# Wheeled Autonomous Mobile Robots for Use in Harsh Environments: A Survey of Recent Publications

by

Susan M. Larkin

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING

APPROVED:

Dr. Charles Reinholtz, Chairman

Dr. Pushkin Kachroo

Dr. William Saunders

June, 1996

Blacksburg, Virginia

**Keywords:** Autonomous Mobile Robots, Sensors, Path Planning, Navigation, Motion Control, System Architecture

# WHEELED AUTONOMOUS MOBILE ROBOTS FOR USE IN HARSH ENVIRONMENTS: A SURVEY OF RECENT PUBLICATIONS

by

Susan M. Larkin

Committee Chairman: Charles F. Reinholtz
Mechanical Engineering Department

## Abstract

Research in the area of autonomous mobile robots has increased over the last several years. Autonomous mobile robots are now being used in a wide variety of applications, including nuclear plant maintenance, interplanetary exploration, military missions and smart highway systems. This thesis is a survey of recent publications, 1990-1996, of wheeled autonomous mobile robots for harsh environments. Various sensing, navigation, and motion control strategies commonly used on autonomous mobile robots are compared. The integration of all three systems in a system architecture is also presented. Following a general discussion of autonomous mobile robot technology, vehicles that have entered the Unmanned Ground Robotics Competition are presented as a focused study of the application of this broad field of research.

# Acknowledgments

I would first like to thank my committee chairman, Dr. Charles Reinholtz for not only serving as my advisor, but for always being a source of encouragement about school, jobs and life. I also want to thank Dr. Reinholtz for agreeing to take on the autonomous vehicle project. It has truly been an enjoyable learning experience for all of us. I would also like to thank Dr. Pushkin Kachroo and Dr. William Saunders for serving on my committee and for the helpful comments.

Thanks to Greg, Steve, Tony, Randy, Dana and Paul for all the good times and conversations in the office and for making me a much stronger WVU fan than I already was! I want to thank all the members of the autonomous vehicle team. It has been a great year! I would also like to thank Scott Hudson for his help scanning pictures.

I would like to thank my fiancé, Tray, for all of his encouragement and support and for making graduate school a lot more fun. I could have never made it without you! I would like to thank my family for all of their support over the years. Dean, thanks for always putting things in perspective and keeping me humble. Cory, thanks for always being there when I needed you and for truly being my best friend. I am deeply grateful to my parents, Tony and Vesta, for all their love and encouragement. You both have taught me the truly important things in life and I love you very much!

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Over the past several years, research in the area of autonomous mobile robots has become an increasingly popular topic. Robots are now being used in a wide variety of industrial and military settings. This thesis is a survey of publications in the area of wheeled autonomous mobile robots from over the past six years, with an emphasis on robots designed for harsh environments.

This chapter starts with the definitions of an autonomous mobile robot and a harsh environment. General background information of robotics research is presented followed by the motivation for this research. Finally, an outline of the remainder of the thesis is presented.

## 1.1 What is an Autonomous Mobile Robot?

What is an autonomous mobile robot? The answer to this question will be related back to the definitions of each word. Webster's Ninth New Collegiate Dictionary lists one of the definitions of the word autonomous as "undertaken or carried on without outside control: SELF-CONTAINED". The word mobile is defined as "capable of moving or being moved: MOVABLE". One of the definitions of the word robot is "an

automatic apparatus or device that performs functions ordinarily ascribed to human beings or operates with what appears to be almost human intelligence". Therefore, using dictionary definitions, one could reason that an autonomous mobile robot is an "intelligent device that can move around, without any human interaction or outside information, and perform tasks that have been previously performed by humans". This combination of activities includes being able to sense its surroundings and make intelligent decisions.

Others authors present the same requirements for an autonomous mobile robot. Kanade et al. (1991) state that "An intelligent autonomous system must have a capable mechanism for mobility and manipulation, a sensing and perception system, and an intelligent decision-making capability, all integrated into a self-contained system." Selfridge and Franklin (1990) assert that robots should be "responsible" because we want robots to be an assistant to us. This means that the robot can make different judgments and choices in different situations. Giralt et al. (1993) describe autonomous robots as "machines endowed with the capacity to reason about a task and about its execution, by intelligently relating perception to action."

## 1.2 What is a Harsh Environment?

The title of this thesis includes the word "harsh" as a classification for environments for autonomous mobile robots. To help classify this type of environment, a survey was

completed on how recent scientific publications classified different situations. The three words commonly used to describe different environments were harsh, hostile, and hazardous. A hazardous environment is an area where it is not safe for a human. Environments referred to as hazardous include areas using nuclear radiation, environments in the deep sea, or the conditions on other planets (Harrigan, 1989; Froschle et al., 1993; Rocheleau and Crane, 1991; Sharma et al., 1991; Visinsky et al., 1993; Smith et al., 1992; Harrigan, 1993). A hostile area puts the robot in a situation that could be harmful, but danger is less severe or imminent. Hostile environments include areas with high temperature or pressure, underwater, but not deep sea, and radioactive areas with less risk than hazardous environment conditions (Fogle, 1992; Slifka et al., 1993; Rogers, 1992; Canistraro et al., 1992; Allen et al., 1991). A harsh environment is classified as one having high temperatures, pressures or loads, an area subject to adverse weather conditions or one that is difficult to traverse (Yializis, 1990; Mitchell, 1990; Bender and Zarlingo, 1992). Applying this definition to robotics, this is an environment that could cause problems for the robot but does not put it in any physical danger.

To summarize, for the purpose of this thesis, the following classifications were made based on the existing literature. A hazardous environment, such as a nuclear reactor or another planet, puts the robot in "physical" danger. Areas that are very "uncomfortable" for the robot and are possibly "life threatening" are considered hostile. Harsh environments could be difficult to traverse, such as rough terrain, but do not put the robot in any danger.

## 1.3 Background

Interest in having autonomous mobile robots perform everyday tasks is not a new idea. In science fiction, robots have been doing undesirable human work for years. One of the most notable science fiction writers in the area of robots is Isaac Asimov. He is credited with first using the term "robotics". He formulated the following three laws that all robots should obey:

1. A robot must not harm a human being, or, through inaction, allow one to come to harm.
2. A robot must always obey human beings unless that is in conflict with the First Law.
3. A robot must protect itself from harm unless that is in conflict with the first or second law.

The worlds that Asimov has created are far from a reality. In the real-world, simple tasks such as having a robot learn to follow a line can be difficult to duplicate.

There are four areas for classifying current robotic systems. The first is teleoperated robots. This is where a robot is operated with human intervention, i.e. using a joystick controller. Teach robots form a second class of robots. These robots are initially taught a sequence of motions using human intervention, but this action can be stored and repeated by the robot. At the next level are robots programmable at the task level. This class of robots does not possess a significant decision-making capability but can perform simple tasks such as "move from point A to point B along a straight line". The fourth classification of robots have advanced decision-making capabilities. These robots can

navigate in complex environments and perform functions such as "stay between the lines" or "avoid the obstacle". The robots considered in this thesis fit into the last category.

Outdoor navigation presents problems because of changing lighting conditions, variations in the form of objects, and the dynamic nature of the environment (Hampapur et al., 1993). Smart highway systems are being developed to improve the safety and comfort of driving. These systems propose to reduce the problems caused by mental stress, fatigue, misjudgments, and inattention while driving (Maruya et al., 1991). Robots capable of off-road navigation are also being studied. They must be able to navigate in unstructured conditions. Jarvis (1993) discusses systems needed to convert common, outdoor vehicles into autonomous mobile robots. He describes how a dune buggy and an electric golf cart were modified to operate autonomously.

There are still many problems to be solved in the area of autonomous robotics research. Stentz et al. (1993) list the three limitations that hinder vehicle performance. They are sensing deficiencies (the inability to measure salient terrain properties); invalid models (the inability to predict the interaction between the vehicle and the terrain); and finite computing resources (limited processing cycles for computing this interaction in real-time). These problems will continue to be investigated in the years to come.

## 1.4 Motivation

The need for a survey in the area of autonomous mobile robots was initially identified when discussing different types of autonomous mobile robot systems for a design project

at Virginia Tech. The applications for autonomous mobile robots are numerous and involve diverse areas of research. Because of this diversity, robotics research is usually focused on a small subset of the ongoing work. A survey of recent publications, the last six years, was selected because the technology in this area is progressing so quickly that earlier sources can become outdated. It would be overwhelming to include every article published in this time frame. For this survey, representative articles have been selected that span a wide variety of applications.

## 1.5 Thesis Outline

This thesis is a survey of current applications of wheeled autonomous mobile robots. The articles covered in this thesis are all current publications. This provides a good cross-section of applications that are of current interest. The chapters in this thesis describe the published work by dividing the actions of the robot into three distinct areas: perceiving, thinking, and acting.

Chapter 2 contains a discussion of sensors that are on autonomous vehicles to acquire information about the environment. This is the "perceiving" action of the robot. This discussion doesn't include sensing systems that use an external source for information, such as beacon-based or global position systems, only sensing systems completely on-board the vehicle. Chapter 3 describes the "thinking" of the robot. This includes the task of converting the sensor information into a map for the robot to use for navigation.

Chapter 4, the motion control chapter, describes how the actual movement of the robot is carried out. This is the "acting" function of the robot. The hardware systems of the robots are not described in detail because this information is available in various other sources. Jones and Flynn (1993) and Critchlow (1985) provide discussions of various hardware components on robotic vehicles. Chapter 5 describes the system architecture. This is how all the functions of the robot are coordinated to produce the desired action. Chapter 6 is a survey of publications that describe how perceiving, thinking and acting are combined into one robotic vehicle. The vehicles described are those that have entered the Unmanned Ground Robotics Competition. A summary and concluding comments are contained in Chapter 7.

# Chapter 2

# Sensors

The first task of an autonomous mobile robot is "perceiving". A robot can become familiar with its surroundings through the use of sensing devices. There are a wide variety of sensors used in robotic applications. Sensing devices range from tactile sensors, which evaluate the environment by bumping into obstacles, to ultrasonic sensors (sonar), which measure the distance between the robot and an object, to video cameras which can evaluate the entire scene at once. Although a large variety of sensors exist, they can generally be classified into one of two categories: active and passive. Active sensors emit and receive a signal that is sent into the environment, while passive sensors only observe the conditions of the scene or vehicle. This chapter will introduce the differences between active and passive sensors and describe in greater detail many of the types of sensors that are commonly used on mobile robots. Sensor fusion, which is the integration of information from several types of sensors for an accurate representation of the surroundings, will also be discussed. The last section contains tables comparing some of the advantages and disadvantages of each sensor discussed.

## 2.1 Active Sensors

Active sensors are used to determine the state of the environment by emitting a signal and using the return information for navigation decisions. Common types of active sensors are ultrasonics, laser range finders, and radar systems. Some advantages of active sensing techniques are that they are less sensitive to environmental disturbances, such as illumination and reflectivity. The major drawback of active sensors is that they are susceptible to signal interference when several sensors are used. Unfortunately, because the sensing area of these devices is often narrowly focused, multiple sensors are needed. This increases the cost of the system and creates the problem of "sensor fusion", which will be discussed later.

### 2.1.1 Ultrasonics

Ultrasonic sensors, also called sonar, are commonly used for obstacle avoidance on autonomous mobile robots. This type of sensor, shown in Figure 2.1, is used because it is relatively inexpensive and provides a good representation of the environment (Figueroa and Mahajan, 1994). Liu and Lewis (1994) explains that sonar sensors can be used to determine the location of obstacles, but not their shape. They can also be used to determine position with respect to a goal, but not absolute position.

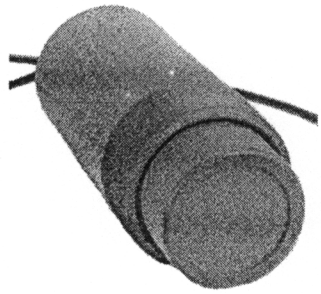

Figure 2.1: Ultrasonic Sensor

Ultrasonic sensors operate by sending a high-frequency acoustic signal and recording the time for an echo to be returned. If no echo is received, it is assumed that no obstacle is in the range of the sensor. If an echo is detected, the time of flight of the signal is used to calculate the distance to the object. This form of sensing provides a "contact-free" environmental representation. It produces good results for medium range distance measurement, 1 to 6 m (Hugli et al., 1994). Ultrasonic sensors have advantages over other sensing devices because they are cheaper than cameras or laser range finders, have a longer range for obstacle detection than most optical sensing techniques, and are not as sensitive to false signals as infrared sensors (Smith et al., 1991). Sonar can also be used effectively to scan a wide area from one reading because it has a "comparatively large beam aperture" (Lim and Cho, 1993).

The problems that arise with using ultrasonic sensors for obstacle detection come in two categories: problems with the sensor itself, and problems with the readings. Some of the criticisms of ultrasonic sensors are that they are slow, noisy, produce unreliable data,

have poor angular and directional resolution, and that the echo amplitude produced by the obstacle may be too small to be detected by the receiver (DeSaussure, et al., 1989; Sethi and Yu, 1990; Lim and Cho, 1993; Guldner et al., 1994; Kurz, 1995). Good (1992) says that poor sonar readings can be caused by four major problems: spatial uncertainty, specular reflections, high-order specular reflections, and interference. Spatial uncertainty occurs because the information that a sonar receives doesn't come from a single direction, but from a 30 degree cone with side lobes. Hence, a detected object is at a known distance, but its position within the cone is unknown. Specular reflection is when a false reading is reported from the echo bouncing off of another surface before received by the sensor. A high-order specular reflection comes from the acoustic pulse hitting three or more objects before being received. The specular reflection problems can occur in all environments, but are more likely to happen when the surfaces of the obstacles are rough, such as outdoors. Tunstel et al. (1994) compare this problem to a person in a house of mirrors where our visual perception gets confused even though is it normally very accurate. The last problem, interference, can come from two sources: noise in the environment, or noise from other ultrasonic sensors, commonly called "crosstalk".

The problems of spatial uncertainty, specular reflections and high-order specular reflections can be reduced, but not eliminated, by taking multiple readings from each sensor and using an average value to get a more accurate representation of the surroundings. Crosstalk of the sensors can often be reduced by using an appropriate "firing" strategy that separates nearby sonar from firing consecutively. Borenstein and

Koren (1995) present a approach to reduce the interference of ultrasonic sensor readings by one to two orders of magnitude called "eliminating rapid ultrasonic firing" (EERUF). The proposed method uses a comparison of consecutive readings along with alternating delays to eliminate noise from the environment and "crosstalk" between sensors.

Two features of ultrasonic sensors that affect the measurement accuracy are the frequency and the beam width. The operating frequency of the sensor is determined by the mechanical design. Since reflectivity is a function of frequency, "lower frequencies can have reduced reflections from some porous targets, while higher frequencies reflect well from most target materials" (Shirley, 1989). Ultrasonic sensor with higher frequencies also work better in an acoustically noisy setting than lower frequency sensors. The beam width of ultrasonic sensors is also a design factor. A narrow beam is less affected by background ultrasonic noise and will also operate over a longer range. The problem with narrow beams is that they are "more influenced by the angle at which the signal hits the obstacle" (Taylor and Kriegman, 1992). Wider beams will detect smaller objects, but are less accurate (Weigl et al., 1992).

Several factors play a role in the effectiveness of ultrasonic sensors. Shirley (1989) provides a good discussion of these environmental and obstacle conditions. Temperature drops cause an increase in the time for an echo to be received because the speed of sound decreases. Ultrasonic sensors need to be shielded around the back and sides to prevent radio frequency interference. Special moisture-resistant sensors should be used in wet environments The target that the sensor can detect depends on the composition, shape,

and orientation to the sensor; some textures will produce a weaker echo. A target of virtually any shape can be detected ultrasonically if a sufficient echo returns to the sensor. Targets that are smooth, flat, and perpendicular to the sensor's beam produce stronger echoes than irregularly shaped targets. A larger target will produce a stronger echo than a smaller target.

Sonar-based sensors should continue to be popular in robotic applications. Their low cost, accurate environmental representation, and availability make them a good choice for navigation.

### 2.1.2 Laser Range Finders

Laser range finders are another popular sensing device used on autonomous mobile robots. An example is shown in Figure 2.2. The two methods employed by laser range finders for obstacle detection are triangulation and time of flight. Triangulation uses geometry to measure the distance to an obstacle by using the relationships of the other known parameters. Using two sensors that are at fixed angles and known distance apart to observe an object, the distance of each sensor to the object can be determined by geometry. This is shown in Figure 2.3. The location of the sensors is known, and angles $\alpha$ and $\beta$ are the known rotations needed to point the lasers at the target. The distance, l, is also known. The distance to the obstacle from each sensor can then be calculated. Triangulation is used by Hugli et al. (1994) to determine the geometry of obstacles.

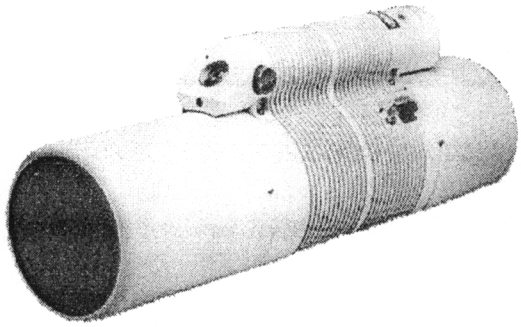

Figure 2.2: Laser Range Finder

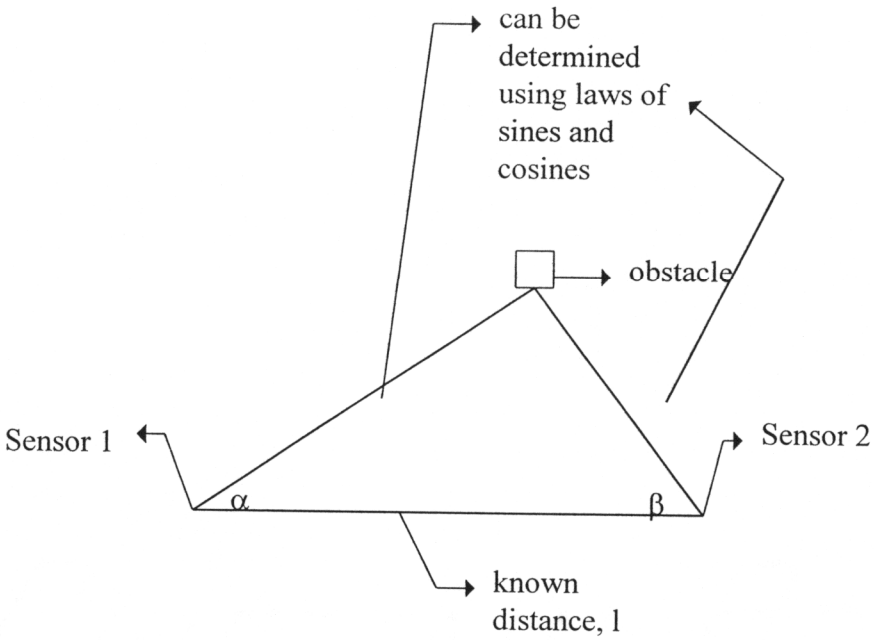


Figure 2.3: Triangulation

Lux and Schaefer (1991) use a laser range finder based on time of flight for distance measurement. The time between the emitted pulse and the received pulse gives a unique distance estimate. The beam is narrow and gives a good estimate of the distance to the obstacle. Time-of-flight laser range finders differ from sonar in the type and frequency of the transmitted wave.

Some of the advantages of using a laser range scanner over ultrasonics are high resolution, excellent accuracy, and relatively long measurement range (Critchlow, 1985; Liu and Lewis, 1994). Laser range finders also have several advantages over other light intensity sensors and cameras. The images produced are easier to process than ones from monochrome video cameras and they are insensitive to illumination, making modeling of an object easier. Many of the calibration problems and the computational burden associated with computer vision are avoided because geometry is directly measured. Also, laser range finders do not require matchable features in both images like stereo vision (Lux and Schaefer, 1991; Kelly et al., 1993; Thorpe et al., 1991; Ikegami et al., 1990).

The main drawbacks with laser range finders are their high cost and slow response time (Figueroa and Mahajan, 1994; Weigl et al., 1993). Because these sensors emit a laser beam, they could also introduce an environment that is unsafe for people (Weigl et al., 1993; Salem and El-Khamy, 1995). Laser range scanners also have difficulty in an environment with metallic surfaces because light can be reflected in different directions (Salem and El-Khamy, 1995). Although a laser range finder has advantages over using a

video camera, its environmental representation is lower in resolution and is sensitive to noise (Yang and Wu, 1995). For these reasons, Yang and Wu (1995) suggest that range finders can not be used on mobile robots as the only sensing system. This may be true for some applications, but research shows that several mobile robots have been able to navigate without the use of other sensors.

Laser range scanners are often coupled with video cameras to get an accurate three-dimensional environmental representation. Morgenthaler et al. (1990) state that laser range sensors have been used in systems designed to detect and avoid on-road obstacles and to classify terrain for off-road navigation. Xie et al. (1994) use a laser range finder that has two mirrors to control the scanning directions in the X and Y directions. The orientations of the mirrors can be dynamically controlled by an application program. The time-of-flight measurement is used to determine depth. They obtained a maximum sensing range of 85 feet with this system.

A commonly used laser range scanner has been developed by the Environmental Research Institute of Michigan (ERIM). This sensor was developed for outdoor autonomous mobile robot navigation applications (Gowdy et al., 1990). There are three main problems when using this sensor in an outdoor terrain: sensor limitations (problems with the data due to the laser range finder itself), rough terrain issues (problems due to the shape of the terrain, including occlusions and undersampling), and problems because of the speed of the vehicle (moving while taking images and looking ahead a great distance) (Stentz et al., 1993). Stentz et al. (1993) and Kelly et al. (1993) provide an excellent

discussion of these problems and the proposed solutions. A summary of these problems and solutions follows.

The first problem, sensor limitations, can be further divided into problems of ambiguity, secularity, mixed pixels, divergence, and accuracy. Ambiguity occurs since the ERIM scanner produces a pixel reading that could be in an infinite sequence of possible ranges spaced apart by the wavelength of the signal. Specular surfaces (such as wet asphalt or water puddles) do not reflect enough of the signal back to the sensor to register a range reading. Two problems are created since the laser beam is being scanned. The first is a mixed pixel. This occurs when the beam strikes the edge of an object and the object reflects part of the signal and the background reflects another part. This leads to an intermediate value which is neither the distance to the object nor to the background. The second problem is angular resolution. At larger distances, the spread in the signal makes it difficult to detect sharp edges.

Rough terrain causes problems with the imaging mechanics, sensor mounting angle, terrain shape, oversampling, undersampling, and range shadows. The best perspective for viewing the terrain is overhead, but this may be impossible on a mobile robot. Even though the field of view is evenly sampled (in pan and tilt angles), uneven sampling of the ground can occur in a rough terrain. Oversampling means that more measurements are taken than needed to identify the object. This occurs most often with protruding objects, such as trees and rocks. Gaps in the data can also be present. This could cause a surface that is tilted away from the sensor, such as small hills and bumps, to be

undersampled. Some terrain regions could be entirely hidden from the sensor, such as the interiors of potholes and the ground behind trees and other objects.

Scanning time, latency, vertical field of view, and throughput are problems that arise from higher speeds of the vehicle. Higher speeds reduce the amount of time available to process the data into a useful representation before the robot must make a decision. If an obstacle is detected, a sufficient distance to bring the vehicle safely to a stop must be maintained. The limited vertical field of view (30 degrees for the ERIM) means that the entire area from the vehicle's location cannot be measured in a single image. Thus, previous images must be retained to evaluate the terrain. Inaccuracies in measuring the vehicle's position and orientation can lead to errors between successive images, resulting in false range discontinuities in the data. Also, the time required for image digitization can cause objects in the image to appear compressed unless the data is corrected to account for vehicle motion.

The perception systems used by Stentz et al. (1992) and Kelley et al. (1992) propose solutions to help eliminate the effects of some of these problems. The ambiguity is removed by searching the image and looking for an abrupt reduction in range. The data beyond this interval is considered inaccurate and is discarded. To remove bad data caused by specular reflections or noise, a filter is applied. To correct the problems caused by the terrain and vehicle speed, a method that samples the vehicle's position at regular intervals during digitization to remove motion distortion is proposed. The pixels are then grouped and the maximum and minimum elevation values are retained. This reduces

oversampled areas to a uniform covering in the ground plane. Empty cells correspond to undersampled or shadowed regions on the terrain. Small runs of empty cells are linearly interpolated, while larger runs are marked as range shadows.

Laser range finders provide a good representation of the surroundings but do so at a considerable cost. Laser range finders are used often, and work well, when used in conjunction with cameras.

### 2.1.3 Optical Proximity Sensor

An optical proximity sensor is a good way to detect if an object is in the way of a mobile robot. An example of this type of sensor is shown in Figure 2.4. This sensing method involves detecting light and transmitting it into a signal that can be interpreted for obstacle detection. This is usually accomplished by measuring the differences in the light intensity as it is transmitted and reflected. An arrangement that uses a light-emitting diode (LED) as its light source and a photosensitive cell is described in Critchlow (1985). A lens is used to project the light source at a point where the light will be reflected from an object back through another lens to the photosensor.

This sensing system has the advantage of being low cost, so several sensors focused at different ranges could be used for better obstacle detection (Critchlow, 1985). The major drawback of this system is that no range information is determined, only the presence or absence of an obstacle is reported.

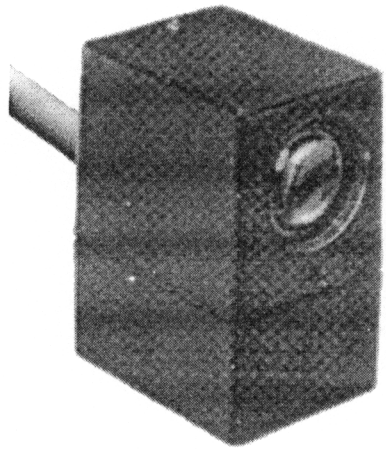

Figure 2.4: Photosensor

### 2.1.4 Structured Light

Structured light is a system composed of a light source and a camera used for active sensing. This sensing technique is a good method for obtaining range and depth information when used on a mobile robot. Structured light operates by projecting a light pattern on the scene to be viewed. These light patterns are regular patterns, such as bars, grids, or circles and are usually projected at a 30 to 45 degree angle from the vertical to the scene. The displacement from the front edge to the rear edge of the object is perceived. The depth and range information is obtained by measuring the displacements along the projected beam. "The projected light pattern may be generated by a scanning laser, so that the location is controlled accurately, or by a rotating slit scanner, or more simply by a conventional slide projector. Greater displacement is observed when the camera is farther from the object or when the displacement angle is increased. However,

there is a loss of light intensity at greater distances and possible obscuring of parts of the image if the displacement angle is too large" (Critchlow, 1985). With the right setup, structured light can produce very accurate information with a wide field of view and a long depth measurement (Pears and Probert, 1993). Lasers are commonly used as the light source because they can provide very intense light bars. This is helpful in many applications because it produces more reliable measurements (Critchlow, 1985).

Santamaria and Arkin (1995) list some of the disadvantages of structured light in their applications. The major problems they had were that the system was very sensitive to noise and lighting conditions. As with most laser based sensors, specular reflection was also a problem. Some of the noise and lighting problems were reduced, but not eliminated by using a filter. The specular reflection problems can also be reduced by using averaging techniques. An algorithm to help with the positioning and misalignment problems was also presented.

Pears and Probert (1993) use structured light for obstacle detection on a mobile robot. They use a laser diode for the light source and a lateral-effect photodiode (LEP) for image position measurement. They chose the LEP over a CCD because it is less expensive, easier to use, and provides better position resolution at shorter ranges. The disadvantage is that a low-noise circuit design is needed

A grid encoded structured light computer vision method is described in Yeung et al. (1993). This system is used on a remote control mobile robot for three-dimensional object feature recognition. The results are interpreted by a person operating the vehicle.

A stroboscopic projector is used to project a rectangular grid on the objects. From the way the grid deforms, enough visual information is available for a person to deduce the position and orientation information of the illuminated objects.

Another approach using structured light used by Santamaria and Arkin (1995) helps to recognize irregularities on the surface of drums. The systems they discuss projects the laser stripes either horizontally or vertically. The horizontal setup consists of a laser and a camera to produce straight lines in the image plane, while the vertical scan produces conic curves on the camera plane and uses two lasers and one camera. Stuck et al. (1994) use a structured light system that consists of two components: a sensor head, which holds a CCD camera with a modified lens incorporating a dual-aperture iris and a laser projector that projects a vertical stripe of light. This setup enables the range finder to measure the range and angular position data of an object that is being illuminated by this stripe.

Structured light is used in a variety of applications. It is an effective way to acquire good depth information, but, as with most light-based sensing systems, at a considerable expense.

### 2.1.5 Infrared Sensors

The use of infrared sensors is another popular sensing technique used in mobile robots. The accuracy of this type of sensor is usually proportional to the cost and the

usefulness varies with application. An infrared camera is shown in Figure 2.5. Infrared sensors operate by measuring the light emitted by an infrared diode and reflected back by the environment. Infrared sensors are typically used for close-range distance measurements, 0.3-1.2 m (Hugli et al., 1994). Although this short range is typical, several infrared arrangements have been developed to achieve a longer scanning range.

Figure 2.5: Infrared Camera

Infrared detectors can be used for road navigation to detect "hot objects", that could be in blind spots. This includes objects such as bike pedals that "must be identified from background clutter" (Abbott et al., 1995). Infrared images have also been used by Matthies et al. (1995) for terrain classification. They use near infrared technology to determine the differences between soil and vegetation areas. This can be accomplished because vegetation is very reflective in the near infrared. An infrared range scanner has been developed at Oxford University that has a higher data acquisition rate, an extended

operational range, and non specular reflection characteristics (Borthwick and Durrant-Whyte, 1994). Yeung et al. (1993) use a infrared ranging system consisting of eight sensors arranged to give a 90 degree field of view. The individual fields of view of several of the sensors were designed to intersect to get a better coverage of the environment.

An infrared range scanner is used on the mobile robot discussed by Vandorpe and Van Brussel (1994) to provide a complete view of the environment. An infrared scanner was selected over ultrasonic sensors because is can get a better picture of the environment with a fast scanning speed and has a narrow beam.

One of the biggest problems with infrared sensors is that their reliability is highly dependent on the reflectivity of the targets (Tunstel and Jamshidi, 1994). Another drawback is the loss of accuracy of the infrared sensor if dust particles are in the air. Kachroo (1995) also says that "infrared sensors typically require that the reflectivity of the target remain approximately constant. Specular or reflective surfaces in the beam of the sensor can potentially cause an erroneous reading. Also, the output is not typically linear. Care needs to be taken to set up the sensor such that its optimum range coincides with a linear portion of its response curve."

Korba et al. (1994) present an survey of several light-based ranging systems. A comparison is made of three commercial infrared sensors and two camera systems with infrared auto-focus technology. The commercial sensors varied in cost from $350 to $4000. The sensors also varied in sensing range, proportional to price, from 10 cm to

13,000 cm. The advantages of the commercial infrared sensors is that different colors didn't pose a problem. They also have a large volume of coverage, but they have very slow response times and supply little information regarding the actual distance of a target within the sensing volume. The camera systems were tested by viewing the light emanating from the auto-focus unit of the camera (using a hand-held infrared viewer). The cameras cost $200 and $300. They both used triangulation methods to determine the distance to a target. The effective beam width for the cameras was around 6. These systems have a narrow field of view and are less affected by target reflectivity than the commercial infrared sensors.

Infrared sensors can be useful in applications needing only a short sensing range. Longer ranges can be achieved, but this greatly increases the cost of the sensing system.

### 2.1.6 Radar

Radar systems are also being used on autonomous mobile robots. They are used as an imaging device to aid in collision avoidance, path-planning, identification, mapping, and object location (Salem and El-Khamy, 1995; Stentz et al., 1993). Radar systems observe the distance to an object by determining the time-of-flight of radio waves reflected from its surface (El-Konyaly et al., 1995b). This is usually accomplished by a pulse radar system that uses a single directional antenna to transmit and receive the waves. Radar systems transmit pulses of electromagnetic waves (usually microwaves), which are

reflected by objects in the path of the beam. Reflections are received by the radar unit, processed electronically, and converted into images. The location of the object is determined from the direction the pulses were transmitted. The radar system is also able to determine the intensity of the reflected signal. This gives an idea of the physical properties of the object's surface (El-Konyaly et al., 1995).

Salem and El-Khamy (1995) list four attributes of a radar sensory system used for object identification on an intelligent vehicle. First, the system is able to obtain data from all directions, either by large beam widths or scanning a narrow beam, without focusing on any one direction. Second, complicated objects are able to be identified without giving preference to a certain surface type. Next, noise can be dealt with in the system without invalidating some of the data. Radar systems are also fast and accurate.

The major drawback to radar systems is their inability to detect boundaries. (Saneyoshi, 1994). Radar systems are useful for object identification, but are highly dependent on the reflectivity of the surface of the object. The types of radar systems that are usually used on mobile robots operate at microwave or millimeter frequencies.

El-Konyaly et al. (1995) states that an advantage of radar operating at microwave frequencies is that the narrow beamwidths needed for range accuracy in cluttered environments can be easily obtained. Since the radar beams are usually scanned, this system can also produce a high resolution representation in any environment. Operating in the microwave frequency also makes the sensor less affected by noise. Salem and El-

Khamy (1995) note that radar can be used in long range (high power) and short range (low power) situations to produce good accuracy and resolution.

Millimeter wave radar systems are also used on mobile robots. This radar system operates on the higher end of the frequency spectrum. It presents many of the same characteristics as microwave radar systems, but has an advantage over microwaves because smaller bandwidths can be obtained and a smaller antenna can be used (Button and Wiltse, 1981). This makes is a very useful system for obstacle detection. Stentz et al. (1993) suggest the use of millimeter wave radar for navigation in an area covered with vegetation to see terrain irregularities. Some of the disadvantages of millimeter wave radar are the high cost, low reliability and availability, short operation range, and sensitivity to inclimate weather (Button and Wiltse, 1981).

Although mostly used for obstacle detection, radar can be used for other measurements. A ground speed radar system is used by Koskinen et al. (1993) on an outdoor mobile robot to provide a "contact-less" speed measurement. Such a system can be used to detect situations where wheels are slipping.

Radar provides a good form of sensing for obstacle detection. Although suffering from high cost, it gives accurate information with good resolution.

## 2.2 Passive Sensors

Passive sensors are non-emissive sensors that observe the state of the environment or the vehicle for navigation. The main advantage of passive sensing techniques are that no signals are emitted into the surroundings, so the presence of other sensors does not have a negative effect on the system. Passive sensing can also be preferable to active sensing in areas where people are present "because it is not desirable to shoot laser beams or ultra-sonic rays into the public domain" (Sung, 1990).

The passive sensors that are discussed fall in two categories: sensors used to map the environment and ones used to determine the conditions of the robot. The first type include video cameras, which take pictures of the entire scene, and tactile sensors used to aid in navigation by actually touching the obstacles. Sensors used to evaluate the "state" of the vehicle include encoders for distance measurement and tilt sensors to determine if the robot is navigating over an uneven area. A more detailed discussion of both types of passive sensors follows.

### 2.2.1 Cameras

Computer vision systems have become a common sensing technique on autonomous mobile robots. A typical small, commonly used camera is shown in Figure 2.6. Computer vision can be used for a variety of navigation tasks, including obstacle detection and avoidance and boundary line detection. The wide variety of sensing

applications that computer vision provides makes it an excellent choice for navigation. Heikkonen and Oja (1993) state that the main goals of robot vision system are to "gather visual information in front of the vehicle and to provide a robust description of the scenes." This description of the scenes includes information about the locations and types of objects detected and road boundary lines used for road-following.

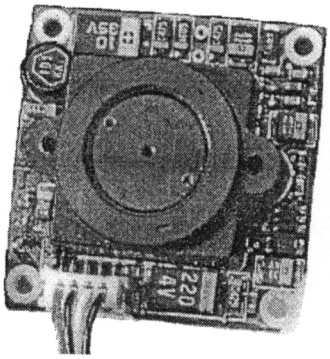

Figure 2.6: Camera

Passive vision is sensing that involves a video camera and some form of video processor for basic image processing and recognition (Hugli et al., 1994). The process first involves capturing an image using a frame grabber board on a computer. After the image is acquired, it is typically preprocessed to smooth the picture and eliminate any spurious data points. Next, if the surroundings were previously known, the image can be matched with a stored image. If navigating in an unknown environment, the scene information is converted into a map describing the location of obstacles or boundary lines (Ikegami et al., 1990)

Computer vision is an excellent sensor for navigation, but may be the most difficult of all commonly used sensors to implement (Heikkonen and Oja, 1993). Some of the advantages of video technology are the low cost and power consumption and a high degree of mechanical reliability. Cameras also have a high resolution, give a good environmental representation and are not affected by outside noise (Yang and Wu, 1995). Also, because of the passive nature of the camera, it does not produce any signals and the signals emitted from other sensors usually do not interfere with the camera data (Badal et al., 1994). Information about surface markings on objects can also be detected with video images, instead of just depth information (Taylor and Kriegman, 1992). This is useful for object identification.

The biggest disadvantage of vision systems is the large computational requirement associated with their use (Figueroa and Mahajan, 1994). Roberts and Bhanu (1992) also note that state-of-the-art motion analysis techniques for obstacle detection are not robust and reliable enough for many practical applications because unrealistic constraints must be placed on the input data to make them work. Problems with vision systems come from unknown camera motion and incomplete or unclear data. Vision systems present additional problems when used outdoors. Conditions vary and are typically less controllable outside as compared to inside. Additional problems arise with lighting and weather conditions (Huber and Graefe, 1994; Ninomiya et al., 1995; Yokoyama et al., 1993).

The camera information is presented in two separate sections: monocular and stereo vision. Monocular vision uses one or more cameras on the vehicle for unique information from each source, while stereo vision uses two or more cameras to compare the same features.

#### 2.2.1.1 Monocular Vision

Monocular vision can be accomplished with any number of cameras using color, black and white images, or a combination of both. Color vision techniques are becoming more popular on mobile robots. One reason that single-camera color vision is used on mobile robots is because is can be used to classify road and non-road areas fairly quickly (Sung, 1990). Color is also useful for navigation on natural terrain because different areas are able to be identified. This has an advantage over purely intensity-based black and white methods because a global threshold value does not have to be assigned to the entire image (Olin and Tseng, 1991).

Zhang et al. (1993) state that the data from color images must be compressed for processing because of the large amount of information captured in a single color video frame. They use a weighting method to reduce the number of colors used in the image analysis. It is important to compress the image to a reasonable number of colors. If too few colors are used, the areas do not produce clear enough results. When too many colors are used, image processing becomes very time-consuming.

The main weakness of single-camera color images is that three-dimensional data is difficult to reproduce from color vision. A single color camera usually can not be used as the only sensor for navigation. A range-finding device is desirable for obstacle avoidance (Yang and Wu, 1995; Sung, 1990). Using a two-dimensional representation of a three-dimensional environment causes problems when assumptions are made about the world from the lost three dimensional data (Boshra and Zhang, 1994b). Other problems arise because color systems can become confused by shadows, vehicle tracks, tire marks and incomplete road boundaries. Problems may also occur if there are sudden changes in the texture or color of the terrain (Sung, 1990).

The autonomous vehicle applications for single-camera direct vision systems cover a wide range of navigation problems. Morgenthaler et al. (1990) use a video camera to capture images and color segmentation techniques to determine the edge of the road. The locations of the edge of the road is processed by an "inverse perspective algorithm" to transform the data to three-dimensional coordinates. This information is used by the navigation system to plan the path along the road. A singles-color camera used on board a robotic vehicle is described by Crisman and Webb (1991). Every pixel in the image is classified as road or off-road depending on how the intensity of the pixel compares to previous road and off-road images. A more complex vision system is used by Fernandez et al. (1994) for navigation in a mountainous terrain. The main tasks of the vision system are broken into three areas: Image segmentation -classifying the image into either road, shoulder, or obstacle categories), Region classification -a model is built from the

segmented images and Obstacle and Way limits location - this determines the location of obstacles in the picture. This is an example of a vision system used for both obstacle avoidance and boundary recognition.

Ringach and Baram (1994) present a vision approach for obstacle detection based on size change information. By comparing the size and texture changes of obstacles in two successive images, it is possible to tell if the object is getting closer to the vehicle. This technique is based on the principle that "when an observer approaches a textured object, the object's image, defined by its projection onto a small sphere (retina, camera sensor) centered at the observer's location, grows larger and the granularity of its surface texture grows coarser."

A vision system proposed by Thorpe et al. (1991) uses an adaptive color classification system to eliminate many of the problems with color imagery. Color systems that use thresholding techniques with color bands that label pixels with similar intensities in the same classes have problems in shaded areas. Their method minimizes these problems by using multiple color classes. Four color classes are used for road description, and four additional color classes are used to describe off-road objects. "Classified pixels vote for all road locations that would contain them, with votes weighted by classification confidence". The area that is classified as the road is used for steering decisions.

An excellent discussion of various computer vision techniques is presented in Simpson (1991). The research activities of the Strategic Computing Initiative program are discussed. Three different areas of vision were examined: road following, cross-

country navigation, and a combination of the two in a single system to accomplish specific mission goals. The road-following research discussed involves a vehicle driving down a stretch of paved or unpaved road. The issues of outdoor illumination, shadows, the recognition of ground features (such as road boundaries), and the avoidance of obstacles using three-dimensional sensor data have been topics for research. Road following techniques are easier to develop, because it is a relatively well-defined task with clearly defined features of interest. The cross-country navigation research involves driving autonomous vehicles across natural terrain. Navigation must be planned to a goal point that can not be identified from the starting point. This type of navigation requires the robot to map the environment based on sensor input. The system developed classified terrain areas as traversable or untraversable. Complex environments were able to be navigated, but at slow speeds. The last area integrated road following and cross-country navigation. This requires the system to navigate both on and off-road as well as avoid and recognize objects. In this situation, the robot must continually decide the appropriate navigation and sensing procedures to be used.

The vision methods already discussed assume the camera to be at a fixed, known position. One-camera systems where the camera rotates are also being used. This system enables the robot to better perceive the surroundings and is a useful tool for map making. When the camera makes a circular scan of the environment at a constant rate this is called an omni-directional view. Ishiguro et al. (1993b) and Ishiguro et al. (1992) describe

using this type of camera information. This method does not require calibration of as many parameters as stereo vision.

Another interesting approach to direct vision is to use two different cameras, but each with different properties. This enables the vision system to detect different properties for boundary lines and obstacles. Dickmanns et al. (1990) use two cameras with different focal lengths on a robotic vehicle testbed. This enables the system to analyze the surroundings using a wide-angle image for global features like the road boundaries. More detail in an enlarged image helps to identify obstacles. A similar two-camera method is used by Manigel and Leonhard (1992). One camera has a telephoto lens for detecting obstacles. The other camera is equipped with a wide-angle lens for lane detection. Sugiyama et al. (1994) use two cameras for position estimation by taking pictures of landmarks. They use one monochrome camera to take pictures of fluorescent lamps and a color camera to take pictures of man-made color patterns.

Three black-and-white cameras are used by DeSaussure et al. (1989) on a mobile robot. One of the cameras is equipped with a wide-angle lens. This camera is used to scan the surroundings and navigate to the goal. The other cameras are used to determine a pre-determined location from known information. The disadvantage to this multiple camera vision system is that it does not give depth information and it requires a great deal of computer processing.

Four scanning cameras are used by Ishiguro et al. (1993a) to observe the environment using a technique they call "multiple vision agents". Each vision agent controls the

motion of one camera independently from the other cameras. Each agent also has its own computing power. Each camera is independent in the sense that it tracks a different area, tasks are determined by checking the status of the other cameras. Another four-camera vision configuration is discussed by Schneider and Wolf (1994). The main driving camera is mounted on a pan-tilt head such that orientation can be adjusted while the vehicle is stationary. It can also be used while the vehicle is in motion to determine the best viewing angle. Two other cameras are mounted on the outside of the vehicle for navigating in congested areas or through narrow lanes. The fourth camera provides a rear view and is used for backing up.

#### 2.2.1.2 Stereo Vision

Stereo vision systems are an increasingly popular method for mobile robot navigation. This sensing technique is widely used because it is possible to obtain three-dimensional information about the environment. Stereo vision can reproduce the terrain by matching corresponding pixels from two or more cameras looking at the scene from different perspectives (Stentz et al., 1993; Maravall et al., 1991). Funkunga et al. (1991) describe stereo vision as a method "used to obtain depth information and further three-dimensional location and shape information by the principle of triangulation using the difference between two images obtained in different locations." Using a stereo vision method to get a three-dimensional representation of the surroundings will give better results than combining data from a single camera and a range finder. It is much simpler for

navigation to extract three-dimensional information from a three-dimensional image (Sung, 1990).

The biggest advantage of stereo vision is that it can be used to directly sensing terrain or obstacles. This can be accomplished using two or more cameras that are non-emissive, non-scanning, and non-mechanical (Matthies et al., 1995). Stereo vision is also useful for outdoor navigation because the is able to distinguish uneven terrain (Sung, 1990).

Stereo vision suffers from difficulties correlating one camera image to others, decreasing accuracy with increasing range, possible uneven sampling of the terrain, difficulty in ranging bland surfaces, and reliance on ambient lighting (Stentz et al., 1993; Funkunga et al., 1991; Santos-Victor et al., 1995; Thorpe et al., 1991). The biggest problem with stereo vision applications is the large amount of time or computing power required for image processing (Liu and Lewis, 1994; Maravall et al., 1991) This limits the use of stereo vision techniques in real-world applications because of its difficulty for real-time navigation control (Badal et al., 1994).

Stereo vision systems may have any number of cameras. A one-camera motion stereo method is used by Huber and Graefe (1994). A scene is sampled quickly and measures the distance to objects in real-time by comparing of the features in successive image. This method is very accurate and can be used with an uncalibrated camera. This method has an advantage over two-camera systems because with the two-camera method it is difficult to determine the correspondence between points and pixels in each image. The placement of the cameras can also become a problem. "The optimal distance between the

cameras depends on the distance to be measured; if the cameras are too far apart the correspondence problem becomes even more difficult, but if they are too close together the accuracy of the measurement turns poor. On any mobile system the distances to be measured vary greatly, making it impossible to select a single optimal camera configuration" (Huber and Graefe, 1994).

Santos-Victor et al. (1993) use a divergent stereo approach, which is based on the way that insects view the world. This system uses two cameras pointed laterally (like the eyes of an insect) with a non overlapping field of view. The images from the left and right cameras are acquired at the same time and adjusts the speed of the robot based on the distance to the object. The robot moves faster around objects that are perceived as closer. This "reaction time" of the robot changes when the identified "danger" of the obstacle changes.

A three-camera stereo vision system was used on board an autonomous vehicle by Morita et al. (1993) to identify boundary lines of the road and to detect other vehicles. Sung (1990) also use a three-camera stereo vision system. The two outer cameras are black and white and the middle camera is color. This helps distinguish the road from other objects. Color information is used to supplement the black and white information provides a excellent source for environment reconstruction.

### 2.2.2 Tactile Sensors

Tactile sensors are a passive device that are used by mobile robots to actually "feel" their way around the environment. Tactile sensors are based on direct measurement using displacement or force transducers. That means that the sensor measures the "mechanical deformation [caused] by a force acting directly on the transducer itself" (Ruocco, 1987) This type of sensing is only acceptable if it is used in a situation where it is safe for the robot to collide with obstacles when mapping the environment. The advantages of tactile sensors are their low cost and ease of implementation. They are able to detect objects in an unknown environment where vision is obstructed and they are useful in determining the location of a robot in a known environment (Boshra and Zhang, 1994a; Malik, 1991).

### 2.2.3 Encoders

Autonomous mobile robots often use encoders to aid in position estimation. These sensors are typically mounted on the output shaft of a drive motor or axle. Shaft encoders are either absolute and incremental. With an absolute encoder, the signal produced by the sensor is a binary code that corresponds to a particular absolute orientation of the shaft (Jones and Flynn, 1993). Incremental encoders use optical sensors to "pick up on/off pulses as the relative motion takes place" (Beckwith et al., 1993). Incremental encoders provide relative, or displacement information only. An incremental encoder is shown in Figure 2.7.

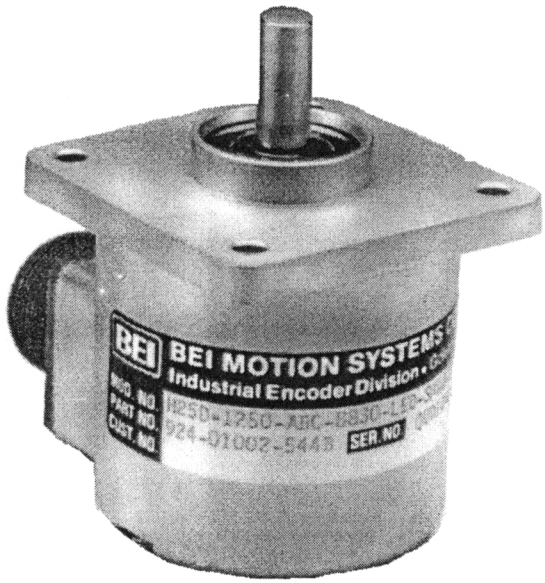


Figure 2.7: Optical Encoder

### 2.2.4 Odometry

Odometry is a general term applied to any method that indicates the distance traveled by the vehicle. Most odometry systems on robots use encoders to measure wheel revolution (Borenstein and Feng, 1995a). Baxter and Bumby (1995) use a combination of shaft encoders and infra-red sensors for distance measurement. Odometry systems are independent of external sensors, simple, inexpensive, and the feedback can generally be incorporated in real-time. A drawback is that these systems accumulate error very rapidly due to wheel slippage and mechanical tolerances (Borenstein and Feng, 1995; Baxter and Bumby, 1995). Baxter and Bumby (1995) states that odometry also presents inaccurate

information when a the robot navigates over an uneven surface because the rotation of the wheels do not accurately coincide with the straight-line distance traveled.

### 2.2.5 Potentiometers

A potentiometer, shown in Figure 2.8, is a sensor used for position measurement. Potentiometers are the most straightforward form of position measurement because each position of the shaft produces a unique resistance and voltage output (Jones and Flynn, 1993). Potentiometers are essentially mechanically variable resistors used to control signal currents and small control currents. Potentiometers are an inexpensive way to measure position accurately but can suffer from poor resolution, non-linearity, and sensitivity to noise (Craig, 1989; Critchlow, 1985). The greatest disadvantage of potentiometers is that they are only able to rotate through a limited number of revolutions before hitting a hard stop.

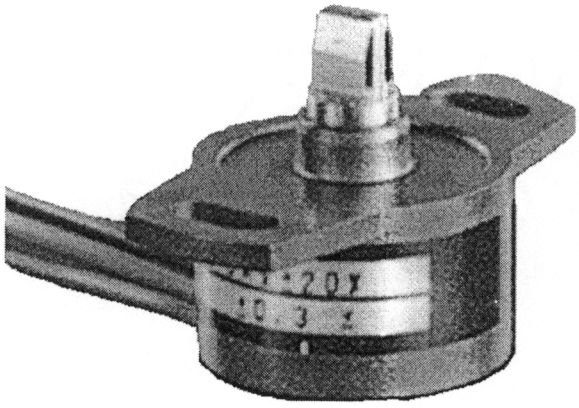

Figure 2.8: Potentiometer

### 2.2.6 Altimeters

Altimeters are used to measure the altitude of the robot. An example of an altimeter is shown in Figure 2.9. One type of altimeter, an aneroid altimeter, measures the difference in a pre-set and the current air pressure to determine the current altitude.


Figure 2.9: Altimeter

### 2.2.7 Compass

A compass can be used on a mobile robot to give the orientation of a vehicle relative to the earth's magnetic field. Problems arise when using a compass, especially for indoor applications. "Magnetic fields from electrical wiring, structural steel in buildings, and even the metal components of the robot itself can all produce large errors in the compass reading" (Jones and Flynn, 1993). A compass is able to provide orientation information at a reasonable cost.

### 2.2.8 Tilt Sensor

A tilt sensor can be used to determine the orientation of a vehicle relative to the gravity field. It is able to determine if the vehicle is navigating over an incline. A common sensor of this type is the electrolytic-tilt sensor, shown in Figure 2.10. "This sensor has two or more electrodes immersed in a conductive fluid. The conduction (resistance )between the electrodes is a function of the orientation of the sensor relative to gravity.

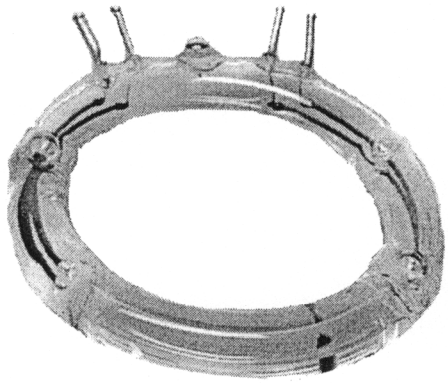

Figure 2.10: Tilt Sensor

### 2.2.9 Accelerometers

Accelerometers, such as those shown in Figure 2.11, are devices that can be used to measure the acceleration of the robot. Some accelerometers produce an analog signal proportional to the rate of acceleration. Others are digital devices that indicate when a preset acceleration rate is exceeded. Beckwith et al. (1993) give an example of an

accelerometer that uses a preloaded electrical contact. It works, "in theory, when the effect of the inertia forces acting on the spring and mass exceed the preload setting, contact will be broken, and this action may then be used to trip some form of the indicator."

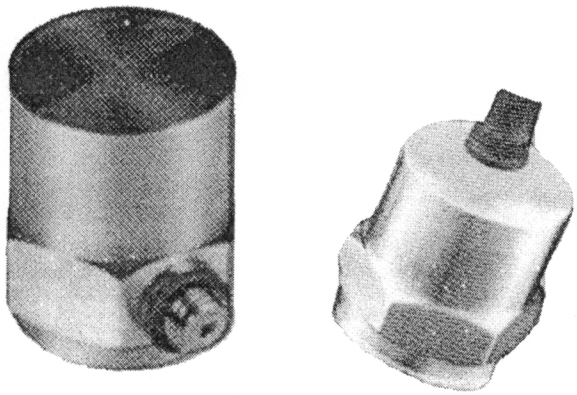

Figure 2.11: Accelerometers

### 2.2.10 Gyroscopes

A gyroscope-based sensor can be used to give an estimate of orientation (Xu, 1991). "Mechanical gyroscopes use the principle of conservation or angular momentum to keep one or more internal axes pointed in the same direction as the exterior of the gyroscope, the gyroscope case, translates and rotates" (Jones and Flynn, 1993). This sensing device helps determine the rotation of a mobile robot relative to a fixed coordinate system. An example of a gyroscope is shown in Figure 2.12.

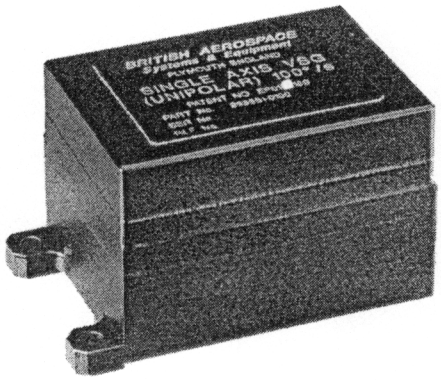


Figure 2.12: Gyroscope

### 2.2.11 Tachometers

Tachometers are used on mobile robots as velocity sensors. Shown in Figure 2.13, these sensors are small voltage generators that provide an analog signal proportional to the shaft velocity (Craig, 1989; Critchlow, 1985). These are simple and inexpensive sensors to implement.

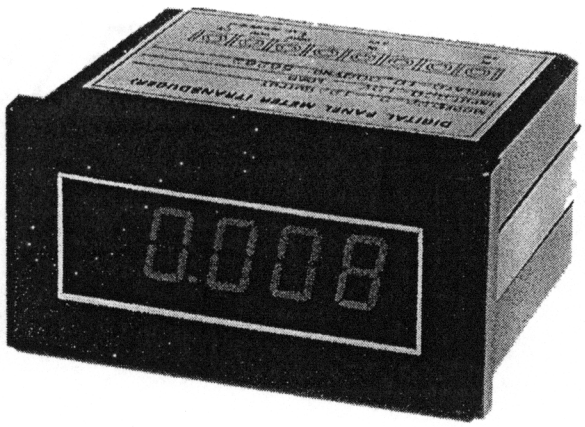


Figure 2.13: Tachometer

## 2.3 Sensor Fusion

For autonomous mobile robots, information from a single sensor is usually insufficient to obtain the information necessary for navigation. Using only one sensor limits navigation to the precision and accuracy of that single sensor. This information is usually incomplete and uncertain (Xu, 1991). Data from several sensors can be "fused" together to provide information that is more accurate and reliable than can be obtained from a single sensor. Sensor fusion combines data taken from a set of sensors into a form that can be used by the robot to produce a desired robotic action (Murphy, 1990).

Xu (1991) classifies the information obtained from various sensors into three classes: complete, partial, and indirect. A complete estimate provides a full description of location or relationship. An encoder is an example of a complete sensor because just the encoder information can be used for dead reckoning. A sensor that gives a partial estimate can not give a complete description, but gives some details. For example, a gyroscope can be used to obtain orientation information, but not position. Indirect sensors can not give specific location information, but can be used to improve the information given from other sensors. For example, a tilt sensor can not be use alone to identify where the robot is but can determine that its location has a certain incline. The right combination of all three types of sensors is often most effective for robot navigation.

Kadonoff and Parish (1995) recommend that sensing for each property be performed with at least two different sensors. At least one of the two sensors for each measurement

should be external; such as ultrasonics, laser range finders, and cameras. This helps reduce errors that can arise with internal sensor measurements. Even though they accumulate errors, internal sensors, such as, gyroscopes, accelerometers, inclinometers, are still used to establish robot position because they can produce a fast, dynamic measurement largely independent of noise and interference from other sensors.

The control scheme for sensor fusion defines the way that the sensory information will be processed. The criteria for a sensor fusion control scheme is discussed in Murphy (1990). The six requirements are:

1. The input must come from real world, asynchronous sensors.
2. The scheme uses both redundant and complementary sensors for complex situations.
3. The system must be capable of processing quantitative, qualitative, and subjective information.
4. The sensor information must be presented in a way that it can be shared by multiple tasks.
5. The scheme must provide for robustness.
6. Knowledge of the environment should also be used whenever possible to reduce processing time.

The control schemes for sensor fusion can be grouped into three categories. The first is a physical science model. This method uses the physical characteristics of the environment and sensors. This gives quantitative information, such as geometric features. The second category, hierarchical representation, is a model that treats sensors

as generic data objects and describes a hierarchy of increasing complexity of representation and the information independent of the task. The last control category is action-oriented perception. In this process, all sensor information is examined with respect to perceptual context, the actions to be performed, the perceptual process to be carried out, and the impact of the environment on actions and perceptual processing (Murphy, 1990)

The sensor fusion model developed by Murphy (1990) executes this process in three steps: collection, assembly, and fusion. The three states of sensor fusion discussed are: 1. complete sensor fusion, 2. fusion with disagreement and recalibration of dependent sensors, and 3. fusion with disagreement and suppression of disagreeing sources. An excellent discussion of sensor fusion and some of the approaches that are taken can be found in Murphy (1990).

Depending on the application and the information needed, a wide variety of sensor combinations are used for sensor fusion. The most common combination of sensors is a visual sensor and a range-finding device, such as ultrasonics or a laser range finder. Using these two sensing devices, intensity and range information can be combined for a good environment representation (Xie et al., 1994). After testing with only one type of sensor and finding that this "imposed severe limitations" on their experiments, Olin and Tseng (1991) combined laser range finder and added color video. Sugiyama et al. (1994) fuse data from three laser range finders and twelve ultrasonic sensors for navigation. Vandorpe and Van Brussel (1994) use information from both encoders and gyroscopes

for position estimation. Boshra and Zhang (1994b) fuse data from visual and tactile sensors for robot localization.

The need for using multiple sensors on autonomous mobile robots will continue to increase as the complexity of the tasks required of the robots increase. By designing several sensor systems on the robot, it is possible to combine different data from several sensors to detect properties of the environment that a single sensor could not detect. It also creates a more robust system by using partially redundant information to verify data in a noisy, uncertain environment (Kamberova et al., 1991).

## 2.4 Summary

To conclude the chapter on the "perception" part of the robot, Tables 2.1 and 2.2 list some of the advantages and disadvantages of each sensor discussed. Sensing is the first task for autonomous mobile robot navigation and builds the ground work for navigation and control systems.

Table 2.1: Active Sensor Comparisons

| Sensor | Advantages | Disadvantages |
|---|---|---|
| Ultrasonics | Inexpensive, Easily Obtainable, Good for Obstacle Detection | Low Resolution and Precision, Affected by Noise and Specular Reflections |
| Laser Range Finders | High Resolution, Good Accuracy, Insensitive to Illumination | Expensive, Slow Response Time, May Create a Hazardous Environment |
| Optical Proximity Sensor | Inexpensive, Easy to Use Multiple Sensors | Can Not Determine Range Information |
| Structured Light | Accurate, Excellent Environment Representation | Expensive, Sensitive to Noise and Lighting Conditions, Specular Reflection |
| Infrared Sensors | Accurate at Short Ranges, Fast Response Time | Expensive, Short Sensing Range, Highly Dependent on Reflectivity of Surroundings |
| Radar | Accurate Information, Good Resolution, Good for Cluttered Environments | Expensive, Low Availability, Affected by Noise, High Power Requirement |

Table 2.2: Passive Sensor Comparisons

| Sensor | Advantages | Disadvantages |
|---|---|---|
| Direct Vision | Good Accuracy and Resolution, Easy to Classify Different Areas, Not Affected by Other Sensors | Hard to Obtain 3D Information, Affected by Lighting Conditions |
| Stereo Vision | Provides 3D Information, Not Affected by Other Sensors | Difficult to Relate Images, Affected by Lighting, Large Computation Time |
| Tactile Sensors | Inexpensive, Easy to Implement, Useful in Partially Blocked Areas | Useful Only in Applications where the Robot can Collide with the Obstacles |
| Encoders | Easy to Implement, Independent of Other Sensors, Inexpensive | Accumulates Errors |
| Odometer | Independent of Other Sensors, Simple, Inexpensive | Accumulates Errors |
| Potentiometer | Inexpensive, Produce Unique Resistance and Voltage Outputs | Low Resolution, Sensitive to Noise, Finite Rotation |
| Altimeter | Useful for Outdoor Mountainous Environments | Not Useful in Many Applications |
| Compass | Inexpensive, Good Sensor for Orientation Measurement | Problems when Using Near Ferrous Material or Magnetic Fields |
| Tilt Sensor | Good for Navigating Over Inclines | Not Useful in Many Applications |
| Accelerometer | Inexpensive Way to Measure Acceleration | Only Signals when the Desired Acceleration has Been Reached |
| Gyroscope | Easy way to Measure Orientation | Not Useful in Many Applications Due to Drift |
| Tachometer | Easy to Implement, Inexpensive | Only Useful for Robots at Higher Speeds |

# Chapter 3

# Mapping and Navigation

After sensing, the next function an autonomous mobile robot must perform is mapping and navigation. This is the "thinking" action of the robot. This chapter discusses how the robot takes the information acquired from the sensors and processes it to form a map of the environment. This map is then used to determine the best path to navigate to the next location, to avoid obstacles, or to reach a goal. Janet et al. (1995) list five requirements for this process. First, the method must always be able to find the best path in realistic, static environments. Second, it must also be expandable to dynamic environments. Third, it must be compatible with and improve the location referencing of the robot. Fourth and fifth, the method must minimize the complexity, data storage, and computation time required and the maps must be able to be created easily and must be simple for the robot to understand.

This chapter is divided into two sections: mapping and navigation. The mapping section will discuss different forms of maps used to represent the environment surrounding a robot and the navigation section will describe several path planning techniques.

## 3.1 Mapping

The mapping function of a mobile robot is an important task. The map forms the basis for all decisions regarding the actions of the robot. There are numerous different map representations being used today. These maps can be split into two classes: scene maps and grid maps. Scene maps show geometric information about the environment. They can be used for several different navigation strategies. Scene maps are divided into three groups: Full Scene, Edge Images, and Features. Grid maps are the most widely used of the mapping techniques. Grid maps models divide the area into smaller regions and assign a probability of the region being free or occupied by an obstacle. The different types of grid maps are certainty, histogram, and topographical/elevation.

Kelly et al. (1993) provide an excellent discussion on how to correct errors usually associated with mapping in an rugged outdoor terrain. This method incorporates numerous methods to eliminate problems with motion distortion, shadows, map fusion and edge smoothing. Each type of map is discussed and examples are presented.

### 3.1.1 Scene Maps

Scene maps use sensor information to produce a full or partial picture of the environment. Full scene maps are used create an image representing the entire scene to match with previous images for navigation or localization. Edge image maps show the extracted edges of boundary lines or obstacles. Feature maps use only certain characteristics of the environment to classify or describe the relationships of detected

obstacles by their attributes or topological relations. A brief survey of applications of these types of maps are presented in the next three sections.

#### 3.1.1.1 Full Scene Maps

Full scene maps are most commonly used for matching previously known information with recently acquired sensor information. They have also been used to classify areas into safe or unsafe regions. These maps are usually generated by a vision system, although other sensing systems have also been used. This is a very accurate method for navigation, but is computationally expensive. Also, all of the obtained information is not needed for navigation.

One type of full scene map proposed by Davis (1991) used a vision system, a scene model planner and a scene model verifier. The planner uses knowledge about the current scene, previous images, and the navigation goals to decide what objects to search for and approximately where to find them. This is matched with a previously known road map to determine the local navigation task. The verifier then checks sensor data to make sure the scene predictions made by the planner are valid.

A mobile robot system to recognize pre-programmed locations is discussed in Braunegg (1993). A stereo video image system is used to identify locations by comparing current images with previous images. Malik and Prasad (1991) present a mapping algorithm that can be used with or without boundary restrictions. This method scans the area and classifies it as either free space, where the robot is free to explore, or

occupied space containing obstacles. A navigation method used by El-Konyaly et al. (1995) uses a radar to build a three dimensional map of the environment. Collision-free sectors are identified and used for path planning. The path for the robot is chosen by identifying several subgoals based on how steep and rough the surface is. The robot proceeds to each subgoal on the way to the final goal.

Feng and Krogh (1990) use a local map of the environment that shows the location of obstacles for navigation. This map is constantly updated by the sensing system. A navigator is used to select different subgoals for the robot on the way to its final goal based on the local map. Ninomiya et al. (1995) discuss a navigation technique that uses a full-scene scan and several different techniques to extract information from a camera image. A road is followed using an edge extraction system, moving obstacles are then detected using an optical flow module and obstacles are detected by matching several successive images using stereo vision.

#### 3.1.1.2 Edge Image Maps

Edge image maps use sensor information to extract the edges of boundary lines or obstacles. These extracted images are then used for obstacle avoidance, to produce forces at the boundaries, and to compare points with known locations. These maps are less difficult to process than full scene maps, but they are more susceptible to sensor errors. Many edge detection maps require the robot to stop in order to run a sequence to extract the edges (Martinez et al., 1994) A method to convert grid-based world maps to line-

based world maps in real time is discussed in Zhao and BeMent (1990a). Line-based models require less computer memory enabling a larger area to be mapped. This makes line-based methods more efficient for planning a global path. A sample edge image map is shown in figure 3.1.

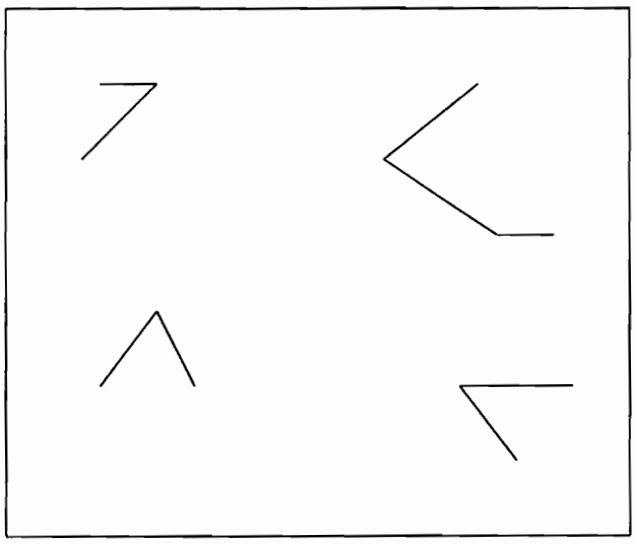

Figure 3.1: Edge Image Map

Sung (1990) describes a stereo vision system that stores the locations of extracted edge points in a table. These points are compared with known points for robot navigation. Veelaert and Peremans (1995) also use a navigation system based on extracted edges. The edges and walls are detected and generate contour lines. The contour lines show multiple paths that the robot can follow to reach the next checkpoint.

#### 3.1.1.3 Feature Maps

Feature maps only use certain characteristics of the environment for mapping. These characteristics show attributes, such as obstacle shape or color, and the topologic relations of obstacles and areas. There are several different types of feature maps that are used to describe the surroundings.

One type of feature maps is a topologic map. This map contains information about all fixed entities in the area, such as distinguishable locations and regions linked by topologic (connectivity) relations. This map represent an object by showing the vertices as points and the edges as being connective to the rest of the world. This map is qualitative and relates to the way people think and map (Dudek et al., 1993). Sugiyama et al. (1994) use a three-level path planning structure with three different maps. The global path planner uses what the authors label as a connection map. This map contains descriptions of topological relations among rooms and hallways. This planner only plans the order the rooms will be traversed by the robot. This information is passed to the local path planner that creates a geometrical map of known, static obstacles. Subgoals are planned and passed to the motion control module. Obstacle avoidance is carried out by creating a sensing map with a certainty factor applied to each cell (this will be discussed later). This information is also used for motion control. An "exploration tree" type map is used by Dudek et al. (1993) for robot navigation. This is a line-based map that shows the connectivity to all locations. The robot visits all locations in the environment and uses the information obtained to generate and constantly update the exploration tree. This

map is a "collection of possible hypotheses about the correct map, given the data accumulated thus far in the exploration." The root of the tree is where the robot started. Each level signifies the robot going to an unexplored edge. The nodes of the tree show the connectivity of the world.

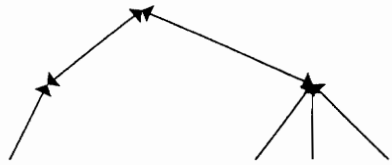

Figure 3.2: Exploration Tree

Froom (1991) proposes a navigation strategy for an autonomous mobile robot using only local information. This strategy uses the geometric information taken from sensors to move through the environment. This information is stored from several trials to get a "minim-time trajectory" for the robot. A navigation system using self-organizing maps is discussed in Heikkonen and Oja (1993). These maps are created for collision avoidance based on the geometric information the robot learns during several navigation attempts. This allows for a continuous learning process for the robot and enables it to easily adjust to unknown environments. The environment information is obtained from a camera and obstacle information is obtained from tactile sensors. The robot moves about the environment recording information and trying to avoid obstacles. If the vehicle collides with an object, it backs up and starts again. A local map of obstacle geometry for navigation is used on a mobile robot described by El-Konyaly et al. (1995b). The path

for obstacle avoidance created by this map is monitored for constant feedback is a servo controller loop.

A mapping system that names obstacles based on their geometry is a key feature of an object classification system described by Bobick and Bolles (1992). This system uses multiple descriptions of objects. Objects are assembled and displayed in representation space. Each representation is composed of smaller object identifiers, such as 2D-BLOB, 3D-BLOB, STICKS, etc. The description of each object is updated by new information, but the old descriptions are kept because they may be more useful in certain navigation tasks.

A good example of classifying areas by their attributes is the vision system discussed by Fernandez et al. (1994). This system performs three tasks: image segmentation, region classification, and obstacle and way limits location. The image segmentation first examines the color of each pixel and groups pixels into regions. These regions are then combined to form an image of the road. Objects and road regions are then classified. Areas that are flat are considered road and uneven areas are considered obstacles. The boundary lines for the road and object edges are extracted as the last task. Another navigation procedure is discussed in Roberts and Bhanu (1992). The technique discussed determines which points taken from sensor data have the most distinguishing features and uses these points for the analysis. A "measure of goodness" is given to each area, and the points with the highest "interestingness" value are reported. These points are then observed from several different positions of the robot. The selected points are then

matched and tracked in consecutive frames. Using the matched images, the range of objects can be calculated and used to create an obstacle map. This information is used by the motion analysis algorithm to make steering decisions.

### 3.1.2 Grid Maps

Grid maps represent the robot's surroundings by transforming the area into a grid. This grid is composed of cells that identify the areas of the environment as being safe or dangerous for the vehicle to enter. In a certainty grid each cell is labeled depending on the probability of an object being present. A histogram grid is much like the certainty grid, but differs in the way it is created and updated. Topographical and elevation maps describe the environment in terms of the terrain. Each type of grid map is explained and examples of each type are given.

#### 3.1.2.1 Certainty Grid

Based on the articles surveyed, a certainty grid based map was the most popular type of grid map. The introduction of the certainty grid can be found in Moravec and Elfes (1985) and Moravec (1988). This type of map is a representation of the probability of an obstacle being in a selected location. The probability value can be assigned by different representations. The most common represents an obstacle in a cell with a value of one and empty spaces with a zero value. Some methods also assign probabilities between zero and one to cells based on the possibility of having an obstacle at that location. Other

methods assign names or colors to cells rather than numbers. Cells are often classified as FREE or OCCUPIED, FULL or EMPTY. The most common grid of this type is the color certainty grid where obstacles are represented by black cells, free space by white, and uncertain areas by gray. Figures 3.1 and 3.2 show examples of certainty grids with number and color values. A 1 represents an obstacle, a 0 a free area, and a 0.5 an undecided area in the first figure. In the second figure, the cells known to be occupied are shown in black, free cells in white, and undecided cells in gray.

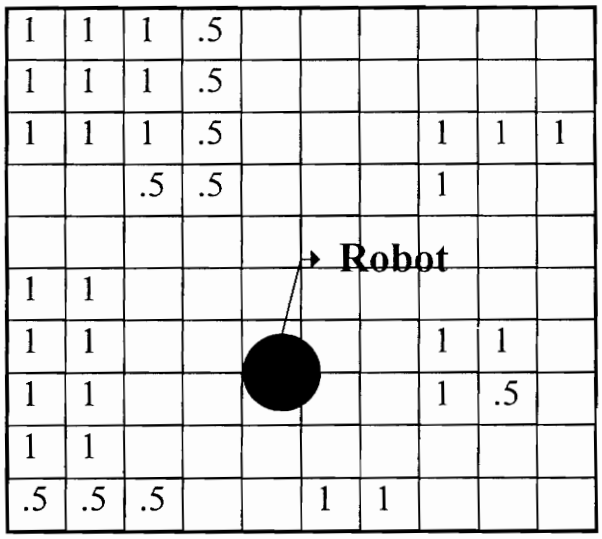


Figure 3.3: Number Based Certainty Grid

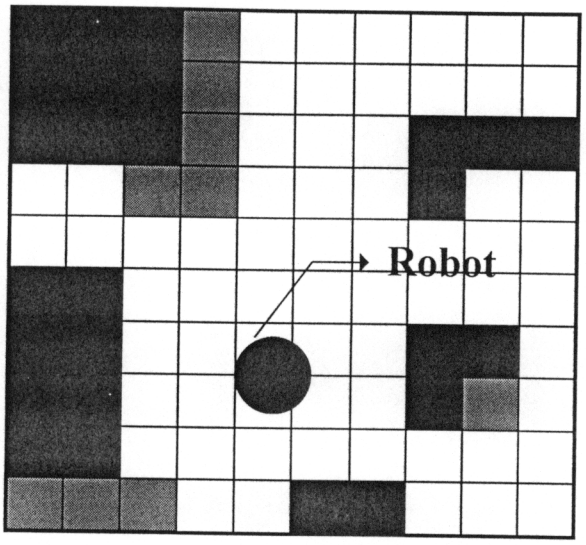


Figure 3.4: Color Based Certainty Grid

Certainty grids are popular because they can generate accurate maps from incomplete sensor data and information from several sensors can be easily combined. Also, the robot can navigate more easily at this "intermediate complexity level" than it can with more complicated map representations (Good, 1993; Dodds, 1990). The cells in this grid can also be constantly updated when a robot leaves a cell by marking it with a high probability of being empty (Good, 1993). A disadvantage is that if the robot is using a certainty grid of only the local area, the robot must occasionally stop to update the map (Martinez et al., 1994).

Good (1993) used a certainty grid that assigns a value of zero to all empty cells, and a value of one to all occupied cells. All other cells are assigned a value between zero and one representing the probability that the area is occupied. A background value was

initially assigned to all of the cells. Two separate, independent grids were used. Using only one grid, the measure of confidence related to the probabilities is lost. The first grid represents the probability values. The second grid relates the confidence with the probability values in each grid. The confidence in the probability values is the difference between the occupied and empty probabilities. A confidence factor is assigned to each sonar reading. This factor is combined with the confidence factor of the certainty grid elements to determine the weights of the new versus old sonar readings from the grid. Because the uncertainty of the location of obstacles changes with respect to the position of the robot, the probability of the object remaining in the same location decreases and the probability of the object being in one of the cell's neighbors increases. This situation is dealt with by taking a weighted average of each cell and a weighted average of the neighbors. One of the weights is based on the distance the robot has moved since the last decay of grid values. Confidence values are used to weight the neighbor cells before they are averaged. The changing location of the robot also decays the confidence values. "By linking the decay of the confidence values to the decay of the occupancy values, the confidence value of a cell should approach 0 at the same time the occupancy value approaches the local background." A drawback to this plan is that prior map information is lost after each decay process. This process can be turned off if past maps are needed.

Baxter and Bumby (1995) use a certainty grid representation to aid with the obstacle avoidance for a fuzzy controller. A grid representation is made of the area around the vehicle. A "reverse" value is used in this method because a zero entry means there is an

obstacle, a one indicates no obstacle, and a value in between zero and one represents the possibility of an obstacle or a obstacle at a long distance away. This method is good for use with a fuzzy logic controller because several sensors can be used for a good environment representation. Also, navigation and obstacle avoidance algorithms can use the map without needing to know how the information was acquired, thus improving portability, and the probabilistic nature of the information makes it ideal for processing by a form of probabilistic or fuzzy logic.

Shiller and Gwo (1991) use a graph map that assigns a mobility of zero to inaccessible regions and a one to perfect roads. Other regions are denoted with a value between zero and one. Several paths to the goal are generated and feasible paths are determined based on the optimal motion time. An obstacle detection system using a vision system is described in Zhao and Yuta (1993).

Siemiatkowska (1994) uses a local map that labels each cell free or occupied. These labels are represented as fuzzy sets and used for the motion control of the robot. A path planning approach discussed in Hebert (1994) uses range or camera information for map building. Every pixel in the image is transformed into local coordinates and stored in a cell in the grid. The grid is checked and traversable and untraversable cells are identified. This information is used for path planning. A navigation system discussed in Stentz and Hebert (1995) classifies each cell on a map as traversable, high-cost, or untraversable. High-cost cells are located within a selected radius of untraversable cells. A path is selected by evaluating the safety of paths. The safest path is the one with the lowest

"cost". Hou and Zheng (1991) describe a path planning technique that uses hexagonal decomposition. Hexagonal cells are labeled as PASSABLE or IMPASSABLE. They suggest using hexagonal cells over traditional square grid cells because the hexagon can be always be divided into smaller hexagons or half hexagons, each cell has six neighbors making the searching procedure easier, and the distance from the center of a hexagon to the center is its neighbors are the same.

A color based mapping procedure was used by Lux and Schaefer (1991). A vision system would label each pixel of an obstacle map green is for a safe area, red for an unsafe area, and black if no information on that area is available. A gradient method is used to analyze the data and decide what color each pixel should be. A green pixel could be changed to red if the area appears to be elevated. A series on connected green points were used as a safe terrain to cross. A mapping technique used by Weigl et al. (1993) uses sensor information to classify obstacle areas. A black area on the map indicated that a cell is occupied, a striped cell shows that the cell is partially occupied, and dots in the cells mean that the status in unknown. This approach can be used to represent the entire environment for robot navigation. Ishiguro et al. (1993b) use a camera to acquire omnidirectional information about the surroundings. This information if transformed into a global map by fusing data taken at several observation points. The map contains black and gray points, based on the certainty of an obstacle in that location. This information can be used to determine the position of the robot at different observation points and the absolute position of the robot. A probability map is used by Lim and Cho (1993) to show

the location of objects. This map fills each cell with gray levels according to the occupancy probability. White cells are empty and black cells are occupied. The robot navigates to each subgoal based on the intensity of each cell.

An occupancy grid, presented in Elfes (1989), is based on the certainty grid concept. Borenstein and Koren (1991) state that the occupancy grid is similar to the certainty grid, but models the sonar sensors with Gaussian uncertainty and applies a more rigorous mathematical model to recurring readings. The classifications for the cells in the grid are the same as for the certainty grid and often the two are used interchangeably.

Gourley and Trivedi (1994) describe a using a occupancy grid with a potential field method. The occupancy grid labels each cell with the probability of it being occupied. A probability of 0.7 or greater assumes an obstacle, 0.3 to 0.7 is an unknown state of the cell, and 0 to 0.3 is considered not occupied. The robot's current and desired position are selected and the two points are connected by the path with the least potential. Yeung et al. (1993) use an infrared sensor to obtain a view of the environment and create an occupancy grid that contains a probability number that represents the likelihood of a cell being occupied. The transformation of occupancy grids is discussed by van Dam et al. (1994). They use an ego-centered robot to create an occupancy grid. This ego-centered occupancy grid needs to be updated every time the robot's perception changes. This is achieved by merging the grid with new sensor information from the robot. Path planning is carried out by training neural networks on the transformed occupancy grids.

A decomposition grid representation model is used in Yang and Wu (1995). This map uses a series of two-dimensional occupancy grids generated from different sensors at different resolution levels. The maps are fused together and the cells are connected for each level. The cells are labeled as free, obstacle, or undecided for navigation. Koskinen et al. (1993) use an occupancy grid for obstacle avoidance. By dividing the area in front of the robot into a square grid, each cell is labeled as free space or obstacle. The labeling depends on if the selected probability value is less then or greater than a threshold value. "If an echo is received corresponding to a certain cell, its occupancy probability value is increased, otherwise it will be decreased by using a probabilistic estimation method."

#### 3.1.2.2 Histogram Grid

Another new map building method called Histogramic In-Motion Mapping (HIMM), introduced by Borenstein and Koren (1991), uses a two-dimensional cartesian histogram grid for obstacle representation. Each cell in the histogram grid has a certainty value with the probability of an obstacle being at that location. It differs from a in the way it is created and updated. With the certainty grid, all of the cells affected by a range reading are updated at the same time. This mapping method is computationally intensive and hard to implement for real-time robot motion. The HIMM method updates only one cell for each range reading. This results in cells with a high certainty value being the actual location of the obstacle. Empty sectors are also determined from sensor reading with both methods, but the certainty grid predicts a probability for each cell while the HIMM

approach only updates the cells that are on the acoustic axis of the sensor. The HIMM method also uses a different approach for assigning values to the cells. Cells are incremented by three and decremented by one with a maximum certainty value for a cell of fifteen and a minimum cell value of zero. "Increments are larger than decrements because only one cell is incremented for each reading, whereas multiple cells are decremented" (Raschke and Borenstein, 1990). A sample histogram grid is shown in Figure 3.5. This graph shows how only cells on the acoustic axis of the sensor are used to update the grid. This grid shows that certainty values of the cells that detect an obstacle are incremented by 3 and free cells are decremented by 1.

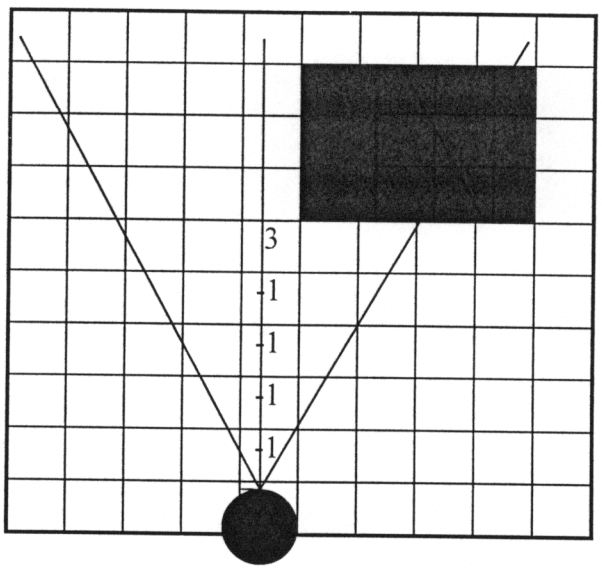


Figure 3.5: Sample of Histogram Grid Updates

This type of map is more effective for real-time control of the robot, but is less accurate when the robot is stationary (Borenstein and Koren, 1991). Another advantage

of this approach is that low certainty values are given to cells that were incremented due to sensor error or moving objects. A disadvantage is that low certainty areas next to high certainty areas result in high certainty values and cause obstacles to be represented larger than they actually are (Borenstein and Koren, 1991).

Borenstein and Koren (1990a) introduce the HIMM method for real-time map building using a histogram grid. This approach allows for quick and accurate mapping and safe navigation of the robot toward a target. This approach is integrated with the Vector Field Histogram method for real-time obstacle avoidance. Vandorpe and Van Brussel (1994) use a histogram grid map for an autonomous mobile robot. Positive numbers represent an occupied space and negative numbers indicate a free space. A path is planned from the goal point back to the robot by looking to move to increasingly smaller numbers. Stuck et al. (1994) use a histogram grid for collision avoidance. The histogram grid obtains a probability distribution by continuously and rapidly acquiring sensor data while the robot moves. This grid contains an occupancy value from 0 to 14 depending on the certainty of the location of an obstacle in a cell. Path planning is accomplished by starting at the goal position and checking to see if cells are safe or unsafe. A safe cell at the goal is selected as the best cell to start navigation. The best cell is expanded by looking at the neighbors of that cell and deciding where to expand. The final output is a path from the robot's current position to the goal.

#### 3.1.2.3 Topographical/Elevation Maps

When a robot is navigating in an outdoor, mountainous, unstructured terrain, a variety of factors can pose a threat to the robot. This type of cross-country path planning has to consider factors such as rocks that the vehicle can not climb, ditches that could trap the vehicle, or inclines that will tip the vehicle over (Gowdy et al., 1990). To deal with these difficult areas, maps that provide information on obstacle locations and terrain are needed. These maps are elevation and topographical maps.

A cartesian elevation map is used by Olin et al. (1991). Creating this map involves taking sensor data and transforming it into cartesian coordinates. This produces a downward-looking terrain representation. Areas where too few samples are available to produce a clear representation are interpolated from known areas. An area is sampled several times to get a good picture of the environment. Errors from the sensor are reduced by averaging several maps. If an object is only detected in one map, this virtually eliminates any effect that it has on navigation. The path for navigation is selected by determining several subgoals. The robot moves toward the subgoals using a reactive planning method that adjusts the speed of the vehicle based on the amount of information available. It slows down as it nears the end of a map and may even stop, if necessary, to wait for a new path.

Gowdy et al. (1990) also use a cartesian elevation map to show the height of the terrain for robot navigation. The values for unknown cells use an interpolation method that checks the value of its neighbors and randomly assigns one of those values to the

cell. Several maps are also fused together for a better representation of the surroundings. Each area is then classified as traversable or nontraversable considering the elevation of areas, the location of obstacles, and the turning radius of the vehicle. To better represent the terrain, different maps with varying resolutions are used. This map structure is called a terrain pyramid with the top map showing the minimum and maximum elevation values over the entire area.

A digital elevation map is used by the robot described in Wright and Simeon (1993). This map is a grid that shows the elevation values at all locations. This information is used for path planning by dividing a cubic area into subcubes. The method used is an oct-tree representation. This consists of analyzing the cube and declaring it FREE or MIXED based on the location of obstacles. If a cube is MIXED it is divided into eight smaller cubes and each area is analyzed again. This method can also be used to divide the areas into different numbers of cubes. A quad-tree representation simply divides each area into four cubes instead of eight cubes.

A digital elevation map is also used on the robot discussed by Talluri and Aggarawal (1993). The location of the vehicle is determined by comparing images from a known map to newly acquired images. The first step of this method is to obtain images in the four geographic directions. One of the images is then compared with the digital elevation map. Based on the maximum height of the area, certain locations can be ignored and the areas the robot could be in are marked. This is repeated for all four images. After the set of all possible locations is determined, the horizontal line contours (HLC's) are then

extracted. These HLC's are compared with the actual image HLC using a curve matching technique. The error between the actual image and the observed image is calculated and the image with the lowest error is considered the location of the vehicle. Figure 3.6 shows an example of actual and extracted HLC's.

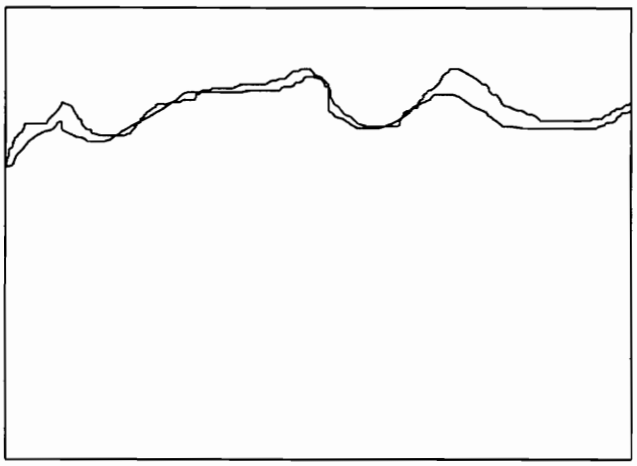

Figure 3.6: Horizon Line Contours

An interesting approach for mapping the environment is proposed by Tsukamoto et al. (1992). They use a linguistic method to make a topographical representation of the environment like humans would relate it to each other. Examples of the words are TOP to represent the top of a mountain, VAL for the point of a valley, RID is for a ridge line, and SLO is for a slope. These descriptions are used to create a topographic map by dividing the area into a grid. The height of each location is measured and it is labeled by determining the feature point. Adjacent areas with the same labels are grouped together in "clusters" and numbered from zero according to height. Objects can also be identified

and represented in the same manner. This map can aid the autonomous navigation of a mobile robot in a mountainous terrain. An example of this type of map is shown in figure 3.7.

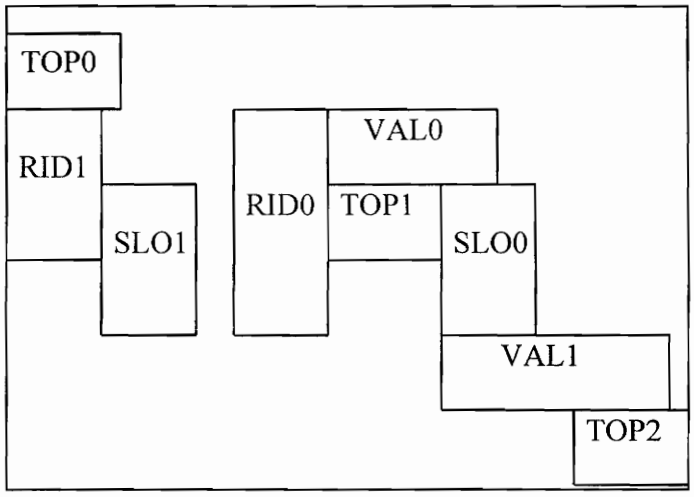


Figure 3.7: Linguistic Topographical Map

A terrain map is used by Brumitt et al. (1993) for cross-country navigation of a mobile robot. A terrain map is produced by a laser range finder on board the vehicle. The path planner generates a path using the terrain map and tests it for safety. If it fails, additional paths are tested and modified to find a safe path. An obstacle-avoidance system used by Badal et al. (1994) takes stereo images and extracts points above the ground plane. A type of terrain map that shows a grid representation of obstacles, an Instantaneous Obstacle Map, is created that can be transformed into a steering vector. Obstacles are prioritized by their distance away from the vehicle. Sharper turns to avoid the obstacles

are made when the obstacles are viewed as being closer. Stentz et al. (1993) describe a cross-country navigation system for an autonomous mobile robot. This system uses a laser range finder to make a local map of the robot's surroundings that records the minimum and maximum elevation values. This map is to be fused into a global map. The path is planned by checking the kinematic, dynamic, and static constraints from the cells on the map.

## 3.2 Navigation

Navigation is defined by Schalit (1992) as "the act of performing all necessary actions to reach a physical location, the goal." This function of the robot takes the information from the map builder and finds the safest and fastest path to navigate while avoiding obstacles. "The goal of navigation in uncertain environments is to make the mobile robot arrive at [a] destination point without colliding with obstacles" (Beom and Cho, 1995).

Navigation strategies have developed considerably over the past several years. Some of the first mobile robots used Wire Guided Navigation strategies. These wire guided systems involved the robot following a wire on a predetermined path. This type of navigation is useful in factory and warehouse environments where the robot always traverses the same path. This type of system is reliable, but its applications are limited (Inigo and Alley, 1990). Such systems also suffer because they lack flexibility, can be

expensive, and do not possess obstacle avoidance capabilities (Inigo and Alley, 1990; Kang et al., 1995).

Current strategies are more flexible. These enable the robot to be fully autonomous. The following sections present the navigation strategies in three parts: map searching, potential fields, and neural networks. Map searching techniques use a chart or graph that shows free and occupied spaces and plans a path around these areas. Gradient summation methods apply attractive forces, on the robot, to the goal position and repulsive forces to obstacles. The resulting force plans the path to the goal. Recently, a new method has been used for path planning called neural networks. Neural networks are used to train the robot to identify certain situations in different environments. The use of fuzzy logic on autonomous mobile robots is also a new area of research. Fuzzy logic can be classified as a navigation strategy for obstacle detection and avoidance or as motion control to generate steering commands. Because most fuzzy logic applications in this survey are used to generate steering commands, this topic is presented in the motion control chapter with both areas of fuzzy logic being discussed. An explanation of each navigation strategy and current implementations follows.

### 3.2.1 Map Searching

Map searching techniques are the most popular form of robot navigation. This is because it is simple to create a good map of the environment to begin the navigation process. This method involves moving the robot from one free space to another by some

method. Certainty grids are a common map used for this purpose because it is easy to interpret the map information. The two areas of map searching discussed are cell-based and vector-based map searching. Cell-based map searching techniques involve looking at the areas directly around the robot to plan the next position. Vector-based map searching generates a series of vectors that navigate the robot around obstacles.

#### 3.2.1.1 Cell-based Map Searching

Cell-based map searching is one of the easiest forms of navigation to implement on an autonomous mobile robot. Most cell-based navigation plans follow the same simple rule: move to the adjacent cell that is least likely to contain an obstacle. Figure 3.8 shows a robot navigating around obstacles. The light gray squares show the path of the robot. Notice that the robot will move closer to squares that are undecided instead of known obstacle locations. Some of the simple methods of cell-based navigation were presented in the mapping section when discussing the use of the map. A few additional examples are included in this section.

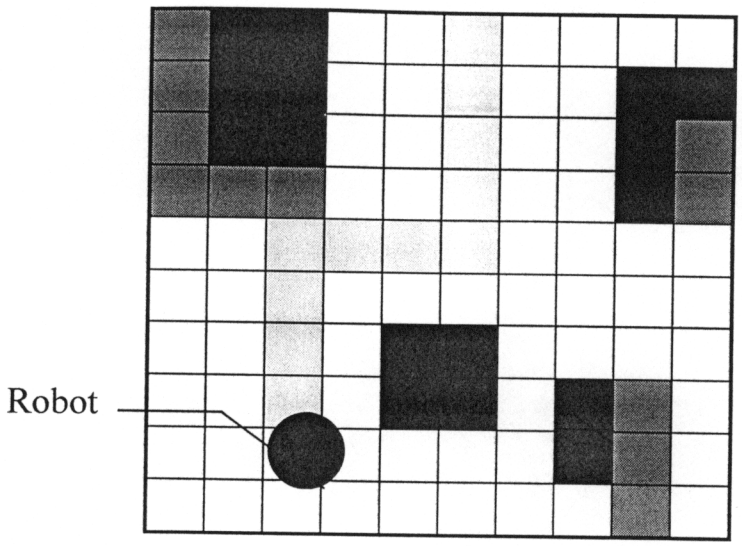


Figure 3.8: Cell-based Map Searching

A navigation system discussed by Good (1993) developed four navigators that were based on the certainty grid approach. The first was a local navigator that had three separate grids: a direction grid contained destination information with a potential value assigned to each cell, obstacle locations were stored in the certainty grid, and a travel grid kept information about the path previously traveled by assigning high potential values to areas that have been entered. These three factors are combined to get a single "badness potential". The robot follows the direction of lower "badness". The second navigator is a backtrack navigator. This navigator selects subdestinations and sends this information to the local navigator. If a subdestination turns out to be a dead end, then the robot goes back to a previous subdestination and tries again. The path navigator was the third type of navigator developed. This navigator used the direction grid to show both known paths and possible paths. The last navigator, the incremental navigator, would update the

direction grid every time the robot stopped. Since the robot stopped often, the grid was constantly updated. This navigator proved to be the most effective and easiest to implement.

Cho and Lim (1995) use a certainty grid method for robot navigation "based on the distance from the current position to the boundary between known and unknown regions on the way to the destination." They use information from sonar sensors to plan two paths around an obstacle, one to the left and one to the right. The path with a point closest to the goal is selected. A path planning method used by Prabler and Milios (1990) combines the idea of a graphing technique with the potential field approach. In the generated map, each cell contains the probability an object is in that area. A potential is associated with each cell. The potential of each cell is recalculated based on the potential of its neighbors. Navigation is performed by assigning a zero to the starting cell and a negative ten to the goal cell. The shortest distance between the start and goal cells is selected for the path.

The terrain pyramid described by Gowdy et al. (1990) is used for navigation by taking the requirements requested by the planner and searching for areas that meet the criteria. The planner will ask for an area with a minimum and maximum elevation. If the observed area meets the requested elevations, then the robot is safe to move. The area is then divided into quadrants. Each quadrant is checked and compared with the requested minimum and maximum values and either put on a list of acceptable areas or divided into another set of quadrants. This is repeated until all of the maps on the terrain pyramid

have been checked. The robot is then able to determine which areas are safe to navigate through without getting trapped at an obstacle.

A method using a digital elevation map described by Wright and Simeon (1993) first identifies a start and goal position. Using the start position, the neighbors are analyzed to determine FREE areas. When checking a neighbor, if it is FREE the "Father" areas are also checked to determine the largest FREE area to enter. If it is MIXED the "Sons" are checked to determine other FREE areas. All neighbors around the start position are checked. This process continues until a sequence of FREE areas are found that allow the robot to safely move to the goal position.

Schalit (1992) uses a homogrophic sensor map for navigation. This map plots the impact points of a range finder on a flat horizontal surface in front of the sensor to get a grid-like representation. The navigation task is divided into five modes. In the first mode, reflex navigation, the robot responds to the sensor information; the environment is never modeled. The next level, line of sight navigation, enables the robot to track a goal. The short-term memory and long-term memory use store areas of failure to correct these problems, and specific information of landmark and edge locations, respectively. The full navigation mode uses maps to plan a complete path.

A distance value model for autonomous mobile robot navigation was discussed by Ikegami et al. (1990). This is a grid-based environmental representation that determined the shortest distance between a boundary line and the free and occupied areas. The sign on the graph is positive for an occupied area and negative for a free area. The distance

information is extended so unknown areas and ambiguous information can be dealt with. An optical flow map is used by Attolico et al. (1990) for obstacle detection. This map is a two-dimensional description of "the apparent velocities of the motion of brightness patterns in the image observed when a camera is moving relative to the scene being imaged." Each point on the grid is assigned a two-dimensional flow velocity, which is used for obstacle avoidance. The navigation plan is carried out in three steps. A global map containing geometric information of the environment, obtained by a vision system, is used for path planning. A local map of the immediate area around the robot is made by the plan interpreter and obstacle detection modules. A control system provides feedback on the current status of the robot and can identify if the path planner needs to create a new map. The map built by Guldner et al. (1994) used the method of wave propagation. A grid map of the environment was taken and a direction was assigned to each cell. A directional field was created to be used for navigation. This method would direct the robot to a desired goal. Another wave propagation method is used by Knick and Schlegel (1994). A laser range finder is used to map information in a grid that classified each cell as free or occupied. Wave propagation is used to generate a direction from the location of the robot. An optimal path is found by determining the most distant point that can be reached by a straight line for an intermediate goal and constantly updating the map until the goal is reached.

#### 3.2.1.2 Vector-based Map Searching

Vector-based map searching uses the relationships and the geometry of the obstacles in the environment for navigation. Using these relationships, a vector path is found to navigate from a start position to the goal. Two types of navigation strategies using vectors are presented: vector field histograms and traversability vectors.

##### 3.2.1.2.1 Vector Field Histogram

To eliminate problems associated with using potential fields, an obstacle avoidance method called the Vector Field Histogram (VFH) was developed by Borenstein and Koren (1990b). This method produces a smooth, non-oscillatory motion of the robot for navigation (Koren and Borenstein, 1991).

There are three levels of data used for this navigation process. At the highest level, a two-dimensional cartesian histogram grid is constantly updated in real-time with the on-board sensing system. At the second level, the histogram grid is reduced to a one-dimensional polar histogram constructed around the current position of the robot. The basis for this transformation is that the occupied cells produce a large vector magnitude close to the robot and smaller ones when they are far away. The value for the polar obstacle density is given for each sector. At the lowest level, a sector with a low obstacle density close to the target is selected and used for the values for the drive and steering controllers. These values are determined from the peaks and valleys of the polar

histogram that represent high and low obstacle density, respectively (Borenstein and Koren, 1990b). This step-by-step process consists of:

1. Reading the sonar information
2. Updating the Histogram Grid
3. Creating the Polar Histogram
4. Determining the free sector and steering direction
5. Calculating the new speed
6. Communicate with the motion controller.

An introduction to this method and sample results can be found in Borenstein and Koren (1990). Figure 3.9 shows a robot moving a histogram grid with certainty values. The smooth motion of the robot is shown.

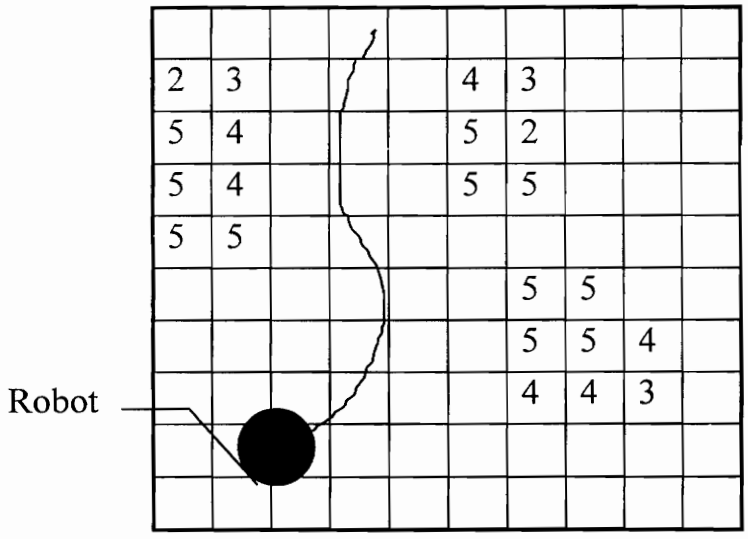


Figure 3.9: Vector Field Histogram

The strengths of the vector field histogram method come from the fact that an obstacle representation is maintained at both the world-model level and the intermediate data-level. This means that the vehicle navigates around clusters of data points that represent obstacles and ignores single points caused by errors. Oscillations of the vehicle when going through tight areas are also eliminated because the second level information, the polar histogram, is still available (Borenstein and Koren, 1990). Although this method eliminates many problems associated with potential field methods, the vehicle can still fail from the "trap problem" (Zhao and BeMent, 1990). This is where the robot gets stuck in a dead end situation and unable to find a way out to head toward the goal.

A method to eliminate the trap problem is presented in Zhao and BeMent (1990). This method uses two maps at the same time. An upper-level, grid-based, map is used for obstacle avoidance and a lower-level, node-based, map is used for trap recovery. The grid map is constructed from ultrasonic sensor readings. Each cell is given a certainty value from zero to fifteen depending of the likelihood of an obstacle being present in the cell. This information is used for obstacle avoidance on the way to the goal. The node-based map is used when a trap is detected. A search window is generated to look for paths to get out of the trap and a viewing window is defined by the area covered by the ultrasonic sensors. A trap recovery algorithm is used to get out of the trap. The four steps are:

1. Concurrent reduction of the grid-based map to a node-based map
2. Determination of a search window in the node-based map in conjunction with a

temporary search goal when a trap is detected

3. Generation of an optimal set of via points (shortest distance) inside the search window
4. Evaluation of candidate via points along the robot's path and via point elimination for those points not qualified.

The node-based map is created by reducing the certainty values of groups of cells to a value of one to indicate an obstacle and a value of zero for a free area. The obstacle nodes and its neighbors are labeled as "forbidden" nodes. For real-time control, a smaller search window is used to aid with the path planning. Via points are generated as intermediate goals to reach the final goal. The navigation system then uses the vector field histogram for obstacle avoidance and moves through the via points toward the goal.

The advantage of this approach is that it allows more information about the robot's surroundings to be used. Real-time navigation is achieved because the vector field histogram is constantly using updated sensor information for obstacle avoidance and switching to the trap recovery algorithm when a trap situation is detected (Zhao and BeMent, 1990b).

Another vector-based method is presented by Warren (1991). This method creates a vector from the start position to the goal of the robot. If any part of this vector crosses "forbidden space", a new vector is created to determine an intermediate goal. The previous goal is stored in a stack. New vectors are created until an intermediate goal can be reached. The robot is then moved to this goal. A previous goal is then taken from the

stack. The robot either moves to this goal or creates more intermediate goals. This is repeated until the initial goal is reached.

#### 3.2.1.2.2 Traversability Vectors

The use of traversability vectors is another technique for mobile robot navigation. This method uses the "spatial relations" between geometric obstacles for navigation. The spatial knowledge is described by the visibility and interference of the obstacle. "T-vectors provide a utility, efficiency and mathematical stability for collision detection and visibility that cannot be matched by commonly used algebraic approaches in static and dynamic environments" (Janet et al., 1995).

A method for using T-vectors developed by Luo and Pan (1990) is based on the assumption that the obstacles can be represented by a set of half-planes. The coefficients of these half-planes are stored as the basic world description and can be updated with changes in the obstacles. This enables the system to constantly process new information and use the most recent to plan the best path.

The obstacles are described first by using "algebraic expressions to model convex polygonal obstacles as an intersection of second-order inequalities" (Luo and Pan, 1990). The coefficients are stored as the description of the world. Using these coefficients, a traversability vector is created. The spatial knowledge of obstacle locations interfering with the path of the robot is used to plan a safe path. The location of obstacles can be updated easily in the world model. A new obstacle can be added by locating its vertices

in cartesian coordinates and computing the coefficients of the inequalities. Obstacles no longer in the path of the robot can easily be removed by simply deleting the coefficients. This enables the system to be able to adapt to a static or dynamic environment. Once the knowledge of all of the obstacles is computed, path planning consists of moving the robot to a goal position while avoiding obstacles. The shortest path is selected by computing the total distance of each possible path and choosing the one with the minimum value. The planning system is able to easily adapt to changes in obstacle information, making it suitable for static or dynamic environments. Figure 3.10 shows a polygonal obstacle with the traversability vectors shown. Each of the ten regions divided by the extended boundaries are numbered.

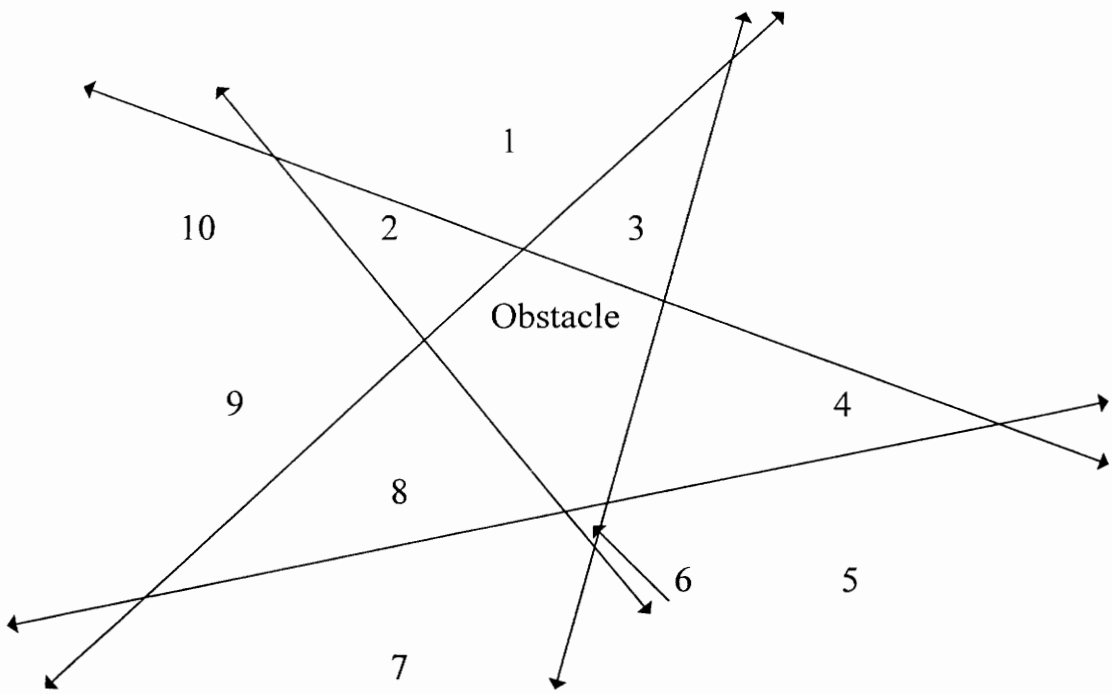


Figure 3.10: T-vectors and Created Regions

Janet et al. (1994) use traversability vectors based on decomposing concave objects can be into convex polygon sets. The edges of these polygons can be classified as non-traversable. "If all vertices of the polygon are in the same half-plane than it can be concluded that no collision exists" (Janet et al., 1994). An efficient path can then be planned around these obstacles. The obstacles are mapped on an visibility graph, or V-graph, that connects two vertices that are visible to each other.

A different graphing technique using T-vectors, an essential visibility graph or EVG, is used by Janet et al. (1995). The difference is that, with a V-graph, all path segments, optimal and redundant, are included. The EVG creates a non-redundant map by constructing a network where every node is an optimal point. This makes the EVG graph simpler, smaller in size, and easier for the robot to use.

The traversability vector method is good for avoiding obstacles that can be represented by polygons. It is not good for boundary line detection. This makes this method useful only for specific applications.

### 3.2.2 Gradient Summation Methods

Gradient summation methods are a commonly used form of mobile robot navigation. These methods operate on the concept of assigning attractive forces to the goal position and repulsive forces to the obstacles. The combination of these forces produce a path that will safely navigate the robot to a goal point. There are three main concepts that use this approach: artificial potential field methods, navigation templates and vector force field

methods. They all operate on the above principle, but vary in the way the forces are assigned.

#### 3.2.2.1 Artificial Potential Fields

Assigning artificial potential fields is the most commonly used gradient summation method. This method applies an attractive force on the robot from the goal and repulsive forces from the obstacles. The sum of the forces determines the path for the robot to follow. The initial development of this approach can be found in Khatib (1985) and Andrews and Hogan (1983). This method is simple and easy to implement, but needs accurate descriptions of obstacle locations for successful navigation (Koren and Borenstein, 1991; Martinez et al., 1994). This method is responsive to the current state of the environment and is able to effectively navigate around newly sensed obstacles while moving (Prabler and Milios, 1990). Payton et al. (1990) describe this method as "a phantom compass that always gives the general idea of the right way to go." Figure 3.11 shows the artificial potential field approach applied to a simple environment. The desired path is shown by the direction of the arrows.

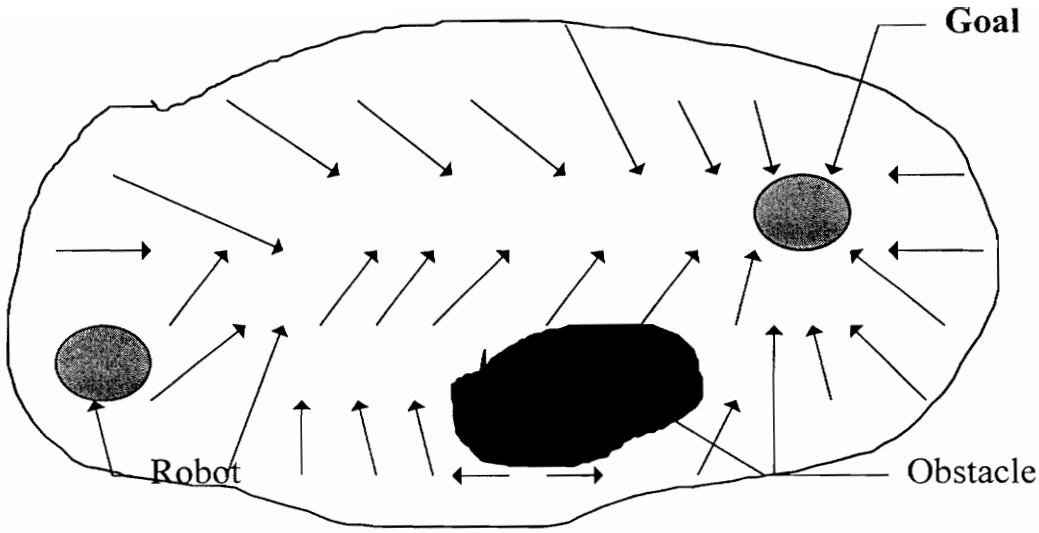


Figure 3.11: Artificial Potential Field

One of the advantages of artificial potential field approach is that the navigation plan is formed from the information provided from the current location of the robot. This means that as the robot's perception of the environment changes its navigation plan also changes, forming a new path on recent sensor information. Even though this method is not as thorough as map searching techniques, the algorithms used to generate paths using potential field methods are fast and efficient (Slack, 1993; Hou and Zheng, 1991).

There are several drawbacks with artificial potential field approaches. The biggest problem is the one of local minima (Siemiatkowska, 1994; Reid, 1993; Hou and Zheng, 1991; Prabler and Milios, 1990; Janabi-Sharifi and Vinke, 1993). This occurs when the attractive and repulsive forces sum to zero and the robot would not move in any direction. Examples of this problem can occur when two obstacles that are close together form a high potential ridge between them with a local minima in between or if the goal is located

directly behind an obstacle and the resulting force directs the robot toward or away from the obstacle (Reid, 1993). Eliminating the local minima problem is a current research topic. Two proposed solutions are to redefine potential functions and to use efficient search techniques. By defining potential functions with "circular thresholds" some local minima problems are eliminated. This solution is computationally expensive and will only work in cases where the obstacles have a simple geometry. A variety of search techniques have been proposed, but they are usually slow and unreliable (Janabi-Sharifi and Vinke, 1993).

There are several other problems with the potential field method. The robot oscillates in narrow passages or in the presence of obstacles and is unable to pass between closely spaced obstacles. Obstacles that are not in the way of the robot can also produce adverse effects on the path. The robot is unable to reach a goal if it is behind a wall. And there is no plan for motion decisions to pass an obstacle on the right or left, which can leave the robot trapped. This method only generates a possible path to the goal, not the optimal route (Koren and Borenstein, 1991; Prabler and Milios, 1990; Bonasso et al., 1992; Payton et al., 1990).

Hou and Zheng (1991) describe a new technique for navigation that uses the potential field method and a hierarchical cell decomposition techniques with a hexagon cell structure. Each cell is labeled as PASSABLE or IMPASSABLE. The PASSABLE cells are also labeled empty or mixed. Navigation is accomplished by moving to the cell with the lowest potential. If no sequence of empty cells can be found, mixed cells are

decomposed into smaller cells and relabeled until a safe path can be planned. Two methods were used to eliminate potential field problem of local minima. To prevent the robot from moving back and forth between the same two points, the potential of last point visited will be set to infinity for the next search round so it will not be selected as the next point. The robot can also continuously visit a series of points and get trapped. This is prevented by setting a point as IMPASSABLE after it has been visited twice.

Janabi-Sharifi and Vinke (1993) describe a hybrid navigation approach that combines the artificial potential field and simulated annealing methods for local and global path planning. This plan eliminates local minima problems. Another method that eliminates local minima problems is discussed by Reid (1993). The potential field map that is generated is a gray scale image with values ranging from 0 at the goal to 255 within the bounds of the obstacle. This map, formed by averaging images, should contain no local minima.

Payton et al. (1990) discuss the advantages of combining multiple potential fields. They cite the example of using two separate potential fields, one to keep the vehicle on a route and one to navigate to observation points. The potential field guiding the robot along the path would be the primary navigation source when an observation is needed. Then, the second potential field would be the guiding mechanism until the observation point was reached. The first potential field would then take over again.

#### 3.2.2.2 Navigation Templates

Navigation templates are a relatively new form of mobile robot navigation. This method was developed based on the idea that a new navigation plan should not have to be made every time the robot's perception of the world changes. Navigation templates incorporate the changes into the current navigation plan. This way, the navigation plan is constantly modified with different views of the environment (Slack, 1990). The idea for navigation templates "were inspired by the deliberate action that can result through the use of explicit navigation plan, the simplicity of the potential field and like approaches for describing plans of action, and the power provided by a situational characterization" (Slack, 1990).

Navigation Templates are similar to gradient summation navigation techniques where obstacles act as repulsors and goals act as attractors. Navigation Templates (or NaT's) are used to construct qualitative navigation plans. There are two types of NaTs: substrate NaT's (or s-NaTs) and modifier NaTs (or m-NaTs). S-NaTs are used to characterize the navigation task and m-NaTs are used to model obstacles and modify the navigation plan. S-NaTs assign a gradient at every position to determine the direction of travel for the robot. There are three types of s-NaTs: direction, position and trajectory. Position s-NaTs are where all of the vectors are directed at a single goal point. Direction s-NaTs are where all of the vectors point in the same direction. The third type of s-NaTs, trajectory, try to plan a path to the goal, but are not widely used (Gat, 1993). M-NaTs are obstacle descriptions that are used to modify the navigation plan of the s-NaT's. Each m-NaT

identifies a body, a spin and a safe distance from obstacles. The body is defined by a function that calculates the left and right extremes of the object. The spin, clockwise or counterclockwise, associated with the obstacle determines if the object should be avoided to the left or right. The safety zone is the area that the robot should avoid, but may enter if necessary to navigate between two closely spaced obstacles (Gat, 1993; Slack, 1993). There are four classes of m-NaTs: the object is not in the way with the given position outside the danger zone, the object is not in the way with the given position inside the danger zone, the object is in the way with the given position outside the danger zone or the object in the way with the given position inside the danger zone. Figure 3.12 shows the direction the robot is guided when navigating between two obstacles. The two m-NaTs and the s-NaT direct the robot between the obstacles.

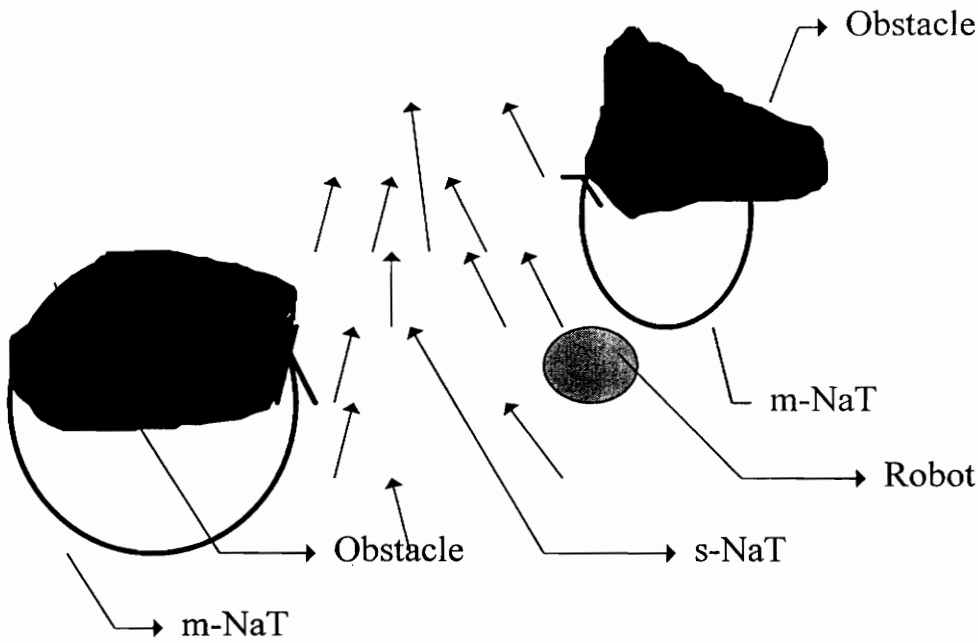


Figure 3.12: Navigation Templates

An s-NaT and a number of m-NaTs are used to create the navigation plan. A combining function is used to transform the NaTs into a preferred direction of travel for the robot. This is based on the geometric properties of the m-NaTs that relate them to the direction of the s-NaT. The preferred direction of travel is determined by first isolating the "immediate navigation objective". This objective satisfies the constraints of the m-NaT. The second step sets the initial values of the lower end of the range of angles. The combining function does not transform the NaTs into a gradient field, but computes the preferred direction of travel at the current position of the robot (Slack, 1993).

Problems with other potential field approaches are avoided by using navigation templates because the nature of the descriptions of the robot's relationship with obstacles (i.e., an m-NaT's spin) (Slack, 1990). Gat (1993) lists three major advantages of the NaT method. First, it does not suffer from the local minima problems of artificial potential field approaches. Second, reactive control can be used because the processing is very fast. Third, high-level control can be used with the spin assignment. The drawback of the NaT method is that an accurate sensing system that can produce a description of the environment in terms of symbolic descriptions of obstacles is needed on-board the vehicle (Gat, 1993).

Slack (1990) describes a local navigation system (LNav) used for navigation in a simulated environment. A sensing system is used to collect data and store it in a grid-based model of the environment that is constantly updated. Uncertainty due to sensor errors or obstructed areas is reduced by sensing the surroundings several times. Detected

obstacles are also given an expiration date because of the changing, irregular environment. When an obstacle is detected, an s-NaT is associated with it. A new m-NaT is also assigned to this obstacle. This new m-NaT is included in the navigation plan and the spin is determined by observing the shortest distance around the obstacle. This system is capable of reacting to changes in the environment by adding, deleting, or changing the features of obstacles. Slack (1993) implements the navigation template on an autonomous mobile robot. Sonar readings were taken to detect obstacles and group the obstacles into larger logical obstacles. The relationship between the robot, the goal, and each obstacle group is compared to assign a clockwise or counter-clockwise spin to the large obstacle groups to use for navigation.

Bonasso et al. (1992) use navigation templates for robot navigation with a reactive system. Sonar is used to determine the robot's position. This information is used to assign NaT's to obstacles and to develop a navigation plan to reach a goal position. The navigation plan is constantly updated with new sensor information, enabling the robot to adjust to changing environments. A NaT algorithm discussed by Gat (1993) operates in two phases. In the first phase, the constraints on the robot's motion assigned by m-NaTs are combined. This produces an estimate of the current direction the robot is traveling. The second phase uses the safety zones of the m-NaTs' to modify the determined travel direction so the robot avoids the safety zones, if possible.

#### 3.2.2.3 Vector Force Fields

The vector force field method is a method that applies an attractive force to the robot from the goal and repulsive forces from the obstacles. The forces are combined to get a resultant force that determines the steering angle and velocity of the robot (Beom and Cho, 1995). The development of this theory can be found in Borenstein and Koren (1989). This method is designed for real-time obstacle avoidance for fast mobile robots. The vector force field method can produce fast, continuous, smooth motion of the robot among unexpected obstacles (Koren and Borenstein, 1991).

The vector force field method uses a two-dimensional histogram grid with each cell containing a certainty value that is constantly updated by the sensing system (Koren and Borenstein, 1991). The potential field concept is applied to the histogram grid by identifying active cells that contain obstacles in the area surrounding the robot. Each active cell exerts a repulsive force toward the robot. All of the repulsive forces add to a resultant repulsive force. An attractive force is applied to the vehicle from the goal cell. The resulting vector gives the robot a new direction. This method is good for navigation in open areas, but won't allow the robot to pass through narrow passages and has instability of motion in narrow corridors (Borenstein and Koren, 1990b). Figure 3.13 shows an example of the calculation of a new steering vector using the vector force field method.

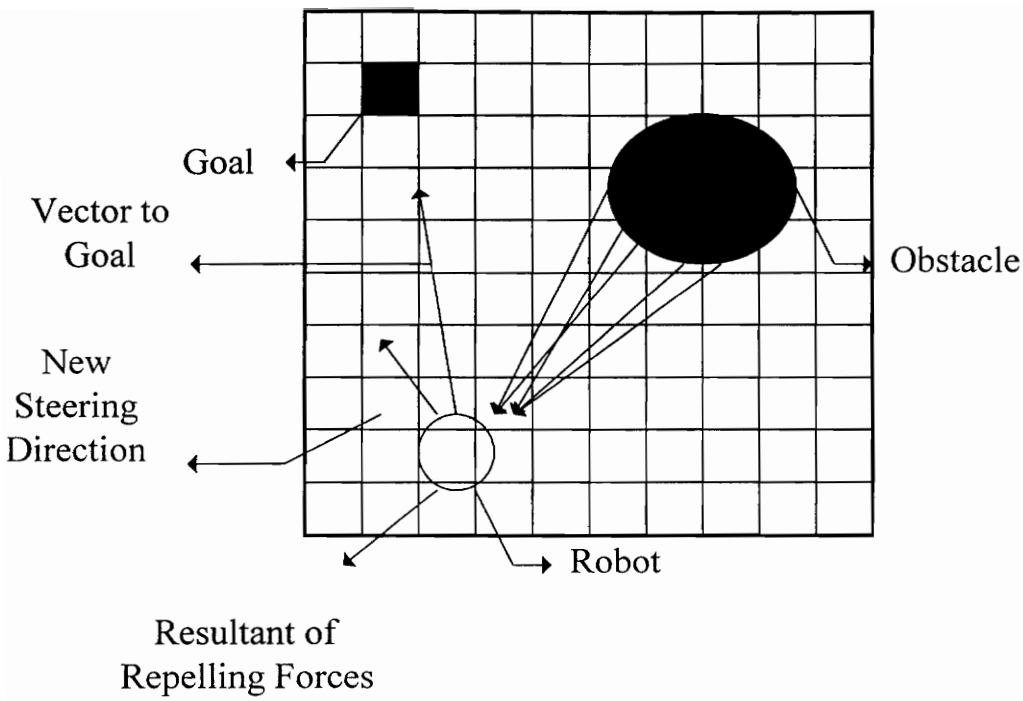


Figure 3.13: Vector Force Field

Schneider and Wolf (1994) use a Virtual Force Field (VFF) method for path planning on an autonomous mobile robot. Information about the surroundings are taken by ultrasonic sensors. This data is stored as a digital map. The path planner uses the weighted sensor readings for navigation. The vehicle can easily navigate in areas where there is no immediate goal or a preassigned destination point is given to the robot. The navigation of the robot consists of moving on the planned path and stopping to plan a new path.

El-Konyaly et al. (1995a) use the vector force field method for obstacle detection and avoidance for a mobile robot. The process is accomplished by using a video camera to obtain information about the environment. This information divides the grid into cells assigned a value between zero and one to represent the occupancy level of the cell.

Obstacles are represented by black spots on the grid and stored in memory. This information is updated with each new camera images. The obstacles in each cell of the grid are assumed to carry an electric charge that exerts a repulsive force on the robot. Usually, the magnitude of this force is inversely proportional to the square of the distance between the robot and the object in the cell. This representation produces a large repulsive force when the robot is near obstacles. This can be a disadvantage because it may cause the robot to veer far away from the goal when trying to avoid the obstacle. They propose using an exponential function for the magnitude of the force. This enables the robot to avoid obstacles only by the necessary distance.

A vector force field obstacle avoidance algorithm has been developed by Gourley and Trivedi (1994). By using only the current sensor readings, only the area immediately surrounding the robot is used for navigation. This algorithm produces a fast reaction time for the robot which is useful when objects quickly appear in the environment. Shahidi et al. (1991) present an algorithm to use geometric information about the environment to produce a vector force field for mobile robot navigation. This vector field method relies on the fact that charges of the same sign repel and charges of opposite signs attract. Using mathematical functions, a positive charge is assigned of the robot and obstacles and a navigate charge is assigned to the goal. This combination of charges will lead the robot to the goal while avoiding obstacles. Local minima problems can be avoided if the attractor is strong enough. The vector field is created by a weighted sum of the functions

assigned to the obstacles and the goal. By making the weighting factor on the goal high enough, the flow of the robot's path is toward the goal.

### 3.2.3 Neural Networks

Neural networks are a relatively new approach to navigation of autonomous mobile robots. The basis for an artificial neural network was formed by looking at "biological computers" - the nervous system and the brain. This lead to the design of intelligent systems based on neural systems. The discussion of one of the most popular neural network systems, multi-layer systems, can be found in Narendra and Parathasathy (1990). These models use neurons as the processing elements that receive data or signal inputs from other neurons. The network is formed by grouping together and connecting multiple neurons (Pandya and Luebbers, 1991; Penna and Wu, 1994). "To achieve full autonomy requires not only the ability to train individual expert networks, but also the ability to integrate their responses." (Pomerleau, 1994) Neural networks are function approximators that map the input to the output. Figure 3.14 is an example of how a simple neural network system is arranged. The two input nodes on the left are transformed into the five output commands on the right. The middle layer is called the hidden units the network. The number of hidden units corresponds to the dimension of the function that is being fit. If a high dimension data function is used to fit data that would be better represented by a low dimensional function, it will be unable to

extrapolate from or would interpolate between the given data points (Davis and Stentz, 1995). It can be seen that all decisions are highly interconnected.

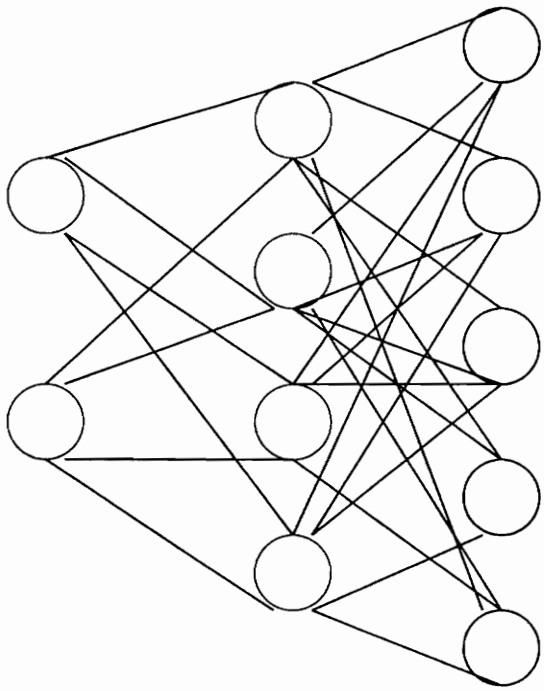

Figure 3.14: Neural Network Architecture

Neural networks can be trained to identify situations by showing the system many examples from numerous classes. The robot then uses these stored examples to navigate in different environments. Neural networks have been used for several tasks, but pattern classification and recognition with vision systems has been the most common. The ability of a network to adapt, learn and integrate information from several input sources makes neural networks a good method for robot navigation (Pandya and Luebbers, 1991).

Artificial neural networks have several advantages over other navigation strategies. One advantage is their ability to deal with scene variations such as changes in illumination. Another advantage is that the neural network can recognize scenes even with low resolution images. This makes the system faster and can cut the cost of an expensive sensing system. Other environment problems, such as noise and variability are less of a problem for neural networks. Neural networks use simple, interconnected processing elements rather than the complex programs used by traditional navigation approaches. This enables complex operations like coordinate transformations, trajectory control, inverse kinematics and inverse dynamics can be performed implicitly, since the neural nets transform the inputs into desired outputs non-algorithmically, using the stored pattern of weights. Complex nonlinear control functions can also be learned without explicit programming by updating the weights in response to an internal or external teacher (Pandya and Luebbers, 1991). The biggest advantage of neural networks is their ability to adapt to new situations if the vehicle has been trained on a similar pattern. That means that neural networks can identify scenes with just a few distinct features. This enables the trainer to only use a few images to train the network and the system is able to design a control system with a complex environmental model (Inigo and Torres, 1994; Pandya and Luebbers, 1991).

The biggest drawback to neural networks is that they can not create new paths if the robot gets trapped unless they have specifically been trained for this situation. Because only a certain set of images are used to train the robot, the robot is unable to safely

navigate through an environment with different features. Training the network can also be a problem. Multiple situations have to be taught to the vehicle to navigate in a simple environment and the number of training sets greatly increases for more complex situations. This training can be a slow process and it can be difficult to develop a system to interpret the information (Pomerleau, 1991).

Banta et al. (1993) discuss the implementation of neural nets on board an autonomous navigation testbed for several applications. The first neural net developed was an area classifier. This neural network was able to identify different type of situations that the robot might encounter while navigating in an underground mine. Some of these situations include a crossing, left turn, right turn, and a split to the front, right, or left. This is accomplished by taking the data from ultrasonic sensors and sending it to a single-layer network. The output is "clustered" together and identified as a certain situation. A second neural network was used for position estimation. This system would take the output from the sonar sensors and compute the distance and orientation of the vehicle from the centerline of the hallway. This application of neural networks proved to be difficult to configure and train and was sensitive to the length of the hall. Because of these problems, Banta et al. (1993) concluded that neural nets did not have any advantage over using strictly geometry-based calculations.

A landmark recognition system was also developed by Banta et al. (1993) using a single video camera and neural networks. Images were taken by the camera and processed to recognize several images. Additional neural networks were used to track the

recognized landmarks for position estimation relative to the landmark. At least two images from different locations are needed for an accurate position estimate. A neural network controller is used to keep the camera pointed at the landmark during motion. The inputs are the present and past locations of the landmark image, the current pan and tilt angles, and the present and past two x and y positions and orientation. Two separate networks were designed to control the pan and tilt of the camera. They are both feedforward networks that give the number of degrees to move the axis in each time step.

The last neural network system incorporated on the vehicle was used to produce a polynomial approximation of the probability density function of the noise. Two cascaded neural nets were used. The first separates the noise by magnitude and outputs the node whose mean is closest in magnitude to the current noise sample, and those to either side of it. The second network creates a histogram of the outputs from the first net. "The outputs from this layer are fed back to its input with a weight of less than one. If stimulated by an input, the output of this memory layer goes high, and then slowly decays if not stimulated again within the time constant of the memory. The composite output of the second layer is a set of coefficients for the noise probability density function descriptor. The descriptor is a polynomial whose terms are the same exponential functions used as the actuation functions in the first layer network." (Banta et al., 1993).

The neural networks worked well on pattern recognition problems, such as the Neural Area Recognition system, the Landmark Recognition system and the Noise Distribution Function Estimator. Difficulties were encountered training these networks, but the

performance of the networks was worth the long training process. In other applications, such as the early steering control system and the Neural Position Estimator other methods could have achieved better results.

Siemiatkowska (1994) uses a neural network for path planning. A map is created defining each cell as free or occupied; the uncertainty for this label is also determined. Each cell corresponds to a neuron for navigation purposes. The neighbors to each neuron are determined and grouped in "neighborhoods". The neuron that corresponds to the global position of the robot is the source of activation. Neurons are activated if the cells they represent are not occupied and occupied cells do not activate their corresponding neurons. All neighbors are checked and the activation process continues until the neuron representing the current location of the robot is activated or no neurons change state. "The first condition means the completion of the first part of the path planning, the second one means that there is no free way between the robot and the goal" (Siemiatkowska, 1994). The path planning process was completed by checking the cells neighboring the robot and moving to the one with the highest level of activation. This neuron activation is then set to zero so the robot will not backtrack. This process continues until the robot navigates to the goal or until the neurons remaining have an unknown status causing the path to be only partially planned. New paths are generated with new maps formed by updated sensor information. This path planning process has advantages over potential field approaches because:

1. This method is highly parallel.

2. An optimal free path can be found in an unknown, changing environment.
3. The robot can recover from a dead end by backtracking without needing any additional information and can automatically find a new path
4. The algorithm can recognize is situation if there is no free path between the robot and the goal.
5. This approach doesn't suffer from local minima.

Kim et al. (1995) have developed an autonomous mobile robot for road following using neural networks. The system uses a single camera for input data. The camera data is processed so only two window strips are sent to the neural network. The entire camera image would make the system slow and require extensive training of the neural network. This information is sent to the Intersection Detection Neural Network (IDN) which identifies the intersection by checking examples. If the vehicle is at an intersection, the path planner uses the road map to determine the direction the vehicle controller should follow. If the vehicle is not at an intersection, the vehicle proceeds in a straight line. "The path planner simply decides what to do based on the road map knowledge. The vehicle velocity is automatically slowed down to a predetermined value near intersections and otherwise kept to a constant value" (Kim et al., 1995). The Steering Control Networks were trained while driving the vehicle by remote control. The camera data is the neural net input and the remote control commands are the desired output. The Intersection Detection Network inputs are the preprocessed intersection/non-intersection images. The desired outputs are positive one for an intersection and negative one for a

non-intersection. The steering control network learns to "produce this steering when the road looks like this". A neural network straight, NNS, system is trained by looking at scenes while driving through intersections or by following a road in places other than intersections. The neural network left, NNL, and neural network right, NNR, are trained while turning left and right at intersections, respectively. "After learning, NNS, NNL and NNR look at the input scene and generate proper steering commands to achieve desired maneuvers." (Kim et al., 1995)

Inigo and Torres (1994) describe a neural network approach that also used one camera. The networks use translation and perspective information of the images to determine the position and proper alignment of the robot. Three different neural networks, called modules, are used for the mobile robot navigation command: the alignment module (AM), the location module (LM), and the obstacle recognition module (ORM). Their actions are interrelated and combined into a single command. The AM command consist of turn left, turn right, continue forward, and stop. The AM is used to properly align the robot using translation information. The LM contains information about the coordinates of the robot referenced by the home position to estimate the position of the robot. The abscissa and ordinate parameters are set if the robot is aligned along the path. This minimizes the effect of translation and identifies if the robot is nearing the end of a hallway or corner because the features will become larger in successive images. The LM is trained to identify the position of the robot after turning the corner. This enables the robot to identify different halls and position itself in a new location. The ORM sets an

"object present" parameter equal to zero if the area is clear and one if and obstacle is detected. The ORM is trained to recognize objects by purely neural-network pattern recognition.

A navigation technique discussed by Thorpe et al. (1991), ALVINN, is also based on neural networks. ALVINN is trained by a person steering the vehicle over part of the road to be followed. The inputs to the system are the preprocessed camera image and the steering angle. This information is used by a back-propagation algorithm that adjusts weights in the hidden units of ALVINN's neural network until values that produce the correct steering response for each image are found. This training session usually requires less than 100 input images and takes less than five minutes. If the robot became misaligined on the road it would be difficult to bring it back to the center. To prevent this many images from different angles were used to train the network so it would be able to recover from errors. ALVINN uses no reasoning and goes straight from camera images to a steering direction with no intermediate geometric or symbolic representation. This enables the system to learn different sets of weights to follow different roads without changing the system. This method has the disadvantage of not being able to use geometric or symbolic input. As an example, the robot is unable to adapt to a situation where it is told that a second road is just like the first, only twice as wide. The advantage is that it is fast and easy to train for a particular road. The learned weights are based on certain features, meaning that the steering output is not affected by lighting or road imperfections.

A system using multiple neural networks is discussed in Pomerleau (1991). Using several networks makes the training process faster because each network is trained on a different aspects of the task. Pomerleau (1991) proposes that each network can be trained in under five minutes for single-lane road driving, highway driving, and collision avoidance. Training "on-the-fly" has been developed to teach each network as a person drives the vehicle. The camera image is the input and the steering direction is the desired output. Several different camera angles are also used during training so the vehicle will be able to recover from errors.

The first step in this navigation process is to obtain data from the on-board sensors which is sent to the driving networks. "The driving networks propagate activation forward through their weights, with each determining what is considers to be the correct steering direction. These steering directions are sent to the arbitrator, which has the job of deciding which network to attend to and therefore how to steer" (Pomerleau, 1991). To aid in the steering decision, a mapping module is also used to provide information about the road types, location of intersections, and permanent obstacles. It recommends a steering direction and gives information about the current driving situation. An inertial navigation system and symbolic landmark recognition system are also used to keep track of the position of the vehicle on a map. An arbitrator uses the information about the capabilities of the driving module and the present terrain to decide what module to use. An obstacle avoidance network is also used. If the area is clear, the network is trained to go straight. If an obstacle is present, the vehicle will swerve to miss it. "By combining

map-related knowledge about the current driving situation with knowledge about abilities and priorities of individual driving modules, the integrated architectures provides ALVINN with capabilities that far exceed those of individual driving modules alone" (Pomerleau, 1991).

This integration of multiple expert networks has the advantage over other plans because is that it is easy to add new modules to the system and can be easily trained in a new area. A disadvantage is that this system relies too heavily on the mapping module. This is not a good solution because accurate information about the road and the position of the vehicle are needed. Another drawback is that each module is suited to only one situation and can not be adapted to other conditions.

NEURO-NAV, an autonomous mobile robot discussed by Meng and Kak (1993) is a neural network system that mimics human behavior. Several small networks were used to train the robot instead of one big network for three reasons. The first was to create generic behaviors by not requiring several networks to agree before the robot could move. Second, smaller neural networks can easily be used in the decision-making process by showing immediate results. A large network often produces an output that can not be influenced by other intelligent agents. Third, smaller neural networks are easier to train.

The design of NEURO-NAV was based on three principles of human behavior. First, humans can reactively navigate based on past experiences and do not need to know the exact coordinates of their position. Second, a person can reach a goal quickly based on a

visual input. Third, information acquired while moving can be stored for other applications.

NEURO-NAV uses these principles for navigation. An exact three-dimensional model of the environment is not needed for navigation; a simple geometric model is all that is required. This map consisting of corridors, junctions, dead ends, and landmarks (e.g., doors, bulletin boards) is used for path planning. Using this model, the controller can issue commands to the robot, such as, "go down corridor A and turn right at the next junction." "Given these commands, the robot travels on a planned path using its ability to perform primitive navigational tasks such as hallway following, i.e., navigating down the hallway without running into walls, and landmark detection" (Meng and Kak, 1993).

## 3.3 Summary

Mapping and navigation are serve as the brains of the robot. By taking the information from a sensing system and creating a map of the environment they formulate the enables a navigation plan. This plan is what determines the success of the robot in reaching its goal. Tables 3.1 and 3.2 lists the advantages and disadvantages of the mapping and navigation procedures described in this chapter.

Table 3.1: Mapping Comparisons

| Mapping Method | Advantages | Disadvantages |
|---|---|---|
| Full Scene Maps | accurate, good for obstacle and boundary line detection | computationally expensive, not all of the obtained information is needed |
| Edge Image Maps | easy to produce, less computer memory is needed | susceptible to sensor error, robot has to stop and think |
| Feature Maps | needs little computer memory | susceptible to sensor error, only shows certain features of an obstacle |
| Certainty Grids | does not need exact sensor data, accurate | computationally expensive, poor for real-time control |
| Histogram Grids | good for real-time control, moving objects are better represented | only good when the robot is moving, areas around objects have higher certainty values than they should |
| Topological/Elevation Maps | accurate, good for outdoors | computationally expensive, not all of the obtained information is needed |

Table 3.2: Navigation Comparisons

| Navigation Strategy | Advantages | Disadvantages |
|---|---|---|
| Cell-based Map Searching | easy to implement, needs a simple sensing system | must stop and think to update the map |
| Vector Field Histogram | can handle random sensor errors, data is available at different levels | can fail by the "trap problem"- dead end |
| T-Vectors | easy to add and remove obstacles | hard to implement, can't be use for real-world control |
| Artificial Potential Fields | fast, efficient, uses only current information | local minima problem, doesn't give the best path |
| Navigation Templates | fast, no local minima problem | needs a very accurate sensing system |
| Vector Force Field | produces fast, continuous motion, real-time control | large forces around obstacles can send robot off the path |
| Neural Networks | produces good results for a set task | can be hard and tedious to train, can not adapt easily to different situations |

Mapping and navigation provide the information needed to control the mechanical systems of the robot. How this information is used for motion control is discussed in Chapter 4.

# Chapter 4

# Motion Control

The last task of an autonomous mobile robot is "acting". This is classified as the motion control of the vehicle. Motion control is the function that either allows the robot to follow a previously generated path or moves reactively based on sensor input. This action is used to safely navigate in an environment while avoiding any unexpected obstacles. Gat (1992) cites three reasons the motion control of an autonomous mobile robots can be difficult. First, the robot must be able to react quickly to changes in the environment. "Elevator doors and oncoming trucks wait for no theorem prover." Second, many featured of the surroundings are unpredictable. This makes it impossible to generate a complete plan in advance. Finally, sensor information can be incomplete and inaccurate, making correct motion planning more difficult. "These are fundamental problems because they cannot ever be engineered away." No matter how powerful a computer we build, a finite amount of time will allow only a finite amount of computation. No matter how good a sensor we may build, there is always information that it cannot deliver because the relevant situation is hidden behind a wall or across town. No matter how good our domain theory may be, many important aspects of the world simply cannot by predicted reliably" (Gat, 1992).

This chapter presents two approaches to motion control: conventional control and fuzzy logic. Conventional control methods used open or closed-loop systems to produce the desired motion of the robot. Fuzzy logic is a new control method based on the way humans react to certain situations. This type of control is called intelligent control. The use of neural networks on mobile robots is also often classified as an intelligent control strategy. Neural networks were included in the navigation discussion instead of the motion control chapter because they are most often used to recognize a certain situation and produce a pre-determined motion. This means that the robot can't actively plan different paths if it becomes trapped.

There are other factors that affect the motion control system. These include constraints placed on the generated trajectories, errors in the system, the cost of traversing a selected path, and the drive system used on the vehicle. The influence of these factors will be discussed following the material on control methods. The chapter will conclude with a brief comparison of control techniques.

## 4.1 Conventional Control

Conventional motion control techniques for autonomous mobile robots are either open-loop or closed-loop. An open-loop controller simply takes the input to the system and applies a control law to produce the desired action. No information is passed back to the controller regarding errors that could make the current status of the robot different

than anticipated. A closed-loop system uses either a feedback or feedforward controller to constantly check the system for errors.

Most autonomous mobile robots have a multi-level approach that uses a combination of open-loop and closed-loop control for the different systems on-board the vehicle. Even if an open-loop is used for one function, the motion of an autonomous mobile robot is a closed-loop system with feedback coming from the sensing system. The discussion presented on open-loop and closed-loop control is focused only on the motion control of the robot.

### 4.1.1 Open-Loop Control

Open-loop control systems carry out the motion of the robot without feedback on the current status of the vehicle. This is an ineffective form of navigation in complex environments. The absence of feedback means that the robot does not know if the desired commands are being completely carried out. Figure 4.1 shows a typical open-loop control structure.

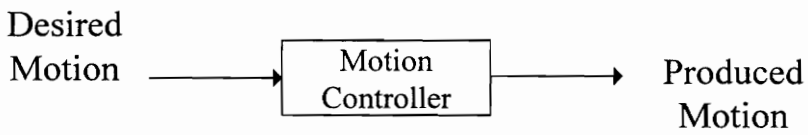


Figure 4.1: Open-Loop Control Structure

Even though this type of control is not useful in complex situations, it is still used for simple motion control problems. Lee et al. (1993) present a motion control plan based on three simple commands: go straight, turn, and spin. A robot with two differentially driven wheels is used. Once the robot observes the scene, a command is carried out. The "go straight" command just continues to move the robot on its current course. The robot is commanded to "turn", the desired steering angle is given and the robot moves until it is at that position. When the robot "spins", the position of the vehicle does not change. The robot will pivot around its center until the spin angle is reached. Given the initial and final positions of the robot, the robot generates a "smooth function", a continuous mathematical function, to move to the goal.

Gat (1992) uses open-loop control using three levels: a controller to supervise the non-decision making actions, a mechanism for making decision computations, and a sequencing system to control the actions of the other two systems. The three systems developed are a controller, sequencer and a deliberator. The controller reacts to sensor information and actually implements the response in hardware. The sequencer controls the actions of the controller and the deliberator. The sequencer can activate and deactivate modules in the controller and send it parameters to be met. All computations are also initiated by the sequencer. These computations are actually performed by the deliberator. The connection of these three systems controls the motion of the robot.

The steering control of the mobile robot described in Stentz et al. (1993) uses a "pure pursuit algorithm". Motion control is carried out by selecting a point on the desired path

at a fixed distance in front of the robot. A path consisting of a single, circular arc is then calculated. After part of the arc is driven, a new path is created. This system works well unless the desired path changes abruptly.

The steering system used by Stentz and Hebert (1995) combines the recommendations from several modules and combines it into one drive command. Each module "votes" from negative one to positive one on a set of arcs for the vehicle to travel. A value of negative one means the arc should not be driven and a one means that it is a desired path. A weighted sum of the votes from each module are combined to pick the arc with the highest vote to use for a steering reference.

### 4.1.2 Closed-Loop Control

Closed-loop control strategies are the most common form of conventional control used on mobile robots. These methods can use a feedback approach to compensate for the errors in the motion and/or a feedforward compensator to help predict accurate commands. A proportional control loop is a common closed-loop strategy. Proportional control is "a control method in which the drive signal to the actuator increases monotonically and proportionally with the increased difference between the actual output and the desired output" (Critchlow, 1985). Another popular closed-loop control method is proportional integral derivative control (PID). This is "a control method in which the actuator drive signal is generated by a weighted sum of: the difference between the actual output and the desired output; the time integral of the difference and the time derivative

of the difference" (Critchlow, 1985). Variations to this control strategy may just use the integral or derivative functions resulting in PI or PD controllers. A simple closed-loop control system is shown in Figure 4.2.

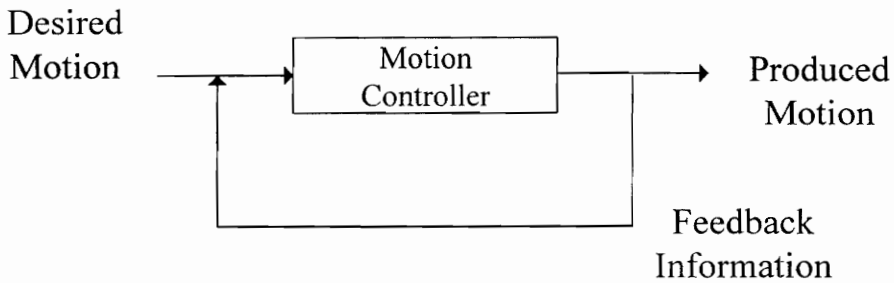


Figure 4.2: Closed-Loop Control Structure

Since this conventional control structure is widely used and can have several variations, several examples of the implementation of this system are presented. A motion control system developed by Shin et al. (1991) separates the steering and velocity control of the robot. A steering planner that uses both a feedback and a feedforward compensator is used. This design was used because, using just feedback control, errors can only be corrected after the fact. The feedforward compensator used a model of the vehicle dynamics and the future path of the robot to plan the next actions of the vehicle. The feedback compensator allows closed-loop compensation between the actual and desired paths of the vehicle. The initial steering command is given by the feedforward compensator. As the robot is moving, the feedback compensator begins correcting the angle. The minimum speed of the vehicle is selected by checking three constraints: the

maximum speed limit in an area, the distance to obstacles and the goal, and the maximum allowable lateral acceleration.

Iida and Yuta (1991) describe the design of a feedforward compensator for steering method using two driving wheels. A feedforward control scheme is used instead of the traditional feedback plan because it is difficult to stabilize the vehicle and continue on the path when the velocities of both wheels interact dynamically. The feedforward compensator uses the inverse vehicle dynamics. The angular velocity of the wheels is determined and converted into the motor torques. A PI feedback loop is then used to correct for errors and noise.

The motion control on the robotic vehicle described by Fernandez et al. (1994) used video image feedback to generate steering angle and velocity commands. The robot plans a steering direction at a desired velocity based on the most recent video information. Two independent control systems are used to calculate the speed and direction. The speed can be set to one of four values: maximum, medium, minimum and stop. The speed is selected based on the distance from the vehicle to an obstacle. There are thirteen steering angles, from -60° to +60°, that can be selected. After a safe distance for the robot to travel is computed, one of the thirteen steering angles is selected. The path of the robot can be modified by constantly checking the maximum safe distance the robot can travel in each steering direction.

El-Konyaly et al. (1995a) have designed a cross-coupling controller for a robot using two driving wheels, to improve on the conventional control techniques. Motion control is

used to generate speed and steering commands that are functions of the drive motor speeds. With conventional control methods, each wheel is in a separate control loop. This method does not allow for each wheel to help compensate for errors in the other wheel. A method using four control loops is presented to correct for this problem. Two independent, conventional closed loop controllers are used to command the velocity of each drive. An additional two independent, control loops are used to determine the speed and orientation of the vehicle.

A mobile robot described by El-Konyaly (1995c) has two front driving wheels and two rear non-driven wheels. One motor is used for speed generation and another is used for steering control. The speed and steering direction of the robot are controlled separately by closed-loop systems. A PI controller is used for speed determination and a PID controller is used to control the steering angle.

Muller (1992) presents a control scheme for steering an autonomous mobile robot around curves using feedback and feedforward controllers. A model of the environment taken from a black and white video camera is used to estimate the steering direction of the vehicle. This direction is given to a feedback controller to produce the correct mechanical motion. When the vehicle is to go around a curve, a feedforward control program takes over. This program is then used to generate the steering angle of the robot. After the turn is complete, the feedback controller takes over again.

The motion control system used by Kim et al. (1993) consisted of three separate controllers. A motor driver controller commands the acceleration and steering motors. It

also communicates with the speed controller. A brake controller pushes or pulls the brake pedal using air pressure. The speed controller determines the speed of the vehicle using an encoder and adjusts the vehicle to the desired speed by sending commands to the brake and motor driver controllers.

Nowinski et al. (1994) designed a conventional controller that takes the distance of the vehicle to a boundary, in the form of visual input, and converts this information into a steering command. A potentiometer was used to provide feedback information on the actual steering angle to a proportional controller. The distance to the boundary line was also measured and fed back by a PD controller. This enabled the vehicle to stay away from the boundaries.

Kim et al. (1995) use a three-part motion control system on an autonomous vehicle. First, a reference trajectory generator produces the desired motion path. It also sends the reference position, reference linear velocity, and reference angular velocity to the motion controller. The speed for each motor is then computed as a group of set points. The motor speed controller adjusts the motor to match the set points. A closed-loop system using an inner current-control loop and an outer speed-control loop is used. The current loop regulates the current driving the motor. The speed control loop uses PID control that reads an encoder and provided feedback information to adjust the speed of the vehicle.

Two main forms of control were used on an autonomous mobile robot developed by Santos-Victor et al. (1995). A navigation control loop is used to change the heading

direction of the robot by applying a rotation speed. A closed-loop control system using a PID controller keeps the robot on the desired path. The velocity is calculated based on the width of the path. If the vehicle was in a narrow space, the speed would decrease. In wide areas, the robot would go faster. This speed was easily adjusted based on the observed surroundings.

Conventional PID control was used by Koskinen et al. (1993) on an autonomous vehicle. This method worked well, except as low speeds "where the centripetal clutch causes strong non-linearity." The steering angle is updated by closed-loop position feedback. Feng and Krogh (1991) use a feedback strategy for conventional steering control of a mobile robot. A desired steering vector is generated at each time interval and used to generate a reference trajectory for the steering angle and wheel velocity. These commands are constantly updated using the feedback loop and new sensor inputs.

## 4.2 Fuzzy Logic

Fuzzy logic is another method used for motion control. Fuzzy systems are based on an intelligent control strategy designed to simulate human performance in the same situation. The introduction of this method can be found in Zadeh (1973). Humans use fuzzy logic when they are driving or parking a car or looking for obstacles. These are actions that are flexible, robust, imprecise and intelligent because they can be modified for different circumstances (Brown et al., 1991). McCarthy and Trabia (1994) explain the

principle behind fuzzy logic by relating its operation to driving a car. If you are driving the car, and the car deviates a "small amount to the left of the target, the normal response is to turn the steering wheel a small amount to the right. If the car starts to deviate a large amount to the right of the desired target, the normal response would be a large turn of the steering wheel to the left." Fuzzy logic controllers implement the same ideas by being able to adjust to changes in the system.

A fuzzy controller usually has four main components: a rule base, a fuzzy inference mechanism, an input fuzzification interface, and an output defuzzification interface. The rule base contains a set of IF-THEN rules , expressed in linguistic form, that have been established to be needed for the task of the robot. This information is used by the fuzzy inference mechanism. The fuzzy inference mechanism makes decisions about what rules are needed and performs the actions signified by the rules. The input fuzzifier takes the numeric inputs to the system and converts them into the fuzzy form needed by the fuzzy inference mechanism. Drawing on the information in the rule base and the inputs from the input fuzzifier, the fuzzy inference mechanism passes the desired output action to the defuzzification interface. The resulting output is a numeric value that is produced by the output defuzzification interface. This value is obtained by combining the decisions made by the fuzzy inference mechanism (Passino, 1995). This process is shown in Figure 4.3.

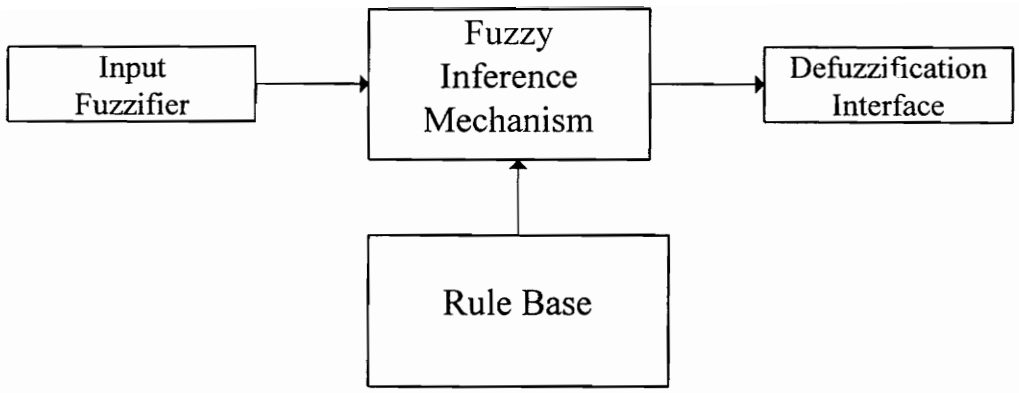


Figure 4.3: Fuzzy Logic Control Process

Passino (1995) uses the example of a cruise-control system on a car to show typical fuzzy rules. The rule base would contain a set of IF-THEN rules of this general type:

IF e(t) is positive-small and de(t)/dt is positive-medium, THEN u(t) is positive-medium

IF e(t) is positive-small and de(t)/dt is negative-medium, THEN u(t) is positive-small

where e(t) is the error between the desired and the actual

inter-vehicle spacing and u(t) is the throttle input to the follower vehicle

The first rule means that the actual and desired distance between the lead and follower vehicles is about the same, but the distance error is increasing at a "medium" rate. This produces an output to increase the acceleration by a "medium" amount to close the gap. The second rule also has a small error between the actual and desired distances of the vehicles. Since the distance is decreasing at a "medium" rate then a "small" speed value should be used. The rule base uses linguistic terms like "negative-medium", or "near" and "far" to produce the desired action. This term is defuzzified to produce the corresponding output. The defuzzification process involves looking at the linguistic

result and comparing it to a numeric equivalent. Rules can be combined and used to produce a smooth motion output.

The biggest advantage of fuzzy logic controllers is that a model of the environment is not needed (Beom and Cho, 1995). Fuzzy logic systems allow for a motion control strategy to be modeled using linguistic terminology. This reduces the computational requirements of the system and allows for real-time navigation. New rules can also be easily added to the system. In addition, because of the reactive nature of a fuzzy controlled robot, it is possible for the robot to wander around an unknown environment and map the area (Tunstel and Jamshidi, 1994).

A disadvantage with fuzzy logic controllers is that approximations that are made regarding rule formulations and variable values cause the system "to converge to a stable domain (rather than a point) around the set point" (Brown et al., 1991). This can produce an unpredictable and potentially oscillatory response. This problem can be reduced by making fewer assumptions, but this requires more memory and makes the process slower. If precise control is needed around a set point, Brown et al. (1991) suggest using a local or nested PID controller whose gains are modified using the fuzzy model.

Beom and Cho (1995) propose a new navigation method that uses fuzzy logic and reinforcement learning. This strategy is proposed to eliminate the problems of conflicting results produced by the obstacle avoidance and goal-seeking behaviors of a system. Fuzzy logic and reinforcement learning behaviors are independently used for motion control and mapping, respectively. The navigation strategy used has five parts:

fuzzification of the input/output variables, rule base construction through reinforcement learning, reasoning process, defuzzification of output variables and a behavior selection scheme. The first step of the process, fuzzification of the input/output variables, is accomplished by taking the sensor data and converting it to linguistic values by using a fuzzy operator. These values express the location of obstacles with respect to the vehicle, VN - very near, FR - very far, etc... Each fuzzy set is assigned a real number between 0 and 1 to every element. This value indicates the degree that the element belongs in this fuzzy set. The rule base is constructed using human behavior. This is a set of IF-THEN rules used to control the robot's actions. The reasoning is carried out by checking each rule base to control the actions of each behavior. The variables are defuzzified to determine the output actions. The behavior rules are then learned by moving a small distance and recording its environment. Situations are remembered and can be recalled when similar situations are encountered. Obstacles are avoided while progressing toward a goal point by calculating the time needed to avoid the obstacle based on the distance from the center of the mobile robot to the obstacle and the velocity component toward the obstacle.

Martinez et al. (1994) use fuzzy logic in a reactive control system for obstacle avoidance. The reactive system is implemented by using a sense-act strategy instead of the traditional sense-plan-act strategy. The robot is constantly processing sensor information on the current state of the environment as the robot is moving. If an obstacle is detected, the fuzzy logic controller is used to modify the steering and speed commands.

This behavior enables the robot to avoid obstacles without an explicit map of the environment, greatly reducing the computational demands on the system. Saffiotti et al. (1993) describe a reactive controller that uses fuzzy logic, reactive behavior and goal-oriented behavior for a control scheme on a mobile robot. This system identifies the goals for the robot and determines the best actions to reach that goal. Each point along the way to the goal must use a certain behavior. The behaviors are then combined using fuzzy logic to produce a smooth sequence of actions for navigation.

Aoki et al. (1994) use a three-level fuzzy algorithm to control the velocity and steering of a mobile robot. The lower level decides the control inputs of the velocity and steering. A fuzzy balancer adjusts these inputs so they will not conflict in the middle level. The upper level combines the control input to the goal and the multicontrol inputs for multiple obstacles to move the robot. The velocity and steering signals are calculated by first checking the sensor input for obstacles. If no obstacles are detected, the robot moves toward the goal with increasing speed. If an object is detected, a fuzzy variable called the degree of static danger is calculated. This variable is based on the relative angle between the robot and the object and the distance to the obstacle. If the degree of static danger is high, the velocity is increased or decreased based on the fuzzy parameters of the distance to the obstacle and the estimated time for the robot and the obstacle to collide. The steering angle is determined by a decision table using the degrees of static and dynamic danger of the robot. The fuzzy balancer combines the velocity and steering commands. The steering angle remains the same and the velocity is reduced if an obstacle is in the

front or back of the vehicle. If an object is detected on one of the sides of the vehicle, the velocity remains constant and the steering angle is adjusted. This method of motion control only adjusts the steering control in the fuzzy balancer. The final steering input is a weighted sum of the steering commands needed to avoid all of the obstacles and navigate toward the goal. The velocity is determined by the degrees of danger.

A fuzzy steering controller is used by McCarthy and Trabia (1994). The steering needed to reach the goal is determined by the steering angle and the vehicle orientation correction. The time to reach the target is to be minimized by the fuzzy controller. The distance to the goal is used to stop the vehicle once it gets within a pre-defined distance to the target. Another fuzzy-logic-based controller is discussed in Liu and Lewis (1994). Using fuzzy rules, the robot was able to do three maneuvers: approach-the-goal, docking, and staying on a path. The steering strategy for the approach the goal procedure was very simple. A subgoal somewhere on a generated circle around the obstacle was determined. Based on the sign of the angle generated between the robot and the subgoal, the steering was either adjusted to the left, right, or remained the same. The docking maneuver took place after the robot had moved to the subgoal on the circle around the obstacle. The new steering angle was selected based on how far the robot was from the goal. The last fuzzy logic controller was used to keep the robot on a path. The robot could stay a pre-defined distance from the left or right of a sidewalk, or keep off both sides of a sidewalk. Based on information from a sonar sensing system, the robot would follow the selected path by modifying its steering angle.

Cardoso et al. (1994) also describe the use of a fuzzy logic steering controller. The controller is intended to keep the robot following a white line on the floor that is recognized by the vision system. The fuzzy controller contains four systems. A fuzzification interface takes the turn angle and speed recommendations and converts them into linguistic fuzzy input variables. A knowledge base contains the data that defines the input/output fuzzy sets and the fuzzy rules for the control strategy. A decision logic system then applies the fuzzy rules to the input variables to obtain the output values. The defuzzification interface then converts the output fuzzy sets into output signals for controlling steering and speed of the robot.

Siemiatkowska (1994) uses a fuzzy logic controller for motion control. This controller is designed to determine the move and turn functions of the robot. A set of fuzzy rules are used to interpret the current status of the robot and change the steering or speed commands. The appropriate action is determined by taking the input to the controller and checking all of the rules to find the best output. This output is then defuzzified to produce a real-value output command to the robot.

A motion control scheme used by Burke and Rattan (1993) uses three layers of fuzzy logic control to command the robot to drive straight, turn left, or turn right in continuous motion. The layers are a protection layer, an orientation layer, and a proportional-plus-derivative layer. The protection layer control is used to keep the robot from running into walls. This is carried out by taking the measurements made by the sensing system and using fuzzy logic rules to move the robot away from the walls, if needed. Orientation

control is used to keep the robot in the direction of the goal. Depending on the sensor input and the difference between the current direction of the robot and the position of the goal, the steering of the robot is updated to be in line with the goal. There are three modes that can be selected to provide the input values to the PD controller. A centering mode is used when two walls are detected and the robot moves straight between them. The wall-hugging mode is used when only one wall is detected or the robot is commanded to turn. In the blind mode, there isn't enough information to direct the robot. A PD controller takes the values for the error and the change in error and attempts to drive them to zero. The position of the robot is updated using fuzzy rules.

Knick and Schlegel (1994) use a fuzzy logic controller to compute the steering angle of a mobile robot. A "driver" module generates the path of the robot to the final goal and previously determined intermediate positions. The method used "is comparable to a rubber band which pulls the robot to the destination point." Using this fuzzy logic approach eliminates the need to solve the inverse kinematic problems of the system. Continuous updates can also be produced by continuously passing subgoal points from the path planner to the fuzzy controller.

A fuzzy controller for obstacle avoidance is used by Vandorpe and Van Brussel (1994). The fuzzy controller used the input from the sensing system as inputs for the controller. The fuzzy rules that are generated are based on the distance of the robot from an obstacle. These rules then determine the orientation and velocity of the robot. Wu et al. (1995) developed a fuzzy controller to drive a vehicle to follow a lane and keep it in

the middle of a lane. A table of rules was developed for input to the fuzzy logic controller based on the lateral error and its first difference. This control method proved to be superior to conventional PID control.

Baxter and Bumby (1995) used fuzzy rules to develop guidance plans for a mobile robot. These plans would enable the robot to navigate in an environment with changing conditions. The goal of the system is to emulate the actions that a human would use. The fuzzy controller discussed by Baxter and Bumby (1995) consists of three parts: the navigation control, obstacle avoidance and velocity control. Two principles were used for motion control. The first is to always have the obstacle avoidance structure active without allowing it to not affect the normal operation of the robot. This required the navigation control and obstacle avoidance controllers to be integrated to prevent conflict. The second principle is to always meet the requested steering angle even if the current speed of the vehicle must be reduced to meet the requirement. The navigation controller used the linguistic descriptions large, medium, small and zero to describe the distance from the robot to the goal. A heading error and goal errors, which describe the relationship of the robot to the goal from the fuzzy tables, determine the new steer direction of the vehicle. Obstacle avoidance fuzzy rules contained information about the location of the obstacle and where the robot could not safely steer. The velocity control considers the requested steering angle, the difference between the actual and requested steering angles and the distance to the goal to determine the speed of the vehicle. The speed of the vehicle could be set to either slow or fast. The vehicle remains at a slow

speed unless the fuzzy condition of a small difference between actual and requested steering angles and a large distance to the goal are met.

## 4.3 Additional Motion Considerations

Other than the resulting motion of the robot, several additional factors need to be considered for effective motion control. Certain constraints must be put on the generated motion trajectory to ensure the safety of the vehicle. Errors are inherent to any motion system and must be identified and corrected. The "cost" of the resulting motion must be examined to ensure that it is within specifications. Since a wide variety of different drive systems can be used on a mobile robot, the selection of a certain type can produce different motion capabilities. These factors will be discusses in more detail in the following sections.

### 4.3.1 Trajectory Constraints

Motion control of a mobile robot is affected by the constraints placed on the produced trajectories. If constraints are violated, then the requested motion control is infeasible. These constraints can be classified in three categories: kinematic, static, and dynamic. Kinematic constraints determine if the geometry of the path is safe based on the current position and orientation of the robot. Static constraints are limits that come from the vehicle, such as steering and wheelbase. Constraints that reflect mass and force relations

and frictional interactions of the wheels and the surface represent dynamic constraints (Graettinger and Krogh, 1989; Stentz et al., 1993).

Stentz et al. (1993) invoke all three constraints on the robot system to produce a trajectory safe for navigation on a rough, outdoor terrain. Three kinematic constraints were defined as body collision , wheel obstacle, and minimum turning radius. The body collision constraint limited motion if part of the body of the vehicle would collide with part of the terrain. Motion is not possible if an obstacle is in front of a wheel of the vehicle that the robot can not cross and continue forward. This is the wheel obstacle constraint. The minimum turning radius constraint will stop motion if the planned path has a turn that is smaller than the minimum turning radius of the vehicle. One static constraint was considered, namely, static tip-over. Using this constraint, control is dismissed if the vehicle becomes inclined enough to become unstable and tip. Four dynamic constraints were considered: skid, slide, dynamic tip-over, and sail. The skid constraint limits the forward acceleration of the robot to prevent skidding due to braking or throttling. The slide constraint places limits on the acceleration perpendicular to the direction of motion to prevent lateral wheel slippage (sliding). This perpendicular acceleration limit is also used for the dynamic tip-over constraint. This prevents the vehicle from falling over because of centrifugal force. The sail constraint puts a limit on the speed of the vehicle to keep it from losing traction as is moves over the top of a hill.

Graettinger and Krogh (1989) also incorporate all three constraints in their motion control system. The static constraint sets the upper and lower bounds of the steering

angle based on the turning capability of the steering mechanism. The kinematic constraints set limits on the steering angle rate. This is the rate the steered, front wheel can be moved due to the response of the steering mechanism. Two types of dynamic constraints were used. The first imposed limits on the applied propulsion torque and the second attempted to limit the frictional force for a no-slip assumption.

Static and dynamic constraints are imposed on the mobile robot discussed by El-Konyaly et al. (1995a). A static constraint was set to prevent the robot from exceeding the maximum steering angle. The dynamic constraints were: a sliding constraint that the total friction must satisfy a no slip condition; a tip-over constraint when the vehicle was accelerating, decelerating, or moving around a curve; and a drive torque constraint to ensure the drive torque was within the physical limits of the motors.

For navigation in an outdoor terrain, additional constraints should be imposed on the motion of the vehicle. The undercarriage height of the robot should be considered to prevent obstacles from wedging under the bottom of the vehicle. The maximum step the vehicle can climb over should also be considered. A vehicle tipping constraint needs to be used so the robot will not try to negotiate a slope that is too steep. (Gowdy et al., 1990)

Brumitt et al. (1993) consider points for navigation only if they satisfy four kinematic constraints. The first is the minimum turning radius. This means that the curvature of the path cannot be smaller than the vehicle's minimum turning radius. Second is locomotion support to ensure the vehicle will not attempt to drive over an obstacle. The last two

constraints are classified as body collision. They ensure that the vehicle will not collide with the known terrain areas and that the robot is not navigating in an unknown terrain area. A static constraint is also used to ensure the "vehicle is in a state of static equilibrium at that point" A set of dynamic speed constraints are also defined. A lateral acceleration limit is defined so the vehicle will not slip. Skidding by braking is considered by setting a parallel acceleration limit. The vertical acceleration, which can cause the vehicle to lose traction, is also limited.

Two kinematic constraints for a four wheeled front-wheel-steer robot are described by Vasseur et al. (1992). The first ensures no slippage of the wheels, and the second limits the steering angle of the front wheels. Four dynamic constraints were considered by Shiller and Gwo (1991). Limits on the engine torque were bounded by the maximum engine force, which is a positive torque in the direction of motion, and the maximum braking force, which is a negative torque. Limits on the coefficient of friction, a sliding constraint, ensured that the friction force from the vehicle motion did not exceed set limits. A contact constraint, positive contact between the vehicle and the ground, is met by setting a maximum safe angle for the vehicle. The tip-over constraint is set by determining the force that must be applied to shift all the weight of the vehicle to one side.

### 4.3.2 Errors

"One of the greatest challenges in the motion planning and control of autonomous mobile robots in a-priori, unknown or dynamic environments is to provide the reasoning modules with methods for handling and/or coping with the many imprecisions, inaccuracies, and uncertainties present in the system" (Pin et al., 1992). These system errors are usually in three forms: sensor error, incomplete environment descriptions, and motion control errors (Pin et al., 1992). Sensor errors come in many different forms. These errors were discussed in Chapter 2. Chapter 3 covers mapping systems and problems with a incomplete description of the surroundings. This section presents errors that can be produced during the motion of the robot.

The most common type of motion control error comes from odometry errors. An excellent discussion on how to measure odometry errors and express them quantitatively can be found in Borenstein and Feng (1995). The following discussion is adapted from that reference. There are two types of odometry errors: systematic errors and non-systematic errors. Systematic errors are caused by imperfections with the robot itself and remain constant over time. Non-systematic errors are caused by the properties of the surface on which the robot is moving. Common systematic errors are caused by unequal wheel diameters; misalignment of wheels; uncertainty about the effective wheelbase (due to non-point wheel contact with the terrain); limited encoder resolution and limited encoder sampling rate. Non-systematic errors are most often caused by travel over uneven floors; travel over unexpected objects on the floor; wheel-slippage due to:

slippery floors, over-acceleration, fast turning (skidding), external forces (interaction with external bodies), internal forces (e.g., castor wheels) and non-point wheel contact with the floor.

Systematic errors are most significant when navigating on a smooth, indoor surface. On rough terrain, non-systematic errors are the prevalent source of error. Systematic errors are caused by kinematics imperfections of the robot and accumulate constantly. The most common systematic errors are unequal wheel diameters and the uncertainty about the effective wheelbase.

Odometry errors occur because the rubber tires typically used on mobile robots are hard to produce with exactly the same diameter and compress differently under different loads. The wheelbase is the distance between the contact points of the floor and the two drive wheels of a differential-drive robot. This feature is used to determine the number differential encoder pulses that correspond to a certain amount of rotation of the robot. The uncertainty in the effective wheelbase is caused because the rubber tires contact the floor in a contact area, not at a single point. This uncertainty can be around 1% for commercially available robots.

Another error that is often present on mobile robots is caused by the average wheel-diameter differing from the nominal wheel-diameter. This is called scaling error. If the average wheel diameter is larger than the nominal wheel diameter, the robot will travel farther than instructed or turn too much. This error can be partially corrected by using a tape measure to measure the scaling error in the system.

Other errors are related to the position of the robot. Lee and Williams (1993) classify this error into three types: heading error, longitudinal error and lateral error. Heading error is the difference in the desired and actual orientations of the robot. This is a major problem and can generate lateral error if it is not quickly corrected. This problem can be eliminated by controlling both driving wheels at the same time. The longitudinal error of the vehicle, vertical misalignment, can be corrected by coordinating the actions of both controllers. The lateral error is a function of the heading error and time. This means that lateral error is caused by not correcting the heading error and increases with time.

### 4.3.3 Control "Costs"

Motion control systems receive typically several possible paths for the robot to follow to reach the final goal. The robot must then determine which route to follow. This path selection is based on the "cost" of following each planned path. The cost of traversing each path is based on a set of pre-defined requirements. Paths are usually selected using the shortest path distance as the cost function. Other cost strategies are number of maneuvers for the vehicle, computational time, travel time, maximum velocity, and minimum energy for fuel conservation (Inigo and Alley, 1990)

The robot discussed by Inigo and Alley (1990) used travel time as the cost function for selecting a path. The vehicle would search its world database for the path with the minimal time cost. The cost of the selected path can only be estimated because changes in the environment can cause the robot to veer off the selected path. The cost

consideration used by Shiller and Gwo (1991) was based on the path distance. Once all possible paths to the goal were generated, the robot would estimate the distance to the goal for each route and select the path with the shortest distance. Computational time is another important cost factor. A path with a high computation time for navigation can impede the progress of the vehicle considerably.

### 4.3.4 Drive Systems

There are three main types of drive systems used on autonomous mobile robots: differential, syncro and tricycle. A differential drive uses two drive wheels, each with its own motor. This system has a simple mechanical design that allows the robot to turn in place (Borenstein, 1995). This design does not require many parts, is easy to control, saves computation time, and has a lower cost (Lee et al., 1993). An important design issue to consider when using a differential drive is the location of additional wheels to stabilize the vehicle. One or two caster wheels are usually added to the robot to obtain a triangle or diamond wheel configuration. Jones and Flynn (1993) suggests problems with both of these configurations. The triangle arrangement of the wheels, may still leave the robot unstable. Using a four wheel diamond pattern, if the robot is on an uneven terrain, the two drive wheels may lose contact with the ground an be supported by the two casters leaving the robot unable to move.

A syncro-drive uses three wheel that are both driven and steered. The wheels are linked together by a set of belts, chains or concentric shafts with bevel gears (Zhao and

BeMent, 1992). One motor drives the vehicle and another motor controls the steering. The wheels can turn in any direction, but are coupled to move as a set. The vehicle is able to move in any direction, but the orientation of the vehicle can not be controlled, since only the wheels turn (Borenstein, 1995). This does not have many of the control problems associated with other drive systems, but it is complex to implement mechanically.

A tricycle drive has three wheels in a triangular arrangement. This design can have one wheel steered and driven and the other two fixed, the two rear wheels driven and the front wheel steered, or all three wheels steered and driven (Borenstein, 1995; Jones and Flynn, 1993).

## 4.4 Summary

Motion control is the "acting" task of an autonomous mobile robot. Conventional control systems are widely used on mobile robots, but the development of intelligent controllers is evolving rapidly. Table 4.1 lists some of the advantages and disadvantages of each type of control scheme.

Table 4.1: Motion Control Comparisons

| Control System | Advantages | Disadvantages |
|---|---|---|
| Open-Loop Control | can be used for simple tasks | no feedback information is given, can't be used in complex environments |
| Closed-Loop Control | can use feedback to correct errors or feedforward to try to predict errors | system equations must be modeled, can't adapt quickly to change |
| Fuzzy Logic | no environmental model is needed, uses linguistic terms, real-time control | may need to use conventional control for precision navigation |

As seen from the above table and several of the examples in the fuzzy logic section, a combination of intelligent and conventional control may be the most effective way to control the motion of a mobile robot. This combination of systems will continue to develop as the use of intelligent control methods continues to increase.

# Chapter 5

# System Architecture

The system architecture of an autonomous mobile robot defines the way the activities of the robot are related. The perception, thinking and acting systems of the robot must be coordinated to produce the desired function of the robot. The architecture of the system determines the nature of this interaction.

No one type of system architecture has been universally adopted as the best way to control a mobile robot. Nevertheless, Liscano et al. (1995) have established a general set of requirements for an architecture system. These requirements should enable the autonomous mobile robot to navigate in different and changing environments. Two groups of attributes have been defined: behavior and design attributes. Behavior attributes produce the desired behavior of the mobile robot and design attributes are needed for a good control architecture.

Six behavior attributes were identified as necessary for the operation of a mobile robot. Reactivity is the first. Reactivity is needed because the state of the world is constantly changing and the robot must be able to adapt to unexpected change. The robot needs intelligence to deal with new situations. This ability helps the robot achieve a goal by dealing with different situations. Centralized global reasoning is a feature that uses a

global, high-level decision-making module to help the robot understand its overall situation. This lets the robot coordinate several independent activities. A robot must also be capable of dealing with conflicting actions and perform several tasks. This is classified as multiple-goal resolution. The system must also be robust enough to handle imperfect inputs, unexpected events, uncertainties, and sudden malfunctions. The robot system must also be reliable. This means that is should perform at the same level in different situations.

Since a mobile robot must constantly reevaluate its surroundings, system designs that can be easily modified are needed. Five design attributes are listed as requirements for a mobile robot system architecture. The first is modularity. This is needed because smaller sub-systems are usually designed, implemented and debugged separately. Systems must also be flexible, because constant changes usually need to be implemented. An architecture should also be easy to expand. This enables the vehicle to constantly add new information needed for different situations. The architecture must also be able to adapt to changes in the environment by modifying the current strategy to adapt to the situation. Multisensor integration, or sensor fusion, is the ability of the robot to combine information from several sensors. This is usually needed for real-time mobile robot control.

There are three commonly used system architectures: hierarchical, behavioral, and blackboard. Table 5.1, taken from Liscano et al. (1995), compares these three strategies based on the six behavior and five design requirements.

Table 5.1: System Architecture Attributes

| | **Hierarchical** | **Behavioral** | **Blackboard** |
|---|---|---|---|
| Reactivity | Medium | High | Medium |
| Intelligence | Sequential | Emergent | Distributed |
| Global | Yes | No | No |
| Multiple Goals | Yes | Difficult | Yes |
| Robustness | Low | High | Medium |
| Reliability | Low | High | Medium |
| Modularity | Yes | No | Yes |
| Flexibility | No | No | Yes |
| Expandability | Yes | Yes | Yes |
| Adaptability | No | No | Yes |
| Sensor | Difficult | Yes | Yes |

As can be seen from the above table, no single architecture has all eleven attributes. That is why "most mobile-robot control systems are hybrid systems combining approaches from hierarchical, behavioral, and blackboard systems" (Liscano et al., 1995).

This chapter discusses the three common system architectures for mobile robots and some of the current applications. The chapter summary contains a table comparing the advantages and disadvantages of each architecture.

## 6.1 Hierarchical Architecture

A hierarchical control structure uses several levels to perform different functions. These levels are connected in a system that has the more complex decisions being made

at the higher levels and systems needing less information at the bottom levels (Passino, 1995). Banta et al. (1992) defines the levels in a hierarchical architecture to include a global planner for reasoning and path planning, a navigator to check the progress of the robot against the map, a sensor management and processing level for obstacle avoidance, and a pilot or actuator control level that does the actual speed control and steering. The decisions are made at the global planner level using information passed from the control and sensing levels. Figure 5.1 shows how information is passed from each level and how the resulting motion information is then used by each level for the next series of decisions.

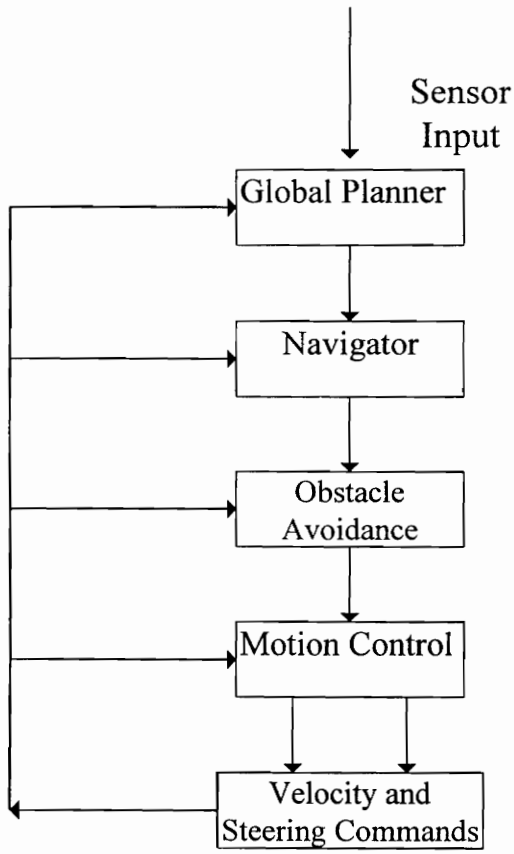


Figure 5.1: Hierarchical Architecture

Hierarchical, or planning systems, systems are user friendly. The system is able to take a "natural language format and use a "command interpreter" at one of the levels to produce the desired motion (Banta et al., 1992). Bay (1995) lists other advantages of planning systems as having a long-term memory to represent the environment, choosing actions to get "long-term" rewards, examining alternate plans, anticipating the result, evaluate their progress and planning future actions. Hierarchical planning systems can be used for a wider variety of situations because it they are a set of specific rules for each situations. Also, paths developed at higher levels can easily be modified by information from other levels (Guldner et al., 1994; Arkin, 1992)

There are several drawbacks to hierarchical systems. The major disadvantage is the large amount of computation time required by the system. Changes to the decisions usually have to pass through several levels in the hierarchy, causing the system to be slow and have a high computational cost (Liscano et al., 1995; Banta et al., 1992). Because of the connection between each level in the hierarchy, if one level fails, the entire system fails. This problem can be reduced by dividing the system into separate modules, but the system does not allow these modules to be modified or moved around very easily (Liscano et al., 1995). Hierarchical systems use "sense-think-act" cycles that are difficult to implement in real-time (Liscano et al., 1995). This can cause the robot to ignore current sensor information and not be able to navigate successfully in a changing environment.

A two-level hierarchical structure was used by Fualdes and Barrouil (1993). One level dealt with symbolic data and the other with numerical data. The symbolic level receives information on the goals to be accomplished. A reasoning agent examines the symbolic data to evaluate the environment and decide the appropriate actions. Because the symbolic information is a collection of numerical data, the numerical level processes this data to update numerical parameters, produce continuous control and send symbolic messages to the symbolic level. Olin and Tseng (1991) discuss a hierarchical system architecture that tightly integrates perception and planning. The higher levels of both the planning and perception systems use symbolic data to make "long-term decisions" and the lower levels use numeric data for immediate planning. At the highest level of the hierarchy, a map-based planner defines the goal to navigate toward based on the map information. The next level, the route-planning level, generates possible paths to reach the goal, selects the shortest path, and defines a set of subgoals for the robot to reach. The lower levels use a local planner makes sure the vehicle is on route to the subgoals and a reflexive planner controls the vehicle in real-time.

A three-level hierarchical path planning structure is used by Sugiyama et al. (1994). A global path planner gives the connection relationships of rooms that are needed to reach the goal by the local path planner. The local path planner plans a path to navigate around the room as subgoals. The obstacle-avoidance planner is used if an object is detected. It reports the location of the object to the local path planner and a new path is found. An algorithm using a three-level hierarchical structure for fuzzy rules is discussed

by Aoki et al. (1994). The lowest level decides the inputs for velocity and steering. In the middle level, a fuzzy balancer adjusts the inputs so they are not in conflict with each other. The top level combines the control needed to reach the goal and to avoid any obstacles to produce the motion of the vehicle. A three-level hierarchical system for path planning is used in Guldner et al. (1994). The global planning level plans the route for the robot to follow to the goal. The local navigator modifies the path, if needed, based on the lower level input. The lower level is the collision avoidance layer. This level uses sensor input to signal if the path needs to be modified.

A four-level hierarchical structure is used by Zhang and Raczkowsky (1994). The first level of the system has a motion command as its input. This command is passed to the Subgoal Planning module to produce points for the robot to follow while avoiding obstacles. Next, a Situation Evaluation of the environment is made. This compares the information from the planned path with current sensor information to determine if a new path plan is needed. This is passed to a Motion Execution level that generates the signals to control the motion of the robot. A navigation system described by Davis (1991) also uses a four-level hierarchical path planning strategy. At the first level, the robot is instructed to "visually sight a set of locations on the surface of the target." This subtask information is passed to a goal point identification level that identified a goal point on the target. A path to reach the goal is planned in the next level. Finally, the target is constantly sensed and used to update the motion of the robot.

## 5.2 Behavioral Architecture

Behavioral architecture is a system that directly related the sensing and acting functions of the robot. This method divides the control of the robot into separate behaviors that are needed for navigation. A behavior architecture system uses the sensor inputs and determines the actions needed. This strategy is also referred to as a reactive architecture. This set of behaviors can be combined without any central intelligent reasoning agent (Fayek et al., 1993). Behaviors can be used for simple motion movement, such as wall following and obstacle avoidance to more complicated navigation tasks by building up layers of behaviors (Liscano et al., 1995; Banta et al., 1992). Behavior control can also be used in situations where it would be difficult to model the environment to use other control strategies. Figure 5.2 shows the typical behavioral plan.

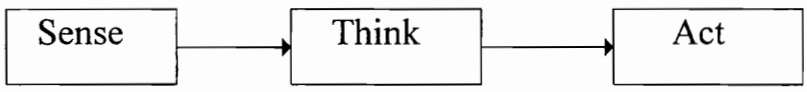


Figure 5.2: Behavioral Architecture

One of the most famous behavioral architectures uses is called subsumption architecture (Brooks, 1986). Subsumption architecture combines sensor input that produces behaviors and real-time control techniques. This architecture uses the sensor input to initiate behaviors. These behaviors are "simply layers of control systems that all run in parallel whenever appropriate sensors fire" (Jones and Flynn, 1993). Behavior

fusion is used to deal with conflicting behaviors instead of the traditional sensor fusion processes. The behavior is selected based on a pre-determined set of priorities. This is a parallel architecture is the sense that all behaviors are running at the same time. The higher priority behaviors control the actions of the vehicle until the sensor input is no longer needed, allowing lower-level priorities to take control (Jones and Flynn, 1993). This architecture is compared to a wiring diagram that connects the outputs of some machines to inputs of others with wires that are used to transmit messages (Brooks, 1989)

Behavior-based architectures are popular because the direct relationship between sensing and action produces a reactive and reliable behavior from the robot. Each sensor contributes to the motion of the vehicle, so if one behavior fails, the other behaviors are only minimally affected (Liscano et al., 1995). A central intelligent control system is not needed to coordinate each behavior action. This simplifies the communication needed among modules and results in a fast reaction time for the vehicle (Banta et al., 1992). Bay (1995) lists another advantages of reactive controllers is that they do not need a world model because their actions are based on sensor input. This eliminates the need for a large amount of memory and allows the robot to adapt to changes in the environment based on current sensor information.

Jones and Flynn (1993) list three advantages the subsumption architecture has for programming robots. First, because of the close relationship of sensing and action, no world model is need for the simple reflexive actions. This reduces the memory needed on-board the vehicle enabling most computations to be performed by simple

microprocessors. A second feature is that it is easy to add new behaviors to the robot. This can be accomplished by writing and wiring in new layers without losing other behaviors. Finally, the robot can still navigate quickly with several behaviors. Since the system computations can be performed in microprocessors, additional processors can easily be added to keep the speed of the robot the same.

One problem with reactive strategies is that for complicated tasks, the user must for decide what behaviors are needed and work backward to program these actions. It can also be difficult to change the programs since "much of the stimulus-response mechanism is hardwired" (Banta et al., 1992). Guldner et al. (1994) also state that it is almost impossible to predict every behavior that will be needed for navigation causing the rule base to be incomplete. Reactive systems also repeat past mistakes, can not use any learning methods, and make it difficult to implement a goal-oriented behavior in complex environments (Bay, 1995). Fernandez et al. (1994) use a reflexive behavior for navigation but comments that a drawback to this system is that, even though a safe path can be found, it is usually not the optimal path for navigation.

Agah and Bekey (1994) use a behavior based architecture that allows for both reflexive and premeditated behaviors. The operation of the robot is based on behavioral assemblies. A behavioral assembly is a collection of separate behaviors, called behavioral statement, with each containing the relationships of the perceptions and the resulting desired actions. Each behavioral statement consists of a statement type, an address, a perception and an action. The resulting behavioral assembly is selected based

on the current task and state of the robot. Each component in the architecture system has its own memory to exchange information with other modules. This structure enables the robot to record situations that were encountered more than a set number of time and store them in a database for future use.

Hugli et al. (1994) use a behavioral approach for navigation using a vision system. A behavior is defined as "an independent stereotyped action that is maintained by a specific perceived stimulus." This is implemented by having a visual pattern stimulate the action of the robot. When the vision system detects the stimulus, a action by the robot is initiated. This action is continued as long as the stimulus exists. Murphy (1995) describes the use of a behavioral architecture on a vehicle used to navigate around an outdoor obstacle course. This architecture was selected because the exact layout of the course did not need to be known in advance for successful navigation. Tunstel and Jamshidi (1994) present a strategy that incorporates fuzzy reasoning into a behavior control system. This combination is used for autonomous mapping of the environment. Fernandez et al. (1994) use a reflexive behavior to navigate a robot in a mountainous environment. The current condition of the robot's surroundings are evaluated. This information is used to plan a path for the robot. The robot stays on the same path until new sensor information reveals that the scene has changed and a new path needs to be computed.

Jorg (1995) states that, for a robot to be fully autonomous, it must be capable of goal-oriented behavior and reflexive behavior. This combination hierarchical and behavior

architectures is being used for a variety of applications. This idea is implemented by Jorg (1995) by using a hierarchical architecture for goal-oriented navigation and a reflexive behavior for obstacle avoidance. The hierarchical system had two levels: a strategical level and a tactical level. The strategical level performed the global planning. This is used at the tactical level, along with information about the current state of the local environment, for local navigation. A reflexive behavior is implemented at an operational abstraction level. This behavior is needed due to the uncertain, changing state of the environment.

Banta et al. (1992) present an architecture that combines ideas behind hierarchical and subsumption systems. This "mode-based" approach selects the mode to operate based on its current situation. The reactive mode is used to follow walls , but if unexpected problems occur, a higher level takes over, like in a hierarchical system. Chun and Jochem (1995) discuss a architecture for a mobile robot that is also a combination of a hierarchical structure and the behavior method. A three-level architecture is used. The lowest level is the control layer. This layer contains the low-level functions of the system, such as closing servo loop. The middle layer is the local action layer. The behavior and reflexive commands are at this level. The global action layer is the highest level. This level performs the "thinking" tasks for the robot such as path planning.

Payton et al. (1990) use an architecture that closely ties in sensing and action for real-time robot control. This system is hierarchical in nature, but the lowest level is similar to subsumption architecture. This level used separate behaviors based on sensor

information for its particular decision-making needs. This made the system faster, because an accurate model of the environment did not need to be obtained to produce correct navigation action. This eliminated the problems and delays caused by sensor fusion.

## 5.3 Blackboard Architecture

The blackboard architecture uses a blackboard, or system of posted "notes", to control the actions of several systems to produce a desired action. This structure enables several independent activities to run at the same time and use a common control unit to coordinate the reasoning and decision-making. An activity contains a perception, action, and partial decision making process that enables the robot to perform a specific function, such as tracking a landmark or traversing a hallway (Liscano et al., 1995). The blackboard decides which activities should be active based on the sensor input. "The blackboard system acts as a high-level reasoning agent, while the activities perform the desired functionalities of the vehicle in real time" (Fayek et al., 1993).

Blackboard systems are flexible, making them useful for a variety of applications and making sensor data easy to integrate. Blackboard architectures also keep all information about the environment on the blackboard. This lets all of the sources have access to all data. This makes it effective for navigation because the robot has the ability to manage several processes. This places a high computational demand on the system and

eliminates the reactive behavior capability of the robot. Therefore, most blackboard systems must be modified for real-time control (Liscano et al., 1995).

The blackboard system discussed by Liscano et al. (1995) uses the blackboard to coordinate the real-time activities of the system and not for decision making. The activities constantly post their current state on the blackboard, but not all communication between the modules of each activity go through the blackboard, enabling the system to exhibit reactive behavior. This system is like a traditional blackboard architecture because the information from each module is stored in a central location, but the individual modules do not work together to solve the same problem. A set of pre-defined rules are used to determine which activity will control the motion of the robot.

Davis (1991) used a set of interacting blackboards for the system architecture. These blackboards contained explicit representations for objects and processes. Each blackboard is divided into areas that correspond to a class of observations and contain the rules that define the characteristics of that class. This information is used for navigation and obstacle avoidance problems.

A system that operates like a blackboard architecture is the Learning Classifier System (LCS). This system is uses "posted-message communications", but the message board is wiped clean at every time interval, thereby requiring no persistent shared sources" (Bay, 1995). The Learning Classifier System also combines characteristics from hierarchical and behavioral architectures. A rule-based system behavior system is used to allow for reactive behaviors, but the structure of the system also uses basic learning techniques.

The uses can weigh different parameters to make the vehicle goal-oriented or situation-oriented. Learning Classifier System also pass messages to aid with the navigation of the vehicle. Bay (1995) implements the Learning Classifier System on several interacting mobile robots.

## 5.4 Summary

The system architecture of the robot controls how the individual actions are related. The three most common architectures hierarchical, behavioral, and blackboard, were discussed along with several applications on real-world mobile robot systems. Table 5.1 compared the three architecture based on the eleven attributes defined by Liscano et al. (1995). Table 5.2 is a short list of some of the advantages and disadvantages of each structure.

Table 5.2: System Architecture Comparisons

| System Architecture | Advantages | Disadvantages |
|---|---|---|
| Hierarchical | goal-oriented, plans each action, no set rule base | high computational requirement, slow, hard to implement in real-time |
| Behavioral | no central intelligence agent, no world model, small memory requirement | hard to modify, incomplete rule base, can't learn from mistakes |
| Blackboard | stores all environment information, flexible, easy to integrate sensor data | high computational time, hard to implement in real-time |

One, or a combination, of the above architectures is used on mobile robots to integrate the perceiving, thinking and acting systems. Chapter 6 explains the integration of all these systems on one type of mobile robots.

# Chapter 6

# UGR Competition

This chapter describes how the topics discussed in the previous chapters have been integrated to design and build an autonomous mobile robot. A brief survey has been compiled of vehicles that have entered the Unmanned Ground Robotics (UGR) Competition. These vehicles were chosen for the survey because they are all excellent examples of integrating several systems onto a mobile robotic vehicle for collision avoidance and boundary line detection. This chapter contains a overview of the competition and descriptions of several of the vehicles that have been entered.

## 6.1 Competition Overview

The First International Unmanned Ground Robotics Competition was held in 1993 on the campus of Oakland University in Rochester, Michigan. This event was sponsored by the Association for Unmanned Vehicle Systems (AUVS), the U. S. Army Tank-Automotive Research, Development and Engineering Command (TARDEC) and Oakland University (OU). "The purpose of the competition is to encourage university students to participate in designing and constructing of intelligent unmanned robotic vehicles in the early stages of their professional and training careers" (Cheok, 1994).

The objective of the competition is for a vehicle to autonomously navigate around an outdoor obstacle course. The general shape of the course is known beforehand, but the actual dimensions and placement of the obstacles are unknown to the teams. The obstacle course is laid out by two lines that are between ten and twelve feet apart. For the 1993, 1994, and 1995 events, the lines were white painted on grass. The rules for the 1996 competition state that the lines could be white or yellow and laid out on grass or asphalt. The vehicles must be able to stay between these lines and avoid any obstacles in its path. The obstacles blocking the path of the vehicles are hay bales that can be covered in white or red trash bags, sand pits, or inclines for the robot to climb. After three heats, the winner of the event is the team that navigates around the course with the best "adjusted time". If no team completes the course, the vehicle with the highest "adjusted distance" will be declared the winner. The adjusted values are calculated after assessing penalties for colliding with obstacles or having part of the vehicle cross the boundary line. The penalty is a five second or five foot deduction depending on if the course is completed. The courses for the 1993 and 1996 competition are shown in Figures 6.1 and 6.2, respectively.

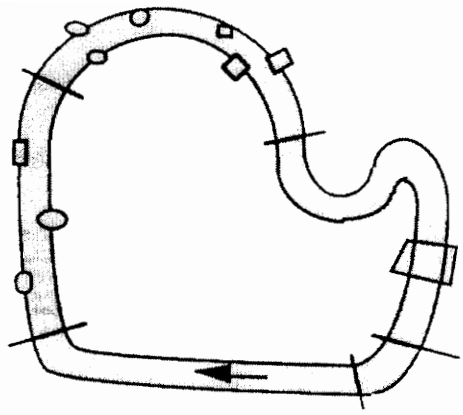

Figure 6.1: 1993 Obstacle Course

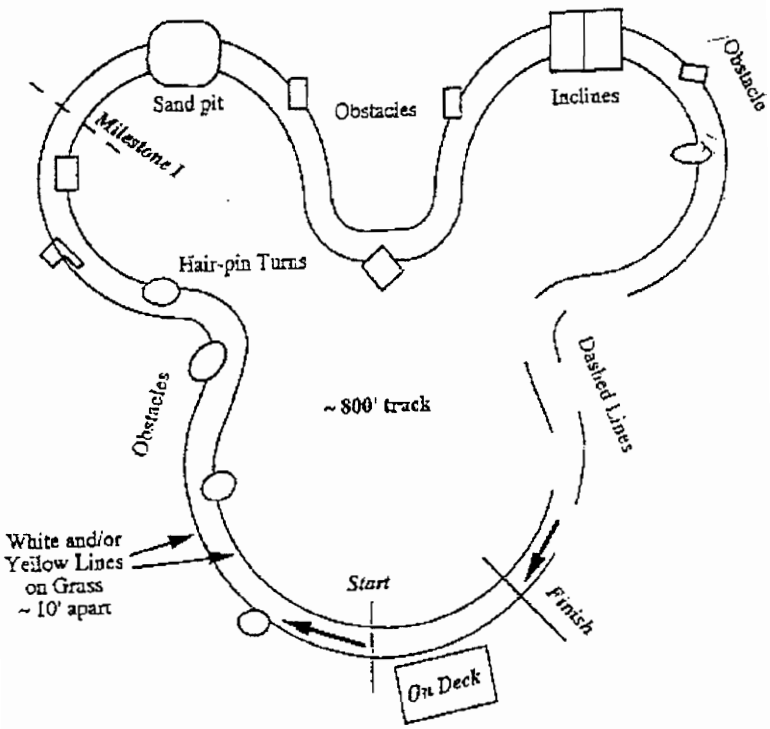


Figure 6.2: 1996 Obstacle Course

The requirement of the vehicle dimensions are that it must be less than six feet in height, five feet in width, and have a length of between three and nine feet. The robot must also be able to carry a twenty pound payload. The robot is also restricted to a five mile per hour speed limit and must be equipped with an on-board and remote emergency stop. The competition will take place regardless of weather conditions, so the vehicle must be protected against rain, wind, and sun.

The students not only compete for school recognition, but also for cash awards. The first place team receives $5000, second place $3000 and third place $2000. The first three competitions were held on the campus of Oakland University in Rochester, MI. The fourth annual event will take place during the summer of 1996 at Epcot Center in Orlando, Florida.

## 6.2 Colorado School of Mines

The Colorado School of Mines has had success with two different entries in the Unmanned Ground Robotics Competition. In 1994, the "Omnibot" was entered and in 1995, "Clementine 2" was entered. They both had the same type of base vehicle, but the navigation strategies were different.

### 6.2.1 Omnibot

"Omnibot" is a battery powered children's "Power Wheels" jeep. A 486 personal computer, with a frame grabber board, is on-board the vehicle. A single ultrasonic sensor is mounted on the front of the vehicle and a video camera is on a raised mount near the center of the vehicle.

Their method for navigating around the course used a reactive architecture. This was accomplished by breaking the tasks into separate behaviors. These behaviors map the input from the sensors and command the motor how to move. The behaviors were designed using "action-oriented perception". This method "emphasizes the importance of sensing in generating the correct response" (Murphy, 1995). Two basic behaviors were implemented on "Omnibot". The first was a follow-line behavior. This uses the information from the vision system and tells the robot to follow the white line on the right side and maintain a specific distance from that line. The second behavior was to move-ahead. This was used for obstacle avoidance and was only enabled when the sonar detected a hay bale. This behavior would help the vision system navigate around the obstacle.

There are three steps to the follow-line behavior. The first uses the camera to observe the environment and computes intensity values for pixels. The brightest 2% of the pixels are selected and compared to a threshold value. After comparing the values, those equal to or above the threshold are considered the white line. The center of the line is then calculated. The motion control of the vehicle consists of taking this value and generating

a steering command five feet from the center of the white line. Because of the small size of the vehicle, this would keep the robot in the center of the course. Problems arise when a hay bale is detected by the vision system. The pixel values for the obstacle would be of the same intensity as the line. At this point, the move-ahead behavior is used. When detected, the obstacle is assumed to be shorter than four feet. The follow-line behavior is then turned off and the move-ahead behavior uses the shaft encoders to move ahead four feet. The follow-line behavior then tries to find the line again.

This navigation strategy proved to be successful, but the robot failed mechanically. Using a commercially made base, the vehicle was not able to cross the sand pit because the wheels did not have enough traction. This navigation method was good for the competition, but would have some disadvantages if used for a broader application. One is that small variances in the course could cause unexpected actions. Corrections could be made in the programming to search for larger sections of white pixels to avoid picking up reflections on the grass or other problems. The biggest drawback is that this method does not have any memory of where the robot has previously navigated. This could be a big disadvantage if the controller gets confused because the vehicle would be unable to back up and perceive the environment again.

### 6.2.2 Clementine 2

"Clementine 2" (C2) is also a children's "Power Wheels" jeep. On-board the vehicle are a Pentium computer with a black and white frame grabber board, a panning

camcorder with a wide angle lens, and one panning ultrasonic sensor on the front of the robot.

A reactive architecture is used for navigation. Four separate behaviors are described as being needed to successfully navigate the course. The follow-path behavior uses the input from the vision system to stay in the center of the path. An avoid-obstacle behavior is used if an obstacle is detected by the sensing system. Pan-camera is used to keep both boundary lines in view. The speed-control behavior was considered for slowing the robot down in situations where there was a tight turn, or when the robot was going up an incline. However, it was not implemented because the maximum speed of the vehicle was 1.5 mph and strict speed regulation was not necessary.

The vision system used three steps for the follow-path behavior. The image was split into six areas where the white boundary lines were expected to be. The edges were then detected and filtered to determine which line in each region best matched the expected input. The lines were then compared to a road model. "Mismatches were expected because the placement of the regions was static; if the course was going up a hill and turning, the line might disappear in the upper half of the image. In that case, the robot would continue on its current course and assume that it would reacquire the path in the next image" (Murphy et al., 1995). An exact model of the environment was not constructed for comparing separate regions, but it was not needed since the processing time was fast enough to correct any mistakes before the vehicle left the course boundary lines. An approximation of the steering angle was also used since the steering system on

the vehicle could not duplicate precise angles. This also sped up the processing time of the computer.

The panning ultrasonic sensor is used for obstacle avoidance. The information from the sonar is mapped using a Vector Field Histogram method. Five readings are taken and areas are labeled as either occupied or empty. The avoid-obstacle behavior determines which of the five areas is the center of the path and searches around this goal. If an obstacle is present, it checks the sectors on either side. If these are clear, the robot will proceed to the empty area. If all areas appear to be occupied, it is assumed that some of the readings are false and the least occupied area is chosen. In case of ties, the vehicle will move to the right.

The motion control of the robot began by first using the follow-path behavior. Direction information can be changed if an obstacle is detected by the avoid-obstacle behavior, or if the pan-camera behavior determines the camera is not pointed in the center of the course. The only mechanical change is the steering angle used to keep the vehicle in the middle of the course.

The biggest drawback of this navigation strategy is that no information about where the robot has already been is stored. This became a problem when only one or neither boundary line of the course was seen by the vision system. Keeping prior knowledge of course boundaries would help the vehicle back-up and try to locate the lines.

## 6.3 Northern Illinois University

Northern Illinois University has entered two different vehicles into Unmanned Ground Robotics Competition. The "NIU Rover" was entered in the 1994 and 1995 competitions and the "Mean Machine" was also entered in 1995. The "NIU Rover" was a golf cart that was designed to be robust. This vehicle was chosen because it had a reasonable cost, could navigate on a rough terrain, and could be easily automated (Nagy and Bock, 1995). The "Mean Machine" was designed specifically for the UGR competition. This "Power Wheels" Jeep was designed to be inexpensive.

### 6.3.1 NIU Rover

The "NIU Rover" is an electric golf cart. By using this existing base, the designers only needed to automate three systems: the steering, the accelerator, and the brakes. One of the design considerations was to ensure that each system could be overridden by human control. They also considered, cost, dimensional constraints, power consumption, and ease of computerizing the control of additional components.

The original braking system on the golf cart has a foot pedal, like the brakes on a car. They added a linear actuator for autonomous operation. The operation of the linear actuator is overridden if someone steps on the brake pedal. The steering system consists of a DC servo motor with high rpm and low torque to move the steering wheel. This is accomplished by mounting the motor on the steering column and driving it through a chain. The position of the motor is controlled with a commercially purchased motion

control card. Position feedback is provided by an optical encoder that is mounted on the motor. To "home" the steering system, two switches on the frame of the cart give left and right extreme positions when a pin in the steering frame strikes them. The steering wheel is still able to be turned by an on-board operator. The acceleration on the golf cart is controlled by a foot pedal. To accomplish this autonomously, the motion control card was also used for accelerator control. "Rather than mechanically depress or release the accelerator pedal, we decided to turn a precision 10 turn potentiometer with the accelerator axis motor" (Nagy and Bock, 1995). Manual control can be obtained by changing potentiometers through the motion control card.

In addition to the two emergency, on-board and remote, stop capabilities, a safety circuit was built. The robot will stop if this circuit detects a computer fault, power off, or an emergency stop activation. This is accomplished by "overriding any computer accelerator and brake commands and replacing them with a zero accelerator input to the original... speed controller, and by directly applying the brake actuator to depress the brake pedal fully" (Nagy and Bock, 1995).

A camera is used as the sensor on board the robot. A 486 computer is also used with a frame grabber card for image processing. The navigation approach is described as "deliberative." Using this concept, the robot attempts to model the surroundings, plan motions in it, and carry out the planned motions. This is executed by taking the video input and attempting to model the environment. The computer then computes the

coordinates for the next motion segment. The motion control card coordinates all the actuators to carry out the planned motions.

This simple method presents a number of problems. By only using vision as the only sensing system, collision avoidance could be a problem. Problems could also arise if one or both of the boundary lines are out of the field of view of the camera.

### 6.3.2 Mean Machine

The second vehicle, the "Mean Machine", is a children's "Power Wheels" vehicle. Shown in Figure 6.3, this vehicle was chosen because of its stability, ruggedness, and low cost. It is also easy to automate with only the steering and acceleration systems to convert.

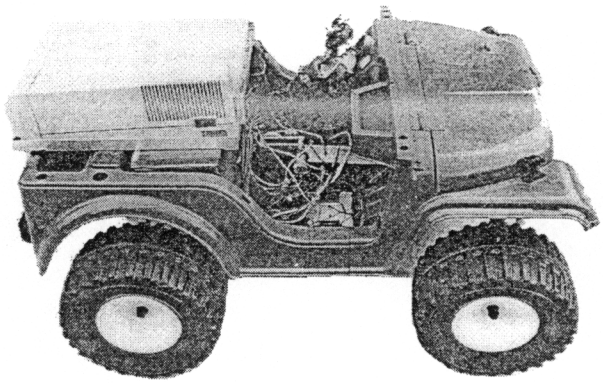

Figure 6.3: The "Mean Machine"

The steering system is a heavy duty DC windshield wiper motor. This drives a spur gear through a pinion to rotate a rod attached to the steering linkage. Limit switches are used to limit the motion to prevent damage to the steering mechanism. The accelerator

pedal was pressed through the use of a small motor that would push a lever just above the pedal. This lever would then press the accelerator pedal.

The "Mean Machine" used two sensors, a back-up scanner for obstacle avoidance and a red LED optic sensor for boundary line detection. The back-up scanner is essentially an ultrasonic sensor. It uses ultrasonic waves to determine the distance to an object. The control computer is a PC XT. A DC to AC converter is used to power the computer. A reactive navigation strategy using both sensors in a binary fashion is used on this robot. The back-up scanner is used as a binary switch to indicate if an obstacle was too close, not the exact distance to the object. If an obstacle is detected as being too close, the software is be adjusted to move the vehicle around the obstacle and then straighten out on the course. During this process, the robot will stop, turn its wheels and start back. The LED output indicated the presence or absence of a white line. It is mounted to one side of the vehicle, but can only see one line at a time. The navigation approach is to continually "bump" into the course path line, veer away, and then turn back in again to bump into the line again further down the course path. This is repeated until an obstacle is detected. This strategy also consists of stopping, adjusting the steering angle, and restarting. The obstacle avoidance function takes precedence over the white line detection strategy.

The main disadvantage is the constant stopping and restarting required by the vehicle. This builds a large time loss while going around the course. Having a sensor see only one line of the course presents its own problems. If the sensor loses that line, or if the

lines are dashed, there is no way to recover from going outside the boundary lines of the course.

## 6.4 Oakland University

Oakland University entered a robot named "UGLY" (UnGuided Land Yacht) in the 1993 Unmanned Ground Robotics Competition. "UGLY" is an electric, four-wheeled golf cart. On-board the vehicle are two cameras, a frame grabber card, five ultrasonic sensors, a forward/reverse controller, and a steering controller. Fuzzy logic was used to combine the sensor and actuator information. The on-board computer system was a 486 personal computer. Its functions were to process camera images, poll the sonar sensors, and compute the desired steering and speed values. The two microprocessors controlled the steering and speed systems. The steering system was the original rack and pinion design of the golf cart. This was driven by a 12 V DC motor. The speed regulation came from varying the output of six car batteries connected in series.

The navigation of the vehicle consisted of fusing the information from the vision and ultrasonic sensors. This enabled the vehicle to "perceive its environment, navigate around the track, and avoid obstacles on the competition course" (Cheok, 1994)

The motion of the vehicle is supervised by a three-tier hierarchical controller. The first level uses the information from the sensing systems and uses a supervisory fuzzy logic to decide which rules fit the current situation of the vehicle. Next, a command fuzzy logic takes the processed vision and sonar information and computes the desired

steering angle and speed. A direct control logic is then used to drive the speed and steering motors.

Fuzzy logic provides a different approach to the path planning and motion control of the vehicle than used by other teams. This could be a very effective strategy, but could be difficult to implement for navigation.

## 6.5 University of Cincinnati

The University of Cincinnati's 1995 entry into the competition was also a golf cart converted into a robotic vehicle. This three wheeled golf cart contained a 36 V, 55 amp, traction motor. A 486 computer was used for speed and steering control. For speed control, a commercially designed device was used. This was a silicon controlled rectifier, pulse-width modulation controller that received a 0-10 volt signal from a motor controller. An encoder was mounted on the front wheel for feedback information. The speed controller required three switches to be powered in a certain order before the vehicle would operate.

The original steering system on the golf cart was a rack and pinion design. This was changed to a lead screw design that used a linear actuator. A brushless motor amplifier controlled the motor and an encoder was mounted on the steering wheel for position feedback. The original braking system of the golf cart was kept intact. This was made autonomous by using a motor attached to a cable to pull the brake pedal down.

Obstacle avoidance was accomplished using six ultrasonic sensors. This sensor information was managed by a micro-controller that computed all of the timing and steering information. This information was relayed to the 486 computer. "The computer is always scanning for steering corrections due to obstacles. Once the angle is detected, the control program stops its calculations and corrects the steering. While the vehicle is in the obstacle avoidance mode, it completely ignores lane markers" (Matthews et al., 1995). This design makes the obstacle avoidance system separate from the other processing systems. Such an approach could be used on any robot.

The vision system on the vehicle consists of a CCD camera with a fish-eye lens. Matthews et al. (1995) state that the fish-eye lens was needed for higher speed and tight turns. The location of the camera is angled down at 32 degrees and panned to the right at 30 degrees. This configuration only gives a four foot wide view of the ground, which presented problems for the vehicle.

The motion control is accomplished using a hierarchical control system. The cart starts out by going straight at top speed and doesn't change until feedback from one of the sensor systems signals that a change is needed. At the next level, the vision system can correct the steering and, at the highest level, the ultrasonic sensors can override the straight motion and the vision system.

Improvements being made to the vehicle include adding more sensors at smaller angles. This need arises because their sensors were mounted low and picked up tall grass as an obstacle. The vision system was subject to periods of "white out" because of the

180 degree field of view of the fish-eye lens. They propose to correct this problem with a shade and a filter.

## 6.6 University of Colorado at Boulder

The University of Colorado at Boulder also entered a small "power wheels" vehicle called "RoboCar" into the 1995 competition. This was a small battery powered vehicle with a single 486 computer for all computations. This vehicle also contained a video camera that panned back and forth, frame grabber board for image processing and a single, panning ultrasonic sensor for obstacle detection. An encoder wheel was also used to determine the distance the vehicle had traveled.

The small vehicle was used because it's small size makes it easier to move around the track. The three speeds on the vehicle, low, high, and turbo, are electronically switched by checking the encoder for the desired velocity. By purchasing a vehicle with a chassis already intact, the team only had to reinforce the existing design to support the equipment. There were a few disadvantages to this small, commercially produced vehicle. One was the lack of a good suspension system. This was not a big factor since in the competition, the vehicle was moving a relatively slow speeds, but it could present a problem at faster pace. The three speeds on the vehicle were not enough to operate under all conditions. The biggest problems were the limitations of the drive motors. These prevented the vehicle from reaching the five mile per hour speed limit and also presented difficulties when going over a rough terrain or up inclines.

A panning video camera with a wide angle lens was used for image processing, a panning ultrasonic sensor was used for obstacle avoidance and a shaft encoder was used to obtain distance information. It was later discovered that additional ultrasonic sensors were needed because the panning sensor would sometimes miss obstacles when the vehicle was moving fast.

Three different battery systems were used on the vehicle. This enabled all of the equipment to be powered without interfering with each other. The drawback was that several different battery systems were needed. The rear wheels were powered by six volt batteries. By placing the batteries in series or parallel, the motors could run at 6 volts for low speed, 12 volts for higher speeds, or 18 volts for "turbo" operation. The vehicle also had reverse capabilities. A 12 volt battery was used to provide power to the emergency stop system. The third battery system consisted of two motorcycle batteries connected to an inverter to supply AC power to the computer and other electronics.

A commercially bought "servo motion control component packaged in a compact high integration module which includes a 600 watt brush motor servo amplifier, DSP motion controller, microcontroller, along with RS-232 and Controller Area Network (CAN) interfaces" was used to control the various systems on the vehicle (Gifford et al., 1995). Three of these devices were used get feedback from the encoder, control the panning motion of the camera and for steering control. A microcontroller was used to control the ultrasonic sensor, the panning of the sensor, the system power relays, battery voltage monitors, light emitting diode (LED) indicators, a horn and engine sound that was used

for software testing, and a serial interface to the PC. The panning ultrasonic sensor was a unique design. In the automatic mode, used for the competition, the sensor would fire at five different angles. The five ultrasonic sensor readings were processed and arranged in a polar occupancy grid. To account for vehicle motion, obstacles were increased in size (in software) as the scan proceeded from one side to the other. The LED's were used to signal if the battery voltage dropped below a designated value.

The system architecture was based mostly on a hierarchical architecture, but used a subsumption strategy to avoid the obstacles. Since relying on a purely reactive scheme is too slow for real-time navigation, they used known information directly in the programming such as the color of the grass and white lines, estimated size of the obstacles, and expected color of the sand.

For path planning, the University of Colorado team developed three separate strategies. The first was a Follow-the-Line concept. This strategy involved following one of the white lines on the course boundary. If this line was lost, the camera was simply panned to the other side to detect the other line. If neither line could be detected, the robot would move forward at a slow pace and continue panning the camera until a line was found. The second path planning strategy used was a "Stay-Centered" approach. The camera was tilted downward, straight in front of the vehicle, and panned in the direction the wheels are turned. A potential field approach was used by identifying the boundary lines as vectors perpendicular to the direction of motion. This method encountered problems with local and global minima/maxima and was not as effective as

the "Follow-the-Line" concept. The neural net approach suffered from the inability to conduct the numerous training sessions required for effective navigation. They point out that "it is important to train the neural net not only with the desired stay-centered and avoid-obstacle capability, but even more vital is to train the net with a myriad of scenarios that could possibly go wrong" (Gifford et al., 1995). Due to the problems of the last two methods, the "Follow-the-Line" strategy was used.

The procedure used for motion control included first mapping the ultrasonic readings. The vision information was processed and the steering angle was computed. This information was compared with the sonar information to see if the desired steering angle would cause the robot to collide with an obstacle. If the angle was not acceptable, the software would take over until the obstacle has been avoided. If no obstacles were in the path of the vehicle, it could proceed along the desired trajectory.

This vehicle was successful, probably as a result of the variety of applications that were investigated. By trying three different path planning methods, a successful solution was found. The small vehicle was a great advantage, but power issues need to be considered for navigating on a rough terrain.

## 6.7 Virginia Tech

Virginia Tech plans to have two vehicles for entry in the 1996 Unmanned Ground Robotics Competition. The first, BOB (Beast of Burden), is a children's all-terrain

vehicle. CALVIN (Computerized Autonomous Land Vehicle with Intelligent Navigation), is a golf cart sized vehicle.

### 6.7.1 BOB

The base of "BOB" is a children's all-terrain vehicle. A pentium computer, three car batteries, an inverter, an electric motor and controller are on board the vehicle. Three types of sensors are used on the vehicle: ultrasonics, tactile, and a vision system. Navigation involves a planning system using input from all sensors for environment reconstruction. A picture of "BOB" is shown in Figure 6.4.

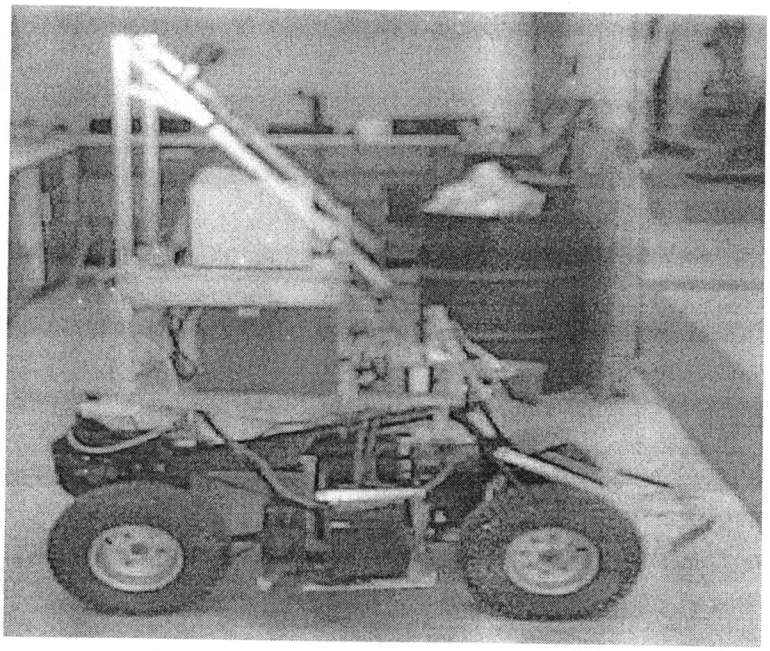

Figure 6.4: "BOB"

The original gasoline engine on "BOB" was replaced by electric motor for better speed control of the vehicle. Three car batteries are on the vehicle. Two are used to

power the inverter and the third is used to supply power to the steering motor. A 120 MHz pentium computer on BOB does most of the processing. A microcontroller is also on board for controlling the ultrasonic sensors. The steering system consists of a DC motor with potentiometer feedback. A vibration isolated plate has been built on the vehicle for carrying of the computer and other equipment. A Lexan shell provides the covering for the vehicle to protect equipment in adverse weather conditions.

The camera is the main sensor on the vehicle. It is used to detect the boundary lines of the course. This is accomplished by using a planning system that processes an image using smoothing and thresholding techniques. The edges are then found and a polynomial function is computed as the optimal path for the robot to follow. A new steering vector is then computed based on the difference between the actual and desired path.

Obstacle avoidance is achieved using other sensing systems. Six ultrasonic sensors are placed on the front bumper of the vehicle. These are used as the primary source for obstacle information. Two tactile "banks", each consisting of four sensors, are used in cases where the sonar sensor fail to detect obstacles. If the robot collides with an obstacle, the tactile sensor banks will provide information showing which side of the vehicle the obstacle is located. This information is compared with the vision system and a new steering angle is found.

The motion control of the vehicle is accomplished by continuously checking the steering angle of the vehicle. The computer receives camera information from the frame

grabber and ultrasonic feedback from the micro controller. Based on the location of the boundary lines and obstacles, a new steering angle is determined. The speed of the vehicle is also adjusted by comparing the steering angle and the location of the obstacles.

### 6.7.2 CALVIN

"CALVIN", shown in Figure 6.5, is based on a three-wheel golf cart. On board "CALVIN" is a personal computer, car battery, two inverters, and an electric motor. Sensing involves two cameras, six ultrasonic sensor, and tactile sensors. A reactive system is used by the vision system for planning a path based on the boundary lines. This information is then updated by other sensors for the determination of the final steering angle.

The original gasoline engine on "CALVIN" was replaced by an electric motor to eliminate problems associated with controlling the speed of the engine. The original brake system on the cart was retained, but foot actuation was replaced by a pneumatic braking system. The rack and pinion steering was replaced with a linear actuator. A Lexan shell was built to protect the electronic equipment.

Two cameras are used on-board for boundary line detection. The reactive system involves taking the pixel data and generating boundary lines. A desired steering vector is then generated based on the location of the lines.

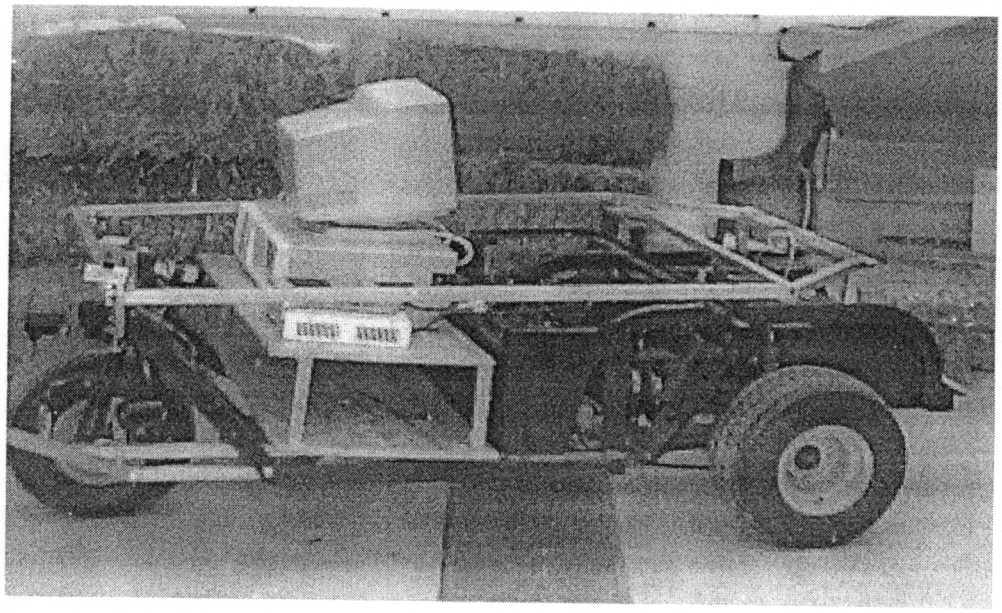

Figure 6.5: "CALVIN"

The steering vector produced by the vision system can be modified by the ultrasonic or tactile sensors. Both sets of sensors are placed on the front bumper of the vehicle. The sonar provide the main source for obstacle detection. The ultrasonics are polled individually to see if an obstacle is detected. Tactile sensors are arranged on the front bumper as a last resort in case the robot misses an obstacle with its vision and ultrasonics. This information is used to update the steering angle determined by the vision system.

## 6.8 West Virginia University

West Virginia University entered a vehicle called the "ANT 2" (Autonomous Navigation Testbed 2) into the 1995 competition. Shown in Figure 6.6, their vehicle is unique because it is tracked rather than wheeled. The base is a Honda Power Carrier

construction vehicle that was stripped down to the tracks. The original vehicle contained a gasoline engine that was replaced by two, one horsepower DC motors. This enables each track to be driven independently through 30:1 worm drive gear reducers and a chain/sprocket system. Power is provided by eight 12V car batteries connected in series. Two personal computers are mounted on the vehicle, one 486 and a 386. The 486 computer contains a frame grabber card is used for vision processing. The 386 computer handles the sensing needs and communicates with the vision computer. The sensors on board the vehicle are a video camera on a pan/tilt head, seven ultrasonic sensors, pitch and roll inclinometers, and a battery voltage monitor.

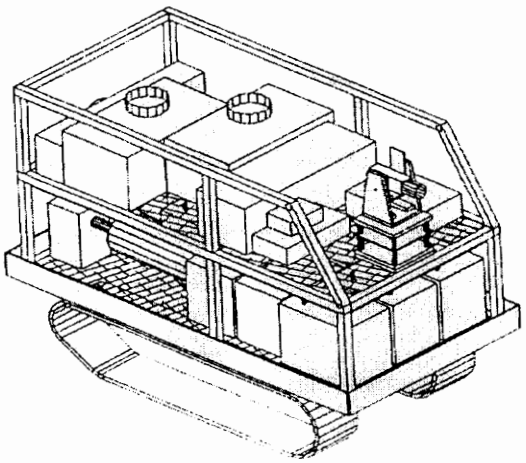

Figure 6.6: "ANT 2"

The robot is controlled by a "main control program". This program calls the other programs that perform the vision and sensor processing, map representation and fusion, path planning functions and motion control. Two modes of operation have been developed, a "Program Editing Mode" and a "Contest Mode". Banta (1995) says that the

"general strategy for the Contest Mode is to construct a map of the robot's position in the environment using sensor inputs, to identify obstacles and their locations relative to the robot on the map, to plan a collision-free course past those obstacles and along the track, and to issue appropriate commands to the motor controllers to move the robot along the desired trajectory."

The primary sensing is performed by the video camera. The image is captured and reduced in resolution to help eliminate any bad information. The information is thresholded to locate the small groups of pixels that could be boundary lines or larger groups that are obstacles. This image is interpolated to make a constant image. A video map of ones and zeroes is created and passed to the map fusion algorithm, along with the equations of the left and right boundary lines. The ultrasonic sensors are placed on the front and sides of the vehicle. They are fired individually and the information is combined to form a sonar map. This information is fused with the video map to create one representation for path planning.

These two maps were fused together after correcting the distortion of the video images and making the sonar data into a similar map. The goal of the group was to compare the newly created map with a previous map and with estimates of the motion of the robot between images. The "features of the new map would be overlaid on features from the old map and the motion estimates are corrected to achieve the best fit" (Banta, 1995) The final map used for path planning, used the information from the sensor maps and the line equations. To aid the robot if both boundary lines are not in the field of view, the

information is placed on a general grid and the robot is given its location between the boundary lines and its angle relative to the path. Obstacles from the current and prior maps are shown and scrolled down with the motion of the robot.

A modified potential field approach was used for path planning. They use the "particle in the stream" effect, where obstacles generate circular rotation fields to push the vehicle around obstacles. The problem of local minima is avoided with this approach.

The motion control involved calculating the location of the robot relative to the path. Feedback is given if the motion is along the desired path. If the robot is not on the correct path, the speed remains constant, and all turns are at a fixed radius of curvature. The motors are then driven by solid state pulse width modulated controllers that use tachometer feedback to close the loop.

Some of the improvements being made to this vehicle include reducing the battery voltage to 48 volts. This will create a safer and lighter robot. They also plan to rebuild the frame to help reduce weight and lower the center of gravity for more stability on inclines. Since software is the "brains" of the vehicle, it will be constantly and critically improved.

## 6.9 Summary

These are only seven of the approximately fifteen schools that have entered the Unmanned Ground Robotics Competition. As can be seen from the above summaries, the approaches used by the different schools varies greatly. Some vehicles were designed

and built from the ground up, other teams modified a store-bought base. The size ranged from the large golf cart size of Virginia Tech to the small children's toy of the Colorado School of Mines. Most of the teams used a video camera for image processing, but obstacle avoidance was accomplished using a camera, or any number of ultrasonic sensors. The path planning strategies range from the simple, University of Colorado at Boulder's follow-the-line strategy, to more complex, "particle in the stream" plan by West Virginia University. The motion control for most teams consisted of a simple feedback loop reporting the difference between the actual and desired steering angle. Safety was a concern for all teams. A good design for safety was by Northern Illinois University. They ensured safety by making human overrides for each system.

Everyone who has participated in these events has an excellent opportunity to explore different areas of mobile robotics. Cheok (1994) lists the five main benefits of this project as: introducing students to intelligent robotics problems early in their college career, helping students become aware of advances in sensors, computers actuators, and system integration technology, exposing students to methodology advances in artificial intelligence and intelligent control disciplines, helping professors who sponsor the teams find research opportunities in terms of topics, concepts, and possibly funding, and showing that useful autonomous applications can be found in military application, exploration application and commercial application.

# Chapter 7

# Summary

The development of autonomous mobile robots is an area that has been growing in interest. Robots are now being used in manufacturing environments, nuclear waste disposal locations, and even to go to Mars. As the desire to eliminate functions that humans find difficult or tedious continues to increase, robot technology will remain in high demand. This thesis presented a survey of approximately 200 articles in the area of robotics. The perceiving, thinking, and acting system of a mobile robot were discussed. The integration and coordination of these behaviors into a system architecture was also considered. A smaller survey of vehicles that have entered the Unmanned Ground Vehicle Competition was included to show the integration of all the systems into an autonomous mobile robot. Some final conclusions drawn from the articles surveyed and closing remarks are included in the remainder of this chapter.

In order to determine what topics in the areas of mobile robots were popular, the keywords used to refer to each article were compiled into a list of top keywords for each year. Table 7.1 shows the top five keywords for each year.

Table 7.1: Top Keywords

| Year | Top Keywords |
|---|---|
| 1990 | autonomous<br>ultrasonics<br>sensors<br>intelligent<br>real-time |
| 1991 | mobile robot<br>navigation<br>sensors<br>algorithms<br>computer vision |
| 1992 | mobile robot<br>autonomous<br>algorithms<br>navigation<br>control system |
| 1993 | mobile robot<br>navigation<br>autonomous<br>algorithms<br>computer vision |
| 1994 | navigation<br>mobile robot<br>motion control<br>robotics<br>autonomous |
| 1995 | mobile robot<br>fuzzy<br>collision avoidance<br>motion control<br>navigation |

This table brings up several expected conclusions, but additional keywords present several interesting points. Based on the articles used in this survey, the areas of mobile robots, navigation and motion control have been topics widely considered over the past

few years. "Mobile robots" was by far the top keyword used over this six-year span. This was as expected, but looking a the keyword lists over the course of the years and not just the top five from each year, the introduction of new concepts can be distinguished. The evolution of fuzzy logic and neural networks is clearly defined. In the early 90's these two keywords were not used much at all. In 1993 the use of fuzzy logic and neural networks can be seen to be more popular with 1995 having many articles on both areas. The use of more advanced control systems, such as learning systems, show an increase in publications in 1994. Although this survey is by no means complete, the trends shown by these keywords provide a good idea of the development of robotic technology.

A search using the citation index on all of the articles referenced was performed to see what authors and/or journals are commonly cited. The citation index is a listing of the sources that have referenced a particular article in a specified year. There is a citation index for each year. Therefore, to do a search on an article published in a certain year, the article must be looked-up in that year and each year after that. If an article was published in 1992, the years 1992, 1993, 1994, 1995 and 1996 must be searched to see how many times the article was referenced by other authors.

Since the majority of the references included in this thesis were published in the 1990's, this search did not produce helpful information. Since most of the research was new, there had not been much time for the work to be seen by other authors. Most often, when the article was cited, it was in another publication by the same author. One reference did stand out as being used often, Brooks (1986). This article is where the

subsumption architecture is first introduced. This is also an older article having more time to be absorbed into the research community. From the articles used in this survey, the leading American universities in the area of mobile robotics research are Carnegie Melon University, Massachusetts Institute of Technology, and the University of Michigan. Each of these universities has several articles that were included in the survey.

Interest in the area of mobile robotics will continue to increase. The larger demand for mobile robots will continue to push current applications leading to more advanced systems. Although the robots depicted in science fiction books are still far from a reality, it can be guaranteed that technology will not rest until fantasy finally becomes reality.

# References


1. Abbott, D., Yakovleff, A., Moini, A., Nguyen, X. T., Blanksby, A., Beare, R., Beaumont-Smith, A., Kim, G., Bouzerdoum, A., Bogner, R. E. and Eshraghian, K., "Biologically Inspired Obstacle Avoidance- a Technology Independent Paradigm," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 2-12.

2. Agah, A. and Bekey, G. A., "Novel Cognitive Architecture for Simulated Robots in an Artificial World," *Proceedings - IEEE International Conference on Robotics and Automation*, Part 3, 1994, pp. 2309-2314.

3. Allen, D. C., Hayward, M., Sievers, R. H. and Cook, L., "Hardening Automated Equipment for Hostile Environments," *Proceedings of the Conference on Remote Systems Technology*, November, 1991, pp. 36-42.

4. Andrews, J. R. and Hogan, N., "Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator," *Control of Manufacturing Processes and Robotic Systems*, 1983, pp. 243-251.

5. Aoki, T., Matsuno, M., Suzuki, T. and Okuma, S., "Motion Planning for Multiple Obstacles Avoidance of Autonomous Mobile Robot Using Hierarchical Fuzzy Rules," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems,* 1994, pp. 265-271.

6. Arkin, R. C., "Homeostatic Control for a Mobile Robot. Dynamic Replanning in Hazardous Environments," *Journal of Robotic Systems*, Vol. 9, No. 2, March 1992, pp. 197-214.

7. Attolico, G., Caponetti, L., Chiaradia, M. T. and Distante, A., "Robot Vision System for Obstacle Avoidance Planning," *Mobile Robots V, SPIE,* Vol. 1388, 1990, pp. 50-61.

8. Badal, S., Ravela, S., Draper, B. and Hanson, A., "Practical Obstacle Detection and Avoidance System," *IEEE Workshop on Applications of Computer-Vision - Proceedings*, 1994, pp. 97-104.

9. Banta, L. E., "Undergraduate Robot Design at West Virginia University," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 202-208.

10. Banta, L., Moody, J. and Nutter, R., "Neural Networks for Autonomous Robot Navigation," *Manufacturing Systems Development and Applications Department Conference Record-IAS Annual Meeting*, Vol. 3, 1993, pp. 2459-2462.

11. Banta, L. E., Nutter, R. S. and Xia, Y., "Mode-Based Navigation for Autonomous Mine Vehicles," *IEEE Transactions on Industry Applications*, Vol. 28, No. 1, January-February, 1992, pp. 181-185.

12. Baxter, J. W. and Bumby, J. R., "Fuzzy Control of a Mobile Robotic Vehicle," *Proceedings of the Institution of Mechanical Engineers, Part I, Journal of Systems and Control Engineering*, Vol. 209, No. 2, 1995, pp. 79-91.

13. Bay, J. S., "Learning Classifier Systems for Single and Multiple Mobile Robots in Unstructured Environments," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 88-99.

14. Beckwith, T. G., Marangoni, R. D., Lienhard, J. H. V, *Mechanical Measurements, 5th ed.*, Addison-Wesley Publishing Co., Reading, MA, 1993.

15. Bender, D. and Zarlingo, S. P., "Spring Metals for Terminals and Connectors in Harsh Environments," *Proceedings of the 42nd Electronic Components and Technology Conference*, May, 1992, pp. 297-305.

16. Beom, H. R. and Cho, H. S., "Sensor-Based Navigation for a Mobile Robot Using Fuzzy Logic and Reinforcement Learning," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 3, March, 1995.

17. Bobick, A. F. and Bolles, R. C., "Multiple Concurrent Object Descriptions in Support of Autonomous Navigation," *Sensor Fusion V, SPIE*, Vol. 1828, November, 1992, pp. 383-392.

18. Bonasso, R. P., Antonisse, H. J. and Slack, M. G., "A Reactive Robot System for Find and Fetch Tasks in an Outdoor Environment," *Proceedings Tenth National Conference on Artificial Intelligence AAAI,* 1992, pp. 801-808.

19. Borenstein, J., "Control and Kinematic Design of Multi-Degree-of-Freedom Mobile Robots with Compliant Linkage," *IEEE Transactions on Robotics and Automation,* Vol. 11, No. 1, February 1995, pp. 21-35.

20. Borenstein, J. and Feng, L., "UMBmark: A Benchmark test for Measuring Odometry Errors in Mobile Robots," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 113-124.

21. Borenstein, J. and Koren, Y., "Error Eliminating Rapid Ultrasonic Firing for Mobile Robot Obstacle Avoidance," *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 1, February, 1995, pp. 132-138.

22. ----- "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 4, August, 1991, pp. 535-539.

23. ----- "Real-Time Map-Building for Fast Mobile Robot Obstacle Avoidance," *Mobile Robots V, SPIE,* Vol. 1388, 1990, pp. 74-81. (1990a)

24. ----- "Real-Time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments," *IEEE International Convention of Robotics and Automation,* Vol. 1, 1990, pp. 572-577. (1990b)

25. ----- "Real-Time Obstacle Avoidance for Fast Mobile Robots," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 5, September/October, 1989, pp. 1179-1187.

26. Borthwick, S. and Durrant-Whyte, H., "Simultaneous Localization and Map Building for Autonomous Guided Vehicles," *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Vol. 2, 1994, pp. 761-768.

27. Boshra, M. and Zhang, H., "Use of Tactile Sensors in Enhancing the Efficiency of Vision-Based Object Localization," *Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems(MFI '94),* October, 1994, pp. 243-250. (1994a)

28. ----- "Use of Visual and Tactile Data for Generation of 3-D Pose Hypotheses," *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems,* Vol. 1, 1994, pp. 73-80. (1994b)

29. Braunegg, D. J., "MARVEL: A System That Recognize World Locations With Stereo Vision ," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 3, June, 1993, pp. 303-308.

30. Brooks, R., "A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network," *International Conference on Robotics and Automation*, 1989, pp. 692-696.

31. ----- "A Robust Layered Control System For A Mobile Robot," *IEEE Journal of Robotics and Automation*, Vol. RA-2, No. 1, March, 1986, pp. 14-23.

32. Brown, M., Fraser, R., Harris, C. and Moore, C. G., "Intelligent Self-Organizing Controllers For Autonomous Guided Vehicles Comparative Aspects of Fuzzy Logic and Neural Nets," *International Conference on CONTROL, IEE Conference Publication*, Vol. 1, No. 332, March, 1991, pp. 134-139.

33. Brumitt, B. L., Coulter, R. C. and Stentz, A., "Dynamic Trajectory Planning for a Cross-Country Navigator," *Mobile Robots VII, SPIE,* Vol. 1831, 1993, pp. 564-575.

34. Burke, D. M. and Rattan, K. S., "Multi-Layered Motion Controller for a Mobile Robot Implemented With Fuzzy Logic," *Proceedings of the 1993 American Control Conference*, 1993, pp. 2248-2251.

35. Button, K. J. and Wiltse, J. C. ed. *Infrared and Millimeter Waves, Volume 4, Millimeter Systems*, Academic Press, New York, 1981.

36. Canistraro, H. A., Jordan, E. H. and Pease, D. M., "X-ray Based Displacement Measurement for Hostile Environments," *Experimental Mechanics*, Vol. 32, No. 4, December, 1992, pp. 289-295.

37. Cardoso, F., Fontes, F., Paris, C., Oliveira, P. and Ribeiro, M. I., "Fuzzy Logic Steering Controller for Guided Vehicle," *Mediterranean Electrotechnical Conference - MELECON* 2, 1994, pp. 711-714.

38. Cheok, K. C., "Autonomous Unmanned Ground Robotic Vehicle Competition: An Intelligent Control Challenge," *Proceedings of the American Control Conference*, Vol. 1, 1994, pp. 383-387.

39. Cho, D. W. and Lim, J. H., "A New Certainty Grid Based Mapping and Navigation System for an Autonomous Mobile Robot," *International Journal of Advanced Manufacturing Technology*, Vol. 10, No. 2, 1995, pp. 139-148.

40. Chun, W. H. and Jochem, T. M., "Unmanned Ground Vehicle Demo II: Demonstration A," *Mobile Robots IX, SPIE,* Vol. 2352, 1995, pp. 180-191.

41. Craig, J. J., *Introduction to Robotics: Mechanics and Control, 2nd. Ed.*, Addison-Wesley Publishing Co., Reading, MA., 1989.

42. Crisman, J. D. and Webb, J. A., "The Warp Machine on Navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 5, May, 1991, pp. 451- 465.

43. Critchlow, A. J., *Introduction to Robotics*, Macmillan Publishing Co., New York, 1985.

44. Davis, I. L. and Stentz, A., "Sensor Fusion for Autonomous Outdoor Navigation Using Neural Networks," *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, 1995, pp. 338-343.

45. Davis, L. S., "Visual Navigation at the University of Maryland," *Robotics and Autonomous Systems*, Vol. 7, No. 2-3, August, 1991, pp. 99-111.

46. DeSaussure, G., Weisbin, C. R., and Spelt, P. G., "Navigation and Learning Experiments By An Autonomous Robot," *Robotics and Computer-Integrated Manufacturing*, Vol. 6., No. 4, 1989, pp. 295-301.

47. Dickmanns, E. D., Mysliwetz, B. and Christians, T., "An Integrated Spatio-Temporal Approach to Automatic Visual Guidance of Autonomous Vehicles," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 20, No. 6, November/December 1990, pp. 1273-1284.

48. Dodds, D. R., "Terrain Classification in Navigation of an Autonomous Mobile Robot," *Mobile Robots V, SPIE*, Vol. 1388, November, 1990, pp. 82-89.

49. Dudek, G., Freedman, P. and Hadjres, S., "Using Uncertain Sensing Data to Create Reliable Maps: an Algorithm for Exploring/Mapping Unknown Graph-Like Worlds," *Mobile Robots VII, SPIE,* Vol. 1831, 1993, pp. 650-660.

50. Elfes, A., "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer Magazine*, Vol. 22, No. 8, June, 1989, pp. 46-57.

51. El-Konyaly, E. H., Enab, Y. M. and Abdel-Kareem, E. M., "Obstacle Avoidance and Motion Control for Mobile Robots," *Mobile Robots IX, SPIE*, Vol. 2352, 1995,

52. El-Konyaly, E. H., Areed, F., Enab, Y. and Zada, F., "Range Sensory Based Robot Navigation in Unknown Terrains," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 76- 85. (1995b)

53. ----- "Support System for Mobile Robot Steering and Control," *Mobile Robots IX, SPIE*, Vol. 2352, 1995, pp. 24-37. (1995c)

54. Fayek, R. E., Liscano, R. and Karam, G. M., "A System Architecture for a Mobile Robot Based on Activities and a Blackboard Control Unit," *IEEE International Conference on Robotics and Automation,* Vol. 2, 1993, pp. 267-274.

55. Feng, D. and Krogh, B. H., 'Dynamic Steering Control of Conventionally Steered Mobile Robots," *Journal of Robotic Systems*, Vol. 8, No. 5, October, 1991, pp. 699-721.

56. ----- "Satisficing Feedback Strategies for Local Navigation of Autonomous Mobile Robots," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 6, November-December, 1990, pp. 1383-1395.

57. Fernandez, J., Martinez, A. B. and Frau, J., "Visual Guidance of an Autonomous Vehicle in Natural Environments Based on Reflexive Control," *Intelligent Vehicles Symposium*, *Proceedings,* 1994, pp. 103-108.

58. Figueroa, F. and Mahajan, A., "A Robust Navigation System for Autonomous Vehicles Using Ultrasonics," *Control Engineering Practice*, Vol. 2, No. 1, February, 1994, pp. 49-59.

59. Fogle, R. F., "The Use of Teleoperators in Hostile Environment Applications," *Proceedings 1992 IEEE International Conference on Robotics and Automation*, Vol. 1, May, 1992, pp. 61-66.

60. Froom, R., "Acquiring Effective Knowledge of Environment Geometry for Minimum-Time Control of a Mobile Robot," *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, August, 1991, pp. 501-506.

61. Froschle, B., Bruemmer, H. P., Lang, W., Neumeier, K. and Ramm, P., "Fabrication of High Quality MOS Devices for Application in Hazardous Environments Based on RTP Gate Dielectrics With in Situ RTCVD of Polysilicon Gates," *III-V Electronic and Photonic Device Fabrication and Performance Materials Research Society Symposium Proceedings*, Vol. 300, 1993, pp. 587-594.

62. Fualdes, T. D. and Barrouil, C. J., "A Common Framework for Reasoning on Uncertainty Both at Symbolic and Numerical Levels," *Future Generations Computer Systems*, Vol. 9, No. 4, December, 1993, pp. 339-347.

63. Funkunaga, K., Asano, T., Murata, H. and Izumi, M., "Image Matching Using Structures of Edge-Image Graphs," *Systems and Computers in Japan*, Vol. 22, No. 11, 1991, pp. 61-71.

64. Gat, E., " Navigation Templates: Enhancements, Extensions and Experiments," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, 1993, pp. 541-547.

65. ----- "Integrating Planning and Reacting in a Heterogeneous Asynchronous Architecture for Controlling Real-World Mobile Robots," *Proceedings of the Tenth National Conference on Artificial Intelligence*, 1992, pp. 809-815.

66. Gifford, K. K., Deeds, M., van der Hoek, A., Henning, F., Freeman, J., Haussman, G., Kruzminsky, S. and Stoller, S., "RoboCar: Software, Hardware, and Mechanical Design Issues for the University of Colorado Autonomous Rover Vehicle," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 228-238.

67. Giralt, G., Chatila, R., Alami, R., "Remote Intervention, Robot Autonomy, and Teleprogramming: Generic Concepts and Real-World Application Cases," *1993 International Conference on Intelligent Robots and Systems*, 1993, pp. 314-320.

68. Good, T. T., " Blank-Map Orienteering for a Mobile Robot Using Certainty Grids," *Mobile Robots VII, SPIE,* Vol. 1831, 1993, pp. 631-642.

69. Gourley, C. and Trivedi, M., "Sensor Based Obstacle Avoidance and Mapping for Fast Mobile Robots," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, Part 2, 1994, pp. 1306-1311.

70. Gowdy, J., Stentz, A., and Hebert, M., "Hierarchical Terrain Representations for Off-Road Navigation," *Mobile Robots V, SPIE,* Vol. 1388, November, 1990, pp. 131-140.

71. Graettinger, J. T. and Krogh, B. H., "Evaluation and Time-Scaling of Trajectories for Wheeled Mobile Robots," *ASME Journal of Dynamic Systems, Measurement, and Control,* Vol. 11, June, 1989, pp. 184-231.

72. Guldner, J., Utkin, V. I, and Bauer, R., "Mobile Robots in Complex Environments: A Three-Layered Hierarchical Path Control System," *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Vol. 3, 1994, pp. 1891-1898.

73. Hampapur, A., Pingali, G. S. and Jain, R., "Simulation for Outdoor Navigation," *Mobile Robots VII, SPIE,* Vol. 1831, 1992, pp. 552-563.

74. Harrigan, R. W., "Automating the Operation of Robots in Hazardous Environments," *1993 International Conference on Intelligent Robots and Systems*, Part 2, 1993, pp. 1211-1219.

75. ----- "Automated Robotic Radiation Surveys," *IEEE International Conference on Systems Engineering*, August, 1989, PP. 363-366.

76. Heikkonen, J. and Oja, E., "Self-Organizing Maps for Visually Guided Collision-Free Navigation," *Proceedings of 1993 International Joint Conference on Neural Networks,* 1993, pp. 1-4.

77. Herbert, M., "Pixel-Based Range Processing for Autonomous Driving," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*," Part 4, 1994, pp. 3362-3367.

78. Hou, E. S. H. and Zheng, D., "Hierarchical Path Planning With Hexagonal Decomposition, *Proceedings of the 1991 IEEE International Conference on Systems*, Man and Cybernetics, Vol. 2, October, 1991, pp. 1005-1010.

79. Huber, J. and Graefe, V., "Motion Stereo for Mobile Robots," *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 4, August, 1994, pp. 378-383.

80. Hugli, H., Tieche, F. and Facchinetti, C., "Integration of Vision in Autonomous Mobile Robotics," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics,* Vol. 1, 1994, pp. 1047-1052.

81. Iida, S. and Yuta, S., "Control of Vehicle With Power Wheeled Steering Using Feedforward Dynamics Compensation," *Proceedings of the 1991 International Conference on Industrial Electronics, Control and Instrumentation - IECON '91*, Vol. 3, October-November, 1991, pp. 2264-2269.

82. Ikegami, T., Kato, J. and Ozono, S., "Representation of Knowledge and Sensor Information Based on the Distance Value Model for Autonomous Mobile Robot," *Systems and Computers in Japan*, Vol. 21, No. 13, 1990, pp. 54-68.

83. Inigo, R. M. and Torres, R. E., "Mobile Robot Navigation with Vision Based Neural Networks," *Mobile Robots IX, SPIE,* Vol. 2352, 1994, pp. 68-79.

84. Inigo, R. M. and Alley, D., "Path Planning Algorithms for Mobile Robots," *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, September, 1990, pp. 1032-1036.

85. Ishiguro, H., Kato, K. and Tsuji, S., "Multiple Vision Agents Navigating a Mobile Robot in a Real World," *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 1, 1993, pp. 772-777. (1993a)

86. Ishiguro, H., Ueda, K. and Tsuji, S., "Omnidirectional Visual Information for Navigating a Mobile Robot," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, 1993, pp. 799-804. (1993b)

87. Ishiguro, H., Yamamoto, M., and Tsuji, S., "Acquiring Omnidirectional Range Information," *Systems and Computers in Japan*, Vol. 23, No. 4, 1992, pp. 47-56.

88. Janabi-Sharifi, F. and Vinke, D., "Robot Path Planning by Integrating the Artificial Potential Field Approach With Simulated Annealing," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, 1993, pp. 282-287.

89. Janet, J. A., Luo, R. C. and Kay, M. G., " Essential Visibility Graph: An Approach to Global Motion Planning for Autonomous Mobile Robots," *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, Vol. 2, 1995, pp. 1958-1963.

90. ----- "T-Vectors Made Autonomous Mobile Robot Motion Planning and Self-Referencing More Efficient," *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Vol. 1, 1994, pp. 587-594.

91. Jarvis, R., "Converting Outdoor Vehicles Into Autonomous Mobile Robots," *1993 International Conference on Intelligent Robots and Systems*, 1993, pp. 591-597.

92. Jones, J. L. and Flynn, A. M., *Mobile Robots: Inspiration to Implementation*, A. K. Peters, Wellesley, MA, 1993.

93. Jorg, K., "World Modeling for an Autonomous Mobile Robot Using Heterogeneous Sensor Information," *Robotics and Autonomous Systems*, Vol. 14, No. 2-3, May, 1995, pp. 159-170.

94. Kachroo, P., "Setup for Advanced Vehicle Control Systems Experiments in the Flexible Low-Cost Automated Scaled Highway (FLASH) Laboratory," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 269-278.

95. Kadonoff, M. B. and Parish, D. W., "Modular Control Systems for Teleoperated and Autonomous Mobile Robots," *Mobile Robots IX, SPIE*, Vol. 2352, 1995, pp. 267-275.

96. Kamberova, G., McKendall, R. and Mintz, M., "Multivariate Data Fusion Based on Fixed-Geometry Confidence Sets," *Sensor Fusion IV: Control Paradigms and Data Structures, SPIE*, Vol. 1611, November, 1991, pp. 28-34.

97. Kanade, T., Hertzberger, L. O. and Groen, F. C. A., "Second International Conference on Intelligent Autonomous Systems," *Robotics and Autonomous Systems*, Vol. 7, No. 2-3, August, 1991, pp. 83-84.

98. Kang, D., Hashimoto, H., and Harashima, F., "Position Estimation for Mobile Robot Using Sensor Fusion," *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, 1995, pp. 271-276.

99. Kelly, A., Stentz, A. and Hebert, M., "Terrain Map Building for Fast Navigation on Rugged Outdoor Terrain," *Mobile Robots VII, SPIE*, Vol. 1831, 1993, pp. 576-589.

100. Khatib, O., "Real-Time Obstacle Avoidance for Manipulatiors and Mobile Robots," *IEEE Conference on Robotics and Automation*, 1990, pp. 500-505.

101. Kim, K. I., Oh, S. Y., Kim, S. W., Jeong, H., Han, J. H., Lee, C. N., Kim, B. S. and Kim, C. S., "An Autonomous Land Vehicle PRV II: Progress and Performance Enhancement," *1995 International Vehicle Symposium,* 1995, pp. 264-269.

102. Kim, K. I., Oh, S. Y., Lee, J. S., Han, J. H. and Lee, C. N., "An Autonomous Land Vehicle: Design Concept and Preliminary Road Test Results," *1993 International Vehicle Symposium,* 1993, pp. 146-151.

103. Knick, M. and Schlegel, C., "AMOS: Active Perception of an Autonomous System," *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems,* Part 1, Vol. 1, 1994, pp. 281-289.

104. Korba, L., Elgazzar, S. and Welch, T., "Active Infrared Sensors for Mobile Robots," *IEEE Transactions on Instrumentation and Measurement*, Vol. 43, No. 2, April, 1994, pp. 283-287.

105. Koren, Y. and Borenstein, J., "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 2, April, 1991, pp. 1398, 1404.

106. Koskinen, K., Makela, H., Rintanen, K., Halme, A., Ojala, M., Suomela, J. and Schonberg, T., "An Experimental Autonomous Land Vehicle for Off-Road Piloting and Navigation Research," *Proceedings of 1993 International Conference on Systems, Man and Cybernetics*, Vol. 4, 1993, pp. 151-156.

107. Kurz, A., "ALEF: An Autonomous Vehicle Which Learns Basic Skills and Constructs Maps for Navigation," *Robotics and Autonomous Systems*, Vol. 14, No. 2-3, May, 1995, pp. 171-183.

108. Lee, S. S. and Williams, J. H., "Fast Tracking Error Control Method for an Autonomous Mobile Robot," *Robotica,* Vol. 11, Part 3, May-June, 1993, pp. 209-215.

109. Lee, S. S., Williams, J. H. and Rayment, P. J., "An Automatic Guidance System of an Autonomous Vehicle-the Trajectory Generation and the Control Algorithm," *Robotica,* Vol. 11, 1993, pp. 309-314.

110. Lim, J. H. and Cho, D. W., "Experimental Investigation of Mapping and Navigation Based on Certainty Grids Using Sonar Sensors," *Robotica*, Vol. 11, Part 1, January-February, 1993, pp. 7-17.

111. Liscano, R., Manz, A. and Stuck, E. R., "Using a Blackboard to Integrate Multiple Activities and Achieve Strategic Reasoning for Mobile-Robot Navigation," *IEEE Expert,* Vol. 10, No. 2, April 1995, pp. 24-36.

112. Liu, K. and Lewis, F. L., "Fuzzy Logic-Based Navigation Controller for an Autonomous Mobile Robot," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics,* Vol. 2, 1994, pp. 1782-1789.

113. Luo, R. C. and Pan, T., "Maintaining Knowledge for an Adaptive Path Planning System," *Applications of Artificial Intelligence VIII, SPIE*, Vol. 1293, Part 1, April, 1990, pp. 416-426.

114. Lux, P. W. and Schaefer, C. H., "Range Imaging for Autonomous Navigation of Robotic Land Vehicles," *Signal Processing*, Vol. 22, No. 3, March, 1991, pp. 299-311.

115. McCarthy, W. E. and Trabia, M. B., "Path Planning of an Autonomous Vehicle Operating in an Unknown Environment Using a Fuzzy Logic Controller," *Dynamic Systems and Control (Vol. 1 of 2) American Society of Mechanical Engineers*, DSC 55-1, 1994, pp. 379-388.

116. Malik, R., "Location by Collision," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, October, 1991, pp. 877-882.

117. Malik, R. and Prasad, S., "Robot Mapping in Unstructured Environments," *Mobile Robots VI, SPIE,* Vol. 1613, 1991, pp. 181-189.

118. Manigel, J. and Leonhard, W., "Vehicle Control by Computer Vision," *IEEE Transactions on Industrial Electronics,* Vol. 39, No. 3, June, 1992, pp. 181-188.

119. Maravall, D., Mazo, M., and Fernandez, E., "Range Computation for a Mobile Robot Using Stereoscopic Vision," *Proceedings of the 6th Mediterranean Electrotechnical Conference-Melecon '91*, May, 1991, pp. 1242-1245.

120. Martinez, A., Tunstel, E. and Jamshidi, M., "Fuzzy Logic Based Collision Avoidance for a Mobile Robot," *Robotica,* 12, Part 6, November-December, 1994, pp. 521-527.

121. Maruya, Y., Takahashi, H., and Kutami, A., "Development of an Autonomous On-Road Vehicle," *Proceedings of the 6th International Pacific Conference on Automotive Engineering,* October-November, 1991, pp. 833-839.

122. Matthews, B., Ruthemeyer, M., Perdue, D. and Hall, E. L., "Development of a Mobile Robot for the 1995 AUVS Competition," *Mobile Robots X, SPIE*, Vol. 2591, 1995, pp. 194-201.

123. Matthies, L., Kelly, A., Litwin, T. and Tharp, G., "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report," *1995 International Vehicle Symposium,* 1995, pp. 66-71.

124. Meng, M. and Kak, A. C., "Mobile Robot Navigation Using Neural Networks and Nonmetrical Environment Models," *IEEE Control Systems,* Vol. 13, No. 5, October, 1993, pp. 30-39.

125. Mitchell, M., "Adaptation of Electronically Scanned Pressure Measurement Technology at Langley Research Center, " *Instrumentation in the Aerospace Industry, Proceedings of the Thirty-Sixth International Instrumentation Symposium*, Vol. 36, May, 1990, pp. 493-516.

126. Moravec, H. P., "Sensor Fusion in Certainty Grids for Mobile Robots," *Artificial Intelligence Magazine*, Summer, 1988, pp. 61-74.

127. Moravec, H. P. and Elfes, A., "High Resolution Maps from Wide Angle Sonar," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1985, pp. 116-121.

128. Morgenthaler, D. G., Hennessy, S. and DeMenthon, D., "Range-Video Fusion and Comparison of Inverse Perspective Algorithms in Static Images," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 6, November-December, 1990, pp. 1301-1312.

129. Morita, T., Takahasi, H., Satonobu, J., Kojima, K. and Morimoto, Y., "An Approach to the Intelligent Vehicle," *1993 International Vehicle Symposium,* 1993, pp. 426-431.

130. Muller, N., "Feedforward Control for Curve Steering for an Autonomous Road Vehicle," *Proceedings 1992 IEEE International Conference on Robotics and Automation*, Vol. 1, May, 1992, pp. 200-205.

131. Murphy, R. R., "An Artificial Intelligence Approach to the 1994 AUVS Unmanned Ground Robotics Competition," *IEEE International Conference of Systems, Man, and Cybernetics,* October, 1995, pp. 1723-1728.

132. Murphy, R. R., Hoff, W., Blitch, J., Gough, V., Hawkins, D., Hoffman, J. C., Krosley, R., Lyons, T., Mali, A., MacMillan, J. and Warshawsky, S., "Colorado School of Mines Behavioral Approach to the 1995 UGR Competition," *Mobile Robots X, SPIE*, Vol. 2591, 1995, pp. 220-227.

133. Murphy, R. R., "A Control Scheme for Sensor Fusion for Navigation of Autonomous Mobile Robots,", *Sensor Fusion III: 3-D Perception and Recognition, SPIE*, Vol. 1383, November, 1990, pp. 436-447.

134. Nagy, P. V. and Bock, T., "The Northern Illinois University Autonomous Mobile Robots," *Mobile Robots X, SPIE*, Vol. 2591, 1995, pp. 209-219.

135. Narendra, K. S. and Parathasathy, K., "Identification and Control of Dynamical Systems using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4-27.

136. Ninomiya, Y., Matsuda, S., Ohta, M. and Harata, Y., "A Real-Time Vision for Intelligent Vehicles," *1995 International Vehicle Symposium,* 1995, pp. 315-320.

137. Nowinski, G., Traechtler, A. and Lafuente, R., "Design Process for Conventional and Neural Controllers - A Case Study," *IEE Conference* Publication, No. 395, 1994, pp. 159-164.

138. Olin, K. E. and Tseng, D. Y., "Autonomous Cross-Country Navigation: An Integrated Perception and Planning System," *IEEE Expert*, Vol. 6, No. 4, August, 1991, pp. 16-30.

139. Pandya, A. S., and Luebbers, P. G., "Neural Networks for Robot Navigation," *Applications of Artificial Intelligence IX, SPIE*, Vol. 1468, part 2, April, 1991, pp. 802-811.

140. Passino, K. M., "Intelligent Control for Autonomous Systems," *IEEE Spectrum*, Vol. 32, No. 6, June, 1995, pp. 55-62.

141. Payton, D. W., Rosenblatt, J. K., and Keirsey, D. M., "Plan Guided Reaction," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 6, November-December, 1990, pp. 1370-1382.

142. Pears, N. and Probert, P., "Optical Range Sensor for Mobile Robot Guidance," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, 1993, pp. 659-664.

143. Penna, M. A. and Wu, J., "Some Artificial Neural Networks for Visual Navigation," *Journal of Intelligent Material Systems and Structures*, Vol. 5, No. 2, March, 1994, pp. 220-231.

144. Pin, F. G., Watanabe, H., Symon, J. and Pattay, R. S., "Autonomous Navigation of a Mobile Robot Using Custom-Designed Qualitative Reasoning VLSI chips and boards," *Proceedings 1992 IEEE International Conference on Robotics and Automation*, Vol. 1, May, 1992, pp. 123-128.

145. Pomerleau, D. A., "Neural-Network-Based Vision Processing for Autonomous Robot Guidance," *Applications of Artificial Neural Networks II, SPIE*, April, 1991, pp. 121-128.

146. Prabler, E. and Milios, E., "Parallel Path Planning in Unknown Terrains," *Mobile Robots V, SPIE,* Vol. 1388, 1990, pp. 2-13.

147. Raschke, U. and Borenstein, J., "A Comparison of Grid-type Map-Building Techniques by Index of Performance," *IEEE International Conference on Robotics and Automation,* Vol. 2, 1990, pp. 1828-1832.

148. Reid, M. B., "Path Planning Using Optically Computed Potential Fields," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, 1993, pp. 295-300.

149. Ringach, D. L. and Baram, Y., "Diffusion Mechanism for Obstacle Detection From Size-Change Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 1, January, 1994, pp. 76-80.

150. Roberts, B. and Bhanu, B., "Inertial Navigation Sensor Integrated Motion Analysis for Autonomous Vehicle Navigation," *Journal of Robotic Systems*, Vol. 9, No. 6, September, 1992, pp. 817-842.

151. Rocheleau, D. N. and Crane, C. D. III, "Development of a Graphical Interface for Robotic Operation in a Hazardous Environment," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, October, 1991, pp. 1077-1081.

152. Rogers, P. J., "Optics in Hostile Environments," *Specification and Measurement of Optical Systems*, SPIE, Vol. 1781, 1992, pp. 36-48.

153. Ruocco, S. R., *Robot Sensors and Transducers*, Halsted Press: John Wiley & Sons, New York, 1987.

154. Saffiotti, A., Ruspini, E. H. and Konolige, K., "Blending Reactivity and Goal-Directedness in a Fuzzy Controller," *Second IEEE International Conference on Fuzzy* Systems, 1993, pp. 134-139.

155. Salem, F. A. and El-Khamy, S. E., "A New Accurate Radar Sensor Incorporating Matched Frequency Hopping Spread Spectrum (MFH/SS) Technique for Mobile Robot Target Identification," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 170-181.

156. Saneyoshi, K., "3-D Image Recognition System by Means of Stereoscopy Combined With Ordinary Image Processing," *Intelligent Vehicles Symposium, Proceedings*, 1994, pp. 13-18.

157. Santamaria, J. C. and Arkin, R. C., "Structured Light Systems for Dent Recognition: Lessons Learned," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 240-259.

158. Santos-Victor, J., Sandini, G., Curotto, F. and Garibaldi, S., "Divergent Stereo in Autonomous Navigation: From Bees to Robots," *International Journal of Computer Vision,* Vol. 14, No. 2, March 1995, pp. 159-177.

159. ----- "Divergent Stereo for Robot Navigation: Learning From Bees," *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1993, PP. 434-439.

160. Schalit, E., "ARCANE: Towards Autonomous Navigation on Rough Terrains," *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Vol. 3, May, 1992, pp. 2568-2574.

161. Schneider, F. E. and Wolf, H.-L., "Acquisition and Display of a World Model Using an Autonomous Mobile Land Robot," *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems,* Vol. 2, 1994, pp. 769-775.

162. Selfridge, O. G. and Franklin, J. A., "The Perceiving Robot: What Does It See? What Does It Do?" *Proceedings of the 5th IEEE International Symposium on Intelligent Control,* September, 1990, pp. 146-151.

163. Sethi, I. K. and Yu, G., "A Neural Network Approach to Robot Localization Using Ultrasonic Sensors," *Proceedings of the 5th IEEE International Symposium on Intelligent Control,* September, 1990, pp. 513-517.

164. Shahidi, R., Shayman, M. and Krishnaprasad, P. S., "Mobile Robot Navigation Using Potential Functions," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Vol. 3, April, 1991, pp. 2047-2053.

165. Sharma, R., Mount, D. M., and Aloimonos, Y., "Navigation in a Hazardous Environment With Distributed Shelters," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, October, 1991, pp. 883-888.

166. Shiller, Z. and Gwo, Y., "Dynamic Motion Planning of Autonomous Vehicles," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 2, April, 1991, pp. 241-249.

167. Shin, D. H., Singh, S. and Shi, W., "A Partitioned Control Scheme for Mobile Robot Path Tracking," *IEEE International Conference on Systems Engineering*, August, 1991, pp. 338-342.

168. Shirley, P. A., "Introduction to Ultrasonic Sensing," *Sensors*, Vol. 6, No. 11, November, 1989, pp. 10-17.

169. Siemiatkowska, B., "Highly Parallel Method for Mapping and Navigation of an Autonomous Mobile Robot," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, Part 4, 1994, pp. 2796-2801

170. Simpson, R. L. Jr., "Computer Vision: An Overview," *IEEE Expert*, Vol. 6, No. 4, August, 1991, pp. 11-15.

171. Slack, M. G., "Navigation Templates: Mediating Qualitative Guidance and Quantitative Control in Mobile Robots," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 23, No. 2, March/April 1993, pp. 452-466.

172. ----- "Coordinating Sensing and Local Navigation," *Sensor Fusion III: 3-D Perception and Recognition," SPIE*, Vol. 1383, November, 1990, pp. 459-470.

173. Slifka, A. J., Chaudhuri, D. K., Compos, R. and Siegwarth, J. D., "Tribometer for Measurements in Hostile Environments," *Wear,* Vol. 170, No. 1, November, 1993, pp. 39-44.

174. Smith, D. E. and Starkey, J. M., "Overview of Vehicle Models, Dynamics, and Control Applied to Automated Vehicles," *Advanced Automotive Technologies*, Vol. 40, December, 1991, pp. 69-87.

175. Smith, F. M., Backman, D. K. and Jacobsen, S. C., "Telerobotic Manipulator for Hazardous Environments," *Journal of Robotic Systems*, Vol. 9, No. 2, March, 1992, pp. 251-260.

176. Stentz, A. and Hebert, M., "Complete Navigation System for Goal Acquisition in Unknown Environments," *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1, 1995, pp. 425-432.

177. Stentz, A., Brumitt, B. L., Coulter, R. C. and Kelly, A., "An Autonomous System for Cross-Country Navigation," *Mobile Robots VII, SPIE,* Vol. 1831, 1993, pp. 540-551.

178. Stuck, E. R., Manz, A., Green, D. A. and Elgazzar, S., "Map Updating and Path Planning for Real-Time Mobile Robot Navigation," *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Vol. 2, 1994, pp. 753-760.

179. Sugiyama, M., Kawano, Y., Niizuma, M., Takagaki, M., Tomizawa, M. and Degawa, S., "Navigation System for an Autonomous Vehicle With Hierarchical Map and Planner," *Intelligent Vehicles Symposium, Proceedings*, 1994, pp. 50-55.

180. Sung, E., "Stereovision and Colour Segmentation for Autonomous Navigation," *Mobile Robots V, SPIE*, Vol. 1388, November, 1990, pp. 176-187.

181. Talluri, R. and Aggarawal, J. K., "Image/Map Correspondence for Mobile Robot Self-Location Using Computer Graphics," *IEEE Transactions on Pattern Analysis and* Machine Intelligence, Vol. 15, No. 6, June, 1993, pp. 597-601.

182. Taylor, C. J. and Kriegman, D. J., "Structure and Motion Form Line Segments in Multiple Images," *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Vol. 2, May, 1992, pp. 1615-1620.

183. Thorpe, C., Hebert, M., Kanade, T. and Shafer, S., "Toward Autonomous Driving: The CMU Navlab--I: Perception," *IEEE Expert*, Vol. 6, No. 4, August, 1991, pp. 31-42.

184. Tsukamoto, M., Inui, N., Kumeda, K., Kamei, K. and Inoue, K., "Linguistic Topographical Representation for Navigation of Autonomous Robot," *Proceedings of the 1992 Japan- USA Symposium on Flexible Automation*, July, 1992, pp. 1329-1334.

185. Tunstel, E. and Jamshidi, M., "Fuzzy Logic and Behavior Control Strategy for Autonomous Mobile Robot Mapping," *IEEE International Conference on Fuzzy Systems*, Vol. 1, 1994, pp. 514-517.

186. van Dam, J. W. M., Kroese, B. J. A. and Groen, F. C. A., "Transforming the Ego-Centered Internal Representation of an Autonomous Robot With the Cascaded Neural Network," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1994, pp. 667-674.

187. Vandorpe, J. and Van Brussel, H., "A Reflexive Navigation Algorithm for an Autonomous Mobile Robot," *Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI '94),* October 1994, pp. 251-258.

188. Vasseur, H. A., Pin, F. G. and Taylor, J. R., "Navigation of Car-Like Mobile Robots in Obstructed Environments Using Convex Polygonal Cells," *Robotics and Autonomous Systems,* Vol. 10, No. 2-3, 1992, pp. 133-146.

189. Veelaert, P. and Peremans, H., "Situation Recognition for Behavior Based Mobile Robots," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 32-42.

190. Visinsky, M. L., Walker, I. D. and Cavallaro, J. R., "Layered Dynamic Fault Detection and Tolerance for Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, 1993, pp. 180-187.

191. Warren, C. W., "A Vector Based Approach to Robot Path Planning," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Vol. 2, April, 1991, pp. 1021-1026.

192. *Webster's Ninth New Collegiate Dictionary*, Merriam-Webster Inc., Springfield, MA, 1989.

193. Weigl, M., Siemiatkowska, G., Sikorski, K. A. and Borkowski, A., "Grid-Based Mapping for Autonomous Mobile Robot," *Robotics and Autonomous Systems*, Vol. 11, No. 1, May, 1993, pp. 13-21

194. Wright, B. D. and Simeon, T., "Free Space Representation for a Mobile Robot Moving on a Rough Terrain," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, 1993, pp. 37-43.

195. Wu, H., Chang, W., Hangen, H. E. and Xi, P., " A Fuzzy Control Method for Lateral Control of Autonomous Land Vehicle," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 125-131.

196. Xie, M., Trassoudaine, L., Alizon, J. and Gallice, J., "Road Obstacle Detection and Tracking by an Active and Intelligent Sensing Strategy," *Machine Vision and Applications 7 3,* 1994, pp. 165-177.

197. Xu, H., "Efficient Fusion Technique for Disparate Sensory Data," *Proceedings of the 1991 International Conference on Industrial Electronics, Control and Instrumentation - IECON '91*, Vol. 3, October-November, 1991, pp. 2535-2540.

198. Yang, J.-Y. and Wu, Y.-G., "Detection for Mobile Robot Navigation Based on Multisensor Fusion," *Mobile Robots X, SPIE,* Vol. 2591, 1995, pp. 182-192.

199. Yeung, S. K., Mc Math, W. S., Korba, L., Elgazzar, S., Petriu, E. M., Gal, C. and Dancea, I. C., "Multi-Sensor System for Mobile Robot Navigation," *1993 International Vehicle Symposium,* 1993, pp. 455-460.

200. Yializis, A., Powers, G. L. and Shaw, D. G., "A New High Temperature Multilayer Capacitor With Acrylate Dielectrics," *IEEE Transactions on Components, Hybrids and Manufacturing Technology*, Vol. 13, No. 4, December, 1990, pp. 611-616.

201. Yokoyama, T., Tachibana, A., Suzuki, T. and Inoue, H., "Automated Vehicle System Using Both a Computer Vision and Magnetic Field Sensors," *1993 International Vehicle Symposium,* 1993, pp. 157-162.

202. Zadeh, L., "Outline of a New Approach to the Analysis of Complex Systems and Decision Process", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3, 1973, pp. 207-219.

203. Zhang, J. and Raczkowsky, J., "Robust Subgoal Planning and Motion Execution for Robots in Fuzzy Environments," *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Vol. 1, 1994, pp. 447-453.

204. Zhang, Q., Jiang, R., Liu, K. and Jing, Y., "Analysis of Color and Range Image Using PDS," *Mobile Robots VII, SPIE*, Vol. 1831, 1993, pp. 156-164.

205. Zhao, G. W. and Yuta, S., "Obstacle Detection by Vision System for an Autonomous Vehicle," *1993 International Vehicle Symposium,* 1993, pp. 31-36.

206. Zhao, Y. and BeMent, S. L., "Kinematics, Dynamics and Control of Wheeled Mobile Robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, May, 1992, pp. 91-96.

207. ----- "Fast Boundary Extraction for Mobile Robot Sensor-Detected Images," *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, September, 1990, pp. 1050-1055. (1990a)

208. ----- "A Heuristic Search Approach for Mobile-Robot Trap Recovery," *Mobile Robots V, SPIE*, Vol. 1388, November, 1990, pp. 122-130. (1990b)

# Vita

Susan Marie Larkin was born on August 26, 1972 in Welch, West Virginia. She attended Mount View High School and was the Valedictorian of the 1990 graduating class. Susan entered West Virginia University as a University Presidential Scholar in 1990. She earned her Bachelor of Science Degree in Mechanical Engineering in May, 1994 as a Magna Cum Laude graduate. Susan then went on to attend Virginia Polytechnic Institute and State University in the fall of 1994. She graduated with a Master of Science Degree in Mechanical Engineering in June of 1996. Susan began her career with Lucent Technologies, Bell Labs Innovations in Dallas, Texas.