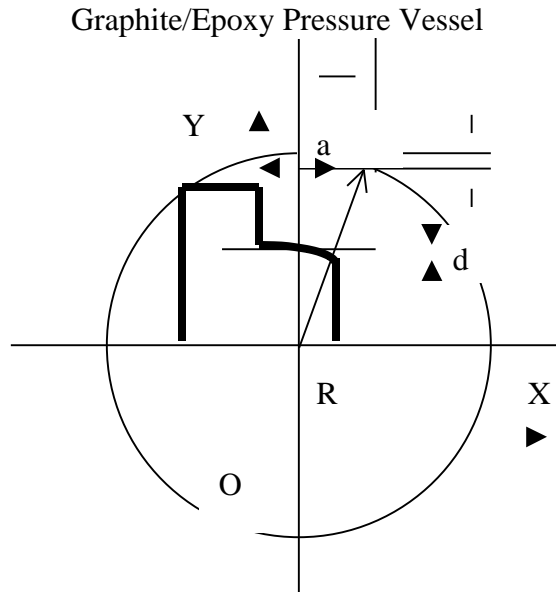


Figure MAF-8. End-cap and its Plug Circle with Radius R

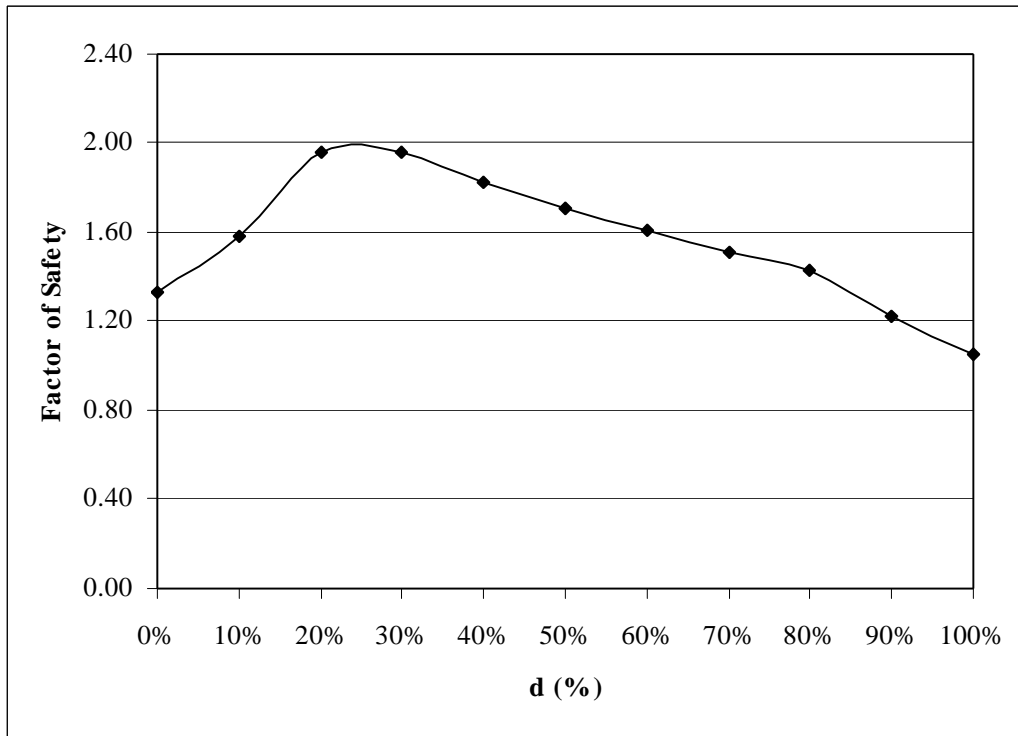


Figure MAF-9. Factor of Safety Based on different d% for the Ti-6Al-4V Pressure Vessel

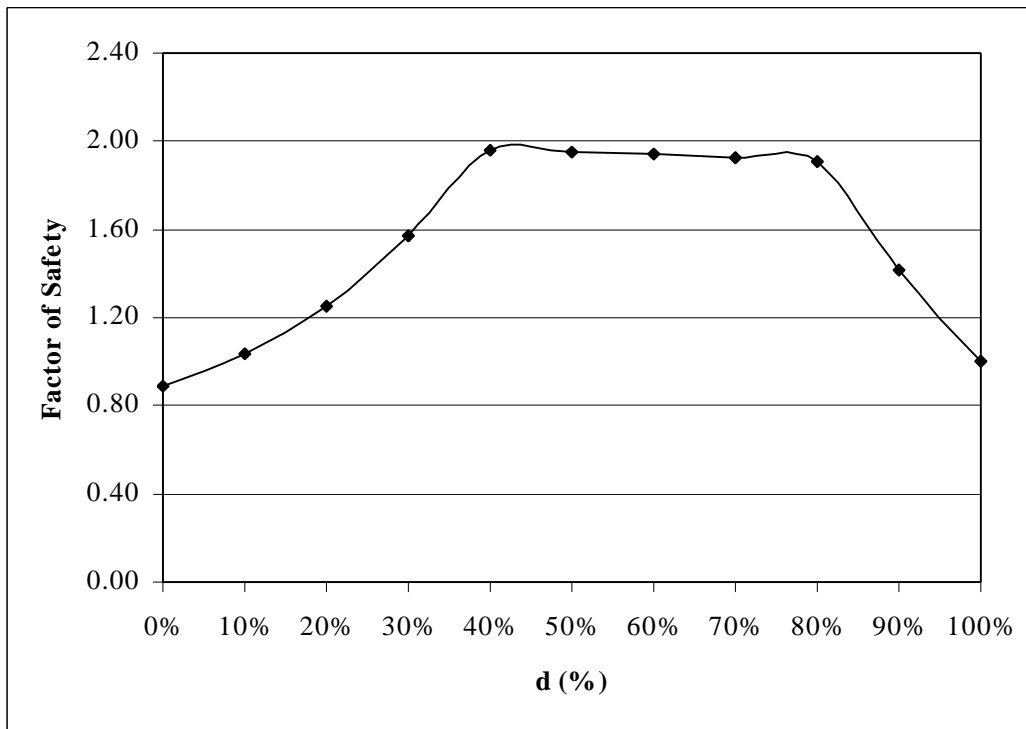


Figure MAF-10. Factor of Safety Based on different d% for the APC-2/AS4 Pressure Vessel

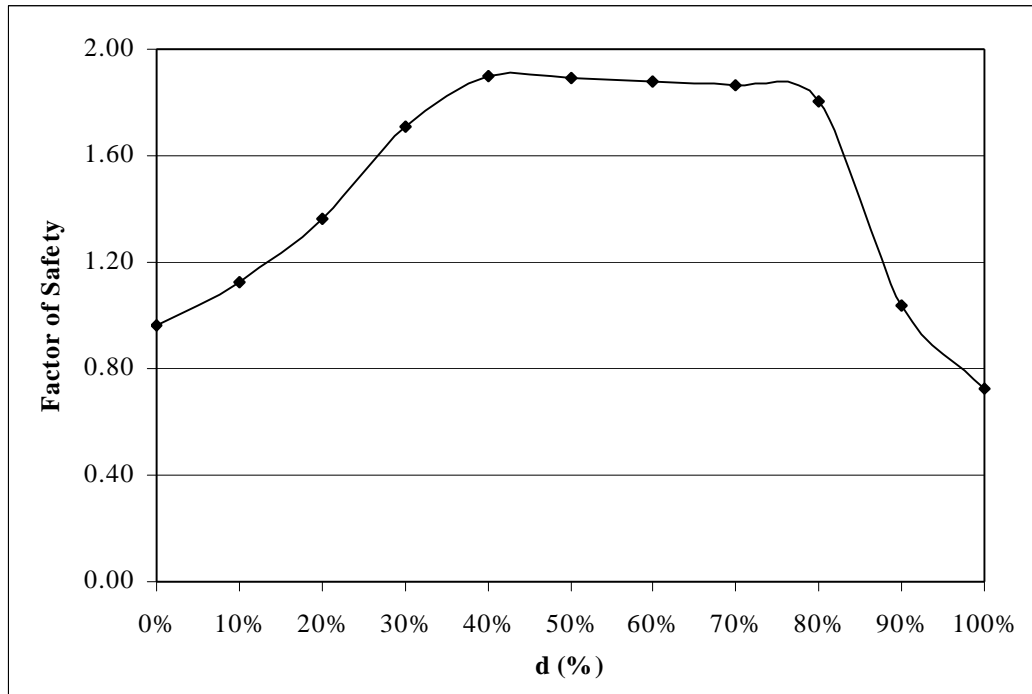


Figure MAF-11. Factor of Safety Based on different d% for the Graphite/Epoxy Pressure Vessel

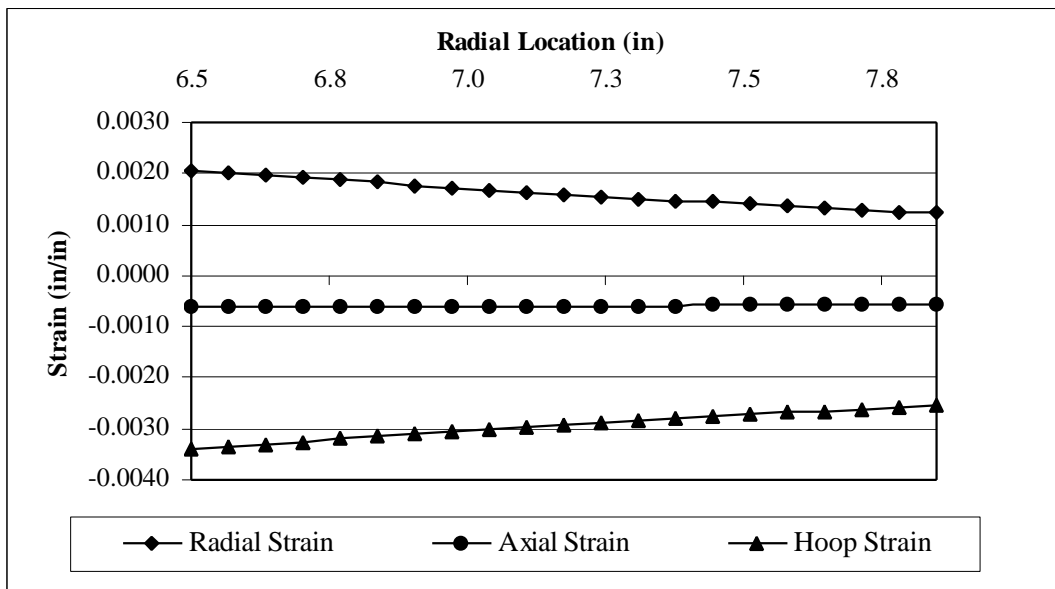


Figure MAF-12. Strain Distributions through the Thickness at the Mid-length of the Ti-6Al-4V Pressure Vessel at 70°F

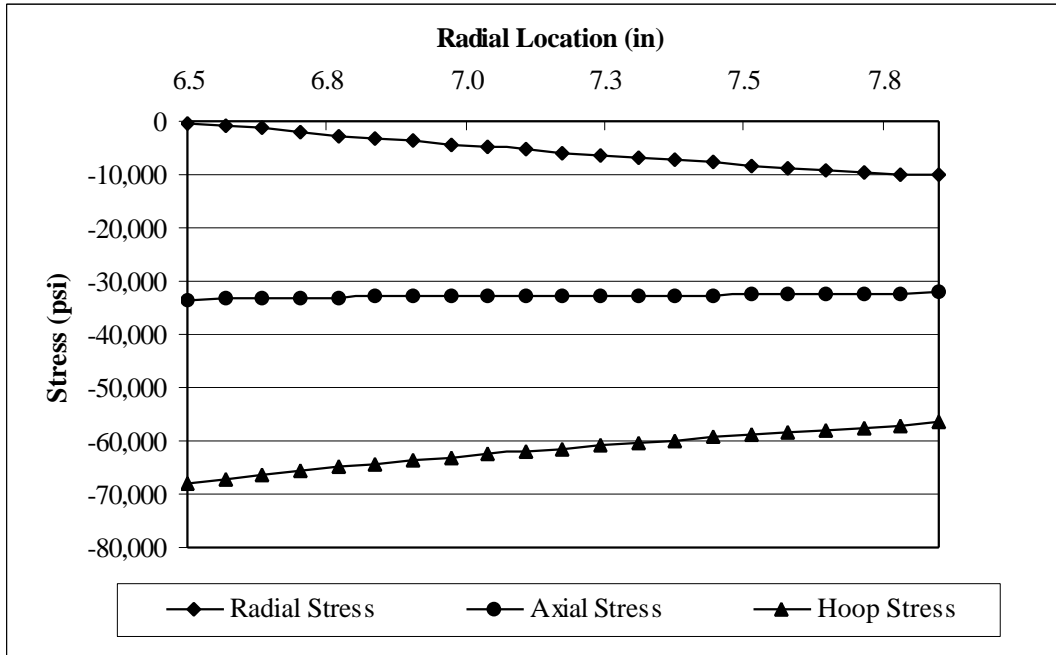


Figure MAF-13. Stress Distributions through the Thickness at the Mid-length of the Ti-6Al-4V Pressure Vessel at 70°F

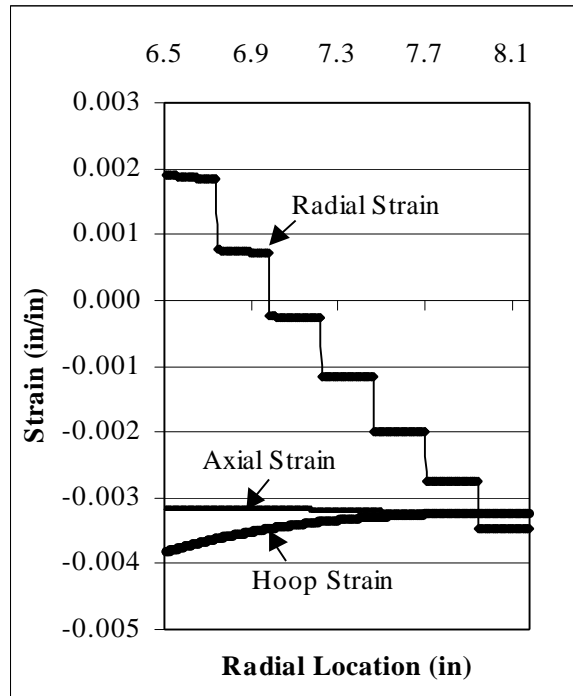


Figure MAF-14. Strain Distributions through the Thickness at the Mid-length of the APC-2/AS4 Pressure Vessel at 70°F

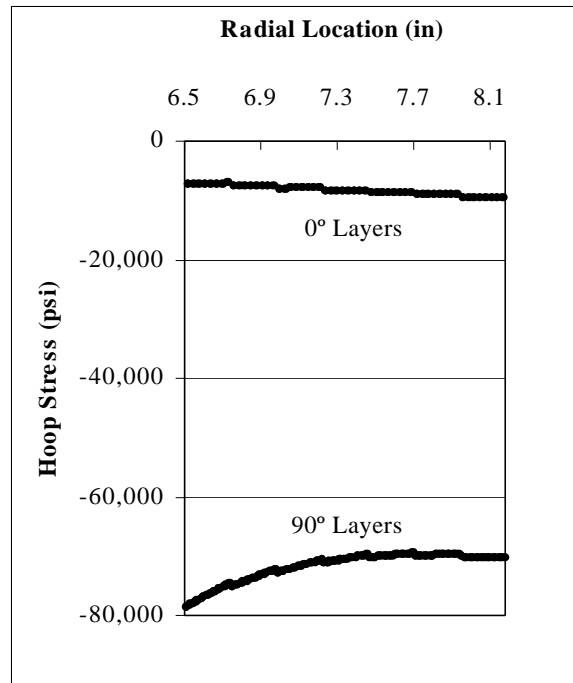


Figure MAF-15. Hoop Stress Distribution through the Thickness at the Mid-length of the APC-2/AS4 Pressure Vessel at 70°F

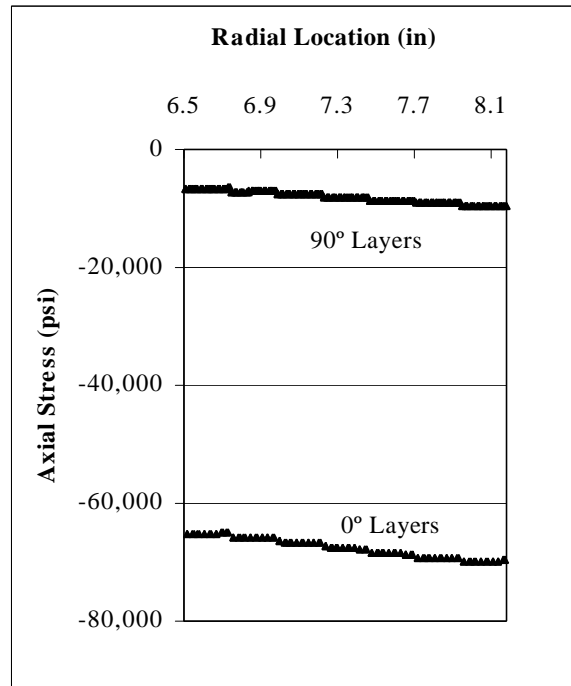


Figure MAF-16. Axial Stress Distribution through the Thickness at the Mid-length of the APC-2/AS4 Pressure Vessel at 70°F

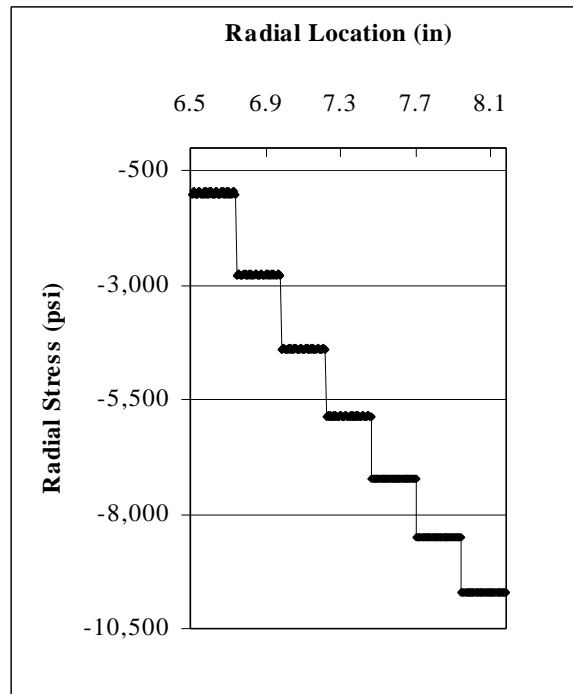


Figure MAF-17. Radial Stress Distribution through the Thickness at the Mid-length of the APC-2/AS4 Pressure Vessel at 70°F

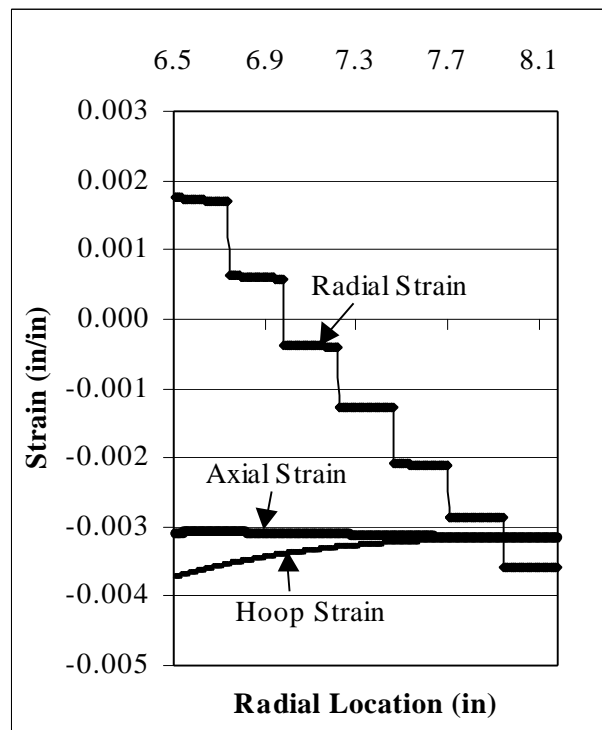


Figure MAF-18. Strain Distributions through the Thickness at the Mid-length of the Graphite/Epoxy Pressure Vessel at 70°F

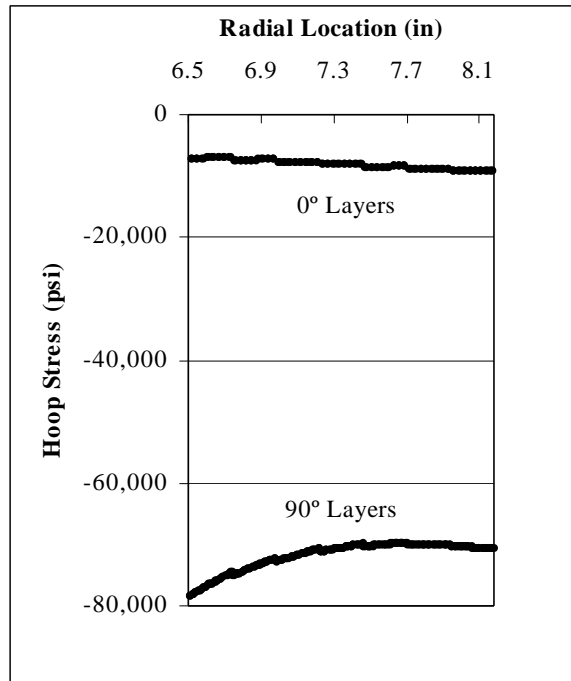


Figure MAF-19. Hoop Stress Distribution through the Thickness at the Mid-length of the Graphite/Epoxy Pressure Vessel at 70°F

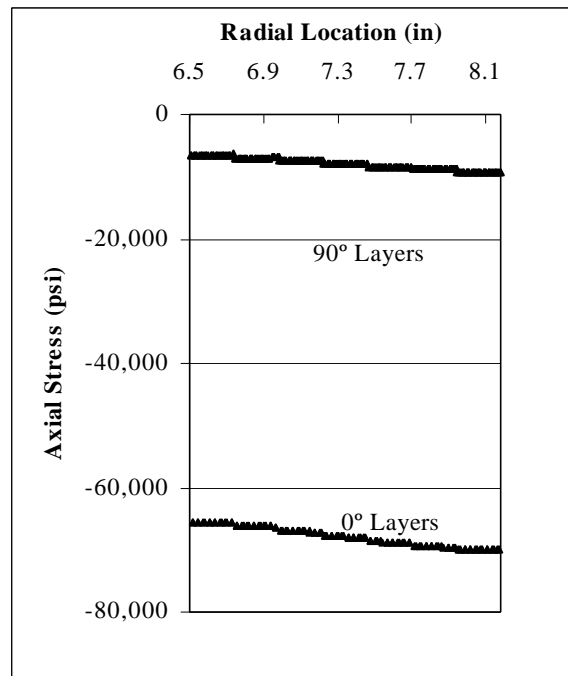


Figure MAF-20. Axial Stress Distribution through the Thickness at the Mid-length of the Graphite/Epoxy Pressure Vessel at 70°F

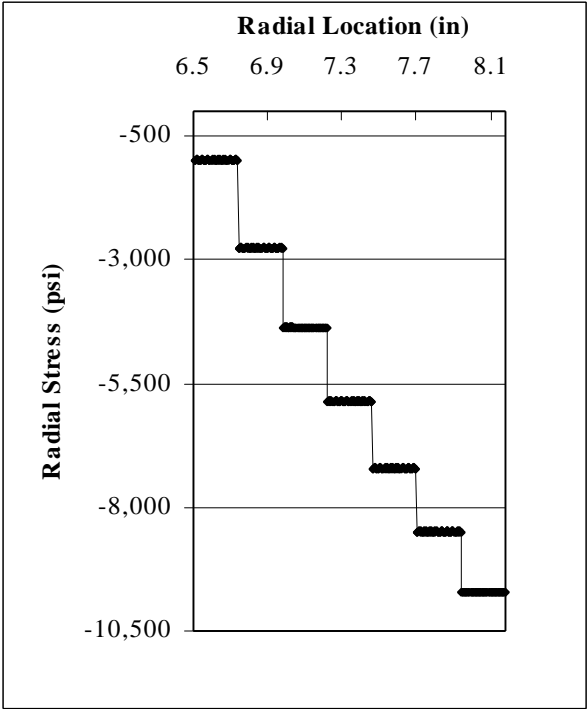


Figure MAF-21. Radial Stress Distribution through the Thickness at the Mid-length of the Graphite/Epoxy Pressure Vessel at 70°F

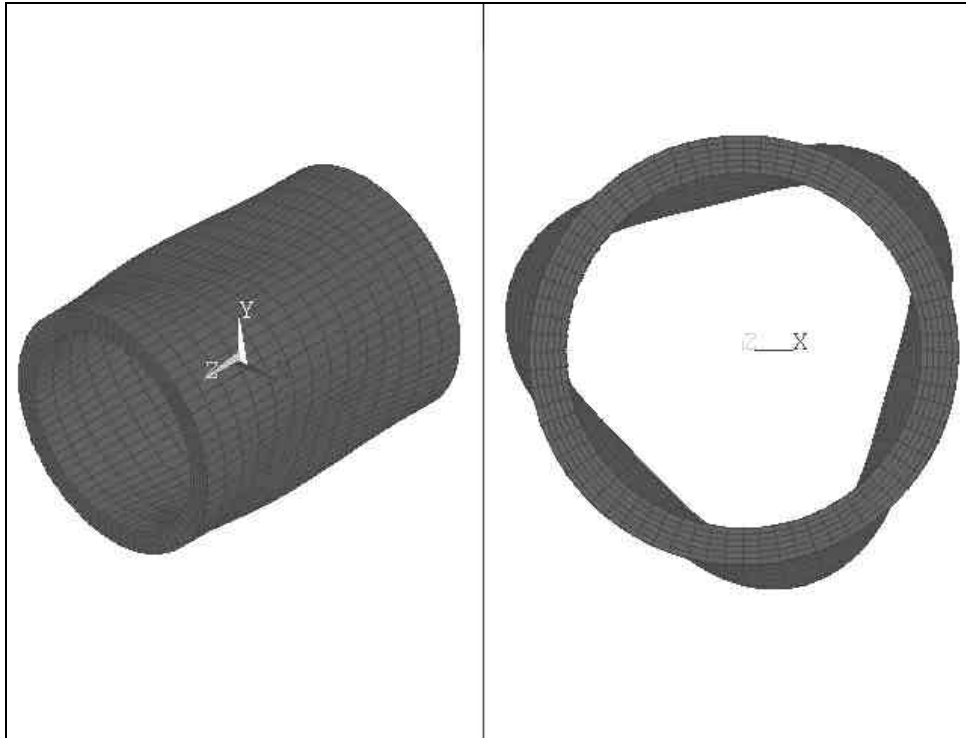


Figure MAF-22. Full Buckling Mode of the Ti-6Al-4V Pressure Vessel

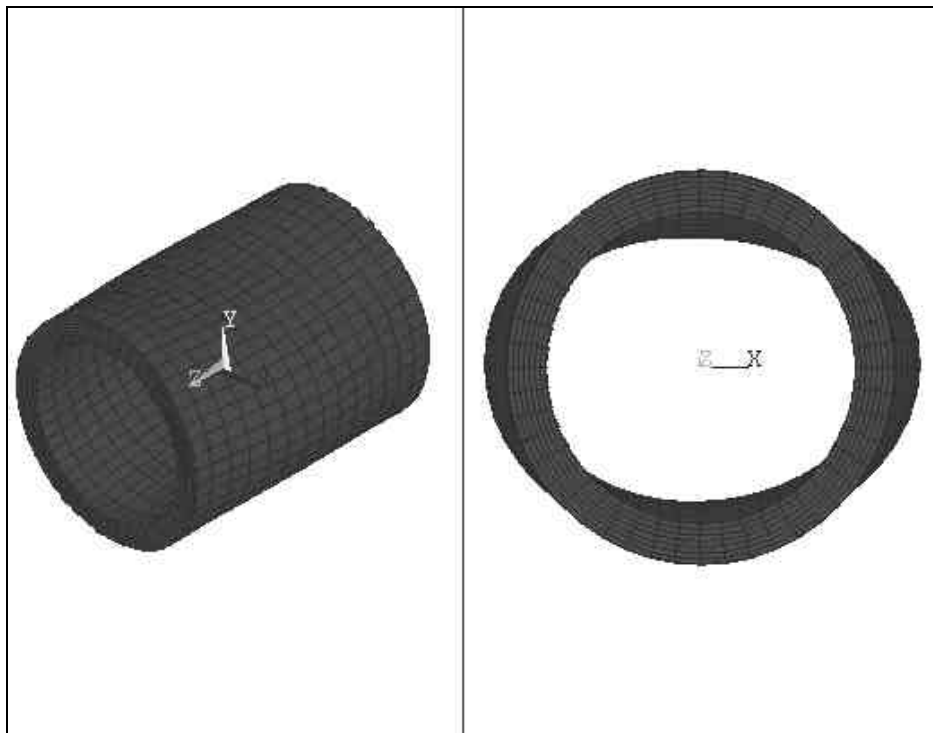


Figure MAF-23. Typical Full Buckling Mode of the Composite Pressure Vessels

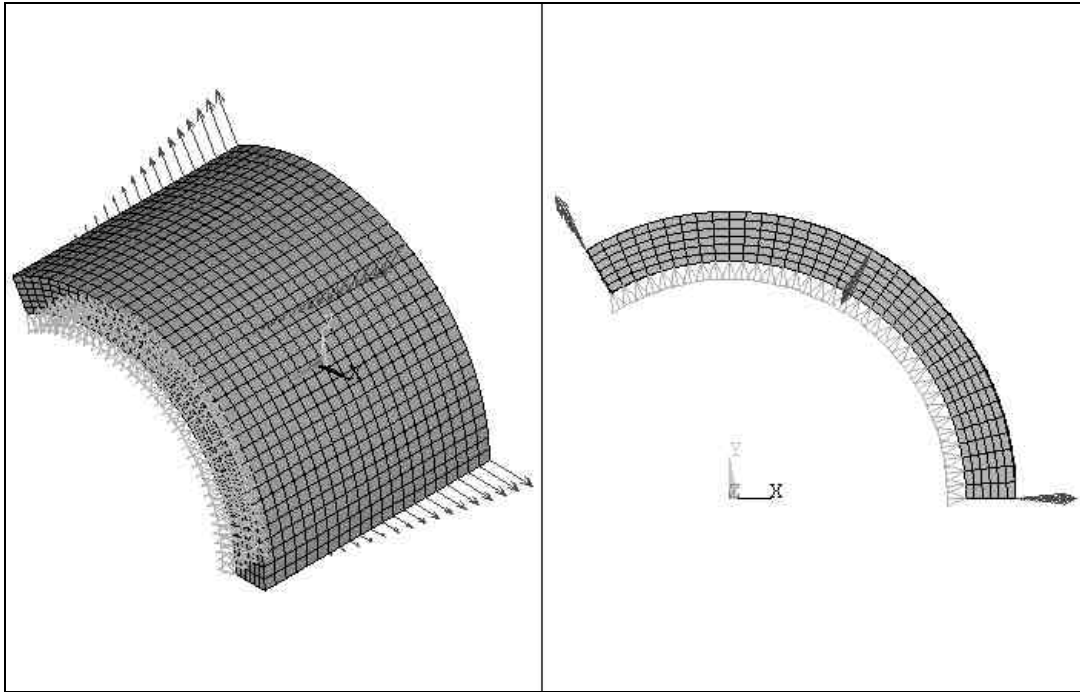


Figure MAF-24. One-Sixth of the Model of the Ti-6Al-4V Pressure Vessel

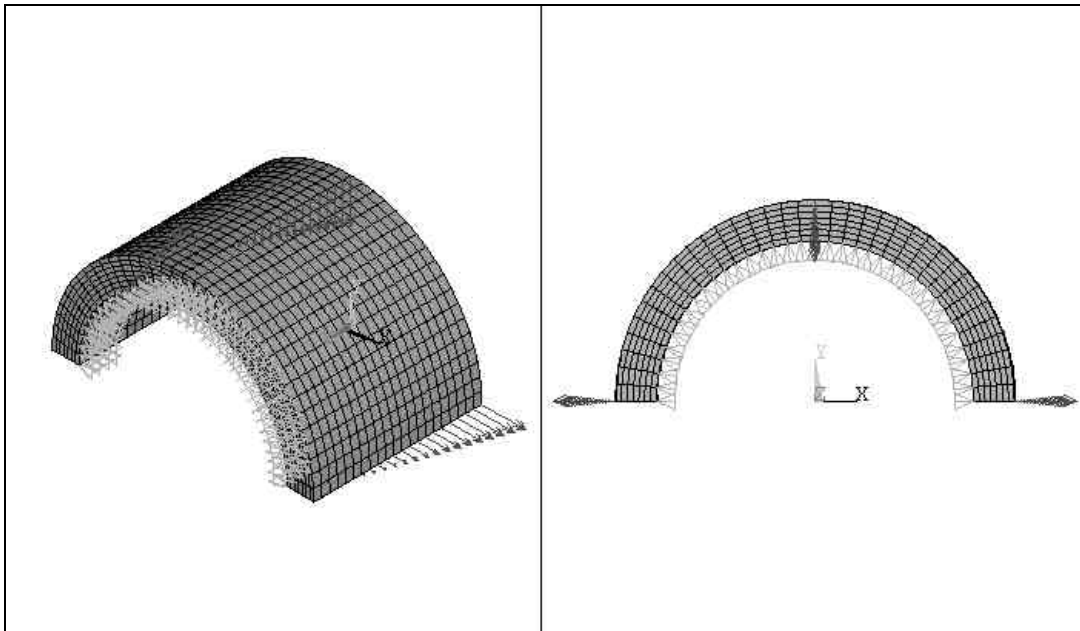


Figure MAF-25. One-fourth of the Model of the Composite Pressure Vessels

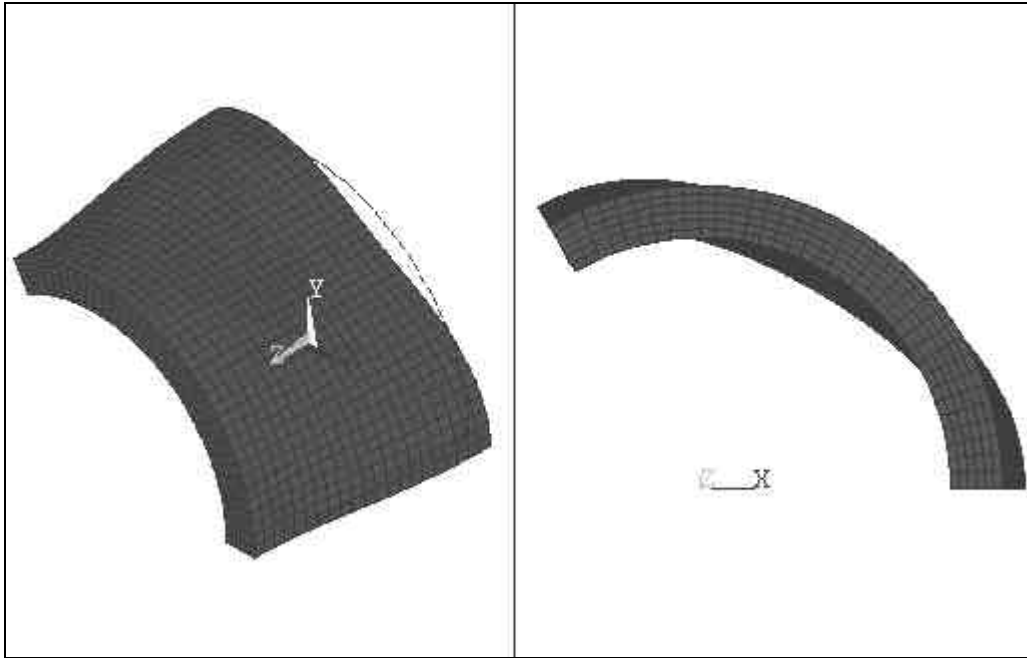


Figure MAF-26. Typical First Mode Shape of One-sixth of the Ti-6Al-4V Pressure Vessel

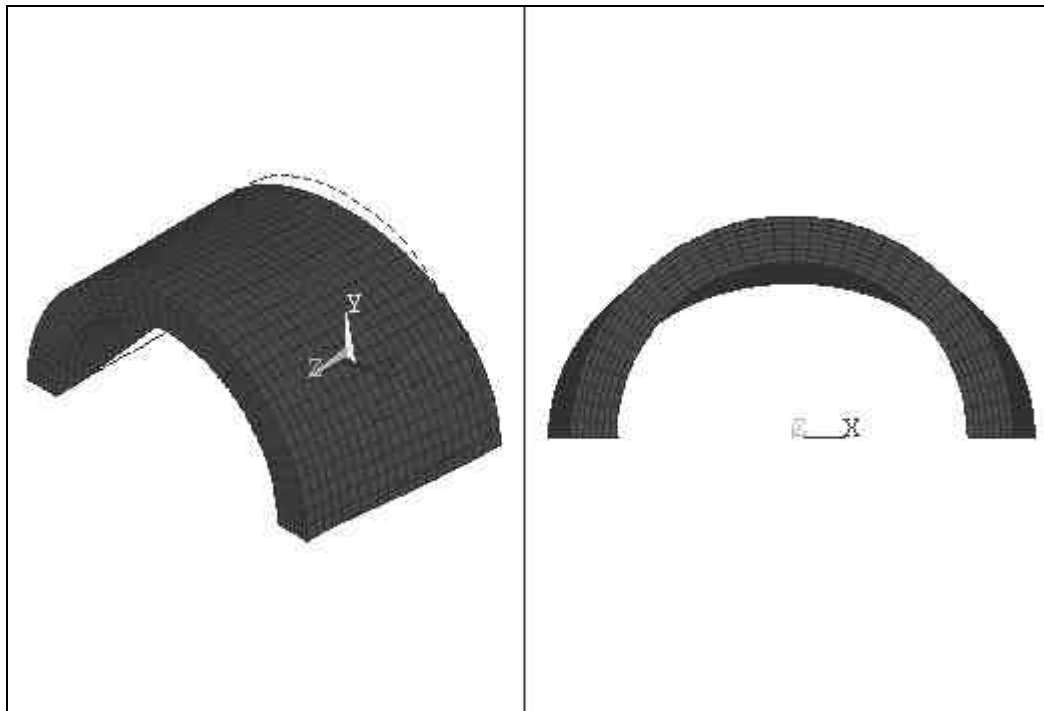


Figure MAF-27. Typical First Mode Shape of One-fourth of the Composite Pressure Vessels

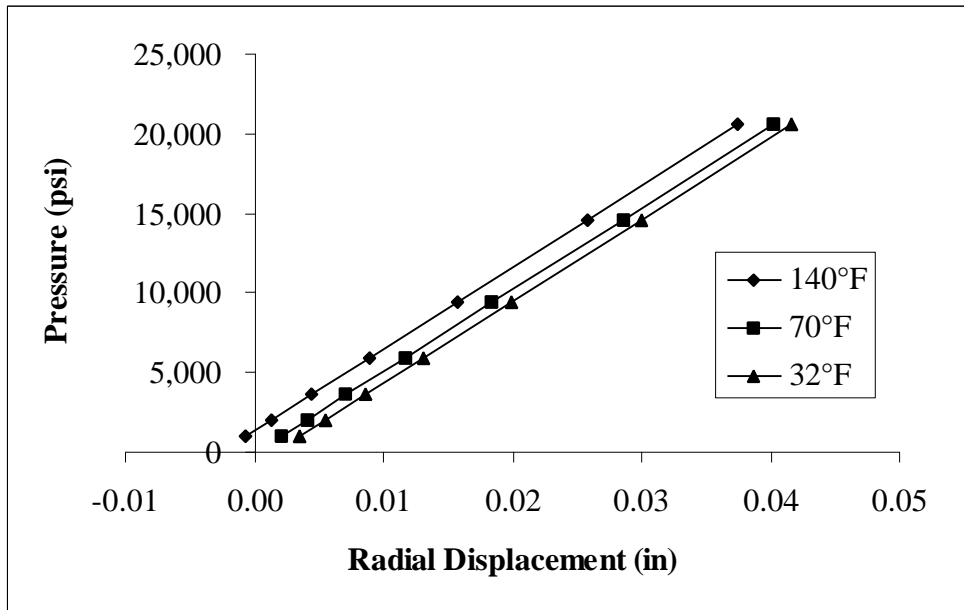


Figure MAF-28. Pressure-Displacement Curve of the Ti-6Al-4V Pressure Vessel Considering Thermal Effects at Mid-length

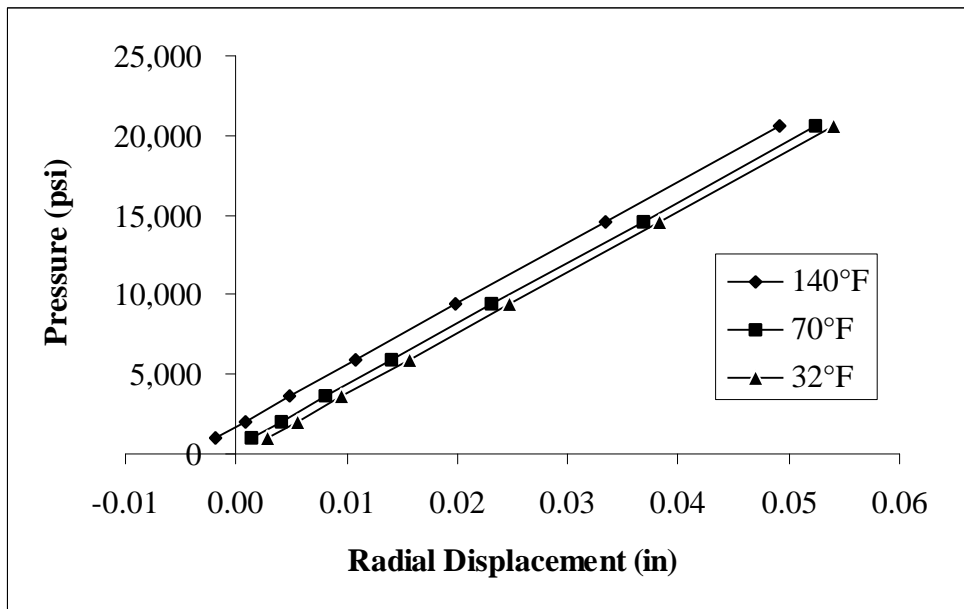


Figure MAF-29. Pressure-Displacement Curve of the APC-2/AS4 Pressure Vessel Considering Hygrothermal Effects at Mid-length

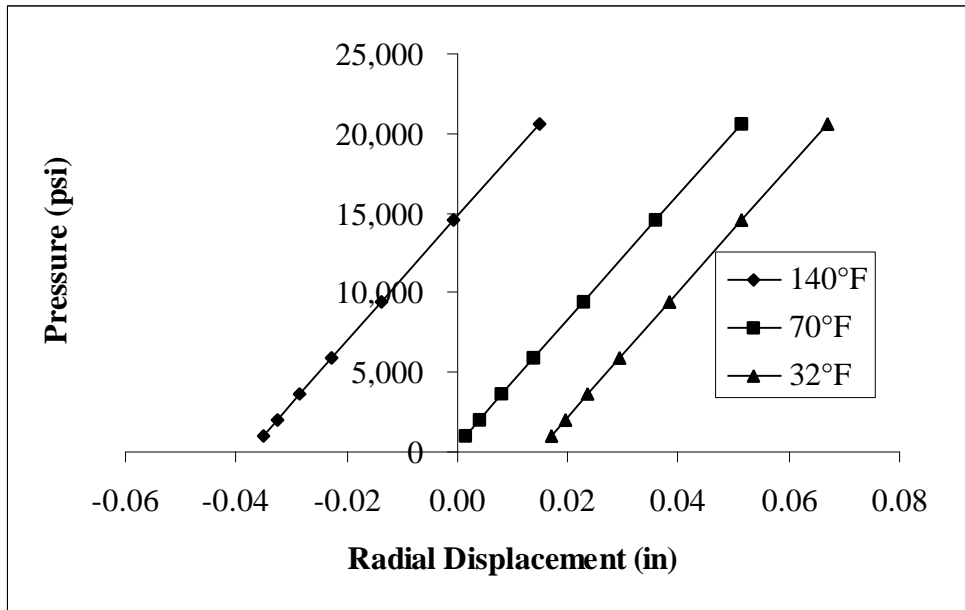


Figure MAF-30. Pressure-Displacement Curve of the Graphite/Epoxy Pressure Vessel
Considering Hygrothermal Effects at Mid-length

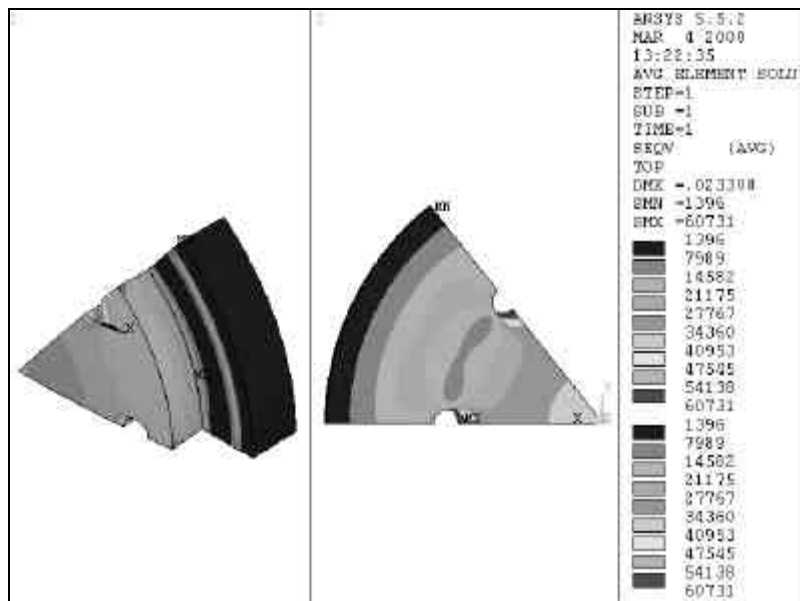


Figure MAF-31. Von Mises Stress Distribution of the End-cap

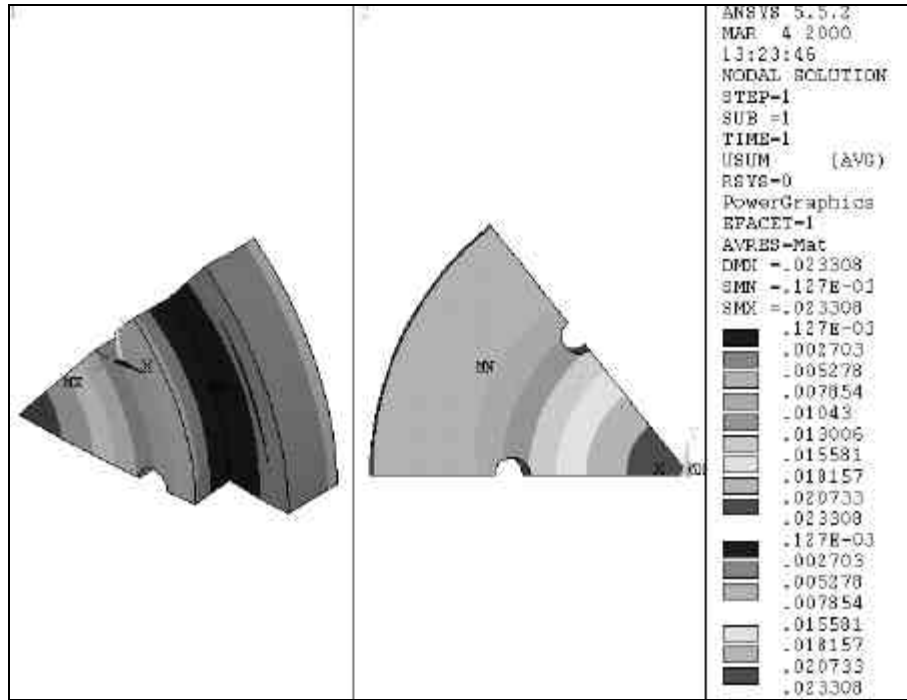


Figure MAF-32. Displacement of the End-cap

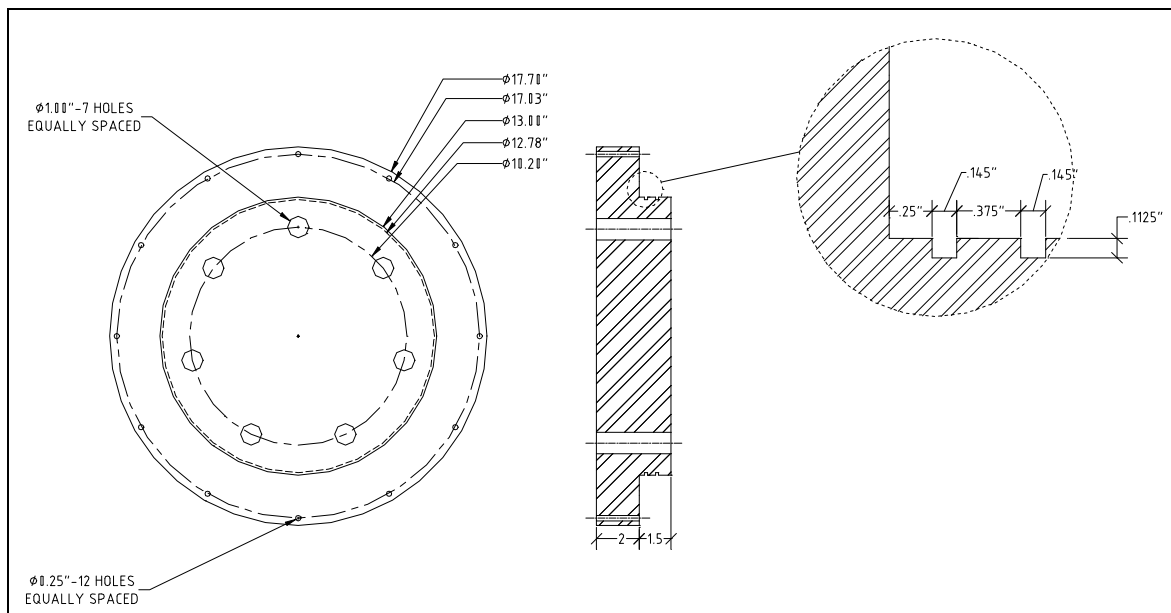


Figure MAF-33. End-Cap Dimensions

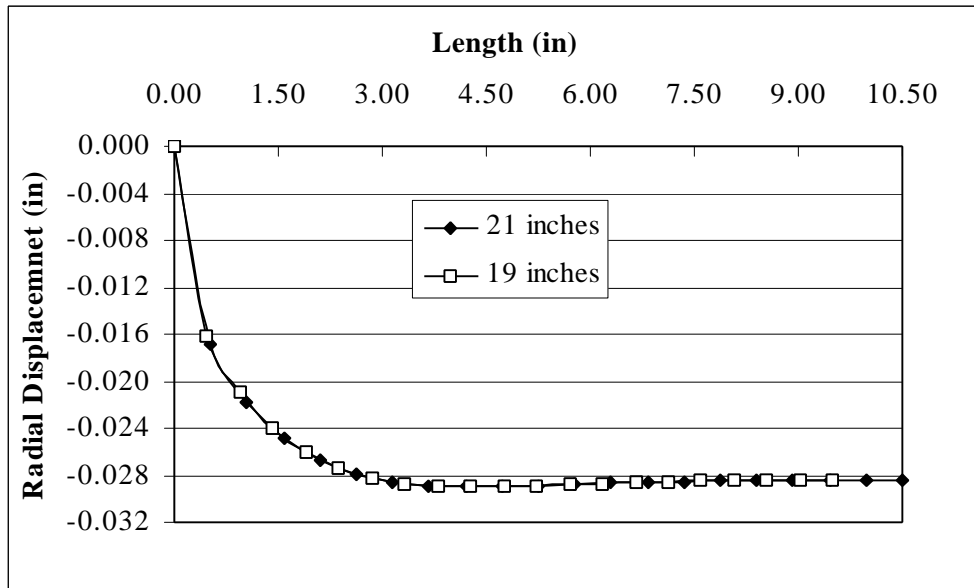


Figure MAF-34. Radial Displacement from the Cylinder End to the Mid-length of the 21” and 19” Pressure Vessels with Edge Simply Supported for SAUVIM Pressure Vessel

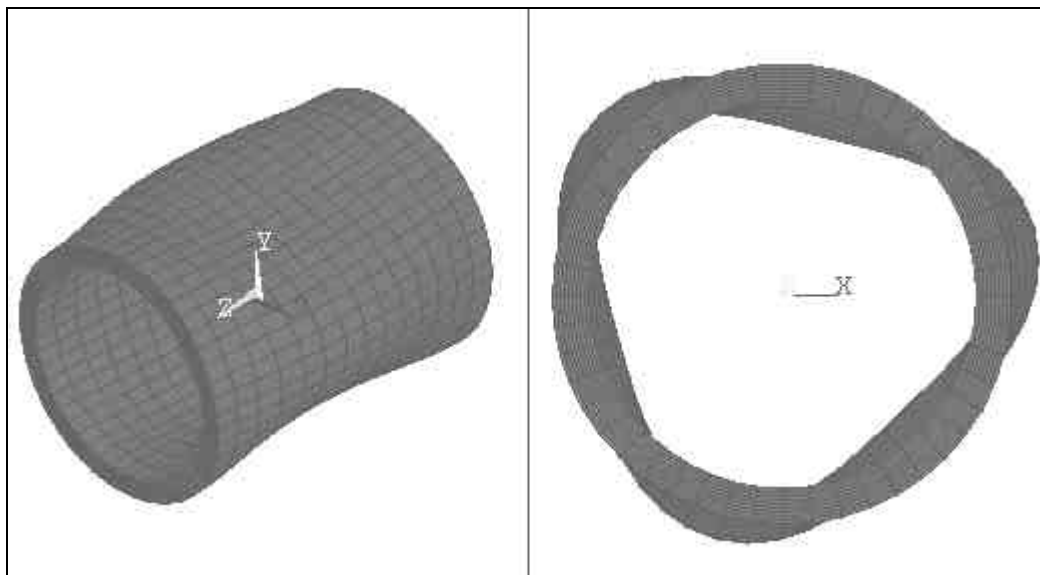


Figure MAF-35. Buckling Mode of the 21” APC-2/AS4 Pressure Vessel for SAUVIM

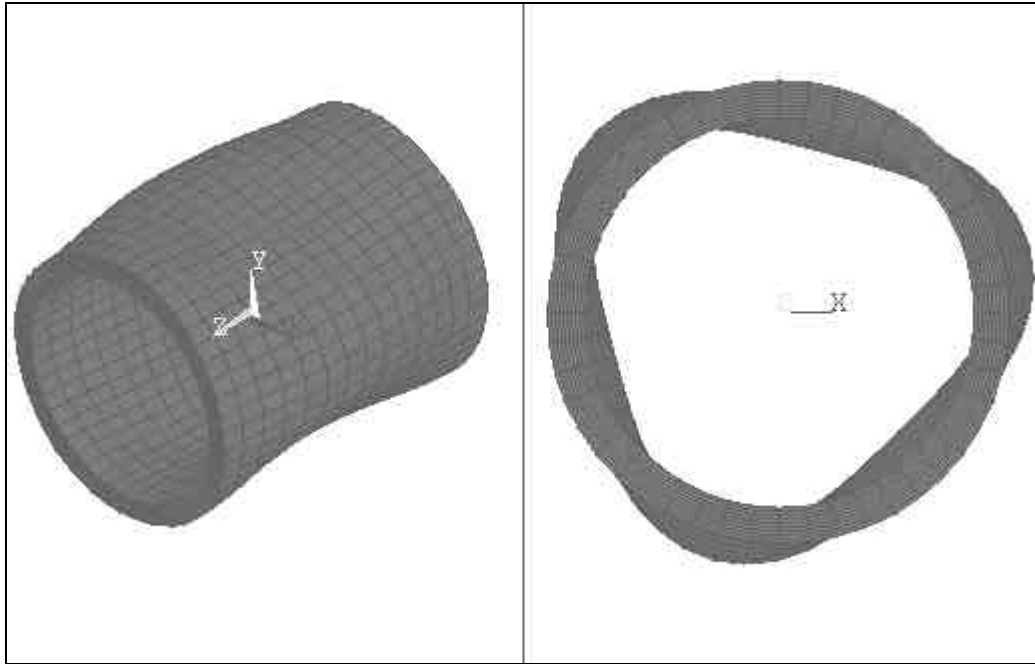


Figure MAF-36. Buckling Mode of the 19" APC-2/AS4 Pressure Vessel for SAUVIM

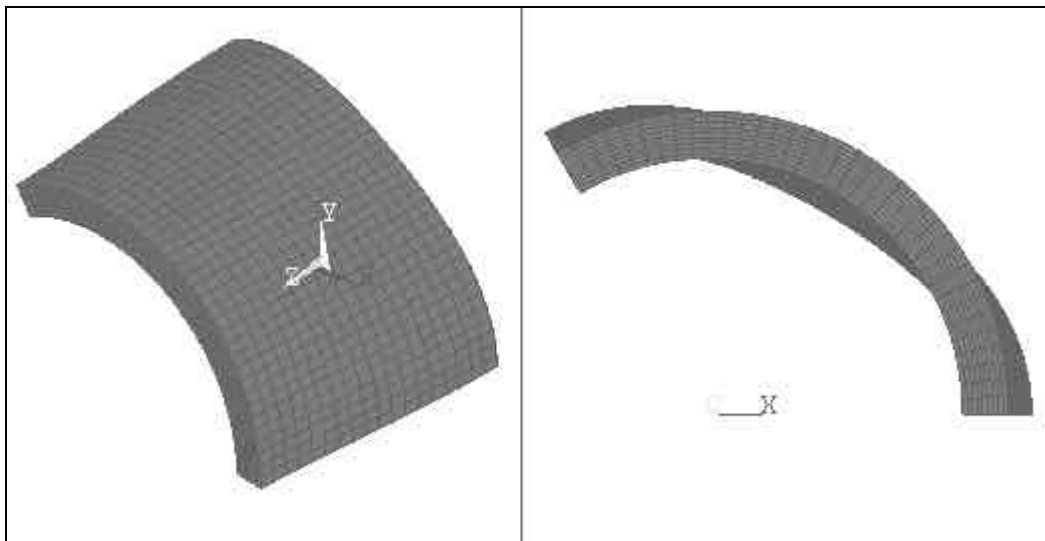


Figure MAF-37. First Mode Shape of the One-sixth of the 21" Pressure Vessel for SAUVIM

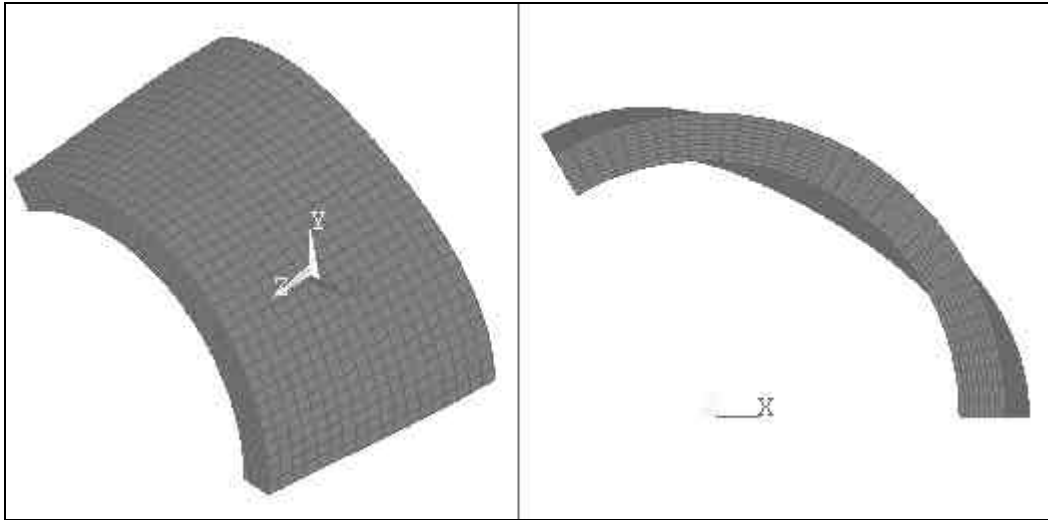


Figure MAF-38. First Mode Shape of the One-sixth of the 19" Pressure Vessel for SAUVIM

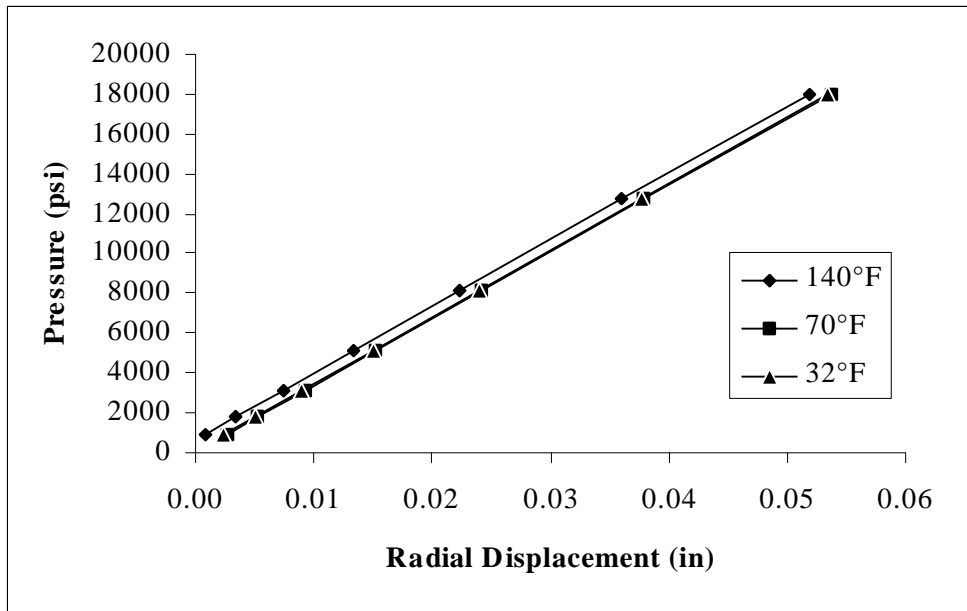


Figure MAF-39. Pressure-Displacement Curve of the 21" APC-2/AS4 Pressure Vessel with Optimum Tapered Radius Considering Hygrothermal Effects for SAUVIM

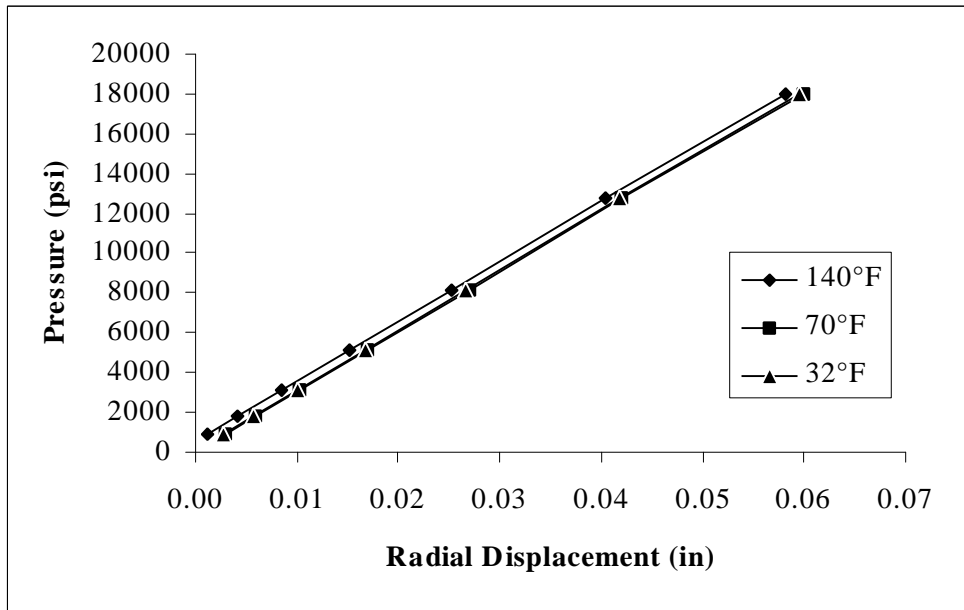


Figure MAF-40. Pressure-Displacement Curve of the 19" APC-2/AS4 Pressure Vessel with Optimum Tapered Radius Considering Hygrothermal Effects for SAUVIM

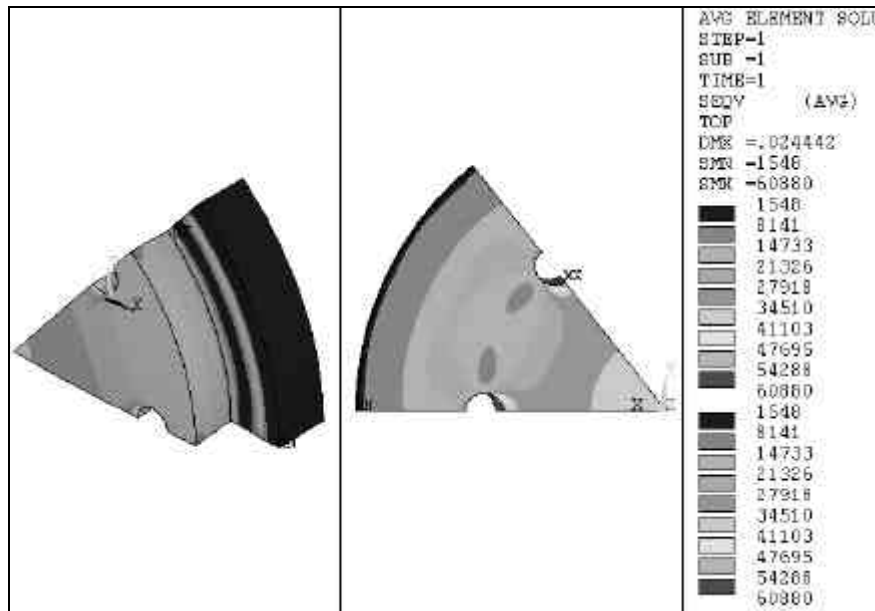


Figure MAF-41. Von Mises Stress Distribution of the End-cap for SAUVIM

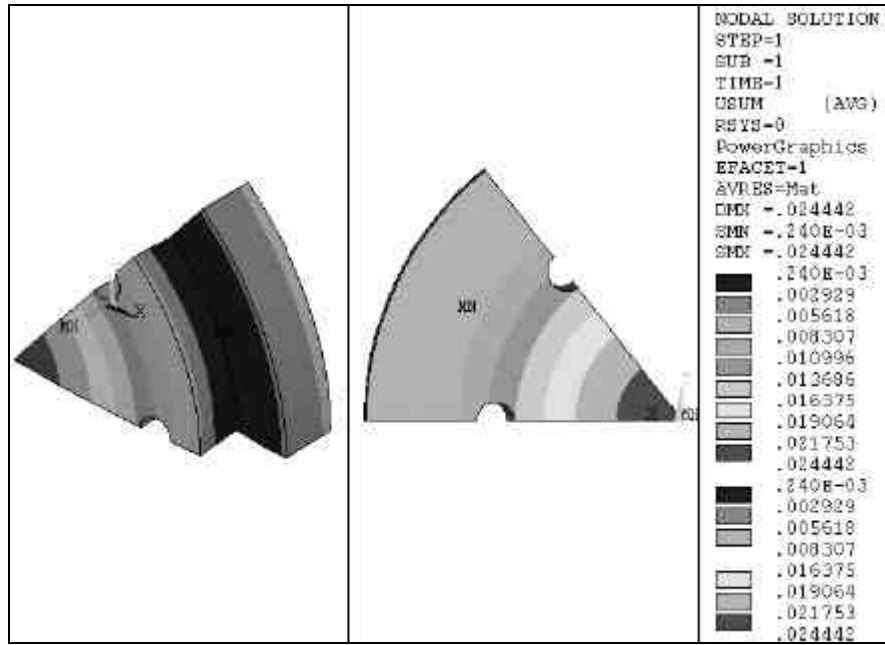


Figure MAF-42. Displacement of the End-cap for SAUVIM

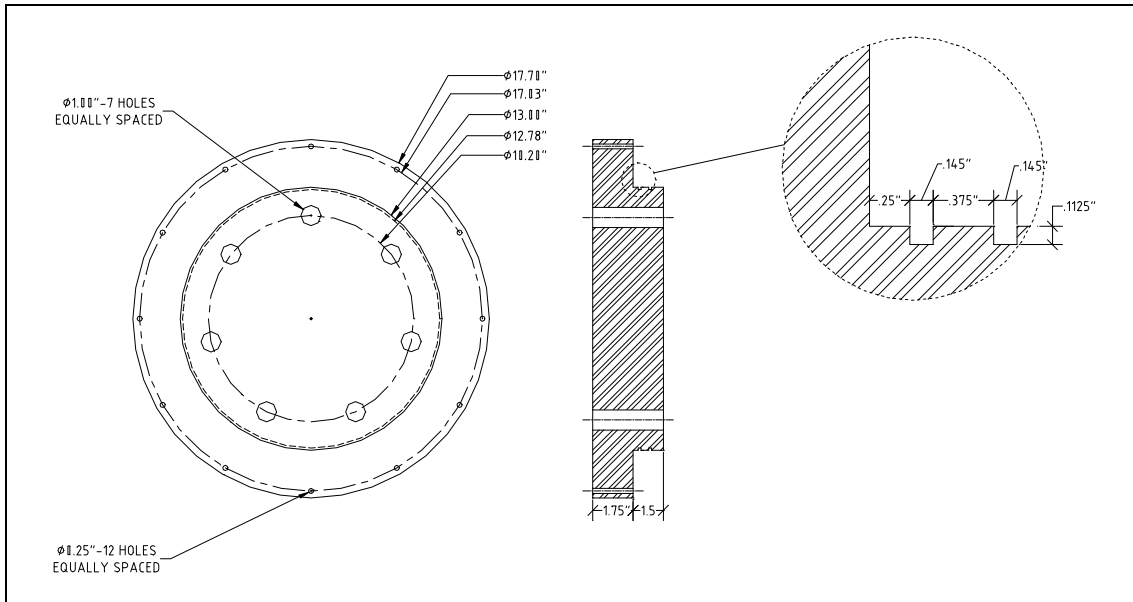


Figure MAF-43. End-Cap Dimensions for SAUVIM

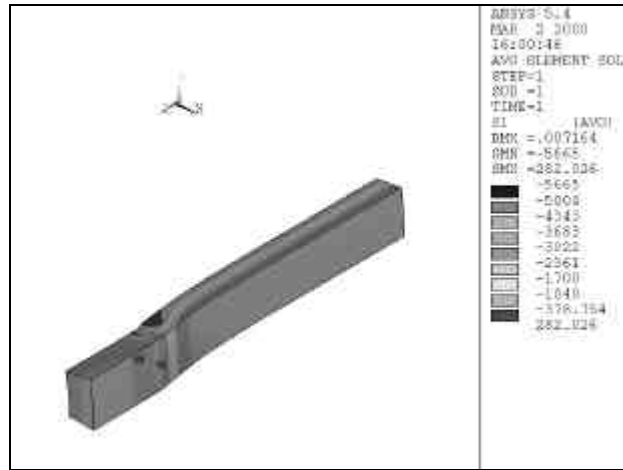


Figure MAF-44. Radial Stress of the Scaled Pressure Vessel

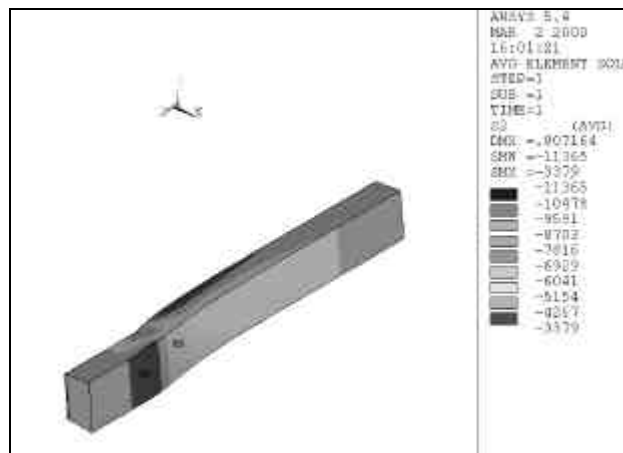


Figure MAF-45. Axial Stress of the Scaled Pressure Vessel

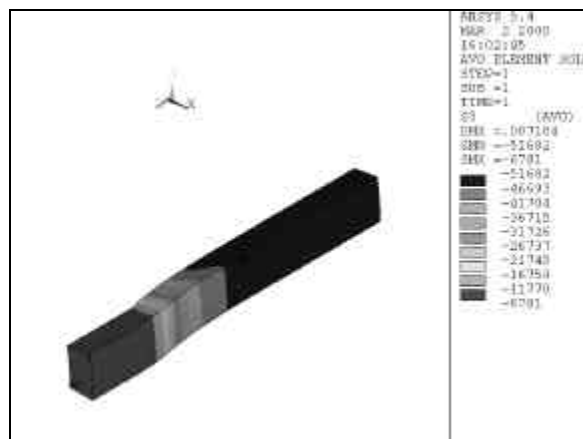


Figure MAF-46. Hoop Stress of the Scaled Pressure Vessel

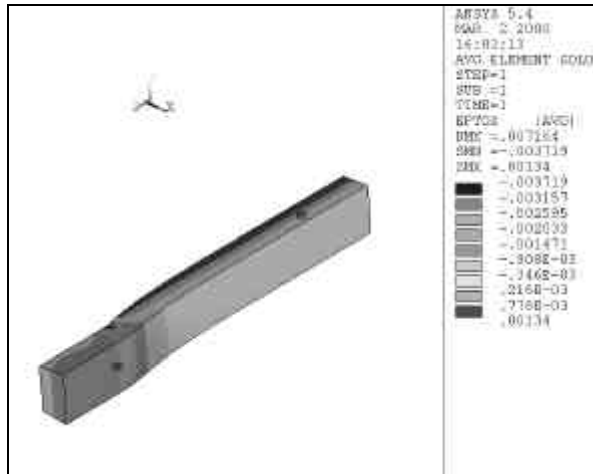


Figure MAF-47. Radial Strain of the Scaled Pressure Vessel

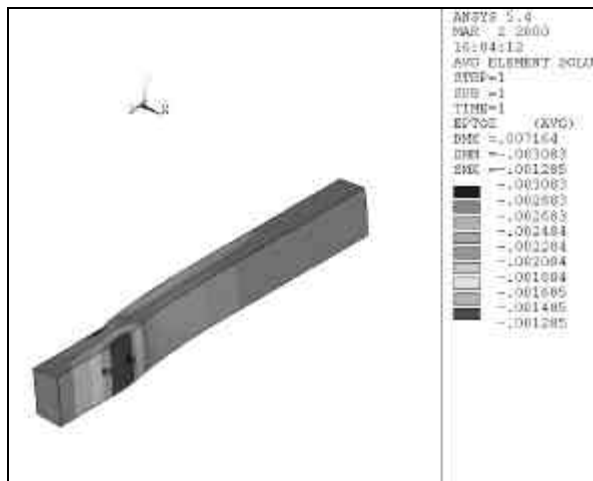


Figure MAF-48. Axial Strain of the Scaled Pressure Vessel

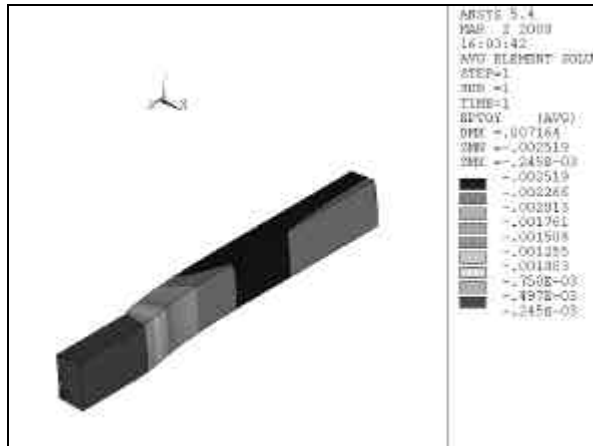


Figure MAF-49. Hoop Strain of the Scaled Pressure Vessel

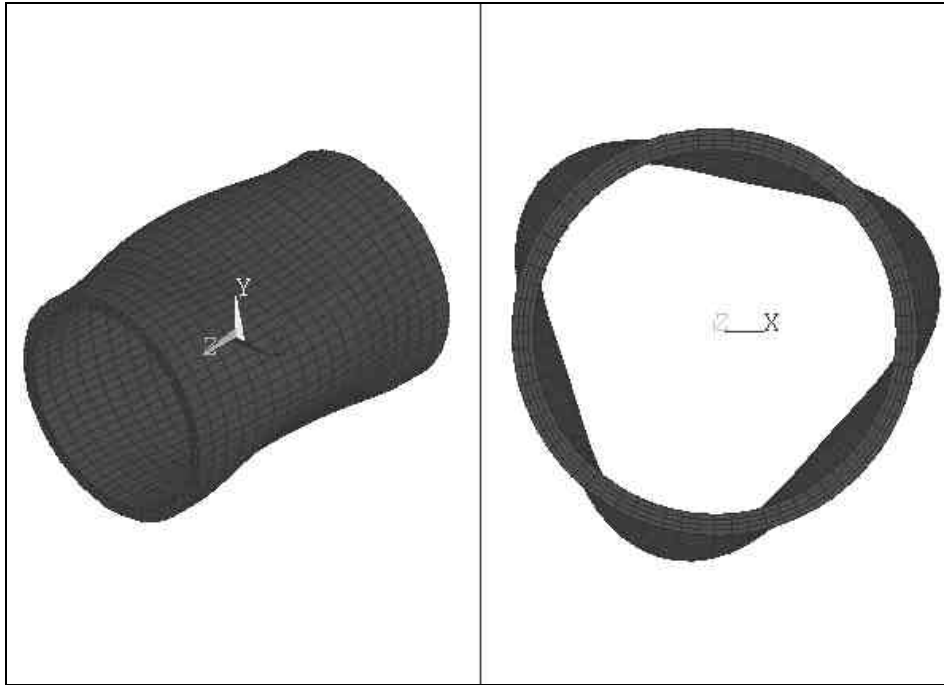


Figure MAF-50. Buckling Mode of the APC-2/AS4 Scaled Pressure Vessel

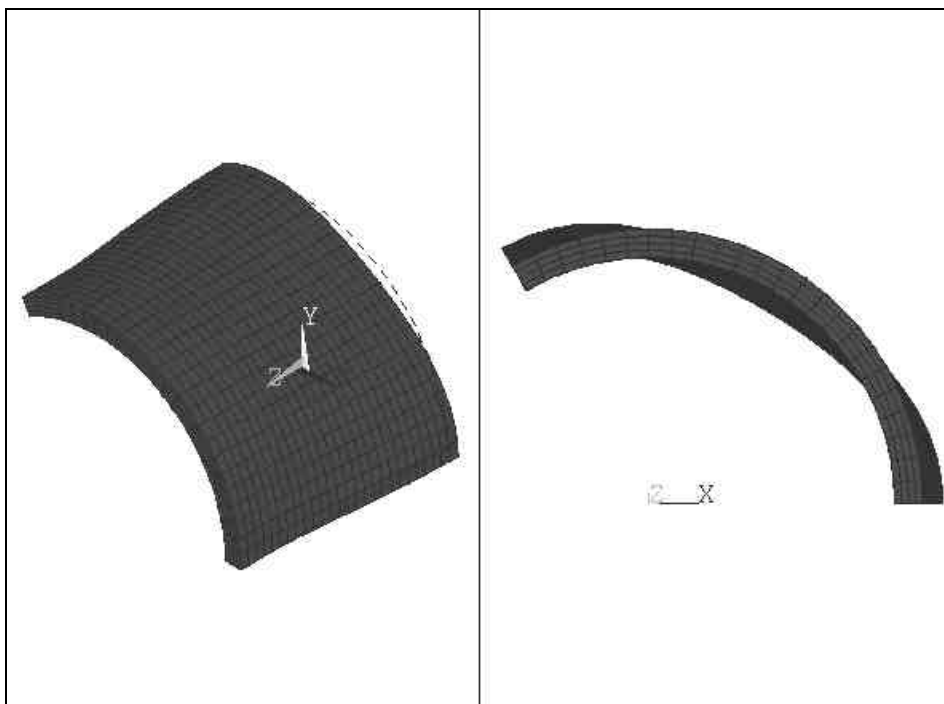


Figure MAF-51. First Mode Shape of the One-sixth of the Scaled Pressure Vessel

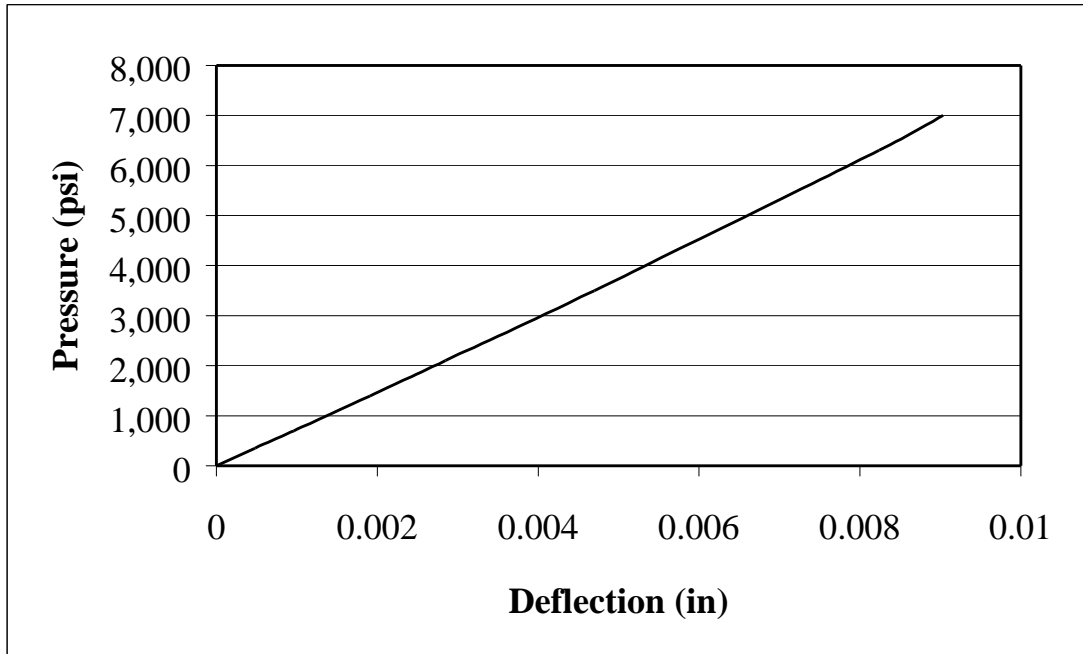


Figure MAF-52. Pressure-Deflection (Mid-length) Curve of Scaled Pressure Vessel

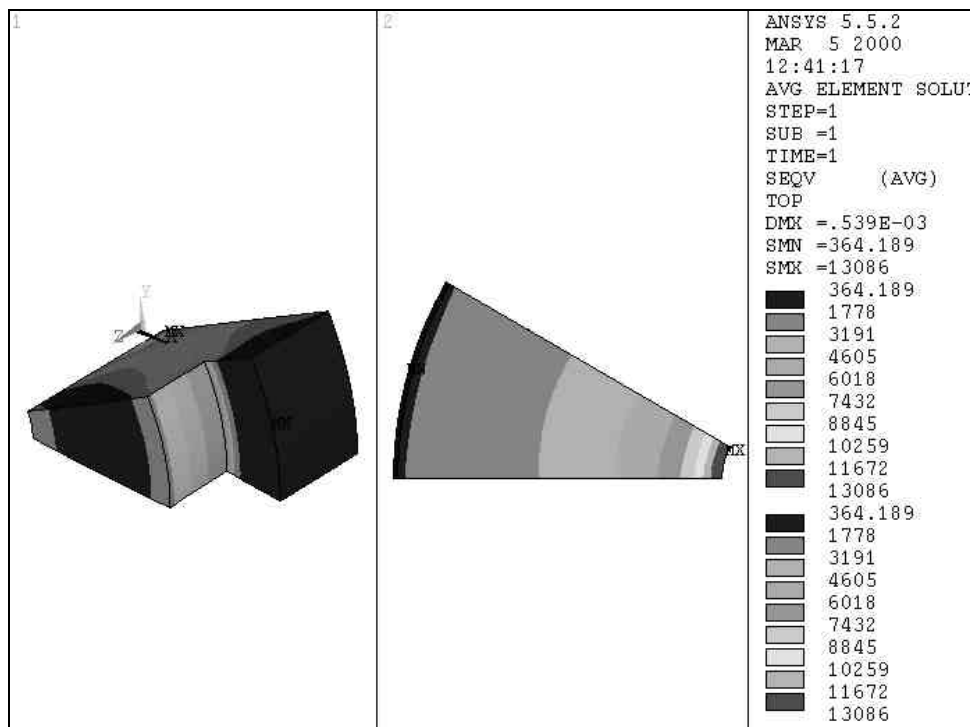


Figure MAF-53. Von Mises Stress Distribution of the Scaled End-cap

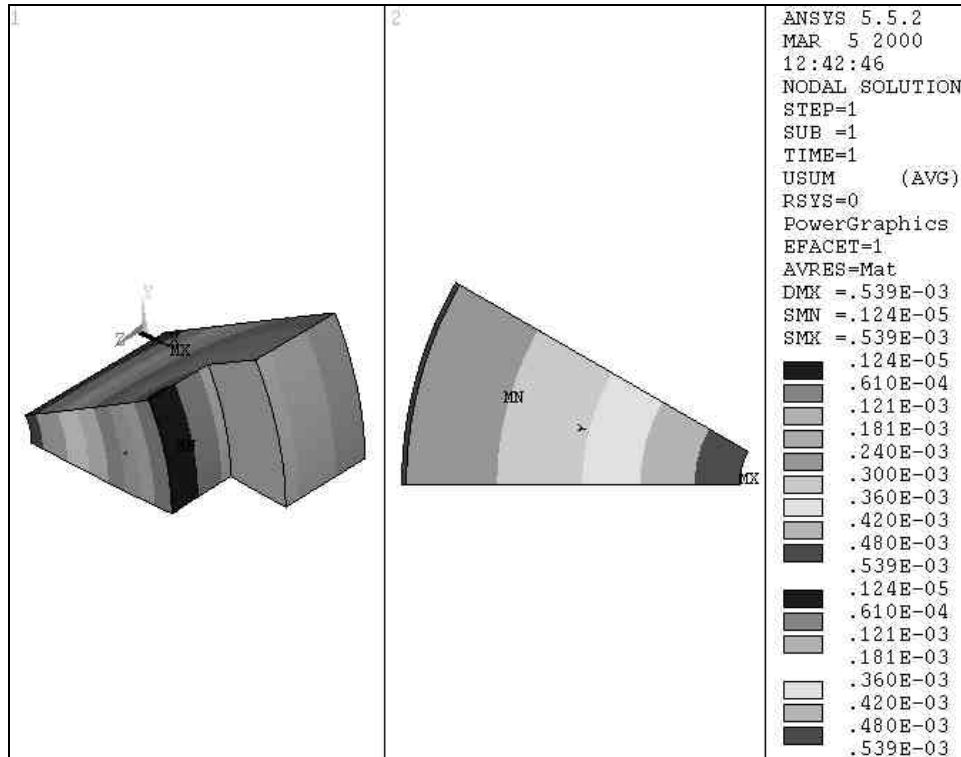


Figure MAF-54. Displacement of the Scaled End-cap

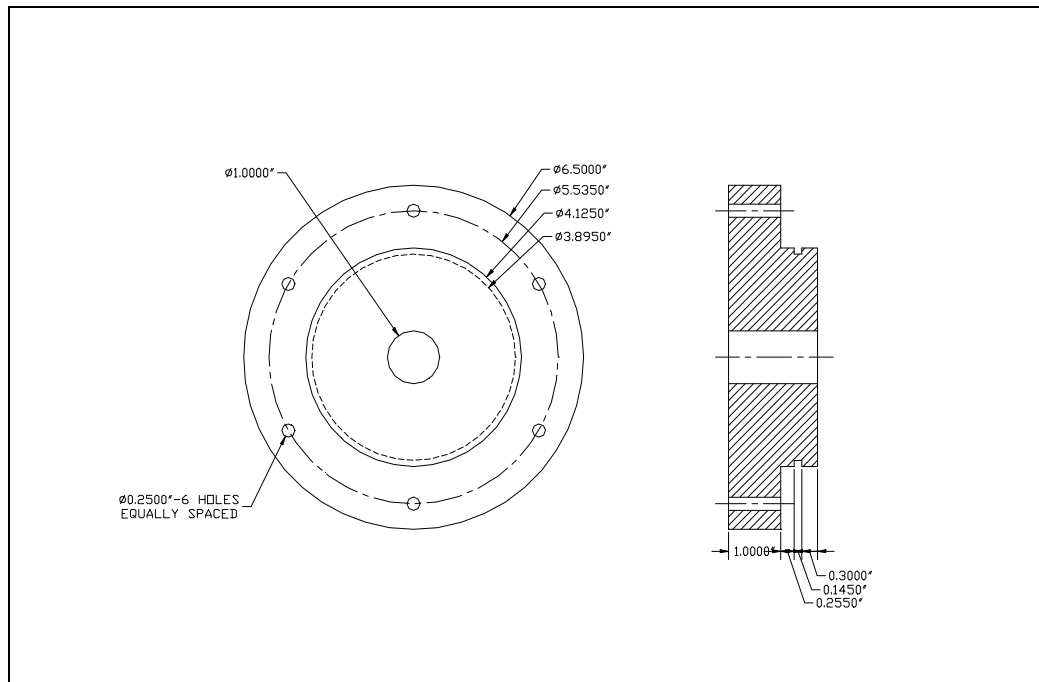


Figure MAF-55. End-Cap Dimensions for Scaled Pressure Vessel

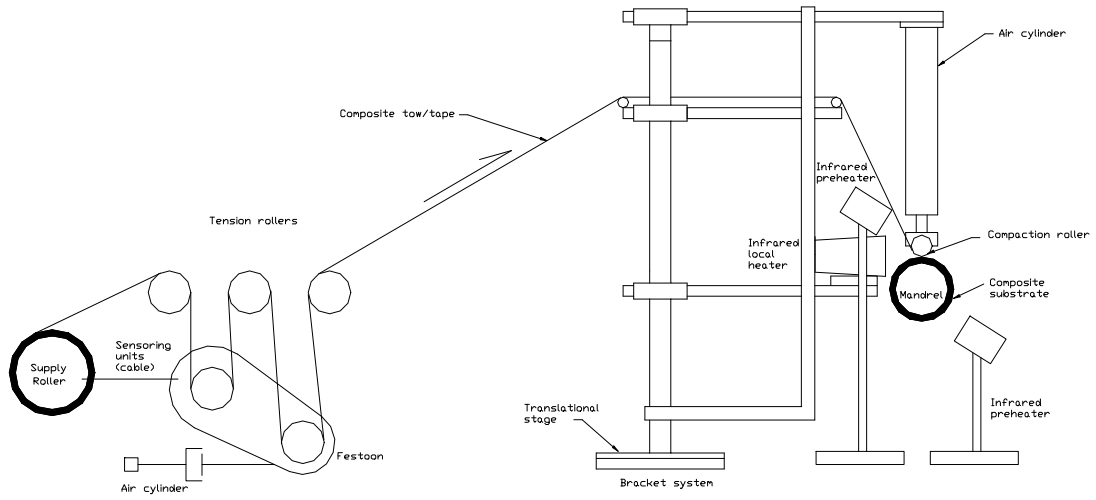


Figure MAF-56. Schematic of the In-situ Thermoplastic Filament Winding Set-up for the Scaled Pressure Vessel

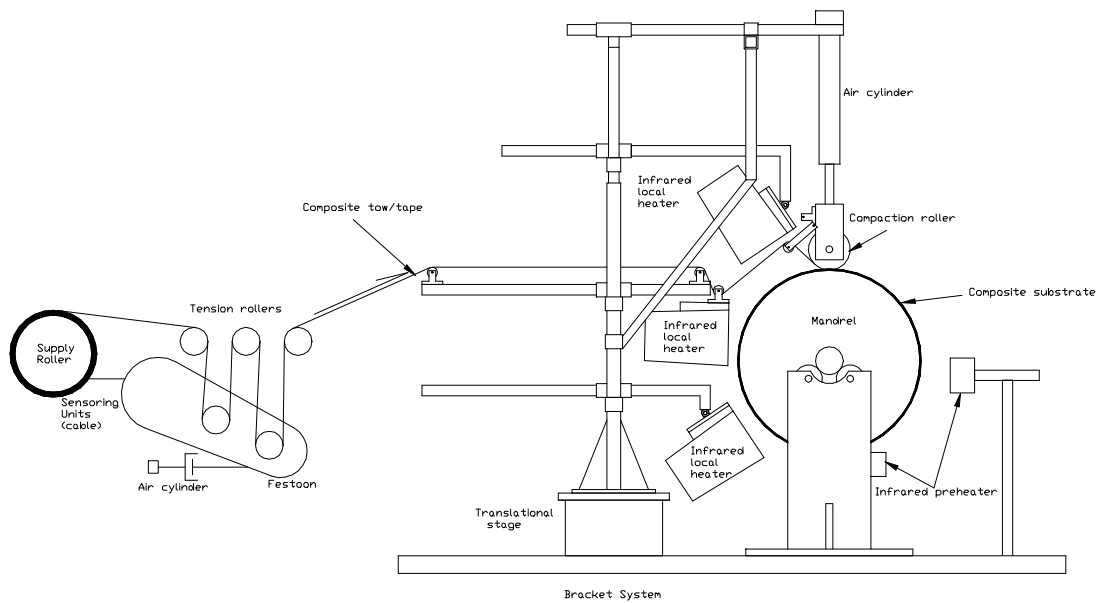


Figure MAF-57. Schematic of the In-situ Thermoplastic Filament Winding Set-up for the Main Pressure Vessel

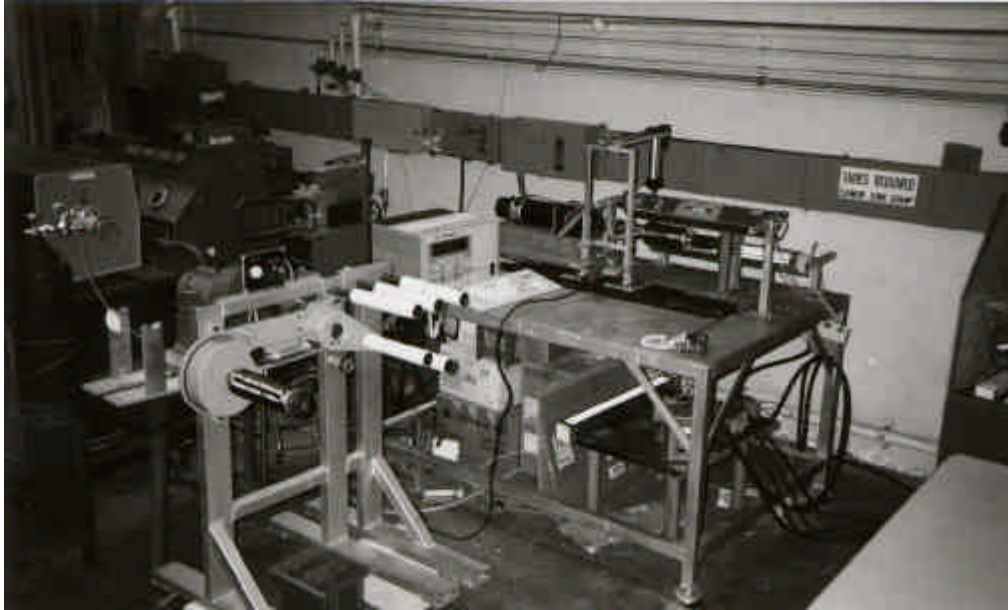


Figure MAF-58. Photograph of the Set-up for the Scaled Pressure Vessel

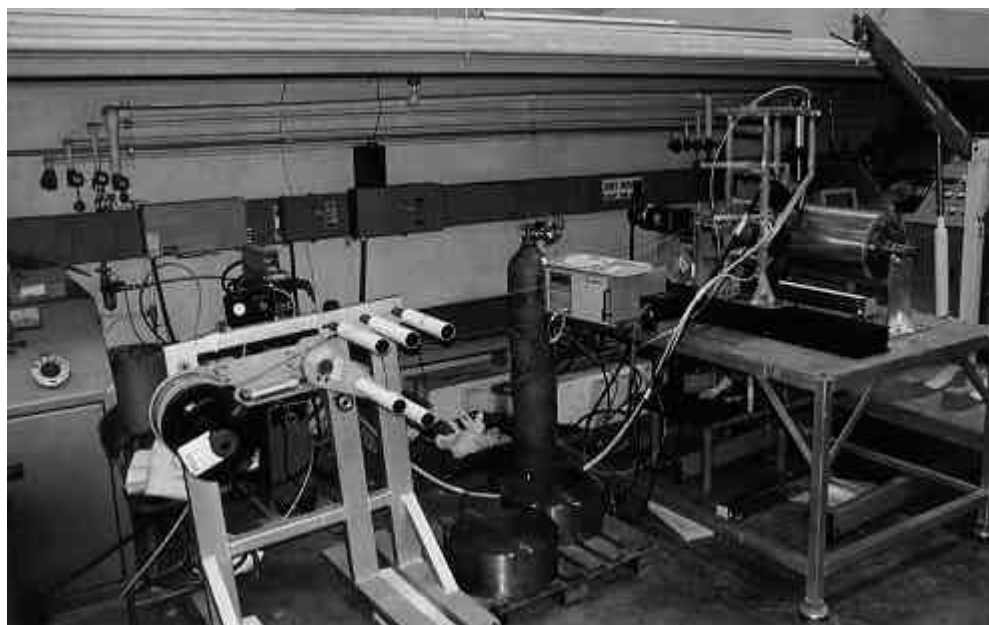


Figure MAF-59. Photograph of the Set-up for the Main Pressure Vessel

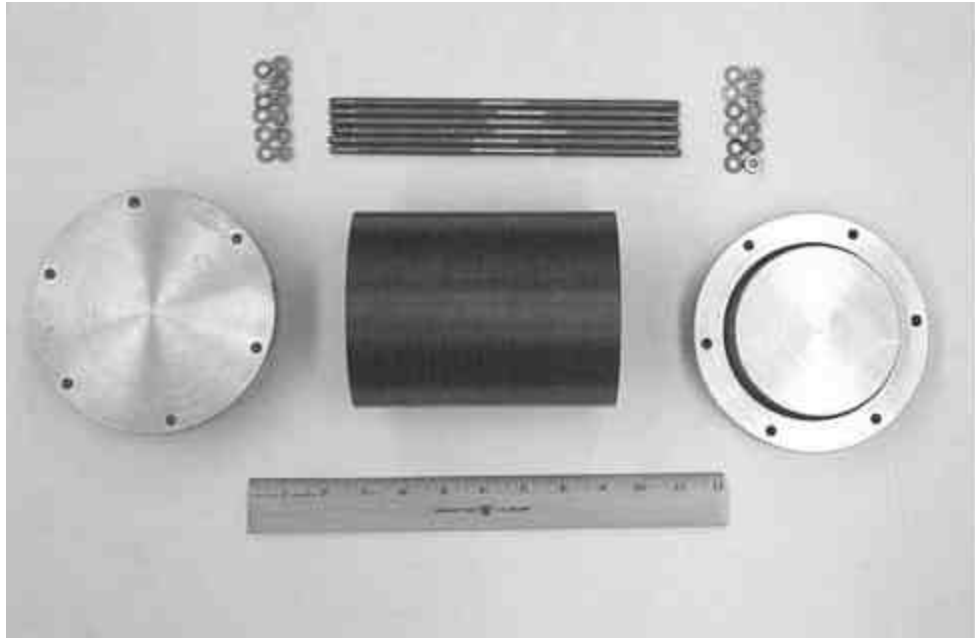


Figure MAF-60. Photograph of the Scaled Pressure Vessel with its End-caps and Tie Rods

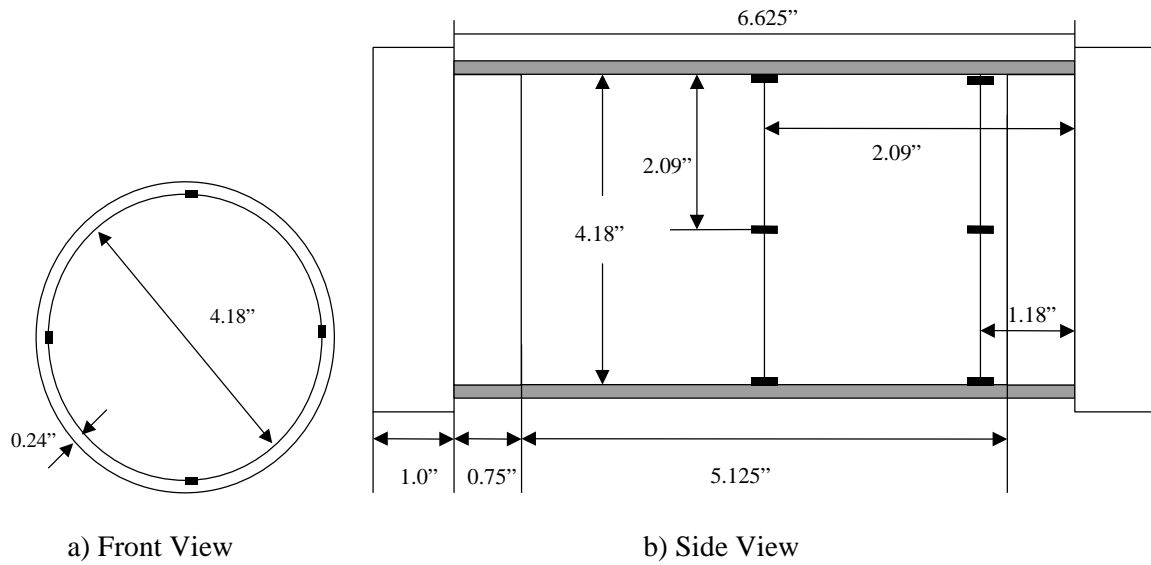


Figure MAF-61. Strain Gage Locations and Dimensions of the Scaled Pressure Vessel with End-caps in Place



Figure MAF-62. Photograph of the Scaled Pressure Vessel and the Connector



Figure MAF-63. Photograph of the Pressure Chamber End-cap and the Connector



Figure MAF-64. Photograph of the Scaled Pressure Vessel, Pressure Chamber, and the Cable



Figure MAF-65. Photograph of the Scaled Pressure Vessel after the Test

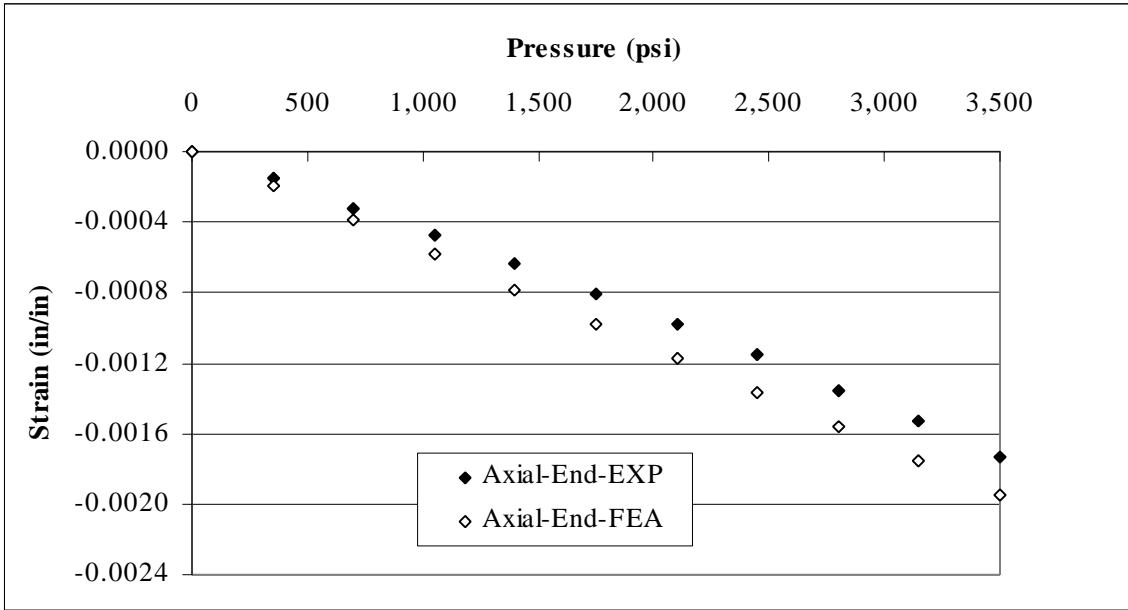


Figure MAF-66. Results of Axial Strain from Experiment and FEA Close to the End-cap

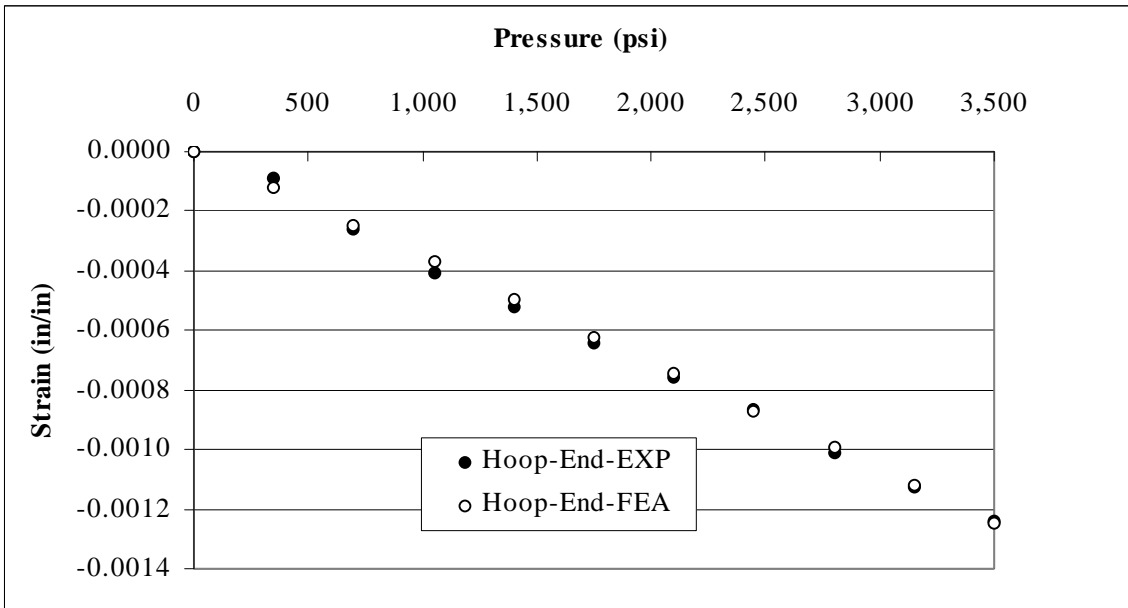


Figure MAF-67. Results of Hoop Strain from Experiment and FEA Close to the End-cap

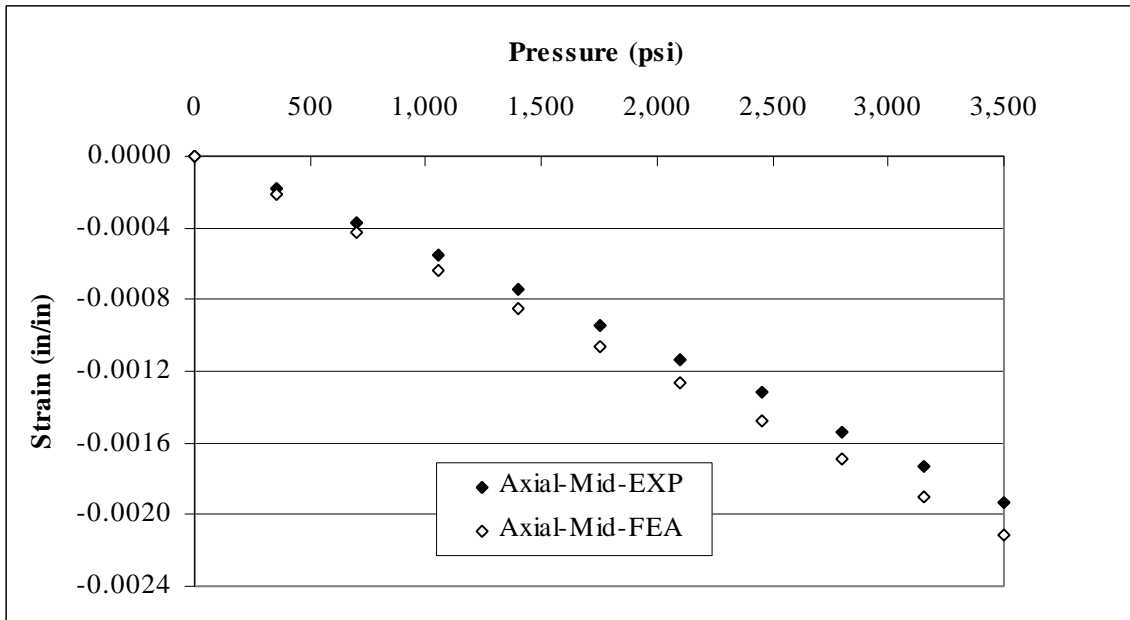


Figure MAF-68. Results of Axial Strain from Experiment and FEA at the Mid-length of the Scaled Pressure Vessel

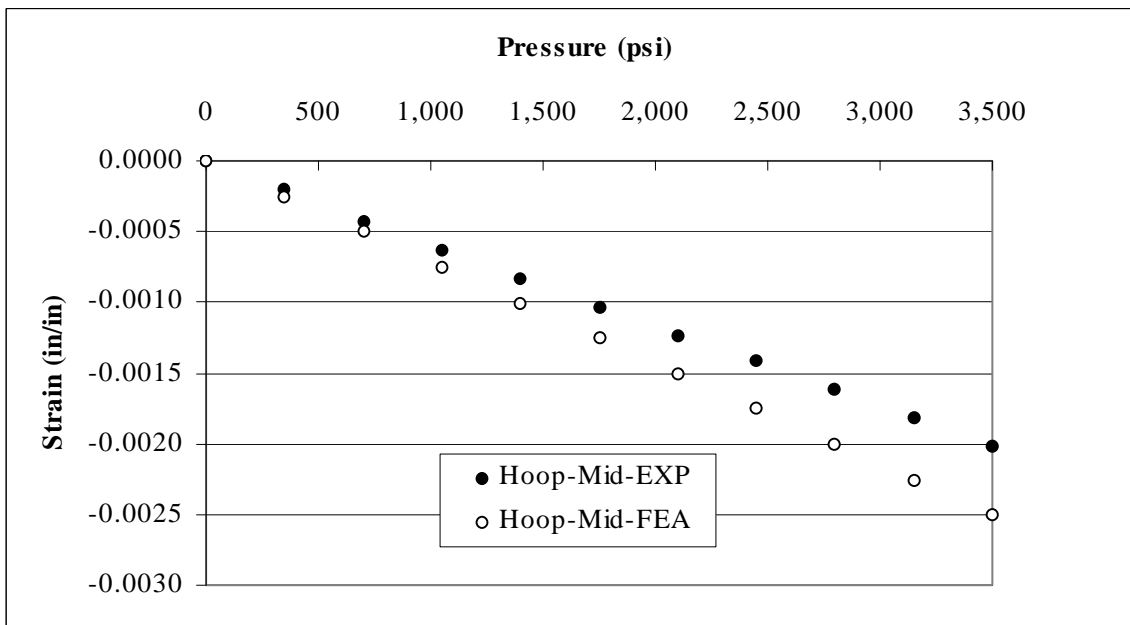


Figure MAF-69. Results of Hoop Strain from Experiment and FEA at the Mid-length of the Scaled Pressure Vessel

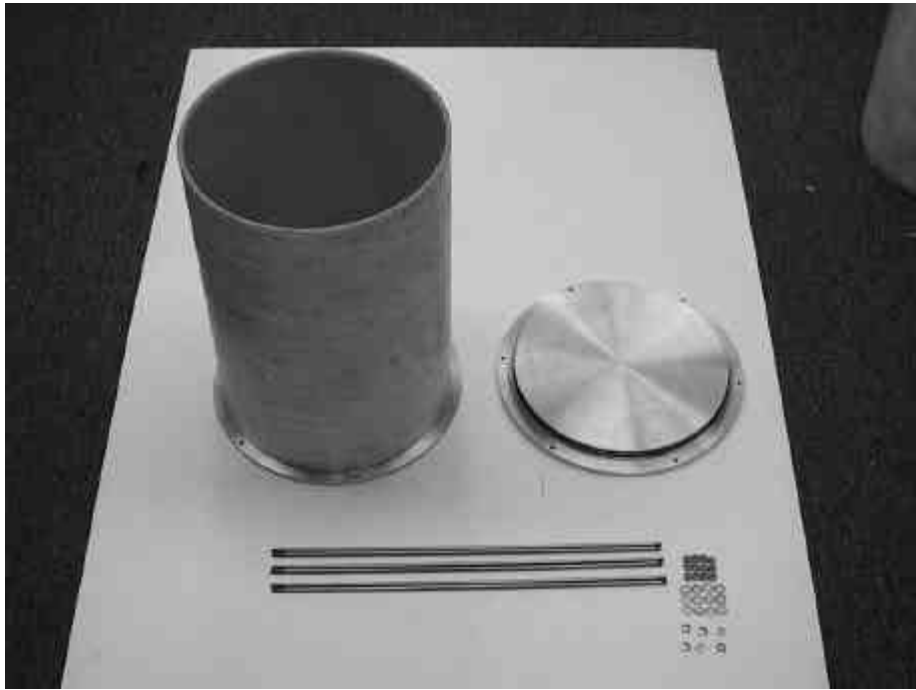


Figure MAF-70. Shallow Water Pressure Vessel with its End-caps and Tie Rods (Ng00a)

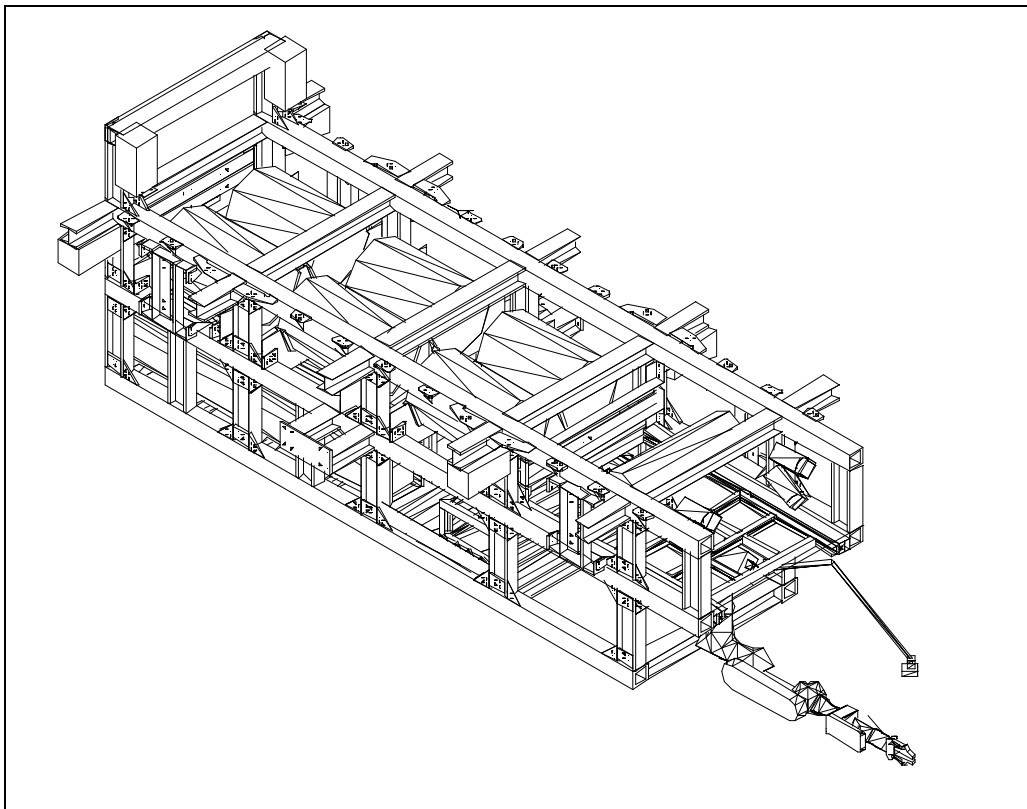


Figure MAF-71. Typical Component Layout of the Vehicle

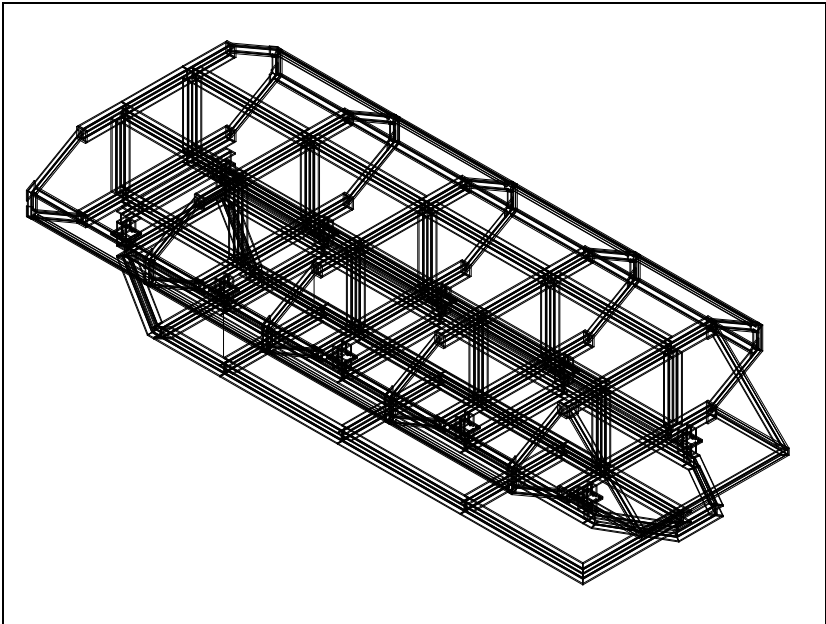


Figure MAF-72. Preliminary Design of the Frame

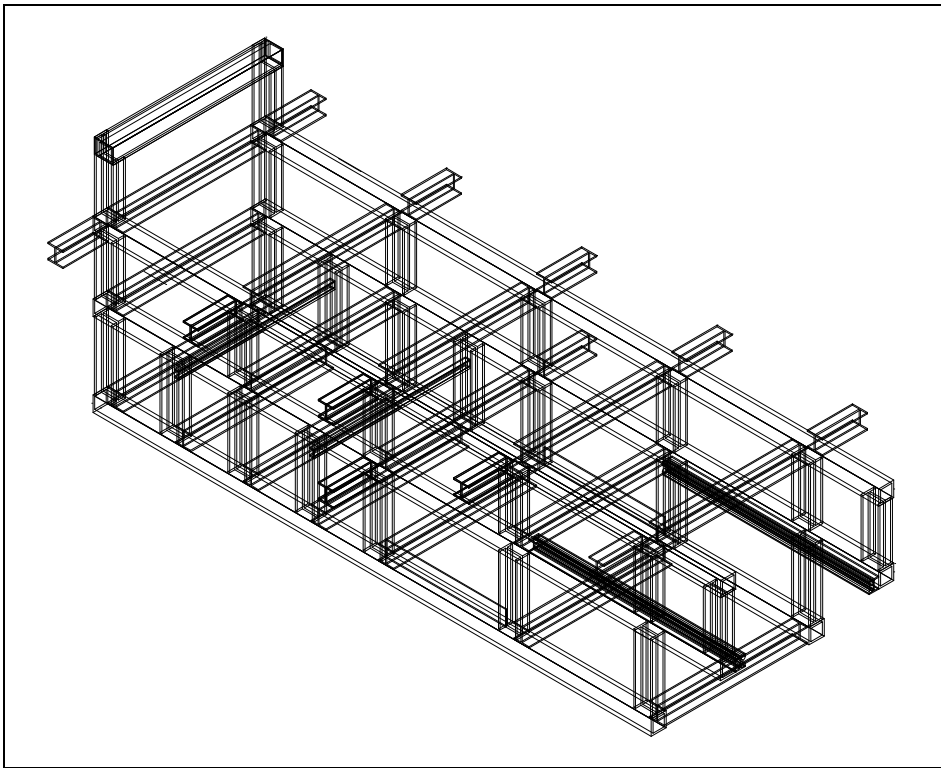


Figure MAF-73. Final Design of the Frame

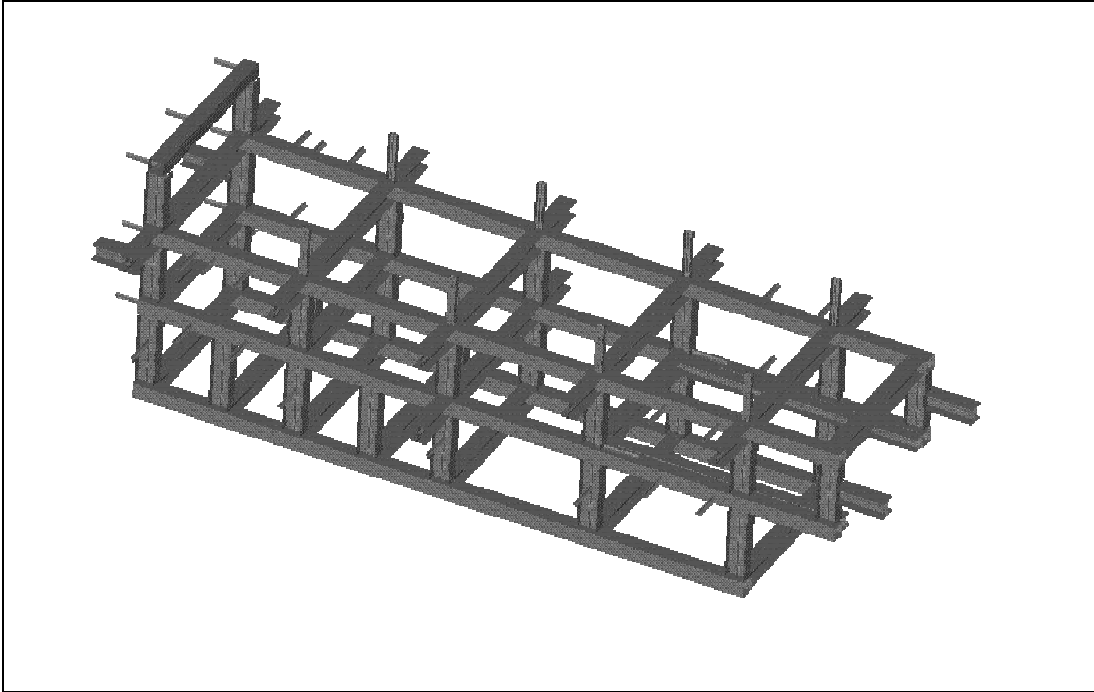


Figure MAF-74. Solid Model of the frame

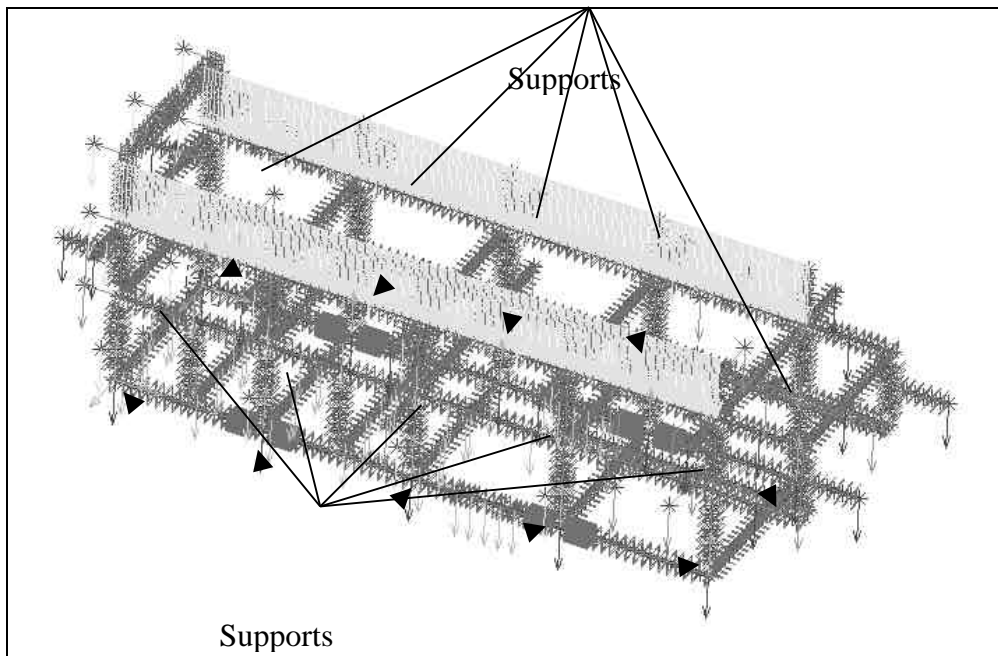


Figure MAF-75. Dry Boundary and Load Conditions with Arms Extended

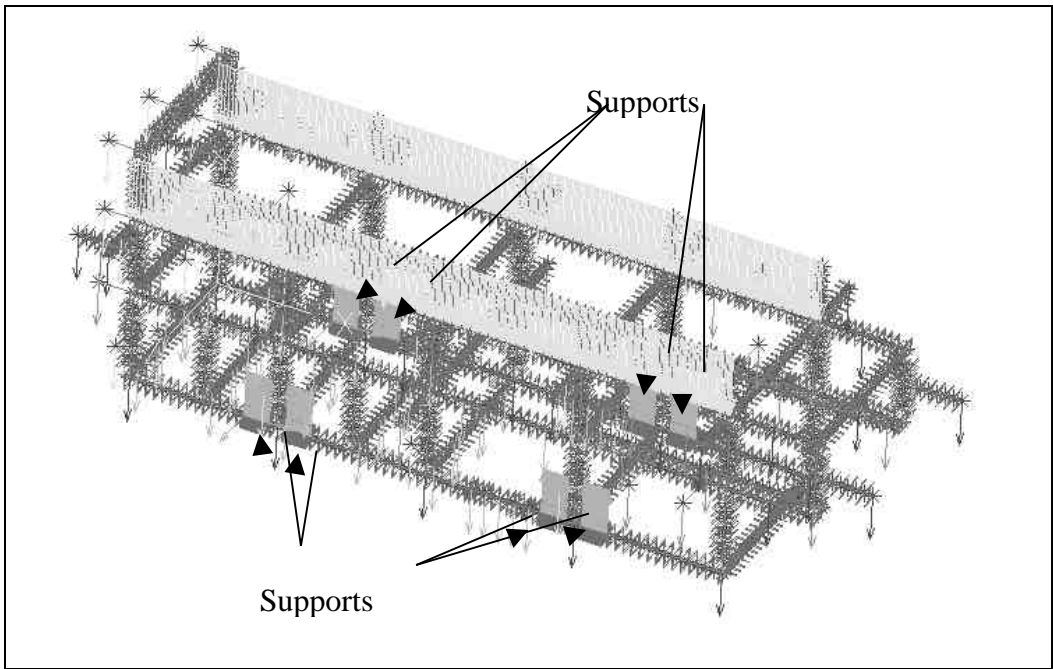


Figure MAF-76. Lifting Boundary and Load Conditions with Arms Extended

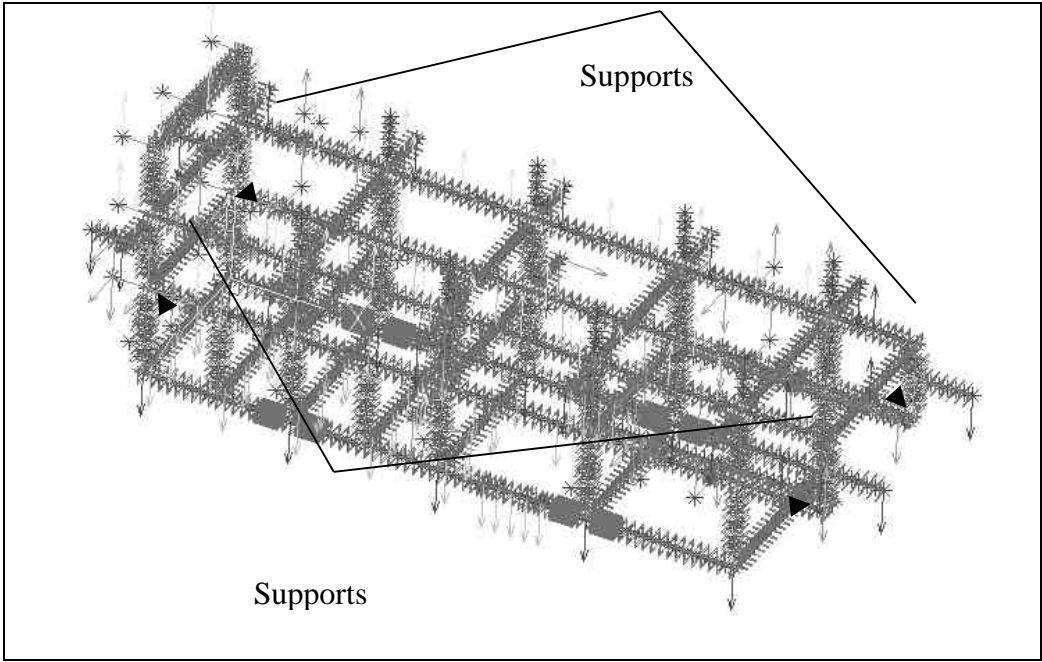


Figure MAF-77. Wet Boundary and Load Conditions with Arms Extended

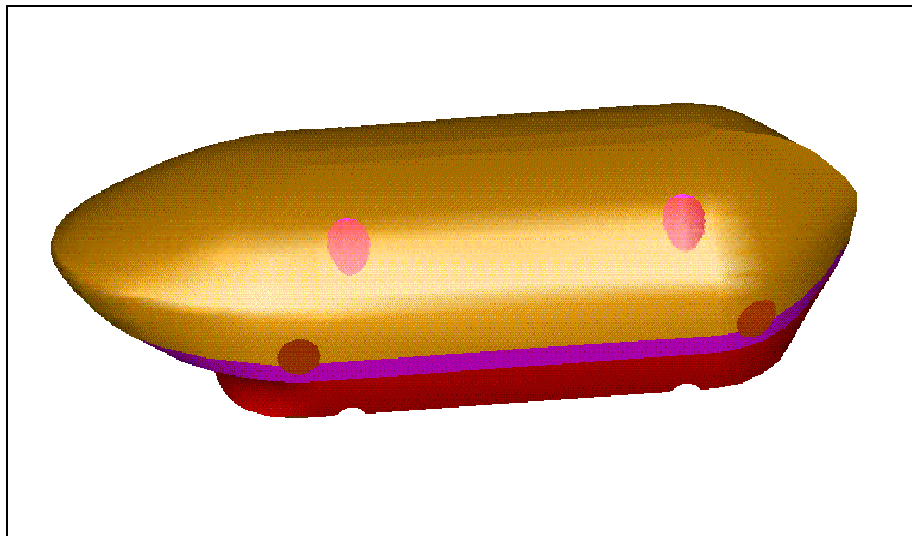


Figure MAF-78. Composite Flooded Fairing (Side/Isometric View)

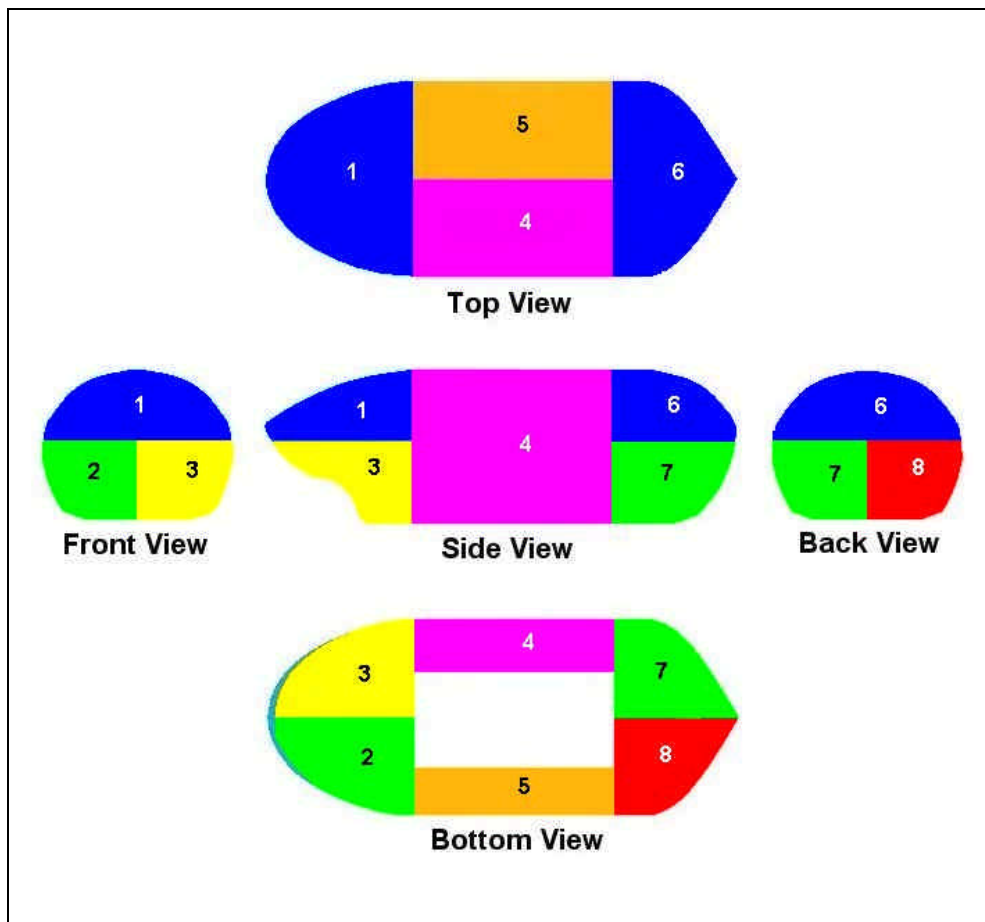


Figure MAF-79. Schematic of 8-Segmented Composite Flooded Fairing

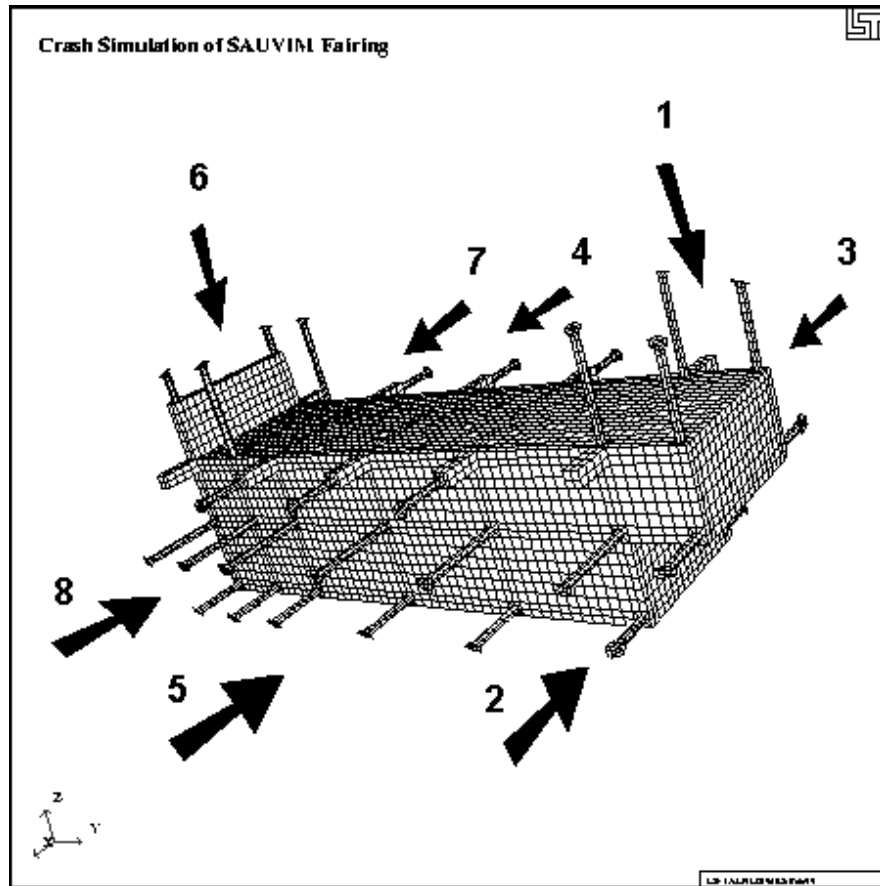


Figure MAF-80. Assembly Order of the Segmented Composite Flooded Fairing on the Meshed Block, Supports, Joints, and Disks used in LS-DYNA3D for the Crash Simulation

Mechanical-Electrical Design (MED)

Project Leader(s): Mr. Oliver T. Easterday, Mr. Michael E. West & Dr. Song K. Choi
Personnel: Mr. Kaikala Rosa, Mr. Dante Julian, Mr. Stacy Hanson & Ms. Elizabeth Shim
Past Project Leader(s): Dr. Curtis S. Ikehara, Dr. Junku Yuh & Mr. Gus Coutsourakis
Past Personnel: Mr. Lawrence Wong, Mr. Mark Fujita, Mr. Dicson Aggabao, Mr. Szu-Min Chang, Ms. Colleen Kaku, Mr. Mike Hall, Mr. Tai Blechta, Mr. Scott Sufak, Mr. Keith Sunderlin, Mr. Clyde Campos, Mr. Richard Antunes, Mr. John Lee, Mr. Scott Sufak, Mr. Daniel Shnidman, Mr. Weston Fujii & Mr. John Lemmond

Objectives

Integrate mechanical and electrical components of the SAUVIM vehicle and provide vehicle infrastructure in terms of structure and power to support research aspects of SAUVIM AUV.

Current Status (Tasks Completed During 8/1/97 – 6/30/02):

MED Group General Systems Status Overview - Mechanical/Electrical:

In this Phase the SAUVIM vehicle detail design, construction and initial static and dynamic testing has progressed. A general overview of the vehicle systems development, both mechanical and electrical, will proceed organized by sub-system. Sub-systems will generally proceed in the following order, based on the stage of development it is in: integration complete, fabrication complete, in-fabrication, entering fabrication/detailed design, design, conceptual design, problem specified/identified. The breakdown and separation between mechanical and electrical systems is not sharply divided so the reader is cautioned that some apparent overlap may occur. Last, a discussion of experience gleaned from the two shakedown tests to date closes to section.

The main emphasis during this time-period of the project has been focused on fabrication of components and vehicle assembly culminating in the first set of wet tests of the vehicle for purposes of establishing static and dynamic stability conducted in November 2001 and May 2002, respectively. Hence, this report summary section will focus mainly around photographs of hardware in various stages of construction with pertinent comments included. Design information and detail can be examined in previous SAUVIM project annual reports if reference is needed.

The reader is referred to the SAUVIM Report Phase-I dated for November 2000 for a discussion of the vehicles function and mission objectives in detail. For reference the system requirements evolved out of the following vehicle objectives as set down in the SAUVIM Vehicle Proposal: ONR Grant Application. Some of these constraints have had a major impact on the evolution of the vehicular systems supporting them and therefore in the evolution of the SAUVIM; these are highlighted underneath each requirement below:

The SAUVIM will seek to achieve the following performance specifications:

<u>Specification:</u>	<u>Phase-I/II-A design</u>	<u>Phase-II-B design</u>
Vehicle Class:	Mid-Depth Intervention AUV	Deep Intervention AUV
Size:	192"L x 95"W x 74"H un-faired 251"L x 95"W x 74"H faired	192"L x 95"W x 74"H un-faired 251"L x 95"W x 74"H faired
Form:	Free-flooded vehicle with fiberglass outer skin, aluminum 6061 structural frame, with six cylindrical vessels	- same -
Weight:	8,500 lbs (dry), -5 lbs (wet)	13,500 lbs (dry), -5 lbs (wet)
Power:	Lead acid batteries banked at 48, 72, and 144VDC on six buses. 12 batteries overall.	Lead acid batteries banked at 48, 72, and 144VDC on six buses. Approximately 70 batteries overall in two fiberglass PBOF enclosures.
Mission duration:	3-6 hours all systems active 24 hours with power rationing	15-18 hours all systems active 1 week or more with advanced power management added.
Powered Cruise Rnge:	2.7 nautical miles powered	8.1 nautical miles powered Unpowered glide slope range varies with depth.
Speed:	1 knot cruise, 2-1/2 knots maximum speed	
Ballasting:	Hard ballast: 1 main/emergency weight - 600lbs, two descent weights - 150 lbs/each, 1 ascent weight - 300 lbs, two metering weight canisters - 60 lbs each. Ballast is scrap iron plates bolted together, metering weights are iron/lead shot	
	Soft Ballast: two 36 gallon tanks (~300 lbs buoyancy each)	
	Floatation: Twelve pieces of foam, six Top foam. About 25% of foam space occupied, ie ~18.5 lbs/ft ³ density	Syntactic foam blocks of same shape as shallow water foam. Foam space is fully occupied, ie ~40 lbs/ft ³ density.
	Pressure vessels: Six E-glass thermo set composite epoxy thermoplastic composite cylindrical, 13"ID x 18" cavity bottles with aluminum lids	Six carbon filament/PEEK bottles with titanium lids 13"ID x 18" cavity
Vehicle Control:	Six Tecnadyne Model 1020 thrusters, two Tecnadyne Model 2010 thrusters. Three articulating fins of about 3ft ² apiece. Main ballast is on a tray with 18.0" stroke allowing trim from 8° nose up to 12° nose down.	
Deployment:	One standard (20') shipping container for vehicle, support equipment and control consoles.	
Command Control	RS-232 and RJ-45 Ethernet tether	Acoustic modem link. Location

	with wireless LAN & RF modem link. Simulated degradation for deep water mission simulation.	transponder, pinger and acoustic ballast release under review.
Robotic Arms:	One 7 joint (d.o.f.) Electrical oil-filled manipulator capable of 21 lbs payload at full extension. One 7 d.o.f. passive arm used for position sensing. Arms are on a deployable platform tray.	
Computer/Control:	Twin-VME backplanes with twin Motorola 68060CPUs linked via RJ-45.	
System Software:	VxWorks RTOS. C-code drivers. Custom high-level interpreted language used for mission planning and reporting.	

As can be seen the integration of all the research systems onto the prototype vehicle involves floatation, ballast, thrust/fins, trim, sensors, cabling, structural frame, and numerous other systems. The breakdown of systems on the vehicle to accomplish these objectives is detailed in Figure MED-1. As can be seen this entails

To obtain a visual overview of the maturing vehicle design, the reader can examine Figure MED-2, which is a conceptual drawing of the complete Phase-II, deep-ocean SAUVIM vehicle complete with the outer fairing. Figure MED-3 shows a 1:12 scale model of the faired vehicle that was made for tank testing for estimating drag coefficient estimates as well as establishing hydrodynamic stability. Figure MED-4 shows the vehicle in an oblique view with all the main components labeled, with the fairing and floatation foam removed for clarity. Figure MED-5 is a photograph of the SAUVIM in a recent state with some of the major assemblies mounted onto the vehicle frame for fitting and testing purposes.

Mechanically, the vehicle is reaching the stage of advanced integration, where many of the major assemblies are being mounted onto and fit as part of the larger vehicle, among the systems at a functional stage of completion or nearing it are: floatation foam, pressure vessels, thrusters, computer, power distribution and battery systems. Entering the stage of being integrated at the present time are the sensor systems, ballast/trim system, and arm computer. The detail design of the vehicle fairing and arm wiring systems are underway.

Meanwhile, the overall electrical system is maturing at this point. Based on the control hierarchy developed by the RDC group, the subsequent breakdown and assignment of functions into the various pressure vessels onboard the SAUVIM has evolved into the electrical system depicted in Figure MED-6. The cabling interconnects between all the pressure vessels and the sensors, actuators and other outboard components are detailed in Figures MED-5 and MED-6. Many of these cables have been fitted onto the vehicle - these include the main data interconnect, switching/control and power feed cables. The SAUVIM presently is in various stages of electrical fabrication, depending on the system. There are custom-built as well as off-the-shelf components being utilized used on the vehicle. Electrical design and layout is performed using PC-based CAD software.

The subsystems will be discussed in detail below proceeding in a general progression based on the degree of completion, with mechanical systems covered first and then electrical. The current state of each system and the next tasks to be completed for it will also be addressed in the next several months for the system in question.

Mechanical Systems – Much progress has occurred on the mechanical side in the past months. Integration of all the major components, save the fairing, onto the frame is complete. The main focus is now on mounting minor components, like sensors and accessories onto the frame as well as finishing out detail design and materials procurement for the fairing. { TC \l3 " }

Frame: The vehicle frame fabrication was completed in 1998 and it is a free-flooded aluminum 6061 alloy structural frame welded with 4043 filler rod. Figure MED-7 shows the frame largely bare of systems components. Design of the frame was performed in-house by the MED group, while analysis for structural strength was performed by the MAF group. The contractor that was involved with the fabrication of the frame was Honolulu Shipyard International (HSI), a subsidiary of Pacific Marine Co.

As can be seen by the CAD drawing in Figure MED-8, the frame weights around 1,200 lbs (dry) and measures around 167” long, 70” wide and 63” high. Loaded with components the SAUVIM should fit into a standard shipping container with around 3” clearance on each side.

One of the first SAUVIM components to be completed, the main activities that have occurred on the frame recently are the following: 1) installation of passive anodic protection, 2) investigation of protective barrier coatings, 3) installation of spray/wash-down system fittings, and 4) work performed for installation of mount-points for various hardware.

It is clear that aluminum is a material prone to corrosion in seawater based on in-house testing of samples, literature, as well as based on experience from some SAUVIM team members with marine hardware in the past. Consultation with other research teams and literature has led us to install commercial marine zinc sacrificial anodes onto the SAUVIM frame to minimize structurally compromising corrosion on the SAUVIM frame. Use of passive anodic protection for bare 6000-series aluminum seems well established. Figure 9, depicts a typical installation of these of which there are eight, evenly distributed around the frame.

A survey of other vehicles with a similar corrosion profile was undertaken in the past with the idea of potentially employing barrier coatings in the form of marine epoxy paints. Literature reviews, past in-house corrosion tests and a survey of research teams with similar aluminum 6000-series alloy vehicle frames, most notably the ARGO team of Wood’s Hole Oceanographic Institute as well as University of Hawaii’s HMRG-II tow-fish vehicle, established that barrier coatings accomplished little other than preserving aesthetics with a corresponding increase in maintenance efforts. Indeed scratches in the coating can concentrate corrosion in spots increasing the chances for structurally significant corrosion as opposed to dispersed pitting which is merely cosmetic in nature. So the SAUVIM frame has been left bare. Minimization of corrosion on the SAUVIM, which appears to be working after two test missions to date, includes the following: utilization of passive anodic protection with zinc, limiting hardware contacting the frame to either 304-, 316- stainless or nonmetallic fasteners, use of water excluding grease (Aqualube “green grease”) on tapped threads and many crevice regions, and liberal wash-down upon recovery.

As to the wash-down process, the SAUVIM has been equipped with a single-point hose barb attachment point for the dual purpose of cooling the bottles during dry operation as well as flushing residual seawater off the frame and components after recovery. This system seemed to work well after the second test. The composite photograph in Figure 10 shows the features of the system, which is merely perforated soaker hose routed around the upper box beam set of the SAUVIM as well as through the interior of the box beams.

The last work on the frame has been minor additions/subtractions to it including a forward instrument rack. For the next several months, additional C-channel framework will be added to the frame to allow for structural underpinning of the fairing, to be added this autumn.

Pressure Vessels: The pressure vessels on the SAUVIM for shallow water testing have been modified from the E-glass thermo-set epoxy group, shown here in Figure MED-11, to Aluminum-6061 machined housings. This was based on cycle- and use testing completed in the February to March timeframe of 2001. Though the pressure vessels could sustain and resist failure due to material strain failure as well as first-mode buckling failure, the usability tests established that the composite vessels were unable to seal reliably due to some design features that eased fabrication difficulties but that compromised the ability of the bottles to seal reliably. On average, statistically the bottles had around a 20% chance of significant leakage; well below what can be tolerated for critical systemic reliability, as leakage is pretty much a mission abort failure. Also, much shock and pounding was required to close and open the bottles and this would compromise the integrity of any delicate instrumentation and electronics to be loaded into the bottles. Figure MED-12 shows one aspect of the testing process, which was a vacuum leak test to establish positive sealing. Detection of leaks was done via a self contained electronic conductivity switches that were sensitive to moisture as well as by water soluble dye lines in paper that were mounted in the bottle. A typical flooding result is depicted in the photograph in Figure MED-13. Massive dissolution and migration of the dye streaks has resulted indicating that a quantity of water sufficient to damage the contained electronic systems has made ingress into the bottle. { TC \12 "}

Possible solutions were looked at; one was a high-risk path that included modifying or re-fabricating a new set of pressure vessels from composites to correct the perceived problems. The chosen solution was the lower-risk one of employing conventional metallic vessels with the idea of revisiting composites with a retrofit bottle for experimentation with non-critical components on the SAUVIM vehicle at a later time.

Initially considered was a design to be made at the UH Marine Center that was designed in-house. Figures MED-14 and MED-15 detail the CAD layout of this design. The depth rating was to be approximately 4000 feet. The material selected was 6061-series aluminum alloy, chosen for ease of fabrication, with a Type-II mil-spec hard anodizing coating to control corrosion. Features of these bottles included a double-radial seal along with an axial seal. Fabrication time, including the obtaining of the material stock was to have been around 5 months, testing was estimated to be another month or so. Given that this was in May of 2001 and the first wet test was slated for December of the year, a third option was explored.

Two companies offering completed pressure vessels were investigated, and were found to be desirable on the basis that design, materials procurement, fabrication, anodizing and testing were all included in the bid amounts making this option only modestly more costly than doing them in-house, but with less risk. These bottles were "turn-key" in nature include certification testing for loading and sealing integrity. Figure MED-16 is a photograph of one of the six bottles manufactured by Prevco, Inc. for the project. The bottles were received in September and cleared up previous problems that were encountered with the composite vessels. Figure MED-17 depicts the bottles under fabrication at the vendors and Figure MED-18 is of the set of six that are being worked with.

The pressure vessels are pretty much complete and integrated onto the vehicle. However, the option to replace one of the bottles, most likely PV#2, which houses the arm motor and joint encoder controllers, will likely be considered at a later date. The depth limit of these bottles was similar to the

UH design, which was a design depth of around 3,700 ft. They were pressure proof-tested to around 4,100 feet. The ultimate failure depth of these bottles which, because of their amply supported endcaps and relatively thick walls, is elastic material failure, was calculated to be in excess of 5,000ft for both the UH design as well as the commodity vendor bottles. This was the option that selected due to the complete system.

Saddle modifications were required to be performed on the vehicle as the pressure vessels had a substantially increased wall thickness while retaining the same interior cavity volume size. This included rerouting of the polyethylene saddles to accommodate the larger outer wall diameter of the bottles as well as adding in aluminum clamps to clamp the substantially heavier bottles. Of course the weight increase also necessitated adjustments in the floatation configuration on the SAUVIM as the empty submerged bottles went from about 30lbs positive buoyancy to about 50lbs negative. A reconfigured saddle is shown in Figure MED-19. The same basic system of locking the pressure vessels down with plastic clamps to provide vibration compliant material via tie rods has been retained.

Floatation: The floatation for the vehicle is largely complete and ready for the shallow water testing that is getting underway on a regular schedule. Floatation foam for the SAUVIM is being acquired in two stages. The first set of foam, currently in use on SAUVIM, is rigid polyurethane foam manufactured (R-3315) by General Plastics Manufacturing Co. of Tacoma WA, is a 300psi rated polyurethane foam with a density of 15.0 lb/ft³. This foam will undergo a 5% volumetric compression at 300psi (with elastic recovery), and will undergo about a 5% weight gain if exposed to 250psi water for a period of 40-50 days, which is a logarithmic absorption limit. SAUVIM carries enough foam to overcome this effect, however, the foam has been barrier coated with a thin layer of brush-on 2-part marine epoxy resin. This will yield an effective depth limit of around 700 feet during testing with this floatation.

Initial shaping was done by use of mockups made out of insulation foam, depicted here in Figure MED-20. These mockups will also be employed as molds for conformal blow tanks as well as in the fairing male mold plug fabrication.

Both the top (dorsal) foam and side foam have been shaped and cut; the top foam has been mounted onto the frame as can be seen from Figure MED-21. Inspection of Figures MED-21 and MED-22, which is a top view of all of the dorsal foam attached, will reveal that both the top and side foam pieces are of a constant cross-section, this serves multiple purposes: first, this will allow for use of the extra shallow water foam as a set of male molds for the vendors who will fabricate the syntactic foam, second, during shallow water trials vehicle trim can not only adjusted by shifting ballast fore and aft, but also by doing the same to sections of the floatation foam. As can be seen from Figure MED-22, much void space remains between the foam spaces - this is due to two reasons: 1) the shallow water foam has about 2.2 times as much buoyancy as the deep water on a volumetric basis. 2) The pressure vessel hardware for the deepwater vehicle will be heavier due to the upgraded bottle walls and lids. These void spaces will be filled in by widening of the foam blocks for the deep-ocean foam, to recover loss of volumetric efficiency. Indeed installation of additional foam low and on the sides of the vehicle will be necessary. This will have the fortunate effect of lowering the vehicular center of volume (COV) closer to the center of mass (COM) and make the vehicle more pitch sensitive. This will allow for it to dive to the bottom and recover on a pitch glide-slope by dropping the forward and main ballast masses respectively.

The second set of foam will be syntactic foam comprised of glass micro-spheres with an epoxy binder these will have a density of around 40 lbs/ft³. Figure MED-23 depicts the deepwater foam, as it will be mounted onto the frame.

A series of likely vendors has been identified for these foam blocks. Extra shallow water foam blocks will be sent to the vendors after completion of the initial shallow water trials and these will serve as male molds for the chosen vendor to form the syntactic block out of. Much more of the foam space volume will be occupied by the syntactic floatation.

Attention was given to the location and sizing of the floatation pieces to ensure vehicle stability and the proper range of pitch and roll sensitivity/stability. The method by which this was tracked is covered under AVehicle Statics@. For the next several months there will not be much done on the floatation of the SAUVIM as it is stable for shallow water testing.

Thrusters: Primary activities on the thrusters has focused on fitting the vendor units, six Tecnadyn 1020 units and two larger 2010 units, into the tube ducts and mounting the ducts, in-turn, onto the frame. The larger 2010 thrusters are rated at around 130/80 lbf (forward/aft) of thrust static each while the smaller 1020 yield around 47/32 lbf static in the forward direction. Actual testing in the pool has confirmed that these values are about 20-25% under the actual output. A typical installation is shown in Figure MED-24. This is a view of the thruster mounted within its tube.

All eight of the thrusters have been mounted to the SAUVIM frame, Figure MED-25, shows the rear-lateral thruster tube mounted to the frame, sans the motor and propeller. Inside the tube all the strut and clamp assemblies are completed and the thrusters have been mounted into the tubes. Figure MED-26 illustrates how these one of these assemblies look prior to installation into a thruster duct. It is a two-piece clamp that grips the thruster body with four struts that run out to the tube.

It is planned to retrofit constant pitch, bi-directional propellers onto the thruster hubs, this will be deferred till after the Arm Computer and manipulator are integrated onto the SAUVIM vehicle.

The thrusters are wired up and running as of this date. No immediate activities are planned on these for the next several months.

{ TC \12 '' }

Fairing: This vehicle component is under detailed design at this stage. Completion of detailed design will be accomplished by mid-August of 2002.

Figure MED-27 is a CAD rendition of the fairing, this was an advanced conceptual drawing that was envisioned. Detailed design of the shallow water fairing, which is planned to be fabricated from E-glass fiber and polyester resin will take place thru the remainder of summer 2002 with fabrication taking place in early to mid fall 2002.

Fins: The fins, bearing structures and power canister detail designs are complete. An isometric view of the location of one of the three fins on the vehicle is shown here in Figure MED-28. { TC \12 " }

Fabrication of the power canisters is underway using the design shown here in Figure MED-29 using an enclosed bi-polar hybrid stepper motor with a planetary gear head with a 1:100 step-down ratio. Figure MED-30, is a photograph of the electrical/electronic components that are within the system. This consists of the motor/gearhead, and housed in the Arm Control computer pressure vessel the motor controller and micro-controller circuitry to drive and pulse the unit. { TC \12 " }

For reasons of standardization the same motor-gearhead will also be utilized to move both the main ballast and arm trays. The mounts, gearing, and shafting to accomplish this is in the detail design and procurement phase. { TC \12 " }

Hard Ballast{ TC \12 " }

Hardballast mounts have been added to the vehicle, these will later become active components upon employment of the SAUVIM in deep-water tests. Figure MED-31 is a photo showing the starboard forward ballast mount in place on the vehicle frame. Preparation of scrap iron plate ballast for the main, ascent, and descent weights is complete. Integration of release mechanisms will be the next step for all of the ballast racks. This is a step that can and will be deferred till initial simulated deep-ocean mission trials.

Detailed design of the release mechanism borrows heavily from the design employed on the *Alvin* submersible. Depicted in Figure MED-32, it is a solenoid-actuated cam release. This release is planned for all four ballast pieces, the two descent, weights, the ascent weight, as well as the main ballast weight.

The main ballast tray has been also installed onto the SAUVIM as well. Currently empty of ballast, it will, like the forward ballast mountings be loaded with ballast upon launching into a deep-water mission trial. The wheels in the design were found to introduce jamming and excessive motion problems and had to be modified in design. They have been replaced with a sliding skid and rail assembly, much like that of the arm tray. Figure MED-33 shows a view of the tray with the redone skid rail assembly visible to the left of the tray.

Detail design of the main ballast tray translation mechanism is underway along with procurement of the mechanical assemblies needed for such. Figure MED-34 is a CAD design overview drawing of the mechanism that will accomplish tray translation. Basically is a twin static lead screw assembly on each side. Two acme nuts enclosed within a thrust plate assembly welded to the ballast tray are spun to advance/retract the ballast tray and accomplish pitch trim, which will be important in setting the glide-slope for deep water missions.

{ TC \12 " }

Soft Ballast: Based on dynamic stability testing within SNUG Harbor last April this system is being added onto the vehicle for the shallow water testing phase. The dropping of iron ballast within the testing location of Ke'ahi lagoon is of dubious legality, so for achieving fine buoyancy trim twin PVC tanks are planned to be added to allow for fine buoyancy trim while SAUVIM is performing shallow water trials. The components and locations for these tanks have been mostly ordered and identified respectively. { TC \12 " }

Figure MED-35 shows some of the main components that will be used in this system. As can be seen the system runs off of compressed air from a conventional SCUBA tank and first-stage regulator, in the lower right corner of the picture. The tanks will be PVC pipes with endcaps of 12.75" diameter by 68" long. Open on the bottom with vent holes the tanks each have about 37.5 gallons capacity, making for a buoyancy of 320 lbs per tank. Supplied at 150psi, the air will be routed thru a stainless steel solenoid valve and coil assembly (third and fourth items on towel from the left), one for each tank. The outlet high in the tank will blow water from the tank. A second pair of low-differential pressure plastic solenoid valves will allow air purge from the tank. Located high on the tank these, will allow for free-flooding of the tanks. One tank will be located forward and one aft, these will allow for the vehicle to also be pitch trimmed while completing shallow water tests.

Integration of these assemblies is currently underway.

Vehicle Support Equipment: Much of this equipment has been produced in reaction to challenges both anticipated as well as unforeseen with the handling and care of the vehicle. Among these items have included: vehicle launch, recovery frame, pressure testing facility, deployment trailer, pressure vessel racks and slings, battery carts, electronics testing racks and testing/tuning tank. Some brief comments on the larger systems are included in this section.

SNUG Facility – All of the mechanical and a significant portion of the electrical fabrication and integration of the SAUVIM is now taking place at the UH Marine facility at SNUG Harbor. This space was almost bare as of the projects transfer to this locale in March 2001. Erected was an climate and dust controlled shack for working on electronics complete with a storage loft above. 3-phase 220VAC and additional 110VAC was run throughout the location to support fabrication activities. Network cabling and conduit was run in anticipation of running the SAUVIM in Predictive Virtual Environment (PVE) and Multiple Vehicle Systems (MVS) projects in late 2002/early 2003. This facility outfitted at minimal cost by working with donated expertise and project personnel time has allowed for a great accelerated pace of vehicle development over the past 12 months. Figure 36 shows the layout of the facility at SNUG Harbor.

In a similar theme, the workspace has been equipped with machine tooling to assist in vehicle fabrication. Figure MED-37 shows the inventory that has been obtained for the vehicle fabrication. Proceeding from upper left corner of Figure MED-37 around the tooling includes the following:

- Bridgeport Series-I 4 axis vertical mill, acquired by internal UH equipment transfer at no cost to project funds. Equipped with digital readout indicator.
- Cincinnati Tooling 10 inch swing engine lathe, acquired by internal UH equipment transfer at no cost to project funds
- Anode table – Steel jig and fabrication table, acquired from scrap equipment at the yard.
- Generic Mill-Drill Machine - acquired with project funds. Installed digital readout indicator.
- Shoptask 3-in-1 Machine - acquired with project funds. Equipped with digital readout and elementary CNC capability.
- Miller Welder – this was transferred from Holmes 140A. Equipped to weld 6000-series aluminum, stainless steel, and carbon steel.

Some in-house fabrication of assemblies for the vehicle will decidedly benefit from this equipment. Also a productive and informal expertise- and equipment-sharing arrangement has been formed with two neighboring projects. These are the Hawaii Undersea Research Laboratory (HURL), primarily funded by NOAA that supports manned submarine (Pisces IV and V) and ROV (RCV-150) operations at sea, and the Hawaii Mapping Research Group (HMRG), which performs near-shore and deep ocean acoustic surveys and mapping with tow-fish arrays and software developed in-house as well as commodity based.

In this area, the next priority is to arrange donation of a standard shipping container to move project stock and supplies into to ease crowding in the shop area. This is planned for late summer.

Also some identified support equipment will be acquired among these are:

- N2 gas bottle and cart – for establishing inert and dry air within the pressure vessels upon completion of vacuum testing of bottles prior to launch.
- Plasma cutter – for time savings and greater ease in fabricating assemblies for the SAUVIM structure.
- CNC software – for transfer of CAD solids into the CNC machine tooling for fabrication of brackets, small pressure housings and the like in-house.

Launch/Recovery Frame – The steel SAUVIM launch recovery frame is completed and in-use. Pictures of it can be seen in the testing section. Completion of the frame included corrosion coating of the unit with asphalt paint.

Pressure Testing facility – The miniature pressure test facility has been moved from Holmes 140A to SNUG Harbor to assist in characterizing materials and certifying depth capabilities of various SAUVIM components. Depicted in Figure MED-38 and MED-39, it is a modified 150 ton jack that is used for testing foam for moisture ingress under static pressure, electronics resistance to pressure, minor camera and light housings and the like. The certification of components is an ongoing process.

Trailer – Funding for a trailer for the vehicle was becoming problematic. No commodity or used trailers meet the needs of load capacity, ease of fitting into the lab (lack of bulk), and budget that was set. A suggestion from one of our welding and machining part-time personnel was to use an aircraft dolly, modified for the SAUVIM. This solution was perfect. An old dolly was donated from Air Services Hawaii, Corp. to the project, under educational tax write-off provisions. Figure MED-40 depicts one of these dollies. Modification to the SAUVIM involved cutting the dolly longitudinally while eliminating the center third of the dolly width, cutting the dolly across the length lengthening the unit by about three feet, adding steel decking (except under the ballast area to facilitate loading), and reinforcing the unit with bracing for the anticipated load. Additionally the entire dolly was corrosion barrier coated with asphalt paint and the bearings were replaced with heavy-duty units in anticipation of increased load.

Bottle Racks and Sling – Handling of the fully loaded pressure vessels for safety reasons necessitated fabrication of a sling and rolling racks to handle the pressure vessels in a safe fashion. The sling is an aluminum bar equipped with nylon straps to load and extract bottles from the vehicle frame using a standard gantry crane. The cart is a castor-equipped unit with saddles to allow safe and easy opening of the pressure vessels as well as for transport around the lab.

Battery Carts – A set of carts was made to allow for easy loading/unloading of batteries onto the vehicle as well as moving them around the lab. Additionally, these carts serve to allow easy handling of the pressure vessel contained electronics equipment when being worked on in the lab.

Test Tank – obtained from the robotics arm group where it was in use in the PUMA arm setup this tank serves duty for tuning of the SAUVIM thrusters and controllers as well as for leak proof testing of various assemblies due for integration onto the vehicle. Depicted in Figure MED-42, the tank here is being utilized for tuning of the SAUVIM longitudinal thrusters.

Other support equipment has been acquired for the project as well, some of these include: vacuum test fittings/pump, oxy-acetylene gas welding assembly and boat-handling equipment.

Minor Equipment: This is a catch-all category for much of the SAUVIM equipment that will be added onto the vehicle. This is a wide range of equipment ranging over the vehicle. Some of the more important ones under development are:

Electrical Junction Boxes – Shown mounted upon the vehicle in Figure MED-43, these boxes allow for breakout and splitting of signals that are routed to various sensor heads and accessories on the SAUVIM Vehicle. Basically equipped with terminal strips inside they allow breaking of SAUVIM signals out into standard cables upon the vehicle. All of the junction boxes are currently mounted just aft of the arm storage space on the SAUVIM vehicle.

Video Cameras and Housings – These are completed in fabrication, save cutting a full set of restraints rods to seal them up with. These have been pressure rated to 4000psi with their current lenses (failure at 4500 psi). The main step to be completed before mounting these on the vehicle is fabrication of a set of articulating mounts for these units. Figure MED-44 shows the typical hardware that will be mounted. SAUVIM will be equipped with one rear-, and two side-facing monochrome cameras on fixed mounts and two articulating forward facing color cameras in a stereo-pair configuration as well as an arm-space monochrome camera. Additionally, as part of the AORD system, one monochrome camera is mounted on the end of the manipulator.

Instrumentation mounts – Much of the SAUVIM vehicle sensor suite will be mounted through the autumn of 2002. These include pressure, sensors, ranging sonars, scan sonars, and as mentioned above video.

Currently mounts are being fabricated for the ranging and scan sonars. These can be seen in Figures MED-45 and MED-46, respectively. Some articulation for mount aiming before launch is a feature in both designs.

Mount provisions for the Mission Sensor Package has been provided for two of the four components so far. Figure MED-47 shows the MSP mount in place on the SAUVIM frame.

Electrical Systems – Tremendous progress on the electrical systems has been realized in the past 12 months, culminating in the first live test in late spring. The design has proceeded from conceptual routing and assignment of electronics and procured items, to detailed wiring bundle design, documentation and fabrication, to mounting of the physical components to main wire-up to bench testing to vehicle operations.

About half of the planned vehicle electronics is installed, mainly those concerning the Navigation CPU and the thruster driver electronics. The vehicle is ready to run in digital open loop dead reckoning mode. Addition of sensors, circuitry and power for achieving closed loop positioning of sensor is the main focus of this area in the next 6 months, to be followed by addition of the Arm Computer and driving electronics into the respective bottles.

Batteries: The batteries are integrated onto the vehicle. A set of three nylon straps was sewed together for each battery to strap them into the aluminum trays that hold them as can be seen in Figure 48. Three batteries fit into each tray and four trays slide into the vehicle to make an array of 12 batteries as can be seen in Figure MED-49. Not shown here is some welding and structural modifications made to allow easier loading of the trays into the vehicle with provision for locking them down. { TC \12 " }

Battery capacity has been tested and it can anticipated that the SAUVIM has about two hours of thrust at 50% of full thrust on the main controllers. For purposes of maintenance ease these commodity batteries will be employed through shallow water testing and integration conducted this year. Literature search and examination of other undersea vehicles, most notably Pieces IV and V, has suggested the SAUVIM can pack 8-10 times the energy storage density in the same volume by switching to twin fiberglass boxes provisioned with lead acid batteries that are used in industrial carts. This will be a worthy upgrade prior to deep water testing and will be conducted in the spring of 2003.

Underwater Cabling: The full set of underwater cabling for routing power, main signals and control are complete. Some additional cabling supplies will need to be purchased for outlying sensors and for interfacing into the junction boxes on the vehicle.

Figure MED-50 shows a typical power cable used to feed the SAUVIM pressure vessels, while Figure MED-51 and MED-52, show a data cable and a close-up of such. Some work to ensure "idiot proofing" and routing information of the cables has to be performed at this time.

Pressure Vessel Electronics: These items make most sense to be broken down by the containing pressure vessel. At the time of this writing three pressure vessels have had all or a significant portion of their electronics installed; these are pressure vessel (PV) #3, PV#5 and PV#6. These are the navigation CPU and two thruster controller bottles.

Some of the process and documentation for the PV #3 will be covered as well as the fabrication.

PV3 – This is the main navigation CPU pressure vessel. And it carries many components as can be seen from previous reports. Basically the signal and power routings within the bottle, between the endlids is presented in Figures MED-53 and MED-54, respectively. Upon performing this step was the determination of how to break signals out of the VME boards, switch and/or condition then (if necessary) and then send them on their way out of the bottle. To accomplish this three layers of interior cabling were employed. { TC \12 " }

The first set, proceeding from outside the bottle inward, is a set of Teflon coated cables that run from the ocean tolerant penetrator and connector bodies mounted on the lids to a set of cannon-plugs formed from plastic with modular pins and sockets mounted on the aluminum component chassis that sits within the pressure vessel. These cables allow easy removal of the lids from the bottles, without necessitating removal of the entire electronics rack. Also they allow, given the use of the modular pin and socket assemblies, the ability to get at the underwater connectors for maintenance and inspection. { TC \12 " }

A second set of connectors takes the power and signals that have come in from the lids and route them to the low level power bus strips or conditioned power buses for the former and the I/O patch panel and/or switched accessories board in the VME rack for the latter. These are integrated onto the chassis and consist of hook-up wire of various gages.

The third set of cables are a set of ribbon cables that are lead out of the VME computers matrix I/O boards and that need to be routed to the conditioning/switching board or just interfaced into hook-up wire. These run between the VME boards and a patch panel, fabricated from red fiberglass to which the ribbon cables lead.

Figure MED-55 is an example of a pin-out assignment of one of the ribbon cables coming off of the VME computer board; one of these drawings exists for each cable assembly within the bottle. Figure MED-56, is similar in nature, but in this case it is for a cable in the second tier, those that run from components in the chassis out to the end panels. Not shown, but similar to this diagram is one more set of diagrams for the cables that interface between the ocean tolerant connections and the inner chassis themselves.

Prior to routing the cables inside the bottle was figuring out where and how the required assemblies would fit into the bottle. Figure MED-57 shows an early mockup model used to assess packing and that was also used in a series of heat rejection tests. Figure 58 is the first attempt a live rack but the Molex molded connectors and D-shell data connectors were found to be clumsy to use for interfacing cables to. So circular plastic cannon plug-type connectors with keying, twist-lock and variable density pins were chosen. What this lead to was a refinement of these racks as shown in Figure MED-59. For PV#3 all of the main electronic components are mounted to an aluminum box sandwiched between aluminum end lids. All of the lid cables run to connectors on the endlids. The VME computer is situated inside the box. The box is open on one side for the VME cables and connectors to reach the computers. Of the three remaining sides, one is devoted to low level as well a switched buses, another is for the DC-DC power supplies needed to condition power for all of the electronics and the remaining side is given over to hosting the three PC-104 computers, video MUX circuitry, stepper controllers, and conversion panel for the ribbon cable to hook-up wire conversion. From left to right on the table in the photograph is the contents of PV#5 (two assemblies), PV#6 and PV#3.

Figure MED-60 shows the final working electronics rack for PV#3 under construction for the mounting of the main components, bus strips, cannon plug connectors and ribbon cabling. Figure MED-61 shows the completed rack being readied for installation into the assigned pressure vessel. Please notice the red fiberglass panel where the ribbon cables are broken out and converted to conventional hook-up wire for routing around to the end lids. For clarity Figure MED-62 shows the same unit being debugged while sitting between its endlids during integration testing. Note the main switching relay, bus strips and power terminals as well as the endlid to aluminum rack wiring set (the black wire bundles running from each endlid).

One of the more involved aspects of wiring this pressure vessel involved the signal switching, fuse protection, and conditioning circuitry. This interface circuitry was enclosed upon a single board installed within the 6U VME rack to save space in a spare slot as is shown here in Figure MED-63. An outside vendor based on a design generated on Visio software and supplied them etched the board. The main features of the board are the power supply connector in the upper right, the switched power and signal connectors shown in the center two connectors, and the VME I/O interface connector (black rectangle in the upper left). The fusing provided for the switched loads is the white cans. The black rectangles is an array of electromagnetic relays to switch loads like the sensors heads, PC-104 computers, ballast drop solenoids and the like. The two empty carries off on the right with support circuitry are for interfacing Basic Stamp micro-controllers for the arm and ballast tray stepper controllers. These here are to avoid bogging down the main VME with interrupts to clock the stepper control circuitry.

Figure MED-64 shows the hierarchy of how power is switched throughout PV#3. Much of this is taken care of from this interface board.

Figure MED-65 is the main top-level circuit diagram for this board. Figures MED-66, MED-67, and MED-68 all detail certain components of the Health Monitoring, Power-up, and Arm/Ballast Tray control circuitry. All of these are included within the board.

PV5 – These bottles uses a platter stack on each end of the bottle similar to that used in missile electronics racks to fasten bus strips and conditioning circuitry to. The platters are held together by tie-rods that are run between spacers and are tapped into an aluminum plate that bolts, in turn, to each end lid. These aluminum blocks are what the main thruster controller and mounted firmly to. This allows for good heat rejection to the surrounding water should it be needed. Figures MED-69 and MED-70 are top and side views of this bottle. The controllers are paired to have one larger controller (intended for the longitudinal thrusters) and one smaller controller for the corresponding vertical controller upon each lid. A pair of pass-thru cables exists to allow lid removal. These are the black connectors snaking back and forth between each lid. This bottle has a main electro mechanical relay to switch power to the main bus. A solid state relay was used but was found to fail after a mere half-dozen cycles due to the large capacitors and initial transient current the draw, on the order of 700-800 amps. This bottle has a small-perforated board that contains the health monitoring circuitry much like that in Figure MED-66. { TC \12 " }

PV6 – is very similar in concept to PV#5 except that all of the electronics is shifted to one end lid. This bottle container four of the smaller controllers hence they are all mounted on one end lid. Otherwise in the circuitry it is very similar to PV#5. A top and side view of this electronic rack is provided in Figures 71 and 72. { TC \12 " }
{ TC \12 " }

Analogue Control: Some circuitry has been prepared to assist in using the vehicle during initial dynamic testing trials as well as early open-loop controls testing. Currently this circuitry is analogue in nature, however, it is being changed over to digital control as this report is in writing. { TC \12 " }

Three main components for this were secured and/or fabricated: an ROV style tether, a surface trailing buoy and a control box.

The ROV tether has provisions for routing Ethernet, serial and moderate amounts of power up and down to the AUV. Currently it runs from PV#3 to the surface buoy

The next item is the buoy. This contains a wireless Ethernet lan hub that will be used for command control from a laptop computer so equipped. Also inside currently is analogue joystick decode circuitry that has provisions for signal gain and offset tuning, and several toggle switches to set buoy interface mode, digital control, thruster, test box and analogue joystick control as well as control power into and the PV#3 bottle.

The last items are two control boxes that both jack into the buoy. The first one is a control box, of which the circuitry is shown in Figure MED-73 that is used to merely run each thruster up and down for diagnostics and tuning. A second box is a control console with a 3-axis joystick, a slide-pot control, and a mushroom (safety switch). The mushroom switch is slaved to the two electromagnetic relays in PV#5 and #6 to prevent a “runaway” incident from happening during initial testing. The three axes of the joystick control fore/aft, yaw left/right and translate left and right motions while the slidepot controls diving behavior of the SAUVIM.

Most of these items are and will house a couple of generations of electronics by which to test the SAUVIM during shakedown trials. Eventually, the buoy will be merely for data collection, the SAUVIM will proceed in AUV mode between tests.

Testing Trials – There have been two tests of the SAUVIM vehicle to date. The first, took place in November of 2001 was a static balance test, and the second in April of 2002 was a dynamic stability and electrical power distribution, control and switching test, as well as further static stability testing and tuning. { TC \13 " }

Static Test (Nov. 2001): Though this is not vehicular hardware system per se, it is an area of the SAUVIM design that has required constant attention since the initial concept designs of the frame and major component placements were being considered throughout the first 12 months of the project. About three major iterations in the vehicular design at the advanced conceptual level were proceeded through and this factor was considered through each.

For purposes of keeping account of the vehicle statics, a coordinate system was defined on the vehicle as detailed in Figures MED-74 and MED-75. The origin point is located at the lowermost, rearmost point of the lowermost, rearmost lateral I-beam member of the SAUVIM vehicle. From here the +x-axis runs forward along the ventral line of the SAUVIM belly, meanwhile +y-axis proceeds out to the right side of the vehicle, therefore by right-hand screw rule the +z-axis proceeds vertically out of the top of the vehicle. This is not in keeping with the standard convention of coordinates systems for AUV control but this one was set up by MED staff for in-house measurement and component tracking and was deemed a more logical one for the task at hand here.

A database in spreadsheet form has been kept of all the larger masses and volumes on the vehicle and the coordinates as to their respective individual center of mass (COM) and centroid of volume (COV). A table of the over one hundred major components (i.e., floatation=21, frame=46, components=40) on the vehicle was compiled and their exact coordinates along with their associated mass and volume were determined. Initially, these were comprised mainly of estimates, but as solid models and actual hardware have been developed over the last 18 months, more accurate estimates have been generated from the CAD software and also by means of direct measurement. The static moment contribution of each component was then computed using the classic mechanics equations.

Corrections in wet mass contribution were made to account for density loss upon immersion in seawater. Basically this was done by accounting for both mass and the volume of the various

components together, from these data the effective specific gravity for the various components could be determined and therefore the wet weight determined by reducing the effective specific gravity (sg. is reduced by 1.0) for the immersed vehicle state. Meanwhile the dry mass state was also tracked as obviously vehicle stability and behavior during craning and launch/recovery operations is critical as well.

The estimated position of these points using data updated in the autumn of 1999 locate the center of mass (immersed SAUVIM) and the center of volume in the locations of the vehicle, these are for the COM at the following coordinates on the vehicle: $x=69.39\text{in}$, $y=-0.88\text{in}$, and $z=32.65\text{in}$ while the COV is at $x=69.39\text{in}$, $y=-1.20\text{in}$ and $z=37.28\text{in}$.

Last November the SAUVIM was dip tested as depicted in Figures MED-76 and MED-77. The vehicle floated with about a 6-degree nose up configuration with about 300lbs positive buoyancy. Application of some 150lbs or iron plates and a personnel standing on the front sank it on an even keel. The experimental determination of the weight and the calculated spreadsheet tally from the design were only off by about 3% of the predicted value. A recording of the vehicle loads and masses and corrective weight applied was kept in a CAD file for future reference as a loading baseline.

For this test the SAUVIM was without the ballast trays.

Dynamic test (Apr. 02): This test was to determine dynamic stability of the vehicle in the water as well as assess the power systems, thrust levels, speed, battery longevity and refine the static trim worksheet. Figure MED-78 shows the launch for this test. Upon launch, the vehicle entered the water to be about 200lbs positively buoyant by calculated estimates. About 150 pounds of buoyancy was eliminated by clamping iron plates onto the top spine of the vehicle. The remaining was eliminated by controlled flooding of four soft ballast tanks, these allowed for fine tuning to neutral buoyancy and for pitch trim elimination. The usefulness of these tanks has established that some type of variable ballast will be extremely useful to have on the vehicle.

Upon completion of static balance trim. the test the buoy and control box, Figures MED-80 and MED-81, were in use as this was the first live test of the vehicle. Figures MED-82 and MED-83 display the vehicle in steady ahead motion and initiating a vertical dive. The vehicle tracked well and had no tendency to pitch, porpoise, or fishtail while executing maneuvers. Estimated speed with the thrusters de-tuned to about 50% full max power was on the order of 1 to 1.25 knots.

The only failure in this test was the soft ballast tanks. Gradual but excessive loss of air through the duration of the test, as shown in Figure MED-82, and implosion for one of the tanks, as shown in Figure 83, sent the vehicle to the bottom for a while. However it was brought to the surface by reeling in the tether. Based on the value of these tanks, albeit with dubious reliability, it has been decided to add a reliable set of soft ballast tanks to the vehicle as was detailed in an earlier section.

Phase II Tasks:{ TC \11 }

- Fairing – complete detail design, procure materials and fabricate.
- Mount and wire-up primary instrumentation (forward and rear scan sonars, six ranging sonars, two pressure sensors, 5 video cameras and INS unit).
- Fabricate, mount, and wire-up soft ballast tankage.
- Fabricate, install and wire-up hard ballast drop mechanisms.

- Add provision for charging of the SAUVIM in the water and switching of the main power relays from the vehicle.
- Integrate arm computer onto vehicle pressure vessel #4.
- Install MSP package onto vehicle and integrate.
- Prepare arm for transfer to vehicle.
- Initiate planned upgrade of main battery banks.
- Investigate mounting of DVL unit.
- Investigate in conjunction with RDC group acoustic modem links for integration onto vehicle a replacement of tether.

References

- Avallone, E.A, and Baumeister, T., Mark's Standard Handbook for Mechanical Engineers - Ninth Edition, McGraw-Hill Book Co., 1987.
- Ayache, N., Artificial Vision for Mobile Robots : Stereo Vision and Multisensory Perception, MIT Press, 1991.
- Beckwith, T.G., and Maranogoni, R.D., Mechanical Measurements, Addison-Wesley Publishing Company, 1990.
- Borland C++ User's Guide, Borland International, Inc., 1992.
- Callister, W.D., Materials Science and Engineering, John Wiley and Sons, Inc., 1991.
- Clayton, B.R. and Bishop, R.E.D., Mechanics of Marine Vehicles, E. & F.N. Spon Ltd., 1982.
- Choi, S.K. and Yuh, J., "Development of an Omni-Directional Underwater Intelligent Navigator", *IEEE Robotics and Automation Magazine*, 1995b.
- Comstock, J.P., Principles of Naval Architecture, Society of Naval Architects and Marine Engineers, 1967.
- Cox, A.W., Sonar and Underwater Sound, Lexington Books, 1974.
- Engelmann, W.H., Handbook of Electric Motors, M. Dekker, 1995.
- Evans, A.J., Basic Digital Electronics - Digital System Circuits and Their Functions, Master Publishing Inc., 1996.
- Fox, R.W. and McDonald, A.T., Introduction to Fluid Mechanics, John Wiley and Sons, Inc., 1992.
- Gere, J.M. and Timoshenko, S.P., Mechanics of Materials, PWS-Kent Publishing Co., 1990.
- Gill, R., Electrical Engineering Handbook, Siemens Co., 1985.
- Hibbeler, R.C., Engineering Mechanics, Macmillian Publishing Co., 1992.
- Hill, P.G. and Peterson, C.R., Mechanics and Thermodynamics of Propulsion, Addison Wesley Publishing Co., 1970.
- Hoerner, S.F., Fluid Dynamic Drag: Practical Information on Aerodynamic and Hydrodynamic Resistance, American Institute of Aeronautics and Astronautics, 1965.
- Howard, G., Automobile Aerodynamics : Theory and Practice for Road and Track, Motorbooks International, 1986.
- Hudson, J. and Luecke J., Basic Commnications Electronics, Master Publsihing, Inc., 1999.
- Hughes, A., Electric Motors and Drives - Fundamentals, Types and Applications, BH Newnes, 1994.
- Incropera, F.P. and DeWitt, D.P., Introduction to Heat Transfer, John Wiley and Sons, 1985.
- Kernighan, B.W. and Ritchie, D.M., The C Programming Language, Prentice-Hall, 1978.
- Krar, S.F., and Amand, J.E., Machine Shop Training, McGraw-Hill Co., 1967.
- Lander, C.W., Power Electronics, McGraw-Hill, 1987.
- Lawry, M.H., I-DEAS Student Guide, Structural Dynamics Research Corp., 1990.
- Linear Circuits Operational Amplifiers Data Book, Texas Instruments, 1992.

Lines, D., Building Power Supplies - Useful Designs for Hobbyists and Technicians, Jerry Leucke Master Publishing Inc., 1997.

Marine Technology Society, ROV Committee, Intervention/ROV '91 Conference & Exposition (9th : 1991 Hollywood, Fla.), ROV Committee and the South Florida Section of the Marine Technology Society, 1991.

Martini, L.J., Practical Seal Design, Marcel Dekker, Inc., 1984.

Mattsson, E., Basic Corrosion Technology for Scientists and Engineers, Ellis Horwood Ltd., 1989.

Microsoft QuickBASIC - Learning to Use Microsoft QuickBASIC, Microsoft Corp., 1988.

Microsoft QuickC - Learning to Use Microsoft QuickC, Microsoft Corp., 1988.

Mims, F., Getting Started in Electronics, Radioshack Co., 1998.

Needler, M.A. and Baker Don E., Digital and Analog Controls, Reston Pub. Co., 1985.

Omura, G., Mastering AutoCAD 13 for Windows95, Windows3.1, and WindowsNT, Sybex, 1995.

Parrish, A., Mechanical Engineer's Reference Book, Butterworths, 1973.

Product and Data Catalog - Reynolds Aluminum Supply Company, Reynolds Aluminum Supply Company, 1989.

Pugh, H., Mechanical Behavior of Materials Under Pressure, Elsevier Publishing Co., 1970.

Porter, H.W., Machine Shop Operations and Setups, American Technical Society, 1967.

Sayers, C., Remote Control Robotics, Springer, 1999.

Scheck, L.A. and Edmondson, G.C., Practical Welding, Glencoe Publishing Co., 1984.

Schlichting, H., Boundary-Layer Theory, McGraw-Hill, Inc., 1979.

Schwartz, M.M., Composite Materials Handbook, McGraw-Hill, 1984.

Schweitzer, P.A., Corrosion and Corrosion Protection Handbook, M. Dekker, 1983.

Serway, R.A. and Faughn, J.S., College Physics, Saunders College Publishing, 1989.

Shigley, J.E. and Mischke, C.R., Mechanical Engineering Design, McGraw-Hill, Inc., 1989.

Smith, C.S., Design of Marine Structures In Composite Materials, Elsevier Applied Science, 1990.

Takashi, K., Electric Motors and their Controls: An Introduction, Oxford University Press, 1991.

Tsai, L.W., Robot Analysis - The Mechanics of Parallel and Serial Manipulators, John Wiley and Sons, Inc. 1999.

Tupper, E., Introduction to Naval Architecture, Butterworth Heinemann, 1996.

Ullman, D.G., The Mechanical Design Process, McGraw-Hill Inc., 1992.

UnderPressure Software Manual, DeepSea Power and Light Co., 1990.

Valentine, R., Motor Control Electronics Handbook, McGraw-Hill, 1998.

Weast, R.C. and Astle, M.J., CRC Handbook of Chemistry and Physics - 61st Edition, CRC Press, Inc. 1981.

Wilson, P.A., International Conference on Maneuvering and Control of Marine Craft, Proceedings of the Second International Conference, 1992.

Yuh, J., Underwater Robotic Vehicles: Design and Control, TSI Press, 1995.

Other sources of information:

Catterson, T. and Kerby, T., Pisces VI layout and design notes.

Dudley, R., Wood's Hole Oceanographics Institute, ALVIN and ABE power and ballast systems notes.

Healy, T., Naval-Post Graduate School, Notes and information on AUV-II electrical and fin layout.

Lee, P.M., VORAM Vehicle Electrical System Layout Drawings.

Rollins, S., PMRF Facility, AMPS vehicle design notes and specifications.

Smith, S., Florida Atlantic University, information on underwater connectors and data routing.

Company Literature that was used for design notes and referencing include:

Advanced Motion Controls

Ansaldo-Maris Co.

Apple Rubber Products
Balmoral
CHAM Co.
Deep Sea Systems
Flotech
Hydro-Seal
ISI Automation
Kaiser Aluminum
Miller
OceanScience
Phillips Electronics
Perry-Tritech
Reynolds Aluminum
SeaCon Brantner
Subconn
Texas Instruments
WPI Burton

Birns
Band-It
DeepSea Power and Light, Inc.
Delta Wire Rope
Honeywell - Data Instruments
Impulse Ent.
JR-3
Kemlon
MCG
Omega
Pelagic Electronics
Precision Associates
Rush Gears
Seal Science
Tecnadyne ,Inc.
Underwater Systems

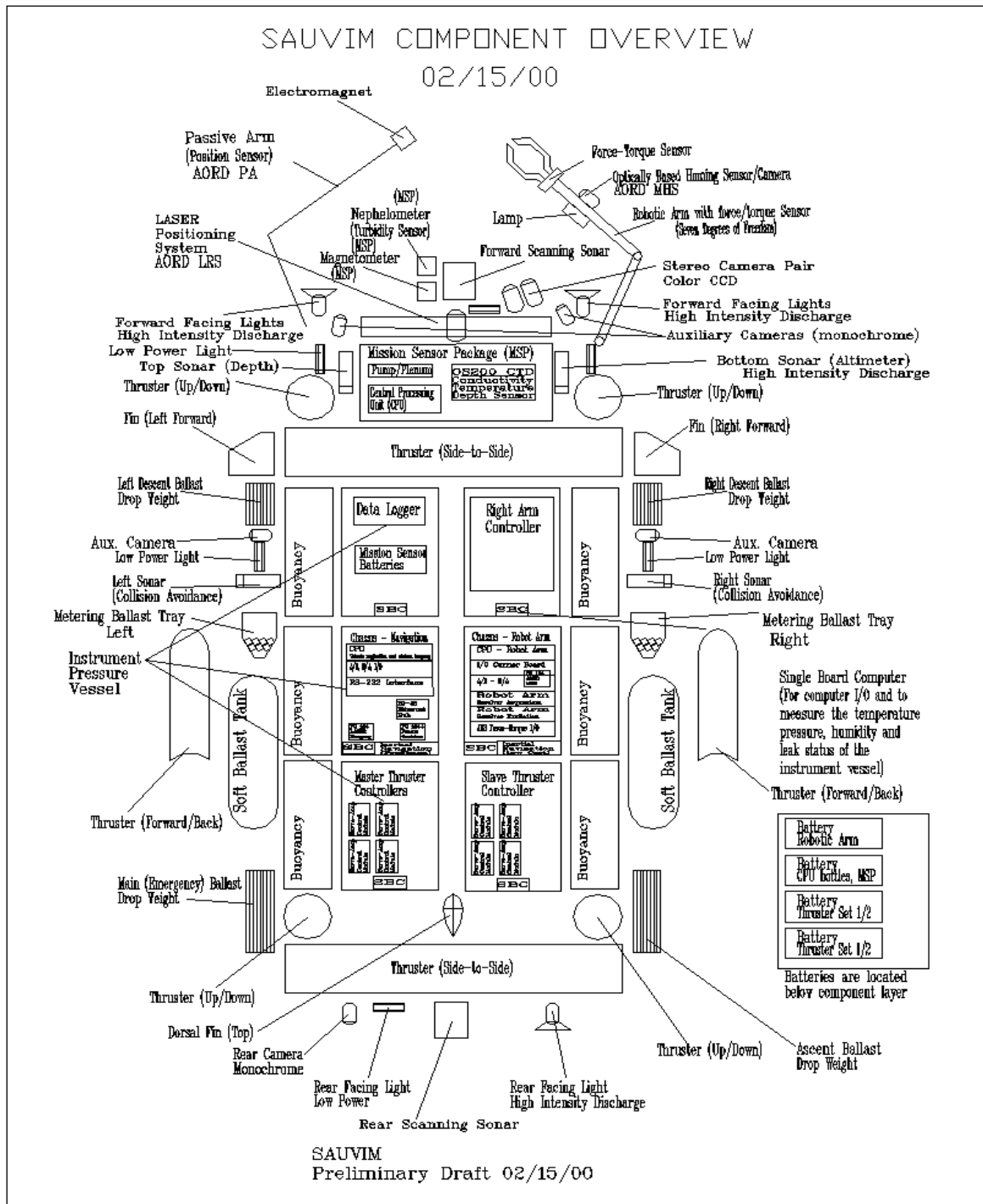


Figure MED-1: SAUVIM AUV with Major Component Systems.

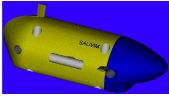


Figure MED-2: Conceptual View of Faired SAUVIM Vehicle.



Figure MED-3: 1:12 Scale Model Plug for Tank Testing.

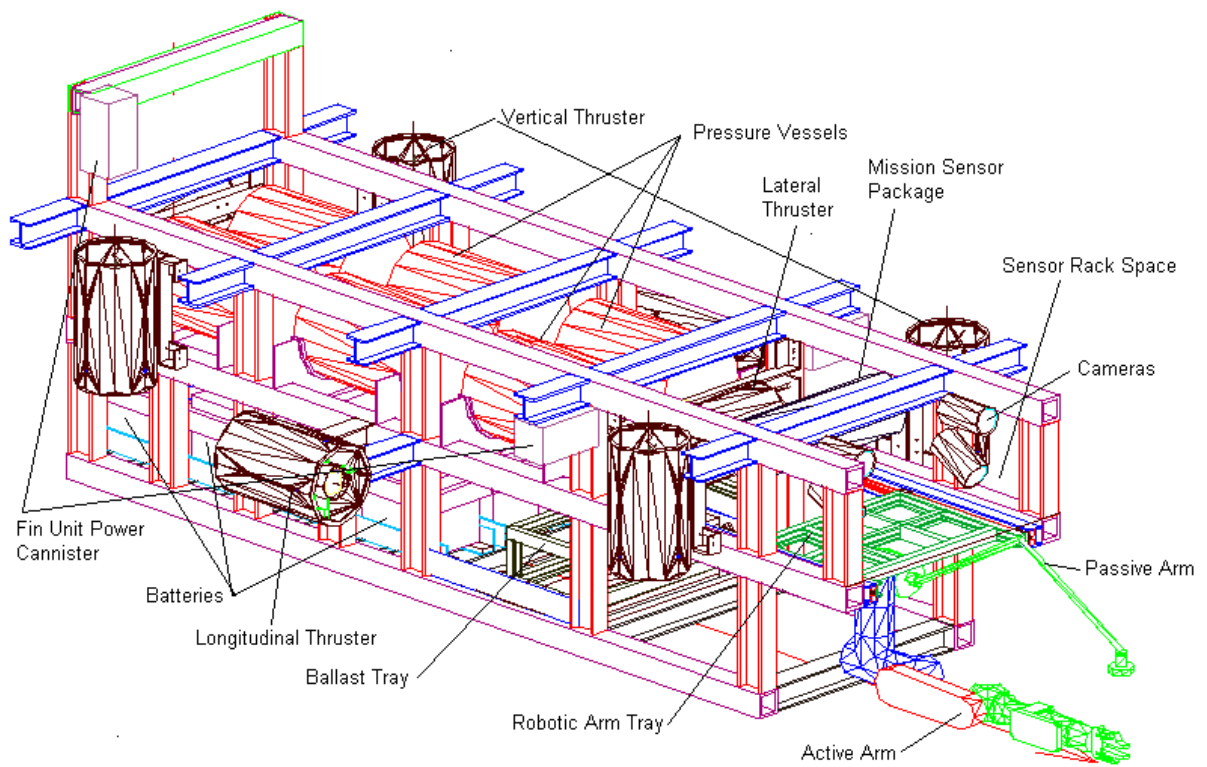


Figure MED-4: Major Component Locations on the SAUVIM



Figure MED-5: SAUVIM with Some Selected Components Mounted.

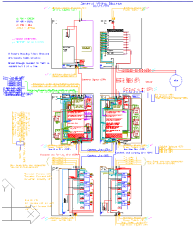


Figure MED-6: SAUVIM Electrical System Overview

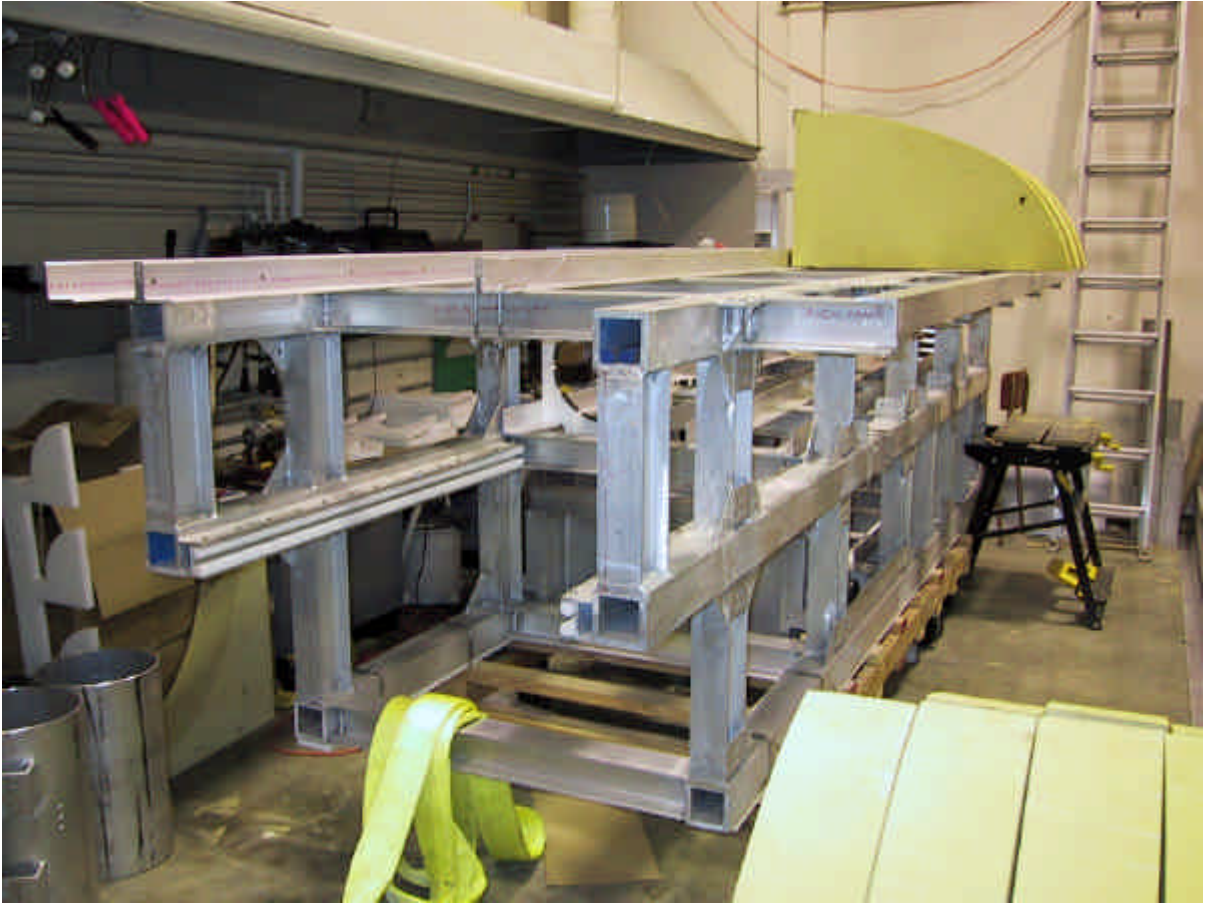


Figure MED-7: SAUVIM Main Structural Frame

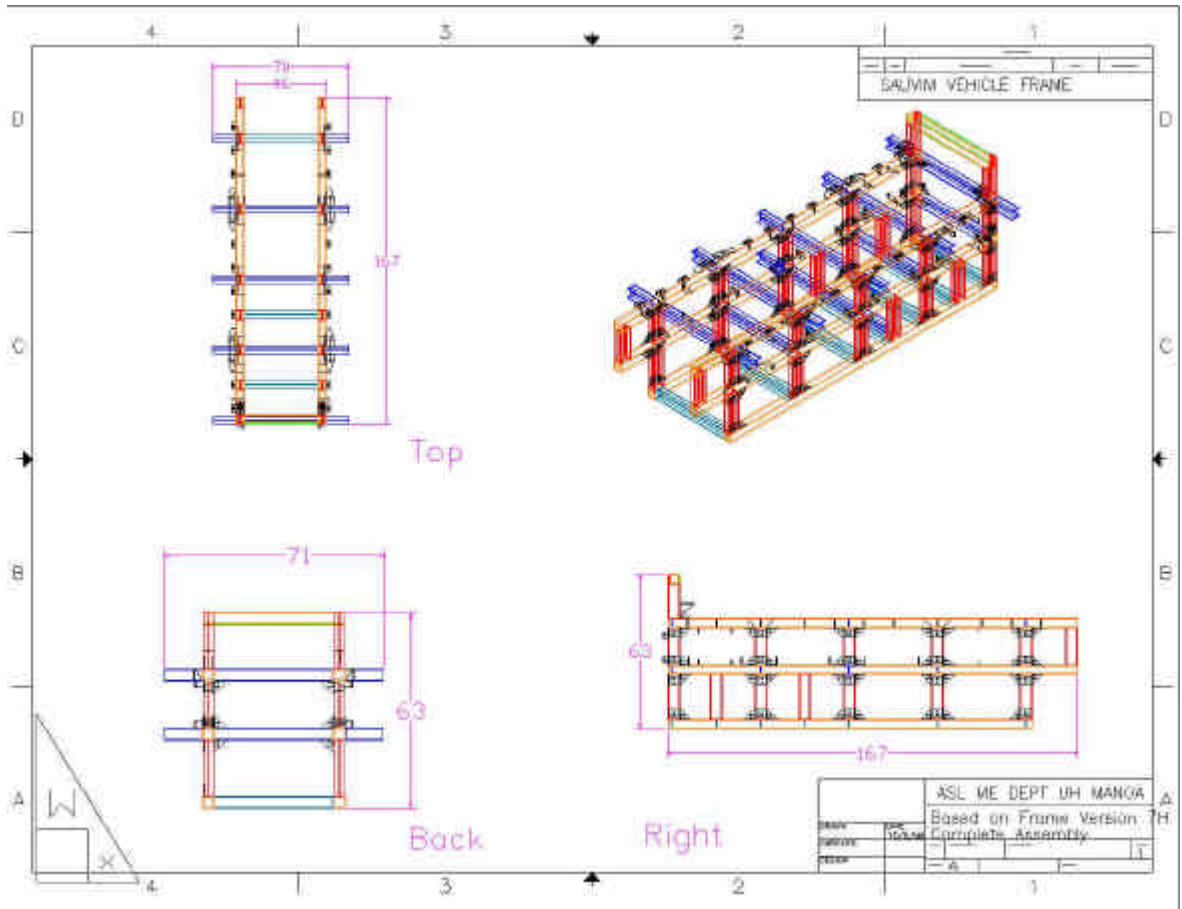


Figure MED-8: CAD Drawing of SAUVIM Frame.

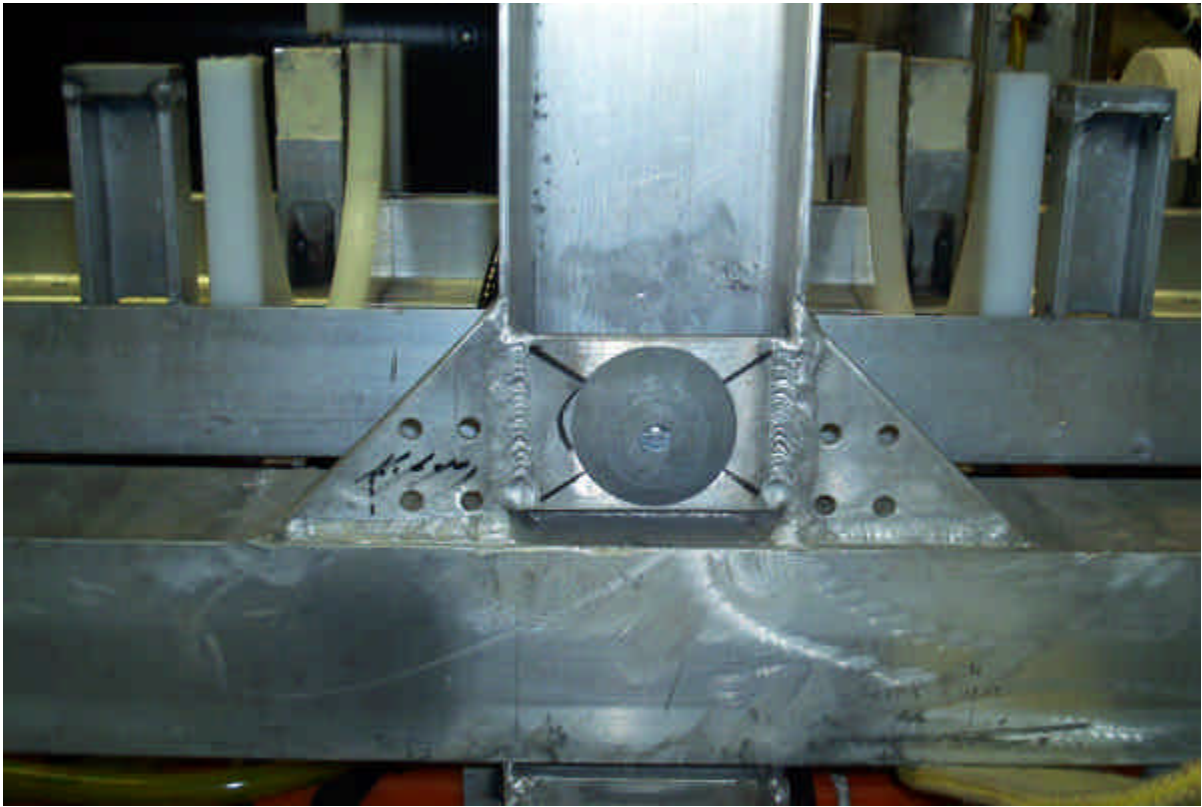


Figure MED-9 - Passive Anodes in Place on Frame



Figure MED-10 - Composite Photos of Flush System.



Figure MED-11 - Composite Bottle Set



Figure MED-12 - Vacuum Hold Test and Lid Installation



Figure MED-13 - Typical Flooded Bottle Test Result.

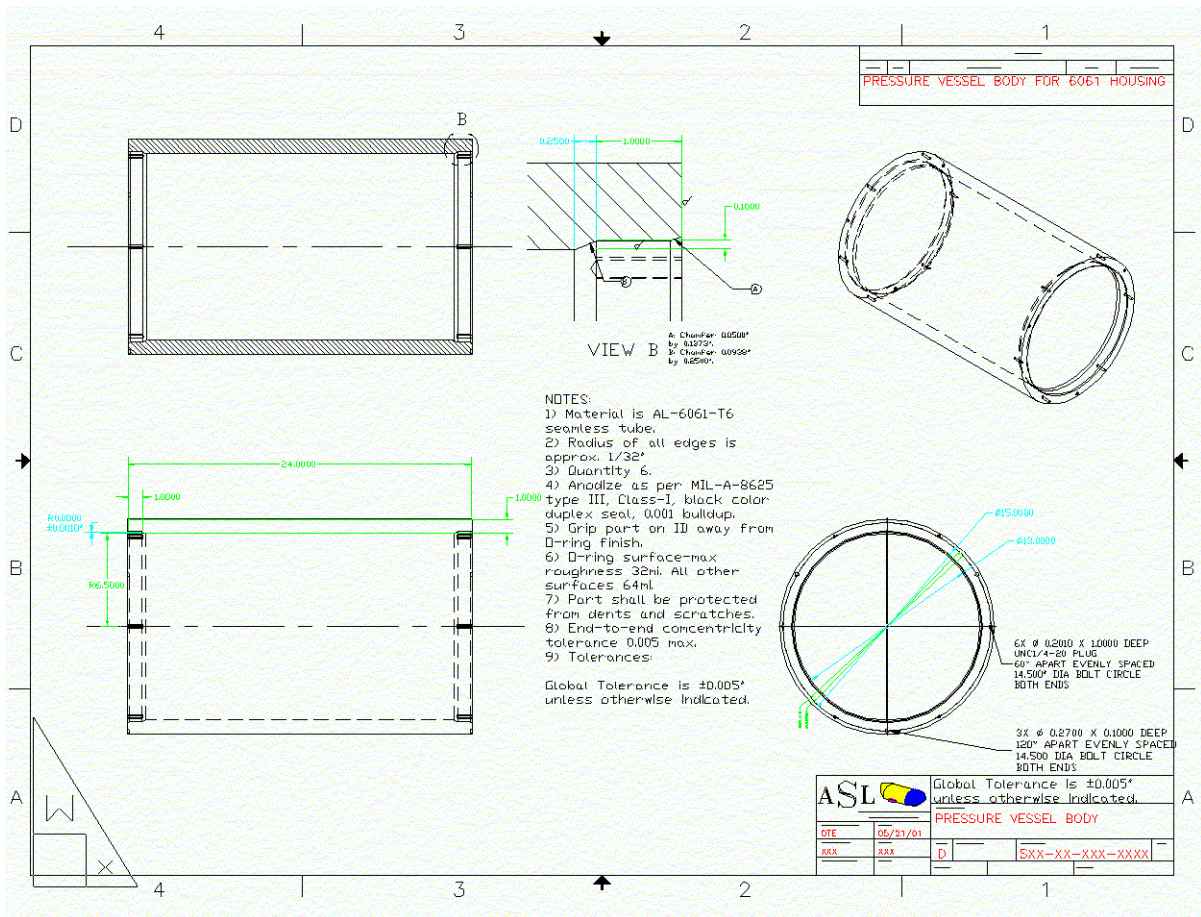


Figure MED-14: UH Pressure Vessel In-House Design of the Body.

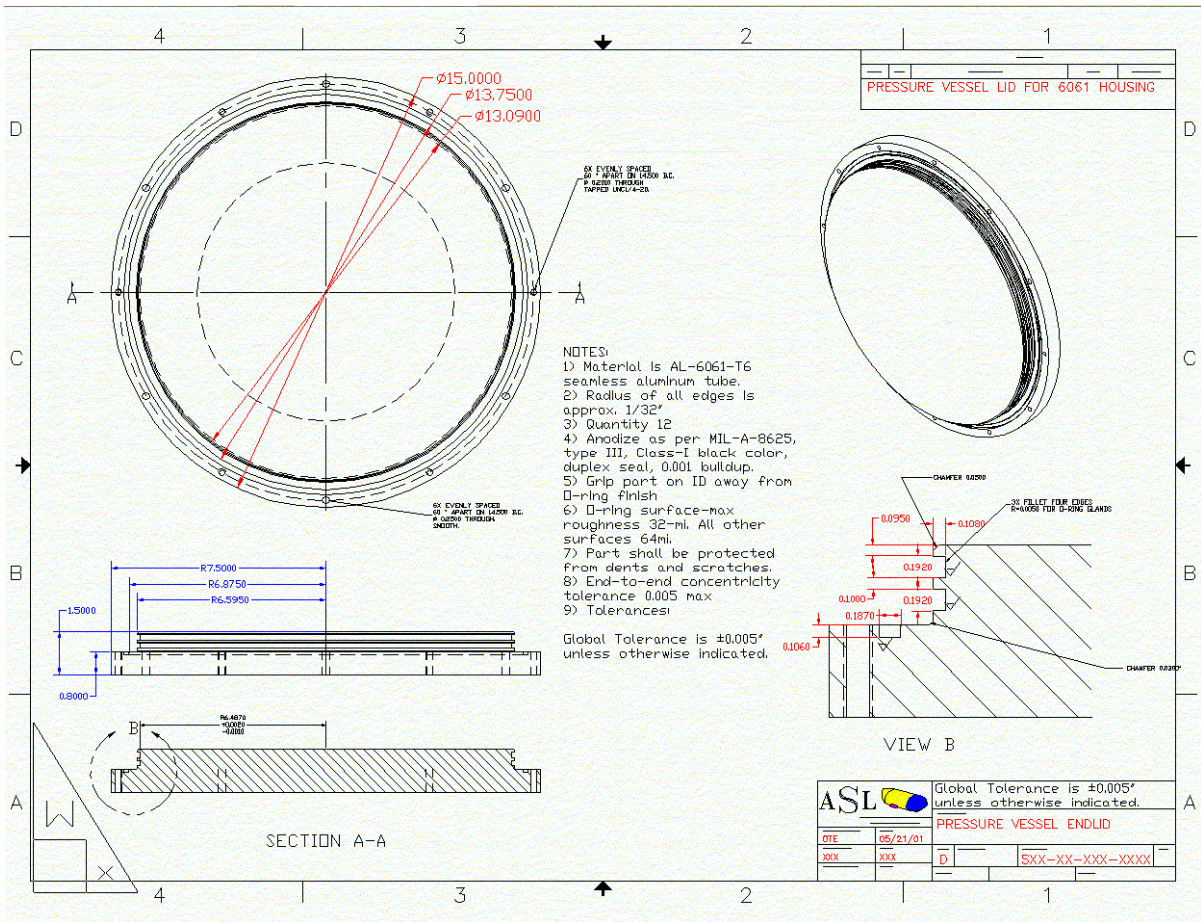


Figure MED-15: UH Pressure Vessel In-House Design of the Endlids.

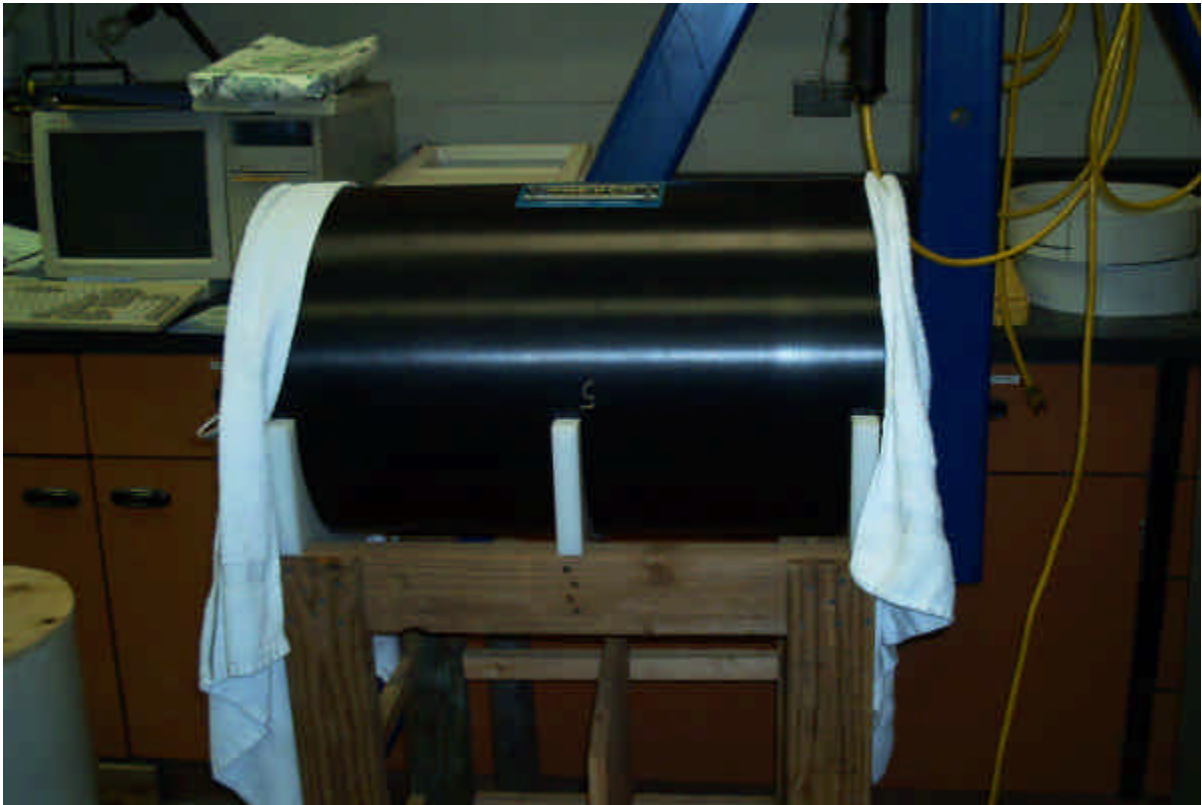


Figure MED-16: Commodity Pressure Vessel (Prevco Inc.).



Figure MED-17: Pressure Vessels Under Fabrication at Vendor.



Figure MED-18 - Complete Set of Commodity Pressure Vessels.

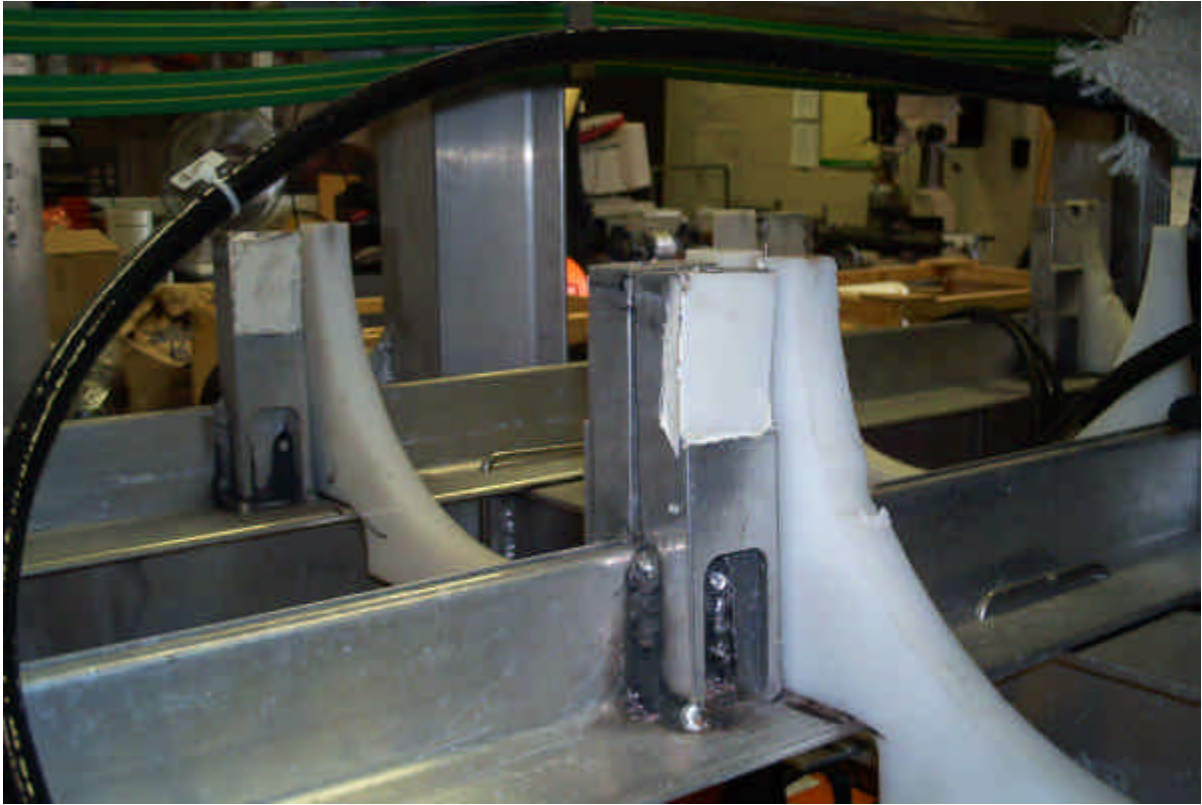


Figure MED-19: Reconfigured Pressure Vessel Saddle

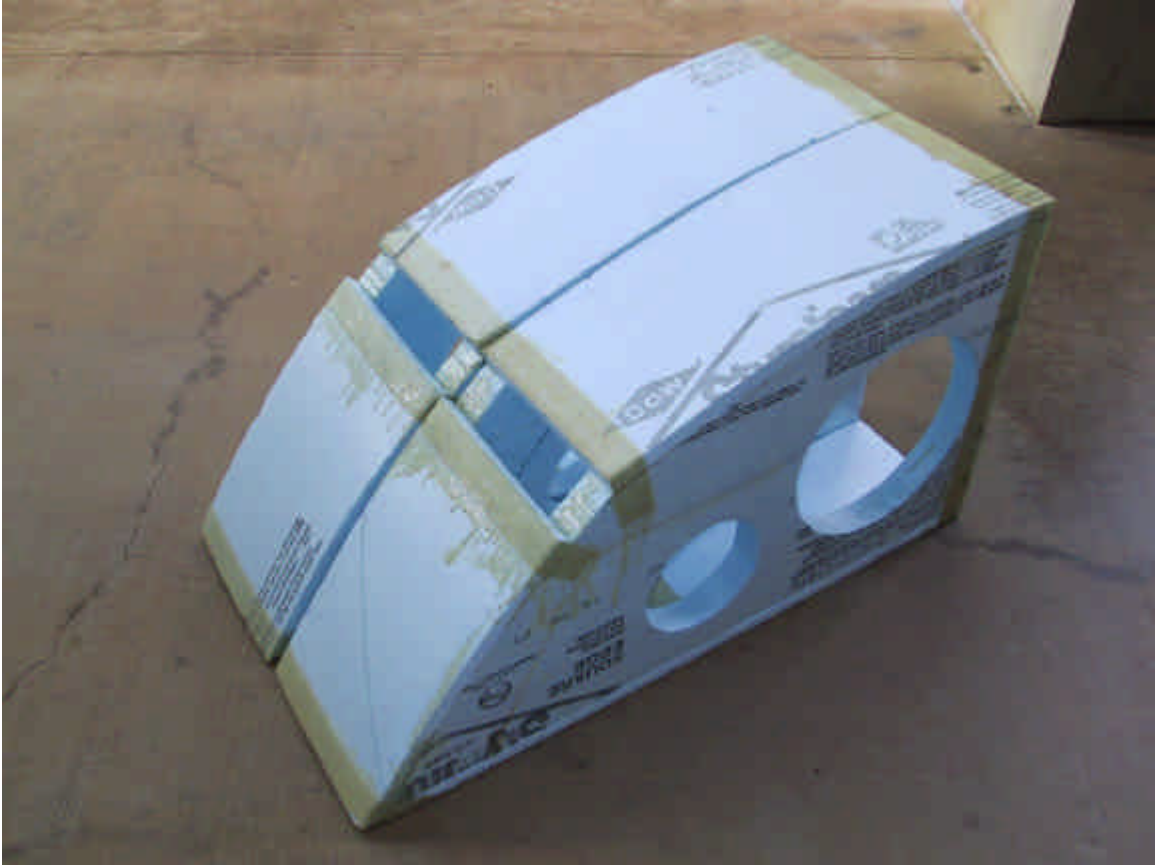


Figure MED-20: 1:1 Model of Dorsal Foam for the SAUVIM Vehicle.

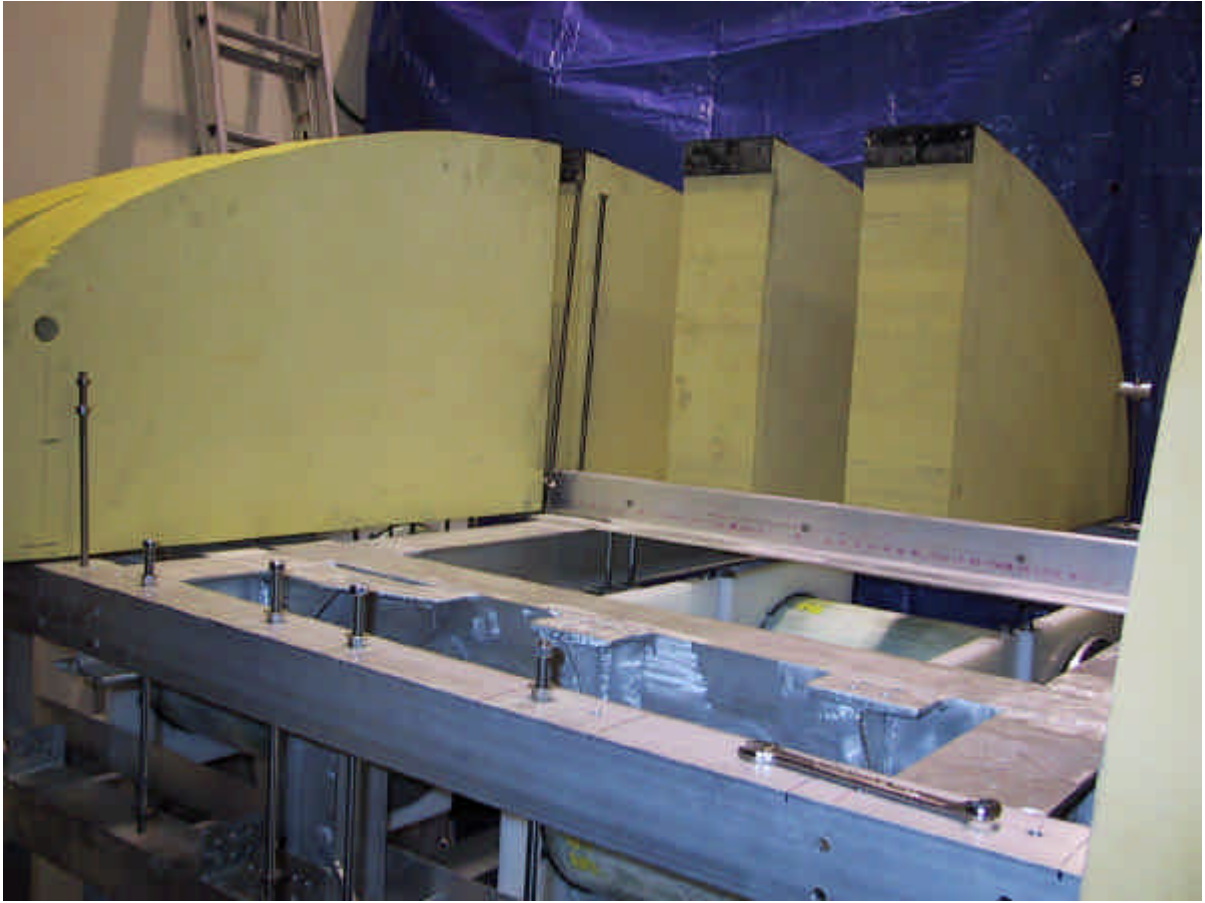


Figure MED-21: View of Dorsal Foam in place on SAUVIM Vehicle.



Figure MED-22: Top View of SAUVIM Dorsal Floatation Foam.

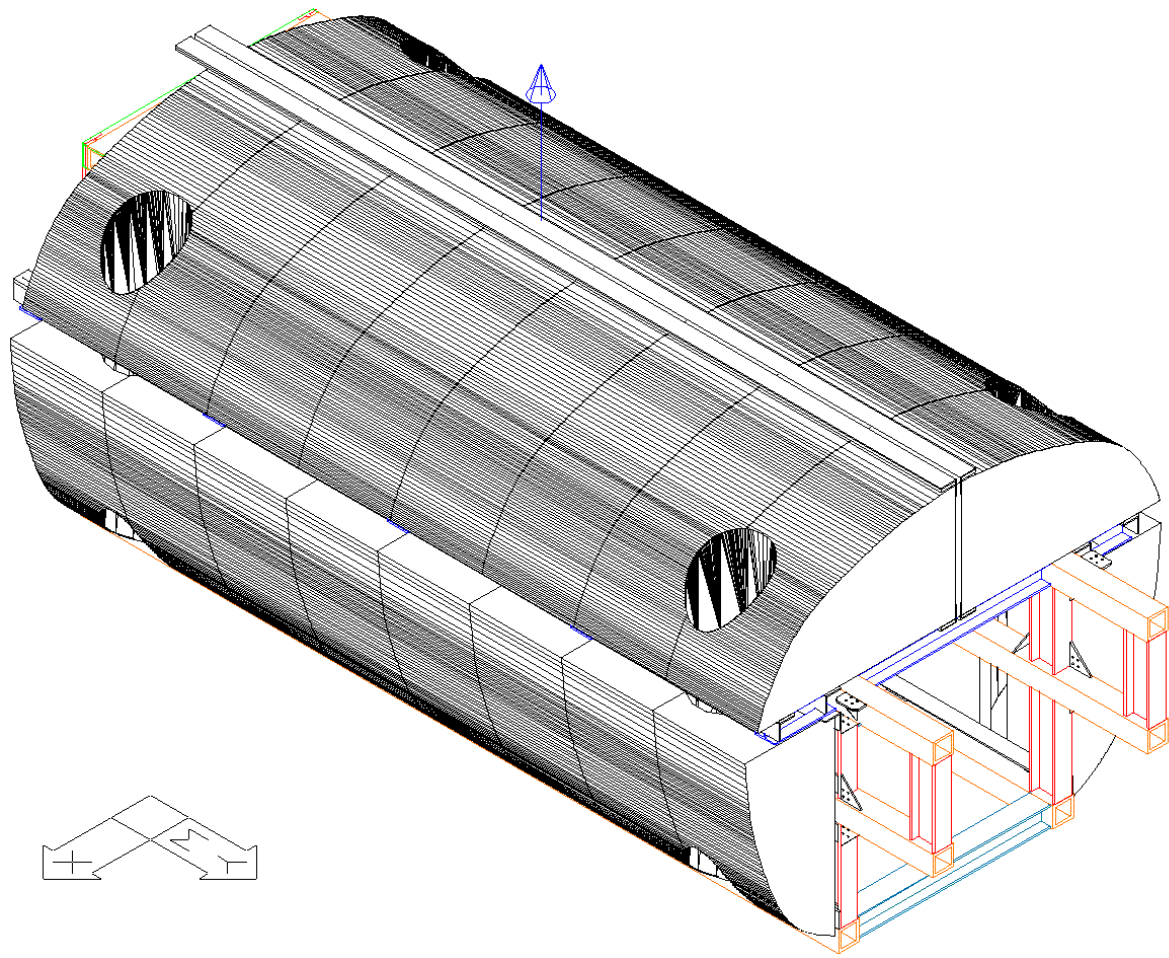


Figure MED-23: Deep Water Foam on SAUVIM.

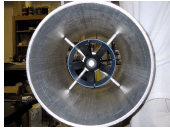


Figure MED-24: View of Thruster from Rear in Tube (1020 Tecnadyne).



Figure MED-25: Thruster Tube Installation, Rear-Lateral Tube.



Figure MED-26: Tecnadine Thruster with Struts.

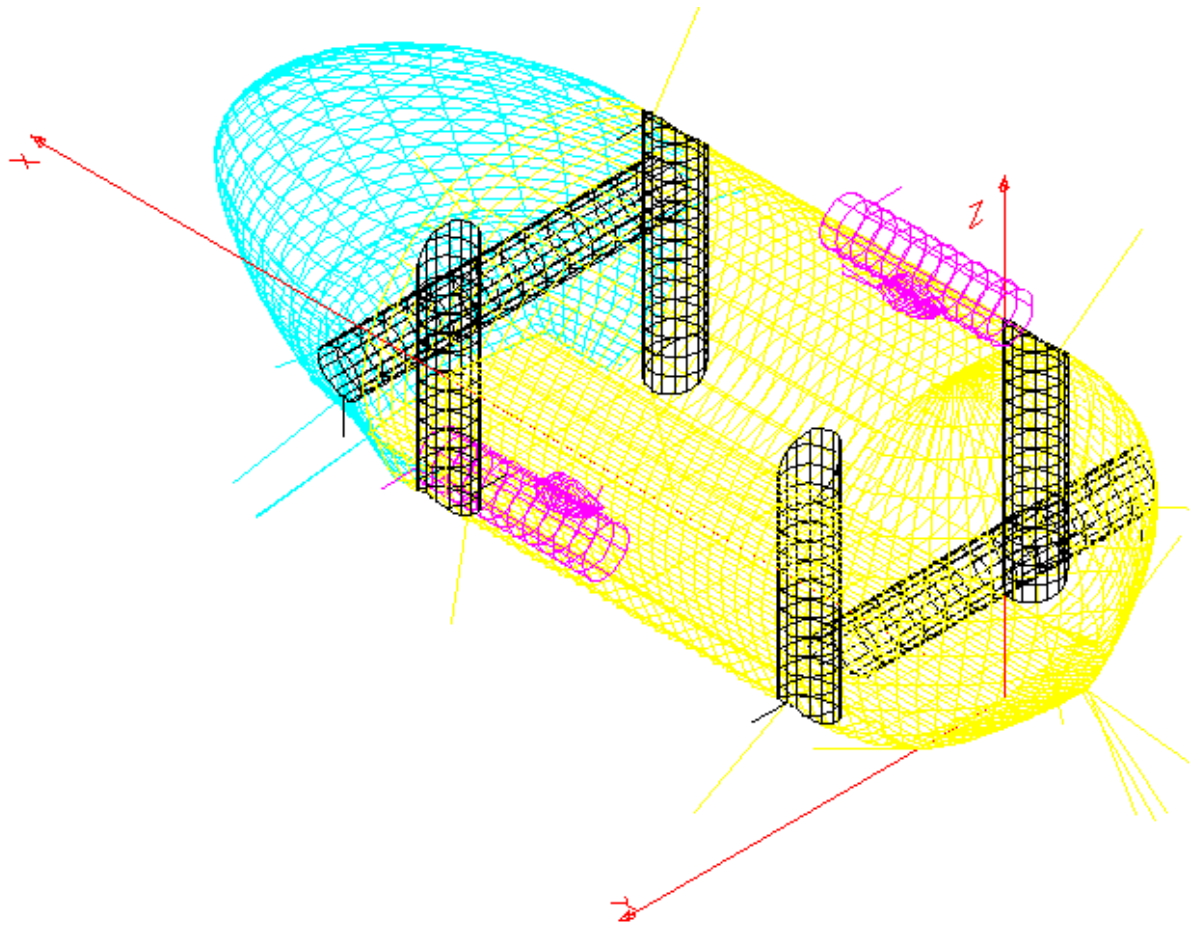


Figure MED-27: Conceptual View of Fairing.

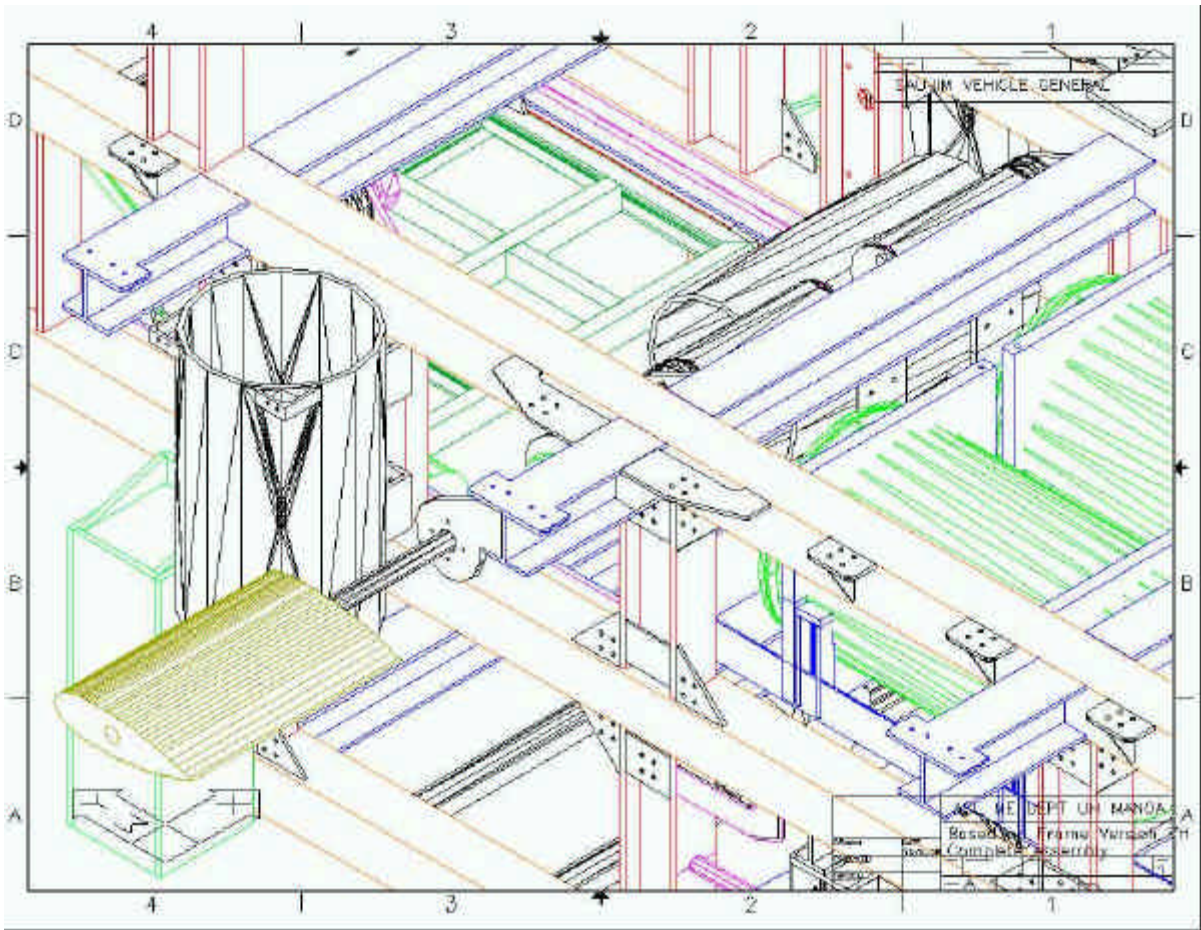


Figure MED-28: Placement of Fins, Bearing Shafts and Power Canisters onto Vehicle.

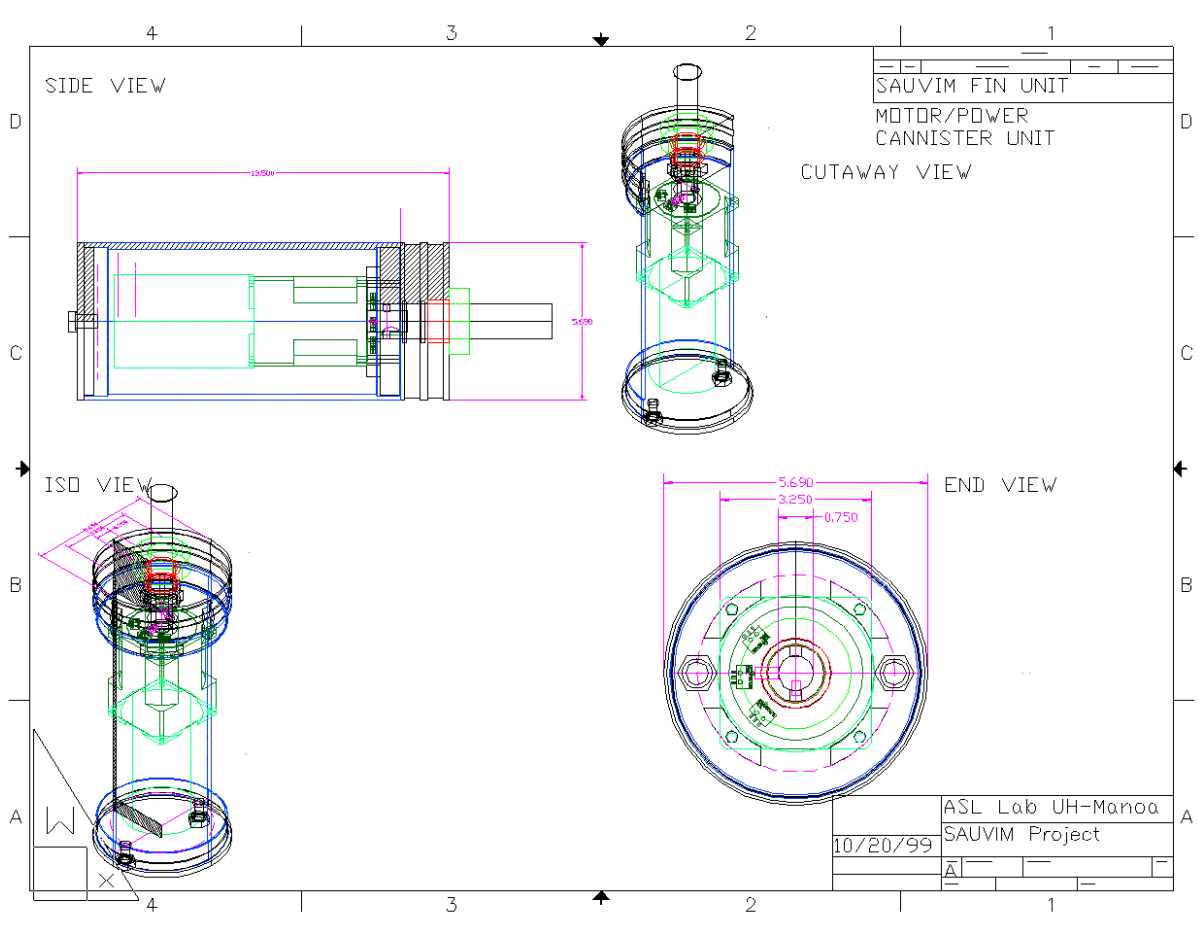


Figure MED-29: Power Canister for Fins, Arm-tray, and Ballast Tray.



Figure MED-30: Servo Motor and Controller



Figure MED-31: Starboard Forward Ballast Mount on Forward Corner of Frame.

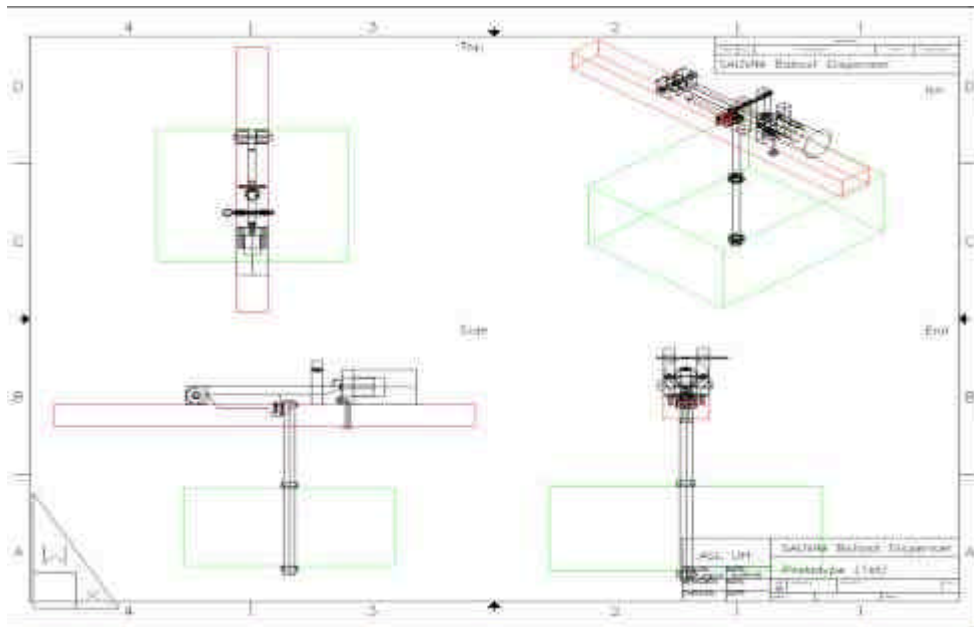


Figure MED-32 - Forward Ballast Mount Drop Mechanism.



Figure MED-33: Main Ballast Tray Side View.

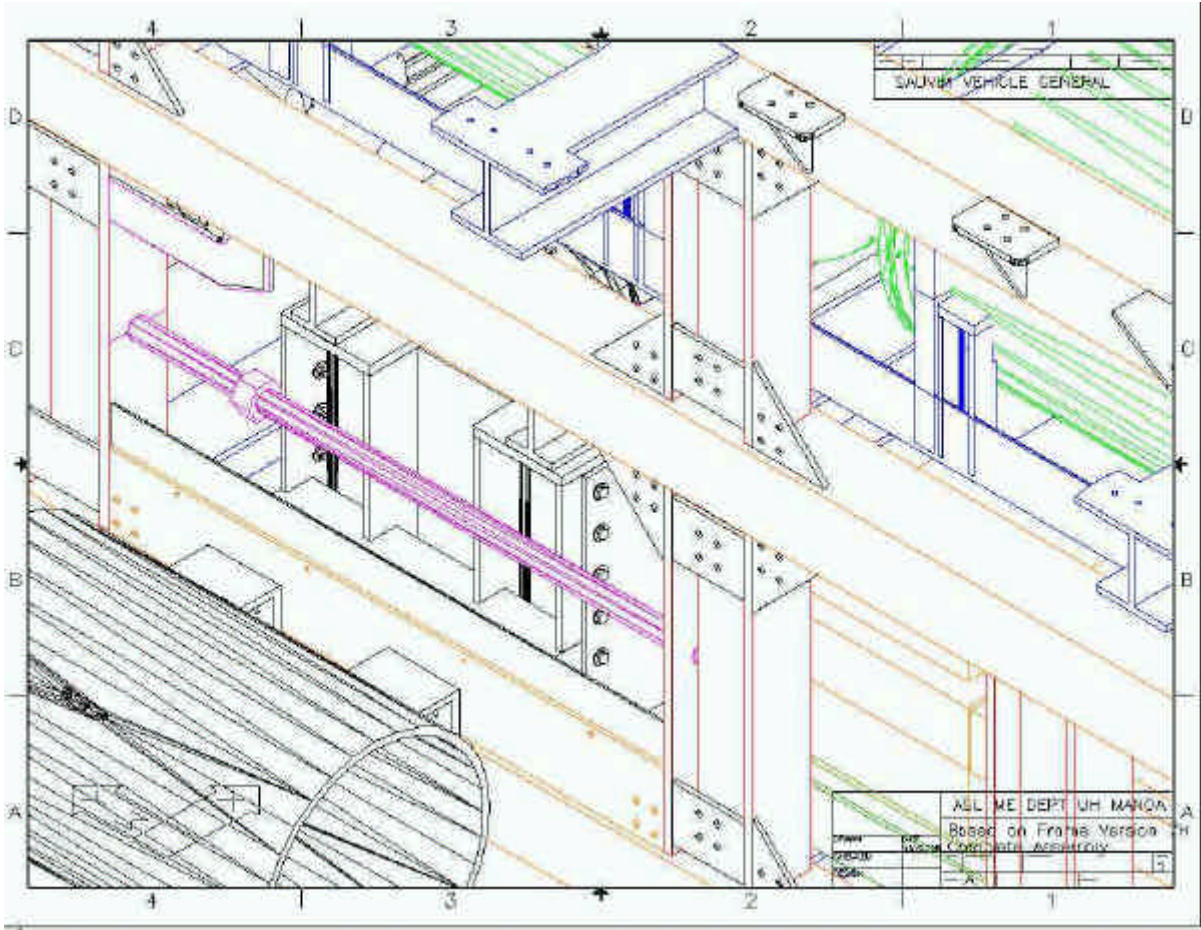


Figure MED-34 - CAD Main Ballast Translation Mechanism.



Figure MED-35: Soft Ballast System Components.

Snug Harbor Floor Plan (08/31/00)
 all dimensions are in inches

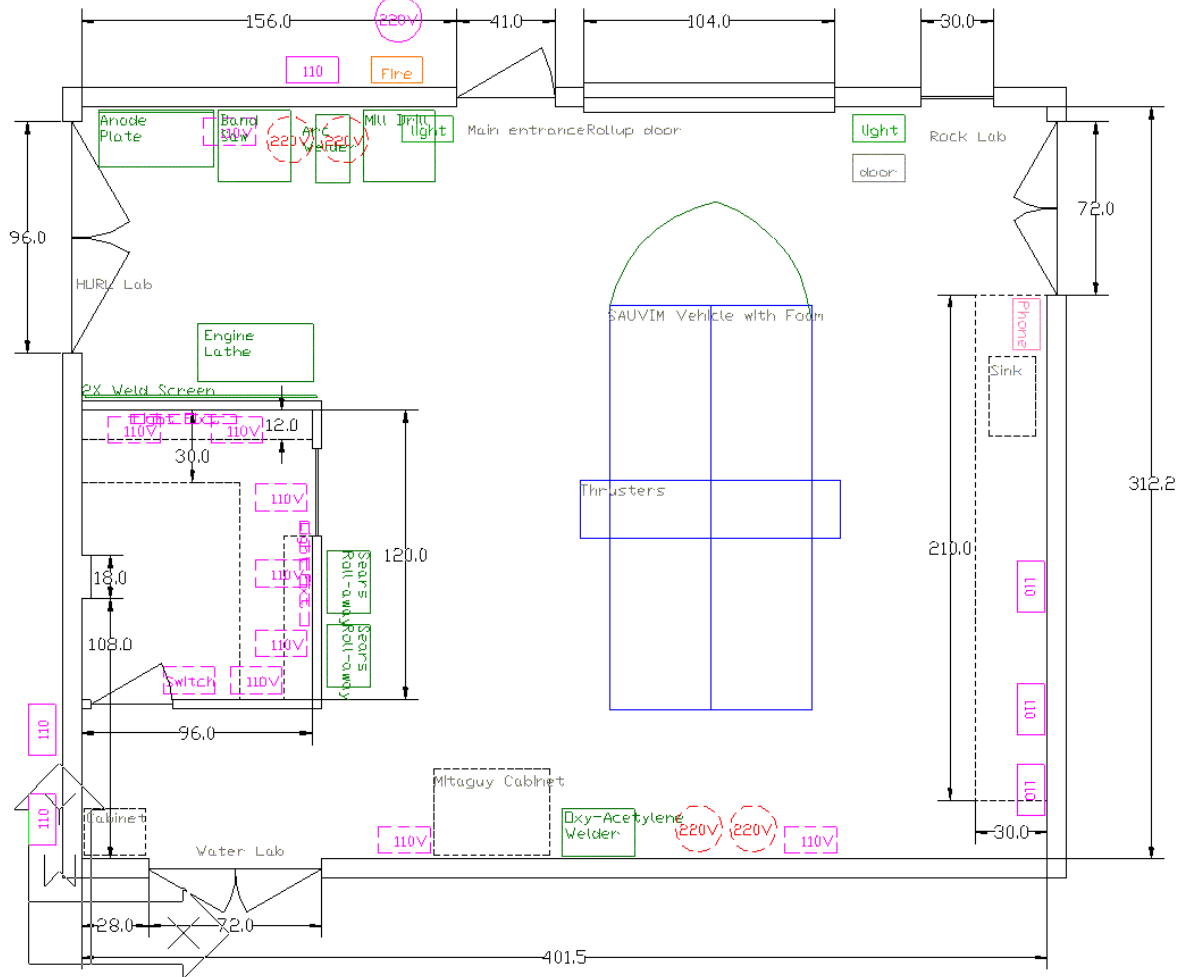


Figure MED-36: Layout of SAUVIM Lab at UH Marine Facility at SNUG Harbor.



Figure MED-37: Machine Tooling to Support SAUVIM Fabrication.



Figure MED-38: High Pressure Test Facility.



Figure MED-39: Detail of High Pressure Test Facility Lid with Electrical Pass-Throughs.



Figure MED-40: Aircraft Pallet Dollies: Before Modification.



Figure MED-41: Finished SAUVIM Trailer from Modified Dolly



Figure MED-42: SAUVIM Testing/Tuning Tank.



Figure MED-43: Electrical Junction Boxes.



Figure MED-44: Camera Hardware and Housing.



Figure MED-45: Tri-tech Ranging Sonar Mount Clamp.



Figure MED-46: Scanning Sonar. Articulating Mount Piece.



Figure MED-47: MSP Main and CTD Sensor Mounts.



Figure MED-48: Battery Straps and Battery in Tray. (tilted view)

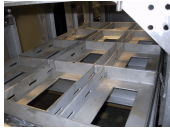


Figure MED-49: Battery Trays in Place on SAUVIM.



Figure MED-50: Underwater Power Cabling.



Figure MED-51: Underwater Data Cabling.



Figure MED-52: Underwater Data Cabling, Close-up View.

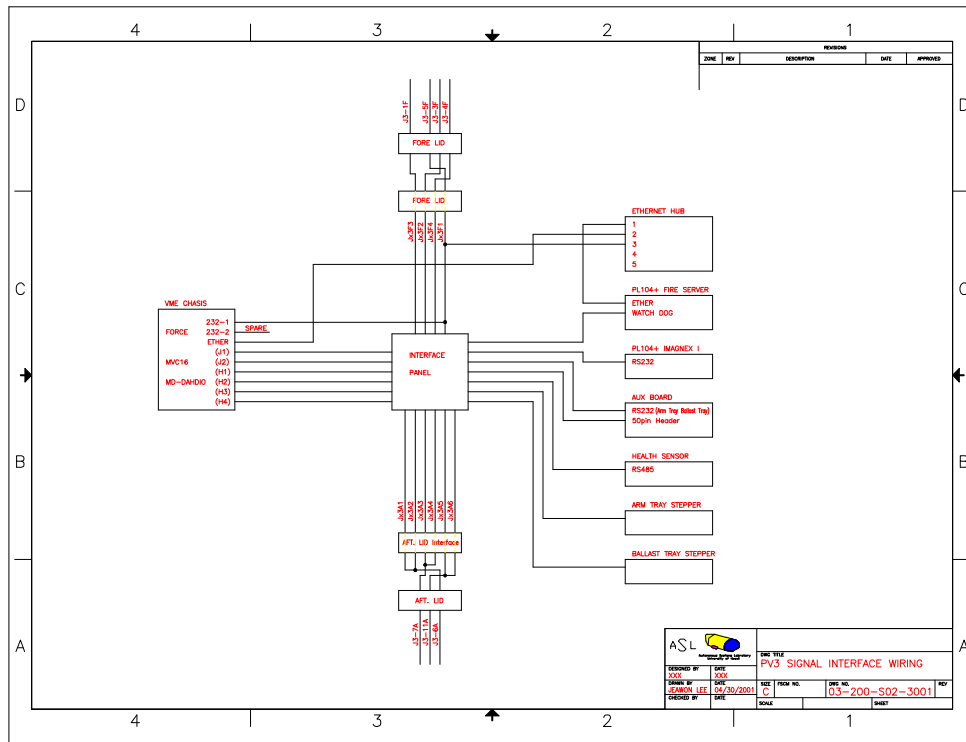


Figure MED-53: PV#3 Signal Routing Diagram.

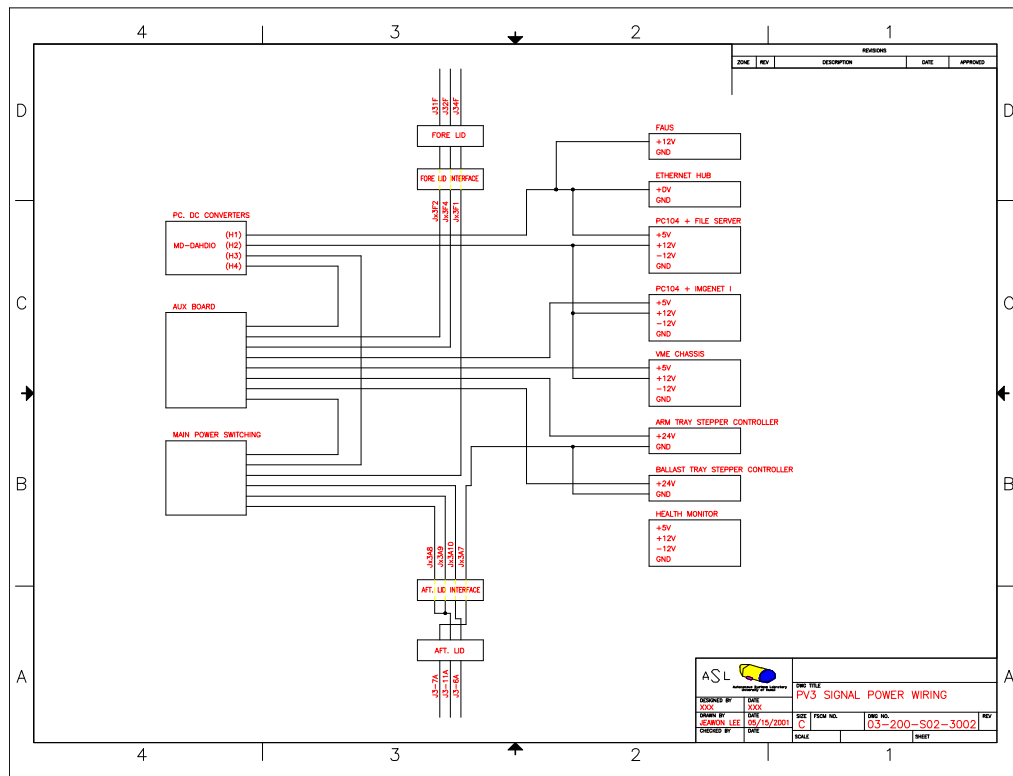


Figure MED-54: PV#3 Power Routing Diagram.

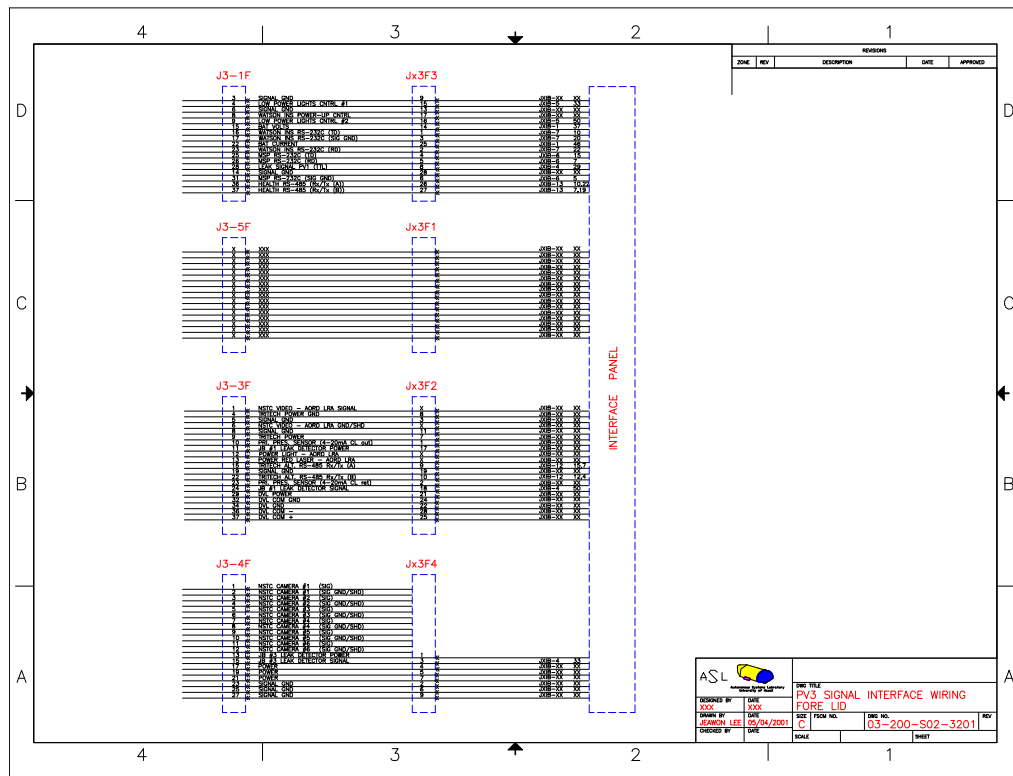


Figure MED-55: Example of PV#3 Cable Breakout to Patch Panel.

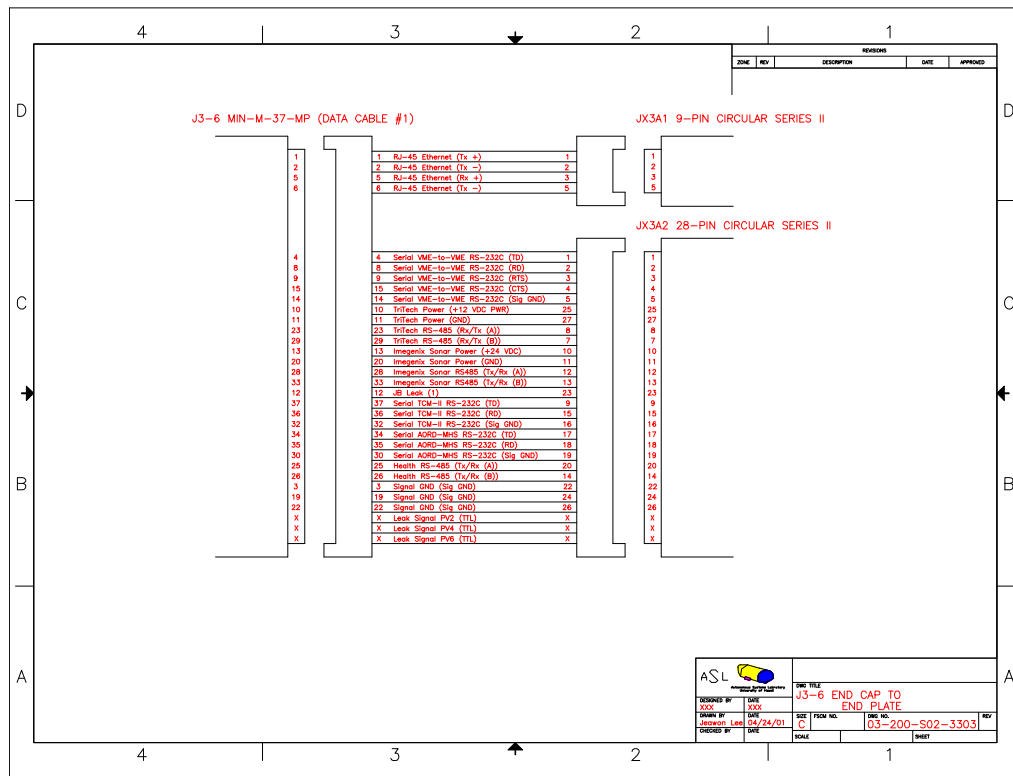


Figure MED-56: Example of a Cable Unit Pin-out within PV#3.



Figure MED-57: Initial Electronics Mounting Mock-up.

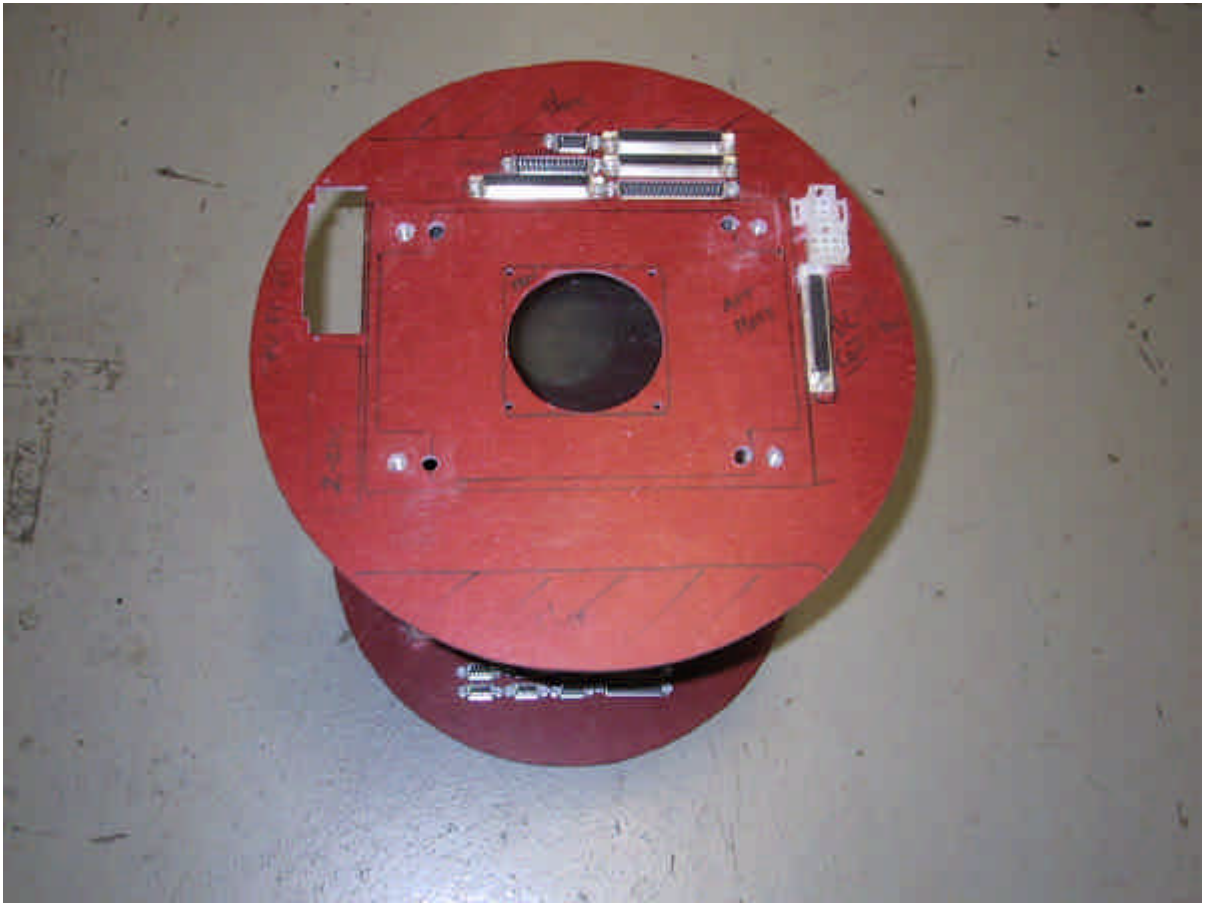


Figure MED-58: First Iteration Electronics Rack Mounting.



Figure MED-59: Working Set of Electronics Racks. Buoy, PV5, PV6, PV3 from lt to right.

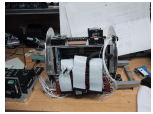


Figure MED-60: Pressure Vessel #3 Electronics Rack, Navigation CPU.

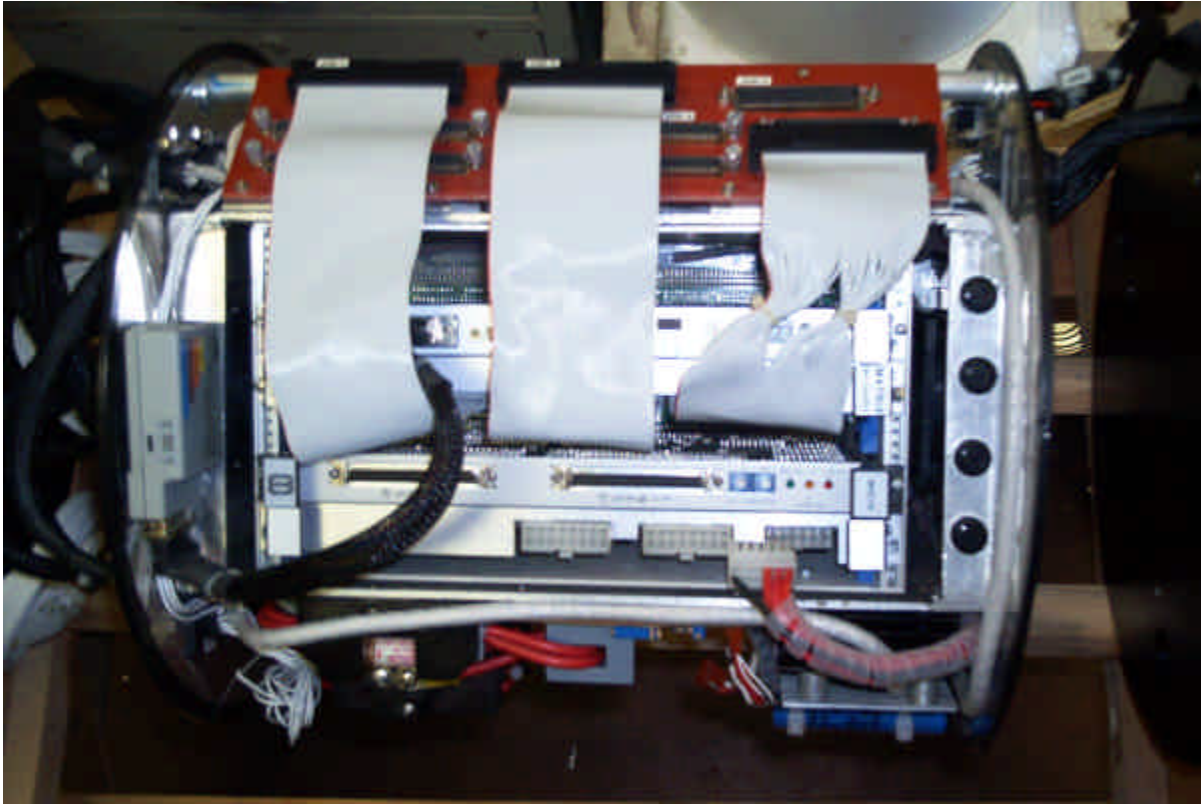


Figure MED-61: PV#3 Navigation Computer Chassis Installed.

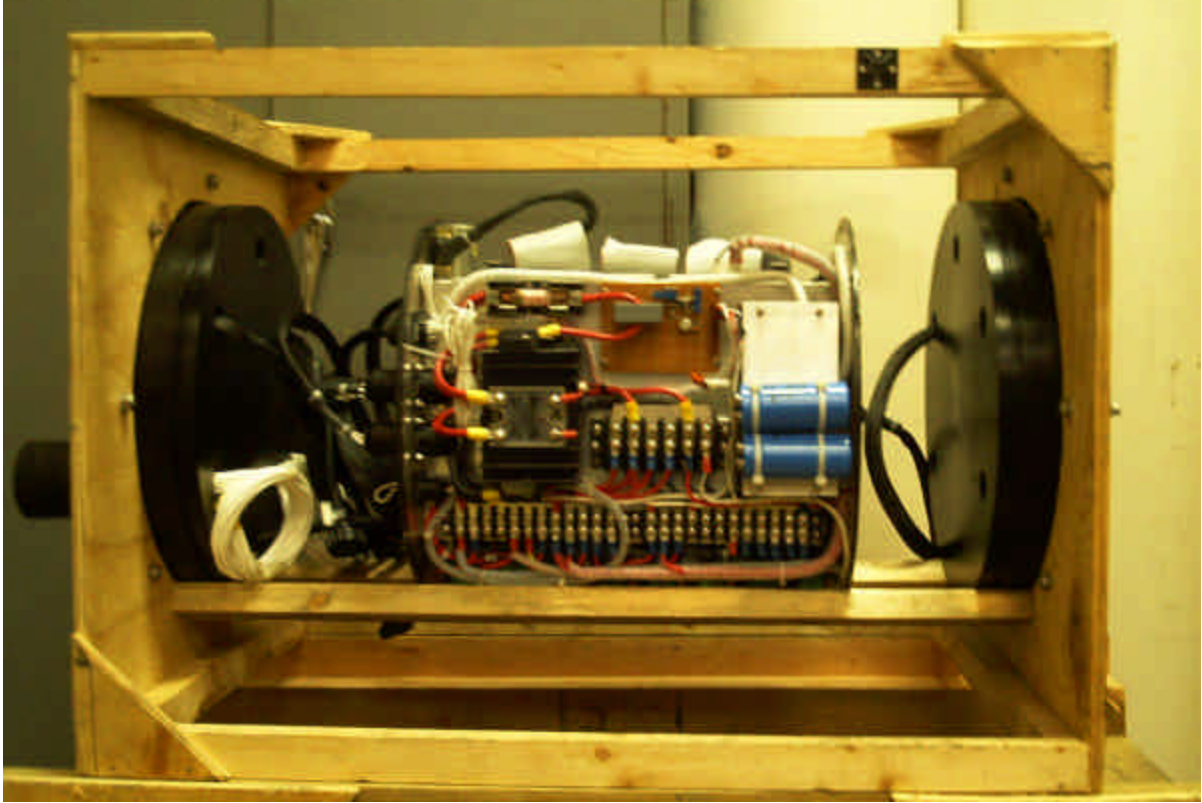


Figure MED-62: Side View of Navigation (PV#3) CPU and Electronics Rack.

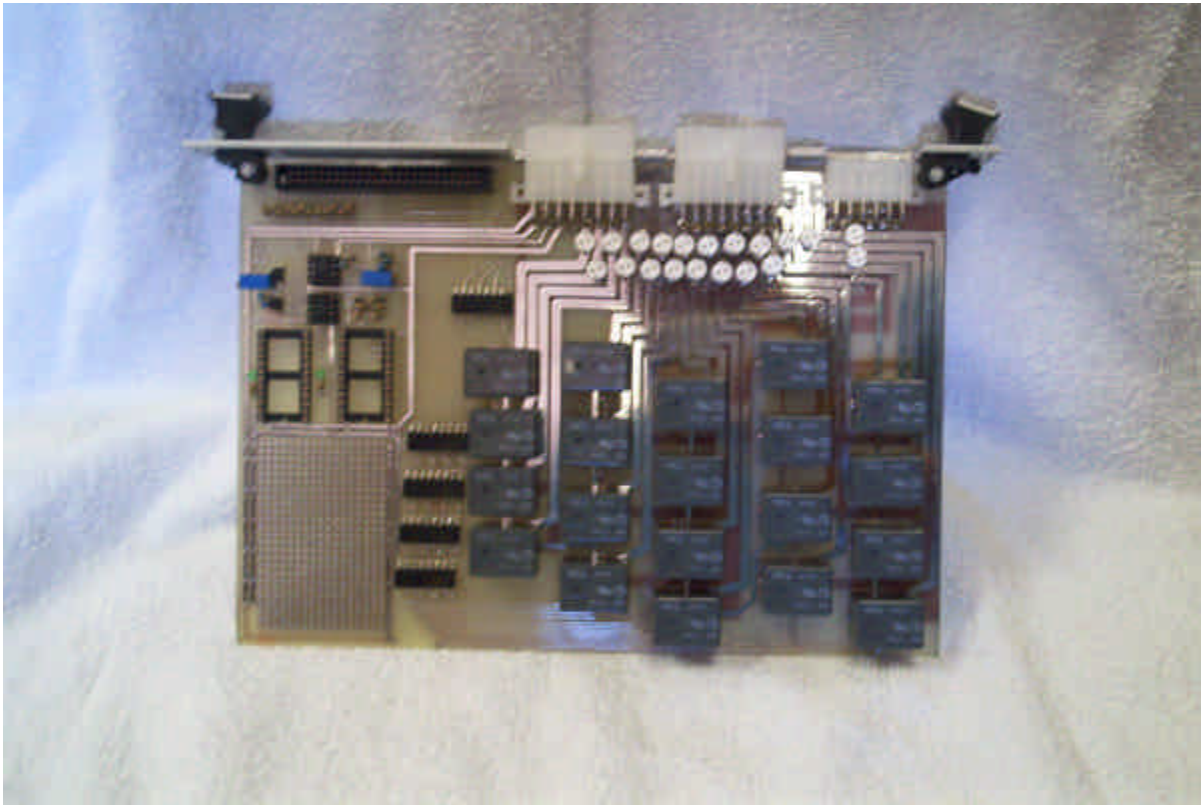


Figure MED-63 - PV#3 Switching/Interface VME Board.

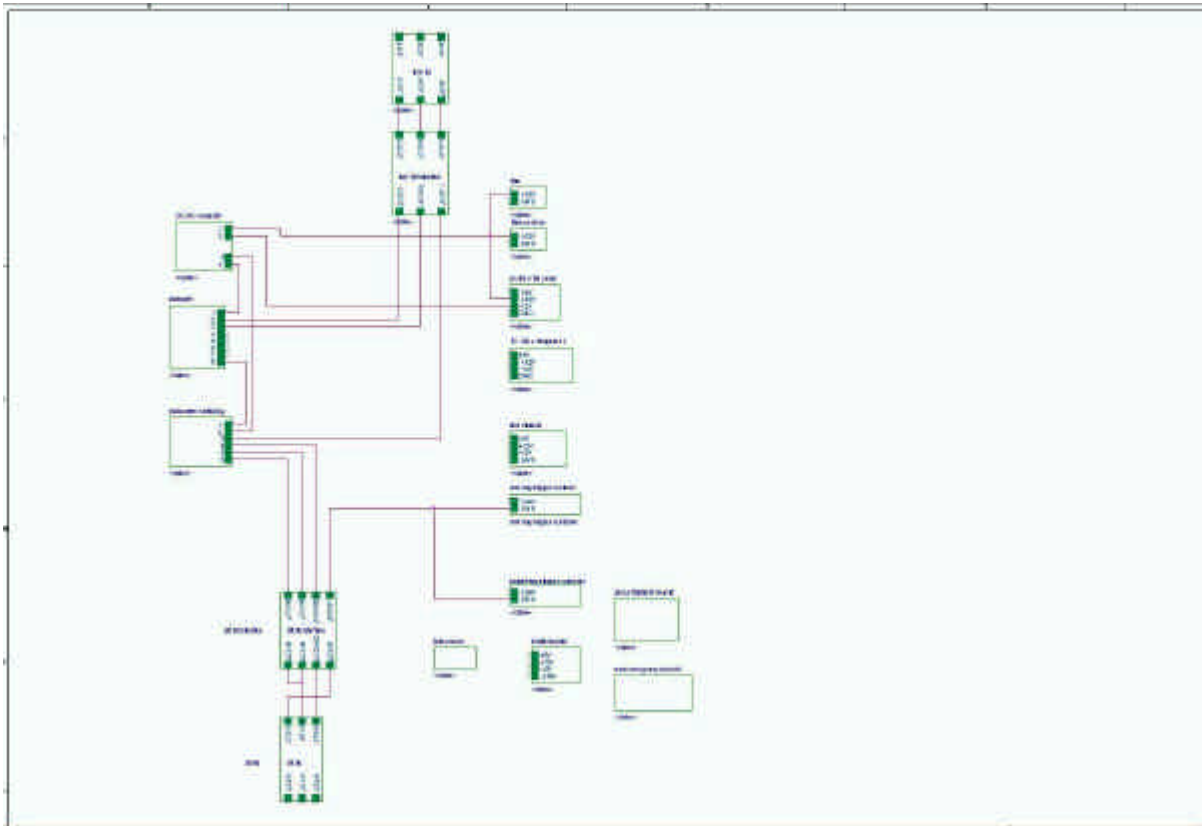


Figure MED-64: Power Switching Hierarchy.

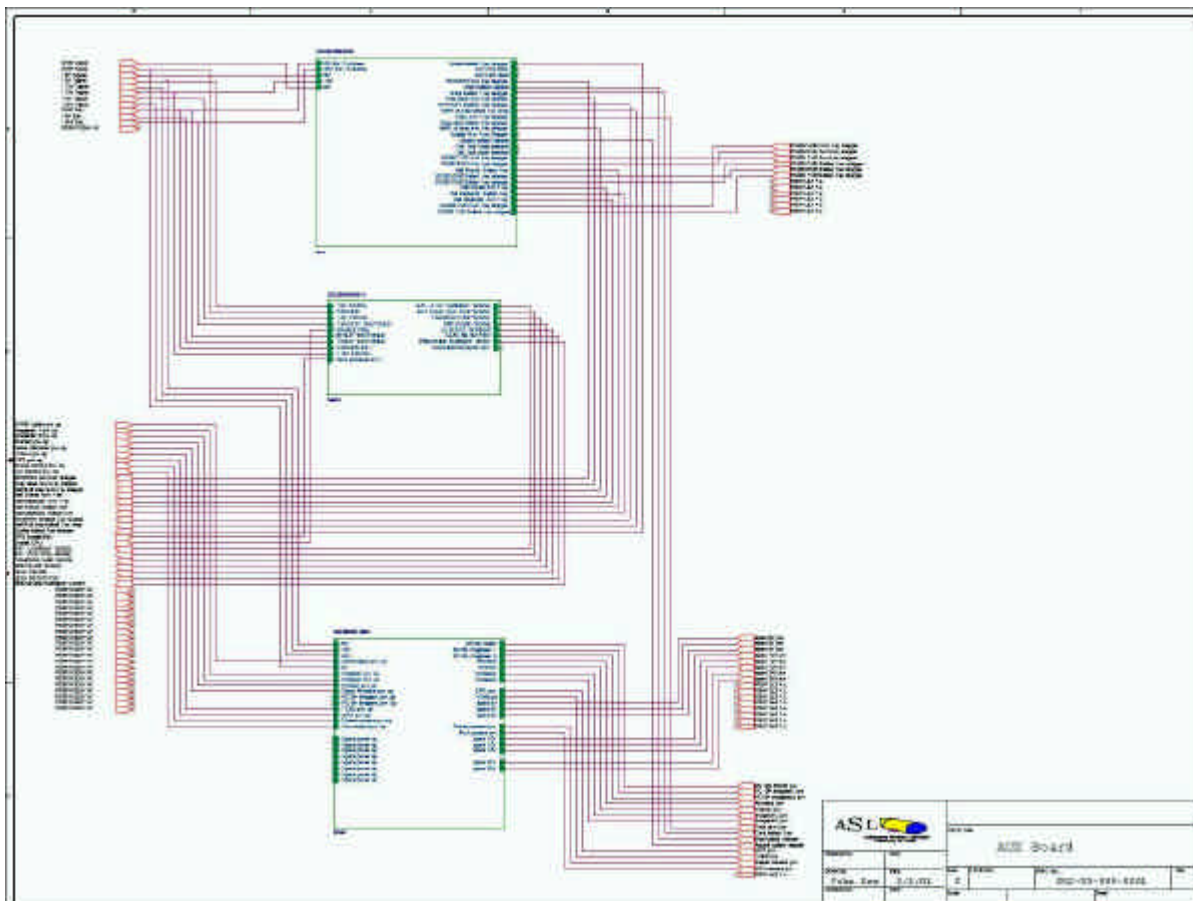


Figure MED-65: Top Level Power Switching Hierarchy.

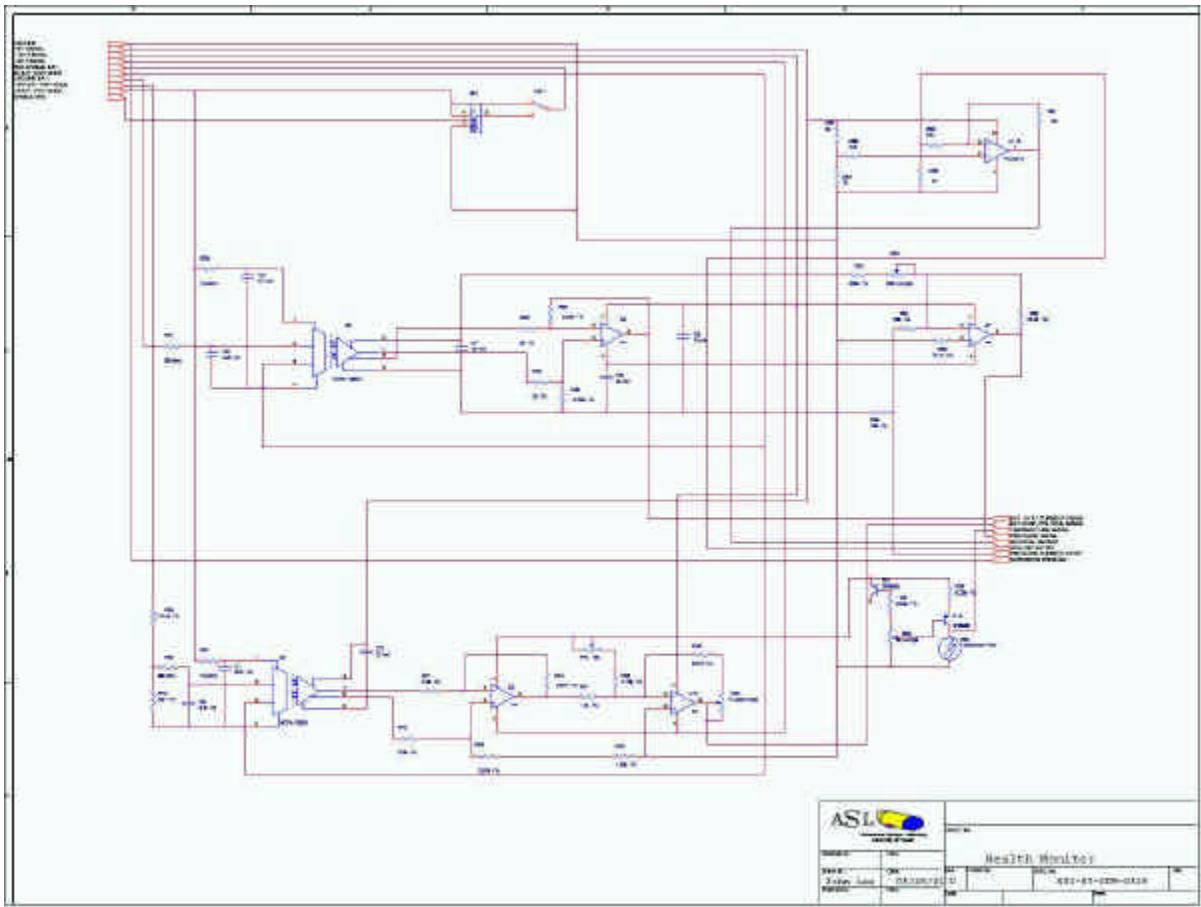


Figure MED-66: Health Monitoring Circuitry.

Figure MED-67: PV#3 Power-up Circuitry.

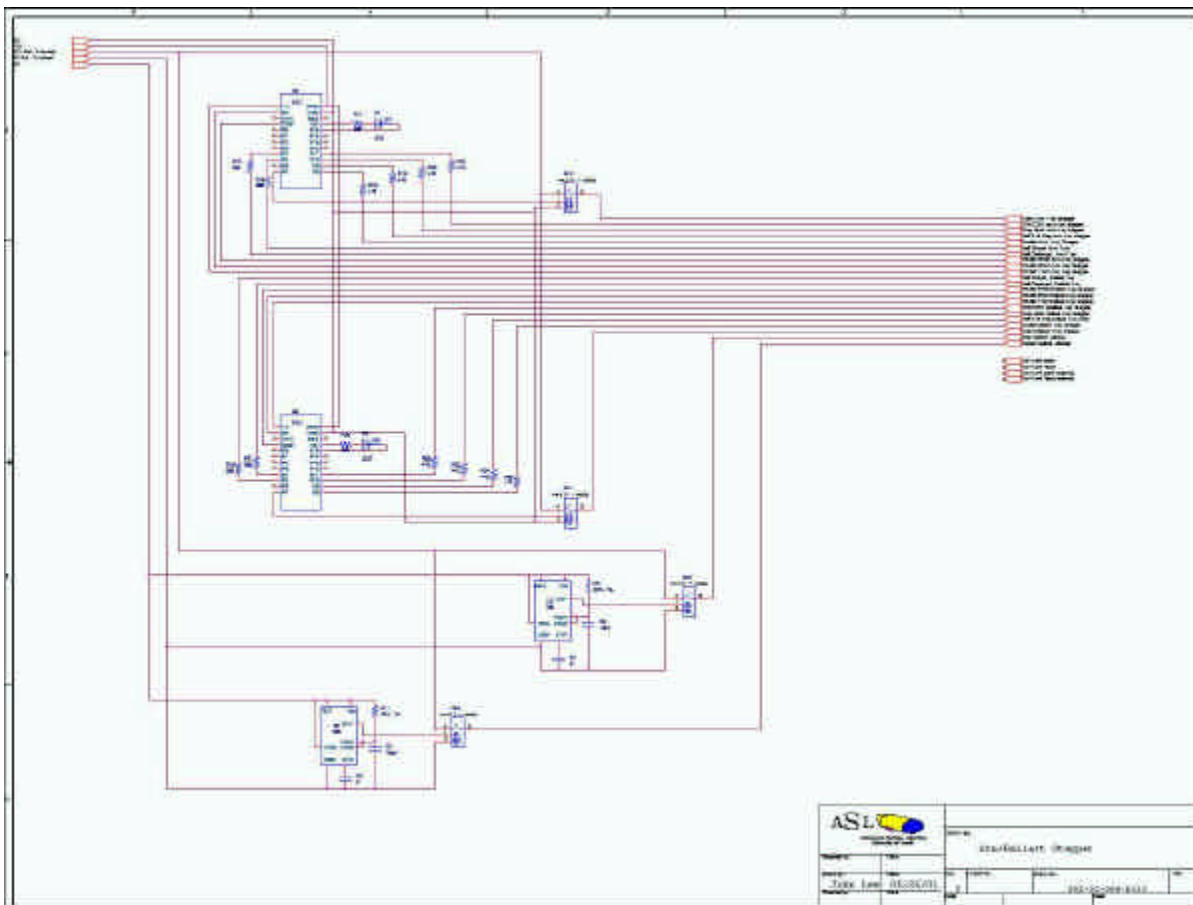


Figure MED-68: PV#3 Arm/Ballast Motor Controller Circuitry.



Figure MED-69: Top View of PV#5 Thruster Controller Electronics.

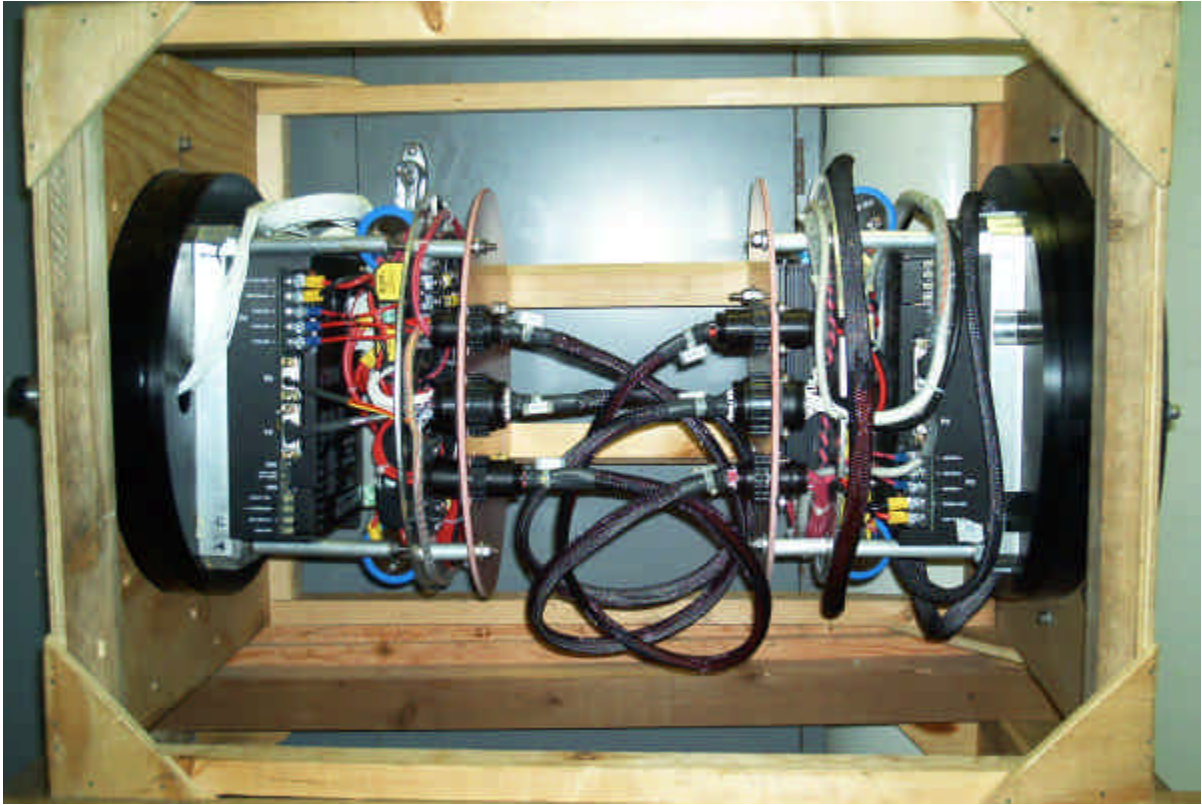


Figure MED-70: Side View of PV#5 Thruster Controller Electronics



Figure MED-71: Top View of PV#6 Thruster Controller Electronics.

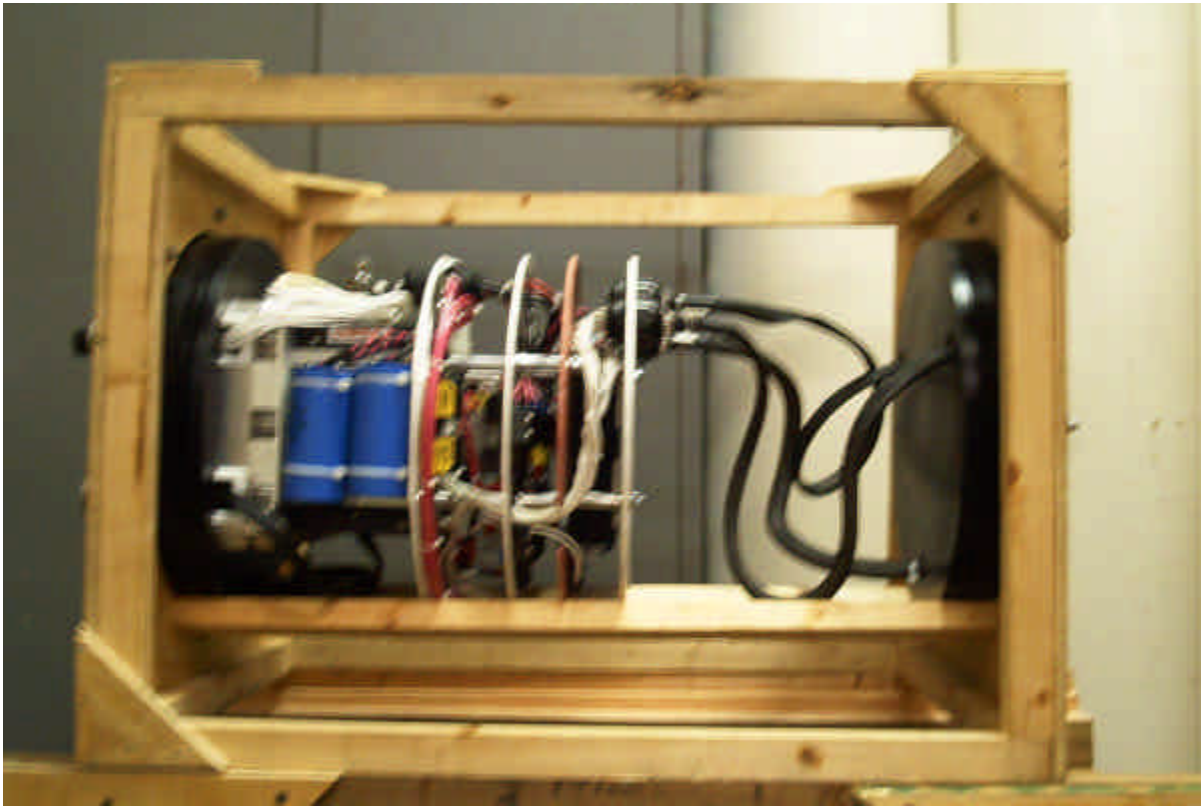


Figure MED-72: Side View of PV#6 Thruster Controller Electronics

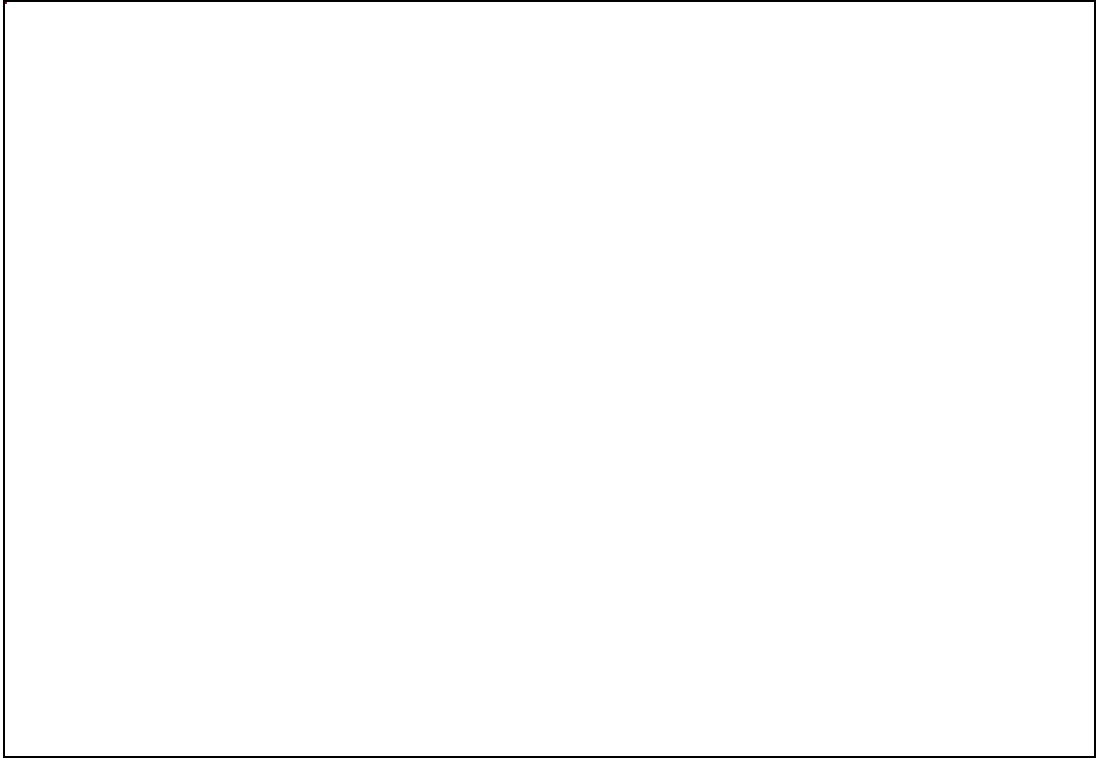


Figure MED-73: Control Box Circuitry

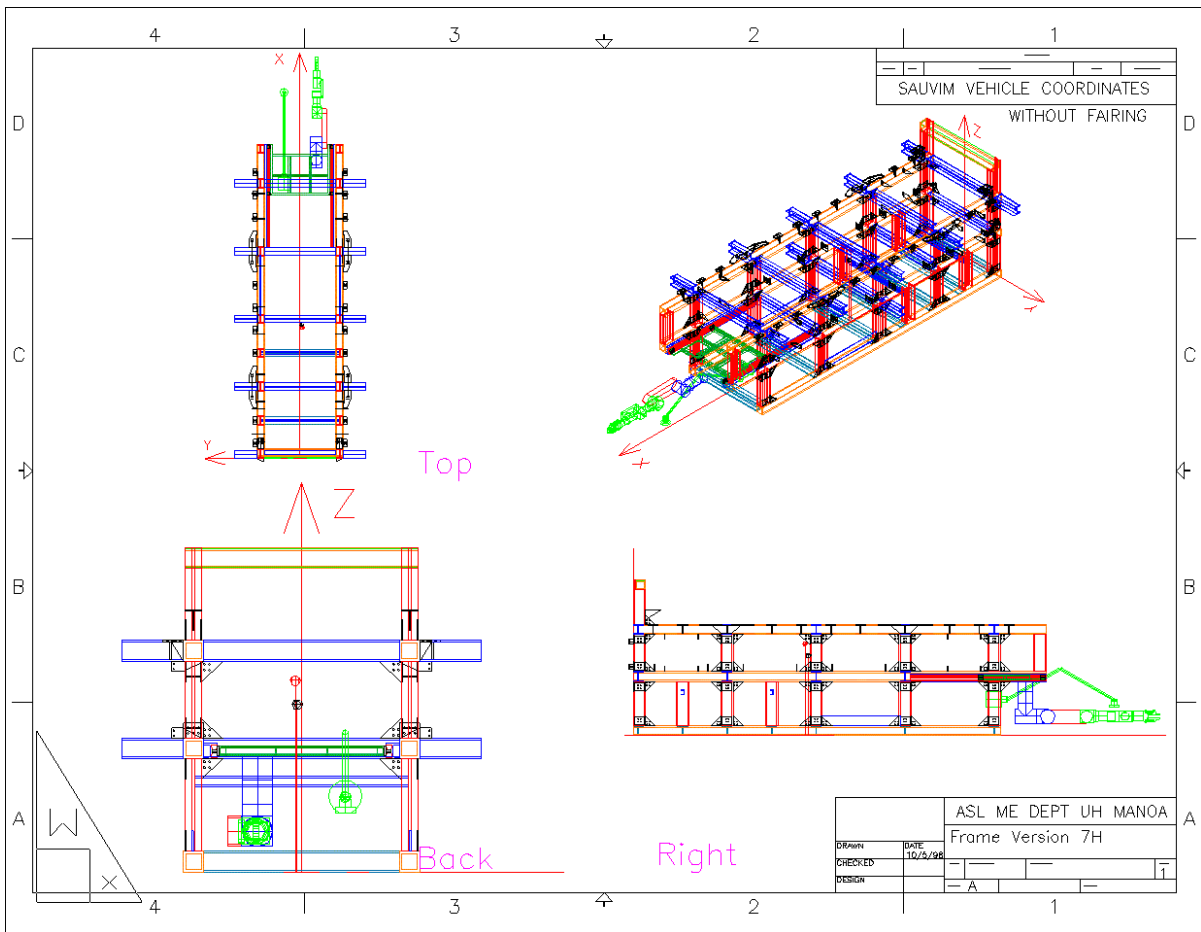


Figure MED-74: Vehicle Frame Schematic

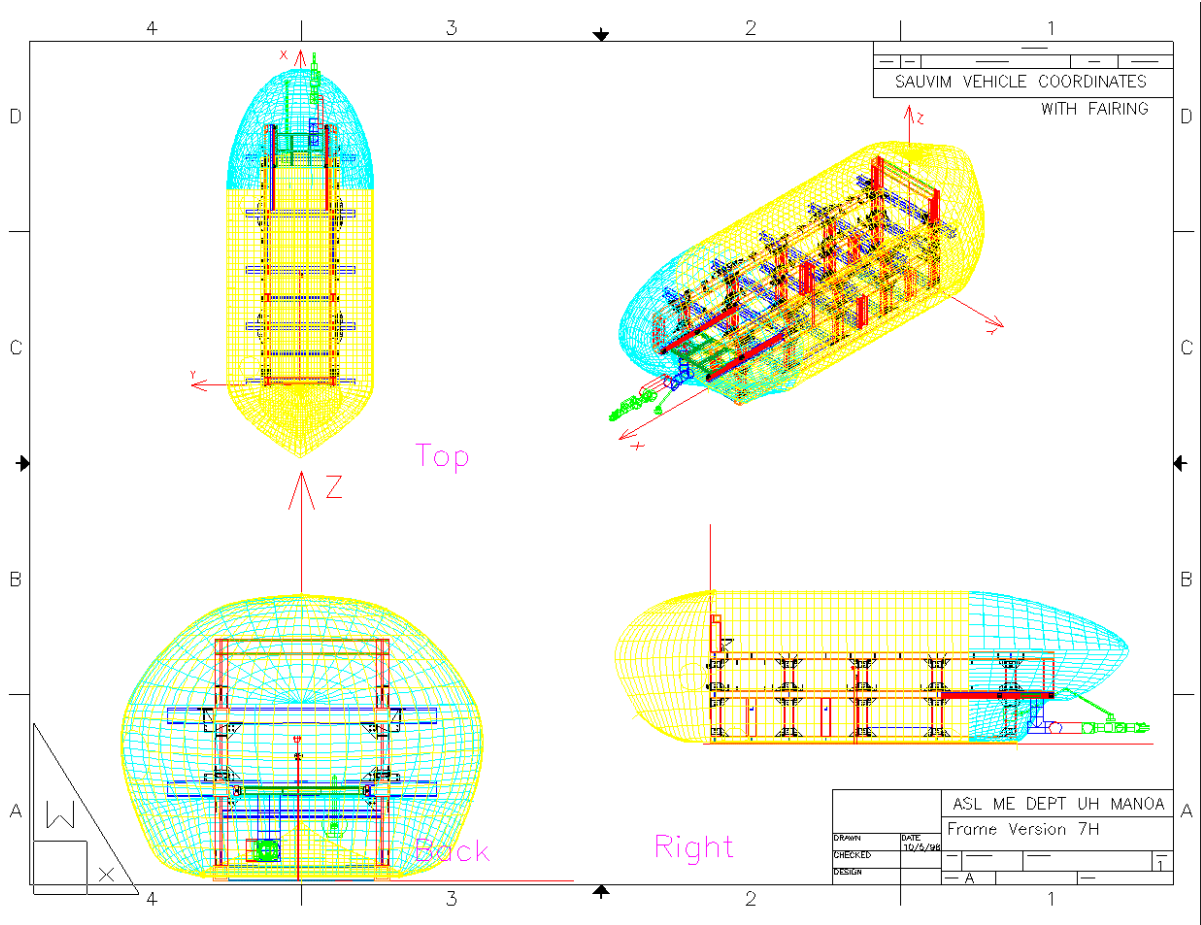


Figure MED-75: Vehicle Frame and Fairing Schematic



Figure MED-76: SAUVIM Deployment



Figure MED-77: SAUVIM Wet-Test



Figure MED-78: SAUVIM Recovery

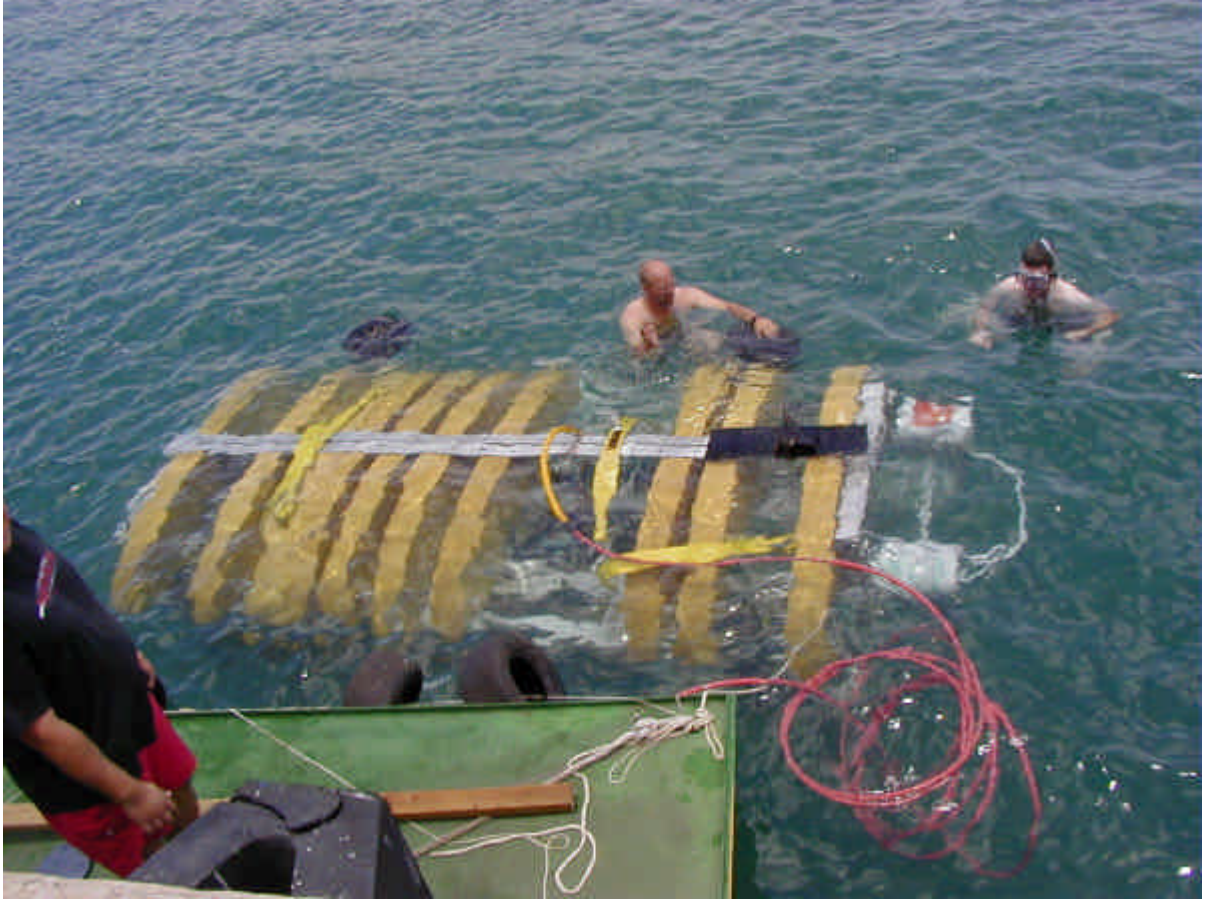


Figure MED-79: SAUVIM Wet Test

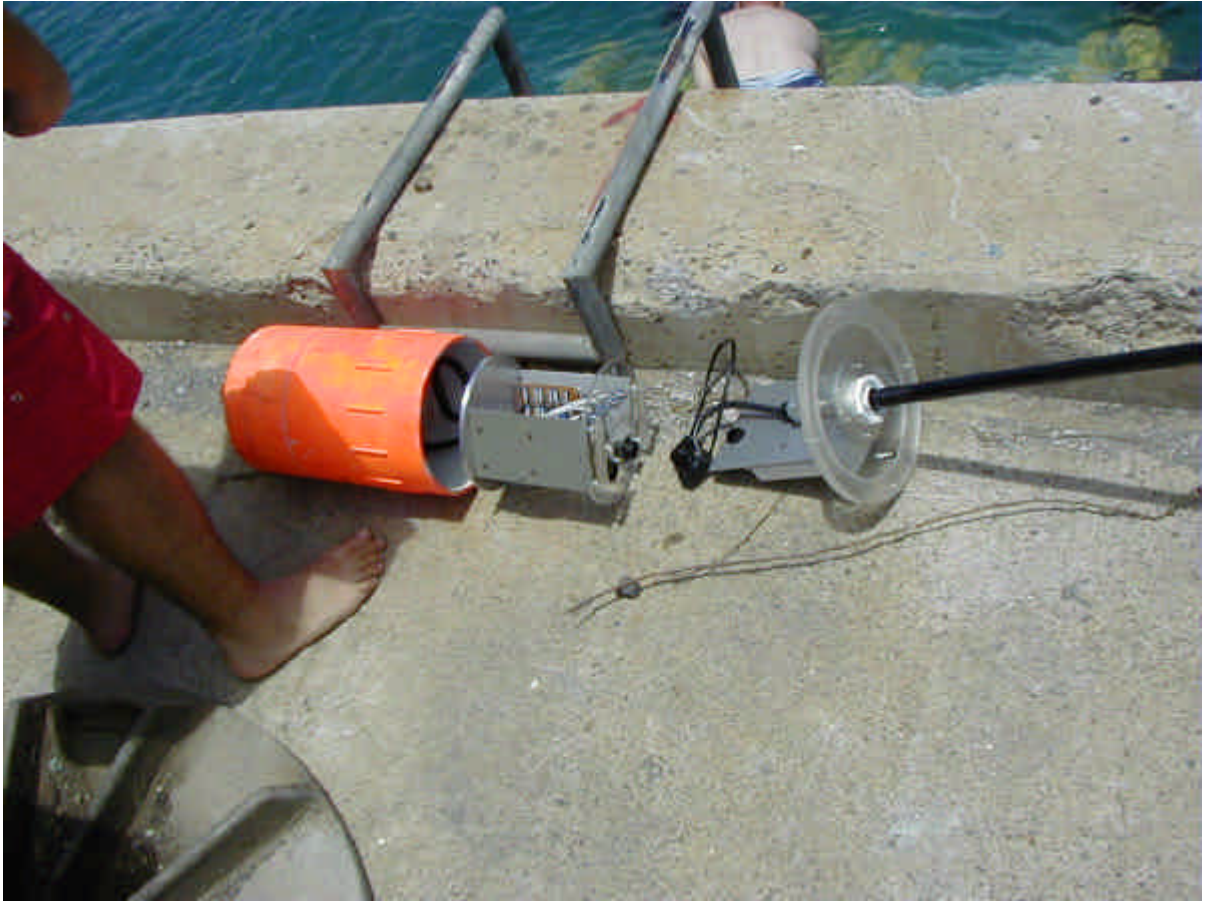


Figure MED-80: SAUVIM Wireless LAN Buoy



Figure MED-81: SAUVIM Analog Joystick Control



Figure MED-82: SAUVIM in Motion



Figure MED-83: SAUVIM in Motion



Figure MED-84: SAUVIM Temporary Ballast Device



Figure MED-85: SAUVIM Temporary Ballast Device

References

- [Aerotech92] Aerotech, "Operation & Technical Manual", 1992.
- [Akiba00] Akiba T and Kakui Yoshimi, "Design and Testing of an Underwater Microscope and Image Processing System for the Study of Zooplankton Distribution", IEEE Journal of Oceanic Engineering, vol. 25, no 1, page 97-104 (2000)
- [Aloimonos97] Aloimonos, Y., Visual Navigation: From Biological Systems to Unmanned Ground Vehicles, Mahwah, NJ, Lawrence Erlbaum Associates, 1997.
- [ANSYS94] ANSYS User's Manual (Version 5.2), Ansys, Inc., Houston, TX, USA, 1994.
- [ANSYS99] ANSYS User's Manual, ANSYS, Inc., Canonsburg, PA, 1999.
- [Antonelli98] Antonelli, G., and S. Chiaverini, "Task-Priority Redundancy Resolution for Underwater Vehicle-Manipulator Systems", in Proceedings of the 1998 IEEE International Conference on Robotics & Automation, pp.756-761, Leuven, Belgium, 1998.
- [Archibald93] Archibald C and Petriu E, "Robot Skills Development Using a Laser Ranger Finder", IEEE 0-7803-1229-5, page 448-452 (1993)
- [Ashby80] Ashby, M. F., and D. R. H., Jones, "Engineering Materials 1: An Introduction to their Properties and Applications", New York: Pergamon Press, 1980.
- [Askeland84] Askeland, D. R., "The Science and Engineering of Materials" California: Wadsworth, Inc., 1984
- [Atari97] Atari A and Dodds G, "Practical Stereo Vision and Multi-Laser Scanning in Object Face Detection and Orientation Determination", IEEE 0-7803-4119-8, page 746-751 (1997)
- [Atari99] Atari A and Dodds G, "Integration of a Stereo Multiple-laser Ranger System and Force Sensor in a Virtual Robotic Environment", IEEE/RSJ Int. Conference on Intelligent Robots and Systems, page 1519-1524 (1999)
- [Auran95] Auran, P.G. and O. Silven, "Ideas for Underwater 3D Sonar Range Sensing and Environmental Modeling", Proceeding of CAMS'95, pp. 284-290, 1995.
- [AUV96] Proceedings of the 1996 Symposium on Autonomous Underwater Vehicle Technology, Monterey, California, 1996.
- [Avallone87] Avallone, E.A, and Baumeister, T., Mark's Standard Handbook for Mechanical Engineers - Ninth Edition, McGraw-Hill Book Co., 1987.
- [Ayache91] Ayache, N., Artificial Vision for Mobile Robots : Stereo Vision and Multisensory Perception, MIT Press, 1991.
- [Beckwith90] Beckwith, T.G., and Maranogoni, R.D., Mechanical Measurements, Addison-Wesley Publishing Company, 1990.
- [Bollinger89] Bollinger, J.G. and N.A. Duffie, Computer Control of Machines and Processes, Reading, MA, Addison-Wesley Publishing Company, 1989.
- [Borland92] Borland C++ User's Guide, Borland International, Inc., 1992.
- [Brush75] Brush, D.O. and B.O. Almroth, Buckling of Bars, Plates, and Shells, New York: McGraw-Hill, Inc., 1975.
- [Brutzman92] Brutzman, D.P., Y. Kanayama & M.J. Zyda, "Integrated Simulation for Rapid Development of Autonomous Underwater Vehicles," Proceedings of the IEEE Oceanic Engineering Society Autonomous Underwater Vehicle 92 Conference, Jun. 1992.
- [Bushnell85] Bushnell, D., "Computerized Buckling Analysis of Shells", Dordrecht: Martinus Nijhoff Publishers, 1985

- [Caimi95] Caimi, F, "Technical Challenges and Recent Developments in Underwater Imaging", Micro-Optics/Micromechanics and Laser Scanning and Shapining, M. Edward Motamedi, Leo Beiser, Editors, Proceedings of SPIE Vol. 2383, 408-418 (1995)
- [Caimi99] Caimi F and Kocak D and Colquitt C, "Design and performances characterization of Simultaneous Reflectance and Surface Mapping laser Scanner for Application in Underwater inspection", in Optical Scanning: Design and Application, Leo Beiser, Stephen F. Sagan, Gerald F. Marshall, Editors, SPIE Vol. 3787, 228-239 (1999).
- [Callister91] Callister, W.D., Materials Science and Engineering, John Wiley and Sons, Inc., 1991.
- [Casalino00] Casalino, G., D.Angeletti, G.Cannata, G. Marani: On the Function and Algorithmic Control Architecture of the AMADEUS Dual Arm Robotic Workcell, SURT 2000, Wailea, Hawaii, June 2000.
- [Chan95] Chan, T.F. and R.V. Dubey, "A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators", IEEE Transaction on Robotics and Automation, vol.11, no.2, pp.286-292, 1995.
- [Chappell99] Chappell, S.C., R.J. Komerska, L. Peng & Y. Lu, "Cooperative AUV Development Concept (CADCON) - An Environment for High-Level Multiple AUV Simulation," Proceedings of the 11th International Symposium on Unmanned Untethered Submersible Technology, Aug. 1999.
- [Chiaverini93] Chiaverini, S. and L. Sciavicco, "The Parallel Approach to Force/Position Control of Robotic Manipulators," IEEE Transaction on Robotics and Automation, vol. 9, pp. 361-373, 1993.
- [Chiaverini97] Chiaverini, S., "Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators," IEEE Transaction on Robotic and Automation, vol. 13, 398-410, 1997.
- [Choi95a] Choi, S.K., J. Yuh, and G.Y. Takashige, "Design of an Omni-Directional Intelligent Navigator, Underwater Robotic Vehicles: Design and Control, TSI Press, pp. 277-297, 1995.
- [Choi95b] Choi, S.K. and J. Yuh, "Development of an Omni-Directional Intelligent Navigator", IEEE Robotics and Automation Magazine, 1995.
- [Choi95c] Choi, S.K., G.Y. Takashige & J. Yuh, "Development of an Omni-Directional Intelligent Navigator," IEEE Robotics and Automation Magazine on Mobile Robots, Mar. 1995.
- [Choi96] Choi, S.K., and J. Yuh, "Experimental Study on a Learning Control System with Bound Estimation for Underwater Robots", International Journal of Autonomous Robots, 3 (2 & 3), pp. 187-194, 1996.
- [Clayton82] Clayton, B.R. and Bishop, R.E.D., Mechanics of Marine Vehicles, E. & F.N. Spon Ltd., 1982.
- [Comstock67] Comstock, J.P., Principles of Naval Architecture, Society of Naval Architects and Marine Engineers, 1967.
- [Coz74] Cox, A.W., Sonar and Underwater Sound, Lexington Books, 1974.
- [Craig86] Craig, J.J, Introduction to Robotics: Mechanics and Control, Reading, MA, Addison-Wesley, 1986.
- [Crawford98] Crawford A and Hay A, "A Simple System for Laser-Illuminated Video Imaging of Sediment Suspension and Bed Topography", IEEE Journal of Oceanic Engineering, vol. 23, no 1, page 12-19 (1998)
- [Cristi96] Cristi, R., M. Caccia, G. Veruggio and A.J. Healey, "A Sonar Based Approach to AUV Localization", Proceeding of CAMS'95, pp. 291-298, 1996.

- [Cunha95] Cunha, J.P., R.R. Costa, and L. Hsu, "Design of a High Performance Variable Structure Position Control of ROV's," IEEE Journal of Oceanic Engineering, vol. 20, no. 1, pp.42-55, 1995.
- [D'egoulange94] D'egoulange, E. and P. Dauchez, P., "External Force Control of an Industrial PUMA 560 Robot," journal of Robotic Systems, vol. 11, pp. 523-540, 1994.
- [de Wit96] Candus de Wit, C., B. Siciliano, and G. Bastin (Editors), Theory of Robot Control, Springer-Verlag, Berlin, Germany, 1996.
- [DeBitetto94] DeBitetto, P.A., "Fuzzy Logic for Depth Control of Unmanned Undersea Vehicles," Proceedings of the Symposium of Autonomous Underwater Vehicle Technology, pp. 233-241, 1994.
- [Deepsea90] UnderPressure Software Manual, DeepSea Power and Light Co., 1990.
- [Doebelin75] Doebelin, E.O., Measurement Systems: Application and Design, New York, McGraw-Hill Book Company, 1975.
- [Dote90] Dote, Y., Servo Motor and Motion Control Using Digital Signal Processors, Englewood Cliffs, NJ, Prentice-Hall Publishing, 1990.
- [Dougherty90] Dougherty, F. and G. Woolweaver, "At-Sea Testing of an Unmanned Underwater Vehicle Flight Control System," Proceedings of the Symposium of Autonomous Underwater Vehicle Technology, pp. 65-73, 1990.
- [Dunningan96] Dunningan, M.W., D.M. Lane, A.C. Clegg, and I. Edwards, "Hybrid Position/Force Control of a Hydraulic Underwater Manipulator," IEEE Proceedings Control Theory and Application, vol. 143, no. 2, pp. 145-151, March 1996.
- [Englemann95] Englemann, W.H., Handbook of Electric Motors, M. Dekker, 1995.
- [Evans96] Evans, A.J., Basic Digital Electronics - Digital System Circuits and Their Functions, Master Publishing Inc., 1996.
- [Ferrerri97] Ferrerri, G., G. magnani, and P. Rocco, "Toward the Implementation of Hybrid Position/Force Control in Industrial Robots," IEEE Transaction on Robotics and Automation, vol. 16, pp. 838-845, 1997.
- [Fox88] Fox J, "Structured light imaging in turbid water", Underwater Imaging, Douglas J. Holloway, Editor, SPIE vol. 980, page 66-71 (1988).
- [Fox92] Fox, R.W. and McDonald, A.T., Introduction to Fluid Mechanics, John Wiley and Sons, Inc., 1992.
- [Gere90] Gere, J.M. and Timoshenko, S.P., Mechanics of Materials, PWS-Kent Publishing Co., 1990.
- [Geyer77] Geyer, R. A., Submersibles and Their Use in Oceanography and Ocean Engineering, Amsterdam, Elsevier Scientific Pub. Co., 1977.
- [Gill85] Gill, R., Electrical Engineering Handbook, Siemens Co., 1985.
- [Goheen98] Goheen, K.R., and R.E. Jeffery, "Multivariable Self-Tuning Autopilots for Autonomous and Remotely Operated underwater Vehicles", IEEE Journal of Oceanic Engineering, vol. 15, pp.144-151, 1990.
- [Goldberg89] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.
- [Harding97] Harding, K.G. and D.J. Svetkoff (chairs/editors), Three-dimensional Imaging and Laser-based Systems for Metrology and Inspection III (Pittsburgh, PA), International Society for Optical Engineering, Bellingham, Washington, 1997.
- [Healy92] Healy, A.J. and D.B. Macro, "Slow Speed Flight Control of Autonomous Underwater Vehicles: Experimental Results with NPS AUV II", Proc. of ISOPE, pp. 523-532, 1992.

- [Healy93] Healy, A.J. and D. Lienard, "Multi-variable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles," IEEE Journal of Oceanic Engineering, vol. 18, no. 3, pp. 327-339, 1993.
- [Hibbeler92] Hibbeler, R.C., Engineering Mechanics, Macmillian Publishing Co., 1992.
- [Hill70] Hill, P.G. and Peterson, C.R., Mechanics and Thermodynamics of Propulsion, Addison Wesley Publishing Co., 1970.
- [Hoerner65] Hoerner, S.F., Fluid Dynamic Drag: Practical Information on Aerodynamic and Hydrodynamic Resistance, American Institute of Aeronautics and Astronautics, 1965.
- [Hollerback87] Hollerbach, J.M. and K.C. Suh, "Redundancy Resolution of Manipulator Through Torque Optimization", IEEE Journal of Robotics and Automation, vol RA-3, No.4, pp. 308-316, 1987.
- [Holman89] Holman, J.P. and W.J. Gajda, Jr., Experimental Methods for Engineers, New York, McGraw-Hill Book Company, 1989.
- [Howard86] Howard, G.. Automobile Aerodynamics: Theory and Practice for Road and Track, Motorbooks International, 1986.
- [Hsu94] Hsu, L., R. Costa, and F. Lizarralde, "Underwater Vehicle Dynamic Positioning Based on a Passive Arm Measurement System", International Advanced Robotics Programme, pp. 23-32, 1994.
- [Hudson99] Hudson, J. and Luecke J., Basic Communications Electronics, Master Publishing, Inc., 1999.
- [Hueber95] Huebner, K. H., E. A., Thornton, and T. G., Byrom, "The Finite Element Method for Engineers," New York: John Wiley & Sons, Inc., 1995.
- [Hughes94] Hughes, A., Electric Motors and Drives - Fundamentals, Types and Applications, BH Newnes, 1994.
- [Hull83] Hull, D., "Axial Crusing of Fibre Reinforced Composite Tubes," Structural Crashworthiness, Eds. N. Jones and T. Wierzbicki, Butterworth., pp. 118-135, 1983.
- [Hyer88] Hyer, M. W., "Respond of Thick Laminate Cylinders to External Hydrostatic Pressure," Journal of Reinforced Plastics and Composites, vol. 7, pp. 321-340, 1988.
- [ICI Thermoplastic Composite92] ICI Thermoplastic Composite, "Thermoplastic Composite Handbook," 1992.
- [Incropera85] Incropera, F.P. and DeWitt, D.P., Introduction to Heat Transfer, John Wiley and Sons, 1985.
- [Ishii94] Ishii, K., T. Fujii, and T. Ura, "A Quick Adaptation Method in Neural Network Based Control System for AUVs," Proceedings of the Symposium of Autonomous Underwater Vehicle Technology, pp.269-274, 1994.
- [Jones92] Jones, D. A., "Principle and Prevention of Corrosion," New York: Macmillian Publishing Company, 1992.
- [Kajita97] Kajita, H. and K. Kosuge, "Force Control of Robot Floating on the Water Utilizing Vehicle Restoring Force," Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems, vol.1, pp. 162-167, 1997.
- [Kato93] Kato, N., Y. Ito, K. Asakawa, and Y. Shirasaki, "Guidance and Control of Autonomous Underwater Vehicle AQUA Explorer 1000 for Inspection of Underwater Cables", Proc. 8th Int. Symposium on Unmanned, Untethered Submersible Technology, Sept. 1993.

- [Kawaguchi96] Kawaguchi, K., C. Ikehara, S.K. Choi, M. Fujita, and J. Yuh, "Design of an Autonomous Underwater Robot: ODIN II," World Automation Congress, Montpellier, France, May 1996.
- [Kernighan78] Kernighan, B.W. and D.M. Ritchie, The C Programming Language, Englewood Cliffs, NJ, Prentice-Hall, Inc., 1978.
- [Klafter83] Klafter, R. D., Robotic Engineering: an Integrated Approach, Prentice Hall, 1989.
- [Klein83] Klein, C.A. and C.S. Huang, "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," IEEE Trans. on System, Man, and Cybernetics, vol. SMC-13, pp. 245-250, 1983.
- [Kocak99] Kocak D and Lobo N and Widder E, "Computer Vision Techniques for Quantifying, Tracking, and Identifying Bioluminescent Plankton", IEEE Journal of Oceanic Engineering, vol. 24, no 1, page 81-95 (1999)
- [Kochin65] Kochin, N.E., I.A. Kibel, and N.V. Rose, Theoretical Hydrodynamics, John Wiley & Sons, 1965.
- [Krar67] Krar, S.F., and Amand, J.E., Machine Shop Training, McGraw-Hill Co., 1967.
- [Kuroda95] Kuroda, Y., K. Aramaki, T. Fujii & T. Ura, "A Hybrid Environment for the Development of Underwater Mechatronic Systems," Proceedings of the 1995 IEEE 21st International Conference on Industrial Electronics, Control, and Instrumentation, Nov. 1995.
- [Lamb45] Lamb, H., Hydrodynamics, Dover, 1945.
- [Lander87] Lander, C.W., Power Electronics, McGraw-Hill, 1987.
- [Lane99] Lane D and Davies: and Robinson G and O'Brien D and Sneddon J and Seaton E and Elfstrom Anders, "The AMADEUS Dextrous Subsea Hand: Design, Modeling, and Sensor Processing", IEEE Journal of Oceanic Engineering, vol. 24, no 1, page 96-111 (1999)
- [Lawry90] Lawry, M.H., I-DEAS Student Guide, Structural Dynamics Research Corp., 1990.
- [Leon95] Leon, G. F. and J. C. Hall, "Case Study-Design and Testing of the Brunswick Graphite Epoxy Composite Ring-Stiffened Thermo set Cylinder," Journal of Thermoplastic Composite Materials, vol. 8., 1995.
- [Lewis84] Lewis, D.J., J.M. Lipscomb, and P.G. Thompson, "The simulation of Remotely Operated Underwater Vehicle", Proceeding of ROV 1984, pp. 245-252, 1984.
- [Lewis89] Lewis, E.V., Principles of Naval Architecture, Jersey City, NJ, Society of Naval Architects and Marine Engineers, 1988-1989.
- [Liegeois77] Liegeois, A., "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-7, No.2, pp.868-871, 1977.
- [Lines97] Lines, D., Building Power Supplies - Useful Designs for Hobbyists and Technicians, Jerry Leucke Master Publishing Inc., 1997.
- [LS-DYNA99] LS-DYNA User's Manual, Livermore Software Technology Corporation, Livermore, CA, 1999.
- [Luanglat97] Luanglat, C. S., and M. N Ghasemi Nejjhad., "A Crash Simulation Study of Composite Materials and Structures for Electric and Hybrid Vehicles," 14th International Electric Vehicle Symposium, Proceedings, pp. 14-17, 1997.
- [Lundgren99] Lundgren, J., and P. Gudmundson, "Moisture Absorption in Glass-Fiber/Epoxy Laminates with Transverse Matrix Cracks," Composites Science and Technology, vol. 59, no. 13, pp. 1983-1991, 1999.

- [Mahesh91] Mahesh, H., J. Yuh, and R. Lakshmi, "A Coordinated Control of an Underwater Vehicle and Robot Manipulator", *Journal of Robotic Systems*, Vol.8, No.3, pp.339-370, 1991.
- [Mallick93] Mallick, P. K., "Fiber-reinforced Composites: Materials, Manufacturing, and Design," New York: Marcel Dekker, Inc., 1993.
- [Marco96] Marco, D.B., *Autonomous Control of Underwater Vehicles and Local Area Maneuvering*, Ph.D. Dissertation, Naval Postgraduate School, 1996.
- [Martini84] Martini, L.J., *Practical Seal Design*, Marcel Dekker, Inc., 1984.
- [Mattsson89] Mattsson, E., *Basic Corrosion Technology for Scientists and Engineers*, Ellis Horwood Ltd., 1989.
- [McLain96] McLain, T.W., S.M. Rock, and M.J. Lee, "Experiments in the Coordinated Control of an Underwater Arm/Vehicle System", *Autonomous Robots* 3, pp. 213-232, Kluwer Academic Publisher, Netherlands, 1996.
- [McMillan95] McMillan, D.O., and R. McGhee, "Efficient Dynamic Simulation of an Underwater Vehicle with a Robotic Manipulator," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.25, No.8, pp.1194-1206, August, 1995.
- [Microsoft88a] *Microsoft QuickBASIC - Learning to Use Microsoft QuickBASIC*, Microsoft Corp., 1988.
- [Microsoft88b] *Microsoft QuickC - Learning to Use Microsoft QuickC*, Microsoft Corp., 1988.
- [Milne-Thomson68] Milne-Thomson, L., *Theoretical Hydrodynamics*, Macmillan, 1968.
- [Mims98] Mims, F., *Getting Started in Electronics*, Radioshack Co., 1998.
- [Mullen99] Mullen L and Contarino M and Laux A and Concannon: and Davis J and Strand M and Coles B, "Moduladet Laser Line Scanner for Enhanced Underwater Imaging", *Airborne and In-water Underwater Imaging*, Gary D. Gilbert, Editor, SPIE vol. 3761, page 2-9 (1999).
- [Nakamura85] Nakamura, Y., and H. Hanafusa, "Task Priority based Redundancy Control of Robot Manipulators", *Robotics Research: The Second International Symposium*, Cambridge, MA: MIT Press, pp.155-162, 1985.
- [Needler85] Needler, M.A. and Baker Don E., *Digital and Analog Controls*, Reston Pub. Co., 1985.
- [Negahdaripour90] Negahdaripour S and Yu C. H. and Shokrollahi A, "Recovering Shape and Motion From Undersea Images", *IEEE Journal of Oceanic Engineering*, vol 15, no 3, page 189-198 (1990)
- [Nejhad91a] Ghasemi Nejhad, M. N., R. D. Cope, and S. I. Guceri, "Thermal Analysis of In-Situ Thermoplastic-Matrix Composite Filament Winding," *ASME Journal of Heat Transfer*, vol. 113, no. 2, pp. 304-313, 1991.
- [Nejhad91b] Ghasemi Nejhad, M. N., R. D. Cope, and S. I. Guceri, "Thermal Analysis of In-Situ Thermoplastic-Matrix Composite Tape Laying," *Journal of Thermoplastic Composite Materials*, vol. 4, no. 1, pp. 29-45, 1991.
- [Nejhad92a] Ghasemi Nejhad, M. N., J. W., Jr., Gillespie, and R. D., Cope, "Prediction of Process-Induced Stresses for In-situ Thermoplastic Filament Winding of Cylinder," *Proceedings of Third International Conference CADCOMP, Computer Aided Design in Composite Material Technology*, pp. 225-253, 1992.
- [Nejhad92b] Ghasemi Nejhad, M. N., J. W., Jr., Gillespie, and R. D., Cope, "Processing Stresses for In-situ Thermoplastic Filament Winding Using the Divergence Method," *Proceedings of ASME Winter Annual Meeting 1992, Heat Transfer Effects in Materials Processing*, Guceri, S. I., and Alam M. K., Eds., HTD-vol. 233, pp. 33-43, 1992.

- [Nejhad93] Ghasemi Nejhad, M. N., "Issues Related to Processability during the Manufacture of Thermoplastic Composite Using On-line Consolidation Technique," *Journal of Thermoplastic Composite Materials*, vol. 6, pp. 130-145, 1993.
- [Nejhad94] Ghasemi Nejhad, M. N., J. W., Jr., Gillespie, and R. D., Cope, "Effects of Processing Parameter on Material Responses during In-situ Filament Winding of Thermoplastic Composites," *International Journal of Materials and Product Technology, Concurrent Engineering of Advanced Materials-Integration of Mechanics and Manufacturing*, vol. 9, no. 1/2/3, pp. 183-214, 1994.
- [Nejhad97] Ghasemi Nejhad, M. N., "Thermal Analysis for Thermoplastic Composite Tow/Tape Preheating and Pultrusion," *Journal of Thermoplastic Composite Materials*, vol. 10, no. 4, pp. 504-523, 1997.
- [Ng00a] Ng, R., A., Yousefpour, M., Uyema, and M. N., Ghasemi Nejhad, "Design, Analysis, Manufacture, and Test of Shallow Water Pressure Vessels using E-Glass/Epoxy Woven Composite Material for a Semi-Autonomous Underwater Vehicle, submitted to the *Journal of Composite Materials*, in review, 2000.
- [Ng00b] Ng, R., M., Uyema, A., Yousefpour, M. N., Ghasemi Nejhad, B., Flegal, and E., Sung, "Manufacturing and Testing of Shallow Water Composite Pressure Vessels for Semi-Autonomous Underwater Vehicle," *World Automation Congress 2000 (WAC 2000)*, in press, June 2000.
- [Nie98] Nie, J., J. Yuh, E. Kardash, and T.I. Fossen, "On-Board Sensor-Based Adaptive Control of Small UUVs in Very Shallow Water", *IFAC Symposium on Control Applications for Marine Systems*, 1998.
- [Nie99] Nie, J., J. Yuh, E. Kardash, and T.I. Fossen, "On-Board Sensor-Based Adaptive Control of Small UUVs in Very Shallow Water", *International Journal of Adaptive Control and Signal Processing*, vol. 13, 1999.
- [Nygards98] Nygards J and Wernersson A, "On Covariances for fusing Laser Ranger and Vison with Sensors Onboard a Moving Robot", *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, page 1053-1059 (1998)
- [Ogata87] Ogata, K., *Discrete-time Control Systems*, Prentice-Hall, New Jersey, 1987.
- [Omura95] Omura, G., *Mastering AutoCAD 13 for Windows95, Windows3.1, and WindowsNT*, Sybex, 1995.
- [Parrish73] Parrish, A., *Mechanical Engineer's Reference Book*, Butterworths, 1973.
- [Pascol93] Pascoal, A., M. J. Rendas, V. Barroso, C. Silvestre, P. Oliveria and I. Lourtie, "Simulation Study of an Integrated Guidance System for an Autonomous Underwater Vehicle", *Acoustic Signal Processing for Ocean Exploration* (Eds. J.M.F. Moura and I.M.G. Lourtie), pp. 587-592, 1993.
- [Pickering97] Pickering, E. R., "Welding Aluminum," *Journal of Advance Materials & Processing*, pp. 29-30, 1997.
- [Porter67] Porter, H.W., *Machine Shop Operations and Setups*, American Technical Society, 1967.
- [Pugh70] Pugh, H., *Mechanical Behavior of Materials Under Pressure*, Elsevier Publishing Co., 1970.
- [Raibert81] Raibert, M.H. and J.J. Craig, "Hybrid Position/Force Control of Manipulators," *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 12, pp. 126-133, 1981.
- [Reynolds89] *Product and Data Catalog - Reynolds Aluminum Supply Company*, Reynolds Aluminum Supply Company, 1989.

- [ROV91] Intervention/ROV'91 Conference & Exposition, Hollywood, Florida, Sponsored by the ROV Committee and the South Florida Section of the Marine Technology Society, 1991.
- [Sagatun92] Sagatun, S.I. Modeling and Control of Underwater Vehicles: Lagrangian Approach, Dr. Ing Thesis, Norwegian Institute of Technology, 1992.
- [Sayers99] Sayers, C., Remote Control Robotics, Springer, 1999.
- [Scheck84] Scheck, L.A.. and Edmondson, G.C., Practical Welding, Glencoe Publishing Co., 1984.
- [Schlichting79] Schlichting, H., Boundry-Layer Theory, McGraw-Hill, Inc., 1979.
- [Schwartz84] Schwartz, M.M., Composite Materials Handbook, McGraw-Hill, 1984.
- [Schweitzer83] Schweitzer, P.A., Corrosion and Corrosion Protection Handbook, M. Dekker, 1983.
- [Serway89] Serway, R.A. and Faughn, J.S., College Physics, Saunders College Publishing, 1989.
- [Shahinpour87] Shahinpour, M., A Robot Engineering Textbook, New York, Harper & Row Publishers, 1987.
- [Shames89] Shames, I.H, Introduction to Solid Mechanics, Prentice-Hall, Inc., 1989.
- [Shen81] Shen, C., and G. S. Springer, "Environmental Effects in the Elastic Moduli of Composite Materials," Environmental Effects on Composite Material, Ed. G. S. Springer, Westport: Technomic Publishing Company, Inc., 1981.
- [Shigley89] Shigley, J.E. and Mischke, C.R., Mechanical Engineering Design, McGraw-Hill, Inc., 1989.
- [Smith90] Smith, C.S., Design of Marine Structures In Composite Materials, Elsevier Applied Science, 1990.
- [Smith96] Smith, J. and K. Sugihara, "GA toolkit on the Web", Proc. of the First Online Workshop on Soft Computing (WSC1), pp.93-98, 1996.
- [Sonmez97] Sonmez, F.O. and H.T. Hahn, "Analysis of the On-line Consolidation Process in the Thermoplastic Composite Tape Placement", Journal of Thermoplastic Composite Materials, v. 10, pp. 543-572, 1997.
- [Sprong89] Sprong, M.W. and M. Vidyasagar, Robot Dynamics and Control, New York, John Wiley & Sons, 1989.
- [SubTech85] Submersible Technology: Proceedings of an International Conference (Subtech '85), Aberdeen, UK, pp. 29-31, 1985.
- [Sugihara97] Sugihara, K. and Yuh, J., "GA-based motion planning for underwater robotic vehicle," Proc. 10th Int'l Symp. on Unmanned Untethered Submersible Technology (UUST-10), Durham, NH, 1997, pp.406-415.
- [Sugihara98a] Sugihara, K., "GA-based on-line path planning for SAUVIM," Proc. 11th Int'l Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-98-AIE), Castellon, Spain, 1998, pp.329-338.
- [Sugihara98b] Sugihara, K. and J. Smith, Genetic Algorithms for Adaptive Planning of Path and Trajectory of a Mobile Robot in 2D Terrain, IEICE Trans. on Information and Systems, to appear 1998 .
- [Sugihara98c] Sugihara, K. and J. Yuh, "GA-based Motion Planning for Underwater Robotic Vehicles", Proc. 10th Int'l Symp. On Unmanned Untethered Submersible Technology (UUST-10), pp.406-415, 1998.
- [Sugihara99] Sugihara, K. and Smith, J., "Genetic algorithms for adaptive planning of path and trajectory of a mobile robot in 2D terrain," IEICE Trans. Information and Systems, Vol. E82-D, No. 1, pp.309-317, January 1999.

- [Svenssoon99] Svensson S. and Lexander J. and Ericson B, "OBSERVATION AND INSPECTION IN SWEDISH WATERS", Underwater Imaging, Douglas J. Holloway, Editor, SPIE vol. 980, page 75-81 (1988).
- [Swartz91] Swartz: and Cummings J, "Laser range-gated underwater imaging including polarization discrimination", Underwater Imaging, Photography, and Visibility, Richard W. Spinard, SPIE vol. 1537, page 42-56 (1991).
- [Tai99] Tsai, L.W., Robot Analysis: The Mechanics of Parallel and Serial Manipulators, John Wiley and Sons, 1999.
- [Takashi91] Takashi, K., Electric Motors and their Controls: An Introduction, Oxford University Press, 1991.
- [Tarn96] Tarn, T.J., G.A., Shoultsand, and S.P. Yang, "A Dynamic Model of an Underwater Vehicle with a Robotic Manipulator using Kane's Method", Autonomous Robots 3, pp. 269-283, Kluwer Academic Publisher, Netherlands, 1996.
- [TI92] Linear Circuits Operational Amplifiers Data Book, Texas Instruments, 1992.
- [Tsai99] Tsai, L.W., Robot Analysis - The Mechanics of Parallel and Serial Manipulators, John Wiley and Sons, Inc. 1999.
- [Tupper96] Tupper, E.C., Introduction to Naval Architecture, Oxford, Butterworth-Heinemann Publishing, 1996.
- [Ullman92] Ullman, D.G., The Mechanical Design Process, McGraw-Hill Inc., 1992.
- [Unimate81] Unimate PUMA Robot: Volume 1 – Technical Manual 398H1, Unimation Inc., Condec Company, Danbury CT, 1981.
- [Unimate84] Unimate PUMA Mark II Robot: 500 Series, Volume 1 – Equipment Manual, Unimation, Westinghouse Corporation, Danbury CT, 1984.
- [Unimate86] Unimate Industrial Robot: Programming Manual, User's Guide to VAL II Version 2.0 (398AG1), Unimation, Westinghouse Corporation, Danbury CT, 1986.
- [Unimate97] Unimate – Supplement to the User's Guide to VAL II: VAL II-IVM PC Supervisor Interface (397W1), Unimation, Westinghouse Corporation, Danbury CT, 1987.
- [Valentine98] Valentine, R., Motor Control Electronics Handbook, McGraw-Hill, 1998.
- [Vieville97] Vieville, T., A Few Steps Towards Active 3D Vision, Springer-Verlag, 1997.
- [Vinson87] Vinson, J.R., and R.L. Sierakowski, The Behavior of Structures Composed of Composite Materials, Netherlands, Kluwer Academic Publishers, 1987.
- [Weast81] Weast, R.C. and Astle, M.J., CRC Handbook of Chemistry and Physics - 61st Edition, CRC Press, Inc. 1981.
- [Wedermann89] Wedermann, C., K. Friedrich, M. Cirino, and R. B. Pipes, "Design and Fabrication an On-Line Consolidation Facility for Thermoplastic Composites," Journal of Thermoplastic Composite Materials, vol. 2, pp. 293-306, 1989.
- [Wells91] Wells W, "Indirect illumination to reduce veiling luminance in seawater", Underwater Imaging, Photography, and Visibility, Richard W. Spinard, SPIE vol. 1537, page 2 (1991).
- [Werdermann89] Werdermann, C., K. Friedrich, M. Cirino, and R.B. Pipes, "Design and Fabrication an On-line Consolidation Facility for Thermoplastic Composites", Journal of Thermoplastic Composite Materials, vol. 2, pp. 293-306, 1989.
- [Whitney87] Whitney, D.E., "Historical Perspective and State of the Art in Robot Force Control," International Journal of Robotic Research, vol. 6, no.1, pp. 3-14, 1987.
- [Wilson92] Wilson, P.A., International Conference on Manoeuvring and Control of Marine Craft, Proceedings of the Second International Conference, 1992.

- [Wit98] Wit, C.C.D., E.O. Diaz, and M. Perrier, "Robust Nonlinear Control of an Underwater Vehicle/ Manipulator System with Composite Dynamics", Proc. IEEE Conf. on Robotics and Automation, pp.452-457, 1998.
- [Yang98a] Yang, K.C., J. Yuh, and S.K. Choi, "Experimental Study of Fault-tolerant System Design for Underwater Robots", Proc. IEEE Conf. on Robotics and Automation, pp. 1051-1056, 1998a.
- [Yang98b] Yang, K.C., J. Yuh, and S.K. Choi, "Experimental Study of Fault-tolerant System Design for Underwater Robots", Journal of System Sciences, 1998b.
- [Yousefpour00a] Yousefpour, A., and M. N. Ghasemi Nejjhad, "Effects of Geometric Optimization of Tapered End-caps on Thick Thermoplastic Composite Pressure Vessels for Deep Ocean Applications," Journal of Thermoplastic Composite Materials, submitted for publication, 2000.
- [Yousefpour00b] Yousefpour, A., and M. N. Ghasemi Nejjhad, "Experimental and Computational Study of APC-2/AS4 Thermoplastic Composite C-Rings," Journal of Thermoplastic Composite Materials, in press, 2000b.
- [Yousefpour00c] Yousefpour, A., R., Ng, M., Uyema, M. N., and Ghasemi Nejjhad, "Design and Finite Element Analysis of Shallow Water Composite Pressure Vessels for Semi-Autonomous Underwater Vehicle," World Automation Congress 2000 (WAC 2000), in press, June 2000c.
- [Yousefpour99] Yousefpour, A., and M. N. Ghasemi Nejjhad, "Testing and Finite Element Modeling of APC-2/AS4 Thermoplastic Composite C-rings," 31st International SAMPE Technical Conference: Advanced Materials & Processes Preparing for the New Millennium, vol. 31, pp. 643-654, Chicago, IL, 1999.
- [Yuh92] Yuh, J., V. Adivi & S.K. Choi, "Development of a 3D Graphic Test Platform for Underwater Robotic Vehicles," Proceedings of the 2nd International Offshore and Polar Engineering Conference, Jun. 1992.
- [Yuh94a] Yuh, J., "Learning Control for Underwater Robotic Vehicles", IEEE Control System Magazine, vol.15, No.2, pp.39-46, 1994.
- [Yuh94b] Yuh, J., Underwater Robotic Vehicles: Design and Control, Workshop on Future Research Directions in Underwater Robotics, TSI Press, p. 361, 1994.
- [Yuh95] Yuh, J., Underwater Robotic Vehicles: Design and Control, TSI Press, 1995.
- [Yuh96] Yuh, J., "An Adaptive and Learning Control System for Underwater Robots", 13th World Congress International Federation of Automatic Control, A, pp. 145-150, 1996.
- [Yuh98a] Yuh, J., S.K. Choi, C. Ikehara, G.H. Kim, G. McMurtry, M. Ghasemi Nejjhad, N. Sarkar, and K. Sugihara, "Design of a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM)," Proceeding of the Underwater Technology '98, 1998.
- [Yuh98b] Yuh, J., J. Nie, and W.C. Lee, "Adaptive Control of Robot Manipulators Using Bound Estimation", IEEE International Conference on Intelligent Robots and Systems, 1998.
- [Yuh99] Yuh, J. and J. Nie, "Experimental Study on Adaptive Control of Underwater Robots," Proceedings of the IEEE International Conference of Robotics and Automation, Detroit, MI, May 1999.
- [Zege91] Zege E and Ivanov A and Katsev I, "Image Transfer Through a Scattering Medium" Springer-Verlag, Berlin, page 277-305 (1991)
- [Zuech88] Zuech, N., Applying Machine Vision, John Wiley and Sons, Inc., 1988.

Phase I: Final Report Appendix

Appendix 1: SAUVIM Velocity Analysis

Appendix 2: SAUVIM Test Plan (Phase I – Shallow Water)

Appendix 1: SAUVIM Velocity Analysis

Motivation:

To get an initial, estimation analysis of the vehicle's performance, these series of simplified calculations were performed. These are **not** intended to be a full-featured dynamic analysis; they are merely reasonable and precise estimates of the following: 1) the acceleration responsiveness of the SAUVIM to the planned thruster units - a set of eight Technodyne Model 1020 brushless motor thrusters; 2) the ultimate cruising speed of the vehicle under neutrally buoyant thrust conditions as well as weighted descent; and 3) an estimate of the rotational (yaw-) responsiveness of the vehicle.

Introduction:

In all cases, we use SAUVIM's response in terms of lineal and angular distance covered versus time elapsed since application of thrust at 100% of the rating supplied on the manufacturer's data sheet for the given thruster set. The other information is detailed in the velocity (or angular speed) versus the time elapsed since the application of the full rated thrust. The initial state of the vehicle in all cases is a full stop position. The SAUVIM vehicle faired is of the following shape:

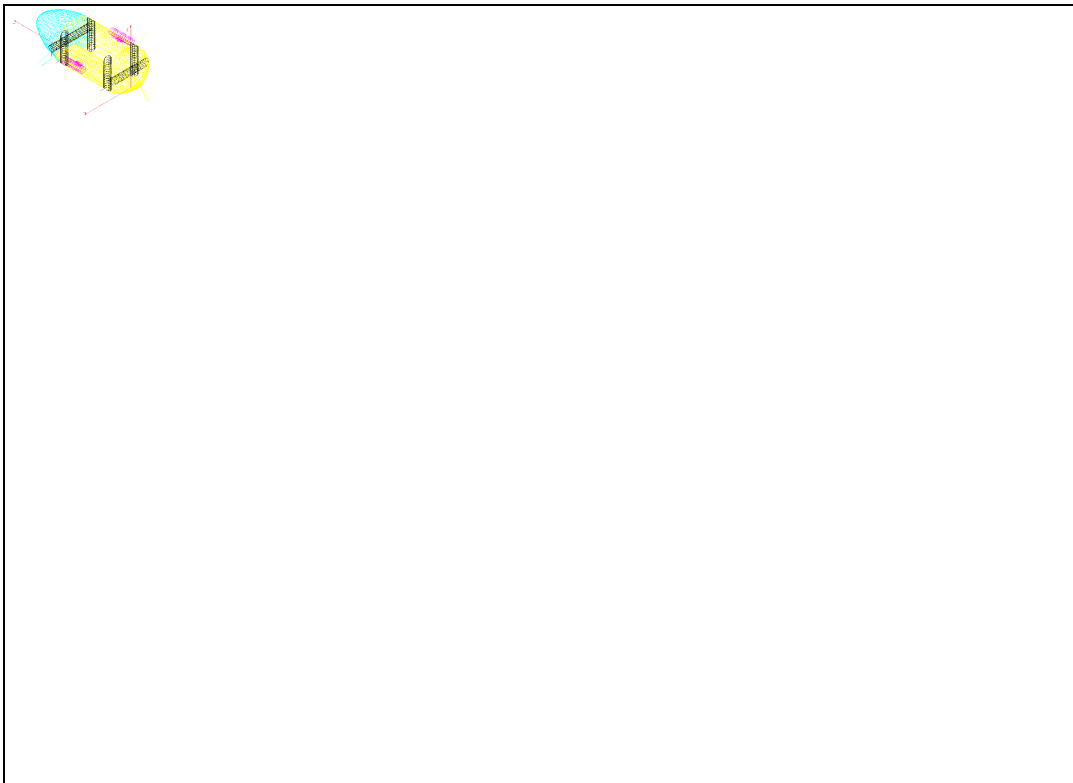


Figure 1: Isometric View of the SAUVIM Fairing with Thruster Tubes

For purposes of this analysis, the longitudinal direction is along the x-axis, the lateral direction is along the y-, and vertical direction in the z-. Four thrusters (the black tubes parallel to the z-axis) point vertical with the more powerful thrust vector pointed down, two point fore and aft (the pink tubes on pylons), the aft direction is the more powerful thrust vector. The two lateral thrusters (black

tubes parallel to the y-axis) face with the more powerful vector pointing along +y, this choice is arbitrary and is made to ensure a balanced pair of thrust in either lateral direction.

Method:

For velocity, a force balance equation was employed: $\text{Force}_{\text{thrusters}} = \text{Drag}_{\text{vehicle}} + (\text{Mass}_{\text{vehicle}})$

$$T_{\text{thrusters-set}} = \frac{1}{2} \rho C_R A \dot{q}^2 + I \ddot{q}$$

where the terms, are F is the maximum thruster output force in Newton (kg_f which is equivalent to 10N), each Technodyne model 1020 outputs 21.4 kg_f (214N or 47 lb_f) in the forward direction and 14.5 kg_f each (145N or 32 lb_f) in the reverse direction. These values were obtained right off of the Technodyne data sheets. Value should be within ±10% of actual value.

ρ, is density of seawater in kg/m³ (=1024 kg/m³). This was obtained from an introductory Oceanography text. Value should be within ±2-3% of actual value.

C_D, is Drag coefficient for SAUVIM in dimensionless form. For the drag in the forward longitudinal direction, 0.35 was used for the faired vehicle and 0.85 was used for the unfaired vehicle. The former number is a composite of the CFD results from CHAM (0.40), HSI (0.25), and the figure cited for a Ford Taurus (0.32). Value may be within ±30% of actual value, these estimates are very preliminary until a combination of thorough CFD study and/or model testing is carried out.

A, is Cross-sectional area from frontal/rearward vantage point in m² (= 3.74 m² frontal, = 10.19 m² lateral, =13.15 m² vertical) These values are derived directly from the ACADr13 model of the SAUVIM fairing. Value should be within ±2% of actual value.

m, is mass of the SAUVIM vehicle includes dry mass as well as entrained water mass within the fairing and the vehicle components, in kg (= 17,800 kg/39,000 lbs for faired SAUVIM, 8,160 kg/18,000 lbs for unfaired SAUVIM). The faired vehicle mass estimate is taken directly from the

v, is velocity of vehicle in m/sec (reported as knots though, initially set to 0 m/sec).

ẍ, is acceleration at a given time in m/sec² (initially set to 0 m/sec²).

Solution for of the differential equation (x-dot (v) and x-dot-dot) proceeds by integration along the time steps using Euler’s method: the initial acceleration and velocity were ‘0’, subsequent steps reference the previous time steps which are spaced at one second intervals. The rest of the terms were treated as constants.

For yaw response the form of the equation is $\text{Couple}_{\text{thruster}} = \text{Drag}_{\text{vehicle-rot.}} + (I_{\text{vehicle}})?$

$$F_{\text{thrusters}} = \frac{1}{2} \rho C_D A v^2 + m \ddot{x}$$

The variables in here are the rotational equivalents of the variables in equation (1), detailed notes on their values will be discussed on the case analysis.

Cases: The following cases have been explored:

- Case I - Forward/Rearward (longitudinal translation) with fairing
- Case II - Forward/Rearward (longitudinal translation) without fairing
- Case III - Lateral Starboard/Port (lateral translation) with fairing.
- Case IV - Lateral Starboard/Port (lateral translation) without fairing.
- Case V - Vertical up/Vertical down (vertical translation) with fairing
- Case VI - Vertical up/Vertical down (lateral translation) without fairing
- Case VII - Yaw response with fairing
- Case VIII - Yaw response without fairing

Assumptions and results pertinent to each will be detailed case-by-case bases. Many items that are in a complete rigorous analysis have been discounted among these are: duct water-mass inertia, vehicle damping coefficient, duct drag losses, C_D variations with velocity change, off-centric application of forces from the center of inertial and drag resistances and resultant thrust reductions off of the maximum to accommodate balancing, reduction in thrust from the Model 1020 as SAUVIM vehicle gains speed and propulsive effective thrust drops off (propeller advance ratio effects).

The following table 1 summarizes the variables used for each case.

Table 1: Different Case Studies for Thruster Tests

Case	Thruster Force	C_D (or C_R)	A	m (or I_{zz})
Units	N		m^2	kg (kg m^2)
I - Forward with fairing	419.8	0.35	3.74	17800
I - Astern with fairing	284.5	0.35	3.74	17800
II - Forward without fairing	419.8	0.85	3.74	8160
II - Astern without fairing	284.5	0.85	3.74	8160
III - Lateral Starboard with fairing	419.8	0.75	10.19	17800
III - Lateral Port with fairing	284.5	0.75	10.19	17800
IV - Lateral Starboard without fairing	419.8	0.80	10.19	8160
IV - Lateral Port without fairing	284.5	0.80	10.19	8160
V - Vertical up with fairing	839.6	1.2	13.15	17800
V - Vertical down with fairing	569.0	1.2	13.15	17800
V - Vertical up without fairing	839.6	1.4	10.00	8160
V - Vertical down without fairing	569.0	1.4	10.00	8160
VII - Yaw with fairing	?	?	10.19	17800
VIII - Yaw without fairing	?	?	10.19	8160
IX - Unpowered 30° descent cruise	N/A	N/A	N/A	N/A

Assumptions/Results: Case-by-case breakdown will proceed.

Case I & II - This is the baseline SAUVIM case where the neutrally buoyant vehicle is accelerated straight forward. It is assumed here, as for all the subsequent cases in this analysis, that the line of action of the thruster vectors is lined up sufficiently close to the center of inertial mass as well as the singular center of drag force action to preclude having to reduce thrust in any of the set to counter the resulting rotational drifts that would occur (e.g. all forces are centric in nature). The two longitudinal thrusters are rated at 47 lb_f/each (214 N) in the forward direction and 32 lb_f/each (145 N) in the reverse direction.

The area, 3.74m², was obtained from the ACADR13 solid model. This is the profile cross-sectional area seen from along the vehicle's X-axis as is standard practice in drag calculations using dimensionless drag data.

For estimating the vehicle mass two methods were used; for the unfaired vehicle the mass was estimated from the itemized tally spreadsheet (Sensit4.wb3) of all the major components with some adjustment made for water that would be entrained within the major cavities of the vehicle (the wiring space above the batteries, around the pressure vessels within the main component box - for details see Figure MED-10). The table below summarizes the approximate void space within each of the major cavity spaces of the vehicle. The foam space cavity is not included as it is assumed to completely occupy SAUVIM's dry mass. The venting value is an estimated guess at the amount of water in a given cavity. It also estimates the water spillage throughout the vehicle components, and therefore, will not contribute to the inertial mass of the vehicle. The approximately 1640 kg figure of entrained water is added to the unfaired SAUVIM mass for the startup run calculations in table 2.

Table 2: Unfaired SAUVIM Entrained Water Mass Estimates

Volume Name	Cavity Height (in)	Cavity Length (in)	Cavity Breadth (in)	Volume (ft ³)	% Volume Occupied	Adjusted Volume	Water in Voids (lbs)	Venting Estimate	Adjusted Mass (lbs)
Battery	21.5	75.0	45.0	42.0	34%	27.52	1717.25	0.50	858.62
Ballast	21.5	32.0	45.0	17.9	11%	15.87	989.99	0.50	494.99
Arm	21.5	32.0	45.0	17.9	8%	16.50	1029.82	0.70	720.87
PV	23.0	165.0	45.0	98.8	17%	82.07	5121.35	0.30	1536.41
Approximate Mass of Water entrained inside Vehicle (lbs)									3610.89
Approximate Mass of Water entrained inside Vehicle (kg)									1639.34

The resulting unfaired SAUVIM effective inertial mass is around 6,900 kg. To give an idea of the relative volume ratio between free space that floods and solid SAUVIM components the approximate volumes of some of the major components is given in table 3.

Table 3: Approximate Volumes of Components

Approximate Volumes of:	ft ³	m ³
12 Batteries	14.47	0.41
6 Pressure Vessels	16.76	0.47
Arms and Tray	1.41	0.042

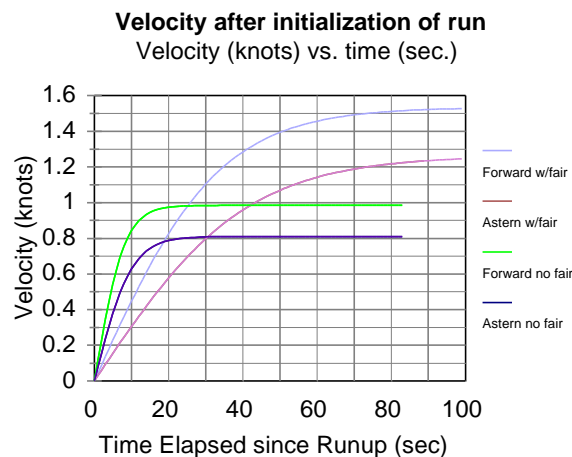
Ballast Tray and Ballast	2.05	0.058
--------------------------	------	-------

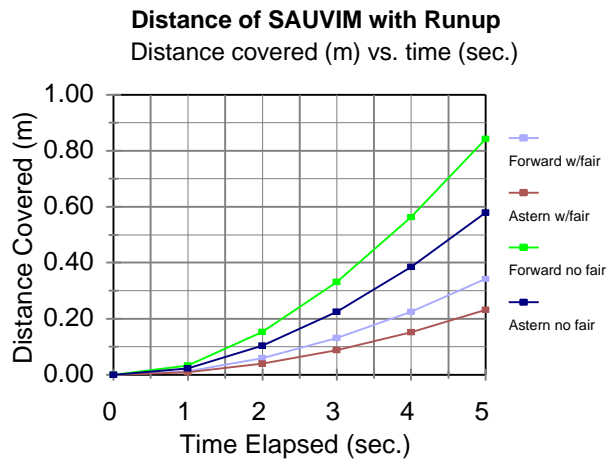
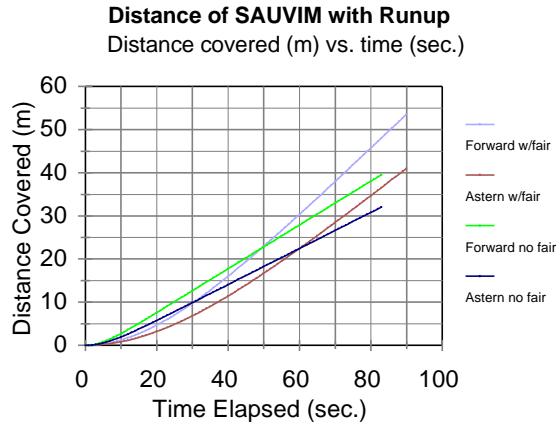
For the faired vehicle the wet mass figure of 17,800 kg was arrived at simply by assuming that the SAUVIM is neutrally buoyant and determining the enclosed volume of water within the fairing shell (done in the ACADR13 model). This assumes a completely stagnant pocket of water within the fairing, which clearly not the case with the actual vehicle as the will be ports, ducts and open areas in the fairing for water to spill through. Since the worst case is to assume a completely sealed fairing, this assumption was made.

Meanwhile, a fairly conservative coefficient of drag was adopted (in the dimensionless form) as well. For forward motion a C_D of 0.35 was used. This was arrived at based on preliminary results from the Phoenix CFD code ($C_D = 0.40$), work done at Pacific Marine with CFD code ($C_D = 0.25$), and some book sources. Chosen from these book sources was the C_D of some concept cars having a very similar form to the SAUVIM fairing ($C_D = 0.35$, 0.23, pg.12-3, $C_D = 0.25$, pg. 12-9 Fluid-Dynamic Drag, Practical Information on Aerodynamic Drag and Hydrodynamic Resistance, Hoerner, S.F., AIAA press - 1965). These values were cited for shapes that are operating in ground effect and therefore only approximate the SAUVIM fairing in a free stream environment, hence the selection of a more conservative value for C_D . The same C_D was used for both forward and rearward motion.

The results of the analysis are summarized in these graphs, the first of which shows the SAUVIM velocity as a function of time elapsed since thruster startup, SAUVIM displacement since thruster startup and the same in a shortened time span. It can be seen with the fairing on that the ultimate forward velocity possible with the twin Technodyne 1020's will be 1.5 knots (0.79 m/sec) forward and a little over 1.2 knots at full reverse (0.65 m/sec). Full speed will be reached after 90 seconds of run up. Without the fairing, acceleration will be much better and the full speed will be reached within 20 seconds, however, high speed will drop to 1.0 knots (0.50 m/sec) and 0.8 knots (0.41 m/sec) for forward and reverse directions, respectively.

It can be seen from the third graph that pulsing the thrusters in the forward direction for the faired vehicle for 3 seconds will result in 15cm of translation, the unfaired SAUVIM will have moved 35cm.



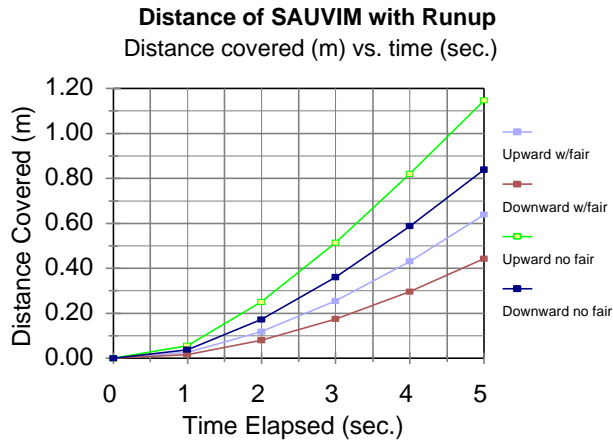
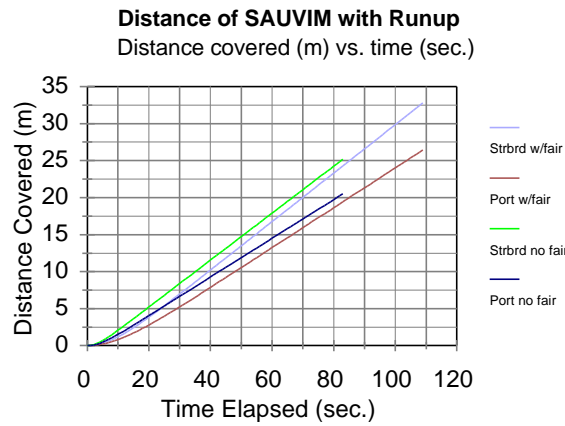
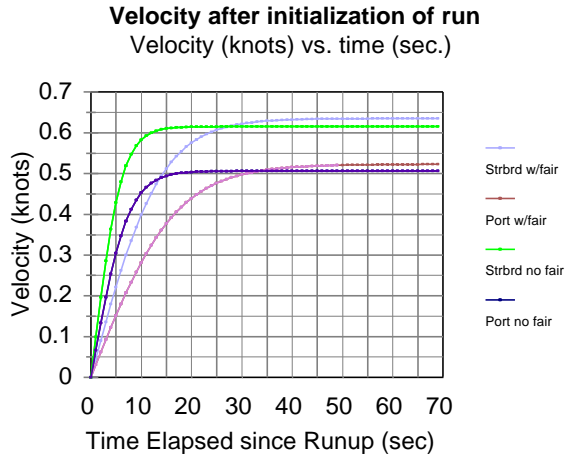


Case III & IV - The mass assumptions are the same as in cases I & II, here lateral motion is concerned. The major changes here concern, profile area, the direction of the Technodyne 1020's favored thrust direction, and C_D changes. For the C_D of both the faired and unfaired vehicle there are no variances with direction to either side as the SAUVIM is symmetric across the XZ-plane. The C_D for the faired vehicle a value of 0.75 was estimated (the data of a circular cylinder of similar aspect ratio in cross-flow with $C_D = 0.70$ was used a basis for this value, from Hoerner, S.F., AIAA press - 1965, pg. 3-16). This value was degraded to 0.80 for the unfaired vehicle to account for sharper edges on the ends of the unfaired vehicle, though the bulk cross-section remains largely unchanged. Though not accounted for in this analysis, due to the relatively complete coverage of floatation foam over the vehicles flooded spaces from this direction the entrained water mass value for the unfaired SAUVIM should probably be adjusted upward.

The more powerful thruster direction of the 1020's was chosen to be applicable for starboard motion, this was chosen arbitrarily; avoidance of any yawing during paired lateral thrusting will probably necessitate orienting the thrusters in this fashion until the symmetrical propellers are retrofitted. The profile area as seen from along the y-axis is 10.19 m^2 . This was obtained from the ACAD R13 fairing model. The results are shown here, it can be seen that the maximum lateral speeds are around 0.63 knots (0.32 m/s) and 0.52 knots (0.26 m/s) for starboard and port directions with the fairing. Without the fairing these values become 0.61 knots (0.31 m/s) and 0.50 knots (0.26 m/s). Note from the first

graph that the maximum speeds are reached within 10 and 30 seconds for the unfaired and faired conditions, respectively.

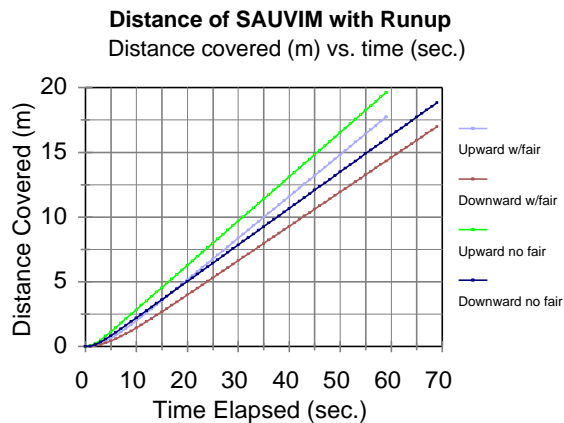
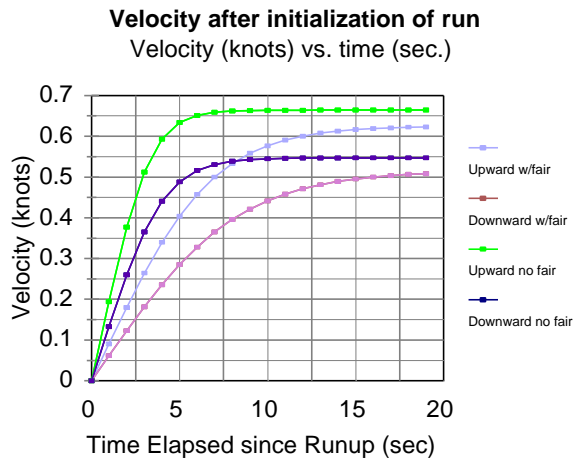
A three second pulse of full thrust will move the vehicle from about 9-28 cm depending on the fairing and favored direction of thrust.

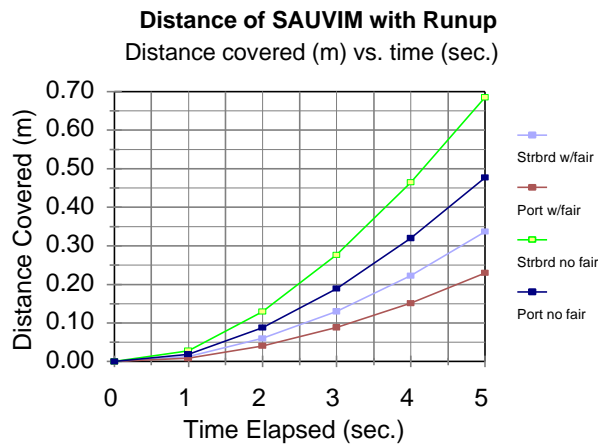


Case V & VI - These cases account for vertical motion. The greater power of the thrusters is due to a set of four Technodyne 1020's being selected for this direction. The favored direction of thrust in this analysis was chosen for the downward direction. This is a design issues but was chosen to fight the gravitational field should the SAUVIM be slightly heavy which due to foam compressibility is a more likely state to be in upon cruise to the bottom. So around 840 N can be applied to move vertically up and about 570 N can be applied in the downward direction. The C_D for this direction was chosen to be 1.2; this was cited for cylinders at moderate Reynolds number flows. At Reynolds flow values typical for our vehicle the C_D for cylinders actually drops to around 0.7, but this is due the migration of regions of separation back on the smooth surface of a cylinder. Since the roughly circular SAUVIM fairing form has edges that trip off flow separation in fixed locations, unlike a smooth cylinder in moderate Reynolds number flow, the higher value for C_D is chosen.

The cross-sectional area is now 13.15 m^2 , this is the profile area of the fairing as seen from the top. It is somewhat less for the unfaired vehicle as the nosecone and tail cone do add about 30% more area to the silhouetted area as opposed to the unfaired vehicle.

The results here indicate the faired vehicle can expect vertical maximum speeds on the order of 0.65 knots and 0.54 knots downward when faired and 0.62 knots and 0.51 knots when unfaired. It can be seen that a three second pulse of the thrusters at full rated load will move the vehicle from about 18-55 cm depending on the thrust direction and the presence of the fairing.





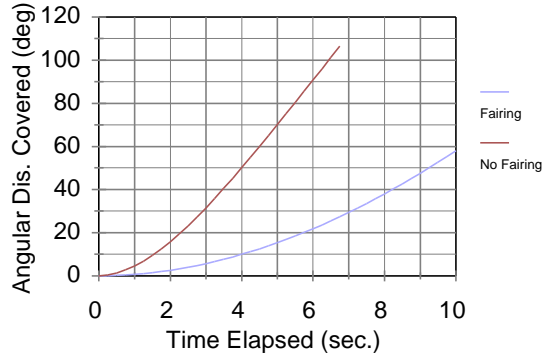
Case VII & VIII - These cases concern yaw rotation. In this case the SAUVIM vehicle pivots about an axis parallel to the z-axis with application of thrust in opposite directions of both the longitudinal and lateral pairs of the thrusters. This has been calculated for both the faired and unfaired vehicle variants. The following assumptions run throughout the models:

- The principle rotational moment for the faired vehicle will assume the fairing volume is a uniform mass with the density of seawater. Treating the volume within the fairing model as a uniform mass and using a solid function to recover the inertial moment derived the moment value.
- C_{D-rot} for the SAUVIM will be that of a rectangular parallelepiped of similar aspect ratio. The value was degraded somewhat for the unfaired vehicle owing to separation around the edges on the aft and forward ends.
- Two sets of thrusters will contribute to the couple moment, the lateral and longitudinal pairs, furthermore no wake coupling effects will be accounted for.
- The principle moment for the unfaired vehicle was found by applying the mass moment formula to the major components that are tracked on the datasheet.
- The couples coming off of the thrusters we using the minimum thrust rating at the shortest moment arm from the inertial axis. For the lateral pair of thrusters this was 14.5 kg of thrust at 60 inches from the inertial axis, for the longitudinal pair it was again 14.5 kg at 60 inches from the inertial axis.
- The inertial axis (I_{zz}) was calculated to be at vehicle coordinates for the X=85in and Y=0 in for the unfaired vehicle, the location of I_{zz} on the faired vehicle was at X=85in and Y=0in again.
- The magnitude of I_{zz} is 41,850 kg-m² for the faired vehicle and reduces to 6,615 kg-m² when the fairing is removed.

The graphs below summarize the results. As expected the faired vehicle has a slower initial response than the unfaired variant; however, the ultimate high rotational speed is not critical as the likely

maneuvers will be completed before obtaining the maximum speed. All of the angular distances are given in degrees.

Angular Distance of SAUVIM with Runup
Angular Dis. (deg) vs. time (sec.)



Velocity after initialization of run
Ang. Velocity (deg/s) vs. time (sec.)



Unpowered Cruise - This case is still under investigation.

General Observations:

A free-flooded fairing incurs both advantages and disadvantages to a vehicle equipped with it, though it may be very useful for open ocean operations, it does result in a real hit on inertial response of the vehicle upon thruster startup and short term pulsing for active station-keeping. The vehicle will not be nearly as responsive to thruster inputs with it installed although cruising range and inertial damping to disturbances will be increased. For a vehicle cruising at near constant speeds or involved in station keeping for extended periods in a steady current using passive inertial damping, the fairing may yield a distinct advantage even though it effectively doubles the vehicle inertial mass from 8,200 kg to 17,800 kg.

Cruising range under full power is affected by the fairing. Consider the longitudinal thrusters only. These thrusters are powered by three lead-acid DeepSea SB-48/18 batteries, arranged in a serial bank to provide 144VDC at 18 Amp-Hours of continuous draw (2.60 kW-hr). This gives the vehicle with

two Technodyne thrusters drawing 1550 Watts continuously the following ranges: 2.70 nautical miles (5.0 km) with the fairing and 1.73 nautical miles (3.2 km) without the fairing.

Recommendations:

The Theoretical Modeling (TM) dynamics group may want to explore the following parametric changes to the SAUVIM vehicle:

Fairing size changes - Fairing size changes in the lineal distances will have a square-law change influence on the profile areas and therefore the magnitude of the drag forces. A 10% reduction in fairing size may reduce hydrodynamic drag to about 81% of the baseline case; meanwhile the inertial mass will drop to about 75% of the baseline case. Maximum cruise speed will climb about 10-12% and initial responsiveness will climb modestly, however, loss of the vehicle expansion/design flexibility that the prototype has will be suffered. The most feasible fairing size change is to redistribute some of the foam up on the top of the vehicle to down within the battery tray area and into some of the larger pockets formed between the pressure vessels. A side effect of doing this relocation would be a shorter separating distance between the centroid of the volume of all the SAUVIM components and the center of mass; this would result in a more tightly responsive vehicle to ballast trim, thruster and fin trim inputs, conversely also to arm inputs and being buffeted by currents, external influences.

Thruster Power Changes: Migrating from the Technodyne 1020 to the 2010 model would nearly quadruple the thrust from each unit (Technodyne 2010 data sheets rating 143 lb_f (650 N) forward and 80 lb_f (364 N) reverse). The Technodyne manufacturing representative has stated that these values are only about 75% of the thrust that the 2010 can actually sustain under continuous load. Raising thruster power by a factor of four will double the maximum speed as drag is a square law dependency on velocity. Note though that the cruising range under maximum cruise speed possible with 2010 units is only about 50% of that with the smaller thruster units running at their maximum rated thrust. Of course economic concerns enter here as the Model 2010 units cost around \$9,500 apiece as opposed to the \$5,800 that the 1020 units run.

Decent Cruise: This is not critical for shallow water variant of the SAUVIM but will become a critical portion of the mission phases as the SAUVIM proceeds into deep-water missions. The ability to glide in a controlled fashion and make course corrections to ensure arrival close to the task site with minimal, if any, thruster application will be critical from the standpoint of the small cruising range imposed by the battery bank energy limits and the minimization of time during which fixed electrical loads (e.g. computers, long-baseline sensors, etc) draw power. Hence further exploration of this mode of vehicle motion warrants conceptual consideration, if not detailed analysis, even prior to commencement of shallow water operations.

Recommended Tasks: These steps will be needed for a more accurate dynamic model the TM-dynamics group should consider the following tasks: 1) locate the center of drag action for the three principle directions, 2) locate the inertial center for both the faired and unfaired vehicles using the AutoCAD fairing model and the Quattro spreadsheet tally of the major SAUVIM component masses, 3) from the former two steps and knowing where the thrusters are located determine the thrust tuning adjustments needed to cancel non-centric effects and 4) determine and map the combined drag/inertial resistance centroid location with velocity location.

Also it will be worth determining the effects of the three planned fins on the vehicle dynamics for both powered and decent cruise. This will be of great value in sizing of the foils for the fin units to ensure the right balance between vehicle-response and vehicle-handling concerns.

Appendix 2: SAUVIM Test Plan (Phase I – Shallow Water)

Objective

- To test essential hardware and software functions and to check the integrity of the system.

In this write-up, it is assumed that the vehicle is completed for experiments. The joystick-based controller will be used in most cases. After the completion of these basic tests, a simple “dead-reckoning” control algorithm and a simple object-following control algorithm will be tested for initial closed-loop control and navigation purposes. The basic tests plans are:

Test Plan 1

Goal - Test the basic emergency handling functions.

The weight dropping functions will be tested. This test is made of two parts. In the first part, weight will be dropped as the vehicle reaches desired depth by monitoring depth sensor. (The desired depth is not determined yet, but it should be limited within 60 ft so that divers can reach to the vehicle for recovery.) The second part will simulate leakage in the pressure vessels. Timer switch can be connected to one of the leakage sensors to simulate the leakage. During the test, battery level will be monitored and logged.

Sensors: depth sensor, leakage sensor, battery gauges

Actuator: weight drop

Test Plan 2

Goal - Test if all the sensors and other hardware devices are working properly and to log acquired data for future analysis.

The sensors, which provide information of vehicle movement, will be checked to see if they provide correct values. These values will be stored in a local storage device and transmitted to the other computer for backup. Thrusters will be turned on in short intervals (for example, 30 seconds for each thruster). As the vehicle moves, the INS and electric compass data will be monitored. Thruster will be operated with open loop controller for the simplicity in early phase of development. The fins will be tested while the vehicle stops and moves.

Sensors: INS, electric Compass

Actuator: thrusters, fins

Test Plan 3

Goal - Test the sonar-based sensors.

The sonar-based sensors such as altimeters and scan sonar will be tested. The vehicle will be fixed at an arbitrary point to minimize disturbance to sensor signals. Operator can place objects in front of each altimeter and check the readings from the sensors. The distance of the objects from the vehicle and the size of the objects are not determined. The readings will be stored in a local storage and transmitted to the remote operator. Because scan sonar will not be used by the first phase, all the data

will be stored in a local storage for future analysis. The data can be analyzed with experimental algorithm or program, but the specific plan is not yet determined.

Sensors: altimeters, scan sonar

Actuator: none

Test Plan 4

Goal - Test the basic vehicle maneuvering function and miscellaneous functions.

The basic maneuvering function will be tested. The vehicle will move using thruster and fins based on the data from sensors. Sonar data will be monitored but will not be used in navigation until next phase starts. Only open loop control will be used. Lights will be turned on and off during navigation. The other sensors, which are not mentioned here, will be monitored and logged for future reference.

Sensors: INS, compass, depth sonar, altimeter, scan sonar, battery level.

Actuator: thrusters, fins, light switch

Publications

- Sugihara, K. and Yuh, J., "GA-based motion planning for underwater robotic vehicle," Proc. 10th Int'l Symp. on Unmanned Untethered Submersible Technology (UUST-10), Durham, NH, 1997, pp.406-415.
- Yuh, J, Choi, S.K., Ikehara, C., Kim, G.H., McMurtry, G., Nejhad, M., Sarkar, N., & Sugihara, K., "Design of a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM)," The Proceedings of the IEEE Oceanic Engineering Society Underwater Technology '98, Tokyo, JAPAN, Apr. 1998.
- Sugihara, K., "GA-based on-line path planning for SAUVIM," Proc. 11th Int'l Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-98-AIE), Castellon, Spain, 1998, pp.329-338.
- Sugihara, K. and Smith, J., "Genetic algorithms for adaptive planning of path and trajectory of a mobile robot in 2D terrain," IEICE Trans. Information and Systems, Vol. E82-D, No. 1, pp.309-317, January 1999.
- Yousefpour, A., and Ghasemi Nejhad, M. N., "Testing and Finite Element Modeling of APC-2/AS4 Thermoplastic Composite C-rings," 31st International SAMPE (Society for the Advancement of Material and Process Engineering) Technical Conference: Advanced Materials & Processes Preparing for the New Millennium, vol. 31, pp. 643-654, Chicago, IL, October 1999.
- Yuh, J. and Choi, S.K., "Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM)," Sea Technology, Oct. 1999.
- Choi, S.K., Easterday, O., Ikehara, C., Coutsourakis, C., & Yuh, J., "Development of SAUVIM," The Proceedings of the Symposium of Underwater Robotic Technologies (SURT) 2000, Maui, HI, Jun. 2000.
- Choi, S.K., Sugihara, K., Menor, S., Nip, A., & Yang, Z., "A Predictive Virtual Environment Monitor for SAUVIM," The Proceedings of the Symposium of Underwater Robotic Technologies (SURT) 2000, Maui, HI, Jun. 2000.
- Kim, T.W., Choi, S.K., Lee, J.W., West, M.E., & Yuh, J., "A Real-Time Distributed Control Architecture for AUVs," The Proceedings of the Symposium of Underwater Robotic Technologies (SURT) 2000, Maui, HI, Jun. 2000.
- Marani, G., Bozzo, T., & Choi, S.K., "A Fast Prototyping Approach for Designing the Maris Manipulator Control," The Proceedings of the Symposium of Underwater Robotic Technologies (SURT) 2000, Maui, HI, Jun. 2000.
- Ng, R., Uyema, M., Yousefpour, A., Ghasemi Nejhad, M.N., Flegal, B., and Sung, E., "Manufacturing and Testing of Shallow Water Composite Pressure Vessels for Semi-Autonomous Underwater Vehicle," World Automation Congress 2000 (WAC 2000), in press, June 2000.
- Sugihara, K. and Yuh, J., "Adaptive, intelligent motion planning system for AUV's," Proc. SURT2000, Maui, Hawaii, 2000.
- Yousefpour, A., Ng, R., Uyema, M., and Ghasemi Nejhad, M.N., "Design and Finite Element Analysis of Shallow Water Composite Pressure Vessels for Semi-Autonomous Underwater Vehicle," World Automation Congress 2000 (WAC 2000), in press, June 2000.
- Choi, S.K., Menor, S. & Yuh, J., "Distributed Virtual Environment Collaborative Simulator for Underwater Robots" The Proceedings of the International Conference on Intelligent Robots and Systems (IROS) 2000, Takamatsu, JAPAN, Nov. 2000

- Choi, S.K. & Easterday, O.T., "The Development of an Underwater Vehicle Monitoring System and Its Sensor Systems," The Proceedings of the International Symposium on Experimental Robotics (ISER), Honolulu, HI, Dec. 2000.
- Ng, R., A., Yousefpour, M., Uyema, and M. N., Ghasemi Nejjhad, "Design, Analysis, Manufacture, and Test of Shallow Water Pressure Vessels using E-Glass/Epoxy Woven Composite Material for a Semi-Autonomous Underwater Vehicle, submitted to the Journal of Composite Materials, in review, 2000.
- Yousefpour, A. and M. N., Ghasemi Nejjhad "Experimental and Computational Study of APC-2/AS4 Thermoplastic Composite C-Rings," Journal of Thermoplastic Composite Materials, in press, 2000.
- Yousefpour, A., and M. N. Ghasemi Nejjhad, "Effects of Geometric Optimization of Tapered End-caps on Thick Thermoplastic Composite Pressure Vessels for Deep Ocean Applications," Journal of Thermoplastic Composite Materials, submitted for publication, 2000.
- Choi, S.K. and Yuh, J., "A Virtual Collaborative World Simulator for Underwater Robots using Multi-Dimensional, Synthetic Environment, "Submitted to the IEEE International Conference on Robotics and Automation (ICRA) 2001, Seoul, KOREA, May 2001.
- Choi, S.K. and Yuh, J. (Editors), Underwater Robotic Vehicle Technologies, TSI Press, Albuquerque, NM, expected to be published 2001.