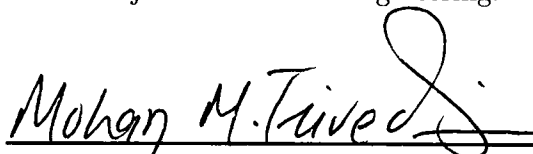


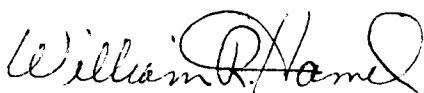
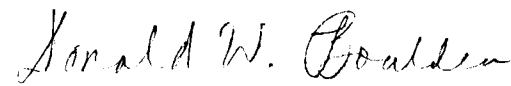
To the Graduate Council:

I am submitting herewith a thesis written by Christopher S. Gourley entitled "Sonar-Based Obstacle Avoidance, Mapping, and Graphical User Interfaces for Integrated Teleautonomous Systems." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.



Prof. Mohan M. Trivedi, Major Professor

We have read this thesis  
and recommend its acceptance:



Accepted for the Council:



Associate Vice Chancellor  
and Dean of The Graduate School

**SONAR-BASED OBSTACLE AVOIDANCE,  
MAPPING, AND GRAPHICAL USER INTERFACES  
FOR INTEGRATED TELEAUTONOMOUS SYSTEMS**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Christopher S. Gourley

May 1994

*Dedicated to my parents.*

## **Acknowledgments**

I would like to thank several of the people for their help and input into this thesis. I can not name everyone but would like to mention a few of those with key input. First, at the Robotics and Process System Division of the Oak Ridge National Laboratory Dennis Haley, Hugh Costello, Phil Bulter, and Tim Noell. and at the Computer Vision of Robotics Research Laboratory at the University of Tennessee, Scott Thayer, Fred DePiero, Nils Lassiter, and Martin Holder. I especially would like to thank my committee members, Dr. Mohan Trivedi, Dr. Bill Hamel, and Dr. Don Bouldin for their inputs and patience in the preparation of this thesis. Also, I would like to thank DOE for the grant which funded this research.

## Abstract

This thesis describes a project whose goal is to move a robot in teleautonomous mode in an unknown environment and find pipes to decommission. The project is divided into several modules to allow for porting to other platforms with a minimum of effort. While the robot moves through the environment, a low-level map, in the form of an occupancy grid, along with detailed location of pipes in the environment are obtained. This thesis deals with the low-level "reflex" of obstacle avoidance that is performed while the robot moves along its path. Also, it is concerned with some of the higher-level tasks involved in path planning and robot motion in order to negotiate through a hazardous environment. Pipe modeling is performed and verification made by overlaying the 3-D models onto actual images of the pipes. These 3-D models are easily edited and passed to previewers so that the operator can use the same controllers to manipulate both real and virtual platforms. The operators interactions with the system is made simple, user-friendly, and efficient using a graphical interface to control high-level tasks, keeping all low-level communications and sensor processing transparent.

The experimental testbed utilizes a wheeled mobile robot with ultrasonic range sensors. A fast and efficient obstacle avoidance algorithm has been created using sixteen ultrasonic range sensors. sensor using weighted sums of the sensor readings. This allows the robot to negotiate around obstacles quickly. This robot may not be capable of precise mapping due to its inaccuracy, but does serve as a very good testbed. Extensive experimentation has been performed with the modeling modules and the mobile platform. The integration has been performed and a testbed of pipes developed to show the operation and capabilities of the initial system. The results show robust operation in a variety of situations within the limits imposed upon it by positional inaccuracy.

## TABLE OF CONTENTS

| CHAPTER                                                                    | PAGE      |
|----------------------------------------------------------------------------|-----------|
| <b>1. Introduction . . . . .</b>                                           | <b>1</b>  |
| 1.1 Research Problem . . . . .                                             | 1         |
| 1.2 Research Methodology and Plan . . . . .                                | 2         |
| 1.3 Research Objectives . . . . .                                          | 5         |
| 1.4 Overview . . . . .                                                     | 7         |
| 1.5 Unique Contributions . . . . .                                         | 8         |
| 1.5.1 Human-Machine System Architecture . . . . .                          | 8         |
| 1.5.2 3-D Graphical Task Modeling . . . . .                                | 8         |
| 1.5.3 Mobile Robotic Platform . . . . .                                    | 9         |
| <b>2. Literature Review . . . . .</b>                                      | <b>10</b> |
| 2.1 Modeling . . . . .                                                     | 12        |
| 2.2 Telerobotics . . . . .                                                 | 13        |
| 2.3 Obstacle Avoidance . . . . .                                           | 15        |
| 2.4 Motion Planning and Navigation . . . . .                               | 23        |
| 2.5 User Interface . . . . .                                               | 25        |
| 2.6 Related Studies and Contributions of this Thesis . . . . .             | 26        |
| <b>3. Research Methodology . . . . .</b>                                   | <b>27</b> |
| <b>4. Hardware Configurations for Testing and Implementation . . . . .</b> | <b>30</b> |
| 4.1 Initial Testing . . . . .                                              | 30        |

| <b>CHAPTER</b>                                           | <b>PAGE</b> |
|----------------------------------------------------------|-------------|
| 4.2 Second Hardware Configuration . . . . .              | 34          |
| 4.3 Mobile Hardware . . . . .                            | 36          |
| <b>5. Workspace Modeling . . . . .</b>                   | <b>42</b>   |
| 5.1 Position and Orientation . . . . .                   | 42          |
| 5.2 Camera Model . . . . .                               | 48          |
| 5.3 Coordinate Frames . . . . .                          | 51          |
| 5.4 Error Correction and Knowledge Enhancement . . . . . | 52          |
| <b>6. Task Planning and Graphical Preview . . . . .</b>  | <b>55</b>   |
| <b>7. Mobile Robotic Subsystem . . . . .</b>             | <b>60</b>   |
| 7.1 Obstacle Avoidance Algorithm . . . . .               | 61          |
| 7.1.1 Obstacle Avoidance . . . . .                       | 63          |
| 7.1.2 Path Planning Strategy . . . . .                   | 68          |
| 7.2 Implementation and System Configuration . . . . .    | 70          |
| 7.2.1 User Interface . . . . .                           | 70          |
| 7.2.2 Communications . . . . .                           | 72          |
| 7.2.3 Robot Motion . . . . .                             | 72          |
| 7.3 Integrated System . . . . .                          | 73          |
| <b>8. Experimentation and Results . . . . .</b>          | <b>76</b>   |
| 8.1 Positioning . . . . .                                | 76          |
| 8.2 Obstacle Avoidance . . . . .                         | 92          |
| 8.3 Integrated System . . . . .                          | 97          |
| <b>9. Concluding Remarks . . . . .</b>                   | <b>100</b>  |

| CHAPTER                           | PAGE |
|-----------------------------------|------|
| 9.1 Future Applications . . . . . | 102  |
| BIBLIOGRAPHY . . . . .            | 103  |
| APPENDICES . . . . .              | 109  |
| A. Error Studies . . . . .        | 110  |
| VITA . . . . .                    | 112  |

## LIST OF TABLES

| <b>TABLE</b> |                                                                                                                                                       | <b>PAGE</b> |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| 2.1          | <i>Occupancy grids vs. Image processing [Elf89] . . . . .</i>                                                                                         | 18          |
| 7.1          | <i>Distance thresholds associated with each sensor to create two rings around<br/>the robot for use in variable speed obstacle avoidance. . . . .</i> | 66          |
| 8.1          | <i>Error measurement in position versus distance traveled. . . . .</i>                                                                                | 82          |
| 8.2          | <i>Error measurement of position and heading versus velocity. . . . .</i>                                                                             | 82          |
| 8.3          | <i>Error measurement of position and heading versus acceleration. . . . .</i>                                                                         | 83          |
| 8.4          | <i>Average speeds of the robot in different environments. . . . .</i>                                                                                 | 94          |
| 8.5          | <i>Distance maintained from objects using the obstacle avoidance algorithm. . . . .</i>                                                               | 96          |

## LIST OF FIGURES

| FIGURE                                                                                                                                                                                                                       | PAGE |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 1.1 <i>A typical unstructured operating environment showing the presents of cylindrical objects including tanks and pipes. These environments are often cluttered and have occlusions and illumination problems. . . . .</i> | 2    |
| 1.2 <i>Teleoperated system dependent upon the human operator for control and perception. . . . .</i>                                                                                                                         | 3    |
| 1.3 <i>Autonomous system in which all control and perception is handled by the robot through pre-programmed algorithms. . . . .</i>                                                                                          | 4    |
| 1.4 <i>Teleautonomous system in which the human and machine interact to create optimal control and perception capabilities. . . . .</i>                                                                                      | 4    |
| 1.5 <i>System design of the integrated system using modular components in which each of the modular subsystems can easily be integrated into another system as needed. . . . .</i>                                           | 7    |
| 2.1 <i>The telerobotic concept in which a human operator is separated from the robot manipulator, controlling it using feedback, such as vision from cameras. . . . .</i>                                                    | 14   |
| 2.2 <i>A sample occupancy grid. This particular grid was made starting the the laboratory on the third floor of Ferris Hall and then allowing the robot to proceed down the hallway. . . . .</i>                             | 17   |
| 3.1 <i>The environment is modeled through the cooperation of human and machine in both high and low level tasks. . . . .</i>                                                                                                 | 28   |

| FIGURE                                                                                                                                                                                                | PAGE |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 4.1 <i>Two degree of freedom stereo vision module to be ported to different robotic platforms for modeling capabilities. . . . .</i>                                                                  | 31   |
| 4.2 <i>The Advanced Servo Manipulator located at the Robotics and Process Systems Division at the Oak Ridge National Laboratory used in initial testing of the modeling subsystem. . . . .</i>        | 32   |
| 4.3 <i>VME rack for real time imaging including processors and image processing boards. . . . .</i>                                                                                                   | 33   |
| 4.4 <i>The ASM-based D &amp; D hybrid telerobotic system. . . . .</i>                                                                                                                                 | 34   |
| 4.5 <i>Cincinnati Milicron T3-726 used for testing the portability of the modeling subsystem. . . . .</i>                                                                                             | 35   |
| 4.6 <i>Hardware configuration for use with the T3-726. . . . .</i>                                                                                                                                    | 35   |
| 4.7 <i>Hardware configuration for the mobile robotic research platform. . . . .</i>                                                                                                                   | 37   |
| 4.8 <i>Mobile robotic platform, ELVIS, used for mapping and obstacle avoidance algorithm testing. . . . .</i>                                                                                         | 37   |
| 4.9 <i>Velocity profile using HP servo chips on board the mobile platform for (1) straight line motion and (2) turning motion. . . . .</i>                                                            | 38   |
| 4.10 <i>Distance calculations via time of flight sensors. . . . .</i>                                                                                                                                 | 39   |
| 5.1 <i>Graphical Task Model Interface used by the operator for manipulation of the pipe model as well as selecting manipulation points along the pipes for radiation scanning or cutting. . . . .</i> | 43   |
| 5.2 <i>Translation and rotation of pipe model into proper position. . . . .</i>                                                                                                                       | 44   |
| 5.3 <i>(1)Rotation about Y by <math>\theta</math>, (2)Rotation about X' by <math>\psi</math>, (3)Rotation about Z'' . . . . .</i>                                                                     | 45   |
| 5.4 <i>Derivation of rotation angles from the desired position of the pipe. . . . .</i>                                                                                                               | 46   |

| FIGURE                                                                                                                                                                                | PAGE |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 5.5 <i>Pipes to be modeled in a typical environment. . . . .</i>                                                                                                                      | 47   |
| 5.6 <i>Pinhole camera model. . . . .</i>                                                                                                                                              | 48   |
| 5.7 <i>Field of view calculation. . . . .</i>                                                                                                                                         | 50   |
| 5.8 <i>Graphical Task Model Interface with color coded pipes overlain onto the actual images of the environment to ease the operators ability to group pipes. . . . .</i>             | 53   |
| 6.1 <i>Graphical model of ASM used in the virtual environment along with real ASM used in initial testing. . . . .</i>                                                                | 56   |
| 6.2 <i>Virtual ASM Interface in which the operator can plan, preview, and teach.</i>                                                                                                  | 57   |
| 6.3 <i>Individual joint control for the ASM Interface. . . . .</i>                                                                                                                    | 58   |
| 6.4 <i>Viewing parameter control for the virtual ASM interface. . . . .</i>                                                                                                           | 59   |
| 6.5 <i>Overhead view of the Graphical model of ASM used in the virtual environment not normally viewed with real ASM. . . . .</i>                                                     | 59   |
| 7.1 <i>Waveforms associated with ultrasonic range sensors. . . . .</i>                                                                                                                | 62   |
| 7.2 <i>Occupancy probability model used in creating occupancy grids from ultrasonic range data. . . . .</i>                                                                           | 64   |
| 7.3 <i>Thresholds for variable speed and reaction determined from the range data of the ultrasonic sensors positioned around the robot. . . . .</i>                                   | 65   |
| 7.4 <i>Sensor vector summation of range readings added with the desired direction to form an overall direction to avoid obstacles while proceeding toward the goal. . . . .</i>       | 67   |
| 7.5 <i>Results of a typical experiment in path planning in which the operator has chosen via points and the obstacle avoidance must take over to avoid an unknown object. . . . .</i> | 68   |

| FIGURE                                                                                                                                                                                                                                                                            | PAGE |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 7.6 Graphical user interface for the mobile robot including an up-to-date occupancy grid, current sensor readings, recent camera view, robot control, and vital information. . . . .                                                                                              | 71   |
| 7.7 Flow chart of typical run of the integrated system. The robot moves into the environment. Candidate objects are found. If they are pipes, they are modeled. Occupancy grids are created in parallel. . . . .                                                                  | 75   |
| 8.1 Test path followed by mobile platform during positional accuracy testing.                                                                                                                                                                                                     | 77   |
| 8.2 Originally proposed robot calibration technique in which hand measurements from a fixed position were to be performed. . . . .                                                                                                                                                | 78   |
| 8.3 Robot geometry used in determining the offset of the internal coordinate frame along with the measurement points used in the calibration. . . . .                                                                                                                             | 79   |
| 8.4 Modified robot calibration incorporating a fourth measurement in order to reduce the amount of computation needed in calculating the robots actual position. . . . .                                                                                                          | 80   |
| 8.5 (1) Average error versus velocity for positional offset, (2) Standard deviation of error versus velocity for angular offset, (3) Average error versus acceleration for positional offset, and (4) Standard deviation of error versus acceleration for angular offset. . . . . | 84   |
| 8.6 Positional error due to acceleration of (1) 100 mm/sec <sup>2</sup> , (2) 250 mm/sec <sup>2</sup> and, (3) 1000 mm/sec <sup>2</sup> . . . . .                                                                                                                                 | 85   |
| 8.7 Angular error due to acceleration of (1) 100 mm/sec <sup>2</sup> , (2) 250 mm/sec <sup>2</sup> and, (3) 1000 mm/sec <sup>2</sup> . . . . .                                                                                                                                    | 86   |
| 8.8 Positional error due to velocity of (1) 200 mm/sec., (2) 250 mm/sec., (3) 500 mm/sec. and, (4) 1000 mm/sec. . . . .                                                                                                                                                           | 87   |

## FIGURE

## PAGE

|      |                                                                                                                            |    |
|------|----------------------------------------------------------------------------------------------------------------------------|----|
| 8.9  | <i>Angular error due to velocity of (1) 200 mm/sec., (2) 250 mm/sec., (3) 500 mm/sec. and, (4) 1000 mm/sec. . . . .</i>    | 88 |
| 8.10 | <i>An occupancy grid create by moving the robot at 150 mm/sec with an acceleration of 500 mm/sec<sup>2</sup>. . . . .</i>  | 90 |
| 8.11 | <i>An occupancy grid create by moving the robot at 350 mm/sec with an acceleration of 1000 mm/sec<sup>2</sup>. . . . .</i> | 90 |
| 8.12 | <i>An edge map showing the walls from the robot moving at 150 mm/sec. .</i>                                                | 91 |
| 8.13 | <i>An edge map showing the walls from the robot moving at 350 mm/sec. .</i>                                                | 91 |
| 8.14 | <i>A map showing the actual walls that the robot has detected as obstacles.</i>                                            | 92 |
| 8.15 | <i>Hallway down which the robot was tested. . . . .</i>                                                                    | 93 |
| 8.16 | <i>Paths followed by the robot when it was in roam mode leaving the lab and proceeding down the hallway. . . . .</i>       | 95 |
| 8.17 | <i>Cluttered laboratory environment in which the robot navigated. . . . .</i>                                              | 95 |
| 8.18 | <i>Pipes to be modeled in a typical environment. . . . .</i>                                                               | 97 |
| 8.19 | <i>Models created from the integrated system with the stereo cameras mounted on the mobile platform. . . . .</i>           | 99 |

# CHAPTER 1

## Introduction

### 1.1 Research Problem

As technology advances, machines are being called upon to aid humans leading to increases in productivity and safety. One problem that has been highly emphasized in recent years is that of environmental safety and restoration. Much effort, time, and money has been infused into this area. The U.S. Department of Energy's Environmental Restoration and Waste Management (ER&WM) program has as one of its important requirements Decontamination and Decommissioning (D&D) [But92] of old nuclear facilities [oE89]. Robots are the perfect instruments to reduce the risk to humans in these hazardous radioactive environments. These facilities are highly unstructured and cluttered, as can be seen by the typical operating environment shown in Figure 1.1, by many cylindrical objects such as pipes. Most have no models available for use in planning strategies to enter the area and dismantle old equipment. Therefore, a means by which a robot can navigate through such an environment containing unknown objects while simultaneously creating maps and data bases as to the contents of the environment would be highly beneficial for robots which would later enter to remove objects. To develop a system to map such a complex area requires the integration of different sensor modalities along with a mobile platform. To lessen the complexity of the system's modeling capabilities, it is limited to modeling of pipes only for reasons to be discussed later.

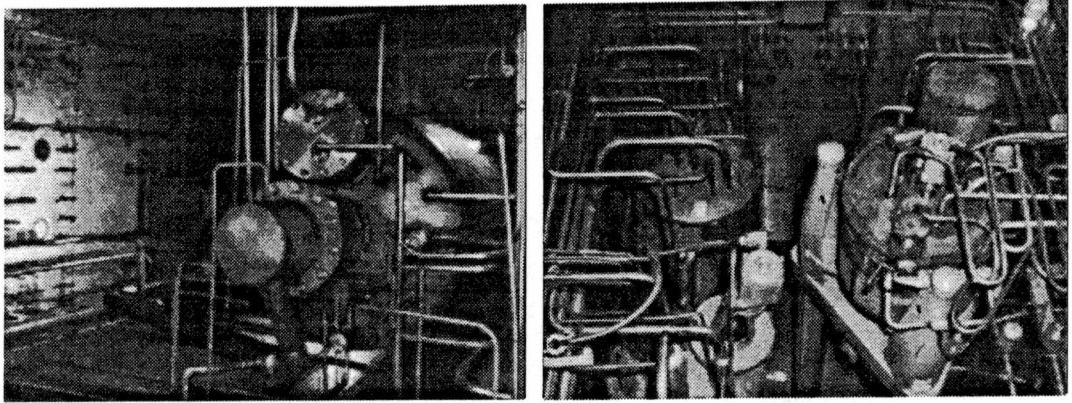


Figure 1.1: *A typical unstructured operating environment showing the presents of cylindrical objects including tanks and pipes. These environments are often cluttered and have occlusions and illumination problems.*

Presented in this thesis is the initial development of such a system. This system is by no means the answer to the problem, but does begin to look at the direction for possible solutions. Also, an initial testbed setup and implementation is used for preliminary experimentation. Much of the state-of-the-art research has not focused on the problem of integration. In the field of mobile robotics, much emphasis has been placed on speed in obstacle avoidance. Task preview capabilities have just begun to enter into the picture as recent advances in the speed of specialized graphics computers have become available. Wrapping all of these into a nicely integrated system is an enormous task which is only begun here.

## 1.2 Research Methodology and Plan

The control of any such system lends itself to different philosophies. In the past, most systems have been developed to be completely autonomous or completely teleoperated, ideologies located at opposite ends of the spectrum. The teleoperated philosophy (see

Figure 1.2) depends completely upon the human operator for control. The operator is

| Human System   |                                                                                   |         |
|----------------|-----------------------------------------------------------------------------------|---------|
|                | Human                                                                             | Machine |
| Perception:    |  |         |
| Task Planning: |  |         |
| Motor/Control: |  |         |

Figure 1.2: *Teleoperated system dependent upon the human operator for control and perception.*

able to view the environment either directly or by means of cameras and manipulate the robot in the environment, like a remote-controlled car. The autonomous philosophy, on the other hand, depends completely upon the robot to make all decisions through AI algorithms (see Figure 1.3). In this case, the operator is only needed to initialize the system and send the robot on its way. Both of these cases give good results in some situations, but they are limited in others; thus, recently a third philosophy has entered the ring, that being teleautonomous control, which attempts to merge the best of both the previous manners of thinking (see Figure 1.4). Here the best of both worlds is exploited. The operator and the machine perform the tasks most suited for each allowing overall optimal performance.

| Machine System |       |                                                                                     |
|----------------|-------|-------------------------------------------------------------------------------------|
|                | Human | Machine                                                                             |
| Perception:    |       |  |
| Task Planning: |       |  |
| Motor/Control: |       |  |

Figure 1.3: *Autonomous system in which all control and perception is handled by the robot through pre-programmed algorithms.*

| Human-Machine System |                                                                                     |                                                                                       |
|----------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|                      | Human                                                                               | Machine                                                                               |
| Perception:          |  |   |
| Task Planning:       |  |                                                                                       |
| Motor/Control:       |                                                                                     |  |

Figure 1.4: *Teleautonomous system in which the human and machine interact to create optimal control and perception capabilities.*

### 1.3 Research Objectives

What is discussed in this thesis is a modular integrated system that is able to go into an unstructured environment and, with the aid of an operator, create a map of the objects in the area through occupancy grids and detailed graphical models of the positioning of the pipes to be decommissioned. Also, a means is available to cut down on the hazardous waste stream and save money through the use of graphical previewers. In the philosophy of Object Oriented Programming, the system is comprised of several lower-level modules. The components needed for this system are

- Pipe localization,
- Modeling capabilities,
- Preview capabilities, and
- Mobile platform.

This thesis is concerned mainly with the final three components, the first already developed. Each of these modules has a simple high-level interface for the user. All of the low-level communication and algorithms are transparent. The low-level complexity leads to high-level simplicity allowing ease of integration by the user at the highest level.

The system developed is a mobile robot utilizing a set of range sensors and a proximity sensor to safely navigating through the environment while mapping the environment with occupancy grids. A pair of cameras are mounted on the robot to perform localization of pipes using stereo vision. Using this, models of the environment are built which can be used for decontamination and decommissioning of the old pipes [Tha93b, Tha92].

The pipe localization described in this thesis is a very specific task. It can not model other objects which are in the environment at this time. The mobile platform described here does not perform specific object recognition like pipe detection. The overall system, however, is very powerful being capable of navigating through its environment, creating a coarse map of the environment and detailed positioning of pipes in the environment which can later be used for scanning and cutting of the pipes.

In recent years the topic of mobile robotics has become an area of much research interest [CTB92, CT92a], as is evident by the topic of the thesis. Advances in hardware, both sensory and computational, as well as artificial intelligence has cultivated the sudden interest and advancement in this area [DB93]. One niche that this thesis develops is the integration of the various modalities along with the interaction and interface with the human operator. The overall goals of the design were:

1. Simplicity at the highest level,
2. Build the system in such a manner that the use thereof is intuitive, and
3. Each *subsystem* or module of the system is designed to be of optimal use and modularity,

This modularity and high-level interaction gives the advantage at the next level of ease of integration as well as portability. Figure 1.5 shows how each module of the system fits together to form the overall high-level tool.

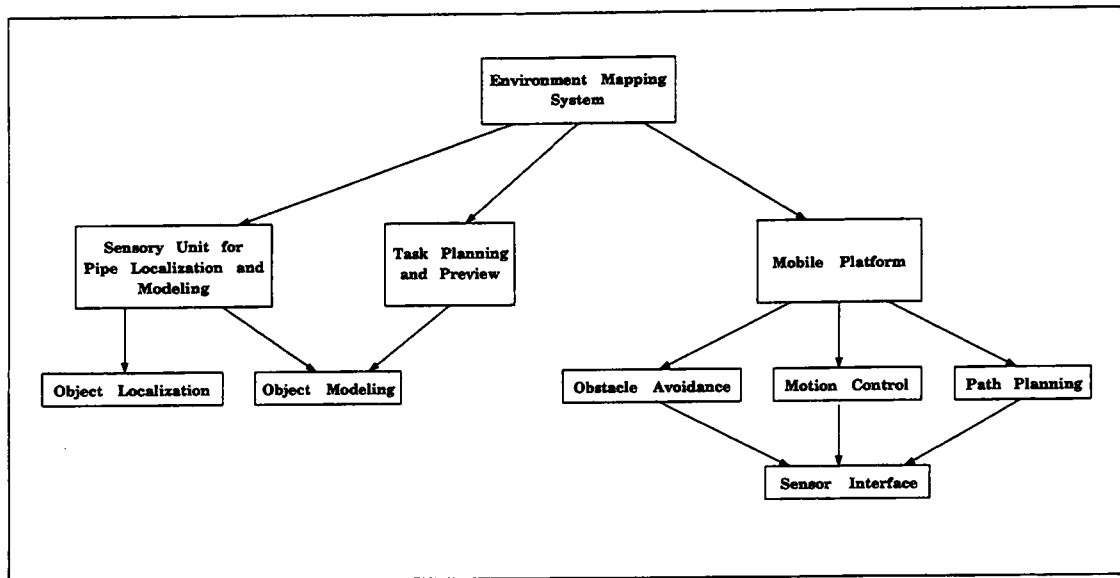


Figure 1.5: *System design of the integrated system using modular components in which each of the modular subsystems can easily be integrated into another system as needed.*

## 1.4 Overview

The remainder of this thesis is presented as follows. First, a review of the relevant literature to this project is discussed. An overview of the hardware used in the development and testing of the system is then presented. Pipe modeling techniques are then discussed followed by task planning and preview capabilities. The next chapter discusses the mobile platform used, along with its mapping and obstacle avoidance abilities. The high-level integrated system is then presented along with experimental results. The last chapter gives a summary and concluding remarks about the research.

## **1.5 Unique Contributions**

The main idea governing this thesis is to allow a simple and high-level interface for the operator to control the robot teleautonomously. Different control philosophies are used so that optimal production can be gained from a variety of situations. The key ingredient to each of the subsystems developed is their ease of integration accomplished through their human-machine interface. This tool in each case operates completely at the high-level. Low-level processing such as opening and closing communication sockets, RPC links across the network, and obstacle avoidance are made completely transparent.

### **1.5.1 Human-Machine System Architecture**

The human-machine system concept and its implementation follows a totally integrated architecture. That is, all subassemblies, interfaces, communication protocols, and algorithms are developed with a modular systems concept for ease of integration and are successfully demonstrated. Interaction between the human and the machine occurs at the highest level possible so as not to involve the human in the mundane and repetitive tasks that need not concern him. Virtual worlds allow the operator to interact more easily with the machine in that the operator does not have to risk damage to the robot or the environment as he would in the real world.

### **1.5.2 3-D Graphical Task Modeling**

This is an important module of the overall system. This is the main medium for human-machine communication in the vision subsystem and for transfer of control between them. It also allows maintenance of 3-D graphical models of the structural and kine-

matic behaviors as well as providing a means of interactions between the robot and its workspace by allowing selection of points on the models to be manipulated.

### **1.5.3 Mobile Robotic Platform**

The mobile platform in itself is unique being the only one of its kind presently, a completely tetherless wheeled mobile platform equipped with multiple sensors controlled from an on board VME rack. The key contributions from the mobile platform research are:

- Reflexive avoidance,
- Variable speed,
- Low-level transparency, and
- High level interface.

The mobile platform is integrated with a vision system that has been tested on a variety of platforms proving itself flexible and robust while being accurate and efficient. Extensive experimentation of both systems separately as well as the integrated system has been performed and will be presented in this thesis.

## CHAPTER 2

### Literature Review

The development of any system would not be wise without studying methods already developed relative to the area of study. Much of the original scope of the project was limited by the initial goal imposed, intelligently cut pipes using a teleoperator. This thesis intends to integrate some low level systems that, at first, may appear to be completely unrelated, that being computer modeling and robotic obstacle avoidance; therefore, this chapter may seem to be rather disjoint at first. Much work has been done in the area of mobile robot navigation. A full review of the work within the last few years in this area would take an entire book. For this reason, only the most notable and successful are discussed in detail. A study of various works also shows that much of the obstacle avoidance is accomplished using ultrasonic range sensors. Some other methods, including using touch and vision sensors has also been done. The most relevant obstacle avoidance methods come from edge-detection, certainty grids, and potential field methods.

Before going into any specific research findings, some general comments about the current state of technology will be stated. One comment that greatly summarizes much of the present work relative to what the general public perceives was made at a recent robotics competition.

I think it's good to have people outside robotics realize how hard the problems are, at least by seeing what the realistic state-of-the-art robots are like

today. We're just not close to R2D2's and C3PO's yet, and people should realize why not. — *Maja Mataric*

To do mobile robot research there are two things that you must have: a mobile robot and sensors. Miller [Mil93] has emphasized the point that the use of complicated sensors leads to complicated algorithms for navigation and planning. He has devised a twelve step program for sensors abusers. It is as follows:

1. Admit you have a problem. Don't be afraid, almost everyone does.
2. Analyze the robot's real goals. Is the robot's goal to build a map and use it to navigate to the office next door or is it just to get to the office next door?
3. Question your need for data. Are the data necessary for your algorithm or necessary to generate figures for your paper?
4. Have the robot remember what it has done—but not for long. Motor history can provide much useful information for your robot and really is simple to gather, but don't become an internal-state abuser!
5. Solve problems mechanically. Some sensing and control problems can be avoided by putting new shocks on your robot, but beware of the monster truck syndrome.
6. Ask if this sensor is the right one for you. If all you want to know is whether there is a big obstacle in front of you, stereo vision might not be the best idea.
7. Don't be afraid to experiment. Sensors don't always deliver what you think they will; test them out.

8. Avoid simulations. They are bound to succeed.
9. Avoid color monitors. They are bound to distract you.
10. Don't be afraid to have your robot ask questions. There are no stupid questions, only stupid robots.
11. Never forget that you have a problem. Just because the price on the camera system has come down does not mean that you have to have it.
12. Acknowledge that there is a higher power; it controls your funding; and it really doesn't care how your robot does something (and usually doesn't care what this something is), just as long as it does it.

From this it is easy to see that the sensors must be chosen that best fits the job. For the task of obstacle avoidance, the robot needs to know the approximate position of an obstacle relative to the robot. Ultrasonic sensors are ideal for this task and are therefore used in this case. They lead to fairly quick and accurate detection of objects. The object geometry and features need not be known; merely the fact that there is an object at position  $(x,y)$  relative to the robot. The sensor selection was also limited by the equipment available. The Labmate platform as well as the ultrasonic sensors were already available and unused.

## 2.1 Modeling

This part of the project was defined by restriction from the contract monitor. Pipes had to be modeled and in this case they were chosen to be modeled as cylinders since

it is simple. The modeling package was the first subsystem to be developed. Not many papers have been written on how to draw a cylinder. Most of the background for this was taken from text books, including Foley [Fol90] and Gonzalez [GW87]. To simplify modeling the 3D model is represented by polygons, more specifically a number of solid rectangles make up the sides of the cylinder. Kim [Kim85] and others have shown that overlaying computer-generated graphics onto actual video displays aids the operator in the comprehension of depths.

## 2.2 Telerobotics

A key ingredient in any system, and especially in a teleoperated system, is the interaction between the human operator and the machine [She89, CT92b].

Telerobotics is a form of teleoperation in which a human operator acts as a supervisor, intermittently communicating to a computer information about goals, constraints, plans, contingencies, assumptions, suggestions and orders relative to a limited task, getting back information about accomplishments, difficulties, concerns, and as requested, raw sensory data—while the subordinate telerobot executes the task based on information received from the human operator plus its own artificial sensing and intelligence. [She74]

Most recent studies have emphasized developing systems which are viewed as integrate human-machine systems [TC93a] performing four tasks:

1. Perception,
2. Motor,

3. Planning, and

4. Control.

In such a system the operator is a key element provided with displays for telepresence and controls for teleoperation. Telerobotic manipulators (see Figure 2.1) have been

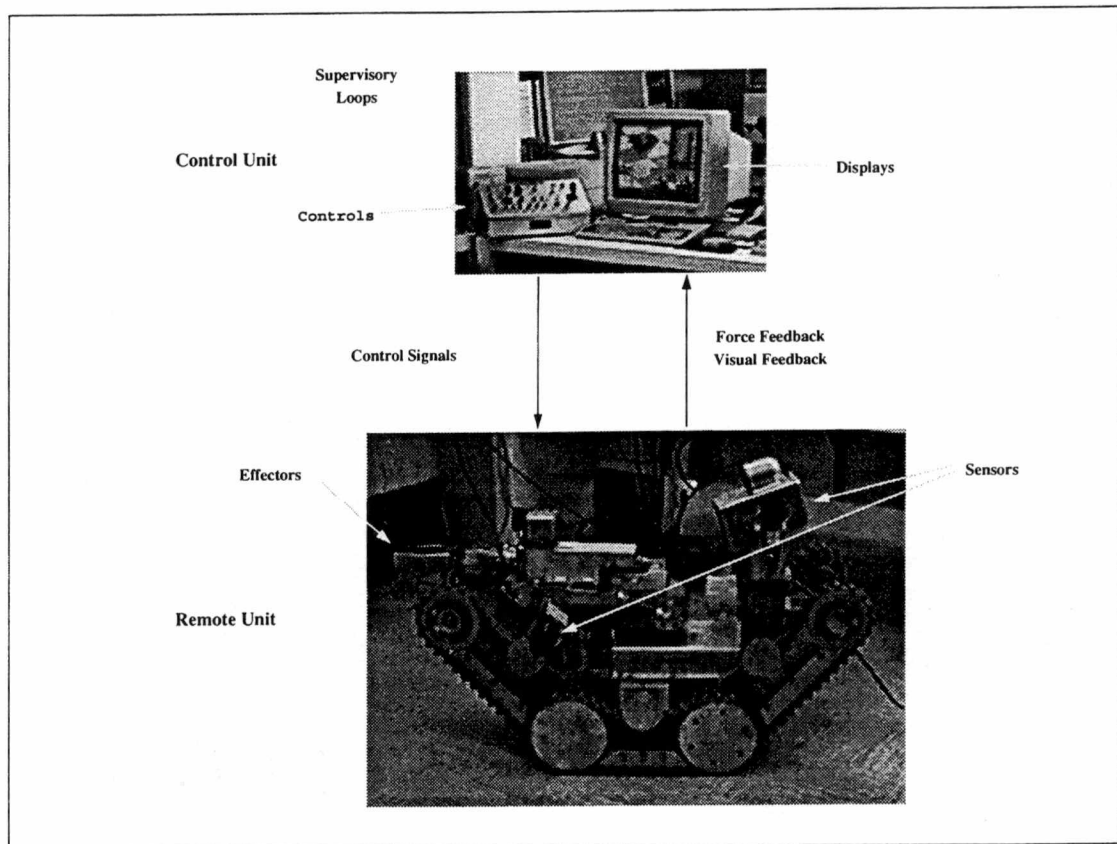


Figure 2.1: *The telerobotic concept in which a human operator is separated from the robot manipulator, controlling it using feedback, such as vision from cameras.*

used in the areas of space, undersea, nuclear power, surgery, construction, policing, and military operations. It is comprised of sensing, actuating, supervisory control, and human-machine interaction. Sensing includes vision, resolved force, touch, kinesthesia, proprioception, and proximity. Recently the sense of sound [Apo92] has been added to

teleoperated systems to assist the operator further.

Teleautonomous control [BK90] is the next logical step in teleoperation allowing the robot to perform tasks with the precision not available with teleoperation alone. For force-feedback in a teleoperator, slop must be built into the system. This automatically removes any ability to perform precise movements repeatedly. The positional control system for a teleoperator does not have to be very precise since the operator can visually correct an errors that occur in the position of the end effector. Teleautonomous control gives some of the control back to the robot.

## 2.3 Obstacle Avoidance

For a mobile robot to move though an environment, it must first be able to sense the objects in the environment and negotiate around them. The sensing can take on many different forms including vision, proximity, range, and touch. Rao [Rao93] uses vision to create graphs of the environment from which obstacle avoidance can then be performed. Also a means by which touch sensors are used to avoid obstacles is also presented. The first papers referred to in this area were from Elfes and his work with occupancy grids which turns out to be the starting point for most of the notable work done recently in the area of obstacle avoidance. Much of the work done in the area of certainty grids was accomplished by Elfes [Elf89]. Due to the noisy information given by ultrasonic sensors, certainty grid were used early on for obstacle avoidance [Elf87]. The occupancy grid is a multidimensional random field that maintains stochastic estimates of the occupancy state of the cells in a spatial lattice. Bayesian estimation procedures allow incremental updating of the occupancy grid using data from all the sensors from

multiple views[Elf89, Elf87]. The basic concept behind the occupancy grid is to create a two dimensional representation of the probability that an area of the grid contains an object. The grid can be developed using a variety of sensors. In this case, only ultrasonic range sensors are used. Using multiple sensors, multiple readings from each sensors, and sensor models taking Gaussian noise into account the occupancy grid can be iteratively updated to include new information. Using prior maps, the robot can begin with a grid containing objects known to exist. This can then be used for path planning. As the robot navigates through the environment, new objects can be added. Decision making as to whether an occupancy cell contains an object can be as simple as a binary function. If the probability of occupancy is greater than one half, the cell is considered occupied; otherwise, it is not. In this project a slightly different approach will be taken. A dead band between a probability of about 0.3 to 0.7 will be used for an unknown state of the cell. Figure 2.2 gives an example of an occupancy grid created in the lab by while the robot was roaming. Different grey levels represent varying occupancy values with white having an occupancy value of 0 (no objects) and black a value of 1 (object present). Each cell represents  $50\text{ mm}^2$  of floorspace.

Using the occupancy grid as a pseudo image, many basic image processing techniques can be used. Table 2.1 (from [Elf89, Elf87]) gives a good comparison between the two. For this reason the occupancy grid is stored using a C++ image class with many basic image processing techniques available.

For Elfes's work, however, the robot was required to stop in order to collect a set of ultrasonic data and then move to the next location since his obstacle avoidance was based upon a completed occupancy grid.

Occupancy grids need not be limited to ultrasonic sensors. Single line stereo has

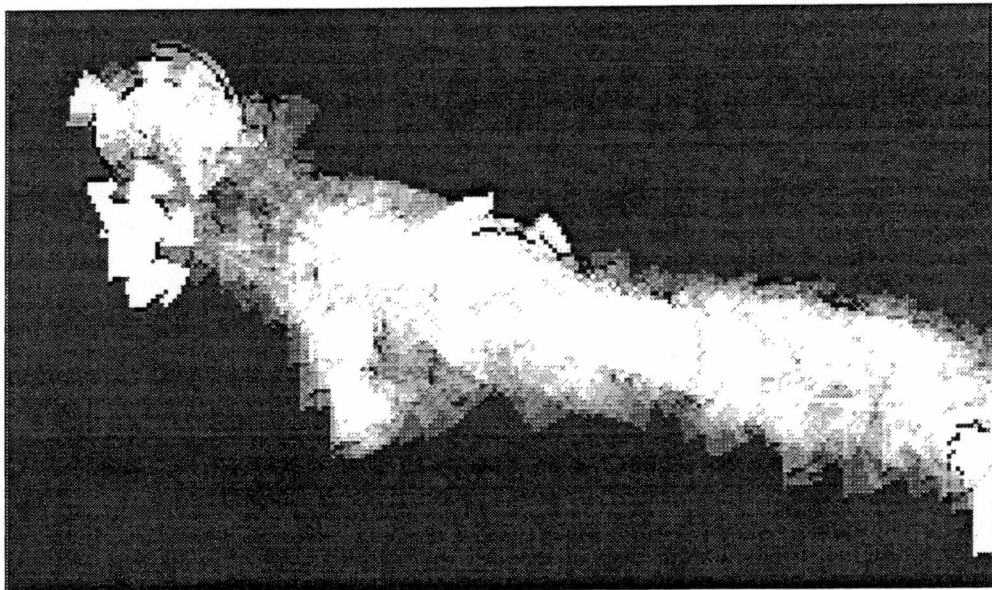


Figure 2.2: *A sample occupancy grid. This particular grid was made starting the the laboratory on the third floor of Ferris Hall and then allowing the robot to proceed down the hallway.*

Table 2.1: *Occupancy grids vs. Image processing [Elf89]*

| Occupancy Grids                                  | Images                |
|--------------------------------------------------|-----------------------|
| Labeling cells as Occupied, Empty or Unknown     | Thresholding          |
| Handling position uncertainty                    | Blurring/Convolution  |
| Removing spurious spatial readings               | Correlation           |
| Motion solving                                   | Region growing        |
| Obstacle growing for path planning               | Edge tracking         |
| Determining object boundaries                    | Edge detection        |
| Extracting and labeling occupied and empty areas | Segmentation/Labeling |
| Prediction of sensor readings from user maps     | Convolution           |
| Incorporation user maps                          | Scan-conversion       |
| Object motion detection over map sequences       | Space-time filtering  |
| Path planning                                    | Skeletonization       |
| Obstacle growing for path planning               | Dilation              |

also been used for the creation of occupancy grids [ME88]. Occupancy grids also need not be limited to two dimensions. Three dimensional occupancy grids have also been researched [Jon93].

The recent advances in technology led the American Association for Artificial Intelligence (AAAI) to sponsor a robot exhibition and competition [DB93] where the leading robots in the country met to determine which was the best. The robots were to navigate in a dynamic unstructured cluttered environment in order to find ten objects (eight feet tall bar-coded poles) and later be able to return to them. The competition's contestants were composed of ten teams from the University of Michigan (Borenstein), SRI International, CMU, MIT/JPL/ISR (Mataric), Georgia Tech/Denning, NASA-JSC, IBM, Brown University, MITRE Autonomous Systems Laboratory, and the University of Chicago. From this list of contenders, the most notable were the entries from Michigan and SRI which placed first and second, respectively [Con92]. These robots are the best of the best and are worth further study. Both rely upon ultrasonic range sensors for obstacle avoidance. SRI's robot, Flakey is a custom-built robot with a maximum velocity of 500 mm/sec. Michigan's Carmel is based on a Cybermotion K2A capable of 780 mm/sec. In competition, Carmel was able to "zip" about the ring at speeds of 300 mm/sec, more than twice as fast as the other robots although it did touch objects. Each robot relies on three computers for control: Flakey has one Sun 3, one Sun Sparc, and a Z80; Carmel has two PC's and one Z80. Obstacle avoidance was accomplished by the use of fuzzy rules for Flakey and by Borenstein's methods (more later) for Carmel. For object recognition, both used CCD cameras, Flakey relying on structure lighting and Carmel on bar codes on the poles. For mapping Carmel relied on dead reckoning and the use of occupancy grids, Flakey used landmark recognition due to its quickly

accumulated dead reckoning error. Neither robot used geometric planning for navigation to a goal point, rather they headed toward the goal and used obstacle avoidance to navigate around any obstacles.

Based on the results of the AAAI competition along with the recent literature, the most noteworthy work in the area of obstacle avoidance has come from Borenstein. Several methods for obstacle avoidance have been devised over the years. One approach is the wall-following method where the robot moves alongside a wall at a specific distance. An obstacle encountered is just considered another wall to follow. Another method is based on edge detection in which edges of an obstacle are determined and the robot tries to steer around the edge. The most notable research in the area of obstacle avoidance for mobile robots has come from Borenstein and Koren as is evident from Michigan's first place win at the AAAI's robot olympics. This method is based upon the certainty grid method that was presented by Elfes and used by Elfes and Maravec for map building is the Virtual Force Field (VFF) method [BK89]. A potential field method in which the idea of imaginary forces from obstacles are exerted on the robot was suggested by Khatib [Kha85] and Krogh [Kro84]. Brooks [Bro86] also has used force field methods treating each reading as a repulsive force. In his implementation the forces are summed as vectors and if this exceeds a threshold, the robot stops and turns into the direction of the resultant vector. The VFF method is most similar to this method. It uses the values in the occupancy grid instead of the sensor readings. In this method, as the robot moves through the environment certainty grids are created. At the same time, a  $33 \times 33$  window in the grid, with the robot located at the center, is used to calculate a repulsive force to the robot, the magnitude of which is proportional to the cell weight and the distance from the robot. From this repelling force and an attracting force toward a

target a resultant direction for robot motion is determined. The force is given by

$$\mathbf{F}(i, j) = \frac{F_{cr} C(i, j)}{d^2(i, j)} \left\{ \frac{x_t - x_0}{d(i, j)} \hat{x} + \frac{y_t - y_0}{d(i, j)} \hat{y} \right\} \quad (2.1)$$

where

|            |                                           |
|------------|-------------------------------------------|
| $F_{cr}$   | force constant (repelling)                |
| $d(i, j)$  | distance between cell (i,j) and the robot |
| $C(i, j)$  | certainty level of cell (i,j)             |
| $x_0, y_0$ | robot's present coordinates               |
| $x_t, y_t$ | coordinates of cell (i,j).                |

This method was later modified into the Vector Field Histogram (VFH) method [BK91b, BK91a]. This adds a polar histogram to the VFF method with  $n$  sectors of width  $\alpha$  relative to the robot's position, mapping Cartesian coordinate onto polar. Each sector  $k$  has a value of  $h_k$  representing the polar obstacle density. Candidate "valleys" in the histogram are determined by means of a threshold. The one most closely matching the direction to the target is chosen as the direction of motion. Experimental results with CARMEL have shown average speeds of 0.58 m/sec. through a field of densely spaced obstacles. Both the VFF and VFH methods are local path planners. They do not find an optimal path to the goal point.

Recently, Borenstein has introduced a new method for firing multiple ultrasonic sensors known as error eliminating rapid ultrasonic firing (EERUF) [BK92, KW93]. This method reduces ultrasonic noise caused by environmental noise and crosstalk when multiple sensors are fired simultaneously by comparison of consecutive readings with alternating delays. Experimental results with a TRC LabMate allowed the robot to traverse a densely cluttered environment at an average speed of 0.8 m/s. This method

does require specialized hardware.

Arkin [Ark89] has introduced path planning and navigation based on motor schema motivated by the concept of schemas from psychology. In this case, a schema is defined as a primitive behavior that, when combined with other schemas, can yield more complex behaviors, not unlike subsumption. The obstacle avoidance schema is based upon localized potential fields. Path planning is accomplished by placing the schemas together to implement a piecewise linear path developed by the navigator. Arkin [Ark93] has also introduced new methods for producing avoidance behavior, that being escape and dodge. The robot first determines if a moving object is going to collide with it by taking multiple sensor readings. In dodging, the robot avoids the object by moving in a direction normal to the objects motion (step to the side). In escaping, the robot moves in a direction to maximally increase the distance to the object (run).

Audenaert [Aud92] does not perform obstacle avoidance, but rather has worked strictly on the sensors used for such a task. By using simple signal processing techniques to build specialized hardware in order to send a binary word instead of a set of pulses from an ultrasonic sensor. Matched filters and peak detection is then used for received signals to determine if the signal originated from the sensor. [PAC93] added to this by using two extra receiving transducers. By using specialized geometry for the location of the receivers, object localization was improved as well as discriminating between different types of reflectors [Bru93] then implemented the hardware needed, including the A/D converter and correlation operation, onto a single CMOS standard cell ASIC.

Unfortunately, time does not allow for the implementation of the perfect setup of hardware and software. Sensor readings could be made much faster by the use of specialized hardware and signal processing, however, more time is spent the integration of the

system using off the shelf hardware. Robot geometry and motion capabilities simplify obstacle avoidance. The piano movers problem [SS83] does not present a big problem since the robot can be modeled as a point and the objects grown in the occupancy grid to take into account the robot's size.

Leonard [LDW92] chose ultrasonic sensors because of the sensing capabilities of bats. Bats, however, use frequency modulation, doppler shifts, and other ultrasonic ranging techniques more complex than just time-of-flight. The basic tool for Leonard's approach to navigation is the use of Kalman filtering to assist in location estimation problems and categorize mobile robot navigation as a multitarget tracking problem using walls and other static objects as their targets. This system was still in development when the book was published so experimental results of this method is not available only simulated results and hand drawn models. The algorithm does fail if no targets are available for tracking. They have not been able to get robot to go through a door. The book does give a good look at the use of ultrasonic sonar versus other sensors. Vision is much more computationally expensive and infrared rangefinders are too expensive.

## 2.4 Motion Planning and Navigation

Rao [Rao86] has presented a method for robotic navigation in an unexplored environment. This method relies heavily upon graph theory [Str86, DH73]. The algorithm consists of searching the environment and iteratively updating nodes on the graph such that all obstacles are separated by edges. From the completed graph optimal paths can be chosen to move in the environment. Rao [Rao93] give a good overview of various navigation algorithms based upon graph theory. Most consist of maze searching in order

to create the nodes and edges of a graph.

Gat [Gat93] explores the role of memory (stored internal state) in autonomous control of mobile robots. For control, the robot reacts to local sensor readings and stored internal states. Planning requires the maintenance of these internal states. The high level abstraction of the states is used in planning the movements of the mobile robot. These states can be wrong though so the robot must also rely on local sensor data. "A robot can't drive with its eyes closed, but neither can it reach its destination knowing nothing about the surroundings beyond what is locally perceptible." [Gat93] The system developed performs classical planning at high levels and uses current sensor readings for reactive control.

Steer [Ste89] deals with trajectory planning for a mobile robot when the robot is limited in its path curvature based on the kinematics of the wheeled vehicle by modulating the steer angle of the front wheels. This does not presently apply to the particular platform used here but must be considered when ported to other platforms. Mataric [Mat92] uses Brook's idea of subsumption to create a hybrid system with subsumption reactive low level control and a high level path planner instead of a reactive approach to navigation. Path planning is based on the creation of a graph with nodes representing unique landmarks. The robot proceeds to a goal chosen by the operator by traveling along the edge of the graph. Vieville [Vie93] has developed a means of navigation based on inertial cues from a gyroscope and visual cues based on stereo correlation. Latombe's [LLS91] motion planning is for fixed manipulators in two dimensions, but does incorporate uncertainty in the sensing. Alexandar [AM89] analyzes the kinematics of wheeled mobile robots with the objective of determining the forward and inverse kinematics of the robot. Slippage resulting from incompatible wheel configuration results in failure of

the rolling wheel model and a modeling involving the slippage is developed.

## 2.5 User Interface

In the design of a human-machine [She92] interface one must be careful not to overload the operator with too much information. For example, a modern aircraft could possibly have over 1000 displays. No operator could possibly keep up with all that is going on. Therefore, the interface should be kept simple, yet functional, giving the operator the information he needs to perform the task at hand.

The design of a user friendly interface is highly dependent upon the psychology of human-machine interaction. Also, for this project, the user interface is the level at which the integration is performed. The interface is the high level interaction point between the operator and the machine, in the modeling case the computer and in the mobile platform case the robot. All of the low level tasks and communications is kept transparent to the operator. This gives the operator control without knowing all the where's and why's of how it works reducing the stress on the operator and amount of information of which he must keep apprised.

To reduce operator fatigue, task preview and teach/playback are available. The interaction between human and machine can be enhanced by the use of virtual reality techniques [TC93b, Tri93]. These systems allow for integration of planning and simulation as well as exchangeability of hardware and software between real and virtual environments.

## 2.6 Related Studies and Contributions of this Thesis

So far this chapter has reviewed the relevant state-of-the-art issues. Recent advances in mobile robotics have emphasized speed in obstacle avoidance. Speed for most of these systems is being able to move continuously without having to stop to take sensor readings. Some robots require up to two minutes to receive and process sensor data before they can move. From the robot competition, CARMEL was the fastest of all the robots moving at speeds up to  $300\text{mm/sec.}$ , more that twice as fast as the nearest competitor. The experimental results shown here show this platform to easily achieve these speeds without the aid of the specialized hardware or multiple processors used on other platforms. The introduction of a variable speed also gives the robot the capability of attaining higher average speeds. A key contribution given here is the users interaction with the modules through a graphical interface. The interfaces were designed so that the most used features are readily available and low level processing is kept transparent to the operator.

## CHAPTER 3

### Research Methodology

The application domain considered in this thesis requires a teleautonomous operation of a robotic system to move into a cluttered dynamic unstructured workspace and gain some knowledge of the objects therein. To accomplish this most efficiently, the human and the machine need to cooperate and interact. The existing state-of-the-art research discussed in the previous chapter does not address interaction or integration of systems, only the individual modules. Those that address human-machine interaction do so on such an abstract level that no details or relevance to any specific system is mentioned. Here, the system is designed in such a way that the operator can choose who or what performs the tasks. Figure 3.1 shows the modules of the system interacting with both human and machine to perform the task of environmental modeling.

The experimentation and results of the system presented in this thesis by no means represents the final answer to the problem of entering into an old nuclear facility to perform mapping for D&D. This is a topic which is much beyond the scope of a thesis. Here is developed the philosophies and some of the modules needed on an initial testbed. The robot used does not possess the accuracy or mobility that would be required in a true implementation of an integrated mapping system of this nature. The initial experimentation and development of the system is presented. For the system to be as flexible as possible, the human-machine interaction must be optimal. This requires the use of several different control philosophies which were outlined in the introduction,

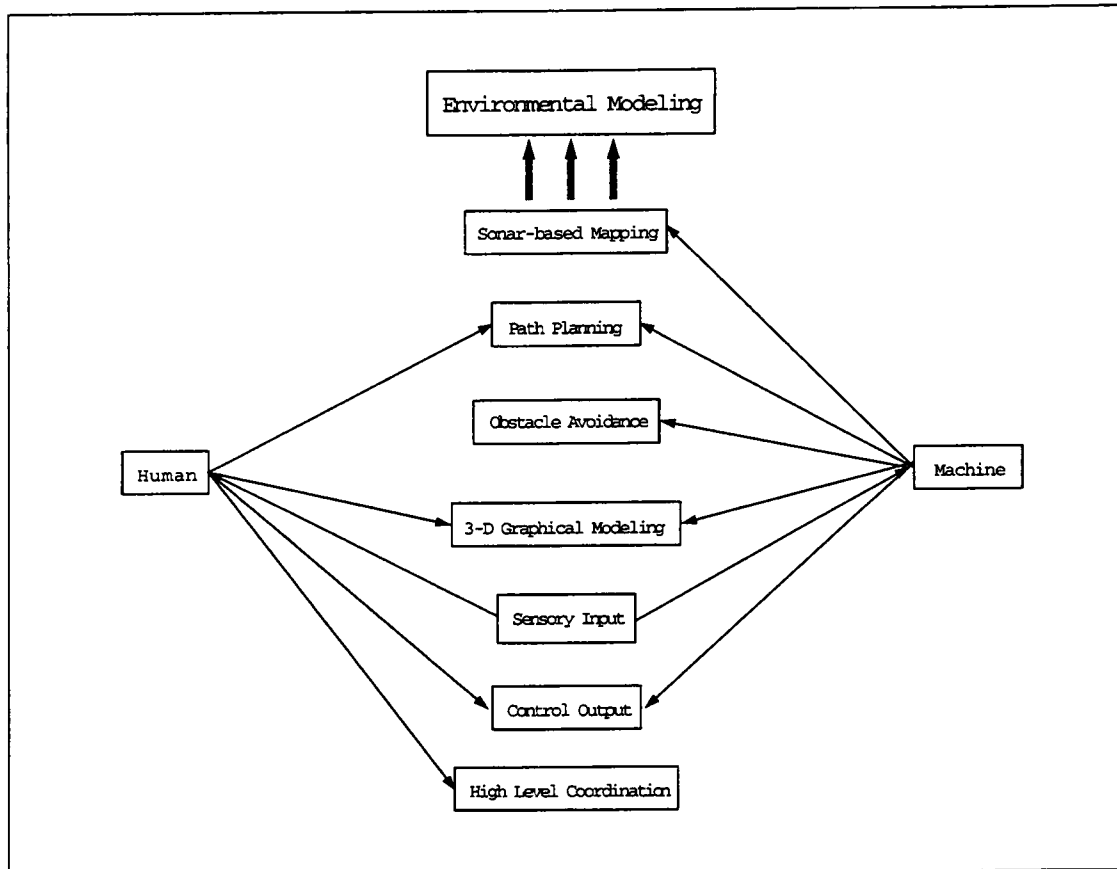


Figure 3.1: *The environment is modeled through the cooperation of human and machine in both high and low level tasks.*

as well as the ease of changing from one of these to another when needed with minimal effort.

## CHAPTER 4

### Hardware Configurations for Testing and Implementation

The hardware used consists of two distinct parts — the sensory module and the robotic platform. The sensory module is part of the modeling system and has been designed to be easily ported from platform to platform. Each robot, however, brings its own distinct control and personality into play.

#### 4.1 Initial Testing

The sensor package developed, consisting of the cameras, stereo software, and pipe modeling software, is completely manipulator independent. The pipe modeling package has been tested on four different hardware setups. Two Panasonic GP-MF502 black-and-white CCD cameras compose the heart of the stereo vision system. These are mounted on a two degree-of-freedom “head” (shown in Figure 4.1). A third camera can be used for verification of the stereo model. A Lasaris variable-diameter, infrared, circle-projecting laser was to be used in conjunction with the stereo unit. The laser is also mounted on the head for use as a marker and a structured light source. This laser proved to be lacking in the abilities promised. Simplification of sensor geometry is achieved by forcing the individual units to lie on the same lateral axis while maintaining parallel transverse axes. The laser occupies the center of the lateral axis with the cameras centered around it.

Mounting bracket design includes two degrees of freedom along the lateral baseline

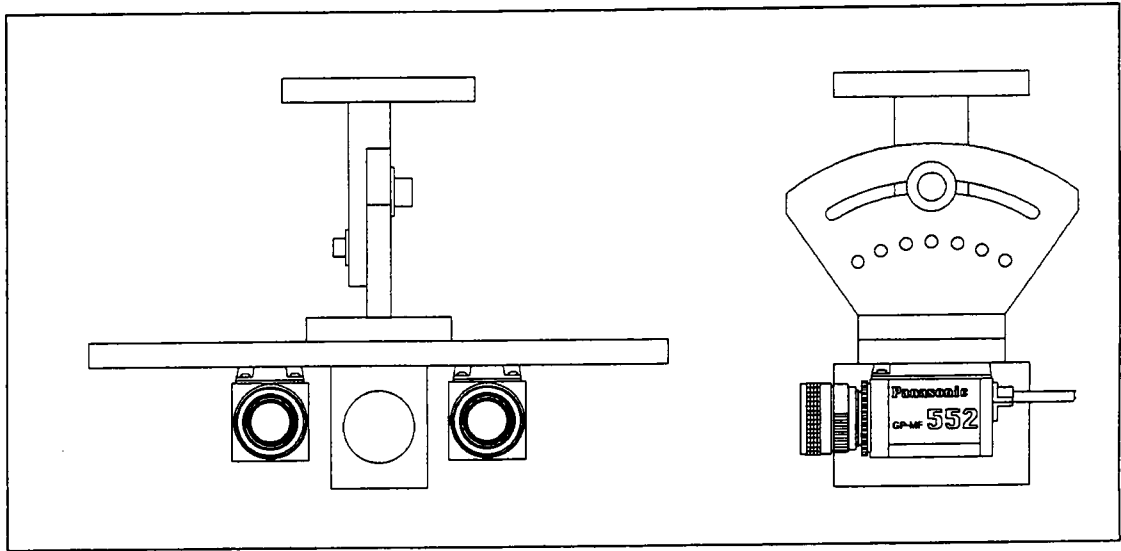


Figure 4.1: *Two degree of freedom stereo vision module to be ported to different robotic platforms for modeling capabilities.*

and pitch on the transverse axes. Figure 4.1 shows the sensory units in one possible arrangement on the mounting bracket. Both degrees of freedom can be manually changed to discrete positions. By restricting these movements to well known locations, accurate calibration of the sensor package becomes a simple matter of describing a transformation between a point in three-dimensional space and a static reference frame on the mounting bracket. Transformations between this reference frame and the individual sensory units become simple translations and camera modeling is simplified.

For initial testing [But93], the package was mounted on the Advanced Servomanipulator (see Figure 4.2), a six degree-of-freedom, remotely maintainable, force-reflecting master/slave teleoperator developed for remote maintenance and handling studies [BC89]. The sensor package is controlled via a VME rack (see Figure 4.3) with two Force 68040 processors. This system uses the VxWorks real time operating system. The master 68040 handles communications, bus arbitration, shared memory access, and

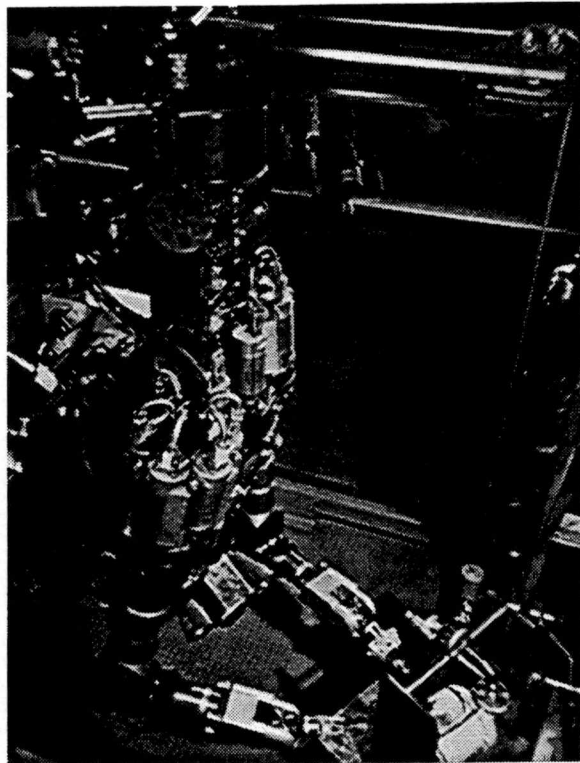


Figure 4.2: *The Advanced Servo Manipulator located at the Robotics and Process Systems Division at the Oak Ridge National Laboratory used in initial testing of the modeling subsystem.*

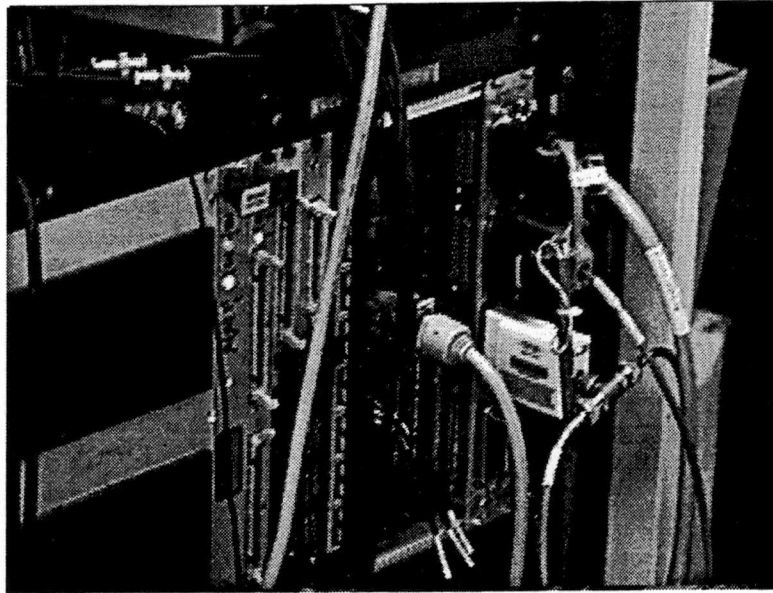


Figure 4.3: *VME rack for real time imaging including processors and image processing boards.*

images acquisition. The remaining 68040 is for use by the system interfacing to the sensor module. For example, this board's serial and parallel ports can control radiation sensors or cut tools. When on the ASM testbed, the slave 68040 is responsible for synchronizing arm control with ASM's MutliBus-based controller. Image capture capabilities are accomplished using Datacube image capture and memory boards. Higher level human-machine interaction (see Chapter 5) occurs on a Silicon Graphics workstation, exploiting the graphical capabilities and builtin VME bus. For this an interface is provided to other systems from the VME rack using a Bit 3 VME-to-VME bus repeater. The Bit 3 board also contains two megabytes of dual-ported shared memory to speed access between the sensor control rack and other VME backplanes. Each user interface uses the X Windows based Motif widget package for the application interface. All code written uses the AT&T 2.0.1 C++ Translator. Figure 4.4 displays a block diagram of

the system hardware configuration.

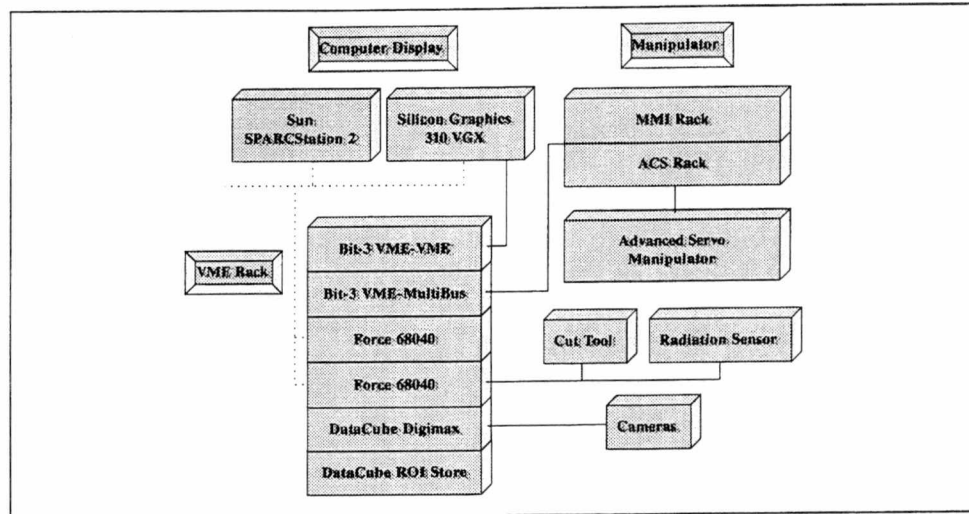


Figure 4.4: *The ASM-based D & D hybrid telerobotic system.*

## 4.2 Second Hardware Configuration

The modeling capabilities were also tested on a standard industrial Cincinnati Milicron T3-726 shown in Figure 4.5. This is the only system in which the manipulation is controlled through the task modeler. All the other systems implemented have separate modules for the manipulator and planning. The control of the Cincinnati Milicron is simple and straight forward. The robot requires a set of six numbers for position and orientation. The kinematics for the motion are embedded in the controller code on the VAX 11/785. This greatly simplifies the task planning. For the manipulation, a file is created from manipulation points chosen using the PMI which contains the positional information for the robot. This file is then fed to the robot and the commands executed.

This setup (see Figure 4.6) was originally designed to show the transferability and

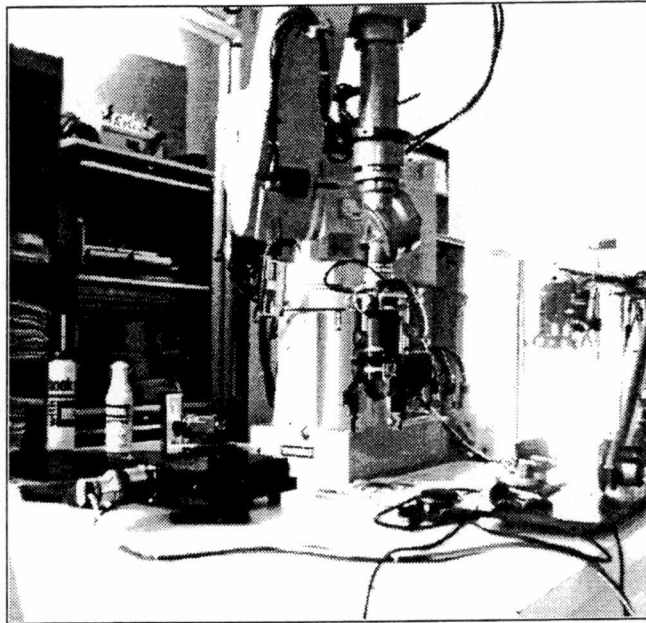


Figure 4.5: *Cincinnati Milicron T3-726 used for testing the portability of the modeling subsystem.*

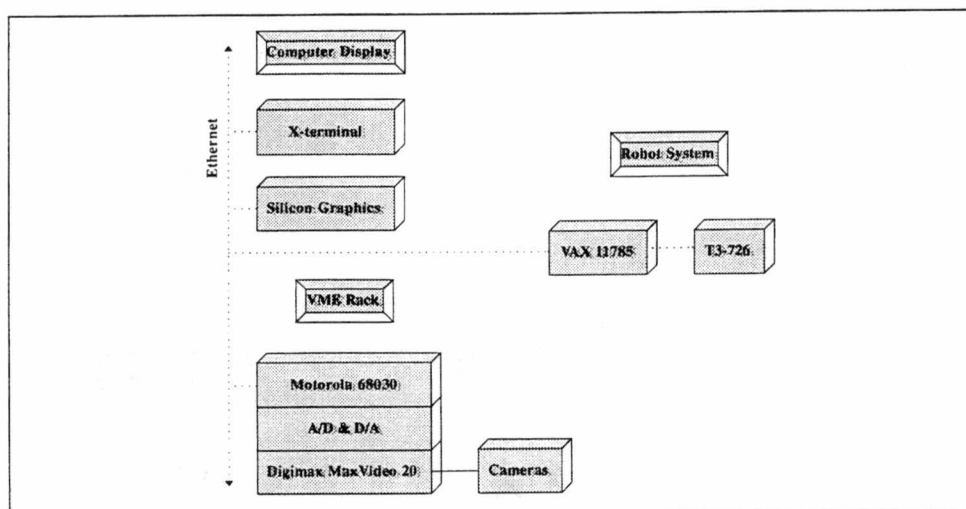


Figure 4.6: *Hardware configuration for use with the T3-726.*

modularity of the vision and modeling system from one platform to another. Therefore, a demonstration was devised in which a ball bearing was picked from the top of a pipe placed at a random position and height and placed into a bin. The pipe on which the bearing is placed is localized with the stereo vision. From this models are created. Scan points and grab points are selected on the models. From these, a set of via points are created and sent to the robot. For these experiments the end effector of the robot serves as a scanner as well as a gripper. Depending on which is selected by the user and sent to the robot position file depends on whether the robot scans the pipe or grips it. When a grab point is selected, the pipe grabbed is automatically placed in either a hot or cold bin depending on the radiation data of the pipe.

### **4.3 Mobile Hardware**

An overall view of this hardware setup is shown in Figure 4.7. This setup represents the movement onto a mobile platform. The mobile platform used for experimentation, known as Experimental Laboratory Vehicle Integrated with Sensors (ELVIS), is shown in Figure 4.8. It is a LabMate robot built by Transitions Research Corporation (TRC) [Tra91a, Tra91b]. This robot is capable of supporting up to 400 pounds. However, increasing the payload affects velocity, acceleration, wheel pressure, and battery life. The manufacturer's maximum recommended speed is one meter per second, although it is capable of speeds up to ten meters per second. For the setup presented here, the heavier payload added by the tetherless VME rack has effected performance (see Chapter 8).

Using the internal robot control boards for servo control, the motion of the robot

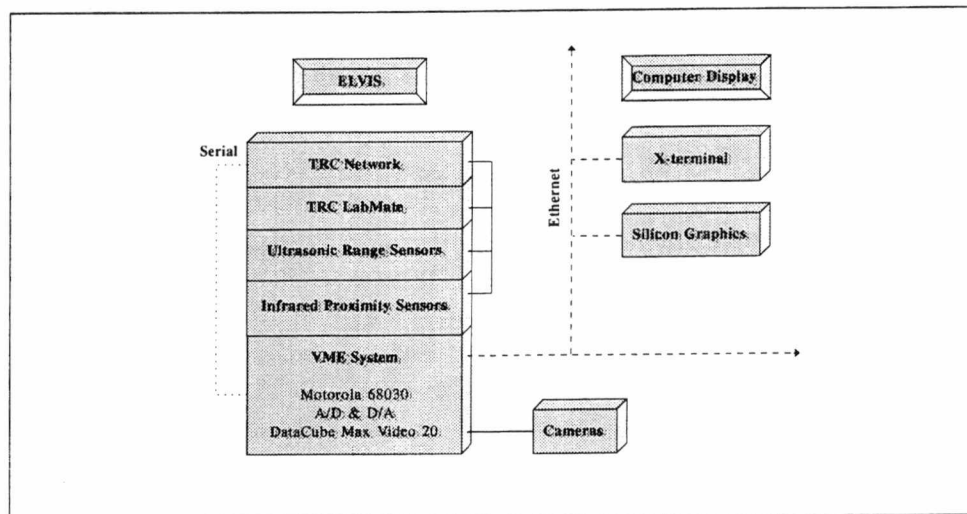


Figure 4.7: *Hardware configuration for the mobile robotic research platform.*

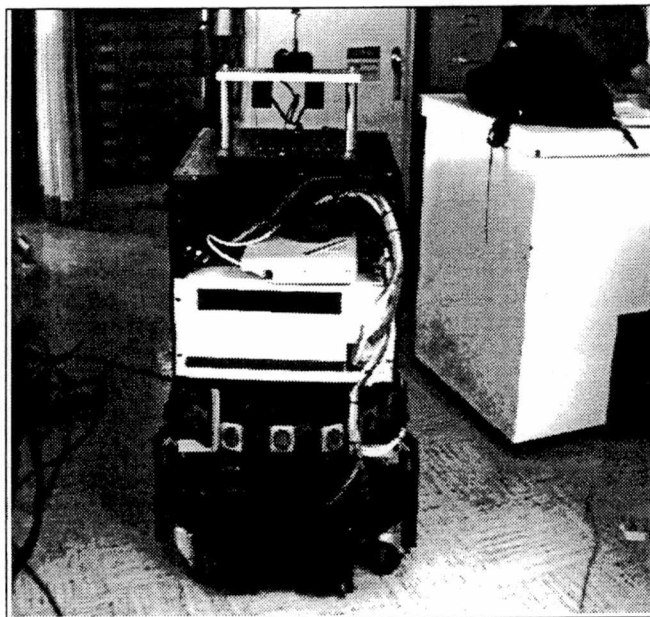


Figure 4.8: *Mobile robotic platform, ELVIS, used for mapping and obstacle avoidance algorithm testing.*

is controlled by setting accelerations, velocities, and turn rates via serial commands. The speed is controlled using the trapezoidal control profile of Hewlett-Packard HCTL-1000 servo motor control chips. These profiles are shown in Figures 4.9 for straight line motion and turns. Torque is available to support up to 0.1g acceleration. Again,

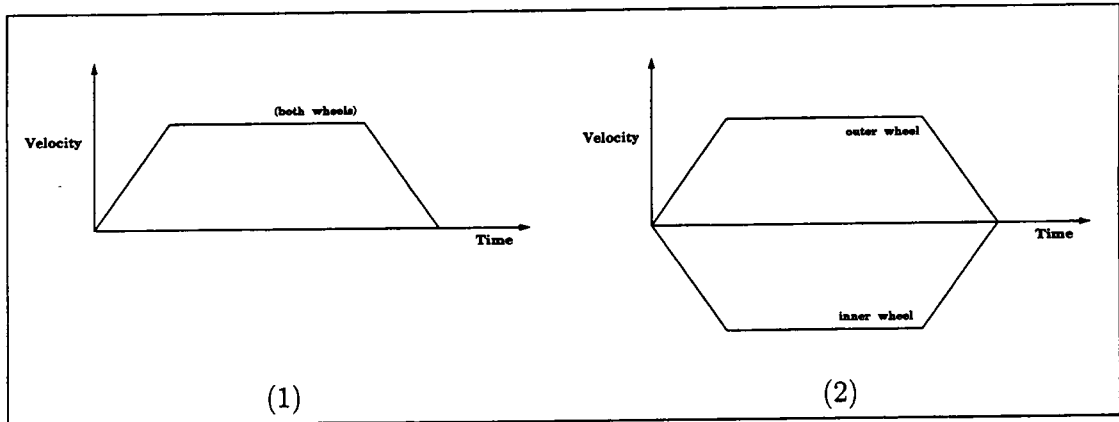


Figure 4.9: *Velocity profile using HP servo chips on board the mobile platform for (1) straight line motion and (2) turning motion.*

this has limited performance of the platform due to the weight. Encoders on the wheel provide a means of measuring position and heading. The control system built into the mobile platform uses these encoders. Wheel slippage causes errors in these readings. This problem is discussed in more detail in Chapter 8. The manufacturer claims typical accuracies of 99%, but does recommend the use of external recalibration after paths of ten meters. The encoder readings are also used for mapping. The current configuration has no means of external calibration, so the readings are considered correct. Many assumptions are made using this platform which are discussed more in Chapter 8 where studies in the positional error are measured.

The sensors used for this configuration are sixteen Polaroid ultrasonic sonar transducers. This subsystem is designed around the Texas Instruments SN28827 Sonar Rang-

ing Module [Tex84]. Communication to this subsystem is also controlled via a serial line which is limited to 9600 baud. The system allows for sensors to be grouped in sets of eight. From each of these sets, only one sensor can be fired at a time. Timeout distance and firing partners can be set, but passing the sensor data limits the speed of the subsystem due to the slow serial link. The results of these limitation are discussed in Chapter 8.

The Polaroid acoustical transducers [Pol84] used are capable of detecting the presents of and distance to objects based on the time of flight method (see Chapter 7 and Figure 4.10). They have an operating range of 0.9 feet to 35 feet (274mm to 10.7

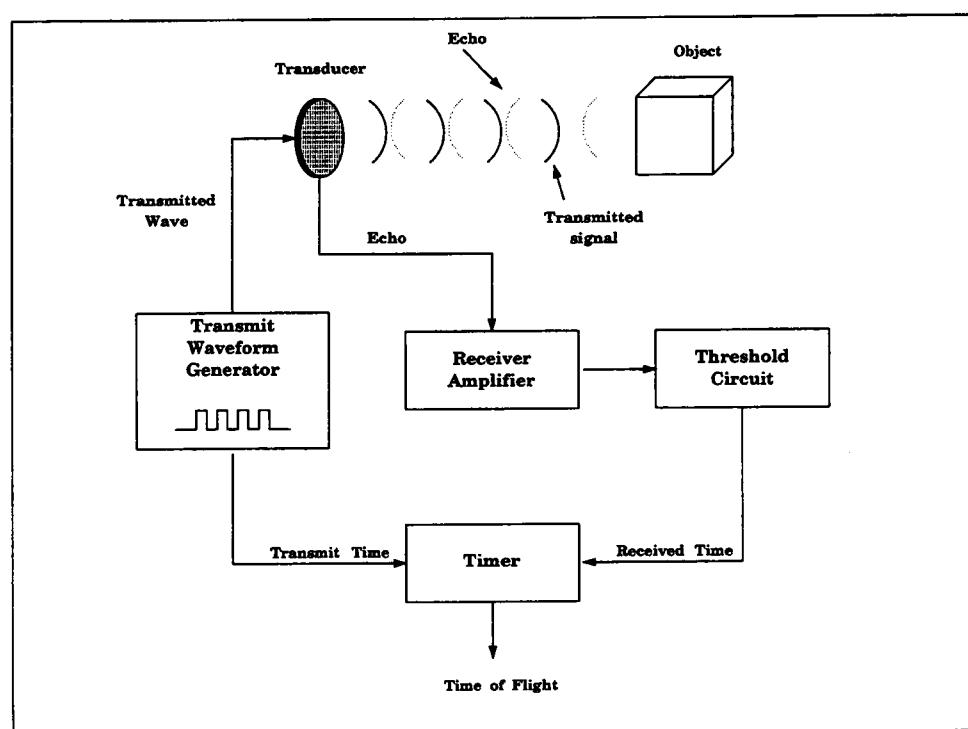


Figure 4.10: *Distance calculations via time of flight sensors.*

meters). To operate, a pulse is transmitted from the transducer and the resulting echo detected. The transducer acts as both transmitter and receiver. The diameter of the

unit determines the acoustical lobe pattern during operation. The transducer emits a high frequency "chirp" lasting about one millisecond. This chirp consist of sixteen pulses. To produce the chirp a 150 volt signal is applied to the transducer after being charged to a DC value of 150 volts. The beam pattern of the transducer is about 30°. (Measured values show 28°.) This pattern can be reduced to about 15° to get higher resolution, but the electronic circuitry used does not allow for this. The transducers are capable of supporting different frequencies, but the TRC board only uses one. Because of the attenuation of the return signal, the echo is amplified. Different circuitry would allow for quicker response times, but that circuitry is not being developed for this thesis.

All of the TRC equipment is controlled through a single serial line using a TRC communication board. The algorithms developed are limited by the response of the transducers and circuitry used. A single infrared proximity sensor is also available. Other hardware, including a DataCube MaxVideo 20 real time image processing board and Panasonic cameras, is also available. The TRC platform is designed to be easy to use. Unfortunately, this leads to a more limited functionality. One such problem limits the speed of the robot. That is the response time to receive information from the sensors is fixed even if the information is available at a higher rate. For this reason, speed had to be scaled accordingly. Specialized hardware would eliminate this problem, but all hardware here is off the shelf readily available equipment. No specialized hardware has been developed.

The user interface resides on the graphics workstation as well as the high-level control for path planning and robot motion. Low level control for the robot and obstacle avoidance is accomplished using a 68030 residing in the VME backplane. The VME rack is currently on board the platform and communication occurs via a wireless ethernet

link. Initial testing was performed with a stationary rack using a wireless serial link. The mobile rack with the wireless ethernet has shown improvements in response time. Higher level task communicate control commands to the 68030, but the low-level task have priority.

## CHAPTER 5

### Workspace Modeling

The Graphical Pipe Model Interface (PMI) consists of a set of graphical tools which allow the operator to edit and manipulate 3D pipe models (see Figure 5.1). The program resides on the Silicon Graphics workstation using the hardware Graphics Library for display of the models. Input to the PMI is taken from the keyboard, from the mouse using a set of push-buttons, or from a six degree of freedom spaceball which allow efficient and effective manipulation of a three dimensional object. The interface receives positional information of the objects from the vision system developed in conjunction with the pipe modeling system [Tha93a] and creates initial models from these parameters.

#### 5.1 Position and Orientation

Borrowing from the teleoperation philosophy, the coordinates of a pipe are interactively determined by the vision system. This information can also be found by different means, such as a range scanner. This tool allows editing of the pipe models and selection of manipulation points along the axis of the pipe. Two points along the axis of least inertia of the pipe are passed to the PMI along with the diameter of the pipe. From this set of coordinates relative to the camera, the pipe is translated and rotated into position as shown in Figure 5.2. From these two points, the location and orientation (six degrees of freedom) of the pipes are calculated [Hor89]. Editing is performed by

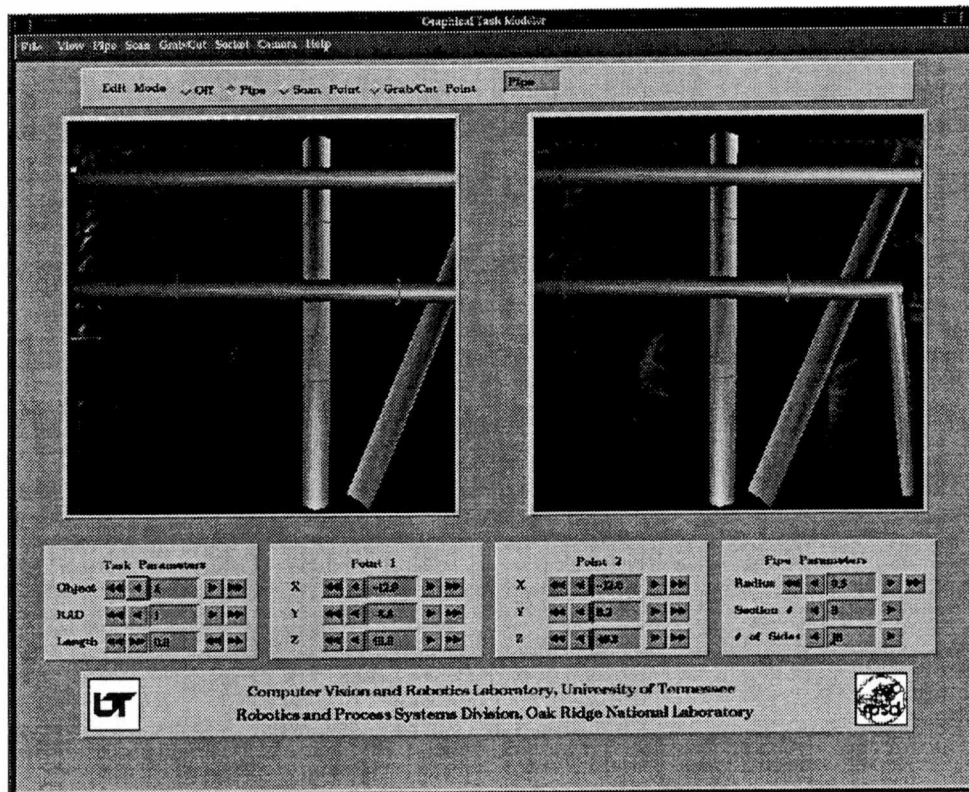


Figure 5.1: *Graphical Task Model Interface used by the operator for manipulation of the pipe model as well as selecting manipulation points along the pipes for radiation scanning or cutting.*

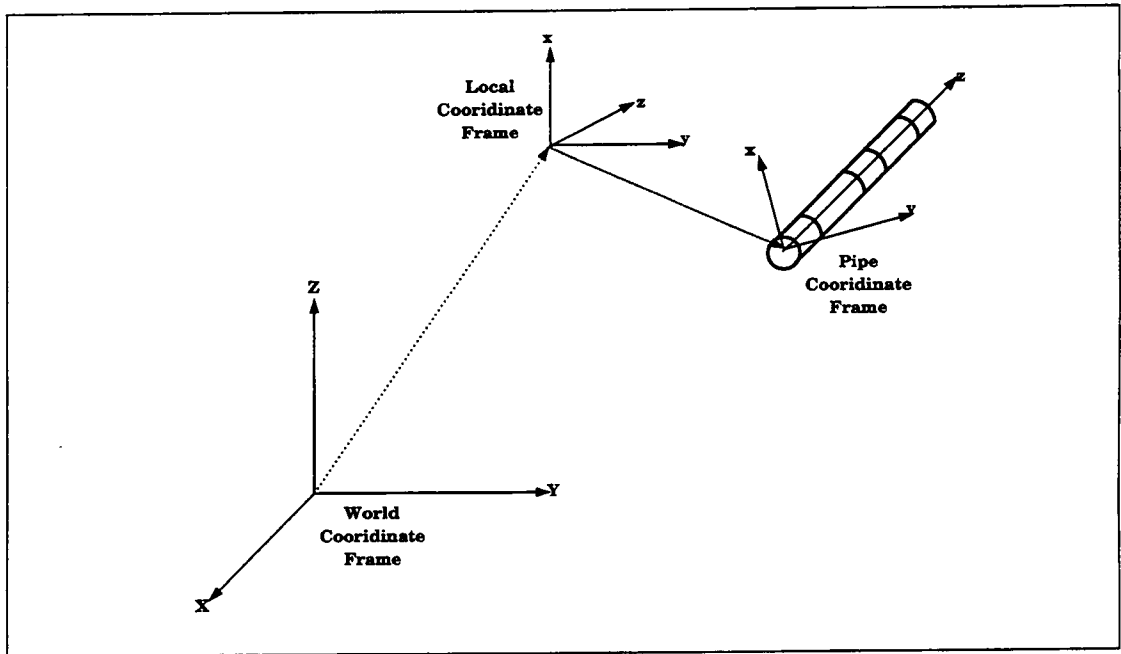


Figure 5.2: *Translation and rotation of pipe model into proper position.*

simply moving the end points of the pipe section. Since the hardware graphics library is being used, the calculations have been optimized to take advantage of the graphics engine. The starting position for the pipe is found by simply translating to the first point given by the vision system. This gives the translation matrix as

$$T_p = \begin{bmatrix} 1 & 0 & 0 & X_p \\ 0 & 1 & 0 & Y_p \\ 0 & 0 & 1 & Z_p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

The SG Graphics Library [Sil89] inherently uses Euler transforms [Yos90] for the rotation matrices used in the video hardware. Therefore a  $Y - X - Z$  Euler rotation was chosen to orient the pipe from its beginning point (see Figure 5.3). The rotation angles for the orientation are found from the difference between the end points of the vector

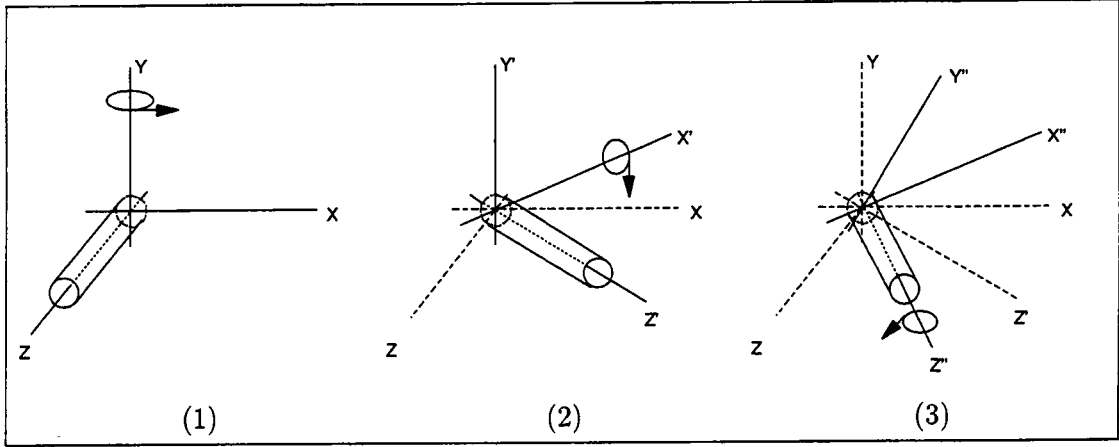


Figure 5.3: (1) Rotation about  $Y$  by  $\theta$ , (2) Rotation about  $X'$  by  $\psi$ , (3) Rotation about  $Z''$

(dark line in Figure 5.4) created from the two points passed to the PMI from the vision system. The equations for the values of  $\theta$  and  $\psi$ , the rotation about the  $Y$  and  $X'$  axis respectively, were derived by back solving from the final position of the pipe to the original coordinate frame [Yos90, Cra89]. Rotation about the  $Y$  axis by  $\theta$  was found using the equation

$$\tan(\theta) = \frac{\Delta x}{\Delta z} \quad (5.2)$$

and rotation about the subsequent  $X'$  axis by  $\psi$  from the equation

$$\tan(\psi) = \frac{-\Delta y}{\sqrt{\Delta x^2 + \Delta z^2}}. \quad (5.3)$$

In these equations  $\Delta x$  is the difference in the  $x$  coordinates of the ends of the pipe and  $\Delta y$  and  $\Delta z$  are defined similarly. From these equations, values are solved for  $\theta$  and  $\psi$  and are as follows:

$$\theta = \text{atan2}(\Delta x, \Delta z) \quad (5.4)$$

$$\psi = -\text{atan2}(\Delta y, \sqrt{\Delta x^2 + \Delta z^2}). \quad (5.5)$$

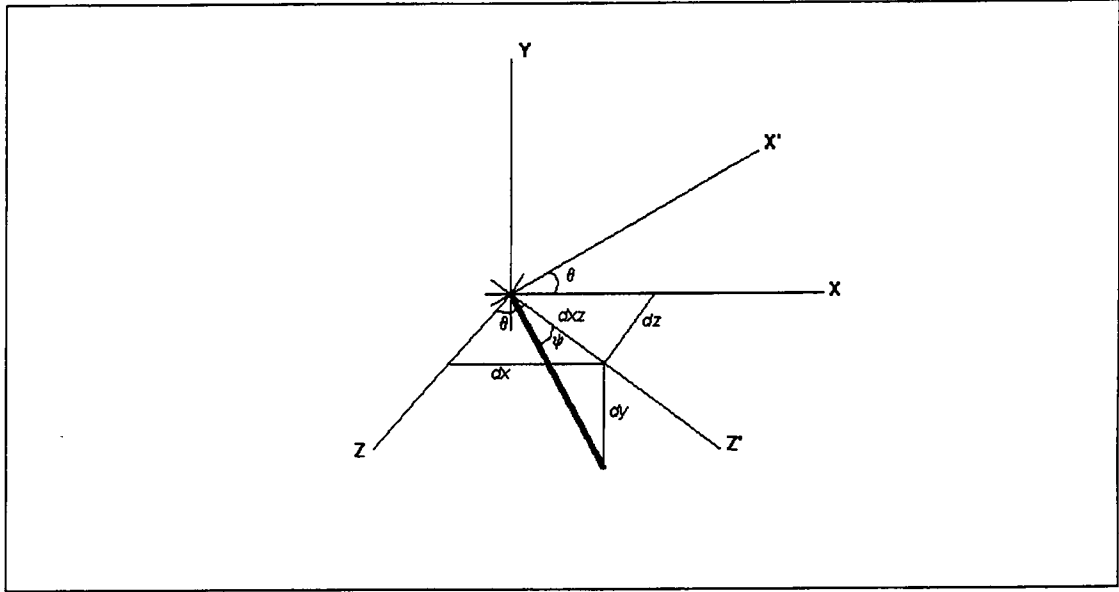


Figure 5.4: *Derivation of rotation angles from the desired position of the pipe.*

The pipe is originally drawn along the  $Z$ -axis; therefore, since a pipe is rotationally invariant along the  $Z$ -axis, no rotation is needed for the last degree of freedom [AK89, GW87]. This results in a rotation matrix given by

$$R_p = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta \cos\psi & \cos\theta \cos\psi & \sin\psi & 0 \\ \sin\theta \sin\psi & -\cos\theta \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

Using these simple equations and the graphics hardware, the pipes can be translated and rotated into the correct orientation very efficiently. The hardware implements these matrices via simple translate and rotate commands needing only the parameters  $X_p$ ,  $Y_p$ , and  $Z_p$  and  $\theta$  and  $\psi$ , respectively. This completes the kinematics needed for proper position and orientation. A set of typical pipes modeled during testing is shown in Figure 5.5.

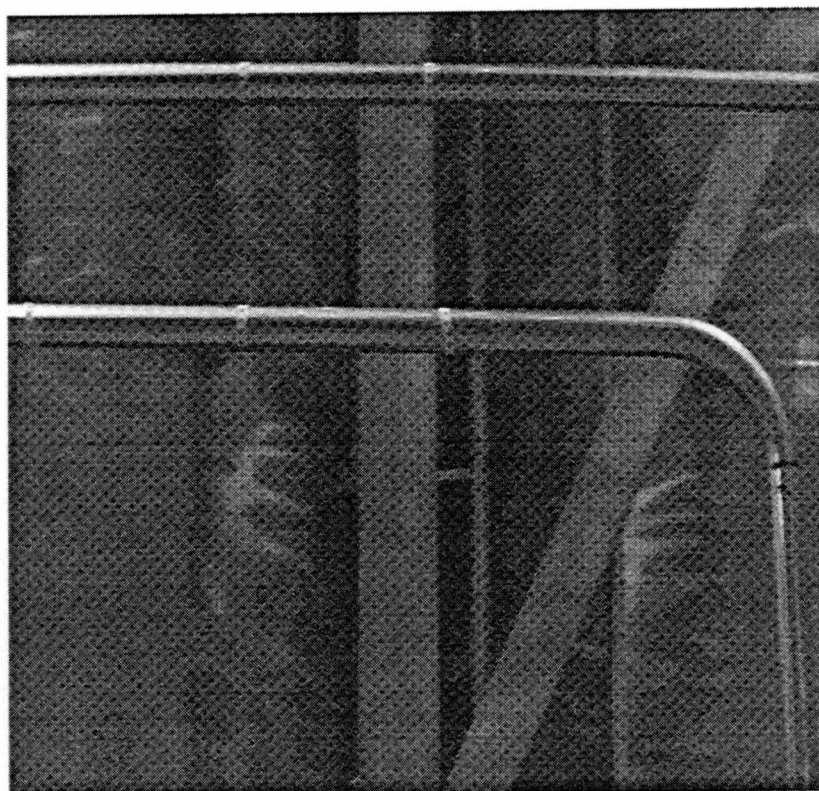


Figure 5.5: *Pipes to be modeled in a typical environment.*

## 5.2 Camera Model

The major problem of overlaying models onto actual images is creating the proper camera models to perform a perspective transform [FGL87, GW87, Fol90]. In this case a pin hole camera model is used (see Figure 5.6). The formation of the complete

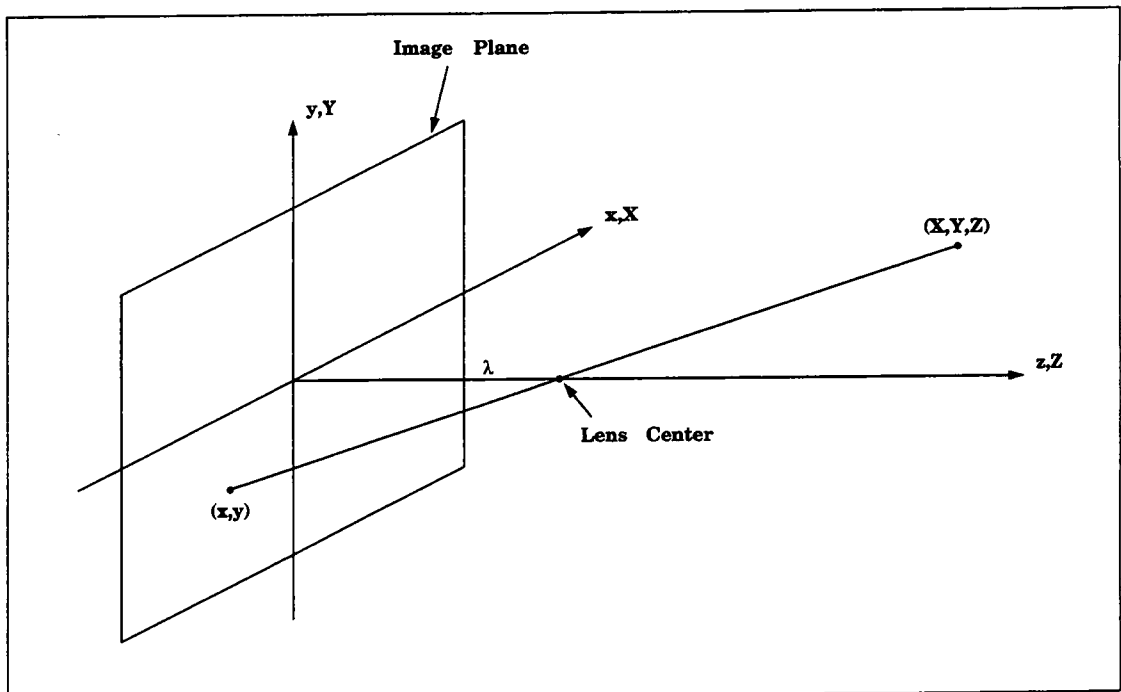


Figure 5.6: *Pinhole camera model.*

camera model involves the calculation of the matrix containing all translation, rotation, and perspective information from the cameras position to that of a reference frame. Gonzalez [FGL87] gives a detailed example of a pinhole camera model, including gimbal offsets and many other specifics. Some of the matrices used in that model are not needed here because of the simplified camera geometry (see Figure 4.1) so a simpler camera model is derived. The perspective transformation matrix, which is the main component

of the camera model, is given by

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix} \quad (5.7)$$

where  $\lambda$  is the focal length of the camera's lens. In this case, the camera's coordinate frame and that of the reference are orthogonal so no rotation matrix is needed between the two. Translation between the coordinate frames and the global reference frame is accomplished using a simple translation matrix

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

In this case values for the offsets are  $X_0 = 2.25$ ,  $Y_0 = -0.5$ , and  $Z_0 = 1.0$  for the left camera. For the right,  $X_0 = -2.25$  and the rest remain the same as the cameras lie on the same transverse axis. A final representation of the camera model,  $c_m$  is then obtained by concatenating these matrices together; therefore,

$$c_m = \mathbf{PT} \quad (5.9)$$

These matrices must now be converted to a format that the graphics hardware can use. The hardware requires the field of view (FOV) from the view point (camera center), the location of the view point ( $X_0$ ,  $Y_0$ , and  $Z_0$ ), and the camera orientation given by a direction. The first parameter calculated was the field of view of the cameras from the focal length of the lens,  $\lambda$ , and the size of the CCD array,  $d$  as shown in Figure 5.7.

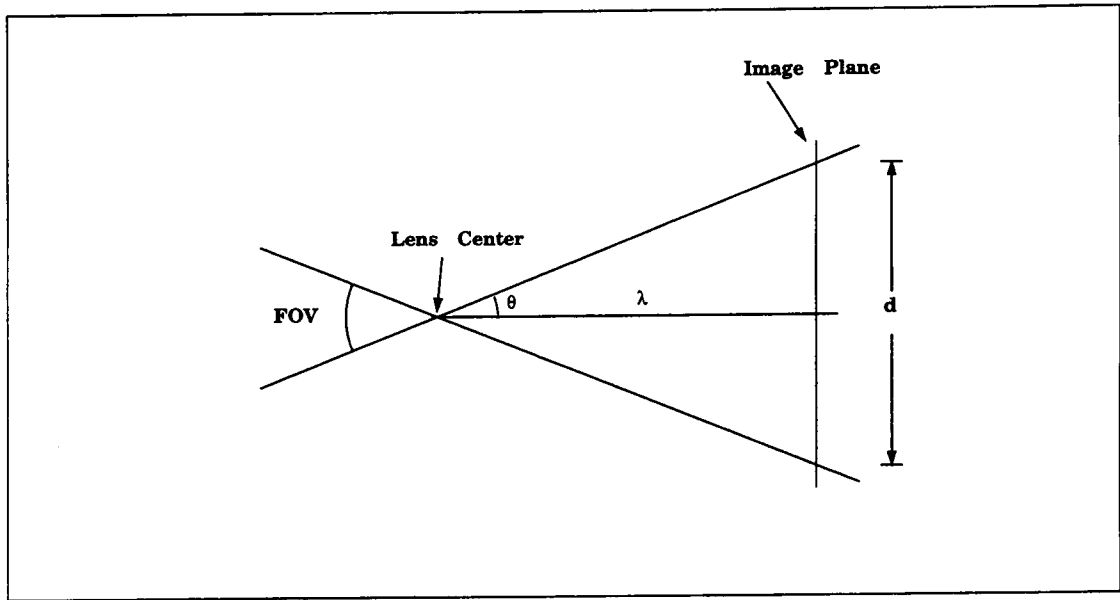


Figure 5.7: *Field of view calculation.*

This is the only parameter that is dependent upon the lens used (in the test case this was 16mm) and the camera. This was calculated by

$$\mathbf{FOV} = 2\theta = \text{atan}\left(\frac{d/2}{\lambda}\right) \quad (5.10)$$

For the test case,  $\lambda = 16\text{mm}$ . and  $d = 6.25\text{mm}$ , so  $\mathbf{FOV} = 22.1^\circ$ . The position of the cameras on the mount determines the  $X_0$ ,  $Y_0$ , and  $Z_0$  offsets and are as given previously. The orientation of the cameras is also determined from the mount because they point directly in the  $Z$  direction with no rotation about the  $X$  and  $Y$  axes. Also, the pixel ratio of the CCD array itself had to be modeled to account for the rectangular ( $13\text{mm} \times 10\text{mm}$ ) elements. This was accomplished by simply adding an aspect ratio

to the transform matrix in the form of

$$\mathbf{A_r} = \begin{bmatrix} 1.3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.11)$$

The resulting matrix containing all the translations, rotations, and transforms is

$$\mathbf{c_m} = \mathbf{PTA_r} = \begin{bmatrix} 1.3 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & \frac{-1}{\lambda} & \frac{-Z_0}{\lambda} + 1 \end{bmatrix} \quad (5.12)$$

From this simple pin hole camera model, 3D object models can be overlain on the image and the operator can manipulate the models with visual verification. Implementation of the above matrices is again greatly eased by the graphics hardware found on the graphics workstations. Projection transformations such as perspective and translations are built into the hardware and are made with a single function call. The PMI allows the operator to change the camera model if needed.

### 5.3 Coordinate Frames

In the PMI there are two coordinate frames which are used. These are the local coordinate frame and the global coordinate frame. The local coordinate frame is that of the sensor array. In the model database both local and relative global coordinate positions are stored. However, the global coordinates are not as precise as the local frame. In initial testing global positioning was based on the position of the gantry on

which the ASM is mounted. The positioning of the gantry is only repeatable within two inches. Hence, the operator must recalibrate the arms before a scan or cut is performed (see Chapter 6). For detailed work, the operator must use the local coordinate frame. For the demonstration with the T3-726, calibration with respect to a fixed global reference was performed with a meter stick. This method, though very crude, performed reliably every time during demonstrations. The dual-arm setup is very well calibrated but not yet tested. The global positioning is integrated into the modeling system for use in mapping with the mobile system. In this case the global coordinates are taken from wheel encoder readings. More about the positioning problems will be discussed in Chapter 8 regarding the experimentation with the mobile platform, but much is not done to try to solve this huge problem in the experimentations performed.

#### **5.4 Error Correction and Knowledge Enhancement**

Due to the limited view of the stereo cameras, a single pipe may be seen from different views as two or more different pipes. The PMI can be used to merge the pipe back into one. Also, if a pipe extends out of the view of the cameras, which is very common, the PMI can be used to lengthen the pipe. If a bend occurs in the pipe and is only visible in one of the stereo images, the operator can use his knowledge gained from the vision system as to the  $z$ -depth and diameter of the pipe that is visible in both images and using the PMI, add the bend in the pipe. An example of this is shown in the lower right corner of the right image in Figure 5.1. Also provided is the ability to color code the pipes (see Figure 5.8). This aids the operator in that objects can be quickly and easily grouped using color cues. With the ASM, the color coding was used

for radiation coding (see Chapter 6).

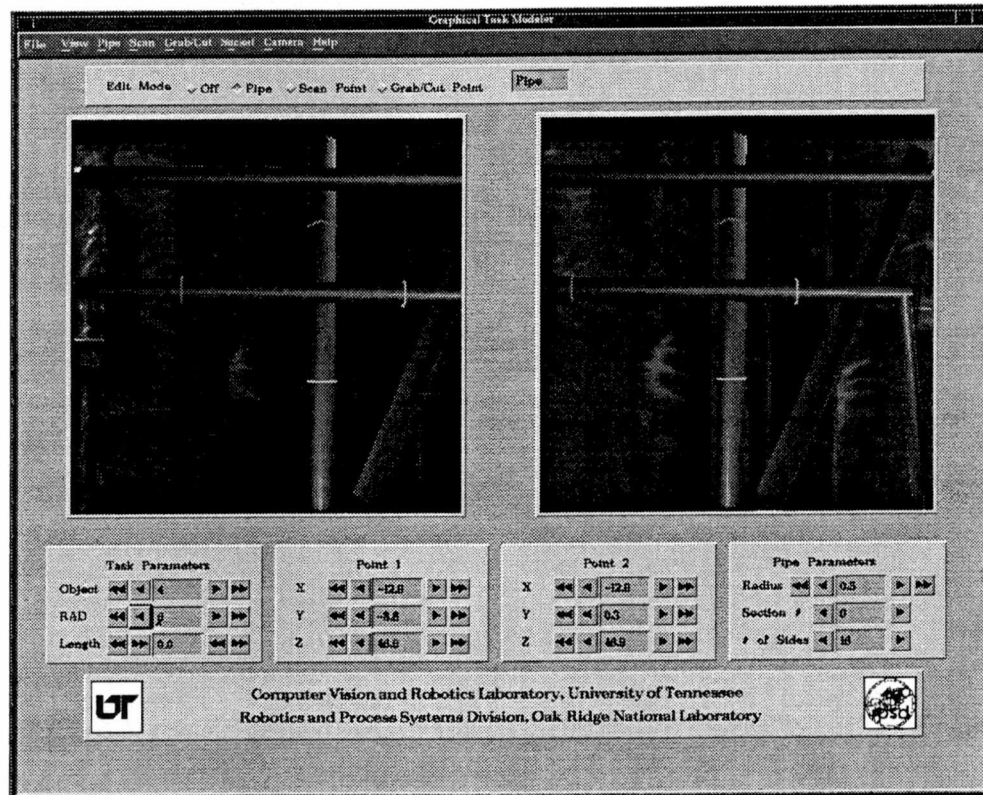


Figure 5.8: *Graphical Task Model Interface with color coded pipes overlain onto the actual images of the environment to ease the operators ability to group pipes.*

The PMI allows for reading, writing, and appending as many model databases as needed. Since the system may be used over several days at a time or at different locations, a means of storing the data obtained is needed. The format of the file is ASCII so the operator can read the task model as well as place comments in the file. This database can also be used with other programs such as previewers. The models created have been passed into IGRIP for use in manipulator studies of the dual-arm system. The PMI also allows the operator of chose scan points as well as cut and grab points along the axis of the pipe. These points can be used to store positions that the

robot can use for manipulation points.

Also the operator has the capability to view the images and models in a variety of different manners. For example, either wire frame or solid models can be viewed. For stereo viewing, a red/green display is available for use with stereo glasses. This gives the operator the perception of actual 3D depth.

## CHAPTER 6

### Task Planning and Graphical Preview

A model of the environment is easily built by the operator using the interactive vision and pipe modeling systems. To aid an operator using the overall system, the PMI also incorporates the ability to select points on the object models. This allows for many possible scenarios. These points can be past to the manipulator system to create path planning. This can be used to create a "scan plan" for radiation scanning of the environment and radiation encoding of the models. This is accomplished by the operator choosing beginning and ending scan points on the graphical task model. The information is then passed to the manipulator, in this case the ASM. The ASM will then be placed near the area to be scanned, calibrated using the vision system, and robotically scan the pipe using the radiation tool. The information gained from the radiation scan will be added to the model database. The graphical model of the ASM for use in a virtual mode is shown in Figure 6.1 along with the actual ASM.

In a similar manor, the PMI can be used to select cut and grab points along the axis of the pipe. Therefore, robotic cutting of the pipes takes place instead of teleoperated cutting, and, since the radiation data is stored in the model database, the pipes can be deposited into the proper hot or cold bin for disposal depending upon the color coding. This will serve to cut down on the amount of hazardous waste that must be decontaminated.

The databases created by the Task Modeler can be incorporated into other programs

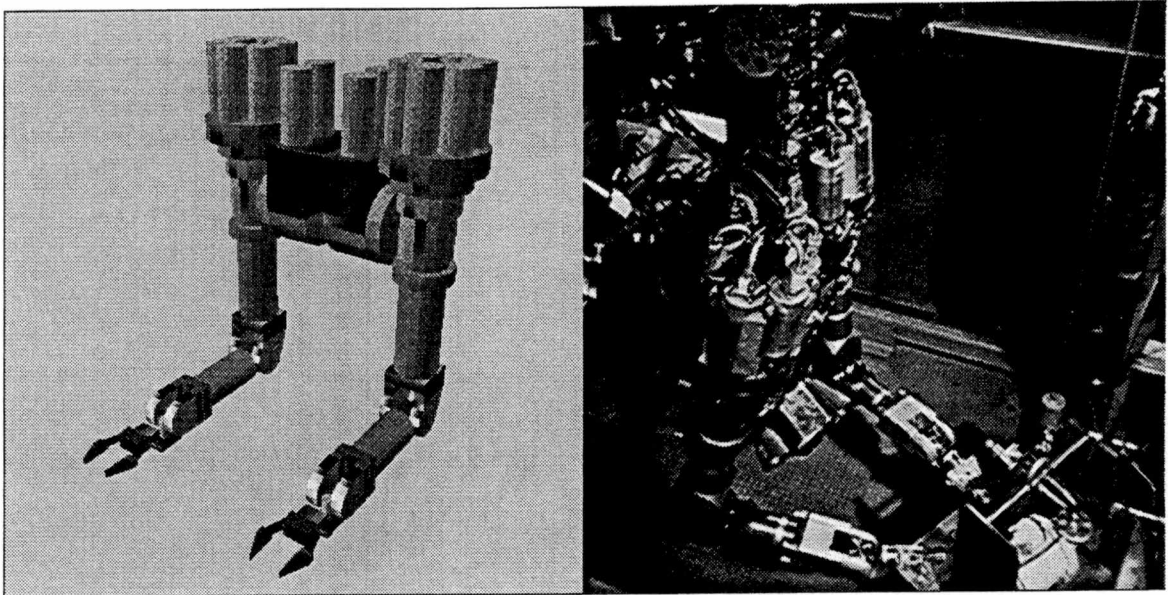


Figure 6.1: *Graphical model of ASM used in the virtual environment along with real ASM used in initial testing.*

such as graphical previewers and IGRIP. A graphical model of the ASM is available for the operator to use in a virtual mode (see Figure 6.2). A one-to-one joint position is available between the model and actual ASM. This allows the model to be controlled from the master. This allows the operator the ability to view the motion he plans to carry out in a virtual world using the same controller as in the real world. He can then tell if the motion is possible. He may find that a pipe will not allow him to make the move or that it is not possible for the move he had planned to make to be done. This allows for safer use of the manipulator. Also, using this mode, task planning can be carried out in a teach/playback mode. Here, the operator can perform a task which may need to be done many times, record (teach) this task, and then allow robotic playback leaving the operator free of repetitious work.

Each joint is individually controllable via sliders on the user interface. A closer view

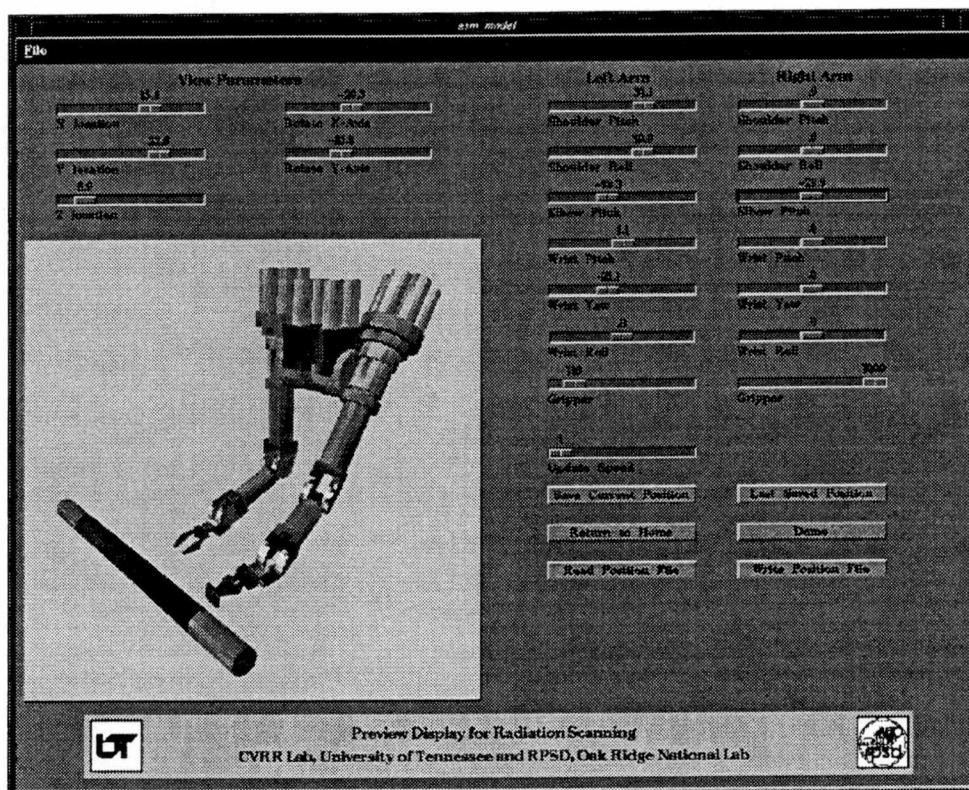


Figure 6.2: Virtual ASM Interface in which the operator can plan, preview, and teach.

of the controls is shown in Figure 6.3. The viewing position in the virtual world is also

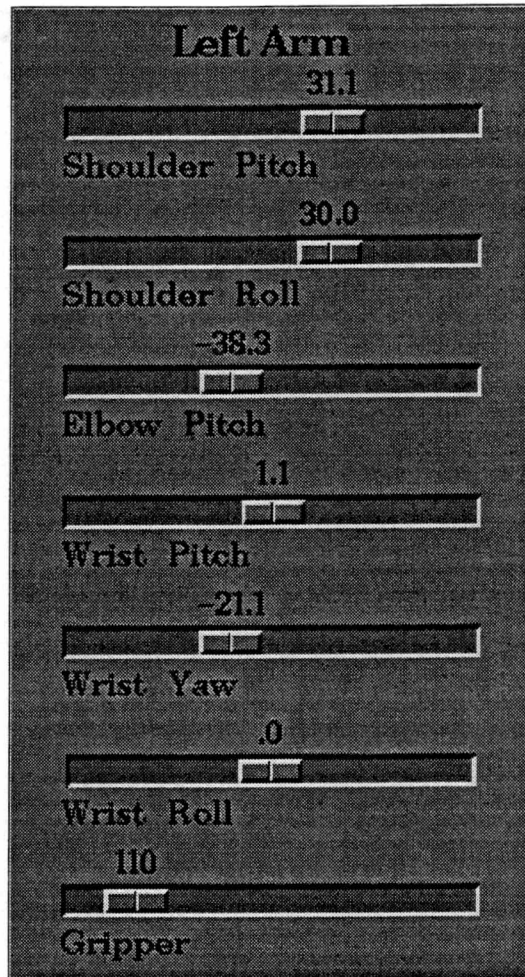


Figure 6.3: *Individual joint control for the ASM Interface.*

controllable via a set of sliders that are able to change the X, Y, and Z position of the view as well as rotate the view about the X and Y axis (see Figure 6.4). This allows the operator to view the environment from positions not normally or easily available in the real world giving him more flexibility in planning and preview (see Figure 6.5).

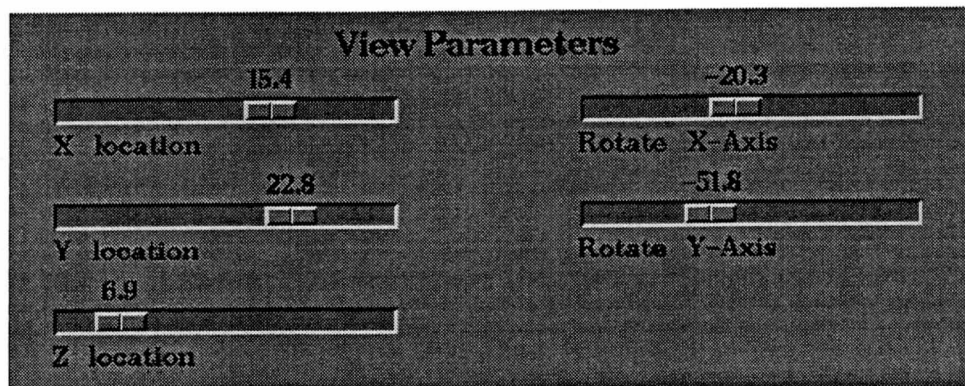


Figure 6.4: *Viewing parameter control for the virtual ASM interface.*

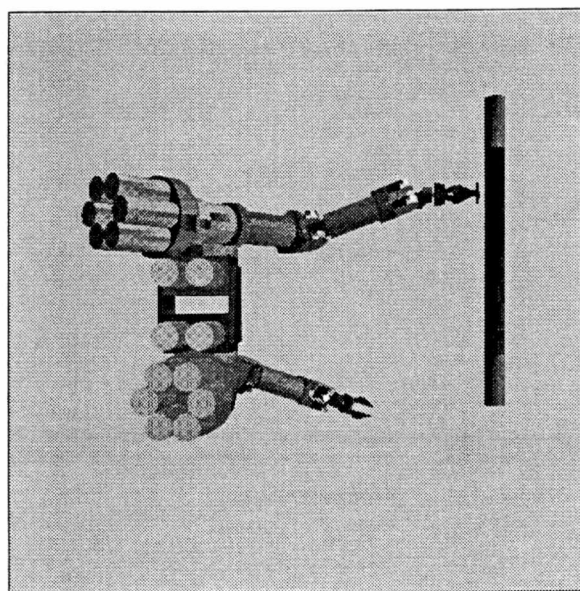


Figure 6.5: *Overhead view of the Graphical model of ASM used in the virtual environment not normally viewed with real ASM.*

## CHAPTER 7

### Mobile Robotic Subsystem

For movement of the system onto a mobile platform the objective again is to create a subsystem with a very simple modular high-level interface for the user [Gou94, GT94]. The lower-level task are to be transparent. The obstacle avoidance is considered to be a low-level task, like a human reflex. The user of the system should never have to worry about this reflex, just the high-level tasks. To assist in this and for ease in integration onto other platforms the overall subsystem itself is keep as modular as possible. They are as follows:

- User interface,
- Communications,
- Robot motion,
- Sensor reading,
- Obstacle avoidance, and
- Path planning.

The user interface and path planning are the only higher-level modules with which the user interacts, the rest are lower-level tasks that the user can take for granted. Only the robot motion module is manipulator dependent. All others are designed to be ported to other platforms with a minimum of effort using the philosophy presented earlier.

## 7.1 Obstacle Avoidance Algorithm

To avoid obstacles, they must first be sensed. Ultrasonic sensors (described in Chapter 4) give a fairly simple, quick, and cheap means of measuring distance to an object. The details of the object are not needed only position. To use a sensor, one needs to understand how the sensor works. A range sensor measures the distance from a reference point to an object that is in the field of view of the sensor [FGL87]. This project uses ultrasonic range sensors utilizing the “time of flight” concept (see Figure 4.10). In this the distance is measured based on the time elapsed between the transmitted signal and the returned echo. For obstacle avoidance, the shape characteristics of the objects in the environment are not needed, merely the distance to the objects. Since the speed of sound is known for transmission through air, a simple calculation involving the transmit and receive times of the signal and its echo gives a distance to the reflecting surface. Ultrasonic sensors are inherently noisy and have a low resolution. For this reason, they have primarily been used for proximity sensing for tasks such as navigation and obstacle avoidance. The operation of an ultrasonic sensor can be described by the use of Figure 7.1 from [FGL87]. The first waveform, *A*, is the gating signal that is used to clock the transmitted signal. This gives rise to waveform *B* which is the transmitted signal and the resulting echo. The pulses shown in *C* result from either a transmitted or received signal. To determine if the signal is outgoing or incoming a time window *D* is introduced. This gives a minimum and maximum detection time,  $\Delta t_1$  and  $\Delta t_1 + \Delta t_2$ . An echo received while *D* is high produces the signal *E* which is the received echo. This signal is reset at the end of the gating signal *A*. To use the sensor purely as a proximity sensor and not a range sensor, the signal *F* can be set with *E* and reset on the rising

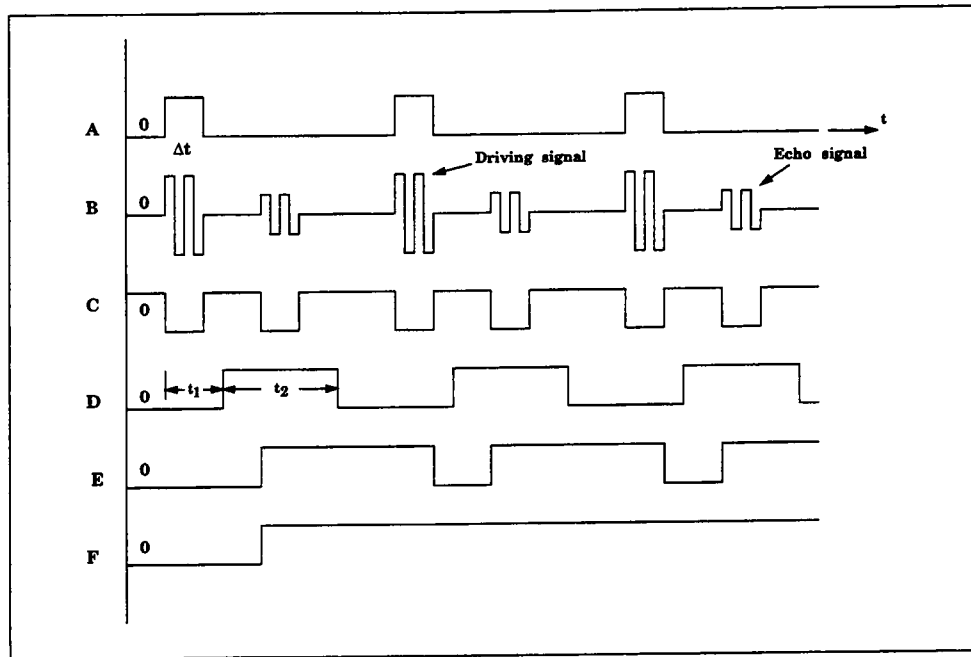


Figure 7.1: *Waveforms associated with ultrasonic range sensors.*

edge of  $A$  when  $E$  is low. This will give a purely binary output from the sensor when used for a proximity sensor instead of ranging.

Many different configurations have been devised for the use of ultrasonic sensors for gather range data. This paper does not attempt to use all of these methods due to limited hardware configurations described earlier. The algorithms however still apply no matter how the data is obtained. Quicker methods of acquisition do lead to higher speeds in the robot due to the simple fact that the data is available sooner. This thesis does not attempt to introduce new methods acquisition of ultrasonic range data, merely to use the data for tasks of mapping and obstacle avoidance. The ideal setup would contain multiple frequency sensors with matched correlation chips to distinguish signals for multiple firing. Also, by firing opposite sensors simultaneously, eight sensors could be fired increasing the response time by a factor of eight. Special interface hardware could

also speed the 9600 baud transfer speeds. This does require specialized hardware which is not currently readily available. Only the first echo received is taken into account by the hardware, though the sensor is capable of taking multiple readings with a depth resolution of about 3 in. due to the length of the signal burst transmitted.

There are sixteen ultrasonic range sensors arranged around the top of the robot. As with most ultrasonic range sensors, they are noisy. For the mapping this is taken into account by taking multiple readings and forming occupancy grids. For the obstacle avoidance, however, there is not time to take many readings, so it is assumed that one is sufficient. Reading the sensors is fairly simple through the communications software. The timeout distance and update speed are user selectable. However, there is a fixed time delay with the current hardware due to the 9600 baud serial link used for data transfer.

#### **7.1.1 Obstacle Avoidance**

Real-time obstacle avoidance presents the problem of navigating around unknown objects in a dynamic environment. The virtual force field (VFF) method described by [BK89] uses an occupancy grid to determine the position of obstacles in the environment. A VFF is then calculated from these obstacles and the robot is "pushed" away from them. For this implementation the occupancy grid probability profile used for mapping is shown in Figure 7.2. The obstacle avoidance algorithm implemented here is similar to this concept but varies in that it uses only current sensor readings to calculate the VFF instead of using occupancy grids. The algorithm takes into account immediate proximity as to whether to react to an object. With the robot being used for experimentation, the reliability of the positioning is very low (see Chapter 8). Slippage,

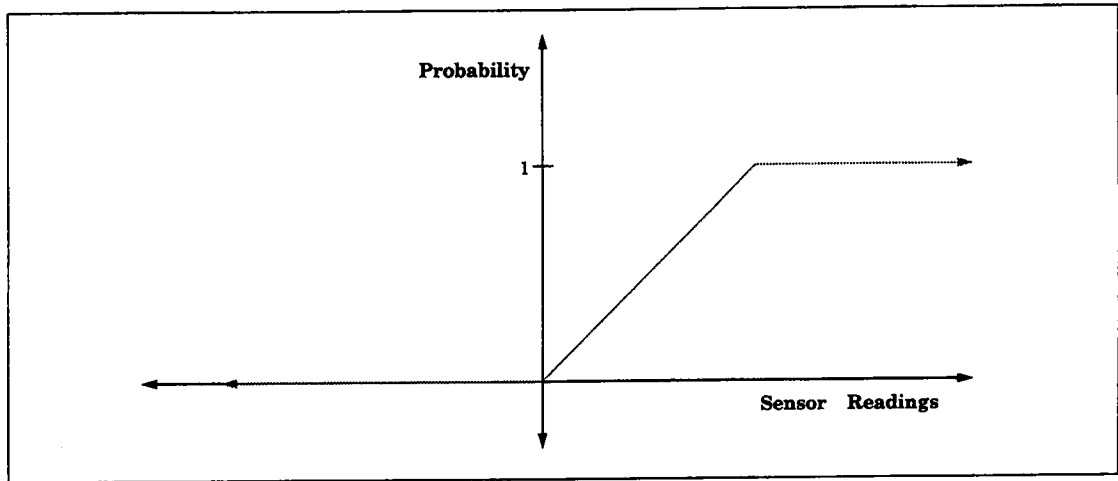


Figure 7.2: *Occupancy probability model used in creating occupancy grids from ultrasonic range data.*

accumulated error, and taking the robot to different floors makes the use of occupancy grids to determine VFF's difficult. Borenstein [BK90] has developed specialized hardware and algorithms to reduce error in sensor readings by taking multiple readings, but does not take wheel slippage into account. Positional error is accounted for by updating the occupancy grids. This is alright for local positioning but for an overall global system presents errors. This does take more computational time, and since this module of the project is sharing computer resource with much more complicated modules, the amount of computational time used needed to be as low as possible. Therefore, a robust, reactive algorithm has been developed for obstacle avoidance which does not rely on occupancy grids. This algorithm allows for a much quicker reaction time to objects which quickly appear in the environment. The current working algorithm is as follows.

1. Initialize system. Set update speed and timeout distances for sensors.
2. If obstacle avoidance is on, read sensors; else, end.

3. If sensor reading is under lowest threshold stop, calculate new heading, turn, proceed at lowest speed. Goto 2.
4. If reading is under second threshold, calculate new heading, veer, and proceed at normal speed. Goto 2.
5. Proceed at highest speed. Goto 2.

Each sensor has two thresholds assigned to it creating two rings around the robot (see Figure 7.3). Since the robot moves forward, the front sensors have a higher threshold

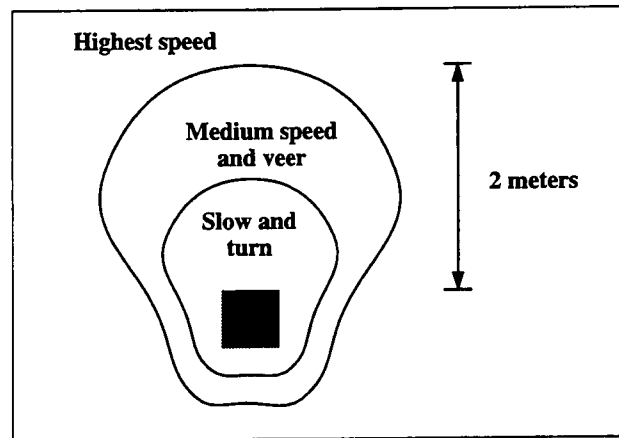


Figure 7.3: *Thresholds for variable speed and reaction determined from the range data of the ultrasonic sensors positioned around the robot.*

and the rear have the lowest. The two sets of thresholds allow for variable speed settings by having two levels of impending obstacles. Most other mobile platforms performing obstacle avoidance only move at one speed [Con92]. Table 7.1 gives the threshold values that are associated with each of the sensors on the robot. When none of the sensor have readings that are within the highest threshold, no obstacles are near and the robot moves at its highest speed (currently 700 mm/sec). When there is a reading inside the

Table 7.1: *Distance thresholds associated with each sensor to create two rings around the robot for use in variable speed obstacle avoidance.*

| Sensor      | Inner threshold | Outer threshold |
|-------------|-----------------|-----------------|
| 0 (left)    | 300             | 600             |
| 1           | 500             | 1000            |
| 2           | 850             | 1500            |
| 3           | 1000            | 1750            |
| 4 (forward) | 1000            | 2000            |
| 5           | 1000            | 1750            |
| 6           | 850             | 1500            |
| 7           | 500             | 1000            |
| 8 (right)   | 300             | 600             |
| 9           | 250             | 500             |
| 10          | 250             | 500             |
| 11          | 250             | 500             |
| 12 (back)   | 250             | 500             |
| 13          | 250             | 500             |
| 14          | 250             | 500             |
| 15          | 250             | 500             |

highest threshold the robot slows (350 mm/sec) to accommodate a possible obstacle and veers to try to miss it. If reading occurs within the lowest threshold the speed is set to its lowest (150 mm/sec) and a new heading vector is calculated. This heading is based on a scaled inverse of the readings from each sensor (see Figure 7.4).

$$\text{Heading} = \text{angle}\left(\sum_{i=0}^{16} w_i s_i\right) + \text{Desired} \quad (7.1)$$

where  $w_i$  is the weight associated with each sensor and  $s_i$  is the inverse of the vector reading of the sensor. By inverting the reading closer objects will apply larger forces.

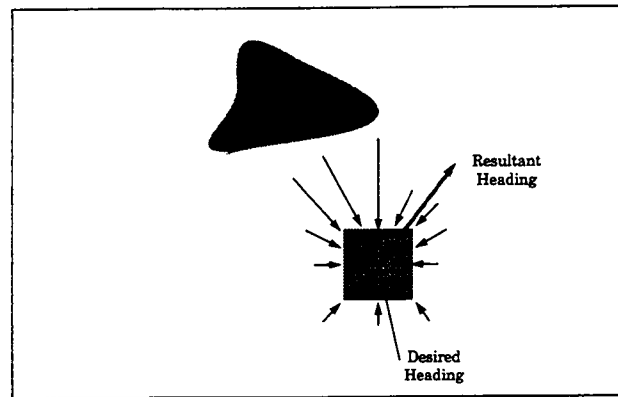


Figure 7.4: *Sensor vector summation of range readings added with the desired direction to form an overall direction to avoid obstacles while proceeding toward the goal.*

In doing this, each sensor gives a weighted vector directed toward the robot. Closer objects give rise to larger magnitude vectors. A resultant is calculated and added with a vector in the direction of the motion desired. The angle of this vector determines the new heading. The maximum angle allowed is 45 degrees and the minimum 5 degrees to try to damp oscillations.

This algorithm makes the obstacle avoidance completely reactive in that no previous knowledge of the environment is needed. With a sensor update rate of four hertz, it is a low-level reflex relying on no previous knowledge of the environment. Using this alone

the robot can “roam” while creating a map of the environment. Obstacle avoidance is no problem if accurate positioning is not known.

### 7.1.2 Path Planning Strategy

Known obstacles are taken into account in path planning (see Figure 7.5). The path planning is based on the original occupancy grid either created earlier or given. Path

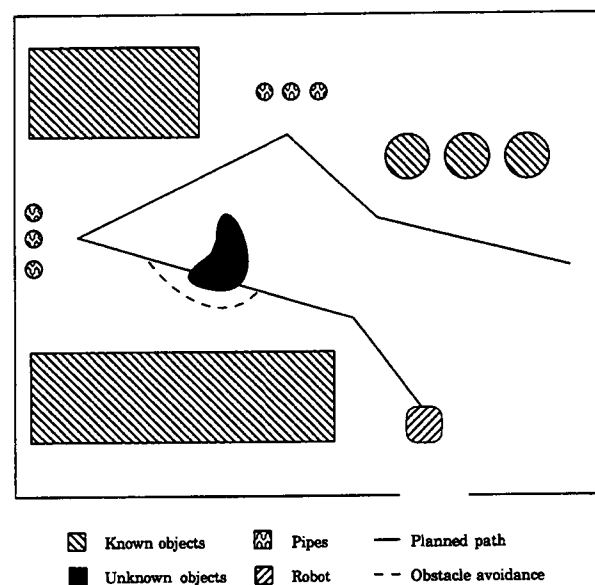


Figure 7.5: *Results of a typical experiment in path planning in which the operator has chosen via points and the obstacle avoidance must take over to avoid an unknown object.*

planning is accomplished in three ways. The simplest path planning is to plot a straight line from the start point to the goal. The obstacle avoidance algorithm can then do the rest. As the robot encounters unknown objects in the environment, the obstacle avoidance algorithm will take over. After the robot has steered clear of any danger, it will again try to proceed straight toward the goal.

Another possibility is based on the philosophy of the potential field method. Using

the potential method concept to produce the least potential path between to point, simple image processing techniques are used to produce such a path on an occupancy grid. The occupancy grid will be binarized with all cells known to be occupied (probability greater than 0.7) and those which are still questionable (probability between 0.3 and 0.7) will be one level (background or 0) and the cells known and those which are not occupied will be the other (foreground or 1). Since object appear as background and unoccupied space appears as the foreground, the image will first be eroded to grow the occupied areas to allow a margin of area to account for the size of the robot. Two points in the image on the foreground will then be chosen, the robot's current position and the desired position. Now, a thinning or skeletonizing technique is used leaving the two points connected. This will produces a graph from which the smallest path can be chosen. The line connecting the two points should be the least potential path between them. All known obstacles will be avoided. This technique is not yet fully implemented due to positioning errors in the robot, but does prove to be promising. as large as possible into the empty space. The boundaries along the regions will then also be the same graph as mentioned earlier and the same results will be obtained. The last path planning method is based on the philosophy of teleoperation. This method allows the user to manually layout a path for the robot to follow on the user interface. The path planning really suffers from the lack of positional accuracy. A path that is over ten meters can essentially be rendered useless by accumulation of error if the accelerations and velocities are set too high. Arriving at the actual desired point is rare in these situations.

Most of the prevalent path planning technique now being used are based on graph theory techniques [TG74]. Some of these were discussed in Chapter 2. They basi-

cally consist of building nodes of possible intersection where the edges connecting them separate obstacles in the environment.

## **7.2 Implementation and System Configuration**

### **7.2.1 User Interface**

The graphical user interface allows control of higher-level functions in the project, such as path planning, while the low-level control remains transparent although still controllable, such as the timeout distance on the sensors. The interface (shown in Figure 7.6), was created using GL/Motif widgets to run on a graphics workstation and also, without graphics, on any X-terminal. The interface gives the user the ability to change the robot speed, sensor timeout distance, and sensor update speed as well as teleoperated and teleautonomous control of the robot. A real-time visual display of the range sensor data is provided along with controls to set the high-level sensor update rate. An environmental map in the form of the latest occupancy grid with the present position of the robot and the path overlaid. The view of the cameras on the robot can be shown, but this is not presently real-time since the video update rate is so slow over the wireless ethernet connection. A software joystick is available for the user to control the motion of the robot when not autonomously moving (see Section 7.2.3). An information window supplies the user with the status of the system. Path planning via points can be selected by the user by simply placing points on the map of the occupancy grid (see Section 7.1.2). From the user interface the occupancy grid can be saved and previously constructed grids can be loaded.

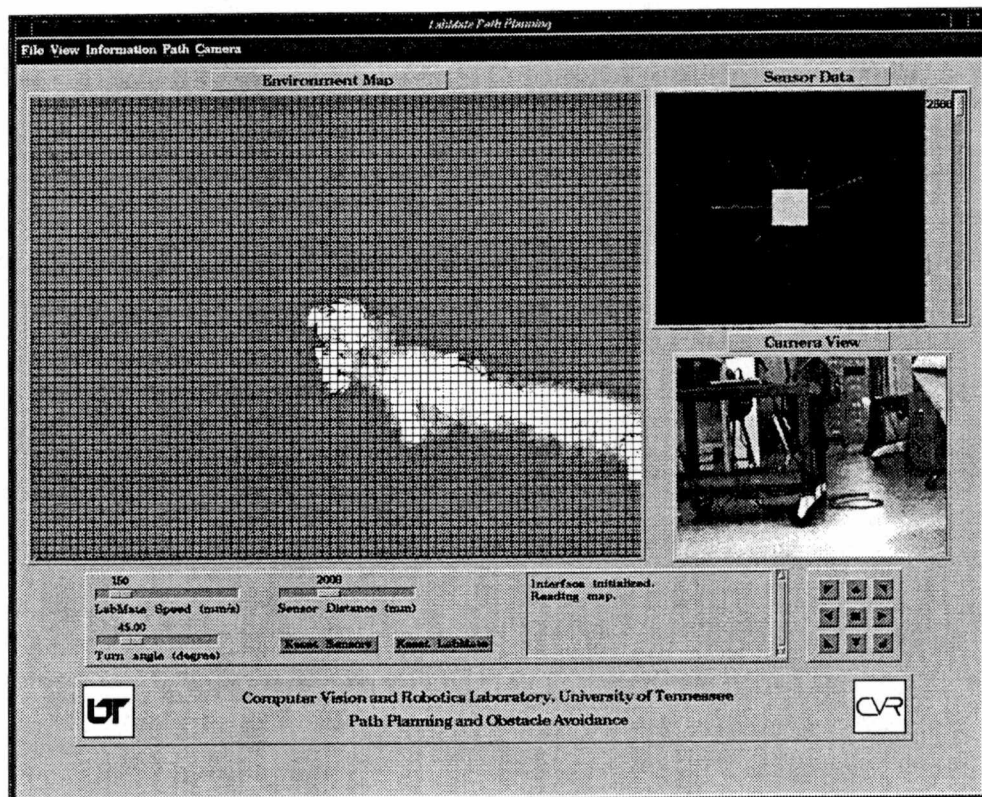


Figure 7.6: Graphical user interface for the mobile robot including an up-to-date occupancy grid, current sensor readings, recent camera view, robot control, and vital information.

### **7.2.2 Communications**

The communications module is user transparent. It is a RPC based client/server program that passes information between the user interface and other higher-level routines residing on the workstation and the lower-level controls on the VME rack over the ethernet. The server resides on the VME side and receives simple commands from the client to move the robot and read the sensors and encoders returning robot position, heading, and sensor readings. Communication to the network board takes place from the VME through a serial link. The update speed of the RPC calls is set through a user selectable timer. The communications module allows the robot to be controlled from any computer.

### **7.2.3 Robot Motion**

Positioning of the robot is currently open loop relying on dead reckoning from encoder readings on the wheels. Although programmable, only the speed and direction of the robot are ever changed to keep the algorithms simple and efficient. Many different parameters can be set on the robot and many different functions are available. This project uses only a subset of the possible commands to make the control of the robot as simple as possible. The robot motion occurs in four modes, one teleoperated, one teleautonomous and two autonomous.

The teleoperated mode uses the software joystick described in Section 7.2.1. In this mode, the user simply moves the robot. This is useful for negotiating the robot when it is with in site. The obstacle avoidance algorithm can be used here so the robot can automatically change headings to avoid objects — a teleautonomous approach. The third mode is an autonomous mode used in the path planning. In this mode the robot

proceeds to a goal point avoiding obstacles (see Section 7.1.1). The last mode is a roaming mode where the robot does not have an ending destination. The robot can roam freely using the obstacle avoidance algorithm. This allows the occupancy grid to be built while the robot is unsupervised or in an unknown environment.

### 7.3 Integrated System

The stereo sensory system was designed to be completely modular. It is completely self contained. The only link needed to the higher-level system is the model information obtained and a coordinate transform for the model parameters based on the location of the sensory module with respect to the global coordinate frame utilized, in this case the encoder readings from the robot. The robot module was also designed to operate independently if needed.

The key to the system integration is the transparency of the lower-level systems [Gou94]. The operator has the advantage of taking the low-level systems for granted. They were designed for modularity and are capable of independent operation allowing the integration to occur at a higher-level. Graphical workstation interfaces allow the operator to control each system and their interaction as he sees fit in a given situation. There are three main high-level modules:

- Robot control,
- Stereo localization, and
- Visual verification.

A typical scenario consists of the following. The operator teleautonomously guides

the robot into the work space while the robot is still visible. After the robot is out of sight of the operator, he then can continue operation telerobotically relying on the range data from the ultrasonic sensors and occasional visual feedback, or he can place the robot in completely autonomous mode and allow the robot to roam freely searching for pipes. For the robot to find a pipe using the stereo vision, the pipe must be in the appropriate visual range of both cameras. This corresponds to a distance of one to three feet directly in front of the robot. Therefore, when an object falls within this range the robot stops to use the stereo cameras to determine if a pipe is present. From this data and that gathered from the ultrasonic sensors, maps of the work space and the pipes contained within it can be created. These can also be passed to graphics packages such as IGRIP for viewing and further task planning and preview.

A flow chart of the operation of the integrated system is given in Figure 7.7. The robot moves into the environment. If an object is found in front of the robot images are taken and if pipes are present they are localized using the stereo algorithms. The pipe models can then be used in preview programs. After the pipes have been modeled the robot can then move again to find more candidate objects. At the same time this loop is occurring, the robot is also creating maps of the environment in the form of occupancy grids.

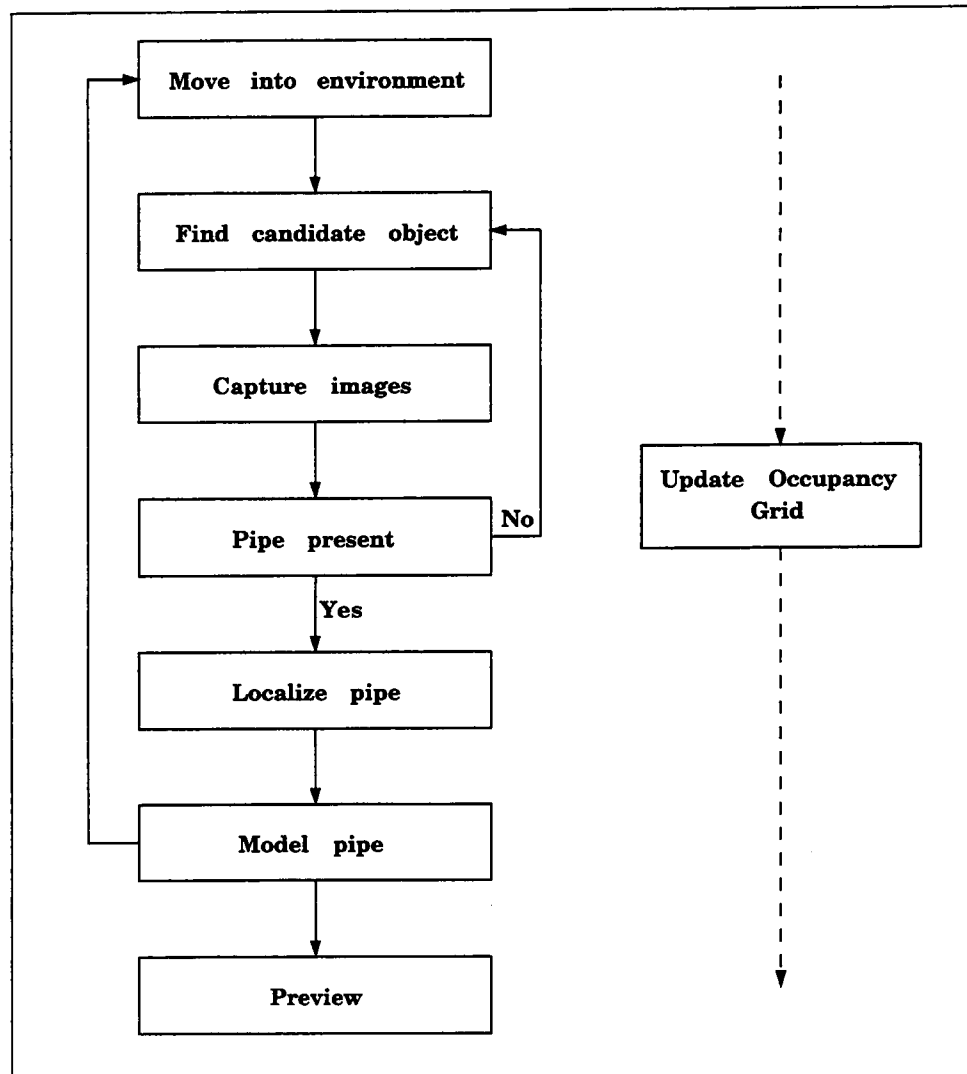


Figure 7.7: Flow chart of typical run of the integrated system. The robot moves into the environment. Candidate objects are found. If they are pipes, they are modeled. Occupancy grids are created in parallel.

## CHAPTER 8

### Experimentation and Results

Extensive experimentation has been performed using the mobile system. The robot has been run both indoors and out. The algorithms work very robustly. The only problems occurring have been due to the short comings of ultrasonic sensors in general as well as the positional encoders. Some objects which absorb sound well are not seen until the last moment. The update speed of the sensors along with the variable speed has allowed the robot to proceed without incident. The only object not seen by the robot was one corner. This problem however is repeatable and could be a communications problem with the serial link that was used during testing.

#### 8.1 Positioning

One major problem that has occurred with the mobile platform is the inaccuracy of the position. A problem with the robot is its lack of positional information. This is well known to be a major problem of which much work has been focused. In order for robust positional accuracy, the causes behind the errors in the positioning must be discovered. Here, encoders on the wheels of the robot are used to determine location and direction through dead reckoning. The major contributor to the errors is wheel slippage. Alexander [AM89] has tried to model slippage using dynamics equations governing friction and kinematics. In this case the goal is not to model the slippage, but to understand the causes and reduce the contribution of the culprits. For motion

of the robot, both the velocity and acceleration can be controlled. High speed appears to be paid for by a loss in positional accuracy. For producing occupancy maps, a lower speed is possibly needed. For this reason an accuracy test for the mobile platform is proposed. This information is needed to determine if occupancy grid maps are feasible. Others have also encountered many of the same problems. Initial testing showed visually errors of about 0.5 meters in position and 90 degrees in orientation. Therefore, a study on the error accumulated in different situations by the robot has been performed. The

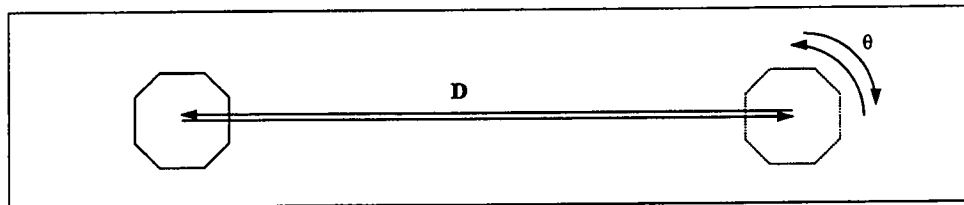


Figure 8.1: *Test path followed by mobile platform during positional accuracy testing.*

robot was moved forward for a distance  $D$ , rotated by an angle  $\theta$ , and then brought back to what it believes is its starting position (see Figure 8.1). From the difference in the position and orientation, information about varying the velocity, acceleration, distance traveled, and rotation angle was obtained with respect to the error accumulated.

The problem is determining the ending position with respect to the original. A “calibration” type method was used (see Figure 8.2). Using this, the original position did not have to be rigidly fixed. To accomplish this task, the following was done:

- First, choose 3 fixed points on the robot. The location of these with respect to the robot’s coordinate frame  $E$  can be measured.
- Using a *calibration stick* measure the distances  $a$ ,  $b$ , and  $c$ . Since these distances are fixed with respect to each other, this determines a transformation from which

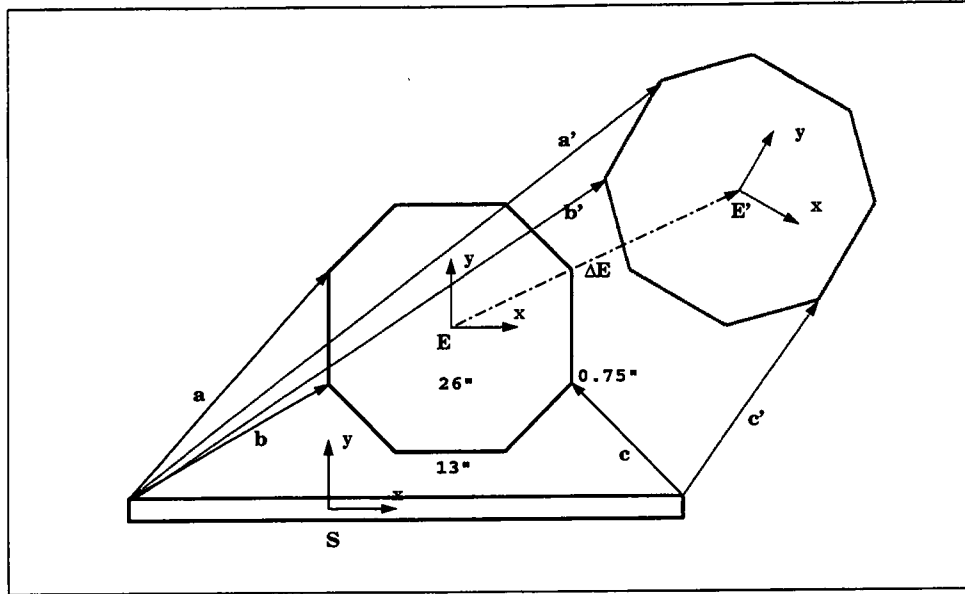


Figure 8.2: *Originally proposed robot calibration technique in which hand measurements from a fixed position were to be performed.*

the coordinate frame  $S$  can be found.

- After the robot has completed its task, the distances  $a'$ ,  $b'$ , and  $c'$  can be used to locate the new frame  $E'$  with respect to  $S$  and thus the original position  $E$ .
- From the differences in the coordinate frames the error in position and orientation can be found.

The figure shows an over exaggerated view of the starting and ending points of the robot to show the geometry used. This was the starting point along with the mobile platform's geometry shown in Figure 8.3 for deriving the needed measurements.

While deriving the equations needed to do the error studies of the robot, it was found that taking one extra measurement would greatly ease the calculations of the angles needed. Referring to Figure 8.4, the error  $\theta$  in the rotation of the robot is

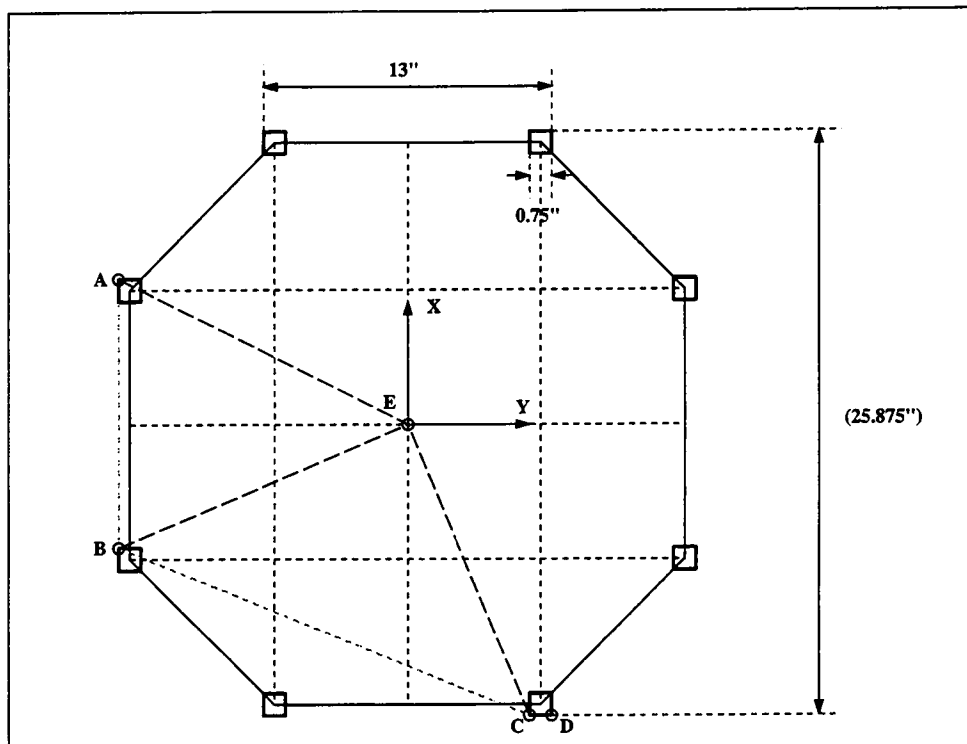


Figure 8.3: Robot geometry used in determining the offset of the internal coordinate frame along with the measurement points used in the calibration.

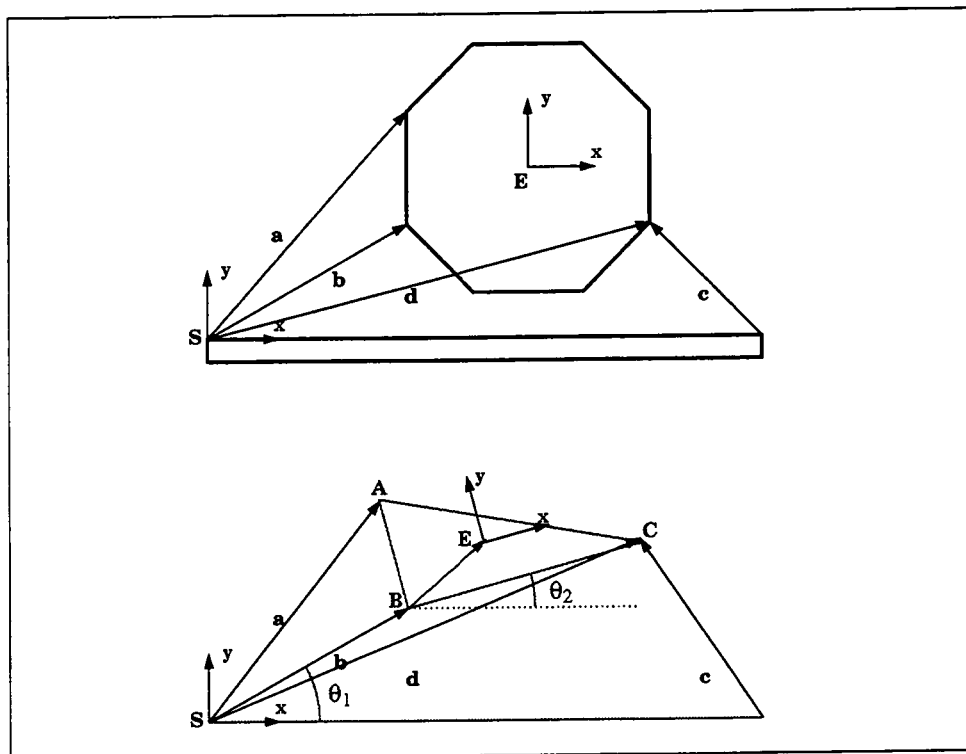


Figure 8.4: *Modified robot calibration incorporating a fourth measurement in order to reduce the amount of computation needed in calculating the robots actual position.*

calculated to be

$$\theta = \theta_1 + \theta_2 - \pi - \psi, \quad (8.1)$$

where  $\psi$  is the angle between the line  $BC$  and the x-axis of the frame  $E$ . The  $X$  and  $Y$  offsets are simply addition of vectors. This measurement can be found redundantly if needed starting with  $a$ ,  $b$ ,  $c$ , or  $d$ . For example, starting with  $b$ ,

$$X_0 = SB\cos(\theta) + BE\cos(\theta + \psi) \quad (8.2)$$

$$Y_0 = SB\sin(\theta) + BE\sin(\theta + \psi). \quad (8.3)$$

From this the radial offset is simply

$$\rho = \sqrt{X_0^2 + Y_0^2} \quad (8.4)$$

These measures can be averaged or to get a better measure of the error, for the  $x$  and  $y$  readings choose the vector which is in the direction of greatest change relative to the direction desired. From data gathered the error of the positional inaccuracy and rotational inaccuracy can be obtained without regard to a fixed starting point or direction.

The manufacturer recommends that dead reckoning using the encoders be limited to about ten meters. After this they suggest using another positioning method such as landmark recognition. Results of initial testing in the robot encoder error has confirmed this fact. The manual measurements made have less precision than the encoder readings, but they are more accurate as they were made from fixed points. They were measured with a resolution of  $\frac{1}{8}$  of an inch where the encoders have a resolution of 1 mm. Several hundred experiments were performed for the positional and rotational error calculations. Some of these are listed in Appendix A. The more interesting results are summarized

in the following graphs and tables. Table 8.1 shows that the error does increase with increasing distance, but this effect appears to be linear and only a minor contributor to the problem. The error, however, can be controlled. Tables 8.2 and 8.3, giving

Table 8.1: *Error measurement in position versus distance traveled.*

| Distance (mm) | $\mu$ | $\sigma$ |
|---------------|-------|----------|
| 2000          | 0.140 | 0.0865   |
| 4000          | 0.472 | 0.633    |

means and standard deviations of errors in location,  $\rho$ , and heading,  $\theta$ , show that by decreasing velocity and acceleration the error is reduced, mainly to reduction in the amount of slippage induced. The effect of increasing the velocity and acceleration of

Table 8.2: *Error measurement of position and heading versus velocity.*

| Velocity ( $\frac{mm}{s}$ ) | $\rho$ (mm) |          | $\theta$ ( $^{\circ}$ ) |          |
|-----------------------------|-------------|----------|-------------------------|----------|
|                             | $\mu$       | $\sigma$ | $\mu$                   | $\sigma$ |
| 100                         | 0.139       | 0.151    | 0.341                   | 0.430    |
| 200                         | 0.178       | 0.250    | 1.453                   | 3.606    |
| 250                         | 0.0805      | 0.542    | 0.955                   | 4.731    |
| 500                         | 0.557       | 0.263    | -1.649                  | 5.342    |
| 1000                        | 1.442       | 2.966    | 1.044                   | 11.346   |

the robot increases the average value of the positional offset. This results in the robot slowly moving to its left after repeated testing, a result which can be verified visually. The offset upon the mean of the angular error tends to be more random in nature, but the variance increase markedly as the velocity and acceleration are increased. These

Table 8.3: *Error measurement of position and heading versus acceleration.*

| Accel. ( $\frac{mm}{s^2}$ ) | $\rho$ (mm) |          | $\theta$ ( $^\circ$ ) |          |
|-----------------------------|-------------|----------|-----------------------|----------|
|                             | $\mu$       | $\sigma$ | $\mu$                 | $\sigma$ |
| 100                         | 0.140       | 0.0865   | 0.328                 | 0.927    |
| 250                         | 0.185       | 0.331    | 1.4324                | 5.434    |
| 1000                        | 0.828       | 2.172    | -0.642                | 8.229    |

trends are shown in the graphs of Figure 8.5. Due to the randomly occurring angular error induced by the oppositely driven wheels during purely rotational motions, accurate translational motions that cover a long distance can result in large positional errors since the heading vector is incorrect. This positional error leads to problems in calculating VFF's from previously obtained data in that an inaccurate occupancy grid will lead to inaccurate obstacle avoidance. It has led to many problems in the path planning aspect of the project. To compensate accelerations and velocities have to be reduced when more precise occupancy grids are needed.

The results of the studies are shown in the figures on the following pages. These graphs show the results of runs of purely translational or rotational motions. In each graph, the resulting encoder reading is shown along with the actual measured value of the error that has occurred due to the motion. Figure 8.6 shows the positional offset induced by varying the acceleration of the robot (effectively, changing the slope of the trapezoidal velocity profile). Clearly, larger accelerations result in larger errors. Figure 8.7 shows the rotational offset produced by the same accelerations. As mentioned previously, these errors tend to be more random, but do show increasing error as the tests proceed. Figure 8.8 shows the positional offset induced by varying the velocity of

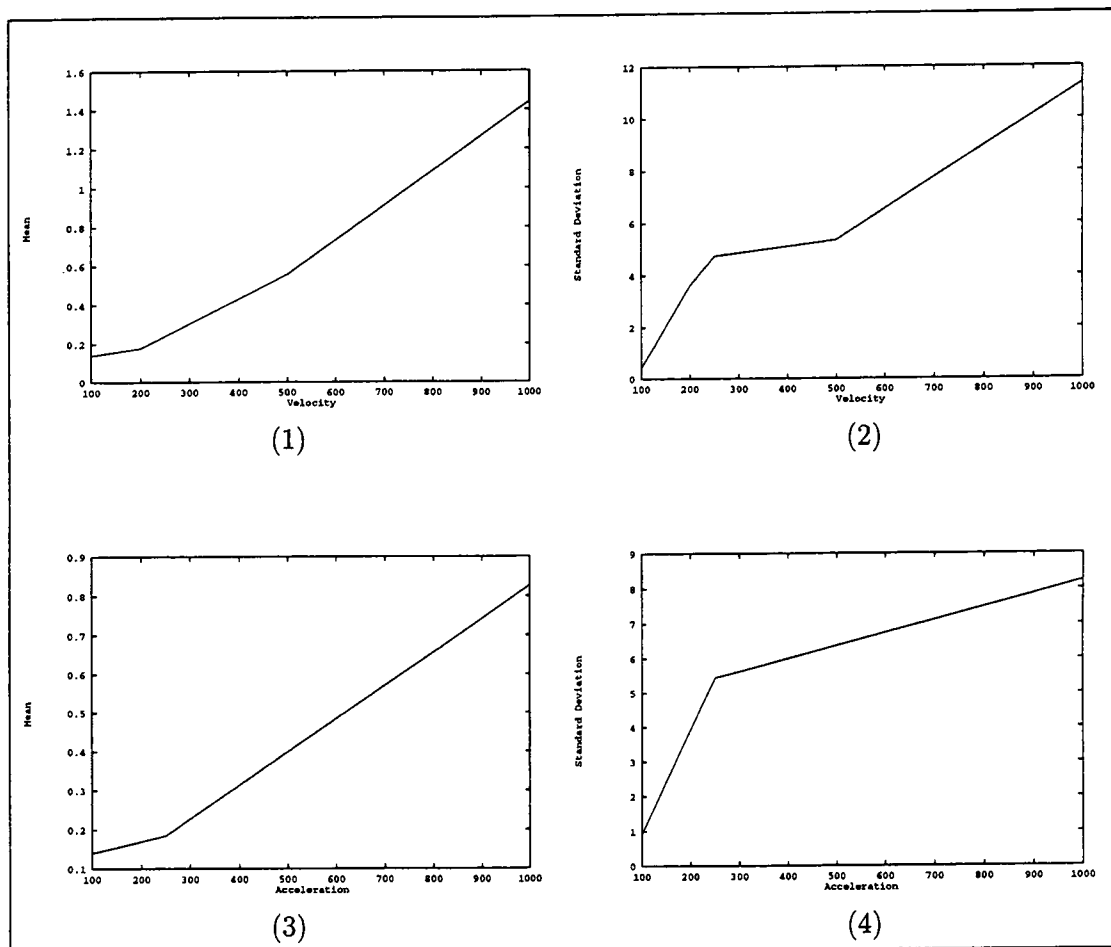


Figure 8.5: (1) Average error versus velocity for positional offset, (2) Standard deviation of error versus velocity for angular offset, (3) Average error versus acceleration for positional offset, and (4) Standard deviation of error versus acceleration for angular offset.

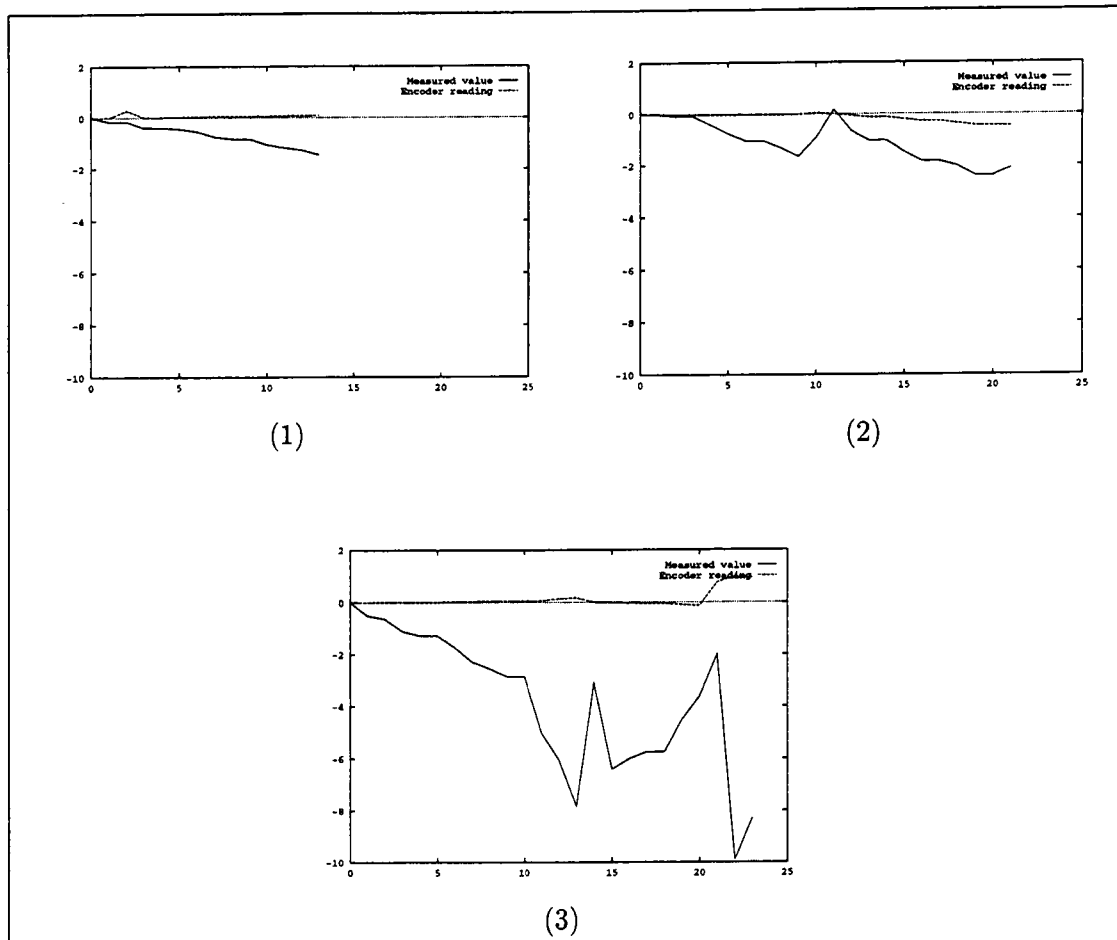


Figure 8.6: *Positional error due to acceleration of (1) 100 mm/sec<sup>2</sup>, (2) 250 mm/sec<sup>2</sup> and, (3) 1000 mm/sec<sup>2</sup>*

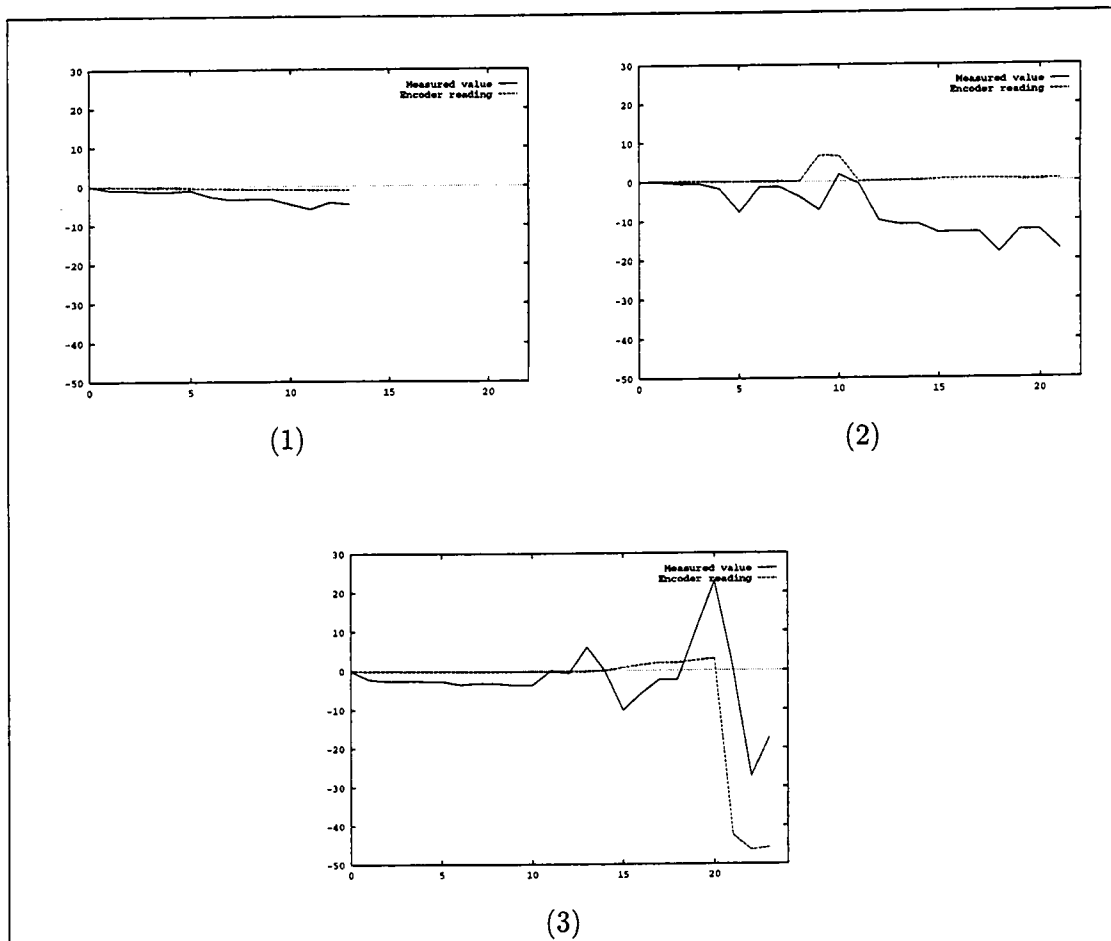


Figure 8.7: Angular error due to acceleration of (1)  $100 \text{ mm/sec}^2$ , (2)  $250 \text{ mm/sec}^2$  and, (3)  $1000 \text{ mm/sec}^2$

the robot (effectively, changing the height of the trapezoidal velocity profile leaving the slope constant). Once again, increase speeds tend to increase the error, although the effect is not as pronounced as with the increasing acceleration. Figure 8.9 shows the

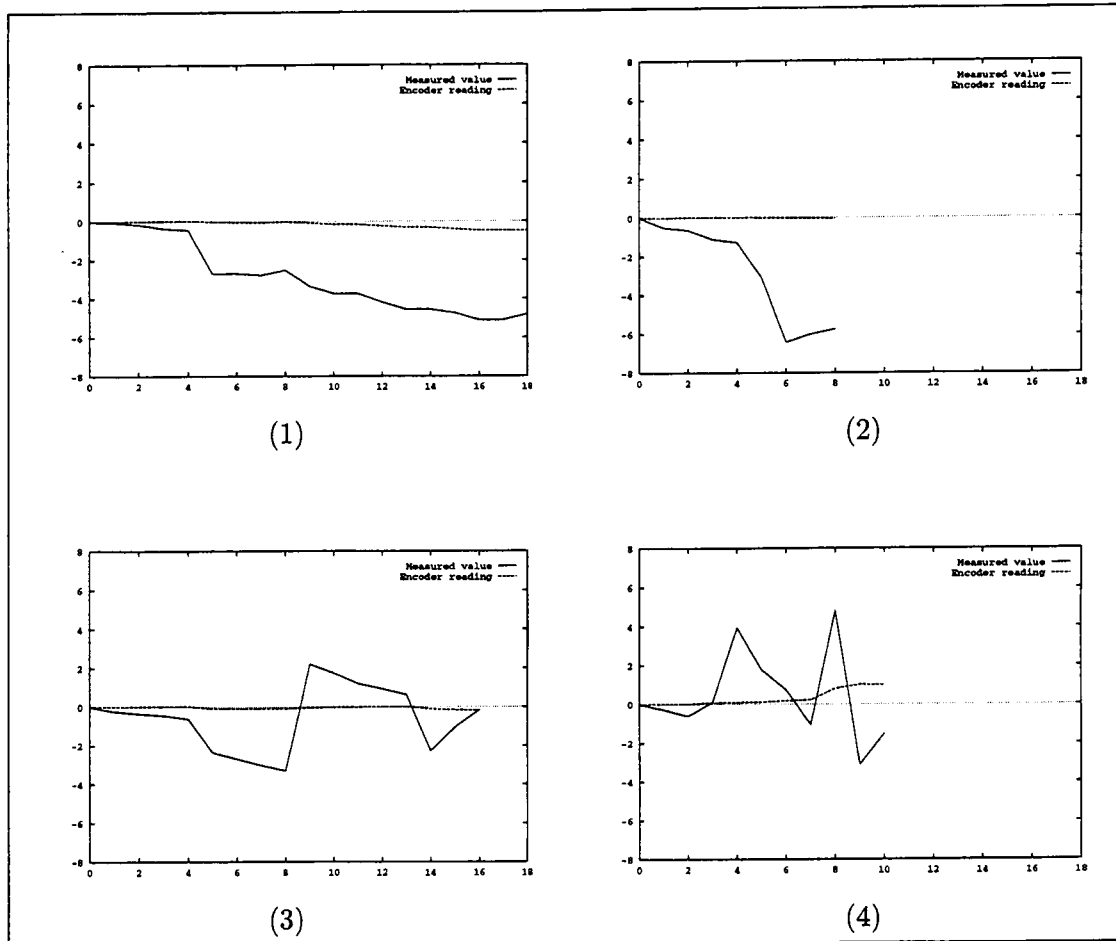


Figure 8.8: *Positional error due to velocity of (1) 200 mm/sec., (2) 250 mm/sec., (3) 500 mm/sec. and, (4) 1000 mm/sec.*

rotational offset produced by the same velocities. Here, as in the changing acceleration, the effect appears more random in nature, but grows as the acceleration is increased.

As can be seen by the graphs, at lower speeds the error is fairly consistent. At higher speeds the error becomes more erratic and larger. If positional accuracy is not

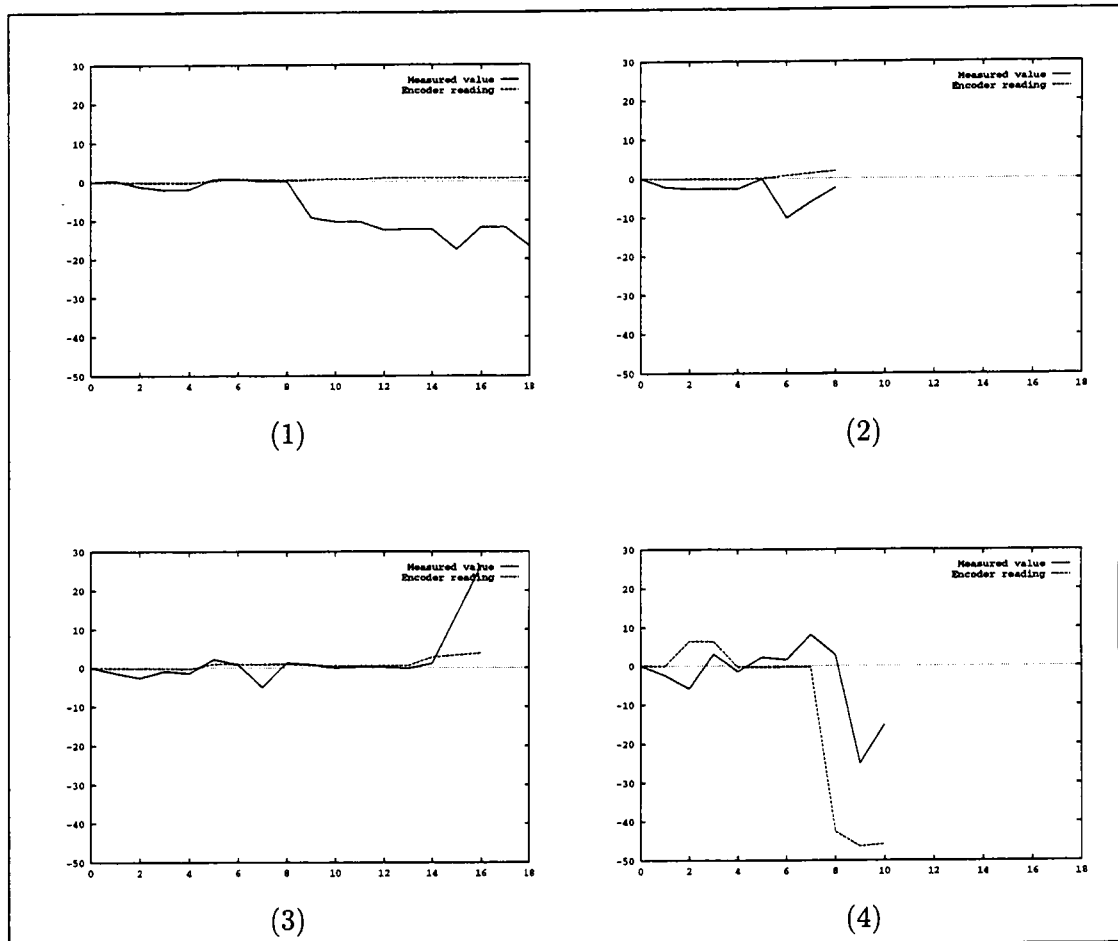


Figure 8.9: Angular error due to velocity of (1) 200 mm/sec., (2) 250 mm/sec., (3) 500 mm/sec. and, (4) 1000 mm/sec.

a consideration, as in a teleautonomous mode, higher speeds can be achieved. For mapping, however, lower velocities and accelerations are used to lower the errors induced by slippage. In each of the graphs, the motor controllers can be seen to do a good job of keeping the robot in what it thinks is the correct position and orientation keeping the error signal low. The failure comes from the wheel slippage giving faulty encoder readings. The wheels can be tightened down to reduce slipping, but this results in the platform being top heavy and tipping over.

The results of the experiments show that to get results that can be accurate for mapping over about 50 meters the speed of the robot must be low (about 150 *mm/sec* maximum) as well as the acceleration (about 250 *mm/sec*<sup>2</sup>). Figures 8.10 and 8.11 show the results of varying the velocity and acceleration upon the occupancy grid maps created. By performing some simple image processing such as grey-level morphology [Pet91], binary erosion and dilation, and edge detection, a map of the obstacles, in this case the walls, can be formed.

As is seen in Figure 8.12, the walls from the slower run are much better defined and much more straight than those from the faster run show in Figure 8.13. Slight errors in the positioning can be seen by the angles in the wall. The maps shown were made by moving the robot from the lab to the end of the hall and back with an error of five to ten pixels from the initial starting position (250-500mm). Higher speeds result in much higher errors (twenty pixels or more – 1 meter). Even five pixels is not good enough for a true system implementation. Accuracies of one to two pixels over hundreds of meters is desired. The ground truth of what the image of the hall should appear as is shown in Figure 8.14. From measuring the actual hall, the distance between the walls should be forty pixels (two meters) across the image. The map made at the lower speed

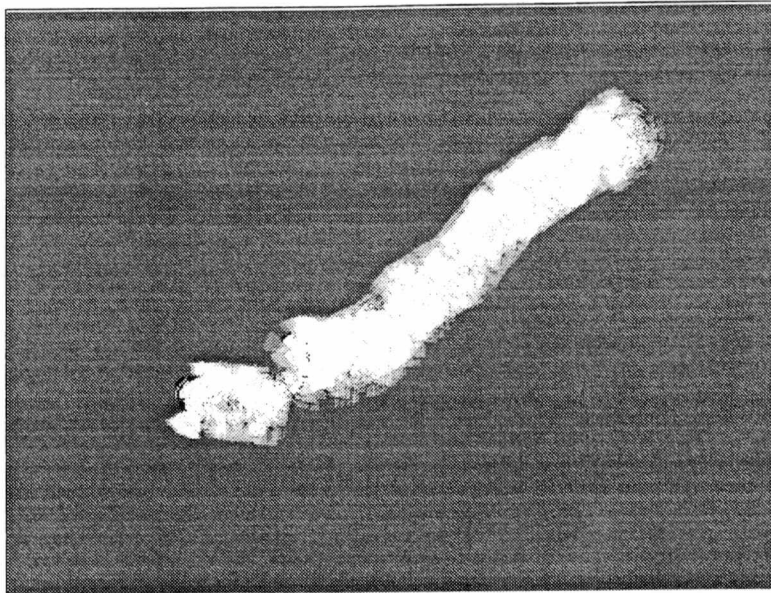


Figure 8.10: *An occupancy grid create by moving the robot at 150 mm/sec with an acceleration of 500 mm/sec<sup>2</sup>.*

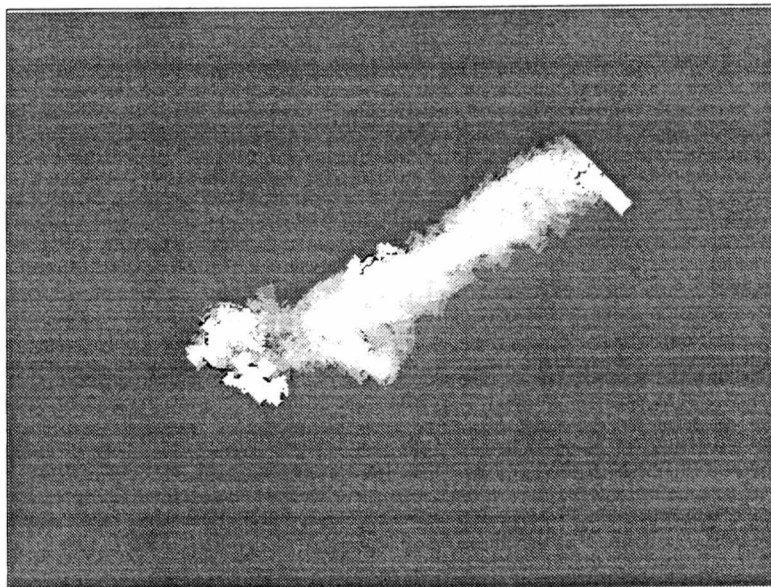


Figure 8.11: *An occupancy grid create by moving the robot at 350 mm/sec with an acceleration of 1000 mm/sec<sup>2</sup>.*

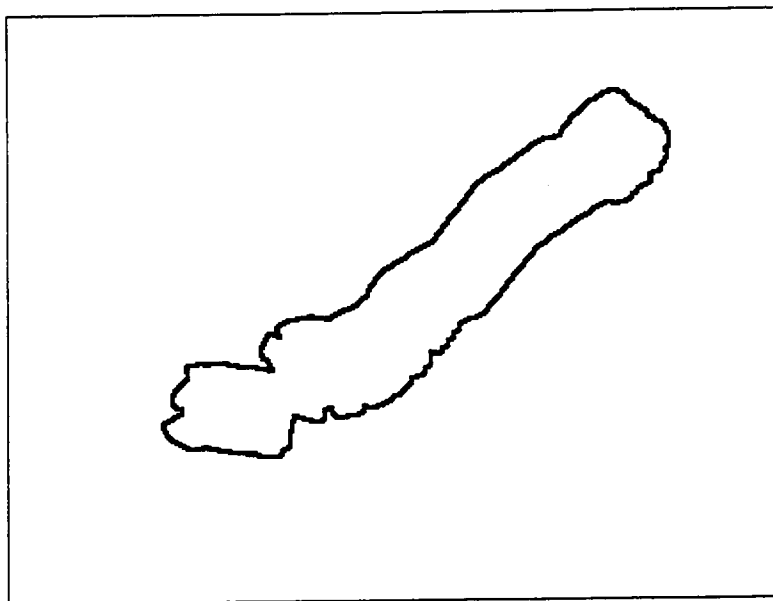


Figure 8.12: *An edge map showing the walls from the robot moving at 150 mm/sec.*

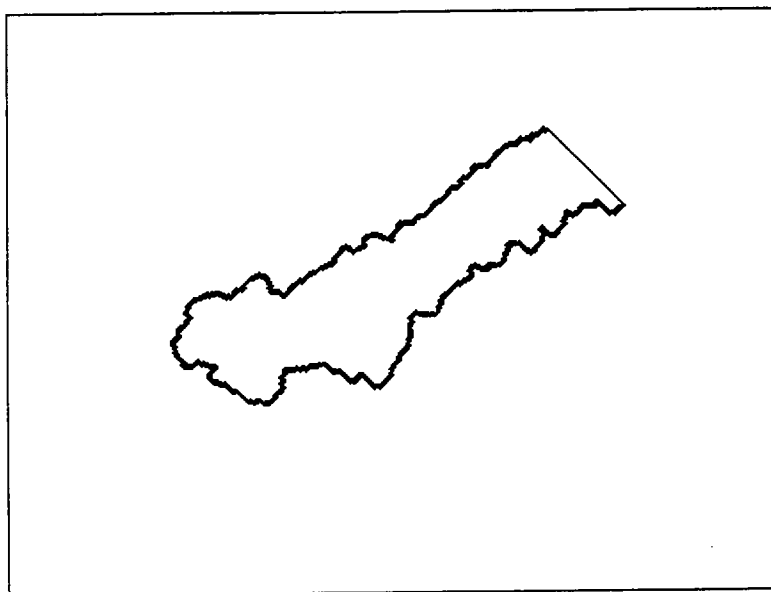


Figure 8.13: *An edge map showing the walls from the robot moving at 350 mm/sec.*

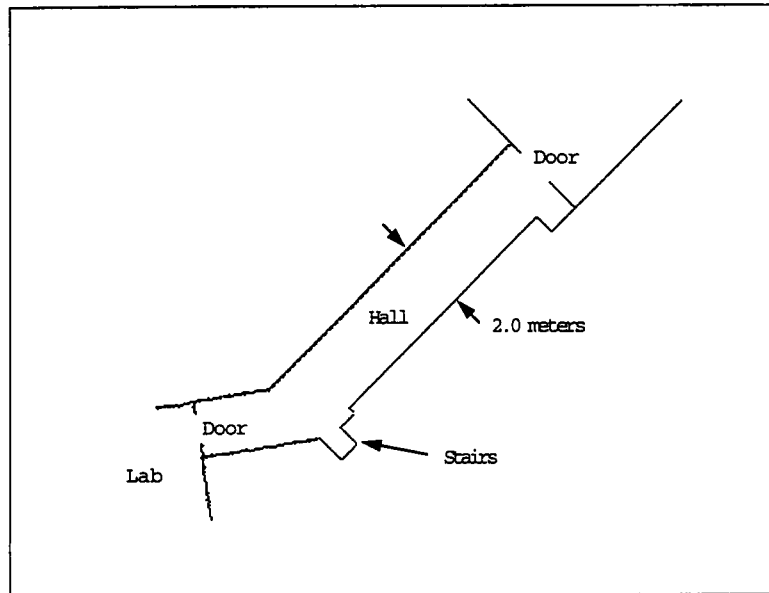


Figure 8.14: *A map showing the actual walls that the robot has detected as obstacles.*

shows a distance of forty-one to forty-two pixels across while the higher speed map has a distance of forty-two to forty-seven pixels across. A view of what the robot views from the cameras as it travels down the hall is also shown in Figure 8.15.

## 8.2 Obstacle Avoidance

The one true test for the obstacle avoidance algorithm is the fact that the robot does not hit objects in the environment. Even with all the limitations currently imposed on the system, it does a satisfactory job negotiating obstacles both inside and out. The robot has been "herded" onto and off of the elevator with no problem. This definitely causes errors in the two-dimensional occupancy grid since different floors have different obstacles. Overall the system has performed very robustly. In most areas the robot averages about 0.5 to 0.6 m/s. However, in smaller inclosed areas such as hallways and



Figure 8.15: *Hallway down which the robot was tested.*

doors, the robot proceed at an average of 0.2 to 0.3 m/s.

The actual speeds involved in the obstacle avoidance, not considering positional errors, is limited mainly by the weight of the robot which limits the acceleration and deceleration. Also, the sensor update speed limits the obstacle avoidance. With sensor readings available sooner, obstacle avoidance can be performed more quickly.

Table 8.4: *Average speeds of the robot in different environments.*

| Distance (m) | Environment       | Avg. Speed (m/s) |
|--------------|-------------------|------------------|
| 20           | Hallway           | 0.322            |
| 100          | Outdoors          | 0.610            |
| 10           | Cluttered Lab     | 0.167            |
| 20           | Building Entrance | 0.667            |
| 1            | Door              | 0.083            |

For timing data the robot was run in many different environments. A summary is given in Table 8.4. Hallway runs where performed outside the laboratory in a hall about 2 meters wide and 20 meters long (see Figure 8.15). Average speed were fairly low due to the proximity of the walls inside the middle ring. Several paths that the robot took down the hall is shown in Figure 8.16. During experimentation in the lab, the robot was never more than a few inches from an object, thus the lower speed. In the lab itself (see Figure 8.17) the robot was allowed to roam, but the confining space led to low speeds since most of the time was spent turning. When approaching doorways, the robot slows and adjusts its heading often further reducing the average speed, but unlike others [LDW92] there is no problem moving through an open door. In a more open environment such as outdoors or at the building entrance few objects are around

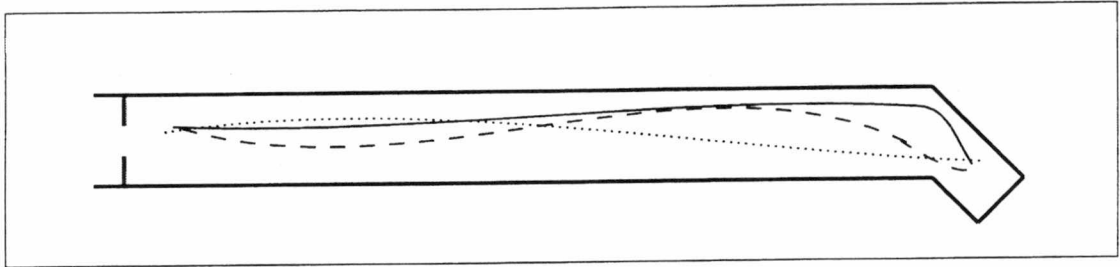


Figure 8.16: *Paths followed by the robot when it was in roam mode leaving the lab and proceeding down the hallway.*

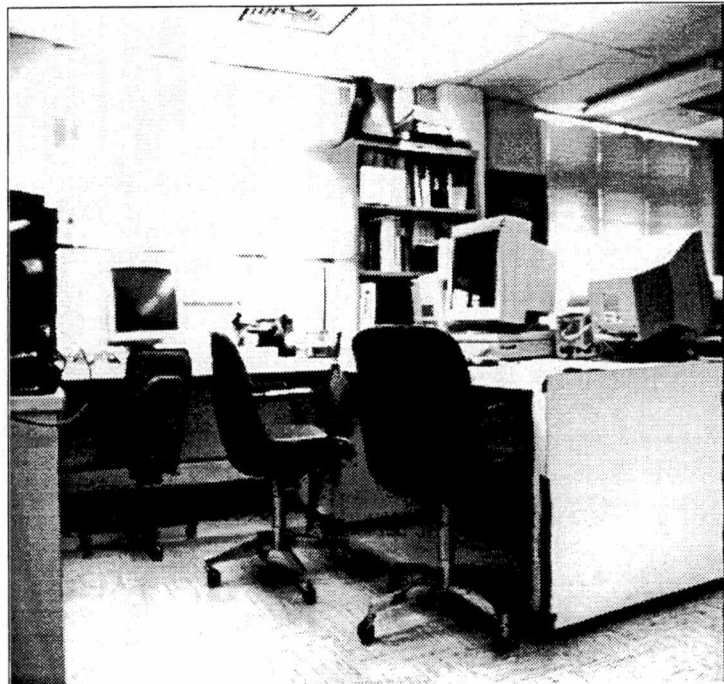


Figure 8.17: *Cluttered laboratory environment in which the robot navigated.*

and speeds are increased. For the outdoor experiments the robot roamed the sidewalks with two people following to make sure the robot did not get into trouble.

Table 8.5: *Distance maintained from objects using the obstacle avoidance algorithm.*

| Number of Walls | Environment | Distance (mm.) |
|-----------------|-------------|----------------|
| 1               | Hallway     | 146            |
| 2               | Laboratory  | 153            |
| 3               | Corridor    | 5              |

To test the absolute limits of the algorithm, the robot, in the roam mode, was sent into a corridor slightly less than a meter wide giving the robot only 5 in. clearance on each side. The end of the corridor was blocked to see if the robot would be able to return. The robot did return every time, but only about 20% of the time was this accomplished without running into one of the walls at the end of the corridor. For most commonly encountered obstacles, such as walls or corners, the robot has no problem in turning away. From Table 8.5 about the closest the robot comes is 150mm.

Also, testing beyond the safe limits of operation, the speeds of the robot were set to 400mm/sec., 800mm/sec., and 1200mm/sec. for the slow, normal, and fast speeds, respectively, and sent down the hallway again for video taping. The robot was able to proceed with little problem, although it was very close to some objects. It would not be safe to allow the robot to proceed at these speeds normally with the limited update rate of the sensors.

### 8.3 Integrated System

The modularity of the design was used to allow for ease of integration. The high level systems co-exist on the same graphical workstation allowing the operator to chose the utilization of each subsystem. Testing was performed on the pipes shown in Figure 8.18. For testing, the operator chose a goal point for the robot. The robot then proceeded to

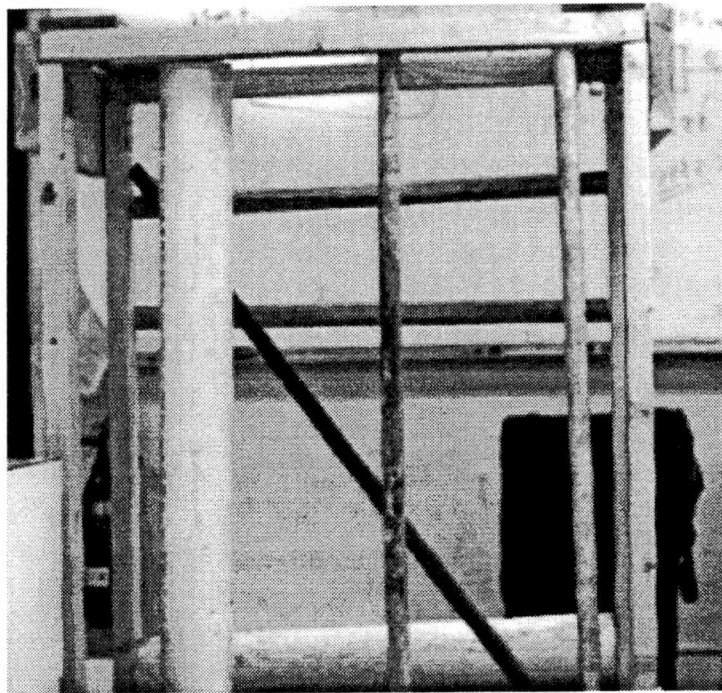


Figure 8.18: *Pipes to be modeled in a typical environment.*

the point and stopped to send images back to the operator (requiring about 25 seconds over the wireless ethernet link.) If pipes were present, which they were, they could then be modeled as outlined previously in Chapter 7 and the flow graph in Figure 7.7.

With each of the user interfaces available to the user, he can interact with them as needed for a particular scenario. For example, at any time the operator can choose

to operate the robot completely manually, teleautonomously, or can allow the robot to move autonomously. When the robot is within visual range, manual and teleautonomous modes are usually the desired interaction philosophy. When the robot is not visible, the operator does not have much choice but to allow autonomous operation. The ease by which the operator can change from one mode to another is a key ingredient in overall efficiency of the integrated system.

The robot was moved both autonomously and teleautonomously to the pipes to perform localization and modeling during testing. Due to the limited nature of the robot, the distance to the pipes was kept low. The results of the modeling are shown in Figure 8.19.



Figure 8.19: Models created from the integrated system with the stereo cameras mounted on the mobile platform.

## CHAPTER 9

### Concluding Remarks

In this thesis has been described a unique modular hardware and software integrated system that can be used for environmental mapping in cluttered and unstructured environments under certain restrictions, such as limited speed and distance. Emphasized are the modular design and the human-machine interaction. Different control philosophies allow optimal production in a variety of situations. The human-machine system concept and its implementation follows a totally integrated architecture. That is, all subassemblies, interfaces, communication protocols, and algorithms are developed with a modular systems concept for ease of integration and are successfully demonstrated.

This thesis deals with the low-level "reflex" of obstacle avoidance that is performed while the robot moves along its path as well as some of the higher-level tasks involved in path planning and robot motion in order to negotiate through a hazardous environment. The overall high-level interaction with the system is made as simple and user friendly as possible using a graphical interface to control high-level tasks and keeping all low-level communications and processing transparent. The robot used for experimentation is a unique wheeled mobile platform and is equipped with many different sensors including ultrasonic range sensors and cameras. A quick and efficient obstacle avoidance algorithm has been created using sixteen ultrasonic range sensor and one infrared proximity sensor using weighted sums of sensor readings and variable speed. This allows the robot to quickly negotiate around obstacles.

Detailed interactive modeling of objects in the environment, in this case pipes, is easily accomplished by the operator. Editing of the models occurs on a visual overlay so that the operator can visually verify the results. From the editor, manipulation points can be chosen which can then be used in either real or virtual environments.

The integrated system is very robust. The vision modules have been tested on many platforms showing transferability with minimum effort. The mobile platform has also proven itself to be robust in its abilities. Although currently limited by weight and sensor bandwidth, the robot has performed as expected within its limits and surprisingly well outside the limits. The mobile platform itself has contributed to the research area

- Reflexive obstacle avoidance,
- Variable speed,
- Transparent low-level control, and
- User friendly interfaces for high-level interaction.

Extensive experimentation has proven the system to be robust and accurate when used within its limits. These limits, however, do not incorporate a complete practical system, but do show the systems feasibility. A practical integrated system would require further development of the subsystems mentioned, especially the mobile platform. Much more precise global positioning is needed in order for the pipe localization to be useful.

The wheeled robot used here will only perform at limited speeds on smooth surfaces. It would not be capable of negotiating through the small spaces and rough floors common in an old nuclear facility. This would require a much smaller robot with better global positioning.

## 9.1 Future Applications

The research presented throughout this thesis lends itself easily to several new research areas. Currently a radiation sensor is being integrated onto the mobile platform. Among other things, this will allow for multiple constraint navigation. Along with occupancy grids, a radiation grid will also be created. This will allow the radioactive areas to act as pseudo objects and the robot can avoid them, or in some cases, the situation may require the robot to enter these areas to reduce exposure to humans.

One major improvement in the system would come from specialized hardware for interfacing to the sensors with built in DSP making the update rate of the obstacle avoidance algorithm much faster allowing for higher speeds on the robot. Higher speeds lead to higher positioning error, thus another area of research into the use of external positioning devices would be helpful. Many global positioning systems are now under development, from the use of satellite to gyroscopes. Since the robot already has cameras, a way of performing localization by visual cues or landmark recognition could be used.

One of the contributions of this thesis has been the idea of varying the speed of the robot with respect to the proximity of obstacles. Fuzzy logic algorithms used to determine the speed of the robot would allow for a smooth transition from the slowest to the fastest speed instead of the discrete jumps that now exists. It would also lead to higher average speeds for the robot.

Another future possibility is to use the octree approach to occupancy maps. This way the environment can be sub-divided into smaller and smaller quadrants as needed for more resolution.

## BIBLIOGRAPHY

## BIBLIOGRAPHY

- [AK89] J. L. Agnew and R. C. Knapp. *Linear Algebra With Applications*. Brooks/Cole Publishing Co., Belmont, California, 1989.
- [AM89] J. C. Alexandar and J. H. Moddocks. On the kinematics of wheeled mobile robots. *International Journal of Robotics Research*, pages 15-27, August 1989.
- [Apo92] M. K. Apostolos, et. al. Multisensory feedback in advanced teleoperations: Benefits of auditory cues. In *Proceedings of SPIE Conference on Intelligent Robotics and Computer Vision*, Boston, MA, November 1992.
- [Ark89] R. C. Arkin. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, pages 92-112, August 1989.
- [Ark93] R. C. Arkin, et. al. Active avoidance: Escape and didging behaviors for reactive control. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(1):175-192, 1993.
- [Aud92] K. Audenaert, et. al. Accurate ranging of multiple objects using ultrasonic sensors. In *Proceedings of the 1992 IEEE Conference on Robotics and Automation*, Nice, France, May 1992.
- [BC89] J. M. Bailey and G. K. Corbet. Dynamic modeling used for the addition of robotic operation to the advanced servomanipulator teleoperator. In *Proceedings of IEEE Southeastcon*, 1989.
- [BK89] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179-1187, 1989.
- [BK90] J. Borenstein and Y. Koren. Teleautonomous guidance for mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1437-1443, November 1990.
- [BK91a] J. Borenstein and Y. Koren. Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 7(4):535-539, August 1991.
- [BK91b] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278-288, June 1991.
- [BK92] J. Borenstein and Y. Koren. Noise rejection for ultrasonic sensors in mobile robot applications. In *Proceedings of the 1992 IEEE Conference on Robotics and Automation*, pages 1727-1732, Nice, France, May 1992.

- [Bro86] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1), 1986.
- [Bru93] M. Brunfaut, et. al. A correlation coprocessor for accurate real-time ultrasonic ranging of multiple objects using the transputer. *Microprocessors and Microsystems*, 17(10), December 1993.
- [But92] P. L. Butler. Fy 1992 robotics technology demonstration for decontamination and decommissioning. Technical report, Oak Ridge National Laboratory and University of Tennessee, 1992.
- [But93] P. L. Butler, et. al. An architecture for robotic system integration. In *Proceedings of the Fifth Topical Meeting on Robotics*, volume 1, pages 279-285, Knoxville, TN, April 1993.
- [Con92] C. Congdon, et. al. Carmel versus flakey: A comparison of two winners. *IEEE AI Magazine*, pages 49-57, Winter 1992.
- [Cra89] J. Craig. *Introduction of Robotics, Mechanics and Control*. Addison-Wesley, second edition, 1989.
- [CT92a] C. Chen and M. M. Trivedi. Mobile robots with articulated tracks and manipulators: Intelligent control and graphical interface for teleoperation. In *Proceedings of SPIE Conference on Intelligent Robotics and Computer Vision*, Boston, MA, November 1992.
- [CT92b] C. Chen and M. M. Trivedi. Virtual and real world operation of mobile robotic manipulators: Integrated simulation, visualization, and control environment. In *Proceedings of SPIE Symposium on Optical Engineering in Aerospace Sensing*, Orlando, FL, April 1992.
- [CTB92] C. Chen, M. M. Trivedi, and C. R. Bidlack. Simulation and graphical interface for programming and operation of sensor-based robots. In *Proceedings of the 1992 IEEE Conference on Robotics and Automation*, Nice, France, May 1992.
- [DB93] T. Dean and R. P. Bonasso. 1992 aaai robot exhibition and competition. *IEEE AI Magazine*, pages 35-48, Spring 1993.
- [DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.
- [Elf87] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3), June 1987.
- [Elf89] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer Magazine*, June 1989.

- [FGL87] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. *Robotic Control, Sensing, Vision, and Intelligence*. McGraw-Hill, 1987.
- [Fol90] A. A. Foley, et. al. *Computer Graphics: Principles and Practice*. Addison-Wesley Publishing Company, Reading, MS, 1990.
- [Gat93] E. Gat. On the role of stored internal state in the control of autonomous mobile robots. *IEEE AI Magazine*, pages 64-73, Spring 1993.
- [Gou94] C. Gourley, et. al. An integrated mobile robotic system for environmental mapping and object localization. In *Proceedings of SPIE Symposium on Optical Engineering in Aerospace Sensing*, Orlando, FL, April 1994.
- [GT94] C. Gourley and M. Trivedi. Sensor based obstacle avoidance and mapping for fast mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.
- [GW87] R. C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley Publishing Company, Reading, MS, 1987.
- [Hor89] B. K. P. Horn. *Robot Vision*. The MIT Press, 1989.
- [Jon93] J. Jones. Real-time construction of three-dimensional occupancy maps. *IEEE*, 1993.
- [Kha85] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, St. Louis, MO, March 1985.
- [Kim85] W. Kim, et. al. Visual enhancements for telerobotics. In *Proceedings of the 1985 IEEE Conference on Systems, Man, and Cybernetics*, Tuscon, AR, November 1985.
- [Kro84] B. H. Krogh. A generalized potential field approach to obstacle avoidance control. In *International Robotics Research Conference*, Bethlehem, PA, August 1984.
- [KW93] D. Kortenkamp and T. Weymouth. Integrating high-speed avoidance, long-range sensing and landmark triangulation to accomplish a find and visit task. Presented at the Second Annual DOE Robotics for Advanced Reactors Program Student Conference, January 1993.
- [LDW92] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, 1992.
- [LLS91] J. Latombe, A. Lazanas, and S. Shekhar. Robot motion planning with uncertainty in control and sensing. *IEEE AI Magazine*, pages 1-47, 1991.

- [Mat92] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Journal of Robotics and Automation*, RA-8(3):304-312, June 1992.
- [ME88] L. Matthies and A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, PA, April 1988.
- [Mil93] D. Miller. A twelve-step program to more efficient robotics. *IEEE AI Magazine*, pages 60-63, Spring 1993.
- [oE89] U. S. Department of Energy. Environmental restoration and waste management robotics technology development program: Five year program plan. Technical report, Oak Ridge National Laboratory, 1989.
- [PAC93] H. Peremans, K. Audenaert, and J. M. Van Camperhout. A high-resolution sensor based on tri-aural perception. *IEEE Transactions on Robotics and Automation*, 9(1), February 1993.
- [Pet91] R. A. Peters. Software for 2d and 3d mathematical morphology. Technical report, Vanderbilt University, 1991.
- [Pol84] Polariod Corporation. *Ultrasonic Ranging System*, 1984.
- [Rao86] S. V. N. Rao, et. al. Robot navigation in an unexplored terrain: Algorithms and analysis. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 1986.
- [Rao93] N. S. V. Rao, et. al. Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms. Technical report, Oak Ridge National Laboratory, July 1993.
- [She74] T. B. Sheridan. *Man-Machine Systems*. The MIT Press, Cambridge, MS, 1974.
- [She89] T. B. Sheridan. Telerobotics. *Automatica*, 25(4):487-507, 1989.
- [She92] T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, Cambridge, MS, 1992.
- [Sil89] Silicon Graphics, Inc. *Silicon Graphics Graphics Library Manual*, 1989.
- [SS83] J. T. Schwartz and M. Sharir. On the piano movers' problem: The case of a rod moving in three-dimensional space amidst polyhedral obstacles. Technical report, Computer Science Department, New York University, October 1983.
- [Ste89] B. Steer. Trajectory planning for a mobile robot. *International Journal of Robotics Research*, pages 3-14, August 1989.

- [Str86] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Wellesley, MA, 1986.
- [TC93a] M. Trivedi and ChuXin Chen. Developing telerobotic systems using virtual reality concepts. In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 352–359, Yokohama, Japan, July 1993.
- [TC93b] M. M. Trivedi and C. Chen. Developing sensor-based robotic systems using virtual reality concepts. In *Proceedings of the Fifth Topical Meeting on Robotics*, volume 1, pages 165–172, Knoxville, TN, April 1993.
- [Tex84] Texas Instruments Corporation. *TI Linear Circuits Data Book*, 1984.
- [TG74] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley Publishing Company, Reading, MS, 1974.
- [Tha92] S. M. Thayer, et. al. Three-dimensional sensing, graphics and interactive control in a human-machine system for decontamination and decommissioning applications. In *Proceedings of SPIE Conference on Intelligent Robotics and Computer Vision*, Boston, MA, November 1992.
- [Tha93a] S. M. Thayer. Design and analysis of human-machine and automatic vision systems for cylinder detection and localization. Master's thesis, University of Tennessee, 1993.
- [Tha93b] S. M. Thayer, et. al. On-line stereo vision and graphical interface for decontamination and decommissioning applications using the advanced servo manipulator. In *Proceedings of the Fifth Topical Meeting on Robotics*, volume 1, pages 287–294, Knoxville, TN, April 1993.
- [Tra91a] Transitions Research Corporation, Danbury, CT. *LabMate User Manual*, 5.211-f edition, 1991.
- [Tra91b] Transitions Research Corporation, Danbury, CT. *Proximity Subsystem User Manual*, 4.6h edition, 1991.
- [Tri93] M. M. Trivedi. Human-robot cooperation using virtual reality concepts. *Robots and Machine Perception*, August 1993.
- [Vie93] T. Vieville, et. al. Autonomous navigation of a mobile robot using inertial and visual cues. In *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 360–367, Yokohama, Japan, July 1993.
- [Yos90] T. Yoshikawa. *Foundations of Robotics: Analysis and Control*. MIT Press, 1990.

## APPENDICES

## APPENDIX A

### Error Studies

The following table contains the measurements and encoder readings made from some of the experimental runs used in determining the positional accuracy of the robot.

| Accl | Vel  | Dist | Angle | Base   | A      | B      | C      | D      | X       | Y       | Ang     |
|------|------|------|-------|--------|--------|--------|--------|--------|---------|---------|---------|
| 100  | 100  | 1000 | 0     | 36.000 | 24.250 | 12.750 | 13.000 | 25.000 | 200.250 | 150.190 | 0.000   |
| 100  | 100  | 1000 | 0     | 36.000 | 24.750 | 13.000 | 13.250 | 24.750 | 200.250 | 150.190 | 359.830 |
| 100  | 100  | 1000 | 0     | 36.000 | 24.750 | 13.000 | 13.250 | 24.750 | 200.600 | 150.190 | 359.750 |
| 100  | 100  | 1000 | 0     | 36.000 | 24.625 | 12.875 | 13.375 | 24.500 | 200.270 | 150.190 | 359.610 |
| 100  | 200  | 1000 | 0     | 36.000 | 24.625 | 12.875 | 13.375 | 24.500 | 200.270 | 150.190 | 359.610 |
| 100  | 200  | 1000 | 0     | 36.000 | 25.125 | 13.125 | 13.750 | 24.375 | 200.280 | 150.190 | 359.510 |
| 100  | 200  | 1000 | 0     | 36.000 | 25.500 | 13.625 | 14.000 | 24.125 | 200.300 | 150.200 | 359.360 |
| 100  | 200  | 1000 | 0     | 36.000 | 25.625 | 13.625 | 14.125 | 23.875 | 200.300 | 150.200 | 359.310 |
| 100  | 200  | 1000 | 0     | 36.000 | 25.750 | 13.750 | 14.375 | 23.750 | 200.310 | 150.200 | 359.210 |
| 100  | 500  | 1000 | 0     | 36.000 | 25.750 | 13.750 | 14.375 | 23.750 | 200.310 | 150.200 | 359.210 |
| 100  | 500  | 1000 | 0     | 36.000 | 26.250 | 14.125 | 14.750 | 23.375 | 200.320 | 150.200 | 359.070 |
| 100  | 500  | 1000 | 0     | 36.000 | 26.500 | 14.500 | 15.000 | 23.125 | 200.330 | 150.210 | 359.000 |
| 100  | 500  | 1000 | 0     | 36.000 | 26.125 | 14.125 | 15.125 | 23.125 | 200.330 | 150.210 | 358.950 |
| 100  | 500  | 1000 | 0     | 36.000 | 26.375 | 14.250 | 15.375 | 22.875 | 200.340 | 150.210 | 358.840 |
| 250  | 200  | 2000 | 0     | 36.000 | 22.625 | 10.625 | 15.250 | 22.000 | 200.250 | 150.190 | 359.970 |
| 250  | 200  | 2000 | 0     | 36.000 | 22.750 | 10.625 | 15.250 | 22.000 | 200.250 | 150.150 | 0.190   |
| 250  | 200  | 2000 | 0     | 36.000 | 22.875 | 10.750 | 15.375 | 21.875 | 200.230 | 150.150 | 0.170   |
| 250  | 500  | 2000 | 0     | 36.000 | 22.875 | 10.750 | 15.375 | 21.875 | 200.230 | 150.150 | 0.170   |
| 250  | 500  | 2000 | 0     | 36.000 | 22.875 | 10.750 | 15.625 | 21.500 | 200.230 | 150.150 | 0.110   |
| 250  | 500  | 2000 | 0     | 36.000 | 22.375 | 10.250 | 15.250 | 21.250 | 200.230 | 150.150 | 0.100   |
| 250  | 500  | 2000 | 0     | 36.000 | 22.625 | 10.500 | 16.250 | 20.875 | 200.230 | 150.150 | 0.130   |
| 250  | 1000 | 2000 | 0     | 36.000 | 22.625 | 10.500 | 16.250 | 20.875 | 200.230 | 150.150 | 0.130   |
| 250  | 1000 | 2000 | 0     | 36.000 | 23.500 | 11.250 | 16.750 | 20.375 | 200.240 | 150.150 | 0.080   |
| 250  | 1000 | 2000 | 0     | 36.000 | 22.000 | 9.875  | 15.625 | 20.375 | 200.250 | 150.160 | 6.580   |
| 250  | 1000 | 2000 | 0     | 36.000 | 21.750 | 9.500  | 15.875 | 21.250 | 200.330 | 150.150 | 6.490   |

| Accl | Vel  | Dist | Angle | Base   | A      | B      | C      | D      | X       | Y       | Ang     |
|------|------|------|-------|--------|--------|--------|--------|--------|---------|---------|---------|
| 250  | 200  | 0    | 45    | 36.000 | 23.250 | 10.875 | 15.125 | 22.125 | 200.250 | 150.190 | 0.020   |
| 250  | 200  | 0    | 45    | 36.000 | 25.125 | 12.875 | 16.375 | 20.625 | 200.220 | 150.120 | 0.120   |
| 250  | 200  | 0    | 45    | 36.000 | 25.375 | 13.125 | 16.875 | 20.125 | 200.190 | 150.050 | 0.250   |
| 250  | 200  | 0    | 90    | 36.000 | 25.375 | 13.125 | 16.875 | 20.125 | 200.190 | 150.050 | 0.250   |
| 250  | 200  | 0    | 90    | 36.000 | 25.375 | 13.125 | 17.250 | 19.625 | 200.130 | 149.990 | 0.520   |
| 250  | 200  | 0    | 90    | 36.000 | 25.250 | 13.000 | 17.625 | 19.250 | 200.080 | 149.930 | 0.590   |
| 250  | 200  | 0    | 120   | 36.000 | 25.250 | 13.000 | 17.625 | 19.250 | 200.080 | 149.930 | 0.590   |
| 250  | 200  | 0    | 120   | 36.000 | 25.250 | 13.125 | 17.750 | 19.000 | 200.000 | 149.890 | 0.540   |
| 250  | 200  | 0    | 120   | 36.000 | 25.500 | 13.375 | 18.500 | 18.500 | 199.930 | 149.850 | 0.460   |
| 250  | 200  | 0    | 180   | 36.000 | 25.500 | 13.375 | 18.500 | 18.500 | 199.930 | 149.850 | 0.460   |
| 250  | 200  | 0    | 180   | 36.000 | 25.125 | 12.875 | 17.750 | 19.000 | 199.920 | 149.840 | 0.490   |
| 1000 | 250  | 2000 | 0     | 36.000 | 30.250 | 18.375 | 16.500 | 26.500 | 200.250 | 150.190 | 0.000   |
| 1000 | 250  | 2000 | 0     | 36.000 | 31.250 | 19.125 | 17.500 | 25.625 | 200.250 | 150.190 | 359.880 |
| 1000 | 250  | 2000 | 0     | 36.000 | 31.500 | 19.375 | 17.875 | 25.375 | 200.260 | 150.190 | 359.900 |
| 1000 | 250  | 2000 | 0     | 36.000 | 31.000 | 18.750 | 17.625 | 25.000 | 200.270 | 150.190 | 359.790 |
| 1000 | 250  | 2000 | 0     | 36.000 | 31.000 | 18.875 | 18.000 | 24.750 | 200.270 | 150.190 | 359.810 |
| 1000 | 500  | 2000 | 0     | 36.000 | 31.000 | 18.875 | 18.000 | 24.750 | 200.270 | 150.190 | 359.810 |
| 1000 | 500  | 2000 | 0     | 36.000 | 31.125 | 19.000 | 18.500 | 24.125 | 200.280 | 150.200 | 359.720 |
| 1000 | 500  | 2000 | 0     | 36.000 | 30.750 | 18.500 | 18.625 | 23.625 | 200.290 | 150.200 | 359.720 |
| 1000 | 500  | 2000 | 0     | 36.000 | 30.875 | 18.625 | 19.125 | 23.250 | 200.300 | 150.210 | 359.720 |
| 1000 | 500  | 2000 | 0     | 36.000 | 30.875 | 18.625 | 19.375 | 22.875 | 200.310 | 150.210 | 359.710 |
| 1000 | 1000 | 2000 | 0     | 36.000 | 30.875 | 18.625 | 19.375 | 22.875 | 200.310 | 150.210 | 359.710 |
| 1000 | 1000 | 2000 | 0     | 36.000 | 27.750 | 15.500 | 19.125 | 21.375 | 200.320 | 150.210 | 359.690 |
| 1000 | 1000 | 2000 | 0     | 36.000 | 23.750 | 14.125 | 18.625 | 20.625 | 200.400 | 150.210 | 359.710 |
| 1000 | 1000 | 2000 | 0     | 36.000 | 22.750 | 10.875 | 18.875 | 19.625 | 200.450 | 150.210 | 359.740 |
| 1000 | 250  | 0    | 90    | 36.000 | 25.000 | 13.125 | 14.000 | 24.375 | 200.250 | 150.190 | 0.050   |
| 1000 | 250  | 0    | 90    | 36.000 | 26.625 | 14.375 | 17.500 | 20.125 | 200.240 | 150.170 | 0.720   |
| 1000 | 250  | 0    | 90    | 36.000 | 25.375 | 13.125 | 16.625 | 21.000 | 200.230 | 150.160 | 1.350   |
| 1000 | 250  | 0    | 90    | 36.000 | 24.125 | 12.000 | 16.000 | 21.625 | 200.220 | 150.140 | 1.960   |
| 1000 | 500  | 0    | 90    | 36.000 | 24.125 | 12.000 | 16.000 | 21.625 | 200.220 | 150.140 | 1.960   |
| 1000 | 500  | 0    | 90    | 36.000 | 19.875 | 8.375  | 13.875 | 23.875 | 200.190 | 150.110 | 2.490   |
| 1000 | 500  | 0    | 90    | 36.000 | 14.625 | 5.625  | 12.625 | 25.375 | 200.170 | 150.080 | 3.010   |
| 1000 | 1000 | 0    | 90    | 36.000 | 24.250 | 12.750 | 12.250 | 25.750 | 200.050 | 151.720 | 317.510 |
| 1000 | 1000 | 0    | 90    | 36.000 | 28.125 | 16.375 | 21.625 | 15.250 | 200.330 | 151.690 | 313.680 |
| 1000 | 1000 | 0    | 90    | 36.000 | 24.625 | 12.500 | 18.125 | 18.625 | 200.340 | 151.640 | 314.330 |

## VITA

Christopher S. Gourley was born in Sweetwater, Tennessee, on August 10, 1970. He received the Bachelor of Science degree in Electrical Engineering from the University of Tennessee, Knoxville in May, 1992, and received the Master of Science degree in Electrical Engineering from the same university in May, 1994. His current research interests are in the fields of computer vision, robotics, and virtual reality. He is a member of Eta Kappa Nu, Tau Beta Pi, and Phi Kappa Phi.